亚马逊云科技

用户指南

Amazon Elastic Compute Cloud

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Elastic Compute Cloud: 用户指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅 中国的 Amazon Web Services 服务入门 (PDF)。

Table of Contents

什么是 Amazon EC2？ 1
功能 1
相关服务 2
访问 EC2 3
定价 4
估算、账单和成本优化 5
资源 5
入门教程 7
步骤 1：启动实例 8
步骤 2：连接到您的实例 9
步骤 3：清除您的实例 12
后续步骤 13
最佳实践 14
亚马逊机器映像 16
AMI 特征 17
启动许可 17
根卷类型 18
识别 AMI 根卷类型 19
虚拟化类型 20
查找 AMI 22
Systems Manager 参数 25
Systems Manager 公共参数 28
Amazon Web Services Marketplace 中的付费 AMI 30
在 Amazon Web Services Marketplace 中销售您的 AMI 31
查找付费 AMI 31
购买付费 AMI 33
检索产品代码 34
使用付费支持 35
付费和支持 AMI 的账单 36
管理您的 订阅 36
AMI 周期 37
创建 AMI 38
创建 Amazon S3 支持的 AMI 46
使用 Windows Sysprep 创建 AMI 83

复制 AMI 98
存储和还原 AMI 113
AMI 世系 122
AMI 水印 127
AMI 使用情况 133
弃用 AMI 156
禁用 AMI 161
取消注册 AMI 167
启动模式 173
UEFI 启动模式的要求 174
AMI 启动模式参数 176
实例类型启动模式 177
实例启动模式 180
操作系统启动模式 182
设置 AMI 启动模式 183
UEFI 变量 189
UEFI 安全启动 190
AMI 加密 204
启动实例场景 205
映像复制场景 207
共享 AMI 209
经过验证的提供商 209
查找共享 AMI 210
准备使用适用于 Linux 的共享 AMI 213
允许的 AMI 214
将您的 AMI 设为公用 233
阻止 AMI 公开访问 236
与组织和组织单位共享 AMI 246
与特定 Amazon 账户共享 AMI 257
取消与您的账户共享 AMI 260
创建共享 Linux AMI 的建议 261
监控 AMI 事件 267
事件详细信息 268
available 事件 269
failed 事件 269
deregistered 事件 270

disabled 事件 ... 271
了解 AMI 账单 ... 271
AMI 账单字段 ... 272
查找 AMI 账单信息 ... 274
验证账单上的 AMI 费用 ... 276
AMI 限额 ... 276
请求增加 AMI 的限额 ... 277
实例 ... 279
实例类型 ... 280
可用实例类型 ... 281
硬件规格 ... 281
虚拟机监控程序类型 ... 281
AMI 虚拟化类型 ... 282
处理器 ... 283
查找实例类型 ... 285
EC2 实例类型查找器 ... 292
Compute Optimizer 建议 ... 294
实例类型更改 ... 296
可突增性能实例 ... 303
GPU 实例 ... 349
EBS 优化 ... 362
CPU 选项 ... 487
AMD SEV-SNP ... 775
处理器状态控制 ... 782
托管式实例 ... 786
托管式实例的账单 ... 787
确定托管式实例 ... 787
托管资源可见性设置 ... 789
开始使用托管式实例 ... 792
嵌套虚拟化 ... 793
工作原理 ... 793
注意事项 ... 793
在启用了嵌套虚拟化的情况下启动新实例 ... 794
将现有实例配置为使用嵌套虚拟化 ... 795
计费和购买选项 ... 796
按需型实例 ... 796

预留实例 799
Spot Instances 857
专属主机 940
Dedicated Instances 982
Capacity Reservations 988
启动模板 1111
限制 1112
权限 1112
控制启动实例 1119
创建 1120
修改（管理版本） 1134
删除 1138
启动实例 1140
教程 1142
实例参数参考 1162
使用启动实例向导启动 1172
使用启动模板启动 1175
从现有实例启动 1181
从 Amazon Web Services Marketplace AMI 启动 1182
连接到您的实例 1184
通用连接先决条件 1186
使用 SSH 连接到 Linux 实例 1191
使用 RDP 连接到 Windows 实例 1205
使用会话管理器进行连接 1214
使用公有 IP 和 EC2 Instance Connect 进行连接 1214
使用私有 IP 和 EC2 Instance Connect Endpoint 进行连接 1245
实例状态更改 1274
按实例状态计费 1275
待处理实例 1276
已停止的实例 1276
休眠实例 1277
重启实例 1277
终止实例 1278
实例状态之间的区别 1278
停止和启动 1280
Hibernate 1298

Reboot 1325
终止 1327
停用 1342
实例自动恢复 1346
实例自动恢复的关键概念 1347
简化的自动恢复机制和基于 CloudWatch 操作的恢复机制之间的区别 1349
构建弹性系统 1350
验证是否已进行自动恢复 1350
简化的自动恢复 1352
CloudWatch 基于操作的恢复 1356
实例元数据 1360
实例元数据类别 1361
动态数据类别 1371
访问实例元数据 1372
配置 IMDS 选项 1409
启动时运行命令 1443
示例：AMI 启动索引值 1467
检测主机是否为 EC2 实例 1471
检查 实例身份文档 1471
检查系统 UUID 1472
检查系统虚拟机生成标识符 1473
实例身份文档 1478
检索实例身份文档 1479
验证实例身份文档 1481
公有证书 1491
STIG 合规性 1495
STIG 设置 1496
下载 STIG 脚本 1507
使用 Systems Manager 应用 STIG 设置 1529
时钟同步 1531
硬件数据包时间戳 1532
闰秒 1533
使用本地 Amazon Time Sync Service 1533
使用公共 Amazon Time Sync Service 1545
比较您 Linux 实例的时间戳 1547
更改实例的时区 1548

管理设备驱动程序 1550
网络驱动程序 1551
显卡驱动程序 1551
存储设备驱动程序 1551
AMD 驱动程序 1552
NVIDIA 驱动程序 1558
在 Windows 上安装 ENA 驱动程序 1601
Windows PV 驱动程序 1618
Amazon NVMe 驱动程序 1646
配置 Windows 实例 1656
Windows 特定的系统设置 1656
适用于 Windows 实例的 Amazon 设备驱动程序 1657
Windows 启动代理 1658
适用于 Windows 的 EC2 快速启动 1797
更改 Windows 管理员密码 1817
添加 Windows 系统组件 1819
在 Windows 上安装 WSL 1823
Windows 实用程序 1825
升级 Windows 实例 1826
执行就地升级 1827
执行自动升级 1831
迁移到基于 Nitro 的实例类型 1840
排查升级问题 1848
教程：将 EC2 实例连接到 RDS 数据库 1849
教程目标 1849
上下文 1849
架构 1849
注意事项 1851
完成教程所需的时间 1851
成本 1851
选项 1：使用 EC2 控制台自动连接 1852
选项 2：使用 RDS 控制台自动连接 1862
选项 3：手动连接 1871
实例集 1881
功能和优势 1881
使用哪种实例集方法？ 1882

配置选项 1883
请求类型 1884
支出限额 1911
基于属性的实例类型选择 1913
实例权重 1946
分配策略 1949
容量再平衡 1954
容量预留 1958
使用 EC2 实例集 1959
EC2 实例集请求状态 1959
EC2 实例集先决条件 1960
创建 EC2 实例集 1964
标记 EC2 实例集 1973
描述 EC2 实例集 1974
修改 EC2 实例集 1979
删除 EC2 实例集 1981
使用竞价型实例集 1985
竞价型实例集请求状态 1986
竞价型实例集权限 1987
创建竞价型实例集 1997
标记竞价型实例集 2005
描述竞价型实例集 2014
修改竞价型实例集请求 2015
取消（删除）竞价型实例集请求 2017
竞价型实例集的自动扩展 2020
监控实例集 2029
使用 CloudWatch 监控实例集 2029
使用 EventBridge 监控您的实例集 2032
教程 2049
教程：将 EC2 实例集配置为使用实例权重 2051
教程：将 EC2 实例集配置为使用按需型实例作为主容量 2054
教程：使用目标容量预留配置 EC2 实例集以启动按需型实例 2055
教程：将 EC2 实例集配置为将实例启动到容量块中 2061
教程：将 EC2 实例集配置为将实例启动到可中断容量预留中 2063
EC2 实例集 CLI 配置示例 2068
示例 1：启动竞价型实例作为默认购买选项 2069

示例 2：启动按需型实例作为默认购买选项 2069
示例 3：启动按需型实例作为主容量 2070
示例 4：使用多个容量保留按需启动实例 2070
示例 5：当总目标容量大于未使用的容量预留数量时，使用容量预留启动按需实例 2074
示例 6：使用容量预留启动按需实例 2077
示例 7：配置容量再平衡以启动替换竞价型实例 2080
示例 8：在容量优化的队列中启动竞价型实例 2082
示例 9：在具有优先级的容量优化队列中启动竞价型实例 2083
示例 10：在价格容量优化的实例集中启动竞价型实例 2084
示例 11：配置基于属性的实例类型选择 2085
竞价型实例集 CLI 配置示例 2087
示例 1：价格最低的可用区 2087
示例 2：列表中价格最低的可用区 2088
示例 3：列表中价格最低的实例类型 2090
示例 4。覆盖请求的价格 2092
示例 5：使用多样化分配策略 2093
示例 6：使用实例权重 2096
示例 7：使用按需容量 2098
示例 8：配置容量再平衡 2098
示例 9：使用容量优化的实例集 2100
示例 10：使用具有优先级的容量优化实例集 2101
示例 11：使用 priceCapacityOptimized 实例集 2102
示例 12：配置基于属性的实例类型选择 2103
队列配额 2104
请求增加目标容量配额 2105
联网 2107
区域和可用区 2108
Regions 2108
可用区 2109
Local Zones 2110
Wavelength 区域 2111
Amazon Outposts 2113
实例 IP 寻址 2114
私有 IPv4 地址 2115
公有 IPv4 地址 2116
公有 IPv4 地址优化 2117

IPv6 地址 2118
多个 IP 地址 2119
EC2 实例主机名 2120
链路本地地址 2120
IPv4 地址 2121
IPv6 地址 2126
辅助 IP 地址 2132
Windows 上的 IPv4 地址 2135
EC2 实例主机名和域 2143
了解 EC2 实例主机名和域 2143
主机名类型 2144
自带 IP 地址 2157
BYOIP 定义 2158
要求和配额 2159
先决条件 2160
载入您的地址范围 2167
使用您的地址范围 2176
弹性 IP 地址 2177
弹性 IP 地址定价 2177
弹性 IP 地址基本信息 2177
弹性 IP 地址配额 2178
关联弹性 IP 地址 2179
转移弹性 IP 地址 2184
释放弹性 IP 地址 2189
将反向 DNS 用于电子邮件应用程序 2190
网络接口 2194
网络接口概念 2196
网卡 2197
每个网络接口的 IP 地址数 2201
创建网络接口 2203
网络接口连接 2208
管理 IP 地址 2211
修改网络接口属性 2215
多个网络接口 2219
请求者托管的网络接口 2221
前缀委派 2223

删除网络接口 2233
网络带宽 2234
可用实例带宽 2235
带宽权重 2237
监控实例带宽 2271
增强联网 2271
弹性网络适配器（ENA） 2272
ENA Express 2345
Intel 82599 VF 2383
监控网络性能 2394
改善 Linux 上的网络延迟 2404
Nitro 性能注意事项 2407
优化 Windows 上的网络性能 2413
Elastic Fabric Adapter 2414
EFA 基础知识 2415
支持的接口和库 2417
支持的实例类型 2418
支持的操作系统 2434
EFA 限制 2435
EFA 定价 2435
EFA 和 MPI 入门 2435
EFA 和 NCCL 入门 2453
开始使用 EFA 和 NCCL 2475
最大化网络带宽 2495
创建并附加 EFA 2509
分离并删除 EFA 2513
监控 EFA 2514
验证 EFA 安装程序 2521
发行说明 2535
EC2 拓扑 2556
工作原理 2557
先决条件 2563
示例 2565
置放群组 2580
置放策略 2581
创建置放群组 2585

更改实例置放 2587
删除置放群组 2588
共享置放群组 2589
Amazon Outposts 上的置放群组 2591
网络 MTU 2592
巨型帧 (9001 MTU) 2593
路径 MTU 发现 2594
设置实例的 MTU 2595
故障排除 2600
Virtual Private Cloud 2601
您的默认 VPC 2601
非默认 VPC 2602
互联网访问 2603
共享子网 2603
仅限 IPv6 子网 2604
辅助网络 2604
什么是辅助网络？ 2604
重要概念 2605
架构 2605
其他注意事项 2605
开始使用 2605
管理辅助网络资源 2607
网络设计最佳实践 2608
问题排查 2609
限额和限制 2610
安全性 2611
数据保护 2612
Amazon EBS 数据安全 2613
静态加密 2613
传输中加密 2614
基础设施安全性 2616
网络隔离 2616
物理主机上的隔离 2616
控制网络流量 2617
故障恢复能力 2618
合规性验证 2619

Identity and Access Management 2620
基于身份的策略 2620
API 的示例策略 2630
针对控制台的示例策略 2673
Amazon 托管策略 2684
IAM 角色 2691
更新管理 2702
Windows 实例的最佳实践 2702
高级安全最佳实践 2702
更新管理 2703
配置管理 2705
变更管理 2706
Amazon EC2 Windows 实例的审计和问责 2706
密钥对 2707
创建密钥对 2708
描述密钥对 2714
删除您的密钥对 2719
在 Linux 实例上添加或更换公有密钥 2720
验证指纹 2722
安全组 2724
概述 2725
创建安全组 2725
更改实例的安全组 2727
删除安全组 2733
连接跟踪 2734
针对不同使用案例的安全组规则 2739
Amazon PrivateLink 2745
创建接口 VPC 端点 2745
创建端点策略 2745
仓储服务 2748
Amazon 存储定价 2749
Amazon EBS 2749
EBS 卷限制 2750
EBS 卡 2753
Amazon EC2 实例存储 2755
数据持久性 2756

实例存储卷限制 2758
SSD 实例存储卷 2762
添加实例存储卷 2766
为 M1 和 C1 实例启用交换卷 2772
初始化实例存储卷 2776
实例存储卷的详细性能统计数据 2777
根卷 2780
具有 Amazon EBS 根卷的实例 2781
具有实例存储根卷的实例（仅限 Linux 实例） 2782
实例终止后保留根卷 2783
替换根卷 2787
卷的设备名称 2796
可用设备名称 2796
设备名称注意事项 2799
块设备映射 2800
块储存设备映射的概念 2800
将块设备映射添加到 AMI 2804
将块设备映射添加到实例 2807
如何为 Windows 实例附加和映射卷 2815
将 NVME 磁盘映射到卷 2816
将非 NVME 磁盘映射到卷 2821
撕裂写防护 2830
支持的块大小 2831
要求 2831
检查实例支持 2832
配置工作负载 2833
Windows VSS EBS 快照 2834
什么是 VSS？ 2835
基于 VSS 的 Amazon EBS 快照解决方案工作原理 2836
VSS 先决条件 2837
创建 VSS 快照 2848
排查 VSS 快照问题 2857
Amazon VSS 解决方案的恢复选项 2860
版本历史记录 2861
对象存储、文件存储及文件缓存 2865
Amazon S3 2866

Amazon EFS 2870
Amazon FSx 2872
Amazon File Cache 2877
管理 资源 2879
为您的资源选择区域 2879
查找资源 2881
控制台步骤 2881
命令行示例 2889
全局视图（跨区域） 2892
Amazon 全局视图 2892
标记资源 2896
有关标签的基本知识 2896
标记 资源 2897
标签限制 2898
标签和访问管理 2899
为资源添加标签以便于进行计费 2899
资源标记权限 2900
添加或删除标签 2903
按标签筛选资源 2907
使用实例元数据来查看标签 2908
Service Quotas 2914
查看当前限额 2914
请求提高 2914
对使用端口 25 发送的电子邮件的限制 2915
监控 资源 2916
监控实例状态 2917
状态检查 2917
状态更改事件 2925
计划的事件 2927
使用 CloudWatch 监控您的实例 2959
实例警报 2960
管理详细监控 2961
CloudWatch 指标 2964
安装和配置 CloudWatch 代理 2978
指标的统计数据 2982
查看监控图表 2991

创建警报 2991
创建停止、终止、重启或恢复实例的警报 2992
使用 EventBridge 自动执行 3004
Amazon EC2 事件类型 3004
使用 CloudTrail 记录 API 调用 3005
CloudTrail 中的 Amazon EC2 API 管理事件 3006
Amazon EC2 API 事件示例 3006
审核使用 EC2 Instance Connect 建立的连接 3008
监控 .NET 和 SQL Server 应用程序 3009
账单和使用情况报告 3010
实例 3011
裸机实例 3011
专属主机 3012
Dedicated Instances 3012
EBS 优化 3012
容量预留 3013
清点您的 EC2 实例 3013
地理位置 3014
硬件规格 3015
映像 3016
IP 地址和 DNS 主机名 3018
安全组规则 3019
用户数据 3021
相关资源 3022
故障排除 3023
实例启动问题 3023
设备名称无效 3024
超出实例限制 3025
实例容量不足 3025
当前不支持请求的配置。请查看文档以了解支持的配置。 3026
实例立即终止 3026
权限不足 3027
Windows 启动后，CPU 使用率短时增高（仅限 Windows 实例） 3028
启动启用 IMDSv1 的实例时失败 3029
实例停止问题 3030
强制停止实例 3031

（可选）创建替代实例 3032
实例终止问题 3035
实例立即终止 3035
延迟的实例终止 3035
已终止实例仍然显示 3037
错误：此实例可能无法终止。修改其“disableApiTermination”实例属性 3037
自动启动或终止的实例 3037
无法访问的实例 3038
实例重启 3038
实例控制台输出 3038
捕获无法访问的实例的屏幕截图 3039
Windows 实例的常见屏幕截图 3041
主机发生故障时的实例恢复 3049
实例出现脱机后又意外重启 3049
Linux 实例 SSH 问题 3050
连接问题的常见原因 3050
连接到您的实例时出错：连接超时 3053
Error: unable to load key ...(错误：无法加载密钥...) Expecting: ANY PRIVATE KEY(需要：任何私有密钥) 3055
错误：服务器无法识别用户密钥 3056
错误：权限被拒绝或 [实例] 端口 22 关闭了连接 3058
错误：未保护的私有密钥文件 3060
错误：私有密钥的格式必须以“-----BEGIN RSA PRIVATE KEY-----”开头，以“-----END RSA PRIVATE KEY-----”结尾 3061
错误：主机密钥验证失败 3061
错误：服务器拒绝我们的密钥或 没有支持的身份验证方法 3062
无法对实例执行 Ping 操作 3063
错误：服务器意外关闭了网络连接 3063
错误：EC2 Instance Connect 的主机密钥验证失败 3063
无法使用 EC2 Instance Connect 连接到 Ubuntu 实例 3065
我丢失了私有密钥。如何连接到我的实例？ 3065
Linux 实例失败状态检查 3071
查看状态检查信息 3072
检索系统日志 3073
排查 Linux 的实例的系统日志错误 3073
内存不足：终止进程 3074

错误：mmu_update 失败 (内存管理更新失败) 3075
I/O 错误（块储存设备故障） 3076
I/O 错误：既不是本地磁盘也不是远程磁盘（破损的分布式块储存设备） 3078
request_module：runaway loop modprobe (在较旧的 Linux 版本上循环旧内核 modprobe) 3079
“严重错误：内核太旧”和“fsck：在尝试打开 /dev 时没有此文件或目录”(内核与 AMI 不匹配) 3080
“FATAL: Could not load /lib/modules”或者“BusyBox”(内核模块缺失) 3081
ERROR：无效内核 (EC2 不兼容内核) 3083
fsck：尝试打开时没有找到此文件或目录... (未找到文件系统) 3084
挂载文件系统时出现一般性错误（挂载失败） 3086
VFS：无法在未知块上挂载根 fs (根文件系统不匹配) 3088
错误：无法确定根设备的主/次编号…（根文件系统/设备不匹配） 3089
XENBUS：设备没有驱动程序… 3090
… 没有检查时，已强制执行检查的工作日 (文件系统检查要求) 3092
fsck 卡在退出状态...（设备缺失） 3092
GRUB 提示 (grubdom>) 3093
提起接口 eth0：设备 eth0 的 MAC 地址与预期不同，驳回。(硬编码的 MAC 地址)。 3096
无法加载 SELinux 策略。计算机处于强制执行模式。正在中断。(SELinux 配置错误) 3097
XENBUS：连接设备时超时 (Xenbus 超时) 3098
Linux 实例从错误的卷启动 3099
Windows 实例 RDP 问题 3101
远程桌面无法连接到远程计算机 3101
使用 macOS RDP 客户端时出错 3103
RDP 显示黑屏而不是桌面 3103
无法使用非管理员用户远程登录到实例 3103
使用 Amazon Systems Manager 解决远程桌面问题 3103
在具有远程注册表的 EC2 实例上启用远程桌面 3107
我丢失了私有密钥。我怎样才能连接到我的 Windows 实例？ 3109
Windows 实例的启动问题 3109
"密码不可用" 3109
"密码尚不可用" 3110
"无法检索 Windows 密码" 3110
"等待元数据服务" 3111
"无法激活 Windows" 3114
"Windows 不是正版 (0x80070005)" 3116
"没有终端服务器许可服务器可提供许可证" 3116
“某些设置由您的组织管理” 3117

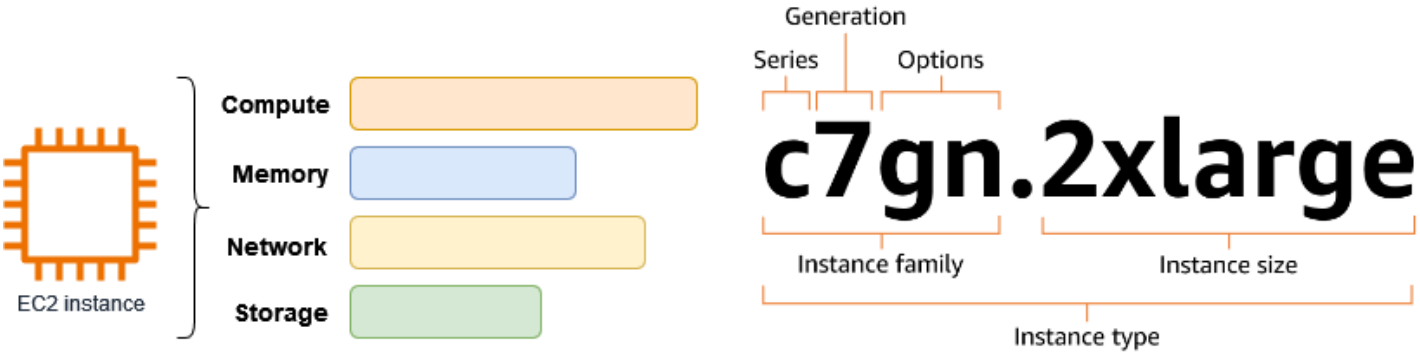
Windows 实例问题 3117
无法将 Amazon Systems Manager 会话管理器连接到 Windows Server 2025 实例 3118
EBS 卷在 Windows Server 2016 和 2019 上不初始化 3118
将 EC2 Windows 实例启动至目录服务还原模式 (DSRM) 3119
实例失去网络连接或计划的任务不按预期方式运行 3122
无法获取控制台输出 3122
网络上不可用的 Windows Server 2012 R2 3123
磁盘签名冲突 3123
通过网络对 Windows 实例进行内核调试 3124
前提条件 3125
步骤 1：在调试主机上安装 Windows 调试工具 3126
步骤 2：设置调试目标 3126
步骤 3：在调试主机上启动调试会话 3128
步骤 4：重新启动调试目标 3129
调试后清理调试设置 3129
限制 3130
附加说明 3130
重置 Windows 管理员密码 3130
使用 EC2Launch v2 重置密码 3131
使用 EC2Launch 重置密码 3136
使用 EC2Config 重置密码 3140
排查 Sysprep 问题 3146
适用于 Linux 实例的 EC2Rescue 3147
安装 EC2Rescue 3148
运行 EC2Rescue 命令 3152
开发 EC2Rescue 模块 3154
EC2Rescue for Windows 实例 3160
使用 EC2Rescue GUI 进行问题排查 3161
使用 EC2Rescue CLI 进行问题排查 3166
使用 EC2Rescue 和 Systems Manager 进行问题排查 3173
EC2 Serial Console 3177
先决条件 3178
配置对 EC2 Serial Console 的访问 3185
连接到 EC2 Serial Console 3195
断开与 EC2 Serial Console 的连接 3204
使用 EC2 Serial Console 对实例进行故障排查 3205

发送诊断中断 3213
支持的实例类型 3214
先决条件 3214
发送诊断中断 3216
文档历史记录 3218
2018 年及更早的历史记录 3244

什么是 Amazon EC2？

Amazon Elastic Compute Cloud（Amazon EC2）在 Amazon Web Services（Amazon）云中按需提供可扩展的计算容量。使用 Amazon EC2 可以降低硬件成本，因此您可以更快地开发和部署应用程序。您可以使用 Amazon EC2 启动所需数量的虚拟服务器，配置安全性和联网以及管理存储。您可以添加容量（纵向扩展）来处理计算密集型任务，例如月度或年度进程或网站流量峰值。如果使用量减少，您可以再次减少容量（缩减）。

EC2 实例是 Amazon 云中的虚拟服务器。启动 EC2 实例时，您指定的实例类型决定了用于您的实例的主机硬件。每种实例类型以不同方式兼顾计算、内存、存储和网络资源。有关更多信息，请参阅 Amazon EC2 实例类型指南。



Amazon EC2 的功能

Amazon EC2 提供以下高级功能：

实例

虚拟服务器。

亚马逊机器映像（AMI）

实例的预置模板，其中包含服务器所需的组件（包括操作系统和其他软件）。

实例类型

实例 CPU、内存、存储、网络容量和图形硬件的多种配置。

Amazon EBS 卷

使用 Amazon Elastic Block Store（Amazon EBS）的数据的持久性存储卷。

实例存储卷

停止、休眠或终止实例时会删除的临时数据的存储卷。

密钥对

实例的安全登录信息。Amazon 会存储公有密钥，而您可以在安全位置存储私有密钥。

安全组

虚拟防火墙，允许您指定可以访问实例的协议、端口和源 IP 范围，以及实例可以连接的目标 IP 范围。

Amazon EC2 支持由商家或服务提供商处理、存储和传输信用卡数据，而且已经验证符合支付卡行业 (PCI) 数据安全标准 (DSS)。有关 PCI DSS 的更多信息，包括如何请求 Amazon PCI Compliance Package 的副本，请参阅 PCI DSS 第 1 级。

相关服务

与 Amazon EC2 结合使用的服务

可以将其他 Amazon Web Services 服务 与使用 Amazon EC2 部署的实例一起使用。

Amazon EC2 Auto Scaling

帮助确保您具有正确数量的 Amazon EC2 实例，可用于处理您的应用程序负载。

Amazon Backup

自动备份 Amazon EC2 实例及其附加的 Amazon EBS 卷。

Amazon CloudWatch

监控实例和 Amazon EBS 卷。

Elastic Load Balancing

可在多个实例间自动分配传入的应用程序流量。

Amazon GuardDuty

检测对 EC2 实例的潜在未经授权或恶意的使用情况。

EC2 Image Builder

自动创建、管理和部署自定义、安全且最新的服务器映像。

Amazon Launch Wizard

调整、配置和部署用于第三方应用程序的 Amazon 资源，而无需手动识别和预置单个 Amazon 资源。

Amazon Systems Manager

使用该安全的端到端管理解决方案在 EC2 实例上大规模执行操作。

其他计算服务

可以使用其他 Amazon 计算服务而不是使用 Amazon EC2 来启动实例。

Amazon Lightsail

使用 Amazon Lightsail 构建网站或 Web 应用程序，这是一个云平台，提供快速部署项目所需的资源，每月价格低廉、可预测。要比较 Amazon EC2 和 Lightsail，请参阅 Amazon Lightsail 或 Amazon EC2。

Amazon Elastic Container Service（Amazon ECS）

在 EC2 实例集群上部署、管理和扩展容器化应用程序。有关更多信息，请参阅选择 Amazon 容器服务。

Amazon Elastic Kubernetes Service(Amazon EKS)

在 Amazon 上运行您的 Kubernetes 应用程序。有关更多信息，请参阅选择 Amazon 容器服务。

访问 Amazon EC2

您可以使用以下界面创建和管理 Amazon EC2 实例：

Amazon EC2 控制台

用于创建和管理 Amazon EC2 实例和资源的简单 Web 界面。如果您已注册Amazon账户，可以通过登录 Amazon Web Services 管理控制台 并从控制台主页选择 EC2 来访问 Amazon EC2 控制台。

Amazon Command Line Interface

让您能够在命令行 Shell 中使用命令与 Amazon 服务进行交互。它在 Windows、Mac 和 Linux 上受支持。有关 Amazon CLI 的更多信息，请参阅 Amazon Command Line Interface 用户指南。您可以在 Amazon CLI 命令参考中查看 Amazon EC2 命令。

Amazon CloudFormation

Amazon EC2 支持使用 Amazon CloudFormation 创建资源。您可以创建一个用于描述 Amazon 资源的模板（采用 JSON 或 YAML 格式），然后 Amazon CloudFormation 会为您预置和配置这些资源。您可以重复使用 CloudFormation 模板来多次预置相同的资源，无论是在同一地区和账户中，还是在多个地区和账户中。有关 Amazon EC2 的受支持资源类型和属性的更多信息，请参阅《Amazon CloudFormation 用户指南》中的 EC2 资源类型参考。

Amazon SDK

如果您倾向于使用特定语言的 API 而非通过 HTTP 或 HTTPS 提交请求来构建应用程序，Amazon 为软件开发人员提供了库文件、示例代码、教程和其他资源。这些库文件提供可自动执行任务的基本功能，例如以加密方式对请求签名、重试请求和处理错误响应，以便您可以更轻松地上手。有关更多信息，请参阅用于在 Amazon 上构建的工具 。

Amazon Tools for PowerShell

一组根据 适用于 .NET 的 Amazon SDK 公开的功能构建的 PowerShell 模块。借助 Tools for PowerShell，您可以从 PowerShell 命令行为针对 Amazon 资源的操作编写脚本。要开始使用，请参阅《Amazon Tools for PowerShell 用户指南》。您可以在 Amazon Tools for PowerShell Cmdlet 参考中找到适用于 Amazon EC2 的 cmdlets。

查询 API

Amazon EC2 提供查询 API。这些请求属于 HTTP 或 HTTPS 请求，需要使用 HTTP 动词 GET 或 POST 以及一个名为 `Action` 的查询参数。有关 Amazon EC2 的 API 操作的更多信息，请参阅《Amazon EC2 API 参考》中的操作。

Amazon EC2 定价

Amazon EC2 提供以下定价选项：

按需型实例

您只需按秒支付使用实例的费用（最少 60 秒），无需长期订阅或支付预付款。

预留实例

可以通过承诺在 1 年或 3 年期限内提供特定的实例配置（包括实例类型和区域）来降低您的 Amazon EC2 成本。

竞价型实例

请求未使用的 EC2 实例，这可能会显著降低您的 Amazon EC2 成本。

专属主机

通过使用完全专供您使用的物理 EC2 服务器（按需付费或作为实惠配套的一部分）来降低成本。您可以使用现有的服务器绑定软件许可证，并获得满足合规性要求的帮助。

按需容量预留

在特定可用区中为 EC2 实例预留计算容量，持续时间不限。

按秒计费

从账单中扣除未使用的分钟数计费和秒数计费。

有关 Amazon EC2 的费用和价格的完整列表以及有关购买模式的更多信息，请参阅 Amazon EC2 定价。

估算、账单和成本优化

要为 Amazon 用例创建估算，请使用 Amazon 定价计算器。

要估算将 Microsoft 工作负载转换为使用部署在 Amazon 上的云原生开源服务的现代架构成本，请使用适用于 Microsoft 工作负载的 Amazon 现代化计算器。

若要查看您的账单，请转到 Amazon 账单与成本管理 控制台中的账单和成本管理控制面板。您的账单中包含了提供您的账单详情的使用情况报告的链接。要了解有关 Amazon 账户账单的更多信息，请参阅 Amazon 账单和成本管理用户指南。

如果您有关于 Amazon 账单、账户和事件的问题，请联系 Amazon Support。

在计算预配置环境的成本时，请记住包括附带成本，例如 EBS 卷的快照存储。

您可以使用 Amazon Trusted Advisor 来优化 Amazon 环境的成本、安全性和性能。

您可以使用 Amazon Cost Explorer 分析 EC2 实例的成本和使用情况。您最多可以查看过去 13 个月的数据，并预测您在接下来 12 个月内可能产生的费用。有关更多信息，请参阅《Amazon Cost Management User Guide》中的 Analyzing your costs and usage with Amazon Cost Explorer。

资源

- Amazon EC2 功能
- Amazon re:Post

- Amazon Skill Builder
- Amazon 支持
- 动手实践教程
- Web 托管
- Windows on Amazon

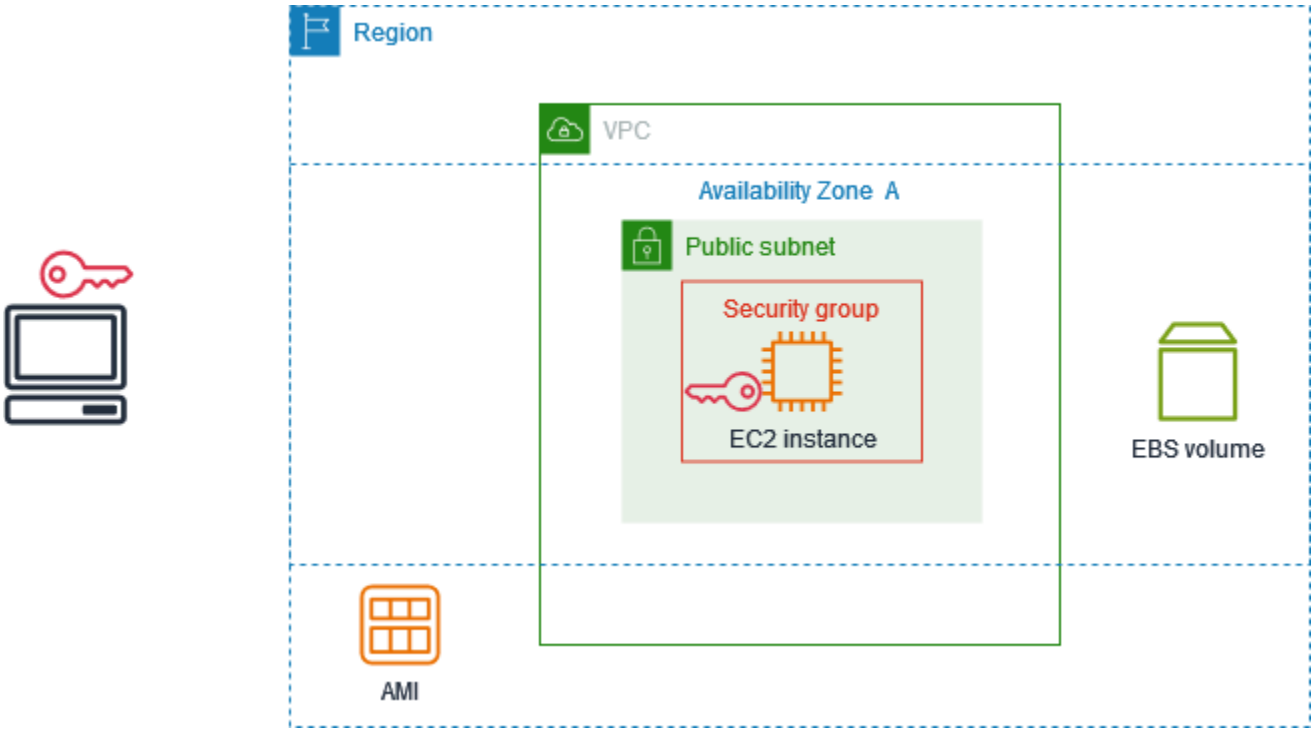
开始使用 Amazon EC2

利用本教程开始使用 Amazon Elastic Compute Cloud (Amazon EC2)。您将了解如何启动和连接到 EC2 实例。实例是 Amazon 云中的虚拟服务器。您可以使用 Amazon EC2 来创建和配置在实例上运行的操作系统和应用程序。

概览

下图演示了本教程中将使用的主要组件：

- 映像：包含要在您的实例上运行的软件（例如操作系统）的模板。
- 密钥对：您在连接到实例时用于证明个人身份的一组安全凭证。公有密钥在您的实例上，而私有密钥在您的计算机上。
- 网络：虚拟私有云（VPC）是专用于您的 Amazon Web Services 账户的虚拟网络。为帮助您快速开始使用，您的账户在每个 Amazon Web Services 区域中都附带一个默认 VPC，并且每个默认 VPC 在每个可用区中都有一个默认子网。
- 安全组：充当虚拟防火墙，以控制入站和出站流量。
- EBS 卷：我们需要映像的根卷。您可以选择性地添加数据卷。



任务

- 步骤 1：启动实例
- 步骤 2：连接到您的实例
- 步骤 3：清除您的实例
- 后续步骤

步骤 1：启动实例

您可以根据以下过程所述使用 Amazon Web Services 管理控制台 启动 EC2 实例。本教程旨在帮助您快速启动第一个实例，因此不会涵盖所有可能的选项。

启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，我们会显示当前 Amazon Web Services 区域（例如，俄亥俄州）。您可以使用所选区域，也可以选择离您更近的区域。
3. 从 EC2 控制台控制面板的启动实例窗格中，选择启动实例。
4. 在 Name and tags（名称与标签）下，为 Name（名称）输入实例的描述性名称。
5. 在 Application and OS Images (Amazon Machine Image)（应用程序和操作系统映像 (Amazon Machine Image)）中，执行以下操作：
   a. 选择快速启动，然后为您的实例选择操作系统（OS）。对于您的第一个 Linux 实例，我们建议您选择 Amazon Linux。
   b. 从亚马逊机器映像（AMI）中，选择的 AMI。
6. 在实例类型下，对于实例类型，选择的实例类型。
7. 在密钥对（登录）下，对于密钥对名称，选择一个现有密钥对，或选择创建新密钥对以创建您的第一个密钥对。

   > **Warning**
   > 如果您选择在没有密钥对的情况下继续（不推荐），则将无法使用本教程中所述的方法连接到您的实例。

8. 请注意在网络设置下，我们已经选择了您的默认 VPC，选择了在我们为您选择的可用区中使用默认子网的选项，并使用一条允许从任何位置 (`0.0.0.0/0`) 连接到实例的规则配置了一个安全组。

> **Warning**
>
> 如果您指定 `0.0.0.0/0`，则会启用来自世界上任何 IP 地址的流量。对于 SSH 和 RDP 协议，您可能考虑这在测试环境下短时间内是可以接受的，但它对于生产环境并不安全。对于生产环境，请确保仅授权从适当的单个 IP 地址或地址范围进行访问。

对于第一个实例，我们建议您使用默认设置。否则，您可以按如下方式更新网络设置：

- （可选）要使用特定的默认子网，请选择编辑，然后选择一个子网。
- （可选）要使用不同的 VPC，请选择编辑，然后选择现有 VPC。如果 VPC 没有配置为公有互联网访问，则您将无法连接到您的实例。
- （可选）要限制特定网络的入站连接流量，请选择自定义而不是任何位置，然后输入您的网络的 CIDR 块。
- （可选）要使用其他安全组，请选择选择现有安全组，然后选择现有的安全组。如果安全组没有允许来自网络的连接流量的规则，则您将无法连接到实例。对于 Linux 实例，您必须允许 SSH 流量。对于 Windows 实例，您必须允许 RDP 流量。

9. 在配置存储下，请注意我们配置了根卷但没有配置数据卷。这对于测试目的来说已经足够了。
10. 查看 Summary（摘要）面板中您的实例配置的摘要，当您准备好后，选择 Launch instance（启动实例）。
11. 如果启动成功，则请从成功通知中选择该实例的 ID 以打开实例页面并监控启动状态。
12. 选中该实例的复选框。初始实例状态为 `pending`。实例启动后，其状态将更改为 `running`。选择状态和警报选项卡。在您的实例通过其状态检查后，该实例就可以接收连接请求了。

步骤 2：连接到您的实例

您使用的过程取决于实例的操作系统。如果您无法连接到实例，请参阅排查 Amazon EC2 Linux 实例的连接问题以获得帮助。

Linux 实例

您可以使用任何 SSH 客户端连接到您的 Linux 实例。如果您的计算机上运行的是 Windows，则请打开终端并运行 `ssh` 命令，以验证是否安装了 SSH 客户端。如果找不到该命令，则请安装适用于 Windows 的 OpenSSH。

使用 SSH 连接到您的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择该实例，然后选择 Connect (连接)。
4. 在连接到实例页面上，选择 SSH 客户端选项卡。
5. （可选）如果您在启动实例时创建了密钥对，并将私有密钥（.pem 文件）下载到运行 Linux 或 macOS 的计算机，则请运行示例 chmod 命令来设置私有密钥的权限。
6. 复制示例 SSH 命令。以下是一个示例，其中 *key-pair-name*.pem 是私有密钥文件的名称，*ec2-user* 是与映像关联的用户名，@ 符号后的字符串是实例的公有 DNS 名称。

```
ssh -i key-pair-name.pem ec2-user@ec2-198-51-100-1.us-east-2.compute.amazonaws.com
```

7. 在计算机的终端窗口，运行您在上一步中保存的 ssh 命令。如果私有密钥文件不在当前目录中，则必须在此命令中指定密钥文件的完全限定路径。

   以下为响应示例：

```
The authenticity of host 'ec2-198-51-100-1.us-east-2.compute.amazonaws.com
 (198-51-100-1)' can't be established.
ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no)?
```

8. （可选）验证安全警报中的指纹是否与您首次启动实例时控制台输出中包含的实例指纹相匹配。要获取控制台输出，请依次选择操作、监控和故障排除、获取系统日志。如果指纹不匹配，则表示有人可能在试图实施中间人攻击。如果匹配，请继续到下一步。
9. 输入 **yes**。

   以下为响应示例：

```
Warning: Permanently added 'ec2-198-51-100-1.us-
east-2.compute.amazonaws.com' (ECDSA) to the list of known hosts.
```

Windows 实例

要使用 RDP 连接到 Windows 实例，您必须检索初始管理员密码，然后在连接到实例时输入此密码。此密码在实例启动几分钟之后才可用。您的账户必须拥有调用 GetPasswordData 操作的权限。有关更多信息，请参阅 用于控制访问 Amazon EC2 API 的示例策略。

管理员账户的默认用户名取决于 AMI 中包含的操作系统（OS）的语言。要确定正确的用户名，请识别操作系统的语言，然后选择相应的用户名。例如，对于英语操作系统，用户名是 `Administrator`；对于法语操作系统，用户名是 `Administrateur`；对于葡萄牙语操作系统，则是 `Administrador`。如果某个语言版本的操作系统没有相同语言的用户名，请选择用户名 `Administrator (Other)`。有关更多信息，请参阅 Microsoft 网站中的 Localized Names for Administrator Account in Windows。

检索初始管理员密码

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择该实例，然后选择 Connect (连接)。
4. 在连接到实例页面上，选择 RDP 客户端选项卡。
5. 在用户名中，选择管理员账户的默认用户名。您选择的用户名必须与用于启动实例的 AMI 中包含的操作系统（OS）语言相匹配。如果没有与操作系统相同语言的用户名，请选择管理员（其他）。
6. 选择获取密码。
7. 在获取 Windows 密码页面上，执行以下操作：
   a. 选择上传私有密钥文件，然后导航至您启动实例时指定的私有密钥（`.pem`）文件。选择文件并选择 Open（打开），以便将文件的全部内容复制到此窗口。
   b. 选择解密密码。获取 Windows 密码页面将关闭，实例的默认管理员密码将显示在密码下，替换先前显示的获取密码链接。
   c. 复制密码并将其保存在安全的位置。需要使用此密码连接实例。

以下过程使用适用于 Windows 的远程桌面连接客户端（MSTSC）。如果使用的是其他 RDP 客户端，请下载 RDP 文件，然后查看 RDP 客户端的文档，了解建立 RDP 连接的步骤。

使用 RDP 客户端连接到 Windows 实例

1. 在连接到实例页面上，选择下载远程桌面文件。完成文件下载后，选择取消，然后返回实例页面。RDP 文件已下载到 `Downloads` 文件夹。
2. 运行 `mstsc.exe` 来打开 RDP 客户端。
3. 展开显示选项，选择打开，然后从 `Downloads` 文件夹中选择.rdp 文件。
4. 默认情况下，计算机是实例的公有 IPv4 DNS 名称，用户名是管理员账户。要改用 IPv6 连接到实例，请将实例的公有 IPv4 DNS 名称替换为其 IPv6 地址。查看默认设置并根据需要进行更改。

5. 选择连接。如果收到一条警告，指出远程连接发布者未知，请选择连接继续进行操作。
6. 输入之前保存的密码，然后选择确定。
7. 由于自签名证书的固有特性，您可能会看到一条警告，指出无法验证该安全证书。请执行以下操作之一：
   - 如果您信任该证书，则请选择是连接到您的实例。
   - [Windows] 在继续操作之前，请将证书的指纹与系统日志中的值进行比较，以确认远程计算机的身份。选择查看证书，然后从详细信息选项卡选择指纹。将此值与操作、监控和故障排除、获取系统日志中的 RDPCERTIFICATE-THUMBPRINT 值进行比较。
   - [Mac OS X] 在继续操作之前，请将证书的指纹与系统日志中的值进行比较，以确认远程计算机的身份。选择显示证书，展开详细信息，然后选择 SHA1 指纹。将此值与操作、监控和故障排除、获取系统日志中的 RDPCERTIFICATE-THUMBPRINT 值进行比较。
8. 如果 RDP 连接成功，RDP 客户端会先显示 Windows 登录屏幕，再显示 Windows 桌面。如果收到错误消息，请参阅the section called “远程桌面无法连接到远程计算机”。完成 RDP 连接后，可以关闭 RDP 客户端。

步骤 3：清除您的实例

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

在您完成为本教程创建的实例后，应通过终止该实例进行清除。如果在清除该实例前要对其执行更多操作，请参阅后续步骤。

一旦实例变为 shutting down 或 terminated 状态，该实例就会停止产生费用，。若要保留实例以供将来使用，但又不产生费用，您可以立即停止实例，稍后再重新启动。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。

要终止您的实例，请执行以下操作：

1. 在导航窗格中，选择 Instances (实例)。在实例列表中选择实例。

2. 依次选择实例状态、终止（删除）实例。
3. 当系统提示您确认时，选择终止（删除）。

   Amazon EC2 关闭并终止您的实例。您的实例在终止之后，短时间内仍将在控制台上可见，然后该条目将自动删除。您不能自己从控制台显示中删除已终止的实例。

后续步骤

启动实例后，您可能想要探索以下后续步骤：

- 借助入门教程了解 Amazon EC2 的核心概念。有关更多信息，请参阅 EC2 实例的启动教程。
- 添加 EBS 卷。有关更多信息，请参阅《Amazon EBS 用户指南》中的创建 Amazon EBS 卷。
- 了解如何使用 Run 命令远程管理您的 EC2 实例。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的 Amazon Systems Manager Run Command。
- 了解实例购买选项。有关更多信息，请参阅 Amazon EC2 账单和购买选项。
- 获取有关实例类型的建议。有关更多信息，请参阅 从 EC2 实例类型查找器获取建议。

Amazon EC2 的最佳实践

为最大程度地利用 Amazon EC2 的优势，建议您执行以下最佳实践。

安全性

- 尽可能将身份联合验证与身份提供商和 IAM 角色结合使用，以管理对 Amazon 资源和 API 的访问权限。有关更多信息，请参阅 IAM 用户指南中的身份提供者和联合身份验证。
- 为安全组实现最严格的规则。
- 定期修补、更新和保护实例上的操作系统和应用程序。有关更多信息，请参阅 更新管理。有关特定于 Windows 操作系统的指导原则，请参阅 Windows 实例的安全最佳实践。
- 使用 Amazon Inspector 可以自动发现并扫描 Amazon EC2 实例是否存在软件漏洞和意外网络暴露。有关更多信息，请参阅 Amazon Inspector 用户指南。
- 根据安全最佳实践和安全标准，使用 Amazon Security Hub CSPM 控件监控 Amazon EC2 资源。有关使用 Security Hub CSPM 的更多信息，请参阅《Amazon Security Hub CSPM 用户指南》中的 Amazon Elastic Compute Cloud 控件。

存储

- 了解根卷类型对数据持久性、备份和恢复的影响。有关更多信息，请参阅 根卷类型。
- 对操作系统与您的数据分别使用单独的 Amazon EBS 卷。确保含有您数据的卷可在实例终止后保留。有关更多信息，请参阅 实例终止时保留数据。
- 使用您的实例可用的实例存储来存储临时数据。请注意，当您停止、休眠或终止您的实例时，系统会删除存储在实例存储中的数据。如果将实例存储用于数据库存储，请确保您拥有一个具有重复因子的集群，从而确保容错。
- 对 EBS 卷和快照进行加密。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 加密。

资源管理

- 使用实例元数据和自定义资源标签跟踪并确定您的 Amazon 资源。有关更多信息，请参阅 使用实例元数据管理 EC2 实例 和 标记 Amazon EC2 资源。
- 查看您的 Amazon EC2 的当前限制。需要时请提前计划请求提高限制。有关更多信息，请参阅 Amazon EC2 Service Quotas。

- 使用 Amazon Trusted Advisor 检查您的 Amazon 环境，然后在有可能节省开支、提高系统可用性和性能或弥补安全漏洞时为您提供建议。有关更多信息，请参阅《Amazon Web Services 支持 用户指南》中的 Amazon Trusted Advisor。

备份和恢复

- 使用 Amazon EBS 快照定期备份您的 EBS 卷，并从您的实例创建 Amazon Machine Image (AMI)，以便保存配置以作为启动未来实例的模板。有关有助于实现此使用案例的 Amazon 服务的更多信息，请参阅 Amazon Backup 和 Amazon Data Lifecycle Manager。
- 跨多个可用区部署应用程序的关键组件，并适当地复制数据。
- 设计您的应用程序，以便在实例重新启动时处理动态 IP 地址分配。有关更多信息，请参阅 Amazon EC2 实例 IP 寻址。
- 监控和响应事件。有关更多信息，请参阅 监控 Amazon EC2 资源。
- 确保您已准备好处理故障转移。对于基本解决方案，您可以手动将网络接口或弹性 IP 地址附加到替换实例。有关更多信息，请参阅 弹性网络接口。对于自动解决方案，您可以使用 Amazon EC2 Auto Scaling。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南。
- 定期测试恢复您的实例和 Amazon EBS 卷的过程，以确保成功还原数据和服务。

联网

- 对于 IPv4 和 IPv6，将应用程序的生存时间 (TTL) 值设置为 255。如果使用更小的值，则存在 TTL 在应用程序流量传输过程中过期的风险，从而导致实例发生可访问性问题。

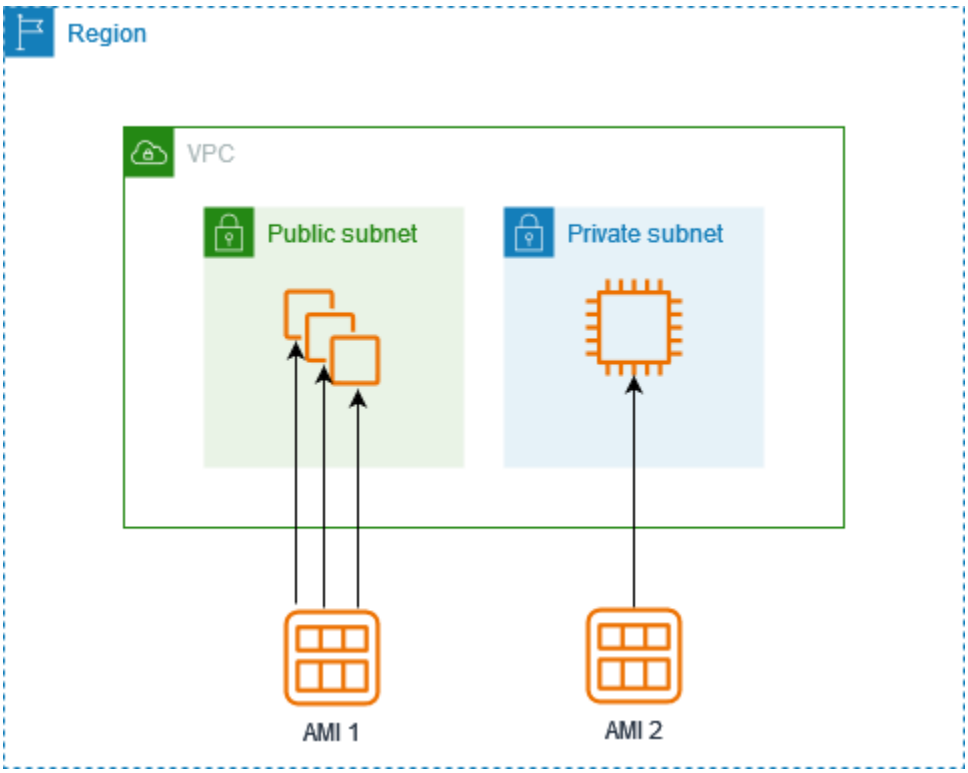
Amazon EC2 中的亚马逊机器映像

亚马逊机器映像（AMI）是一个映像，它提供设置和启动 Amazon EC2 实例所需的软件。每个 AMI 还包含块设备映射，指定要附加到启动的实例的块设备。在启动实例时，您必须指定 (AMI)。AMI 必须与您为实例选择的实例类型兼容。您可以使用 Amazon 提供的 AMI、公共 AMI、其他人与您共享的 AMI 或从 Amazon Web Services Marketplace 购买的 AMI。

AMI 特定于以下各项：

- 区域
- 操作系统
- 处理器架构
- 根卷类型
- 虚拟化类型

在需要具有相同配置的多个实例时，您可以从单个 AMI 启动多个实例。在需要不同的配置的实例时，您可以使用其他 AMI 启动实例，如下图中所示。



您可以从 Amazon EC2 实例创建 AMI，然后使用它启动具有相同配置的实例。您可以将 AMI 复制到另一个 Amazon 区域，然后使用它在该区域中启动实例。您还可以与其他账户共享您创建的 AMI，以便他们可以启动具有相同配置的实例。您可以使用 Amazon Web Services Marketplace 销售 AMI。

内容

- Amazon EC2 中的 AMI 类型和特征
- 查找满足您的 EC2 实例要求的 AMI
- Amazon Web Services Marketplace 中适用于 Amazon EC2 实例的付费 AMI
- Amazon EC2 AMI 生命周期
- 使用 Amazon EC2 启动模式的实例启动行为
- 将加密与 EBS 支持的 AMI 结合使用
- 了解 Amazon EC2 中共享 AMI 的使用情况
- 使用 Amazon EventBridge 监控 AMI 事件
- 了解 AMI 账单信息
- Amazon EC2 中的 AMI 配额

Amazon EC2 中的 AMI 类型和特征

启动实例时，您选择的 AMI 必须与您选择的实例类型兼容。可以基于以下特性选择要使用的 AMI：

- 区域
- 操作系统
- 处理器架构
- 启动许可
- 根卷类型
- 虚拟化类型

启动许可

启动许可决定谁可以使用 AMI 启动实例。您可以将启动许可视为共享 AMI，即当您授予启动许可时，您就是在与其他用户共享 AMI。只有 AMI 的拥有者可以通过指定启动许可来确定其可用性。启动许可分为以下类别。

启动权限	描述
公有	拥有者向所有 Amazon 账户授予启动许可。
显式	拥有者向特定 Amazon 账户、企业或企业部门（OU）授予启动许可。
隐式	拥有者拥有 AMI 的隐式启动许可。

Amazon 和 Amazon EC2 社区提供了大量的公用 AMI。有关更多信息，请参阅了解 Amazon EC2 中共享 AMI 的使用情况。开发人员可以为其 AMI 收费。有关更多信息，请参阅 Amazon Web Services Marketplace 中适用于 Amazon EC2 实例的付费 AMI。

根卷类型

所有 AMI 均可归类为由 Amazon EBS 支持或由 Amazon S3 支持。

- Amazon EBS-backed AMI – 从 AMI 启动的实例的根卷是从 Amazon EBS 快照创建的 Amazon Elastic Block Store（Amazon EBS）卷。同时支持 Linux 和 Windows AMI。
- Amazon S3 支持的 AMI – 从 AMI 启动的实例的根卷是根据一个存储在 Amazon S3 中的模板创建而来的实例存储卷。仅支持 Linux AMI。Windows AMI 不支持根卷的实例存储。

有关更多信息，请参阅 Amazon EC2 实例的根卷。

> **Note**
>
> Amazon S3 支持的 AMI 已达生命周期终止，不建议新使用此类 AMI。仅下列早期实例类型支持此类 AMI：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

下表总结了使用两种类型的 AMI 时的重要区别。

特征	Amazon EBS-backed AMI	Amazon S3 支持的 AMI
根卷	EBS 卷	实例存储卷
实例的启动时间	通常不到 1 分钟	通常不到 5 分钟
数据持久性		

特征	Amazon EBS-backed AMI	Amazon S3 支持的 AMI
	默认情况下，实例终止时将删除根卷。* 默认情况下，在实例终止后，任何其他 EBS 卷上的数据仍然存在。	任意实例存储卷上的数据仅在实例的生命周期内保留。
停止状态	可以处于停止状态。即使实例已停止并且未运行，根卷也会持久保留在 Amazon EBS 中。	不可置于已停止状态；实例正在运行或已终止。
修改	实例停止后，实例类型、内核、RAM 磁盘和用户数据仍可更改。	实例存在期间，实例属性是稳定不变的。
收费	您需要为实例使用、EBS 卷使用以及将 AMI 存储为 EBS 快照付费。	您需要为实例使用以及在 Amazon S3 中存储 AMI 付费。
AMI 创建/捆绑	使用单一命令/调用	需要安装和使用 AMI 工具

* 默认情况下，EBS 根卷的 `DeleteOnTermination` 标志设置为 `true`。有关如何更改此标志以便卷在终止之后保留的信息，请参阅Amazon EC2 实例终止后会保留 Amazon EBS 根卷。

** 仅支持 `io2` EBS Block Express。有关更多信息，请参阅《Amazon EBS 用户指南》中的预调配 IOPS SSD Block Express 卷。

识别由您的 AMI 确定的根卷类型

您用于启动 EC2 实例的 AMI 决定根卷的类型。EC2 实例的根卷是 EBS 卷或实例存储卷。

基于 Nitro 的实例仅支持 EBS 根卷。下列上一代实例类型是支持实例存储根卷的唯一实例类型：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

Console

识别由 AMI 确定的根卷类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 AMIs（AMI），选择 AMI。
3. 在 Details（详细信息）选项卡中检查 Root Device Type（根设备类型）的值，如下所示：
   - ebs – 从此 AMI 启动的实例将获得 EBS 根卷
   - instance store – 从此 AMI 启动的实例将获得实例存储根卷。

Amazon CLI

识别由 AMI 确定的根卷类型

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].RootDeviceType
```

下面是示例输出。

```
ebs
```

PowerShell

识别由 AMI 确定的根卷类型

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image `
    -ImageId ami-0abcdef1234567890).RootDeviceType.Value
```

下面是示例输出。

```
ebs
```

虚拟化类型

亚马逊机器映像使用两种虚拟化类型之一：半虚拟化 (PV) 或硬件虚拟机 (HVM)。半虚拟化和 HVM AMI 之间的主要区别在于它们的启动方式，以及它们能否使用特定硬件扩展（CPU、网络和存储）实现更好的性能。Windows AMI 是硬件虚拟机 AMI。

下表比较 HVM 和 PV AMI。

特征	全虚拟化	PV
描述	硬件虚拟机 AMI 配有一组完全虚拟化的硬件，通过执行映像根块储存设备的主启动记录来启动。通过此虚拟化类型可以直接在虚拟机上运行操作系统而不进行任何修改 (如同它在裸机硬件上运行一样)。Amazon EC2 主机系统可模拟向客户机提供的部分或所有底层硬件。	半虚拟化 AMIs 使用名为 PV-GRUB 的特殊启动加载程序启动，该加载程序开始启动周期，然后对映像链式加载 `menu.lst` 文件中指定的内核。半虚拟来宾可以在没有明确支持虚拟化的主机硬件上运行。有关 PV-GRUB 及其在 Amazon EC2 中的使用情况的更多信息，请参阅 User provided kernels。
支持的实例类型	当前一代的所有实例类型都支持 HVM AMI。	以下上一代实例类型支持 PV AMI：C1、C3、M1、M3、M2 和 T1。最新一代实例类型不支持 PV AMI。
支持硬件扩展	硬件虚拟机客户机可以利用硬件扩展快速访问主机系统上的底层硬件。它们需要使用增强联网和 GPU 处理。要将指令传递给专用网络和 GPU 设备，操作系统必须能访问本机硬件平台；HVM 虚拟化可提供这种访问。有关更多信息，请参阅 Amazon EC2 实例上的增强联网功能。	否，它们无法利用特殊的硬件扩展，例如增强网络或 GPU 处理。
如何查找	使用控制台或 describe-images 命令验证 AMI 的虚拟化类型是否已设置为 `hvm`。	使用控制台或 describe-images 命令验证 AMI 的虚拟化类型是否已设置为 `paravirtual` 。

硬件虚拟机上的半虚拟化

以往，半虚拟化客户机在存储和网络操作方面的性能要优于硬件虚拟机客户机，因为它们可以对 I/O 使用特殊驱动程序，从而避免模拟网络和磁盘硬件的开销，而硬件虚拟机客户机必须将这些指令转换为模拟的硬件。现在，半虚拟化驱动程序可用于硬件虚拟机客户机，因此无法移植到半虚拟化环境中运行的操作系统仍可以通过它们获得存储和网络 I/O 方面的性能优势。借助这些硬件虚拟机驱动程序上的半虚拟化，硬件虚拟机客户机可以获得与半虚拟化客户机相同甚至更佳的性能。

查找满足您的 EC2 实例要求的 AMI

AMI 包括启动实例所需的组件和应用程序，例如操作系统和根卷的类型。要启动实例，必须找到满足您需求的 AMI。

选择 AMI 时，对于要启动的实例，可能需要考虑以下要求：

- AMI 的 Amazon 区域，因为 AMI ID 在每个区域中都是唯一的。
- 操作系统（例如，Linux 或 Windows）。
- 架构（例如，32 位、64 位或 64 位 ARM）。
- 根卷类型（例如，Amazon EBS 或实例存储）。
- 提供商（例如，亚马逊云科技）。
- 其他软件（例如，SQL Server）。

Console

您可以在使用启动实例向导时从 AMI 列表中进行选择，也可以使用映像页面在所有可用的 AMI 中进行搜索。

使用启动实例向导查找快速启动 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航栏，选择您在其中启动实例的区域。您可以选择向您提供的任何区域，无需理会您身处的位置。AMI ID 在每个 Amazon 区域中都是唯一的。
3. 从控制台控制面板中，选择启动实例。
4. 在应用程序和操作系统映像（亚马逊机器映像）下，选择快速启动，选择适用于实例的操作系统 (OS)，然后在亚马逊机器映像（AMI）中，从列表中选择一个常用的 AMI。如果您没有看到想要使用的 AMI，请选择 Browser more AMIs（浏览更多 AMI）以浏览完整的 AMI 目录。有关更多信息，请参阅 应用程序和操作系统映像（亚马逊机器映像）。

使用 AMI 页面查找 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航栏，选择您在其中启动实例的区域。您可以选择向您提供的任何区域，无需理会您身处的位置。AMI ID 在每个 Amazon 区域中都是唯一的。
3. 在导航窗格中，选择 AMI。
4. （可选）使用筛选条件和搜索选项，将显示的 AMI 列表范围限定为仅能查看符合您的标准的 AMI。

   例如，要列出 Amazon 提供的所有 AMI，请选择公有映像。然后使用搜索选项进一步缩小显示的 AMI 列表的范围。选择 Search（搜索）栏，然后从菜单中选择 Owner alias（拥有者别名），然后选择 = 运算符，最后选择值 amazon。要查找与特定平台（例如 Linux 或 Windows）匹配的 AMI，请再次选择搜索栏以选择平台，然后选择 = 运算符，从提供的列表中选择操作系统。
5. （可选）选择首选项图标，以选择要显示的映像属性，例如根卷类型。或者，可以从列表中选择 AMI，然后在 Details（详细信息）选项卡中查看其属性。
6. 选择 AMI 之前，请确认它是由实例存储支持还是由 Amazon EBS 支持并了解此差异的影响，这十分重要。有关更多信息，请参阅 根卷类型。
7. 要从此 AMI 启动一个实例，请选择此实例，然后选择从映像启动实例。有关使用控制台启动实例的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。如果您没有准备好立即启动实例，请记下 AMI ID 以供将来使用。

Amazon CLI

使用 describe-images 命令查找符合要求的 AMI。默认情况下，此命令会返回所有公有 AMI、由您拥有的 AMI 以及与您共享的 AMI。

查找由 Amazon 拥有的 AMI

使用 describe-images 命令和 `--owners` 选项。

```
aws ec2 describe-images --owners amazon
```

查找 Windows AMI

添加以下筛选条件，从而仅显示 Windows AMI。

```
--filters "Name=platform,Values=windows"
```

查找 EBS-backed AMI

添加以下筛选条件，从而仅显示 Amazon EBS-backed AMI。

```
--filters "Name=root-device-type,Values=ebs"
```

PowerShell

使用 Get-EC2Image cmdlet 查找符合要求的 AMI。默认情况下，此 cmdlet 会返回所有公有 AMI、由您拥有的 AMI 或与您共享的 AMI。

查找由 Amazon 拥有的 AMI

使用 Get-EC2Image 命令和 `-Owner` 参数。

```
Get-EC2Image -Owner amazon
```

查找 Windows AMI

添加以下筛选条件，从而仅显示 Windows AMI。

```
-Filter @{Name="platform"; Values="windows"}
```

有关更多示例，请参阅《Amazon Tools for PowerShell 用户指南》中的 Find an Amazon Machine Image Using Windows PowerShell。

相关资源

有关特定操作系统的 AMI 的更多信息，请参阅以下内容：

- Amazon Linux 2023 –《Amazon Linux 2023 用户指南》中的 AL2023 on Amazon EC2
- Ubuntu – Canonical Ubuntu 网站上的 Amazon EC2 AMI Locator
- RHEL – Red Hat 网站上的 Red Hat Enterprise Linux Images (AMI) Available on Amazon Web Services (Amazon)
- Windows 服务器：Amazon Windows AMI 参考

有关您可以在 Amazon Web Services Marketplace 上订阅的 AMI 的信息，请参阅 Amazon Web Services Marketplace 中适用于 Amazon EC2 实例的付费 AMI。

有关使用 Systems Manager 帮助用户找到启动实例时应使用的最新 AMI 的信息，请参阅以下内容：

- 使用 Systems Manager 参数引用 AMI
- 使用 Systems Manager 公共参数引用最新的 AMI

使用 Systems Manager 参数引用 AMI

在 Amazon EC2 控制台中使用 EC2 启动实例向导启动实例时，您可以从列表中选择 AMI，也可以选择指向 AMI ID 的 Amazon Systems Manager 参数（如此部分中所述）。如果您使用自动化代码启动实例，则可以指定 Systems Manager 参数而不是 AMI ID。

Systems Manager 参数是客户定义的键/值对，您可以在 Systems Manager Parameter Store 中创建该键/值对。Parameter Store 提供了一个中央存储来对应用程序配置值进行外部化。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的 Amazon Systems Manager Parameter Store。

创建指向 AMI ID 的参数时，请确保将数据类型指定为 `aws:ec2:image`。指定此数据类型可确保在创建或修改参数时，将参数值作为 AMI ID 进行验证。有关更多信息，请参阅 Amazon Systems Manager 用户指南中对亚马逊机器映像 ID 的原生参数支持。

目录

- 使用案例
- 权限
- 限制
- 使用 Systems Manager 参数启动实例

使用案例

当您使用 Systems Manager 参数指向 AMI ID 时，您的用户在启动实例时可以更轻松地选择正确的 AMI。Systems Manager 参数还可以简化自动化代码的维护。

用户使用更简单

如果您需要使用特定 AMI 启动实例并且该 AMI 定期更新，我们建议您要求用户选择 Systems Manager 参数来查找 AMI。要求用户选择 Systems Manager 参数可确保使用最新的 AMI 启动实例。

例如，您的组织中可能会每个月创建一个新版本的 AMI，其中具有最新操作系统和应用程序修补程序。同时，您要求用户使用最新版本的 AMI 启动实例。为确保用户使用最新版本，您可以创建指向正确 AMI ID 的 Systems Manager 参数（例如 `golden-ami`）。每次创建新版本的 AMI 时，您需要更新参数中的 AMI ID 值，以使其始终指向最新 AMI。您的用户不需要了解对 AMI 的定期更新，因为他们每

次都会继续选择相同的 Systems Manager 参数。为您的 AMI 使用 Systems Manager 参数可让他们更轻松地选择正确的 AMI 来启动实例。

简化自动化代码维护

如果您使用自动化代码启动实例，则可以指定 Systems Manager 参数而不是 AMI ID。如果创建了新版本的 AMI，您可以更改参数中的 AMI ID 值，以使其指向最新 AMI。每次创建新版本的 AMI 时，都不需要修改引用该参数的自动化代码。这样做简化了自动化的维护，有助于降低部署成本。

> **Note**
>
> 当您更改 Systems Manager 参数指向的 AMI ID 时，正在运行的实例不受影响。

权限

如果您在启动实例向导中使用指向 AMI ID 的 Systems Manager 参数，则必须将以下权限添加到您的 IAM 策略：

- `ssm:DescribeParameters`：授予查看和选择 Systems Manager 参数的权限。
- `ssm:GetParameters`：授予检索 Systems Manager 参数值的权限。

您还可以限制对特定 Systems Manager 参数的访问权限。有关更多信息以及示例 IAM 策略，请参阅示例：使用 EC2 启动实例向导。

限制

AMI 和 Systems Manager 参数特定于区域。要跨区域使用相同的 Systems Manager 参数名称，请在各个区域中创建具有相同名称的 Systems Manager 参数（例如，`golden-ami`）。在每个区域中，将 Systems Manager 参数指向该区域的 AMI。

参数名称区分大小写。只有在参数是层次结构的一部分时，参数名称才需要使用反斜杠，例如，`/amis/production/golden-ami`。如果参数不是层次结构的一部分，您可以省略反斜杠。

使用 Systems Manager 参数启动实例

启动实例时，您可以不指定 AMI ID，而选择指定指向某个 AMI ID 的 Systems Manager 参数。

要以编程方式指定参数，请使用以下语法，其中 `resolve:ssm` 是标准前缀，`parameter-name` 是唯一参数名称。

```
resolve:ssm:parameter-name
```

Systems Manager 参数具有版本支持。参数的每个迭代将分配一个唯一的版本号。您可以按如下方式引用参数的版本，其中 `version` 是唯一版本号。默认情况下，未指定版本时将使用参数的最新版本。

```
resolve:ssm:parameter-name:version
```

要使用 Amazon 提供的公有参数启动实例，请参阅使用 Systems Manager 公共参数引用最新的 AMI。

Console

使用 Systems Manager 参数查找 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航栏，选择您在其中启动实例的区域。您可以选择向您提供的任何区域，无需理会您身处的位置。
3. 从控制台控制面板中，选择启动实例。
4. 在应用程序和操作系统镜像（Amazon 机器映像）下，选择浏览其他 AMI。
5. 选择搜索栏右侧的箭头按钮，然后选择 Search by Systems Manager parameter（按 Systems Manager 参数搜索）。
6. 对于 Systems Manager 参数 (Systems Manager 参数)，请选择一个参数。相应的 AMI ID 将显示在 Currently resolves to（当前解析为）下方。
7. 选择搜索。与 AMI ID 匹配的 AMI 将显示在列表中。
8. 从列表中选择 AMI，然后选择选择。

有关使用启动实例向导启动实例的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

使用 Systems Manager 参数启动实例

使用 run-instances 命令和 `--image-id` 选项。此示例使用名为 **`golden-ami`** 的 Systems Manager 参数来指定 AMI ID。

```
--image-id resolve:ssm:/golden-ami
```

您可以为 Systems Manager 参数创建若干版本。以下示例指定了 **golden-ami** 参数的版本 2。

```
--image-id resolve:ssm:/golden-ami:2
```

PowerShell

使用 Systems Manager 参数启动实例

将 New-EC2Instance cmdlet 与 `-ImageId` 参数结合使用。此示例使用名为 **golden-ami** 的 Systems Manager 参数来指定 AMI ID。

```
-ImageId "resolve:ssm:/golden-ami"
```

您可以为 Systems Manager 参数创建若干版本。以下示例指定了 **golden-ami** 参数的版本 2。

```
-ImageId "resolve:ssm:/golden-ami:2"
```

使用 Systems Manager 公共参数引用最新的 AMI

Amazon Systems Manager 为 Amazon 维护的公有 AMI 提供公有参数。您可以在启动实例时使用公有参数来确保使用最新 AMI。例如，公有参数 `/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-arm64` 在所有区域中均可用，并且始终指向给定区域中适用于 arm64 架构的最新版本 Amazon Linux 2023 AMI。

在以下路径中提供公有参数：

- Linux – `/aws/service/ami-amazon-linux-latest`
- Windows – `/aws/service/ami-windows-latest`

有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的使用公共参数。

列出 Amazon Linux AMI

Amazon CLI

列出当前 Amazon 区域中的 Linux AMI

使用以下 get-parameters-by-path 命令。`--path` 参数的值因 Linux AMI 而异。

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-amazon-linux-latest \
    --query "Parameters[].Name"
```

PowerShell

列出当前 Amazon 区域中的 Linux AMI

使用 Get-SSMParametersByPath cmdlet。

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-amazon-linux-latest" | `
    Sort-Object Name | Format-Table Name
```

列出 Windows AMI

Amazon CLI

列出当前 Amazon 区域中的 Windows AMI

使用以下 get-parameters-by-path 命令。`--path` 参数的值因 Windows AMI 而异。

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-windows-latest \
    --query "Parameters[].Name"
```

PowerShell

列出当前 Amazon 区域中的 Windows AMI

使用 Get-SSMParametersByPath cmdlet。

```
Get-SSMParametersByPath `
    -Path "/aws/service/ami-windows-latest" | `
    Sort-Object Name | Format-Table Name
```

使用公有参数启动实例

要在启动实例时指定此公有参数，请使用以下语法 `resolve:ssm:`*`public-parameter`*，其中 `resolve:ssm` 是标准前缀，*`public-parameter`* 是公有参数的路径和名称。

Amazon CLI

使用公有参数启动实例

使用 run-instances 命令和 `--image-id` 选项。此示例为该映像 ID 指定了一个 Systems Manager 公有参数，以使用最新版本的 Amazon Linux 2023 AMI 启动实例

```
--image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64
```

PowerShell

使用公有参数启动实例

将 New-EC2Instance cmdlet 与 `-ImageId` 参数结合使用。此示例为该映像 ID 指定了一个 Systems Manager 公有参数，以使用最新版本适用于 Windows Server 2022 的 AMI 启动实例。

```
-ImageId "resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base"
```

有关使用 Systems Manager 参数的更多示例，请参阅 Query for the latest Amazon Linux AMI IDs Using Amazon Systems Manager Parameter Store 和 Query for the Latest Windows AMI Using Amazon Systems Manager Parameter Store。

Amazon Web Services Marketplace 中适用于 Amazon EC2 实例的付费 AMI

付费 AMI 是在 Amazon Web Services Marketplace 中列出销售的 AMI。Amazon Web Services Marketplace 是一个在线商店，您可以从中购买在 Amazon 上运行的软件，包括可用来启动 EC2 实例的 AMI。Amazon Web Services Marketplace AMI 分为各种类别（如开发人员工具），您可以根据自己的要求查找产品。有关 Amazon Web Services Marketplace 的更多信息，请参阅 Amazon Web Services Marketplace 网站。

您可以在 Amazon Web Services Marketplace 中从第三方购买 AMI，包括具有 Red Hat 等组织的服务合同的 AMI。您还可以创建 AMI 并在 Amazon Web Services Marketplace 中将其出售给其他 Amazon EC2 用户。如果遵循一些简单的指导，为公共使用构建安全、可靠、可用的 AMI 的过程可以很简单。有关如何创建和使用共享 AMI 的信息，请参阅了解 Amazon EC2 中共享 AMI 的使用情况。

从付费 AMI 启动实例与从任何其他 AMI 启动实例的方式相同。不需要额外参数。实例根据 AMI 拥有者设置的费率以及相关 Web 服务的标准使用费（例如，在 Amazon EC2 中运行 m5.small 实例类型的小时费率）来收费。还可能需要支付其他税款。付费 AMI 拥有者可以确认是否使用该付费 AMI 启动特定实例。

> **Important**
>
> Amazon DevPay 不再接受新的卖家或产品。Amazon Web Services Marketplace 现在是通过 Amazon 销售软件和服务的统一电子商务平台。有关如何从 Amazon Web Services Marketplace 部署和销售软件的信息，请参阅在 Amazon Marketplace 中出售。Amazon Web Services Marketplace 支持受 Amazon EBS 支持的 AMI。

内容

- 在 Amazon Web Services Marketplace 中销售您的 AMI
- 查找付费 AMI
- 在 Amazon Web Services Marketplace 中购买付费 AMI
- 从您的实例中检索 Amazon Web Services Marketplace 产品代码
- 为支持的 Amazon Web Services Marketplace 产品使用付费支持
- 付费和支持 AMI 的账单
- 管理您的 Amazon Web Services Marketplace 订阅

在 Amazon Web Services Marketplace 中销售您的 AMI

您可以使用 Amazon Web Services Marketplace 销售 AMI。Amazon Web Services Marketplace 提供组织有序的购物体验。此外，Amazon Web Services Marketplace 还支持 Amazon 功能，如 Amazon EBS-backed AMI、预留实例和竞价型实例。

有关如何在 Amazon Web Services Marketplace 中出售 AMI 的信息，请参阅在 Amazon Marketplace 中出售。

查找付费 AMI

付费 AMI 是可供购买的亚马逊机器映像（AMI）。付费 AMI 也有产品代码。您可以查找可通过 Amazon Web Services Marketplace 购买的 AMI。

Console

查找付费 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMIs。
3. 对于第一个筛选条件，选择公有映像。
4. 请执行以下操作之一：
   - 如果您知道产品代码，请选择 Product code（产品代码），再选择 =，然后输入产品代码。
   - 如果您不知道产品代码，请在搜索栏中指定以下筛选条件：所有者别名=aws-marketplace。根据需要指定其他筛选条件。
5. 保存 AMI 的 ID。

Amazon CLI

查找付费 AMI

使用以下 describe-images 命令。

```
aws ec2 describe-images --owners aws-marketplace
```

输出中包含大量映像。您可以通过指定筛选条件来帮助确定所需的 AMI。找到 AMI 后，请在以下命令中指定其 ID 来获取产品代码。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].ProductCodes[].ProductCodeId
```

下面是示例输出。

```
[
    "cdef1234abc567def8EXAMPLE"
]
```

如果您知道产品代码，可以按产品代码筛选结果。此示例返回具有指定产品代码的最新 AMI。

```
aws ec2 describe-images \
```

```
    --filters "Name=product-code,Values=cdef1234abc567def8EXAMPLE" \
    --query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

PowerShell

查找付费 AMI

使用 Get-EC2Image cmdlet。

```
Get-EC2Image -Owner aws-marketplace
```

输出中包含大量映像。您可以通过指定筛选条件来帮助确定所需的 AMI。找到 AMI 后，请在以下命令中指定其 ID 来获取产品代码。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ProductCodes
```

下面是示例输出。

```
ProductCodeId             ProductCodeType
-------------             ---------------
cdef1234abc567def8EXAMPLE marketplace
```

如果您知道产品代码，可以按产品代码筛选结果。此示例返回具有指定产品代码的最新 AMI。

```
(Get-EC2Image -Owner aws-marketplace -Filter @{"Name"="product-
code";"Value"="cdef1234abc567def8EXAMPLE"} | sort CreationDate -Descending | Select-
Object -First 1).ImageId
```

在 Amazon Web Services Marketplace 中购买付费 AMI

必须先注册（购买）付费 AMI，然后才能使用该 AMI 启动 Amazon EC2 实例。

通常情况下，付费 AMI 的卖方会为您提供 AMI 的相关信息，包括其价格以及购买网站链接。单击该链接时，首先会提示您登录 Amazon，然后可以购买 AMI。

使用控制台购买付费 AMI

可以使用 Amazon EC2 启动向导购买付费 AMI。有关更多信息，请参阅从 Amazon Web Services Marketplace AMI 中启动 Amazon EC2 实例。

使用 Amazon Web Services Marketplace 订阅产品

要使用 Amazon Web Services Marketplace，必须拥有 Amazon Web Services 账户。要从 Amazon Web Services Marketplace 产品启动实例，必须注册以使用 Amazon EC2 服务，并且必须订阅从中启动实例的产品。您可以使用以下方法之一在 Amazon Web Services Marketplace 中订阅产品：

- Amazon Web Services Marketplace 网站：您可以使用一键部署功能快速启动预配置的软件。有关更多信息，请参阅 AMI-based products in Amazon Web Services Marketplace。
- Amazon EC2 启动向导：您可以直接从向导搜索 AMI 并启动实例。有关更多信息，请参阅 从 Amazon Web Services Marketplace AMI 中启动 Amazon EC2 实例。

从您的实例中检索 Amazon Web Services Marketplace 产品代码

可以使用实例元数据检索实例的 Amazon Web Services Marketplace 产品代码。如果实例具有产品代码，则 Amazon EC2 将返回产品代码。有关检索元数据的更多信息，请参阅 访问 EC2 实例的实例元数据。

IMDSv2

Linux

在 Linux 实例上运行以下命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/product-codes
```

Windows

在 Windows 实例上运行以下 cmdlet。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/product-codes
```

IMDSv1

Linux

在 Linux 实例上运行以下命令。

```
curl http://169.254.169.254/latest/meta-data/product-codes
```

Windows

在 Windows 实例上运行以下命令。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/product-codes
```

为支持的 Amazon Web Services Marketplace 产品使用付费支持

Amazon EC2 还使开发人员可以为软件（或派生 AMI）提供支持。开发人员可以创建您可注册使用的支持产品。在注册支持产品的过程中，开发人员会为您提供产品代码，您必须将该代码与您自己的 AMI 关联起来。这样，开发人员就能确认您的实例有获取支持的权限。此外，还能确保您在运行产品实例时，按照开发人员指定的产品使用条款付费。

限制

- 设置产品代码属性后，将不能更改或移除该属性。
- 不能将支持产品用于预留实例。通常情况下，您需按支持产品卖方指定的价格付费。

Amazon CLI

将产品代码关联到 AMI

使用 modify-image-attribute 命令。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --product-codes "cdef1234abc567def8EXAMPLE"
```

PowerShell

将产品代码关联到 AMI

使用 Edit-EC2ImageAttribute cmdlet。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ProductCode "cdef1234abc567def8EXAMPLE"
```

付费和支持 AMI 的账单

在每个月月底，您会收到一封电子邮件，邮件里注明了该月因使用任何付费和受支持的 AMI 所产生的信用卡付费金额情况。这个账单与您的常规 Amazon EC2 账单是分开的。有关更多信息，请参阅 Amazon Web Services Marketplace 买家指南中的为产品付费。

管理您的 Amazon Web Services Marketplace 订阅

在 Amazon Web Services Marketplace 网站上，您可以检查订阅详细信息，查看供应商的使用说明，管理订阅等。

查看 订阅详细信息

检查订阅详细信息

1. 登录 Amazon Web Services Marketplace。
2. 选择 Your Marketplace Account。
3. 选择 Manage your software subscriptions。
4. 会列出当前所有订阅。选择使用说明以查看使用产品的特定说明，例如，用于连接到运行中的实例的用户名。

取消订阅

> **Note**
>
> - 取消订阅不会终止使用该 AMI 启动的实例。我们将继续对您正在运行的实例计费，直至其被终止。您必须终止使用该 AMI 启动的所有实例才能停止对该订阅付费。
> - 取消了订阅之后，您无法再从该 AMI 启动任何实例。要再次使用该 AMI，您需要在 Amazon Web Services Marketplace 网站上或通过 Amazon EC2 控制台中的启动向导重新订阅它。

取消 Amazon Web Services Marketplace 订阅

1. 要停止对订阅计费，请确保您已终止从该订阅运行的所有实例。

   > **Warning**
   >
   > 终止实例是永久且不可逆转的。
   > 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
   > 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

   a. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   b. 在导航窗格中，选择 Instances (实例)。
   c. 选择相应实例，然后依次选择实例状态、终止（删除）实例。
   d. 当系统提示您确认时，选择终止（删除）。
2. 登录到 Amazon Web Services Marketplace，选择 Your Marketplace Account（您的 Marketplace 账户），然后选择 Manage your software subscriptions（管理您的软件订阅）。
3. 选择 Cancel subscription。会提示您确认取消。

Amazon EC2 AMI 生命周期

亚马逊机器映像（AMI）包含了设置和启动实例所需的软件配置。在启动实例时，您必须指定 (AMI)。您也可以使用 Amazon 提供的 AMI，也可以创建自己的 AMI。该 AMI 必须位于要从中启动实例的 Amazon Web Services 区域。

AMI 的生命周期包括创建、复制、弃用、禁用和删除（注销）AMI。

创建 AMI。您可以使用 Amazon 提供的 AMI 来启动实例，也可以根据自己的需要创建自定义 AMI。要创建自定义 AMI，请从某个现有的 AMI 启动一个实例，对其进行自定义（例如，安装软件和配置操作系统设置），然后利用该实例创建一个 AMI。任何实例自定义都将保存到此新 AMI 中，从而确保利用您的新 AMI 启动的实例包含这些自定义。

复制 AMI。您只能在 AMI 所在的 Amazon Web Services 区域使用该 AMI 启动实例。如果需要在多个区域启动具有相同配置的实例，请将该 AMI 复制到其他区域。

弃用 AMI。要将 AMI 标记为已取代或已过时，您可以设置立即弃用或某个未来弃用日期。已弃用的 AMI 会从 AMI 列表中隐藏，但如果用户和服务知道 AMI ID，则可以继续使用已弃用的 AMI。

禁用 AMI。要暂时阻止使用 AMI，可以禁用该 AMI。禁用某个 AMI 之后，其无法用于启动新实例。不过，如果您重新启用该 AMI，其可用于再次启动实例。请注意，禁用 AMI 不会影响已经从中启动的现有实例。

注销（删除）AMI。不再需要某个 AMI 时，可以将其注销，防止将其用于启动新实例。如果 AMI 符合某个留存规则，则会转移至回收站，并且可在留存期届满之前从回收站还原，但在留存期届满后将被永久删除。如果不符合任何留存规则，则会立即被永久删除。请注意，注销 AMI 不会影响利用该 AMI 启动的现有实例。

自动化 AMI 生命周期。您可以使用 Amazon Data Lifecycle Manager 来自动创建、保留、复制、弃用和取消注册 Amazon EBS-backed AMI 及其备份快照。您还可以使用 EC2 Image Builder 自动创建、管理和部署自定义 AMI。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Automate backups with Amazon Data Lifecycle Manager和 EC2 Image Builder 用户指南。

内容

- 创建 Amazon EBS-backed AMI
- 创建 Amazon S3 支持的 AMI
- 使用 Windows Sysprep 创建 Amazon EC2 AMI
- 复制 Amazon EC2 AMI
- 使用 S3 存储和还原 AMI
- 使用 AMI 世系跟踪 AMI 的源
- 使用 AMI 水印来跟踪和识别 AMI
- 管理和监控 AMI 使用情况
- 弃用 Amazon EC2 AMI
- 禁用 Amazon EC2 AMI
- 取消注册 Amazon EC2 AMI

创建 Amazon EBS-backed AMI

您可以从 Amazon EC2 实例或从 Amazon EC2 实例的根卷快照创建自己的 Amazon EBS-backed AMI。

要从实例创建 Amazon EBS-backed AMI，请先使用现有的 Amazon EBS-backed AMI 启动一个实例。此 AMI 可以是从 Amazon Web Services Marketplace 获得的 AMI，可以是使用 VM Import/Export 创建的 AMI，也可以是能够访问的任何其他 AMI。自定义满足特定要求的实例后，创建新的 AMI 并加以注册。然后，即可使用新的 AMI 启动具有自定义项的新实例。

下述过程适用于由加密的 Amazon Elastic Block Store (Amazon EBS) 卷（包括根卷）支持的 Amazon EC2 实例，也适用于未加密卷。

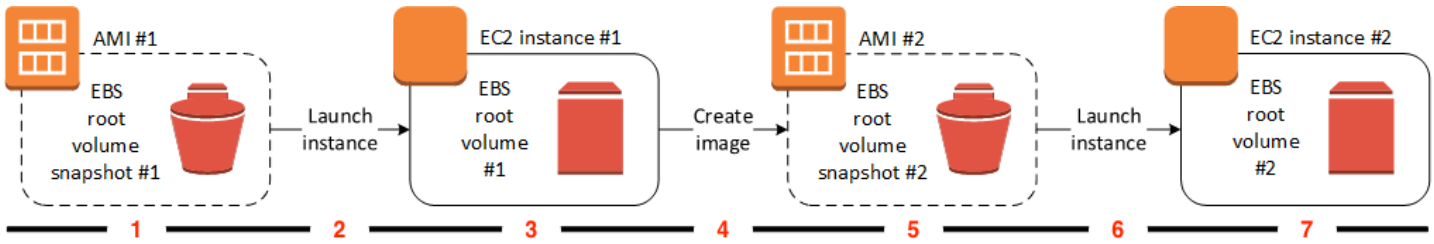
用于 Amazon S3 支持的 AMI 的 AMI 创建过程有所不同。有关更多信息，请参阅 创建 Amazon S3 支持的 AMI。

内容

- 关于从实例创建 AMI 的概述
- 从实例创建 AMI
- 从快照创建 AMI

关于从实例创建 AMI 的概述

以下图表概述了从运行中的 EC2 实例创建 Amazon EBS-backed AMI 的流程：从现有 AMI 开始，启动实例，自定义该实例，从该实例创建新 AMI，并最终启动新 AMI 的实例。图中的数字与以下描述中的数字匹配。



1 – AMI #1：从现有的 AMI 开始

查找类似于您要创建的 AMI 的现有 AMI。这可以是您从 Amazon Web Services Marketplace 获得的 AMI、您使用 VM Import/Export 创建的 AMI 或您可以访问的任何其他 AMI。您将根据自己的需求自定义此 AMI。

在图中，EBS 根卷快照 #1 表示 AMI 为 Amazon EBS-backed AMI，并且有关根卷的信息存储在此快照中。

2 – 从现有 AMI 启动实例

配置 AMI 的方法是从您希望作为新 AMI 基础的 AMI 启动实例，然后自定义该实例（在图中以 3 表示）。然后，您将创建一个包含自定义项的新 AMI（在图中以 4 表示）。

3 – EC2 实例 #1：自定义实例

连接到您的实例并根据您的需求对其进行自定义。您的新 AMI 将包含这些自定义项。

您可以对您的实例执行以下任何操作，以便自定义该实例：

- 安装软件和应用程序
- 复制数据
- 通过删除临时文件和对您的硬盘进行碎片整理来缩短启动时间
- 附加其他 EBS 卷

4 – 创建映像

当您从实例中创建 AMI 时，Amazon EC2 先切断实例的电源再创建 AMI，以确保创建过程中实例上的所有内容均停止并保持一致状态。如果您确信您的实例处于适合 AMI 创建的一致状态，则可以告知 Amazon EC2 不断电和重启实例。一些文件系统 (例如 XFS) 可以冻结和解冻活动，因此能在不重启实例的情况下安全创建映像。

在 AMI 创建过程中，Amazon EC2 会创建您实例的根卷和附加到您实例的任何其他 EBS 卷的快照。在注销 AMI 并删除快照之前，您需要支付快照的费用。如果有任何附加到实例的卷进行了加密，则新 AMI 只会在支持 Amazon EBS 加密的实例上成功启动。

根据卷的大小，可能需要几分钟才能完成 AMI 创建过程 (有时长达 24 小时)。您可能会发现先创建卷的快照，然后再创建 AMI 后会更高效。这样，创建 AMI 时就只需创建小的增量快照，且创建过程完成得更快 (快照创建的总时间保持不变)。

5 – AMI #2：新 AMI

该过程完成之后，您便具有从实例的根卷创建的新 AMI 和快照（快照 #2）。如果除了根卷之外，您还向实例添加了实例存储卷或 EBS 卷，则新 AMI 的块设备映射包含这些卷的信息。

Amazon EC2 自动为您注册 AMI。

6 – 从新 AMI 启动实例

您可以使用新 AMI 启动实例。

7 – EC2 实例 #2：新实例

当您使用新 AMI 启动实例时，Amazon EC2 会使用快照为实例的根卷创建新 EBS 卷。如果您在自定义实例时添加了实例存储卷或 EBS 卷，则新 AMI 的块设备映射包含这些卷的信息，并且您从新 AMI 启动的实例的块设备映射自动包含这些卷的信息。新实例的块设备映射中指定的实例存储卷是新的，不包含用于创建 AMI 的实例的实例存储卷中的任何数据。EBS 卷上的数据会持久保留。有关更多信息，请参阅Amazon EC2 实例上卷的块设备映射。

当您从由 EBS 支持的 AMI 创建新实例时，应该先初始化其根卷及任何额外的 EBS 存储，然后再将其投入生产。有关更多信息，请参阅《Amazon EBS 用户指南》中的初始化 Amazon EBS 卷。

从实例创建 AMI

如果您有现有实例，则可以通过该实例创建 AMI。

Console

创建 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择要从中创建 AMI 的实例，然后选择 Actions（操作）、Image and templates（映像和模板）、Create image（创建映像）。

   > **Tip**
   > 如果该选项处于禁用状态，则表明您的实例不是由 Amazon EBS 支持的实例。

4. 在 Create image（创建映像）页面上，请指定以下信息：

   a. 对于 Image name（映像名称），请输入映像的唯一名称，最多 127 个字符。

   b. 对于 Image description（映像描述），输入映像的可选描述，最多 255 个字符。

   c. 对于重启实例，请保持选中该复选框（默认），或将其清除。

   - 如果选择重启实例，则 Amazon EC2 创建新 AMI 时，会重启该实例，以便其可以在数据处于静态状态时拍摄附加卷的快照，以确保一致的状态。
   - 如果清除重启实例，则 Amazon EC2 创建新 AMI 时，不会关闭和重启实例。

> **Warning**
>
> 如果您清除重启实例，则我们无法保证所创建映像的文件系统完整性。

d. Instance volumes（实例卷）– 您可以修改根卷，并添加其他 Amazon EBS 和实例存储卷，如下所示：

   i. 根卷在第一行中定义。

      - 要更改根卷的大小，请在 Size (尺寸) 中输入所需的值。
      - 如果选择 Delete on Termination (终止时删除)，则当您终止从此 AMI 创建的实例时，将删除 EBS 卷。如果取消选择 Delete on Termination (终止时删除)，则当您终止实例时，不会删除 EBS 卷。有关更多信息，请参阅实例终止时保留数据。

   ii. 要添加 EBS 卷，请选择 Add New Volume（添加新卷）（这将添加一个新行）。对于存储类型，选择 EBS，并填写行中的字段。当您从新的 AMI 启动实例时，额外的卷会自动附加到该实例。您必须格式化并挂载空卷。您必须挂载基于快照的卷。

   iii. 要添加实例存储卷，请参阅 将实例存储卷添加到 Amazon EC2 AMI。当您从新的 AMI 启动实例时，这些额外的卷会自动初始化并挂载。这些卷不包含您的 AMI 所基于的运行实例的实例存储卷上的数据。

e. 快照目标：如果实例卷位于支持 EBS 本地快照的本地区域中，请选择创建 AMI 快照的位置：

   - Amazon Web Services 区域：在卷的本地区域的父区域中创建快照。
   - Amazon 本地区域：在与卷相同的本地区域中创建快照。

> **Note**
>
> 此选项仅在支持 EBS 本地快照的本地区域中出现，并且仅当您的实例是在本地区域中创建时才会出现。如果卷位于某个区域中，则不会出现此选项，快照会自动在与卷相同的区域中创建。有关更多信息，请参阅《Amazon EBS User Guide》中的 Local snapshots in Local Zones。

> **Important**
>
> 实例卷的所有快照都必须位于同一位置。验证现有快照的位置。如果任何现有快照的位置与您选择的目标不同，AMI 创建将失败。

f. 准备好创建 AMI 时，选择 Create image（创建映像）。

5. 若要在创建 AMI 时查看其状态，请执行以下操作：

   a. 在导航窗格中，选择 AMI。

   b. 将筛选条件设置为 Owned by me（我拥有的），然后在列表中查找您的 AMI。

      最初，状态是 `pending`，但过几分钟就会变成 `available`。

6. （可选）若要查看为新 AMI 创建的快照，请执行以下操作：

   a. 记下您在上一步骤中找到的 AMI ID。

   b. 在导航窗格中，选择快照。

   c. 将筛选条件设置为 Owned by me（我拥有的），然后在 Description（描述）列中查找具有新 AMI ID 的快照。

      您从此 AMI 启动实例时，Amazon EC2 使用此快照创建实例的根卷。

Amazon CLI

创建 AMI

使用 create-image 命令。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-web-server" \
    --description "My web server image" \
    --no-reboot
```

PowerShell

创建 AMI

使用 New-EC2Image cmdlet。

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-web-server" `
    -Description "My web server image" `
    -NoReboot $true
```

从快照创建 AMI

如果您有实例根卷的快照，则可以通过此快照创建 AMI。

> **Note**
>
> 在大多数情况下，适用于 Windows、Red Hat、SUSE 和 SQL Server 的 AMI 需要在 AMI 上提供正确的许可信息。有关更多信息，请参阅 了解 AMI 账单信息。从快照创建 AMI 时，RegisterImage 操作会从快照的元数据获取正确的账单信息，但这需要存在相应的元数据。要验证是否应用了正确的账单信息，请检查新 AMI 上的平台详细信息字段。如果该字段为空或不匹配预期的操作系统代码（例如 Windows、Red Hat、SUSE 或 SQL），则 AMI 创建失败，您应丢弃 AMI 并按照从实例创建 AMI中的说明进行操作。

Console

从快照创建 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择快照。
3. 选择从中创建 AMI 的快照并选择 Actions（操作）、Create image from snapshot（从快照创建映像）。
4. 在从快照创建映像页面上，请指定以下信息：
   a. 对于 Image name（镜像名称），请输入镜像的描述性名称。
   b. 对于 Description（描述），请输入镜像的简短描述。
   c. 对于 Architecture（架构），请选择镜像架构。对于 32 位选择 i386，对于 64 位选择 x86_64，对于 64 位 ARM 选择 arm64，或者对于 64 位 macOS 选择 x86_64。
   d. 对于根设备名称，请输入要用于根卷的设备名称。有关更多信息，请参阅 Amazon EC2 实例上卷的设备名称。

e. 对于 Virtualization type（虚拟化类型），请选择从此 AMI 启动的实例要使用的虚拟化类型。有关更多信息，请参阅 虚拟化类型。

f. （仅适用于半虚拟化）对于 Kernel ID（内核 ID），选择镜像的操作系统内核。如果您使用的是实例的根卷快照，请选择与原始实例相同的内核 ID。如果您不确定，请使用默认内核。

g. （仅适用于半虚拟化）对于 RAM disk ID（RAM 磁盘 ID），请选择镜像的 RAM 磁盘。如果您选择了一个特定内核，则您可能需要选择带有可支持该内核的驱动程序的某个特定 RAM 磁盘。

h. 对于启动模式，选择映像的启动模式，或者选择使用默认值，以便在使用此 AMI 启动实例时，它会以该实例类型支持的启动模式启动。有关更多信息，请参阅 设置 Amazon EC2 AMI 的启动模式。

i. （可选）在块设备映射部分中，自定义根卷并添加其它数据卷。

   对于每个卷，您可以指定大小、类型、性能特征、终止时删除的行为以及加密状态。对于根卷，其大小不能小于快照的大小。对于卷类型，默认选择为通用型 SSD gp3。

j. （可选）在标签下，您可以向新的 AMI 添加一个或多个标签。要添加标签，请选择 Add tag (添加标签)，然后输入该标签的键和值。对每个标签重复此操作。

k. 准备好创建 AMI 时，选择 Create image（创建映像）。

5. （仅适用于 Windows、Red Hat、SUSE 和 SQL Server）要验证是否应用了正确的账单信息，请检查新 AMI 上的平台详细信息字段。如果该字段为空或不匹配预期的操作系统代码（例如 Windows 或 Red Hat），则 AMI 创建失败，您应丢弃 AMI 并按照从实例创建 AMI中的说明进行操作。

Amazon CLI

使用 Amazon CLI 从快照创建 AMI

使用 register-image 命令。

```
aws ec2 register-image \
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f}
```

PowerShell

使用 PowerShell 从快照创建 AMI

使用 Register-EC2Image cmdlet。

```
$block = @{SnapshotId=snap-0db2cf683925d191f}
Register-EC2Image `
    -Name my-image `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block}
```

创建 Amazon S3 支持的 AMI

您在启动实例时指定的 AMI 决定根卷的类型

要创建 Amazon S3 支持的 Linux AMI，请通过从 Amazon S3 支持的现有 Linux AMI 启动的实例开始进行。根据您自己的需要自定义该实例之后，请捆绑卷并注册新 AMI，您可以使用该 AMI 启动具有这些自定义项的新实例。

您无法创建 Amazon S3 支持的 Windows AMI，因为 Windows AMI 不支持根卷的实例存储。

> **Important**
>
> 只有以下实例类型支持将实例存储卷作为根卷，并且需要 Amazon S3 支持的 AMI：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

用于 Amazon EBS-backed AMI 的 AMI 创建过程有所不同。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。

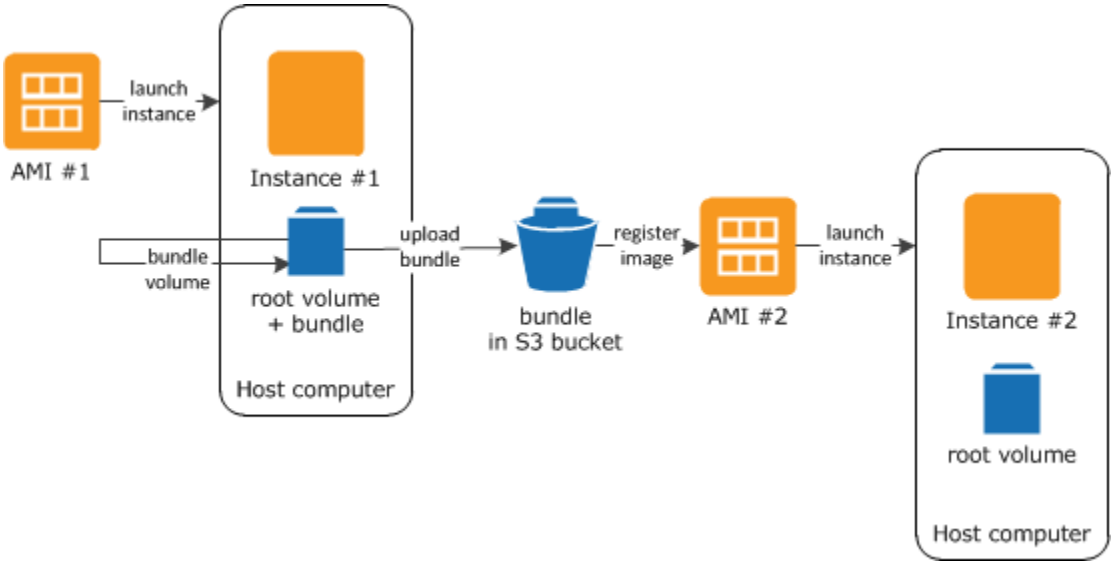
内容

- AMI 创建概述
- 先决条件
- 从 Amazon Linux 实例创建 AMI
- 设置 Amazon EC2 AMI 工具
- Amazon EC2 AMI 工具参考

- 将 Amazon S3 支持的 AMI 转换为 EBS-backed AMI

AMI 创建概述

下图总结了从具有实例存储根卷的实例创建 AMI 的过程。



首先，从类似于您要创建的 AMI 的 AMI 启动实例。您可以连接到您的实例并进行自定义。根据您的需要设置好实例后，您可以捆绑它。完成捆绑过程需要几分钟的时间。该过程完成之后，您会得到一个捆绑，该捆绑由映像清单 (`image.manifest.xml`) 以及包含根卷模板的文件 (`image.part.xx`) 组成。接下来，将该捆绑上传到 Amazon S3 存储桶，然后注册您的 AMI。

> **Note**
>
> 要将对象上载到 Amazon S3 支持的 Linux AMI 的 S3 存储桶，必须为该存储桶启用 ACL。否则，Amazon EC2 将无法在要上载的对象上设置 ACL。如果您的目标存储桶使用存储桶拥有者强制执行的 S3 对象所有权设置，则这将不起作用，因为 ACL 已禁用。有关更多信息，请参阅为您的存储桶控制对象所有权和禁用 ACL。

当您使用新 AMI 启动实例时，我们会使用您上传到 Amazon S3 的捆绑为实例创建根卷。Amazon S3 中的捆绑使用的存储空间会使您的账户产生费用，直到将其删除。有关更多信息，请参阅 取消注册 Amazon EC2 AMI。

如果除了根卷之外，您还向实例添加实例存储卷，则新 AMI 的块设备映射包含这些卷的信息，并且您从新 AMI 启动的实例的块设备映射自动包含这些卷的信息。有关更多信息，请参阅 Amazon EC2 实例上卷的块设备映射。

先决条件

必须先完成以下任务才能创建 AMI：

- 安装 AMI 工具。有关更多信息，请参阅设置 Amazon EC2 AMI 工具。
- 安装 Amazon CLI。有关更多信息，请参阅 Amazon CLI 入门。
- 请确保您具有用于捆绑的 S3 存储桶，并且您的存储桶已启用 ACL。有关配置 ACL 的更多信息，请参阅 配置 ACL。
  - 要使用 Amazon Web Services 管理控制台创建 S3 存储桶，请打开 Amazon S3 控制台（地址为 https://console.aws.amazon.com/s3/），然后选择创建存储桶。
  - 要使用 Amazon CLI 创建 S3 存储桶，请使用 mb 命令。如果您安装的 AMI 工具版本为 1.5.18 或更高版本，则还可以使用 `ec2-upload-bundle` 命令创建 S3 存储桶。有关更多信息，请参阅 ec2-upload-bundle。
- 确保捆绑包中的文件未在 S3 存储桶中加密。如果您需要对 AMI 进行加密，则可以改用 EBS 支持的 AMI。有关更多信息，请参阅 将加密与 EBS 支持的 AMI 结合使用。
- 确保您拥有您的 Amazon 账户 ID。有关更多信息，请参阅 Amazon 账户管理参考指南中的查看 Amazon Web Services 账户 身份。
- 确保您拥有凭证以使用 Amazon CLI。有关更多信息，请参阅《Amazon Command Line Interface User Guide》中的 Authentication and access credentials for the Amazon CLI。
- 确保您拥有 X.509 证书以及相应的私有密钥。
  - 如果您需要创建 X.509 证书，请参阅管理签名证书。X.509 证书和私有密钥用于加密和解密您的 AMI。
  - [中国（北京）] 使用 `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` 证书。
  - [Amazon GovCloud（美国西部）] 使用 `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-gov.pem` 证书。
- 连接到您的 实例并对其进行自定义。例如，您可以安装软件和应用程序、复制数据、删除临时文件及修改 Linux 配置。

从 Amazon Linux 实例创建 AMI

以下过程介绍如何从运行 Amazon Linux 1 的具有实例存储根卷的实例创建 AMI。它们可能不适用于运行其他 Linux 发行版的实例。

准备使用 AMI 工具 (仅限 HVM 实例)

1. AMI 工具需要有 GRUB Legacy，才能正确启动。使用以下命令安装 GRUB：

```
[ec2-user ~]$ sudo yum install -y grub
```

2. 使用以下命令安装分区管理程序包：

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

从具有实例存储根卷的 Amazon Linux 实例创建 AMI

此过程假设您满足先决条件中的先决条件。

在以下示例中，将每个*#######*替换为您自己的信息。

1. 将您的凭证上传到您的实例。我们使用这些凭证确保只有您和 Amazon EC2 才能访问您的 AMI。

   a. 在您的实例上为凭证创建临时目录，如下所示：

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   这使您可以从创建的映像中排除您的凭证。

   b. 使用安全复制工具 (如 scp) 将 X.509 证书和对应的私有密钥从您的计算机复制到实例上的 `/tmp/cert` 目录。以下 `-i` *`my-private-key`*`.pem` 命令中的 scp 选项是您用于通过 SSH 连接到实例的私有密钥，而不是 X.509 私有密钥。例如：

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /
   path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  685     0.7KB/s   00:00
   ```

   此外，由于这些是纯文本文件，所以您可以在文本编辑器中打开证书和密钥，并将其内容复制到 `/tmp/cert` 中的新文件。

2. 通过从您的实例内部运行 ec2-bundle-vol 命令，准备捆绑包以便上传到 Amazon S3。请务必指定 `-e` 选项以排除用于存储您的凭证的目录。默认情况下，捆绑过程不包括可能包含敏感信息

的文件。这些文件包括 *.sw、*.swo、*.swp、*.pem、*.priv、*id_rsa*、*id_dsa* *.gpg、*.jks、*/.ssh/authorized_keys 和 */.bash_history。要包括所有这些文件，请使用 --no-filter 选项。要包括其中部分文件，请使用 --include 选项。

> **Important**
>
> 默认情况下，AMI 捆绑过程在表示根卷的 /tmp 目录中创建经过压缩和加密的文件集合。如果您在 /tmp 中没有足够的可用磁盘空间来存储捆绑，则需要使用 -d */path/to/bundle/storage* 选项指定不同的位置来存储捆绑。某些实例会在 /mnt 或 /media/ephemeral0 上装载您可以使用的临时存储，您还可以创建、连接和挂载新 Amazon EBS 卷以存储捆绑。有关更多信息，请参阅《Amazon EBS 用户指南》中的创建 Amazon EBS 卷。

a. 您必须以根用户身份运行 ec2-bundle-vol 命令。对于大部分命令，您可以使用 sudo 获取提升的权限，但是在这种情况下，您应运行 sudo -E su 以保留环境变量。

```
[ec2-user ~]$ sudo -E su
```

请注意，在 bash 提示符下现在将您标识为根用户，并且美元符号已替换为哈希标签，表示您现在处于 root Shell 中：

```
[root ec2-user]#
```

b. 要创建 AMI 捆绑，请如下所示运行 ec2-bundle-vol 命令：

```
[root ec2-user]# ec2-bundle-vol -k /tmp/cert/pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 123456789012 -r x86_64 -e /tmp/cert --
partition gpt
```

> **Note**
>
> 对于中国（北京）和 Amazon GovCloud（美国西部）区域，请使用 --ec2cert 参数并根据先决条件指定证书。

创建映像可能需要几分钟时间。该命令完成后，您的 /tmp (非默认) 目录会包含捆绑 (image.manifest.xml 以及多个 image.part.*xx* 文件)。

c. 从 root Shell 退出。

```
[root ec2-user]# exit
```

3. （可选）要添加更多实例存储卷，请在 image.manifest.xml 文件中为您的 AMI 编辑块储存设备映射。有关更多信息，请参阅Amazon EC2 实例上卷的块设备映射。

a. 创建 image.manifest.xml 文件的备份。

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

b. 重新设置 image.manifest.xml 文件的格式，使其更易于阅读和编辑。

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/
image.manifest.xml
```

c. 使用文本编辑器编辑 image.manifest.xml 中的块储存设备映射。以下示例显示了 ephemeral1 实例存储卷的一个新条目。

> **Note**
>
> 有关排除的文件的列表，请参阅ec2-bundle-vol。

```
    <block_device_mapping>
      <mapping>
        <virtual>ami</virtual>
        <device>sda</device>
      </mapping>
      <mapping>
        <virtual>ephemeral0</virtual>
        <device>sdb</device>
      </mapping>
      <mapping>
        <virtual>ephemeral1</virtual>
        <device>sdc</device>
      </mapping>
```

```
      <mapping>
        <virtual>root</virtual>
        <device>/dev/sda1</device>
      </mapping>
    </block_device_mapping>
```

d. 保存 image.manifest.xml 文件并退出文本编辑器。

4. 要将捆绑上传到 Amazon S3，请如下所示运行 ec2-upload-bundle 命令。

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name -
m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

> **Important**
>
> 要在US East (N. Virginia)之外的区域中注册 AMI，则必须指定带 --region 选项的目标区域和目标区域中已存在的存储桶路径或可在目标区域中创建的唯一存储桶路径。

5. （可选）将捆绑上传到 Amazon S3 之后，您可以使用以下 /tmp 命令将捆绑从实例上的 rm 目录中删除：

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

> **Important**
>
> 如果您在 -d */path/to/bundle/storage* 中使用 Step 2 选项指定了路径，请使用该路径，而不是 /tmp。

6. 要注册您的 AMI，请按以下所示运行 register-image 命令。

```
[ec2-user ~]$ aws ec2 register-image --image-location amzn-s3-demo-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --
virtualization-type hvm
```

> **Important**
>
> 如果您先前为 ec2-upload-bundle 命令指定了某个区域，请为该命令再次指定该区域。

设置 Amazon EC2 AMI 工具

您可以使用 AMI 工具创建和管理 Amazon S3 支持的 Linux AMI。要使用这些工具，必须在 Linux 实例上安装它们。AMI 工具可作为 RPM 提供，也为不支持 RPM 的 Linux 发行版提供 .zip 格式的文件。

使用 RPM 设置 AMI 工具

1. 使用您的 Linux 发行版的程序包管理器 (如 yum) 安装 Ruby。例如：

```
[ec2-user ~]$ sudo yum install -y ruby
```

2. 使用 wget 或 curl 等工具下载 RPM 文件。例如：

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. 使用以下命令验证 RPM 文件的签名：

```
[ec2-user ~]$ rpm -K ec2-ami-tools.noarch.rpm
```

上述命令应指示该文件的 SHA1 和 MD5 哈希值是 OK。如果该命令指示这些哈希值是 NOT OK，请使用以下命令查看该文件的标头 SHA1 和 MD5 哈希值：

```
[ec2-user ~]$ rpm -Kv ec2-ami-tools.noarch.rpm
```

然后，将您的文件的标头 SHA1 和 MD5 哈希值与以下经验证的 AMI 工具哈希值进行比较，以确认文件的真实性：

- 标头 SHA1：a1f662d6f25f69871104e6a62187fa4df508f880
- MD5：9faff05258064e2f7909b66142de6782

如果您的文件的标头 SHA1 和 MD5 哈希值与经验证的 AMI 工具哈希值相匹配，请继续下一步。

4. 使用以下命令安装 RPM：

```
[ec2-user ~]$ sudo yum install ec2-ami-tools.noarch.rpm
```

5. 使用 ec2-ami-tools-version 命令验证您的 AMI 工具安装。

```
[ec2-user ~]$ ec2-ami-tools-version
```

> **Note**
>
> 如果您收到一个加载错误，例如“cannot load such file -- ec2/amitools/version (LoadError)”(无法加载此类文件 -- ec2/amitools/version (LoadError))，请完成下一步骤以将 AMI 工具安装的位置添加到 RUBYLIB 路径。

6. （可选）如果您在上一步中收到了错误，则将您的 AMI 工具的安装位置添加到您的 RUBYLIB 路径中。

   a. 运行以下命令以确定要添加的路径。

   ```
   [ec2-user ~]$ rpm -qil ec2-ami-tools | grep ec2/amitools/version
   /usr/lib/ruby/site_ruby/ec2/amitools/version.rb
   /usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
   ```

   在以上示例中，以前加载错误中的丢失文件位于 /usr/lib/ruby/site_ruby 和 /usr/lib64/ruby/site_ruby。

   b. 将上一步的位置添加到您的 RUBYLIB 路径中。

   ```
   [ec2-user ~]$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/site_ruby
   ```

   c. 使用 ec2-ami-tools-version 命令验证您的 AMI 工具安装。

   ```
   [ec2-user ~]$ ec2-ami-tools-version
   ```

使用 .zip 文件设置 AMI 工具

1. 使用您的 Linux 发行版的程序包管理器安装 Ruby 并解压缩，例如 apt-get。例如：

   ```
   [ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
   ```

2. 使用 wget 或 curl 等工具下载 .zip 文件。例如：

   ```
   [ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
   ```

3. 将文件解压缩到合适的安装目录，如 /usr/local/ec2。

```
[ec2-user ~]$ sudo mkdir -p /usr/local/ec2
$ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
```

请注意，.zip 文件包含文件夹 ec2-ami-tools-*x.x.x*，其中 *x.x.x* 是工具的版本号（例如，ec2-ami-tools-1.5.7）。

4. 将 EC2_AMITOOL_HOME 环境变量设置为工具的安装目录。例如：

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. 将工具添加到您的 PATH 环境变量。例如：

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. 您可以使用 ec2-ami-tools-version 命令验证您的 AMI 工具安装。

```
[ec2-user ~]$ ec2-ami-tools-version
```

管理签名证书

AMI 工具中的某些命令需要签名证书 (也称为 X.509 证书)。您必须创建证书，然后将其上传到 Amazon。例如，您可以使用第三方工具 (例如 OpenSSL) 创建证书。

创建签名证书

1. 安装和配置 OpenSSL。
2. 使用 openssl genrsa 命令创建私有密钥，并将输出保存到 .pem 文件。我们建议您创建 2048 或 4096 位 RSA 密钥。

```
openssl genrsa 2048 > private-key.pem
```

3. 使用 openssl req 命令生成证书。

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -
out certificate.pem
```

要将证书上传到 Amazon，请使用 upload-signing-certificate 命令。

```
aws iam upload-signing-certificate --user-name user-name --certificate-body
 file://path/to/certificate.pem
```

要列出用户的证书，请使用 list-signing-certificates 命令：

```
aws iam list-signing-certificates --user-name user-name
```

要对用户禁用或重新启用签名证书，请使用 update-signing-certificate 命令。以下命令可禁用证书：

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE --
status Inactive --user-name user-name
```

要删除证书，请使用 delete-signing-certificate 命令：

```
aws iam delete-signing-certificate --user-name user-name --certificate-
id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE
```

Amazon EC2 AMI 工具参考

您可以使用 AMI 工具命令创建和管理 Amazon S3 支持的 Linux AMI。要设置这些工具，请参阅设置 Amazon EC2 AMI 工具。

有关访问密钥的信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。

命令

- ec2-ami-tools-version
- ec2-bundle-image
- ec2-bundle-vol
- ec2-delete-bundle
- ec2-download-bundle
- ec2-migrate-manifest
- ec2-unbundle
- ec2-upload-bundle
- AMI 工具的常用选项

ec2-ami-tools-version

描述

描述 AMI 工具的版本。

语法

ec2-ami-tools-version

输出

版本信息。

示例

此示例命令显示所用 AMI 工具的版本信息。

```
[ec2-user ~]$ ec2-ami-tools-version
1.5.2 20071010
```

ec2-bundle-image

描述

通过回环文件中创建的操作系统映像创建 Amazon S3 支持的 Linux AMI。

语法

ec2-bundle-image -c *path* -k *path* -u *account* -i *path* [-d *path*] [--ec2cert *path*] [-r *architecture*] [--productcodes *code1*,*code2*,...] [-B *mapping*] [-p *prefix*]

选项

`-c, --cert` 路径

用户的 PEM 编码 RSA 公有密钥凭证文件。

必需：是

`-k, --privatekey` 路径

指向 PEM 编码 RSA 密钥文件的路径。您需要指定此密钥解开此捆绑包，因此，请将其保存在安全的地方。请注意，不需要在您的 Amazon 账户中注册该密钥。

必需：是

`-u, --user` 账户

用户的 Amazon 账户 ID (不包含破折号)。

必需：是

`-i, --image` 路径

指向待捆绑映像的路径。

必需：是

`-d, --destination` 路径

要在其中创建捆绑的目录。

默认值：`/tmp`

必需：否

`--ec2cert` 路径

用于加密映像清单的 Amazon EC2 X.509 公有密钥凭证的路径。

`us-gov-west-1` 和 `cn-north-1` 区域使用非默认公有密钥凭证，必须随该选项指定该证书的路径。该证书的路径因 AMI 工具的安装方法而异。对于 Amazon Linux，证书位于 `/opt/aws/amitools/ec2/etc/ec2/amitools/`。如果您将来自 RPM 或 ZIP 文件的 AMI 工具安装在了 设置 Amazon EC2 AMI 工具 中，则证书位于 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`。

必需：仅限 `us-gov-west-1` 和 `cn-north-1` 区域。

`-r, --arch` 架构

映像架构。如果您不在命令行上提供架构，则会在绑定开始时提示您输入架构。

有效值：`i386 | x86_64`

必需：否

`--productcodescode1,code2,...`

在注册时附加到映像的产品代码，用逗号隔开。

必需：否

`-B, --block-device-mapping` 映射

定义块储存设备向此 AMI 的实例公开的方式 (如果其实例类型支持指定的设备)。

指定键值对的逗号分隔列表，每个键是虚拟名称，每个值是相应的设备名称。虚拟名称包括：

- `ami` — 实例所看到的根文件系统设备
- `root` — 内核所看到的根文件系统设备
- `swap` — 实例所看到的交换设备
- `ephemeralN` — 第 N 个实例存储卷

必需：否

`-p, --prefix` prefix

捆绑的 AMI 文件的文件名前缀。

默认：映像文件的名称。例如，如果映像路径为 `/var/spool/my-image/version-2/debian.img`，则默认前缀为 `debian.img`。

必需：否

`--kernel` kernel_id

已淘汰。使用 register-image 设置内核。

必需：否

`--ramdisk` ramdisk_id

已淘汰。使用 register-image 设置 RAM 磁盘 (若需要)。

必需：否

输出

描述捆绑过程的阶段和状态的状态消息。

示例

此示例从回环文件中所创建的操作系统映像创建捆绑的 AMI。

```
[ec2-user ~]$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -r x86_64
```

```
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
Created image.part.06
Created image.part.07
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

ec2-bundle-vol

描述

通过对实例根卷的副本进行压缩、加密和签名来创建 Amazon S3 支持的 Linux AMI。

Amazon EC2 将尝试从实例继承产品代码、内核设置、RAM 磁盘设置和块储存设备映射。

默认情况下，捆绑过程不包括可能包含敏感信息的文件。这些文件包括 `*.sw`、`*.swo`、`*.swp`、`*.pem`、`*.priv`、`*id_rsa*`、`*id_dsa*` `*.gpg`、`*.jks`、`*/.ssh/authorized_keys` 和 `*/.bash_history`。要包括所有这些文件，请使用 `--no-filter` 选项。要包括其中部分文件，请使用 `--include` 选项。

有关更多信息，请参阅 创建 Amazon S3 支持的 AMI。

语法

```
ec2-bundle-vol -c path -k path -u account [-d path] [--ec2cert path] [-
r architecture] [--productcodes code1,code2,...] [-B mapping] [--all] [-e
directory1,directory2,...] [-i file1,file2,...] [--no-filter] [-p prefix]
```

[-s *size*] [--[no-]inherit] [-v *volume*] [-P *type*] [-S *script*] [--fstab *path*] [--generate-fstab] [--grub-config *path*]

选项

`-c, --cert` 路径

用户的 PEM 编码 RSA 公有密钥凭证文件。

必需：是

`-k, --privatekey` 路径

用户的 PEM 编码 RSA 密钥文件的路径。

必需：是

`-u, --user` 账户

用户的 Amazon 账户 ID (不包含破折号)。

必需：是

`-d, --destination` destination

要在其中创建捆绑的目录。

默认值：`/tmp`

必需：否

`--ec2cert` 路径

用于加密映像清单的 Amazon EC2 X.509 公有密钥凭证的路径。

`us-gov-west-1` 和 `cn-north-1` 区域使用非默认公有密钥凭证，必须随该选项指定该证书的路径。该证书的路径因 AMI 工具的安装方法而异。对于 Amazon Linux，证书位于 `/opt/aws/amitools/ec2/etc/ec2/amitools/`。如果您将来自 RPM 或 ZIP 文件的 AMI 工具安装在了 设置 Amazon EC2 AMI 工具 中，则证书位于 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`。

必需：仅限 `us-gov-west-1` 和 `cn-north-1` 区域。

`-r, --arch` 架构

映像架构。如果您不在命令行上提供架构，则会在绑定开始时提示您提供架构。

有效值：`i386` | `x86_64`

必需：否

--productcodescode1,code2,...

在注册时附加到映像的产品代码，用逗号隔开。

必需：否

-B, --block-device-mapping 映射

定义块储存设备向此 AMI 的实例公开的方式 (如果其实例类型支持指定的设备)。

指定键值对的逗号分隔列表，每个键是虚拟名称，每个值是相应的设备名称。虚拟名称包括：

- ami — 实例所看到的根文件系统设备
- root — 内核所看到的根文件系统设备
- swap — 实例所看到的交换设备
- ephemeralN — 第 N 个实例存储卷

必需：否

-a, --all

捆绑所有目录，包括远程装载的文件系统上的目录。

必需：否

-e, --exclude directory1,directory2,...

要从捆绑操作中排除的绝对目录路径和文件的列表。此参数覆盖 --all 选项。指定排除时，随此参数列出的目录和子目录将不会随卷捆绑。

必需：否

-i, --include file1,file2,...

要在捆绑操作中包含的文件的列表。因为指定的文件可能包含敏感信息，若不指定则会从 AMI 中排除。

必需：否

--no-filter

如果指定，则我们不会因为文件可能包含敏感信息而将其从 AMI 排除。

必需：否

`-p, --prefix` prefix

捆绑的 AMI 文件的文件名前缀。

默认值：`image`

必需：否

`-s, --size` size

要创建的映像文件的大小，以 MB (1024 * 1024 字节) 为单位。最大大小为 10240 MB。

默认值：10240

必需：否

`--[no-]inherit`

指示映像是否应当继承实例的元数据 (默认为继承)。如果启用 `--inherit` 但实例元数据不可访问，则捆绑将失败。

必需：否

`-v, --volume` 体积

要从中创建捆绑的装载卷的绝对路径。

默认值：根目录 (/)

必需：否

`-P, --partition` type

指示磁盘映像是否应使用分区表。如果不指定分区表类型，则默认使用卷的父块储存设备上使用的类型 (如果适用)，否则默认为 `gpt`。

有效值：`mbr` |`gpt` |`none`

必需：否

`-S, --script` 脚本

将在捆绑前运行的自定义脚本。该脚本必须获得一个参数，即卷的装载点。

必需：否

`--fstab` 路径

要捆绑到映像中的 fstab 的路径。如果未指定，Amazon EC2 将捆绑 /etc/fstab。

必需：否

--generate-fstab

使用 Amazon EC2 提供的 fstab 捆绑卷。

必需：否

--grub-config

将捆绑到映像中的备用 GRUB 配置文件的路径。默认情况下，ec2-bundle-vol 预计克隆的映像上存在 /boot/grub/menu.lst 或 /boot/grub/grub.conf。该选项可让您指定备用 GRUB 配置文件的路径，将会复制该文件以覆盖默认值 (若存在)。

必需：否

--kernel kernel_id

已淘汰。使用 register-image 设置内核。

必需：否

--ramdiskramdisk_id

已淘汰。使用 register-image 设置 RAM 磁盘 (若需要)。

必需：否

输出

描述捆绑的阶段和状态的状态消息。

示例

此示例通过对本机根文件系统进行压缩、加密和签名创建捆绑的 AMI。

```
[ec2-user ~]$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c
 cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
  Copying / into the image file /mnt/image...
  Excluding:
       sys
       dev/shm
       proc
       dev/pts
       proc/sys/fs/binfmt_misc
```

```
      dev
      media
      mnt
      proc
      sys
      tmp/image
      mnt/img-mnt
1+0 records in
1+0 records out
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
...
Created image.part.22
Created image.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
Bundle Volume complete.
```

ec2-delete-bundle

描述

从 Amazon S3 存储中删除指定的捆绑。删除捆绑后，您不能从相应的 AMI 启动实例。

语法

ec2-delete-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* [-t *token*] [--url *url*] [--region *region*] [--sigv *version*] [-m *path*] [-p *prefix*] [--clear] [--retry] [-y]

选项

-b, --bucket 存储桶

包含捆绑的 AMI 的 Amazon S3 存储桶的名称，后跟可选的以“/”分隔的路径前缀

必需：是

`-a, --access-key` access_key_id

Amazon 访问密钥 ID。

必需：是

`-s, --secret-key` secret_access_key

Amazon 秘密访问密钥。

必需：是

`-t, --delegation-token` token

传递到 Amazon 请求的委托令牌。有关更多信息，请参阅《IAM 用户指南》中的临时安全凭证。

必需：仅当使用临时安全凭证时是必需的。

默认：`AWS_DELEGATION_TOKEN` 环境变量的值 (若已设置)。

`--region`区域

要在请求签名中使用的区域。

默认值：`us-east-1`

必需：若使用签名版本 4 则必需

`--sigv`版本

对请求进行签名时要使用的签名版本。

有效值：2 | 4

默认值：4

必需：否

`-m, --manifestpath`

清单文件的路径。

必需：您必须指定 `--prefix` 或 `--manifest`。

`-p, --prefix` prefix

捆绑的 AMI 文件名前缀。请提供完整前缀。例如，如果前缀是 image.img，请使用 `-p image.img` 而不是 `-p image`。

必需：您必须指定 --prefix 或 --manifest。

--clear

删除指定的捆绑之后删除 Amazon S3 存储桶 (若为空)。

必需：否

--retry

在所有 Amazon S3 错误后自动重试，每个操作最多五次。

必需：否

-y, --yes

自动假定所有提示的回答为 yes。

必需：否

输出

Amazon EC2 显示状态消息以指示删除过程的阶段和状态。

示例

此示例从 Amazon S3 删除捆绑。

```
[ec2-user ~]$ ec2-delete-bundle -b amzn-s3-demo-bucket -a your_access_key_id -
s your_secret_access_key
Deleting files:
amzn-s3-demo-bucket/image.manifest.xml
amzn-s3-demo-bucket/image.part.00
amzn-s3-demo-bucket/image.part.01
amzn-s3-demo-bucket/image.part.02
amzn-s3-demo-bucket/image.part.03
amzn-s3-demo-bucket/image.part.04
amzn-s3-demo-bucket/image.part.05
amzn-s3-demo-bucket/image.part.06
Continue? [y/n]
y
Deleted amzn-s3-demo-bucket/image.manifest.xml
Deleted amzn-s3-demo-bucket/image.part.00
Deleted amzn-s3-demo-bucket/image.part.01
```

```
Deleted amzn-s3-demo-bucket/image.part.02
Deleted amzn-s3-demo-bucket/image.part.03
Deleted amzn-s3-demo-bucket/image.part.04
Deleted amzn-s3-demo-bucket/image.part.05
Deleted amzn-s3-demo-bucket/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

描述

从 Amazon S3 存储下载指定的 Amazon S3 支持的 Linux AMIs。

语法

ec2-download-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* -k *path* [--url *url*] [--region *region*] [--sigv *version*] [-m *file*] [-p *prefix*] [-d *directory*] [--retry]

选项

-b, --bucket 存储桶

捆绑所在的 Amazon S3 存储桶的名称，后跟可选的以“/”分隔的路径前缀。

必需：是

-a, --access-key access_key_id

Amazon 访问密钥 ID。

必需：是

-s, --secret-key secret_access_key

Amazon 秘密访问密钥。

必需：是

-k, --privatekey 路径

用于解密清单的私有密钥。

必需：是

`--url` url

Amazon S3 服务 URL。

默认值：`https://s3.amazonaws.com/`

必需：否

`--region` region

要在请求签名中使用的区域。

默认值：`us-east-1`

必需：若使用签名版本 4 则必需

`--sigv` version

对请求进行签名时要使用的签名版本。

有效值：2 | 4

默认值：4

必需：否

`-m, --manifest` file

清单文件的名称 (无路径)。我们建议您指定清单 (`-m`) 或前缀 (`-p`)。

必需：否

`-p, --prefix` prefix

捆绑的 AMI 文件的文件名前缀。

默认值：`image`

必需：否

`-d, --directory` directory

保存下载的捆绑的目录。该目录必须存在。

默认：当前工作目录。

必需：否

`--retry`

在所有 Amazon S3 错误后自动重试，每个操作最多五次。

必需：否

输出

将显示指示下载过程各个阶段的状态消息。

示例

此示例创建 `bundled` 目录（使用 Linux mkdir 命令）并从 `amzn-s3-demo-bucket` Amazon S3 存储桶下载捆绑。

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/bundles/bundle_name
 -m image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from amzn-s3-demo-bucket to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.00 ...
Downloaded image.part.00 from amzn-s3-demo-bucket
Downloading part image.part.01 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.01 ...
Downloaded image.part.01 from amzn-s3-demo-bucket
Downloading part image.part.02 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.02 ...
Downloaded image.part.02 from amzn-s3-demo-bucket
Downloading part image.part.03 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.03 ...
Downloaded image.part.03 from amzn-s3-demo-bucket
Downloading part image.part.04 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.04 ...
Downloaded image.part.04 from amzn-s3-demo-bucket
Downloading part image.part.05 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.05 ...
Downloaded image.part.05 from amzn-s3-demo-bucket
Downloading part image.part.06 from amzn-s3-demo-bucket/bundles/bundle_name to
 mybundle/image.part.06 ...
Downloaded image.part.06 from amzn-s3-demo-bucket
```

ec2-migrate-manifest

描述

修改 Amazon S3 支持的 Linux AMI（例如，其证书、内核和 RAM 磁盘）以使其支持其他区域。

语法

```
ec2-migrate-manifest -c path -k path -m path {(-a access_key_id -
s secret_access_key --region region) | (--no-mapping)} [--ec2cert
ec2_cert_path] [--kernel kernel-id] [--ramdisk ramdisk_id]
```

选项

`-c, --cert` 路径

用户的 PEM 编码 RSA 公有密钥凭证文件。

必需：是

`-k, --privatekey` 路径

用户的 PEM 编码 RSA 密钥文件的路径。

必需：是

`--manifest` 路径

清单文件的路径。

必需：是

`-a, --access-key` access_key_id

Amazon 访问密钥 ID。

必需：若使用自动映射则必需。

`-s, --secret-key` secret_access_key

Amazon 秘密访问密钥。

必需：若使用自动映射则必需。

`--region` region

要在映射文件中查找的区域。

必需：若使用自动映射则必需。

`--no-mapping`

禁用内核和 RAM 磁盘的自动映射。

迁移期间，Amazon EC2 会将清单文件中的内核和 RAM 磁盘替换为目标区域指定的内核和 RAM 磁盘。除非提供了 `--no-mapping` 参数，否则 `ec2-migrate-bundle` 便可能使用 `DescribeRegions` 和 `DescribeImages` 操作执行自动映射。

必需：若您不提供用于自动映射的 `-a`、`-s` 和 `--region` 选项，则必需。

`--ec2cert` 路径

用于加密映像清单的 Amazon EC2 X.509 公有密钥凭证的路径。

`us-gov-west-1` 和 `cn-north-1` 区域使用非默认公有密钥凭证，必须随该选项指定该证书的路径。该证书的路径因 AMI 工具的安装方法而异。对于 Amazon Linux，证书位于 `/opt/aws/amitools/ec2/etc/ec2/amitools/`。如果您将来自 ZIP 文件的 AMI 工具安装在 设置 Amazon EC2 AMI 工具 中，则证书位于 `$EC2_AMITOOL_HOME/etc/ec2/amitools/`。

必需：仅限 `us-gov-west-1` 和 `cn-north-1` 区域。

`--kernel` kernel_id

要选择的内核的 ID。

> **Important**
>
> 我们建议您使用 PV-GRUB 而不是内核和 RAM 磁盘。有关更多信息，请参阅《Amazon Linux 2 User Guide》中的 User provided kernels。

必需：否

`--ramdisk` ramdisk_id

供选择的 RAM 磁盘的 ID。

> **Important**
>
> 我们建议您使用 PV-GRUB 而不是内核和 RAM 磁盘。有关更多信息，请参阅《Amazon Linux 2 User Guide》中的 User provided kernels。

必需：否

输出

描述捆绑过程的阶段和状态的状态消息。

示例

此示例将 my-ami.manifest.xml 清单中指定的 AMI 从美国复制到欧洲。

```
[ec2-user ~]$ ec2-migrate-manifest --manifest my-ami.manifest.xml
 --cert cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --region eu-west-1

Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-west-1.
```

ec2-unbundle

描述

从 Amazon S3 支持的 Linux AMI 重新创建捆绑。

语法

ec2-unbundle -k *path* -m *path* [-s *source_directory*] [-d *destination_directory*]

选项

-k, --privatekey 路径

您的 PEM 编码 RSA 密钥文件的路径。

必需：是

-m, --manifest 路径

清单文件的路径。

必需：是

-s, --source source_directory

包含捆绑的目录。

默认：当前目录。

必需：否

`-d, --destination` destination_directory

将 AMI 解绑到的目录。目标目录必须存在。

默认：当前目录。

必需：否

示例

此 Linux 和 UNIX 示例解绑 `image.manifest.xml` 文件中指定的 AMI。

```
[ec2-user ~]$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -s mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

输出

将显示指示解绑过程各个阶段的状态消息。

ec2-upload-bundle

描述

将 Amazon S3 支持的 Linux AMI 的捆绑上载到 Amazon S3，并在上载的对象上设置相应的访问控制列表（ACL）。有关更多信息，请参阅 创建 Amazon S3 支持的 AMI。

> **Note**
>
> 要将对象上载到 Amazon S3 支持的 Linux AMI 的 S3 存储桶，必须为该存储桶启用 ACL。否则，Amazon EC2 将无法在要上载的对象上设置 ACL。如果您的目标存储桶使用存储桶拥有者强制执行的 S3 对象所有权设置，则这将不起作用，因为 ACL 已禁用。有关更多信息，请参阅为您的存储桶控制对象所有权和禁用 ACL。

语法

ec2-upload-bundle -b *bucket* -a *access_key_id* -s *secret_access_key* [-t *token*] -m *path* [--url *url*] [--region *region*] [--sigv *version*] [--acl *acl*] [-d *directory*] [--part *part*] [--retry] [--skipmanifest]

选项

`-b, --bucket` 存储桶

用于存储捆绑的 Amazon S3 存储桶的名称，后跟可选的以“/”分隔的路径前缀。如果存储桶不存在，则创建一个 (若存储桶名称可用)。此外，如果存储桶不存在且 AMI 工具版本为 1.5.18 或更高版本，则此命令会为存储桶设置 ACL。

必需：是

`-a, --access-key` access_key_id

您的 Amazon 访问密钥 ID。

必需：是

`-s, --secret-key` secret_access_key

您的 Amazon 秘密访问密钥。

必需：是

`-t, --delegation-token` token

传递到 Amazon 请求的委托令牌。有关更多信息，请参阅《IAM 用户指南》中的临时安全凭证。

必需：仅当使用临时安全凭证时是必需的。

默认：`AWS_DELEGATION_TOKEN` 环境变量的值 (若已设置)。

`-m, --manifest` 路径

清单文件的路径。清单文件是在捆绑过程中创建的，可以在包含捆绑的目录中找到。

必需：是

`--url` url

已淘汰。请使用 `--region` 选项，除非您的存储桶被限制到 EU 位置 (且不是 `eu-west-1`)。`--location` 标记是确定该特定位置限制的唯一途径。

Amazon S3 终端节点服务 URL。

默认值：https://s3.amazonaws.com/

必需：否

--region region

要在请求签名中为目标 S3 存储桶使用的区域。

- 如果存储桶不存在，您也没有指定区域，则该工具将创建无位置限制的存储桶（在 us-east-1 中）。
- 如果存储桶不存在，而您指定了区域，则该工具将在指定区域创建存储桶。
- 如果存储桶存在，而您没有指定区域，则该工具将使用存储桶的位置。
- 如果存储桶存在，并且您指定 us-east-1 为区域，则该工具将使用存储桶的实际位置而不会显示任何错误消息，并将覆盖任何现有的匹配文件。
- 如果存储桶存在，并且您指定与存储桶的实际位置不符的区域（非 us-east-1），则该工具将报错退出。

如果您的存储桶被限制到 EU 位置 (不是 eu-west-1)，请改用 --location 标记。--location 标记是确定该特定位置限制的唯一途径。

默认值：us-east-1

必需：若使用签名版本 4 则必需

--sigv version

对请求进行签名时要使用的签名版本。

有效值：2 | 4

默认值：4

必需：否

--acl acl

捆绑的映像的访问控制列表策略。

有效值：public-read | aws-exec-read

默认值：aws-exec-read

必需：否

-d, --directory directory

包含捆绑的 AMI 段的目录。

默认：包含清单文件的目录 (参阅 -m 选项)。

必需：否

--part part

开始上传指定的段及所有后续段。例如：--part 04。

必需：否

--retry

在所有 Amazon S3 错误后自动重试，每个操作最多五次。

必需：否

--skipmanifest

不上传清单。

必需：否

--location 位置

已淘汰。请使用 --region 选项，除非您的存储桶被限制到 EU 位置 (且不是 eu-west-1)。--location 标记是确定该特定位置限制的唯一途径。

目标 Amazon S3 存储桶的位置限制。如果存储桶存在，而您指定的位置与存储桶的实际位置不符，则该工具将报错退出。如果存储桶存在，而您没有指定位置，则该工具将使用存储桶的位置。如果存储桶不存在，而您指定了位置，则该工具将在指定位置创建存储桶。如果存储桶不存在，您也没有指定位置，则该工具将创建无位置限制的存储桶 (在 us-east-1 中)。

默认：如果指定 --region，则将位置设置为该指定区域。如果未指定 --region，则位置默认为 us-east-1。

必需：否

输出

Amazon EC2 显示状态消息以指示上传过程的阶段的状态。

示例

此示例上传 image.manifest.xml 清单所指定的捆绑。

```
[ec2-user ~]$ ec2-upload-bundle -b amzn-s3-demo-bucket/bundles/bundle_name -m
 image.manifest.xml -a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket amzn-s3-demo-bucket ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

AMI 工具的常用选项

大多数 AMI 工具接受以下可选参数。

`--help, -h`

显示帮助消息。

`--version`

显示版本和版权声明。

`--manual`

显示手动输入。

`--batch`

以批处理模式运行，不显示交互提示。

`--debug`

显示对故障排除可能有帮助的信息。

将 Amazon S3 支持的 AMI 转换为 EBS-backed AMI

您可以将拥有的 Amazon S3 支持的 Linux AMI 转换为 Amazon EBS 支持的 Linux AMI。

> **Important**
>
> 您无法转换不属于自己的 AMI。

将 Amazon S3 支持的 AMI 转换为 Amazon EBS-backed AMI

1. 从 Amazon EBS-backed AMI 启动 Amazon Linux 实例。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。Amazon Linux 实例预先安装了 Amazon CLI 和 AMI 工具。
2. 上传您用于将 Amazon S3 支持的 AMI 捆绑到实例的 X.509 私有密钥。我们使用此密钥确保只有您和 Amazon EC2 才能访问您的 AMI。

   a. 在您的实例上为 X.509 私有密钥创建临时目录，如下所示：

   ```
   [ec2-user ~]$ mkdir /tmp/cert
   ```

   b. 使用安全复制工具 (如 `/tmp/certscp`) 将您的 X.509 私有密钥从您的计算机复制到您实例上的 目录。以下命令中的 *my-private-key* 参数是您用于通过 SSH 连接到实例的私有密钥。例如：

   ```
   you@your_computer:~ $ scp -i my-private-key.pem /
   path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
   user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
   pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  100%  717     0.7KB/s   00:00
   ```

3. 配置环境变量以使用 Amazon CLI。有关更多信息，请参阅 Environment variables。

   a. （推荐）为您的 Amazon 访问密钥、私有密钥和会话令牌设置环境变量。

   ```
   [ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
   [ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
   [ec2-user ~]$ export AWS_SESSION_TOKEN=your_session_token
   ```

b. 为您的 Amazon 访问密钥和私有密钥设置环境变量。

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

4. 为新 AMI 准备 Amazon Elastic Block Store (Amazon EBS) 卷。

a. 使用 create-volume 命令在您的实例所在的同一可用区中创建空 EBS 卷。记下命令输出中的卷 ID。

> **Important**
>
> 此 EBS 卷不小于原始实例存储根卷。

```
aws ec2 create-volume \
    --size 10 \
    --region us-west-2 \
    --availability-zone us-west-2b
```

b. 使用 attach-volume 命令将该卷附加到 Amazon EBS 支持的实例。

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --region us-west-2
```

5. 创建用于捆绑的文件夹。

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. 使用 `/tmp/bundle` 命令将由实例存储支持的 AMI 的捆绑下载到 ec2-download-bundle。

```
[ec2-user ~]$ ec2-download-bundle -b amzn-s3-demo-bucket/bundle_folder/bundle_name
 -m image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --
privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. 使用 ec2-unbundle 命令从捆绑重新构建映像文件。

a. 将目录更改为捆绑文件夹。

```
[ec2-user ~]$ cd /tmp/bundle/
```

b. 运行 ec2-unbundle 命令。

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. 将文件从未捆绑的映像复制到新 EBS 卷。

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

9. 探测所有未捆绑的新分区的卷。

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

10. 列出块储存设备以查找要装载的设备名称。

```
[ec2-user bundle]$ lsblk
NAME         MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/sda    202:0    0   8G  0 disk
##/dev/sda1 202:1    0   8G  0 part /
/dev/sdb    202:80   0  10G  0 disk
##/dev/sdb1 202:81   0  10G  0 part
```

在此示例中，要装载的分区是 /dev/sdb1，但您的设备名称可能有所不同。如果您的卷未分区，则要装载的设备类似于 /dev/sdb (没有设备分区尾部数字)。

11. 为新 EBS 卷创建装载点并装载该卷。

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. 使用您常用的文本编辑器（如 /etc/fstab 或 vim）在 EBS 卷上打开 nano 文件，然后删除实例存储（临时）卷的所有条目。因为 EBS 卷装载在 /mnt/ebs 上，所以 fstab 文件位于 /mnt/ebs/etc/fstab 处。

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/     /           ext4    defaults,noatime  1   1
tmpfs       /dev/shm    tmpfs   defaults        0   0
devpts      /dev/pts    devpts  gid=5,mode=620  0   0
```

```
sysfs       /sys        sysfs   defaults        0   0
proc        /proc       proc    defaults        0   0
/dev/sdb        /media/ephemeral0       auto    defaults,comment=cloudconfig    0
     2
```

在本示例中，应删除最后一行。

13. 从实例中卸载和分离该卷。

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id vol-01234567890abcdef --
region us-west-2
```

14. 按如下所示从新 EBS 卷创建 AMI。

    a. 创建新 EBS 卷的快照。

```
[ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description
 "your_snapshot_description" --volume-id vol-01234567890abcdef
```

    b. 检查快照是否完整。

```
[ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
id snap-0abcdef1234567890
```

    c. 使用 `aki` 命令标识在原始 AMI 上使用的处理器架构、虚拟化类型和内核映像 (describe-images)。对于此步骤，您需要 Amazon S3 支持的原始 AMI 的 AMI ID。

```
[ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-
id ami-0abcdef1234567890 --output text
IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon
 available public machine aki-fc8f11cc instance-store paravirtual xen
```

    在此示例中，架构是 `x86_64`，内核映像 ID 是 `aki-fc8f11cc`。在以下步骤中使用这些值。如果上面命令的输出还列出 `ari` ID，请记下该 ID。

    d. 使用新 EBS 卷的快照 ID 和上一步中得到的值注册新 AMI。如果前一命令输出列出了 `ari` ID，请通过 `--ramdisk-id` *ari_id* 将其包括在后续命令中。

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --
name your_new_ami_name --block-device-mappings DeviceName=device-
```

```
name,Ebs={SnapshotId=snap-0abcdef1234567890} --virtualization-type paravirtual
 --architecture x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
```

15. (可选) 测试了您可以从新 AMI 启动实例之后，您可以删除为此过程创建的 EBS 卷。

```
aws ec2 delete-volume --volume-id vol-01234567890abcdef
```

使用 Windows Sysprep 创建 Amazon EC2 AMI

Microsoft 系统准备（简称 Windows Sysprep）工具会创建操作系统的通用版本，可在捕获新映像之前删除实例特定的系统配置。

我们建议您使用 EC2 Image Builder 自动创建、管理和部署自定义、安全且最新的“黄金”服务器映像，这些映像是预先安装的，并预配置了软件和设置。

您还可以使用 Windows Sysprep 通过以下 Windows 启动代理创建标准化 AMI：EC2Launch v2、EC2Launch 和 EC2Config。

> **Important**
>
> 不要使用 Windows Sysprep 创建实例备份。Windows Sysprep 会删除系统特定的信息；删除这些信息可能会对实例备份造成意外结果。

要排查 Windows Sysprep 问题，请参阅排查 Amazon EC2 Windows 实例的 Sysprep 问题。

内容

- Windows Sysprep 阶段
- 开始前的准备工作
- 使用 Windows Sysprep 和 EC2Launch v2 创建 AMI
- 使用 Windows Sysprep 和 EC2Launch 创建 AMI
- 使用 Windows Sysprep 和 EC2Config 创建 AMI

Windows Sysprep 阶段

Windows Sysprep 会经历以下阶段：

- 一般化：Sysprep 工具删除映像特定的信息和配置。例如，Windows Sysprep 会删除安全标识符（SID）、计算机名称、事件日志和特定驱动程序，等等。完成这一阶段后，操作系统 (OS) 即准备就绪，可以创建 AMI。

  > **Note**
  >
  > 使用 Windows 启动代理运行 Windows Sysprep 时，系统会阻止删除驱动程序，因为 `PersistAllDeviceInstalls` 在默认情况下设置为 true。

- 专门化：即插即用功能会扫描计算机并为检测到的所有设备安装驱动程序。Sysprep 工具会生成操作系统要求，如计算机名称和 SID。您可以选择在这一阶段运行命令。
- 全新体验 (OOBE)：系统运行 Windows 安装程序的一个简化版本并要求您输入系统语言、时区和注册组织等信息。使用 Windows 启动运行 Windows Sysprep 时，应答文件会自动执行这一阶段。

开始前的准备工作

- 执行 Windows Sysprep 前，建议删除所有本地用户账户及所有账户配置文件，而不是将在其下方运行 Windows Sysprep 的单个管理员账户。如果使用其他账户和配置文件执行 Windows Sysprep，可能会产生意外行为，包括配置文件数据丢失或完成 Sysprep 失败。
- 了解有关 Sysprep 概览的更多信息。
- 了解 Sysprep Support for Server Roles。

使用 Windows Sysprep 和 EC2Launch v2 创建 AMI

当您从安装了 EC2Launch v2 代理的实例创建映像时，EC2Launch v2 会在准备映像时执行特定的任务。这包括使用 Windows Sysprep。有关更多信息，请参阅 the section called “Windows Sysprep 阶段”。

内容

- Windows Sysprep 操作
- Sysprep 之后
- 使用 EC2Launch v2 运行 Windows Sysprep

Windows Sysprep 操作

在准备映像的过程中，Windows Sysprep 和 EC2Launch v2 会执行以下操作。

1. 当您在 EC2Launch 设置对话框中选择使用 Sysprep 关闭时，系统会运行 `ec2launch sysprep` 命令。
2. EC2Launch v2 通过读取 `unattend.xml` 的注册表值来编辑 `HKEY_USERS\.DEFAULT\Control Panel\International\LocaleName` 文件的内容。该文件位于以下目录中：`C:\ProgramData\Amazon\EC2Launch\sysprep`。
3. 系统运行 `BeforeSysprep.cmd`。该命令创建一个注册表项，如下所示：

   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 1 /f

   该注册表项禁用 RDP 连接，直到它们重新启用。禁用 RDP 连接是必需的安全措施，因为在 Windows Sysprep 运行之后的第一个启动会话过程中，在短时间内，RDP 允许连接并且管理员密码为空。
4. EC2Launch v2 服务通过运行以下命令来调用 Windows Sysprep：

   sysprep.exe /oobe /generalize /shutdown /unattend: "C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml"

一般化阶段

- EC2Launch v2 删除特定于映像的信息和配置，如计算机名称和 SID。如果实例是域的成员，则会从域中将其删除。`unattend.xml` 应答文件包含影响这一阶段的以下设置：
  - PersistAllDeviceInstalls：此设置防止 Windows 安装程序删除和重新配置设备，这可加快映像准备过程，因为 Amazon AMI 需要运行特定驱动程序，而这些驱动程序的重新检测需要花费一些时间。
  - DoNotCleanUpNonPresentDevices：此设置为当前不存在的设备保留即插即用信息。
- Windows Sysprep 在准备创建 AMI 时会关闭操作系统。系统会启动新实例或启动原始实例。

专门化阶段

系统生成特定于操作系统的要求，如计算机名称和 SID。系统还会根据您在 `unattend.xml` 应答文件中指定的配置执行以下操作。

- CopyProfile：Windows Sysprep 可以配置为删除所有用户配置文件，包括内置管理员配置文件。此设置保留内置管理员账户，这样，您对该账户进行的所有自定义设置就都可以转移到新映像。默认值为 `True`。

CopyProfile 将默认配置文件替换为现有本地管理员配置文件。运行 Windows Sysprep 后登录的所有账户在首次登录时都会收到该配置文件的副本及其内容。

如果您没有要转移到新映像的特定用户配置文件自定义设置，请将此设置更改为 `False`。Windows Sysprep 会删除所有用户配置文件；此举可以节省时间和磁盘空间。

- TimeZone：默认情况下，时区设置为协调世界时 (UTC)。
- 顺序 1 同步命令：系统运行以下命令，以启用管理员账户并指定密码要求。

```
net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /
PASSWORDREQ:YES
```

- 顺序 2 同步命令：系统加密管理员密码。如果未配置 `setAdminAccount` 任务，则此安全措施可防止实例在 Windows Sysprep 完成后处于可访问状态。

  系统会从本地启动代理目录 (`C:\Program Files\Amazon\EC2Launch\`) 运行以下命令。

```
EC2Launch.exe internal randomize-password --username Administrator
```

- 要启用远程桌面连接，系统会将终端服务器 `fDenyTSConnections` 注册表项设置为 false。

OOBE 阶段

1. 系统使用 EC2Launch v2 应答文件指定以下配置：
   - `<InputLocale>en-US</InputLocale>`
   - `<SystemLocale>en-US</SystemLocale>`
   - `<UILanguage>en-US</UILanguage>`
   - `<UserLocale>en-US</UserLocale>`
   - `<HideEULAPage>true</HideEULAPage>`
   - `<HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>`
   - `<ProtectYourPC>3</ProtectYourPC>`
   - `<BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>`
   - `<TimeZone>UTC</TimeZone>`
   - `<RegisteredOrganization>Amazon.com</RegisteredOrganization>`
   - `<RegisteredOwner>EC2</RegisteredOwner>`

> **Note**
>
> 在一般化和专门化阶段，EC2Launch v2 会监控操作系统的状态。如果 EC2Launch v2 检测到操作系统处于 Sysprep 阶段，则会将以下消息发布到系统日志：
> 正在配置 Windows。SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. 系统运行 EC2Launch v2。

Sysprep 之后

Windows Sysprep 完成后，EC2Launch v2 会将以下消息发送到控制台输出：

```
Windows sysprep configuration complete.
```

EC2Launch v2 随后执行以下操作：

1. 读取 `agent-config.yml` 文件的内容并运行配置的任务。
2. 执行 `preReady` 阶段的所有任务。
3. 完成后，将 `Windows is ready` 消息发送到实例系统日志。
4. 执行 `PostReady` 阶段的所有任务。

有关 EC2Launch v2 的更多信息，请参阅使用 EC2Launch v2 代理在 EC2 Windows 实例启动期间执行任务。

使用 EC2Launch v2 运行 Windows Sysprep

按照以下过程，结合使用 Windows Sysprep 与 EC2Launch v2 来创建标准化 AMI。

1. 在 Amazon EC2 控制台中，查找要复制的 AMI。
2. 启动并连接到您的 Windows 实例。
3. 自定义设置
   a. 在 Windows 开始菜单中，搜索并选择 Amazon EC2Launch 设置。有关 Amazon EC2Launch 设置对话框中的选项和设置的更多信息，请参阅 配置 Windows 实例的 EC2Launch v2 设置。
   b. 如果您进行了更改，请选择保存后再关闭。
4. 选择使用 Sysprep 关闭或不使用 Sysprep 关闭。

如果系统询问您是否想运行 Windows Sysprep 并关闭该实例，请单击是。EC2Launch v2 会运行 Windows Sysprep。然后，您退出实例，实例关闭。如果您在 Amazon EC2 控制台中查看实例页面，实例状态会从 `Running` 变为 `Stopping`，再变为 `Stopped`。此时，从该实例创建一个 AMI 是安全的。

您可以使用以下命令从命令行中手动调用 Windows Sysprep 工具：

```
"%programfiles%\amazon\ec2launch\ec2launch.exe" sysprep --shutdown=true
```

使用 Windows Sysprep 和 EC2Launch 创建 AMI

当您从安装了 EC2Launch 代理的实例创建映像时，EC2Launch 会在准备映像时执行特定的任务。这包括使用 Windows Sysprep。有关更多信息，请参阅 the section called “Windows Sysprep 阶段”。

EC2Launch 为 Windows Sysprep 提供一个默认应答文件和批处理文件，来自动执行并保护 AMI 上的映像准备过程。您可以选择性地修改这些文件。这些文件默认位于以下目录：`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep`。

> **Important**
>
> 不要使用 Windows Sysprep 创建实例备份。Windows Sysprep 会删除系统特定的信息。如果您删除此信息，可能会给实例备份带来意想不到的后果。

内容

- 用于 Windows Sysprep 的 EC2Launch 应答文件和批处理文件
- 使用 EC2Launch 运行 Windows Sysprep
- 在启动自定义 AMI 时更新 Server 2016 及更高版本的元数据/KMS 路由

用于 Windows Sysprep 的 EC2Launch 应答文件和批处理文件

用于 Windows Sysprep 的 EC2Launch 应答文件和批处理文件包括：

`Unattend.xml`

这是默认应答文件。如果您运行 `SysprepInstance.ps1` 或者在用户界面中选择 ShutdownWithSysprep，系统会从该文件中读取设置。

`BeforeSysprep.cmd`

自定义此批处理文件，以便在 EC2Launch 运行 Windows Sysprep 前执行命令。

`SysprepSpecialize.cmd`

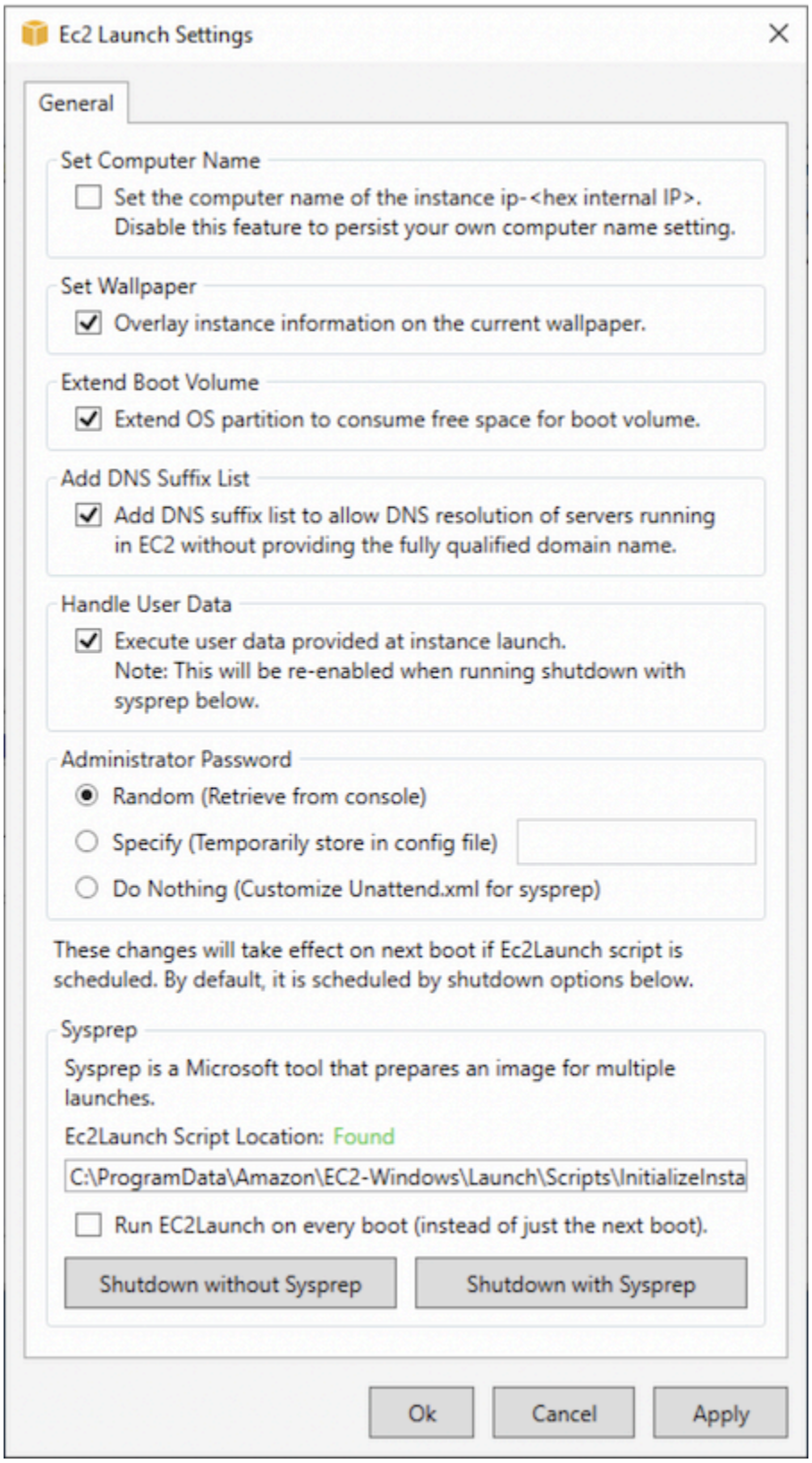
自定义此批处理文件，以便在 Windows Sysprep 专门化阶段期间运行命令。

使用 EC2Launch 运行 Windows Sysprep

在 Windows Server 2016 及更高版本的完整安装（带有桌面体验）上，可以通过手动方式使用 EC2Launch 运行 Windows Sysprep，也可以通过 EC2Launch Settings 应用程序运行 Windows Sysprep。

使用 EC2Launch Settings 应用程序运行 Windows Sysprep

1. 在 Amazon EC2 控制台中，找到或创建 Windows Server 2016 或更高版本 AMI。
2. 从该 AMI 中启动 Windows 实例。
3. 连接到您的 Windows 实例并对其进行自定义。
4. 搜索并运行 EC2LaunchSettings 应用程序。默认情况下它位于以下内容：`C:\ProgramData\Amazon\EC2-Windows\Launch\Settings`。



5. 根据需要选中或清除选项。这些设置存储在 `LaunchConfig.json` 文件中。
6. 对于 Administrator Password，请执行以下操作之一：

- 选择 Random。EC2Launch 生成一个密码并使用用户的密钥对其进行加密。系统会在实例启动后禁用此设置，以便在重新启动或停止再启动实例后该密码仍然存在。
- 选择 Specify 并键入一个符合系统要求的密码。该密码以明文方式存储在 `LaunchConfig.json` 文件中，并且在 Windows Sysprep 设置管理员密码时会被删除。如果现在就关闭，则会立即设置密码。EC2Launch 使用用户的密钥对密码进行加密。
- 选择 DoNothing 并在 `unattend.xml` 文件中指定密码。如果未在 `unattend.xml` 中指定密码，管理员账户会被禁用。

7. 选择 Shutdown with Sysprep (使用 Sysprep 关闭)。

使用 EC2Launch 手动运行 Windows Sysprep

1. 在 Amazon EC2 控制台中找到或创建您要复制的 Windows Server 2016 或更高版本数据中心版 AMI。
2. 启动并连接到您的 Windows 实例。
3. 自定义实例。
4. 在 `LaunchConfig.json` 文件中指定设置。默认情况下，该文件位于 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目录中。

   对于 `adminPasswordType`，请指定下列值之一：

   `Random`

   EC2Launch 生成一个密码并使用用户的密钥对其进行加密。系统会在实例启动后禁用此设置，以便在重新启动或停止再启动实例后该密码仍然存在。

   `Specify`

   EC2Launch 使用您在 `adminPassword` 中指定的密码。如果密码不满足系统要求，EC2Launch 会生成随机密码。该密码以明文方式存储在 `LaunchConfig.json` 文件中，并且在 Windows Sysprep 设置管理员密码时会被删除。EC2Launch 使用用户的密钥对密码进行加密。

   `DoNothing`

   EC2Launch 使用您在 `unattend.xml` 文件中指定的密码。如果未在 `unattend.xml` 中指定密码，管理员账户会被禁用。

5. (可选) 在 unattend.xml 和其他配置文件中指定设置。如果您计划参与安装，则无需在这些文件中进行更改。这些文件默认位于以下目录：C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep。
6. 在 Windows PowerShell 中，运行 ./InitializeInstance.ps1 -Schedule。默认情况下，该脚本位于以下目录中：C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts。该脚本安排实例在下次启动期间初始化。您必须先运行此脚本，然后再在下一个步骤中运行 SysprepInstance.ps1 脚本。
7. 在 Windows PowerShell 中，运行 ./SysprepInstance.ps1。默认情况下，该脚本位于以下目录中：C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts。

您会退出实例并且实例会关闭。如果您在 Amazon EC2 控制台中查看实例页面，实例状态会从 Running 变为 Stopping，然后变为 Stopped。此时，从该实例创建 AMI 是安全的。

在启动自定义 AMI 时更新 Server 2016 及更高版本的元数据/KMS 路由

要在启动自定义 AMI 时更新 Server 2016 及更高版本的元数据/KMS 路由，请执行下列操作之一：

- 运行 EC2LaunchSettings GUI (C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe)，并选择相应选项以便在使用 Windows Sysprep 的情况下关闭。
- 在创建 AMI 之前，运行 EC2LaunchSettings，并在未使用 Windows Sysprep 的情况下关闭。这会将 EC2 启动初始化任务设置为在下次引导时运行，从而根据实例的子网设置路由。
- 在从 PowerShell 创建 AMI 之前，手动重新计划 EC2 启动初始化任务。

  > **Important**
  >
  > 在重新计划任务之间记录默认的密码重置行为。

- 要在发生 Windows 激活失败或与实例元数据通信失败的正在运行的实例上更新路由，请参阅“无法激活 Windows”。

使用 Windows Sysprep 和 EC2Config 创建 AMI

当您从安装了 EC2Config 服务的实例创建映像时， EC2Config 会在准备映像时执行特定的任务。这包括使用 Windows Sysprep。有关更多信息，请参阅 the section called “Windows Sysprep 阶段”。

内容

- Windows Sysprep 操作

- Sysprep 之后
- 使用 EC2Config 服务运行 Windows Sysprep

Windows Sysprep 操作

在准备映像的过程中，Windows Sysprep 和 EC2Config 服务会执行以下操作。

1. 在 EC2 服务属性对话框中选择使用 Sysprep 关闭时，系统会运行 ec2config.exe -sysprep 命令。
2. EC2Config 服务读取 BundleConfig.xml 文件的内容。默认情况下，该文件位于以下目录中：C:\Program Files\Amazon\Ec2ConfigService\Settings。

   BundleConfig.xml 文件包含以下设置：您可以更改这些设置：

   - AutoSysprep：指明是否自动使用 Windows Sysprep。如果从 EC2 Service Properties 对话框运行 Windows Sysprep，则无需更改此值。默认值为 No。
   - SetRDPCertificate：为远程桌面服务器设置自签名证书。这样，您可以安全地使用远程桌面协议 (RDP) 连接到实例。如果新实例应使用证书，请将该值更改为 Yes。此设置不适用于 Windows Server 2012 实例，因为这些操作系统会生成自己的证书。默认值为 No。
   - SetPasswordAfterSysprep：在新启动的实例上设置随机密码，使用用户启动密钥对其加密，并将加密密码输出到控制台。如果新实例不应设置为随机加密密码，请将该值更改为 No。默认值为 Yes。
   - PreSysprepRunCmd：要运行的命令的位置。默认情况下，该命令位于以下目录中：C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd。
3. 系统运行 BeforeSysprep.cmd。该命令创建一个注册表项，如下所示：

   ```
   reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
    fDenyTSConnections /t REG_DWORD /d 1 /f
   ```

   该注册表项禁用 RDP 连接，直到它们重新启用。禁用 RDP 连接是必需的安全措施，因为在 Windows Sysprep 运行之后的第一个启动会话过程中，在短时间内，RDP 允许连接并且管理员密码为空。
4. EC2Config 服务通过运行以下命令来调用 Windows Sysprep：

   ```
   sysprep.exe /unattend: "C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml" /
   oobe /generalize /shutdown
   ```

一般化阶段

- 该工具删除特定于映像的信息和配置 (如计算机名称和 SID)。如果实例是域的成员，则会从域中将其删除。`sysprep2008.xml` 应答文件包含影响这一阶段的以下设置：
  - PersistAllDeviceInstalls：此设置防止 Windows 安装程序删除和重新配置设备，这可加快映像准备过程，因为 Amazon AMI 需要运行特定驱动程序，而这些驱动程序的重新检测需要花费一些时间。
  - DoNotCleanUpNonPresentDevices：此设置为当前不存在的设备保留即插即用信息。
- Windows Sysprep 在准备创建 AMI 时会关闭操作系统。系统会启动新实例或启动原始实例。

专门化阶段

系统生成特定于操作系统的要求，如计算机名称和 SID。系统还根据您在 sysprep2008.xml 应答文件中指定的配置来执行以下操作。

- CopyProfile：Windows Sysprep 可以配置为删除所有用户配置文件，包括内置管理员配置文件。此设置保留内置管理员账户，这样，您对该账户进行的所有自定义都可转移到新映像。默认值是 True。

  CopyProfile 将默认配置文件替换为现有本地管理员配置文件。运行 Windows Sysprep 后登录的所有账户在首次登录时都会收到该配置文件的副本及其内容。

  如果您没有要转移到新映像的特定用户配置文件自定义设置，请将此设置更改为 False。Windows Sysprep 会删除所有用户配置文件；此举可以节省时间和磁盘空间。
- TimeZone：默认情况下，时区设置为协调世界时 (UTC)。
- 顺序 1 同步命令：系统运行以下命令，以启用管理员账户并指定密码要求。

  net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER / PASSWORDREQ:YES
- 顺序 2 同步命令：系统加密管理员密码。如果未启用 ec2setpassword 设置，则此安全措施可防止实例在 Windows Sysprep 完成后处于可访问状态。

  C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator
- 顺序 3 同步命令：系统运行以下命令：

  C:\Program Files\Amazon\Ec2ConfigService\Scripts\SysprepSpecializePhase.cmd

  该命令添加以下注册表项，用于重新启用 RDP：

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f

OOBE 阶段

1. 通过使用 EC2Config 服务应答文件，系统指定以下配置：
   - <InputLocale>en-US</InputLocale>
   - <SystemLocale>en-US</SystemLocale>
   - <UILanguage>en-US</UILanguage>
   - <UserLocale>en-US</UserLocale>
   - <HideEULAPage>true</HideEULAPage>
   - <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
   - <NetworkLocation>Other</NetworkLocation>
   - <ProtectYourPC>3</ProtectYourPC>
   - <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>
   - <TimeZone>UTC</TimeZone>
   - <RegisteredOrganization>Amazon.com</RegisteredOrganization>
   - <RegisteredOwner>Amazon</RegisteredOwner>

   > **Note**
   >
   > 在一般化和专门化阶段期间，EC2Config 服务监控操作系统的状态。如果 EC2Config 检测到操作系统处于 Sysprep 阶段，则会将以下消息发布到系统日志：EC2ConfigMonitorState: 0 Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. OOBE 阶段完成后，系统将 从以下位置运行 `SetupComplete.cmd`：`C:\Windows\Setup\Scripts\SetupComplete.cmd`。在 2015 年 4 月之前的 Amazon 公用 AMI 中，此文件为空，不对映像运行任何操作。在 2015 年 4 月之后的公开 AMI 中，此文件包含以下值：call "C:\Program Files\Amazon\Ec2ConfigService\Scripts\PostSysprep.cmd"。
3. 系统运行 `PostSysprep.cmd`，执行以下操作：
   - 将本地管理员密码设置为不过期。如果密码已过期，管理员可能无法登录。
   - 设置 MSSQLServer 机器名称 (如果已安装)，以便该名称与 AMI 同步。

Sysprep 之后

Windows Sysprep 完成后，EC2Config 服务会将以下消息发送到控制台输出：

```
Windows sysprep configuration complete.
   Message: Sysprep Start
   Message: Sysprep End
```

EC2Config 随后执行以下操作：

1. 读取 config.xml 文件的内容并列出所有已启用的插件。
2. 同时执行所有“Windows 就绪之前”插件。
   - Ec2SetPassword
   - Ec2SetComputerName
   - Ec2InitializeDrives
   - Ec2EventLog
   - Ec2ConfigureRDP
   - Ec2OutputRDPCert
   - Ec2SetDriveLetter
   - Ec2WindowsActivate
   - Ec2DynamicBootVolumeSize
3. 完成之后，将“Windows 准备就绪”消息发送到实例系统日志。
4. 同时运行所有“Windows 就绪之后”插件。
   - Amazon CloudWatch Logs
   - UserData
   - Amazon Systems Manager (Systems Manager)

有关 Windows 插件的更多信息，请参阅使用 EC2Config 服务在 EC2 旧版 Windows 操作系统实例启动期间执行任务。

使用 EC2Config 服务运行 Windows Sysprep

按照以下过程，结合使用 Windows Sysprep 与 EC2Config 服务来创建标准化 AMI。

1. 在 Amazon EC2 控制台中，查找或创建要复制的 AMI。
2. 启动并连接到您的 Windows 实例。

3. 对它进行自定义。
4. 在 EC2Config 服务应答文件中指定配置设置：

   ```
   C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml
   ```

5. 在 Windows 开始菜单中，选择所有程序，然后选择 EC2ConfigService Settings。
6. 在 Ec2 Service Properties 对话框中选择 Image 选项卡。有关 Ec2 Service Properties 对话框中的选项和设置的更多信息，请参阅 Ec2 服务属性。
7. 选择管理员密码选项，然后选择 Shutdown with Sysprep 或 Shutdown without Sysprep。EC2Config 会根据您选择的密码选项编辑设置文件。
   - 随机：EC2Config 生成一个密码，使用用户的密钥加密，并向控制台显示加密的密码。我们会在首次启动后禁用此设置，以便在重启或停止再启动实例后该密码仍然存在。
   - 指定：密码以非加密格式（明文）存储在 Windows Sysprep 应答文件中。下一次运行 Windows Sysprep 时，即会设置管理员密码。如果现在就关闭，则会立即设置密码。服务再次启动时，则会删除管理员密码。务必要记住此密码，因为以后无法将其取回。
   - 保持现有：运行 Windows Sysprep 或重新启动 EC2Config 时，管理员账户的现有密码不变。务必要记住此密码，因为以后无法将其取回。
8. 选择 OK。

如果系统询问您是否想运行 Windows Sysprep 并关闭该实例，请单击是。您会看到 EC2Config 运行 Windows Sysprep。然后，您会退出实例并且实例会关闭。如果您在 Amazon EC2 控制台中查看 Instances (实例) 页面，实例状态会从 `Running` 变为 `Stopping`，然后最终变为 `Stopped`。此时，从该实例创建一个 AMI 是安全的。

您可以使用以下命令从命令行中手动调用 Windows Sysprep 工具：

```
"%programfiles%\amazon\ec2configservice\"ec2config.exe -sysprep""
```

> **Note**
>
> 如果您的 CMD Shell 已在 C:\Program Files\Amazon\EC2ConfigService\ 目录中，则命令中不需要双引号。

但是，请务必确保 `Ec2ConfigService\Settings` 文件夹中指定的 XML 文件选项正确无误，否则您可能无法连接到实例。有关设置文件的更多信息，请参阅 EC2Config 设置文件。有关

配置 Windows Sysprep 并从命令行中运行它的示例，请参阅 `Ec2ConfigService\Scripts\InstallUpdates.ps1`。

复制 Amazon EC2 AMI

如果需要跨多个区域统一配置 Amazon EC2 实例，您可以将单个亚马逊机器映像（AMI）作为模板来启动所有实例。但由于 AMI 属于区域特定的资源，要在特定 Amazon Web Services 区域中启动实例，该 AMI 必须位于该区域。因此，要在多个区域使用相同的 AMI，必须将其从源区域复制到每个目标区域。

用于复制 AMI 的方法取决于您是在同一分区内跨区域复制，还是跨不同分区复制：

- 跨区域复制 – 在同一分区内跨区域复制 AMI，例如，在商业分区内跨区域复制。本主题中介绍的就是这种复制方法。
- 跨分区复制 – 将 AMI 从一个分区复制到另一个分区，例如，从商业分区复制到 Amazon GovCloud (US) 分区。有关此复制方法的更多信息，请参阅存储和还原 AMI。
- 跨账户复制 – 创建由其他 Amazon Web Services 账户与您的 Amazon Web Services 账户共享的 AMI 的副本。本主题中介绍的就是这种复制方法。

对于跨区域和跨账户 AMI 复制，完成复制操作需要的时间为尽力完成的时间。如果需要控制完成时间，您可以指定一个介于 15 分钟到 48 小时之间的完成时限，从而确保在所需的时限内复制 AMI。基于时间的 AMI 复制操作会产生额外的费用。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Time-based copies。

内容

- 注意事项
- 成本
- 授予复制 Amazon EC2 AMI 的权限
- 复制 AMI
- 停止待处理的 AMI 复制操作
- Amazon EC2 AMI 复制工作原理

注意事项

- 复制 AMI 的权限 – 您可以使用 IAM 策略授予或拒绝用户复制 AMI 的权限。从 2024 年 10 月 28 日开始，您可以为源 AMI 上的 `CopyImage` 操作指定资源级权限。新 AMI 的资源级权限与以前一样可用。
- 启动权限和 Amazon S3 存储桶权限 – Amazon 不会将启动权限或 Amazon S3 存储桶权限从源 AMI 复制到新 AMI。复制操作完成之后，您可以将启动许可和 Amazon S3 存储桶权限应用于新的 AMI。
- 标签 – 您只能复制附加到源 AMI 的用户定义的 AMI 标签。不会复制系统标签（前缀为 `aws:`）和由其他 Amazon Web Services 账户 附加的用户定义标签。复制 AMI 时，您可以向新 AMI 及其备份快照附加新标签。
- 基于时间的 AMI 复制配额 – 达到累积快照复制吞吐量配额后，后续基于时间的 AMI 复制请求将会失败。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Quotas for time-based copies。
- 支持的源-目标复制 – 源 AMI 的位置决定您是否可以复制它以及新 AMI 允许的目标：
  - 如果源 AMI 位于某个区域，您可以在该区域内复制它，或者将其复制到另一个区域、与该区域关联的 Outpost 或该区域的本地区域。
  - 如果源 AMI 位于某个本地区域，您可以在该本地区域内复制它，或者将其复制到该本地区域的父区域或具有相同父区域的某些其他本地区域。
  - 如果源 AMI 位于某个 Outpost 上，则您无法复制它。
- 源和目标的 CLI 参数 – 使用 CLI 时，支持通过以下参数来指定要复制的 AMI 的源位置和新 AMI 的目标。请注意，复制操作必须在目标区域中启动；如果省略 `--region` 参数，则目标将采用 Amazon CLI 设置中配置的默认区域。

源至目标	源参数	目标参数
区域到区域	`--source-region`	`--region`
区域到 Outpost	`--source-region`	`--destination-outpost-arn`（Outpost 的 ARN）
区域到本地区域	`--source-region` 必须是本地区域的父区域。	`--destination-availability-zone`（本地区域的名称）或 `--destination-availability-zone-id`（本地区域的 ID）

源至目标	源参数	目标参数
本地区域到区域	`--source-region` 必须是本地区域的父区域。 源本地区域是根据指定源 AMI ID 的位置采用的。	`--region` 必须是本地区域的父区域。
本地区域到本地区域	`--source-region` 必须是本地区域的父区域。 源本地区域是根据指定源 AMI ID 的位置采用的。	`--destination-availability-zone`（本地区域的名称）或 `--destination-availability-zone-id`（本地区域的 ID）

成本

如果未指定完成时间，则复制 AMI 不会产生任何费用。而基于时间的 AMI 复制操作会产生额外的费用。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Time-based copies。

将会收取标准的存储和数据传输费。如果您复制由 EBS 支持的 AMI，则任何其他 EBS 快照的存储将会产生费用。

授予复制 Amazon EC2 AMI 的权限

要复制 EBS-backed AMI 或 Amazon S3 支持的 AMI，您需要以下 IAM 权限：

- `ec2:CopyImage` – 复制 AMI。对于 EBS-backed AMI，它还授予复制 AMI 支持快照的权限。
- `ec2:CreateTags` – 标记目标 AMI。对于 EBS-backed AMI，它还授予标记目标 AMI 支持快照的权限。

如果您要复制由实例存储支持的 AMI，则需要以下额外的 IAM 权限：

- `s3:CreateBucket` – 在目标区域为新的 AMI 创建 S3 存储桶
- `s3:PutBucketOwnershipControls` – 为新创建的 S3 存储桶启用 ACL，以便可以使用 `aws-exec-read` 标准 ACL 写入对象
- `s3:GetBucketAcl` – 读取源存储桶的 ACL

- `s3:ListAllMyBuckets` – 在目标区域为 AMI 查找现有 S3 存储桶
- `s3:GetObject` – 读取源存储桶中的对象
- `s3:PutObject` – 将对象写入目标存储桶
- `s3:PutObjectAcl` – 将新对象的权限写入目标存储桶

> **Note**
>
> 从 2024 年 10 月 28 日开始，您可以为源 AMI 上的 `CopyImage` 操作指定资源级权限。目标 AMI 的资源级权限与以前一样可用。有关更多信息，请参阅《服务授权参考》中的 Amazon EC2 定义的操作下的表中的 CopyImage。

用于复制 EBS-backed AMI 并标记目标 AMI 和快照的 IAM policy 示例

以下示例策略授予您复制任何 EBS-backed AMI 并标记目标 AMI 及其支持快照的权限。

> **Note**
>
> 从 2024 年 10 月 28 日开始，您可以在 `Resource` 元素中指定快照。有关更多信息，请参阅《服务授权参考》中的 Amazon EC2 定义的操作下的表中的 CopyImage。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "PermissionToCopyAllImages",
        "Effect": "Allow",
        "Action": [
            "ec2:CopyImage",
            "ec2:CreateTags"
        ],
        "Resource": [
            "arn:aws:ec2:*::image/*",
            "arn:aws:ec2:*::snapshot/*"
        ]
    }]
```

```
}
```

用于复制 EBS-backed AMI 但拒绝标记新快照的 IAM policy 示例

当您获取 ec2:CopySnapshot 权限时，系统会自动授予 ec2:CopyImage 权限。可以明确拒绝标记新的支持快照的权限，从而覆盖 ec2:CreateTags 操作的 Allow 效果。

以下示例策略授予您复制任何 EBS-backed AMI 但拒绝您标记目标 AMI 的新支持快照的权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:::snapshot/*"
        }
    ]
}
```

用于复制 Amazon S3 支持的 AMI 并标记目标 AMI 的 IAM 策略示例

以下示例策略授予您将指定源存储桶中任何 Amazon S3 支持的 AMI 复制到指定区域并标记目标 AMI 的权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "PermissionToCopyAllImages",
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amis-for-111122223333-in-us-east-2-hash"
            ]
        }
    ]
}
```

要查找 AMI 源存储桶的 Amazon Resource Name (ARN)，请访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台，在导航窗格中选择 AMIs (AMI)，然后在 Source (源) 列中找到该存储桶名称。

> **Note**
>
> 仅在您首次将 Amazon S3 支持的 AMI 复制到单个区域时，才需要 `s3:CreateBucket` 权限。在此之后，将使用已在该区域中创建的 Amazon S3 存储桶来存储您将来复制到该区域的所有 AMIs。

复制 AMI

您可以复制您自己拥有的 AMI，也可以复制其他账户与您共享的 AMI。有关支持的源和目标组合，请参阅注意事项。

Console

复制 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从控制台导航栏中，选择包含 AMI 的区域。
3. 在导航窗格中，选择 AMI 以显示区域中可供您使用的 AMI 列表。
4. 如果看不到您要复制的 AMI，请选择其他筛选条件。您可以按 AMI 我拥有的、私有映像、公共映像和禁用映像进行筛选。
5. 选择要复制的 AMI，然后选择操作、复制 AMI。
6. 在复制亚马逊机器映像（AMI）页面上，指定以下信息：
   a. AMI copy name（AMI 副本名称）：新 AMI 的名称。您可以在名称中包含操作系统信息，因为 Amazon EC2 在显示有关 AMI 的详细信息时不提供该信息。
   b. Description（描述）：默认情况下，描述包括源 AMI 的相关信息，以便您能区分副本和原本。您可以按需更改此描述。
   c. Destination Region（目标区域）：在其中复制 AMI 的区域。有关更多信息，请参阅跨区域复制和跨账户复制。
   d. 复制标签：选中此复选框可在复制 AMI 时包含用户定义的 AMI 标签。不会复制系统标签（前缀为 `aws:`）和由其他 Amazon Web Services 账户 附加的用户定义标签。
   e. 基于时间的复制：您可以指定复制操作是在特定时限内完成还是尽力完成，如下所示：

- 要在特定时限内完成复制，请执行以下操作：
  - 选择启用基于时间的复制。
  - 对于完成时限，输入允许执行复制操作的分钟数（以 15 分钟为增量）。此完成期限将会应用到与该 AMI 关联的所有快照。

    有关更多信息，请参阅《Amazon EBS 用户指南》中的 Time-based copies。
- 要尽力完成复制，请执行以下操作：
  - 保持取消选择启用基于时间的复制。

f. （仅限 EBS-backed AMI）加密 AMI 副本的 EBS 快照：选择此复选框可加密目标快照，或使用不同的密钥对它们进行重新加密。如果启用默认加密，加密 AMI 副本的 EBS 快照复选框处于选中状态，无法清除。有关更多信息，请参阅 加密和复制。

g. （仅限 EBS-backed AMI）KMS 密钥：用于加密目标快照的 KMS 密钥。

h. 标签：您可以使用相同的标签来标记新的 AMI 和新快照，也可以使用不同的标签来标记它们。

- 要使用相同标签标记新的 AMI 和新快照，请选择将映像和快照一起标记。相同的标签将应用于新的 AMI 和创建的每个快照。
- 要使用不同的标签标记新的 AMI 和新的快照，请选择分别标记映像和快照。对新的 AMI 和创建的快照应用了不同的标签。但是，请注意，创建的所有新快照都获得相同的标签；您不能使用不同的标签来标记每个新快照。

要添加标签，请选择 Add tag (添加标签)，然后输入该标签的键和值。对每个标签重复此操作。

i. 准备好复制 AMI 时，选择复制 AMI。

新 AMI 的初始状态是 `Pending`。当状态为 `Available` 时，AMI 复制操作完成。

Amazon CLI

将 AMI 从一个区域复制到另一个区域

使用 copy-image 命令。您必须指定源和目标区域。您可以使用 `--source-region` 参数指定源区域。您可以使用 `--region` 参数指定目标区域（或者省略此参数以采用 Amazon CLI 设置中配置的默认区域）。

```
aws ec2 copy-image \
```

```
    --source-image-id ami-0abcdef1234567890 \
    --source-region us-west-2 \
    --name my-ami \
    --region us-east-1
```

要在复制期间加密目标快照，您必须指定以下额外的参数：--encrypted 和 --kms-key-id。

将 AMI 从区域复制到本地区域

使用 copy-image 命令。必须指定源和目标。您可以使用 --source-region 参数指定源区域。可以使用 --destination-availability-zone 参数指定目标本地区域（可以改用 --destination-availability-zone-id）。请注意，只能将 AMI 从一个区域复制到同一区域内的本地区域。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --destination-availability-zone cn-north-1-pkx-1a \
    --name my-ami \
    --region cn-north-1
```

将 AMI 从本地区域复制到区域

使用 copy-image 命令。必须指定源和目标。您可以使用 --source-region 参数指定源区域。使用 --region 参数指定目标区域（或者省略此参数以采用 Amazon CLI 设置中配置的默认区域）。源本地区域是根据指定源 AMI ID 的位置采用的。请注意，只能将 AMI 从本地区域复制到其父区域。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --name my-ami \
    --region cn-north-1
```

将 AMI 从一个本地区域复制到另一个本地区域

使用 copy-image 命令。必须指定源和目标。可以使用 --source-region 参数指定本地区域的源区域。可以使用 --destination-availability-zone 参数指定目标本地区域（可以改用 --destination-availability-zone-id）。源本地区域是根据指定源 AMI ID 的位置采用的。使用 --region 参数指定目标本地区域的父区域（或者省略此参数以采用 Amazon CLI 设置中配置的默认区域）。

```
aws ec2 copy-image \
    --source-image-id ami-0abcdef1234567890 \
    --source-region cn-north-1 \
    --destination-availability-zone cn-north-1-pkx-1a \
    --name my-ami \
    --region cn-north-1
```

PowerShell

将 AMI 从一个区域复制到另一个区域

使用 Copy-EC2Image cmdlet。您必须指定源和目标区域。您可以使用 -SourceRegion 参数指定源区域。您可以使用 -Region 参数或 Set-AWSDefaultRegion cmdlet 指定目标区域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion us-west-2 `
    -Name my-ami `
    -Region us-east-1
```

要在复制期间加密目标快照，您必须指定以下额外的参数：-Encrypted 和 -KmsKeyId。

将 AMI 从区域复制到本地区域

使用 Copy-EC2Image cmdlet。必须指定源和目标。您可以使用 -SourceRegion 参数指定源区域。可以使用 -DestinationAvailabilityZone 参数指定目标本地区域（可以改用 -DestinationAvailabilityZoneId）。请注意，只能将 AMI 从一个区域复制到同一区域内的本地区域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -DestinationAvailabilityZone cn-north-1-pkx-1a `
    -Name my-ami `
    -Region cn-north-1
```

将 AMI 从本地区域复制到区域

使用 Copy-EC2Image cmdlet。必须指定源和目标。您可以使用 -SourceRegion 参数指定源区域。使用 -Region 参数或 Set-AWSDefaultRegion cmdlet 指定目标区域。源本地区域是根据指定源 AMI ID 的位置采用的。请注意，只能将 AMI 从本地区域复制到其父区域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -Name my-ami `
    -Region cn-north-1
```

将 AMI 从一个本地区域复制到另一个本地区域

使用 Copy-EC2Image cmdlet。必须指定源和目标。可以使用 -SourceRegion 参数指定本地区域的源区域。可以使用 -DestinationAvailabilityZone 参数指定目标本地区域（可以改用 -DestinationAvailabilityZoneId）。源本地区域是根据指定源 AMI ID 的位置采用的。使用 -Region 参数或 Set-AWSDefaultRegion cmdlet 指定目标本地区域的父区域。

```
Copy-EC2Image `
    -SourceImageId ami-0abcdef1234567890 `
    -SourceRegion cn-north-1 `
    -DestinationAvailabilityZone cn-north-1-pkx-1a `
    -Name my-ami `
    -Region cn-north-1
```

停止待处理的 AMI 复制操作

您可以使用以下过程停止待处理的 AMI 复制。

Console

停止 AMI 复制操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航栏中，从区域选择器中选择目标区域。
3. 在导航窗格中，选择 AMIs。
4. 选择要停止复制的 AMI，然后选择操作、取消注册 AMI。
5. 当系统要求您确认时，选择 Deregister AMI（取消注册 AMI）。

Amazon CLI

停止 AMI 复制操作

使用 deregister-image 命令。

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

PowerShell

使用以下方法停止 AMI 复制操作

使用 Unregister-EC2Image cmdlet。

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

Amazon EC2 AMI 复制工作原理

复制源 AMI 将生成完全相同但独立的新 AMI（我们也称之为目标 AMI）。目标 AMI 有自己的唯一 AMI ID。您可以更改或注销源 AMI，这不会对目标 AMI 产生任何影响。反之亦然。

对于 EBS-backed AMI，每个支持快照将会复制到完全相同但独立的目标快照。如果您将 AMI 复制到新区域，则快照是完整的（非增量）副本。如果您加密未加密的备份快照或将其加密到新 KMS 密钥，则快照是完整（非增量）副本。后续的 AMI 复制操作会生成备份快照的增量副本。

内容

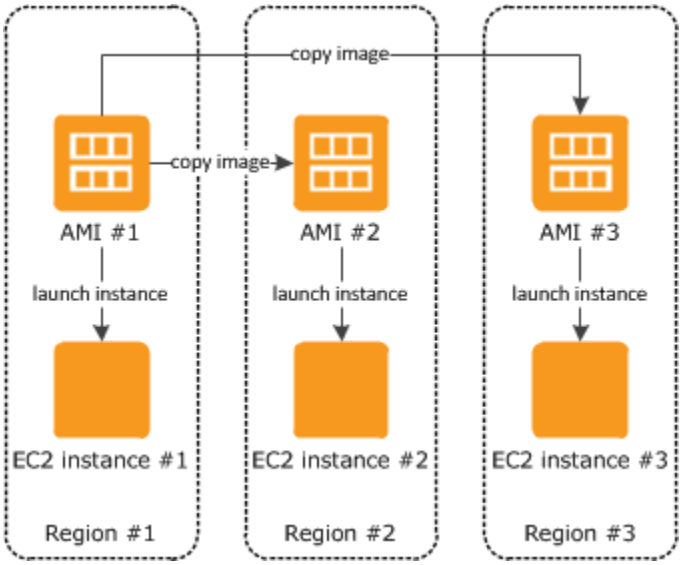
- 跨区域复制
- 跨账户复制
- 基于时间的 AMI 复制操作
- 加密和复制

跨区域复制

跨不同地理位置复制 AMI 具有以下优势：

- 一致的全球部署：通过将 AMI 从一个区域复制到另一个区域，您可以根据相同的 AMI 在不同的区域中启动一致的实例。
- 可扩展性：无论用户身处何处，您都可以更轻松地设计和构建能满足他们需求的全球应用程序。
- 性能：您可以通过分发您的应用程序以及找到较接近您用户的应用程序的关键组件来提高性能。您还可以利用区域特定的功能，例如，实例类型或其他 Amazon 服务。
- 高可用性：您可以跨 Amazon 区域设计和部署应用程序以提高可用性。

下图显示源 AMI、在不同的区域中复制的两个 AMI 以及从它们中启动的 EC2 实例之间的关系。从 AMI 中启动实例时，该实例位于 AMI 所在的区域中。如果您更改源 AMI，并希望在目标区域中的 AMIs 上反映这些更改，您必须将源 AMI 重新复制到目标区域中。



在首次将 Amazon S3 支持的 AMI 复制到一个区域时，我们为复制到该区域的 AMI 创建一个 Amazon S3 存储桶。复制到该区域的所有 Amazon S3 支持的 AMI 存储在该存储桶中。存储桶名称具有以下格式：amis-for-*account*-in-*region*-*hash*。例如：`amis-for-123456789012-in-us-east-2-yhjmxvp6`。

先决条件

在复制 AMI 之前，您必须确保更新源 AMI 的内容以支持在不同的区域中运行。例如，您应更新任何数据库连接字符串或相似的应用程序配置数据，以指向适当的资源。否则，从目标区域上的新 AMI 中启动的实例可能仍会使用源区域中的资源，这可能会影响性能和成本。

限制

- 目标区域限制为 300 个并发 AMI 复制操作。这也适用于基于时间的 AMI 复制操作。
- 您无法将半虚拟化（PV）AMI 复制到不支持 PV AMI 的区域。有关更多信息，请参阅 虚拟化类型。

跨账户复制

如果与您的 Amazon Web Services 账户 共享另一 Amazon Web Services 账户 的 AMI，您可以复制共享的 AMI。这称为跨账户复制。与您共享的 AMI 就是源 AMI。复制源 AMI 时，即创建了一个新的 AMI。新的 AMI 通常称为目标 AMI。

AMI 成本

- 对于共享 AMI，共享 AMI 的账户要支付该区域的存储费用。
- 如果您复制了与您的账户共享的 AMI，则您就是账户中目标 AMI 的所有者。
  - 源 AMI 的所有者需要支付标准 Amazon EBS 或 Amazon S3 传输费用。
  - 您需要支付目标区域中的目标 AMI 存储费用。

资源权限

要复制其他账户与您共享的 AMI，源 AMI 的所有者必须授予您对支持该 AMI 的存储的读取权限，而不仅限于该 AMI 本身的权限。存储是关联的 EBS 快照（用于 Amazon EBS-backed AMI）或关联的 S3 存储桶（用于 Amazon S3 支持的 AMI）。如果共享 AMI 具有加密快照，则拥有者必须同时与您共享一个或多个密钥。有关授予资源权限的更多信息，对于 EBS 快照，请参阅《Amazon EBS 用户指南》中的 Share an Amazon EBS snapshot with other Amazon Web Services 账户；对于 S3 存储桶，请参阅《Amazon S3 用户指南》中的 Amazon S3 的身份和访问管理。

> **Note**
>
> 附加到源 AMI 的标签不会跨账户复制到目标 AMI。

基于时间的 AMI 复制操作

当您为具有单个关联快照的 EBS 支持的 AMI 启动基于时间的 AMI 复制操作时，其行为方式将与单个基于时间的快照复制操作相同，并且具有相同的吞吐量限制。

当您为具有多个关联快照的 EBS 支持的 AMI 启动基于时间的 AMI 复制操作时，其行为方式将与并发基于时间的快照复制操作相同，并且具有相同的吞吐量限制。每个关联的快照都会导致一个单独的快照复制请求，并且每个请求都会占用您的累积快照复制吞吐量配额。您指定的完成期限将应用于每个关联的快照。

有关更多信息，请参阅《Amazon EBS 用户指南》中的 Time-based copies。

加密和复制

下表显示了各种 AMI 复制场景的加密支持。尽管可以复制未加密快照来生成加密快照，但是不能复制加密快照来生成未加密快照。

场景	描述	支持
1	未加密到未加密	是
2	已加密到已加密	是
3	未加密到已加密	是
4	已加密到未加密	否

> **Note**
>
> 在 CopyImage 操作期间加密仅适用于 Amazon EBS-backed AMI。由于 Amazon S3 支持的 AMI 不使用快照，因此您不能使用复制来更改其加密状态。

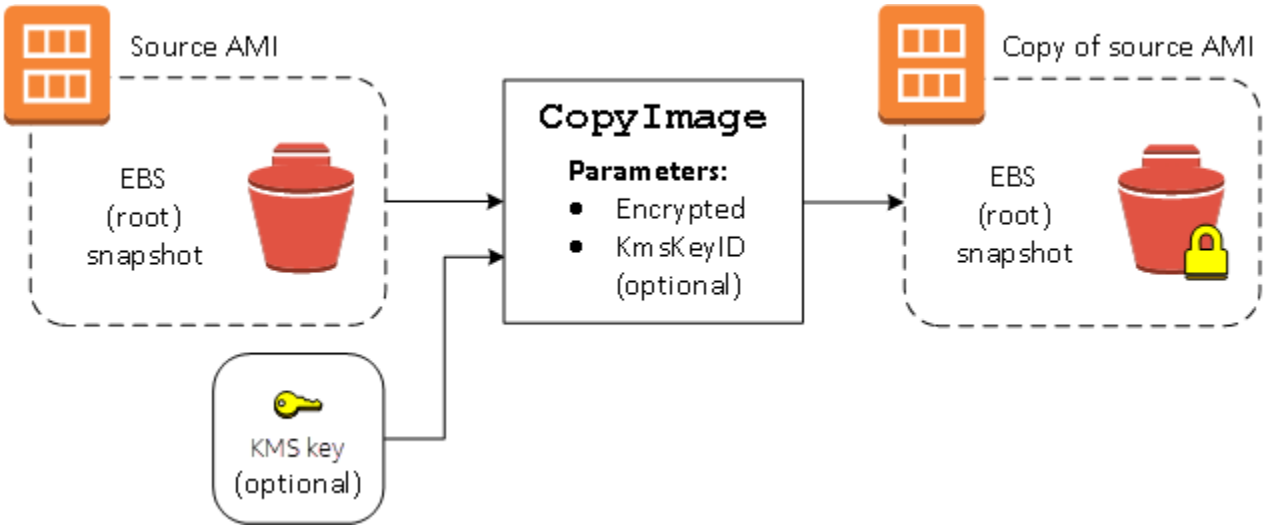
当您复制 AMI 而不指定加密参数时，默认情况下将复制备份快照及其原始加密状态。因此，如果源 AMI 由未加密的快照支持，则生成的目标快照也将处于未加密状态。同样，如果源 AMI 的快照已加密，则生成的目标快照也将使用相同的 Amazon KMS 密钥进行加密。对于多个快照支持的 AMI，每个目标快照都会保留其对应源快照的加密状态。

要在 AMI 复制期间更改目标备份快照的加密状态，可以指定加密参数。以下示例显示非默认情况，其中使用 CopyImage 操作指定加密参数来更改目标 AMI 的加密状态。

将未加密的源 AMI 复制到加密目标 AMI

在这种情况下，由未加密根快照支持的 AMI 会复制到带加密根快照的 AMI。CopyImage 操作将使用两个加密参数（包括一个客户托管密钥）调用。因此，根快照的加密状态发生变化，这样目标 AMI 由与原快照具有相同数据的根快照支持，但使用指定的密钥进行加密。在两个 AMI 中，您都会产生快照的存储成本，以及从任一 AMI 启动的任何实例的费用。

> **Note**
>
> 启用默认加密与将 AMI 中所有快照的 Encrypted 参数设置为 true 具有相同的效果。



设置 Encrypted 参数会加密此实例的单个快照。如果您未指定 KmsKeyId 参数，会使用默认客户托管密钥来加密快照副本。

有关复制带加密快照的 AMIs 的更多信息，请参阅将加密与 EBS 支持的 AMI 结合使用。

使用 S3 存储和还原 AMI

您可以将亚马逊机器映像（AMI）存储在 Amazon S3 存储桶中、将 AMI 复制到另一个 S3 存储桶，然后从 S3 存储桶还原它。通过使用 S3 存储桶存储和还原 AMI，您可以将 AMI 从一个 Amazon 分区复制到另一个分区，例如，从主商业分区到 Amazon GovCloud (US) 分区。您还可以通过将 AMI 存储在 S3 存储桶中为其创建存档副本。

支持使用 S3 存储和还原 AMI 的 API 是 CreateStoreImageTask、DescribeStoreImageTasks 和 CreateRestoreImageTask。

CopyImage 是推荐用于在 Amazon 分区内复制 AMI 的 API。但是，CopyImage 无法将 AMI 复制到另一个分区。

有关 Amazon 分区的信息，请参阅 IAM 用户指南中 Amazon 资源名称(ARN)页面上的##。

> **Warning**
>
> 在 Amazon 分区或 Amazon 区域之间移动数据时，请确保遵守所有适用法律和业务要求，包括但不限于任何适用的政府法规和数据驻留要求。

内容

- 使用案例
- 限制
- 成本
- AMI 存储和还原的工作原理
- 创建存储映像任务

使用案例

使用存储和还原 API 执行以下操作：

- 在 Amazon 分区之间复制 AMI
- 制作 AMI 的存档副本

在 Amazon 分区之间复制 AMI

通过使用 S3 存储桶存储和还原 AMI，您可以将 AMI 从一个 Amazon 分区复制到另一个分区，或从一个 Amazon 区域复制到另一个区域。在以下示例中，您可将 AMI 从主商业分区复制到 Amazon GovCloud (US) 分区，尤其是从 `us-east-2` 区域复制到 `us-gov-east-1` 区域。

要将 AMI 从一个分区复制到另一个分区，请执行以下步骤：

- 使用 `CreateStoreImageTask` 将 AMI 存储在当前区域内的 S3 存储桶中。在此示例中，S3 存储桶位于 `us-east-2` 中。
- 使用 `DescribeStoreImageTasks` 监控存储任务的进度。任务完成后，对象将在 S3 存储桶中可见。
- 使用您选择的程序将存储的 AMI 对象复制到目标分区中的 S3 存储桶。在此示例中，S3 存储桶位于 `us-gov-east-1` 中。

> **Note**
>
> 由于每个分区均需要不同的 Amazon 凭证，因此无法将 S3 对象从一个分区直接复制到另一个分区。跨分区复制 S3 对象的过程不在本文档的讨论范围之内。我们提供以下复制过程作为示例，但您必须使用符合安全要求的复制过程。
>
> - 要跨分区复制某一 AMI，复制过程可能非常简单，如下所示：从源存储桶下载对象到中间主机（例如，EC2 实例或笔记本电脑），然后从中间主机上传对象到目标存储桶。对于该过程的每个阶段，请对分区使用 Amazon 凭证。

- 要实现更持久的使用，请考虑开发一个管理副本的应用程序（可能使用 S3 分段下载和上传）。

- 使用 `CreateRestoreImageTask` 从目标分区中的 S3 存储桶还原 AMI。在此示例中，S3 存储桶位于 `us-gov-east-1` 中。
- 通过对需要检查其状态何时变为可用的 AMI 进行描述来监控还原任务的进度。您还可以通过描述快照来监控构成所还原 AMI 的快照的进度百分比。

制作 AMI 的存档副本

您可以通过将 AMI 存储在 S3 存储桶中为其创建存档副本。AMI 被打包到 S3 内的单个对象中，所有 AMI 元数据（不包括共享信息）都作为存储 AMI 的一部分得到保留。AMI 数据将作为存储流程的一部分进行压缩。由于 AMI 包含可轻松压缩的数据，因此 S3 中的对象较小。为了降低成本，您可以使用更便宜的 S3 存储套餐。有关更多信息，请参阅 Amazon S3 存储类和 Amazon S3 定价

限制

- 要存储 AMI，您的 Amazon Web Services 账户必须拥有该 AMI 及其快照的所有权，或者该 AMI 及其快照必须直接与您的账户共享。如果 AMI 仅公开共享，则无法存储。
- 使用这些 API 只能存储由 EBS 支持的 AMI。
- 不支持半虚拟化 (PV) AMI。
- 可存储的 AMI 的大小（压缩前）限制为 5,000 GB。
- 存储映像请求的配额：1200 GB 的存储工作（快照数据）正在进行中。
- 还原映像请求的配额：600 GB 的还原工作（快照数据）正在进行中。
- 在存储任务的持续时间内，不得删除快照，执行存储的 IAM 委托人必须有权访问快照，否则存储流程将会失败。
- 您不能在同一 S3 存储桶中创建 AMI 的多个副本。
- 存储在 S3 存储桶中的 AMI 无法使用其原始 AMI ID 进行还原。您可以使用 AMI 别名来缓解这种情况。
- 目前，仅支持使用 Amazon Command Line Interface、Amazon 开发工具包和 Amazon EC2 API 存储和还原 API。您无法使用 Amazon EC2 控制台存储和还原 AMI。

成本

使用 S3 存储和还原 AMI 时，您需要为存储和还原 API 使用的服务以及数据传输付费。API 使用 S3 和 EBS Direct API（由这些 API 在内部使用以访问快照数据）。有关更多信息，请参阅 Amazon S3 定价和 Amazon EBS 定价。

AMI 存储和还原的工作原理

要使用 S3 存储和还原 AMI，请使用以下 API：

- `CreateStoreImageTask` – 将 AMI 存储在 S3 存储桶中
- `DescribeStoreImageTasks` – 提供 AMI 存储任务的进度
- `CreateRestoreImageTask` – 从 S3 存储桶还原 AMI

API 的工作原理

- CreateStoreImageTask
- DescribeStoreImageTasks
- CreateRestoreImageTask
- 文件路径

CreateStoreImageTask

`CreateStoreImageTask` API 将 AMI 作为单个对象存储在 S3 存储桶中。

API 创建一个任务，从 AMI 及其快照中读取所有数据，然后使用 S3 分段上传将数据存储在 S3 对象中。API 获取 AMI 的所有组件，包括大多数非区域特定的 AMI 元数据以及 AMI 中包含的所有 EBS 快照，然后将它们打包到 S3 内的单个对象中。数据将作为上传流程的一部分进行压缩，以减少 S3 中使用的空间量，因此 S3 中的对象可能小于 AMI 中快照大小总和。

如果调用此 API 的账户有可见的 AMI 和快照标签，则会保留它们。

S3 中的对象的 ID 与 AMI 的相同，但带有 `.bin` 扩展名。以下数据还作为 S3 对象上的 S3 元数据标签存储：AMI 名称、AMI 描述、AMI 注册日期、AMI 拥有者账户以及存储操作的时间戳。

完成任务所需的时间取决于 AMI 的大小。它还取决于有多少其他任务正在进行，因为任务需要排队。您可以通过调用 `DescribeStoreImageTasks` API 跟踪任务的进度。

所有正在进行的 AMI 的大小总和限制为每个账户 1200 GB 的 EBS 快照数据。进一步的任务创建将被拒绝，直到正在进行的任务低于限制。例如，如果当前正在存储快照数据为 200 GB 的 AMI 和另一个快照数据为 400 GB 的 AMI，则将接受另一个请求，因为正在进行的总量为 600 GB，低于限制。但是，如果当前正在存储的单个 AMI 的快照数据为 1200 GB，则在任务完成之前，进一步的任务都将被拒绝。

DescribeStoreImageTasks

`DescribeStoreImageTasks` API 描述 AMI 存储任务的进度。您可以描述指定 AMI 的任务。如果未指定 AMI，则会获得过去 31 天内处理的所有存储映像任务的分页列表。

对于每个 AMI 任务，响应会指示任务是 `InProgress`、`Completed` 还是 `Failed`。对于任务 `InProgress`，响应会将估计进度显示为百分比值。

任务按反向的时间顺序列出。

目前，只能查看上个月的任务。

CreateRestoreImageTask

`CreateRestoreImageTask` API 启动一个任务，该任务可从先前使用 `CreateStoreImageTask` 请求创建的 S3 对象还原 AMI。

执行还原任务的区域可以与执行存储任务的区域相同，也可以不同。

要从中还原 AMI 对象的 S3 存储桶必须位于请求执行还原任务的相同区域中。AMI 将在此区域中还原。

AMI 将使用其元数据还原，例如与存储的 AMI 值对应的名称、描述和块储存设备映射。名称对该账户在该区域中的 AMI 必须唯一。如果未提供名称，则新 AMI 获得的名称与原始 AMI 的名称相同。AMI 获得在还原流程中生成的新 AMI ID。

完成 AMI 还原任务所需的时间取决于 AMI 的大小。它还取决于有多少其他任务正在进行，因为任务需要排队。您可以通过描述 AMI (describe-images) 或其 EBS 快照 (describe-snapshots) 来查看任务的进度。如果任务失败，AMI 和快照将移至失败状态。

所有正在进行的 AMI 的大小总和限制为每个账户的 EBS 快照数据 600 GB（根据还原后的大小）。进一步的任务创建将被拒绝，直到正在进行的任务低于限制。

文件路径

您可以通过以下方式在存储和恢复 AMI 时使用文件路径：

- 在 S3 中存储 AMI 时，可以将文件路径添加到存储桶名称中。在内部，系统将路径与存储桶名称分开，然后将路径添加到为存储 AMI 而生成的对象密钥中。完整的对象路径显示在 API 调用的响应中。
- 在恢复 AMI 时，由于对象密钥参数可用，可以将路径添加到对象键值的开头。

示例：带有附加文件路径的存储桶名称

存储 AMI 时，请在存储桶名称后指定文件路径。

```
amzn-s3-demo-bucket/path1/path2
```

以下是结果对象键。

```
path1/path2/ami-0abcdef1234567890.bin
```

还原 AMI 时，请指定存储桶名称和对象键。有关示例，请参阅 the section called “创建存储映像任务”。

创建存储映像任务

将 AMI 存储在 S3 存储桶中时，将创建存储映像任务。您可以使用存储映像任务来监控该过程的进度和结果。

内容

- 保护您的 AMI
- 使用 S3 存储和还原 AMI 的权限
- 创建存储映像任务
- 创建还原映像任务

保护您的 AMI

务必确保 S3 存储桶配置有足够的安全性来保护 AMI 的内容，并确保只要 AMI 对象仍保留在存储桶中，安全性将保持不变。如果无法做到这一点，建议不要使用这些 API。确保不允许对 S3 存储桶进行公开访问。我们建议为存储 AMI 的 S3 存储桶启用服务器端加密，但不是必需要求。

有关如何为 S3 存储桶设置适当的安全设置的信息，请查看以下安全主题：

- 阻止对 Amazon S3 存储的公有访问

- 设置 Amazon S3 存储桶的默认服务器端加密行为
- 我可以使用哪个 S3 桶策略来遵守 Amazon Config 规则 s3-bucket-ssl-requests-only？
- 启用 Amazon S3 服务器访问日志记录

当 AMI 快照复制到 S3 对象时，将通过 TLS 连接复制数据。您可以使用加密快照存储 AMI，但是快照会作为存储流程的一部分进行解密。

使用 S3 存储和还原 AMI 的权限

如果您的 IAM 主体将使用 Amazon S3 来存储或还原 AMI，则需要向其授予所需权限。

以下示例策略包括允许 IAM 委托人执行存储和还原任务需要的所有操作。

此外，您还可以创建向主体授予仅访问指定资源权限的 IAM policy。如需了解更多示例策略，请参阅《IAM 用户指南》中的 Amazon 资源的访问管理。

> **Note**
>
> 如果构成 AMI 的快照已加密，或者账户默认启用了加密功能，您的 IAM 主体必须具有使用 KMS 密钥的权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:PutObjectTagging",
                "s3:AbortMultipartUpload",
                "ebs:CompleteSnapshot",
                "ebs:GetSnapshotBlock",
                "ebs:ListChangedBlocks",
                "ebs:ListSnapshotBlocks",
```

```
                "ebs:PutSnapshotBlock",
                "ebs:StartSnapshot",
                "ec2:CreateStoreImageTask",
                "ec2:DescribeStoreImageTasks",
                "ec2:CreateRestoreImageTask",
                "ec2:GetEbsEncryptionByDefault",
                "ec2:DescribeTags",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

创建存储映像任务

要将 AMI 存储在 S3 存储桶中，首先要创建存储映像任务。完成任务所需的时间取决于 AMI 的大小。您可以跟踪该任务的进度，直到其成功或失败。

Amazon CLI

创建存储映像任务

使用 create-store-image-task 命令。

```
aws ec2 create-store-image-task \
    --image-id ami-0abcdef1234567890 \
    --bucket amzn-s3-demo-bucket
```

下面是示例输出。

```
{
  "ObjectKey": "ami-0abcdef1234567890.bin"
}
```

描述存储映像任务的进度

使用 describe-store-image-tasks 命令。

```
aws ec2 describe-store-image-tasks \
    --image-ids ami-0abcdef1234567890 \
```

```
    --query StoreImageTaskResults[].StoreTaskState \
    --output text
```

下面是示例输出。

```
InProgress
```

PowerShell

创建存储映像任务

使用 New-EC2StoreImageTask cmdlet。

```
New-EC2StoreImageTask `
    -ImageId ami-0abcdef1234567890 `
    -Bucket amzn-s3-demo-bucket
```

下面是示例输出。

```
ObjectKey         : ami-0abcdef1234567890.bin
```

描述存储映像任务的进度

使用 Get-EC2StoreImageTask cmdlet。

```
(Get-EC2StoreImageTask -ImageId ami-0abcdef1234567890).StoreTaskState
```

下面是示例输出。

```
InProgress
```

创建还原映像任务

必须为还原的 AMI 指定名称。名称对该账户在该区域中的 AMI 必须唯一。还原的 AMI 将获得一个新 AMI ID。

Amazon CLI

创建恢复映像任务

使用 create-restore-image-task 命令。

```
aws ec2 create-restore-image-task \
    --object-key ami-0abcdef1234567890.bin \
    --bucket amzn-s3-demo-bucket \
    --name "my-restored-ami"
```

下面是示例输出。

```
{
   "ImageId": "ami-1234567890abcdef0"
}
```

PowerShell

创建恢复映像任务

使用 New-EC2RestoreImageTask cmdlet。

```
New-EC2RestoreImageTask `
    -ObjectKey ami-0abcdef1234567890.bin `
    -Bucket amzn-s3-demo-bucket `
    -Name "my-restored-ami"
```

下面是示例输出。

```
ImageId         : ami-1234567890abcdef0
```

使用 AMI 世系跟踪 AMI 的源

AMI 世系通过返回所有原级 AMI 的 ID 和区域来帮助您跟踪 AMI 的源。当您创建或复制 AMI 时，新 AMI 会保留其源（父）AMI 的 ID 和区域。这使您能够跟踪 AMI 的链，直至根 AMI。

主要优势

使用 AMI 世系可以帮助您：

- 跟踪 AMI 衍生品以确保符合内部策略。
- 当在原级 AMI 中发现安全问题时，识别潜在易受攻击的 AMI。
- 在多个区域保持 AMI 源的可见性。

主题

- AMI 世系的工作原理
- 注意事项
- 查看 AMI 世系
- 识别源 AMI

AMI 世系的工作原理

AMI 世系识别用于创建指定 AMI 的父 AMI、父级的父级，依此类推，直至根 AMI。下面将介绍操作方式：

- 每个 AMI 都会显示其源（父）AMI 的 ID 和区域。
- 从所选 AMI 开始，世系条目列表按顺序显示每个父 AMI。
- 世系条目列表可以追溯到根 AMI。根 AMI 可以是以下任一项：
  - 来自经过验证的提供商的公共 AMI（由其所有者别名标识，即 `amazon` 或 `aws-marketplace`）。
  - 没有记录的原级的 AMI。例如，当使用 RegisterImage 直接从一组快照创建 AMI 时，与从实例创建 AMI 时不同，此时没有要跟踪的源 AMI。
  - 一个 AMI，其源 AMI 来自不同的分区。
  - 列表中的第 50 个 AMI。世系列表中最多可以包含 50 个 AMI。

注意事项

- 源 AMI 的 ID 和区域仅适用于使用 CreateImage、CopyImage 或 CreateRestoreImageTask 创建的 AMI。
- 对于使用 CreateImage 创建的 AMI（从实例创建 AMI），源 AMI ID 是用于启动实例的 AMI 的 ID。
- 源 AMI 信息不适用于：
  - 使用 RegisterImage 创建的 AMI，因为它们是根据快照创建的。
  - 对于一些较旧 AMI。
- 在以下情况下会保留源 AMI 信息：
  - 跨多个区域复制 AMI。
  - 源 AMI 已被注销（删除）。
  - 您无权访问源 AMI。

- 每个世系列表最多只能包含 50 个 AMI。

查看 AMI 世系

您可以使用以下方法查看 AMI 的世系。

Console

查看 AMI 的世系

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择一个 AMI，然后选择 AMI 世系选项卡。
4. AMI 世系条目表列出了世系列表中的所有 AMI。
   - AMI ID：世系列表中每个 AMI 的标识符。表中的第一个条目是选定的 AMI，后面是其原级。
   - 源 AMI ID：创建 AMI ID 列中的 AMI 所依据的 AMI 的 ID。破折号（-）表示 AMI 世系列表的结尾。
   - 源 AMI 区域：源 AMI 所在的 Amazon Web Services 区域。
   - 世系等级：世系列表中的位置，其中：
     - 0（输入 AMI）表示您想知道其世系的所选 AMI。
     - 数字越大，显示的原级更老。
     - ***n***（原始 AMI）表示根 AMI，数字表示世系列表可以追溯的层级深度。
   - 创建日期：AMI 的创建时间，采用 UTC 格式。
   - 所有者别名：AMI 所有者的别名（例如 amazon）。破折号（-）表示 AMI 没有所有者别名。

Amazon CLI

查看 AMI 的世系

使用 get-image-ancestry 命令并指定 AMI ID。

```
aws ec2 get-image-ancestry \
    --image-id ami-1111111111EXAMPLE \
    --region us-east-1
```

下面是示例输出。输出按照世系顺序列出 AMI：第一个条目是指定的（输入）AMI，后面依次是其父级、父级的父级，依此类推，直到根 AMI。

```
{
    "ImageAncestryEntries": [
        {
            "CreationDate": "2025-01-17T18:37:50.000Z",
            "ImageId": "ami-1111111111EXAMPLE", // Input AMI
            "SourceImageId": "ami-2222222222EXAMPLE",
            "SourceImageRegion": "us-east-1"

        },
        {
            "CreationDate": "2025-01-17T18:37:50.000Z",
            "ImageId": "ami-2222222222EXAMPLE", // Parent AMI
            "SourceImageId": "ami-3333333333EXAMPLE",
            "SourceImageRegion": "us-east-1"
        },
        ...
        {
            "CreationDate": "2025-01-17T18:37:50.000Z",
            "ImageId": "ami-8888888888EXAMPLE", // Root AMI
            "ImageOwnerAlias": "aws-marketplace",
            "SourceImageId": "ami-9999999999EXAMPLE",
            "SourceImageRegion": "us-east-2"
        }
    ]
}
```

PowerShell

查看 AMI 的世系

使用 Get-EC2ImageAncestry cmdlet。

```
Get-EC2ImageAncestry -ImageId ami-1111111111EXAMPLE
```

下面是示例输出。输出按照世系顺序列出 AMI：第一个条目是指定的（输入）AMI，后面依次是其父级、父级的父级，依此类推，直到根 AMI。

```
ImageAncestryEntries : {
    @{
        CreationDate = "2025-01-17T18:37:50.000Z"
```

```
        ImageId = "ami-1111111111EXAMPLE"    # Input AMI
        SourceImageId = "ami-2222222222EXAMPLE"
        SourceImageRegion = "us-east-1"
    },
    @{
        CreationDate = "2025-01-17T18:37:50.000Z"
        ImageId = "ami-2222222222EXAMPLE"    # Parent AMI
        SourceImageId = "ami-3333333333EXAMPLE"
        SourceImageRegion = "us-east-1"
    },
    ...
    @{
        CreationDate = "2025-01-17T18:37:50.000Z"
        ImageId = "ami-8888888888EXAMPLE"    # Root AMI
        ImageOwnerAlias = "aws-marketplace"
        SourceImageId = "ami-9999999999EXAMPLE"
        SourceImageRegion = "us-east-2"
    }
}
```

识别源 AMI

如果您只需要识别用于创建某个 AMI 的直接父级（源）AMI，则可以使用以下方法。

Console

识别用于创建所选 AMI 的源 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI 以查看其详细信息。

   源 AMI 信息显示在下面的字段中：源 AMI ID 和源 AMI 区域

Amazon CLI

识别用于创建指定 AMI 的源 AMI

使用 describe-images 命令。

```
aws ec2 describe-images \
```

```
    --region us-east-1 \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].{ID:SourceImageId,Region:SourceImageRegion}"
```

下面是示例输出。

```
[
    {
        "ID": "ami-0abcdef1234567890",
        "Region": "us-west-2"
    }
]
```

PowerShell

识别用于创建指定 AMI 的源 AMI

使用 Get-EC2Image cmdlet。

```
Get-EC2Image -ImageId ami-0abcdef1234567890 | Select SourceImageId,
 SourceImageRegion
```

下面是示例输出。

```
SourceImageId           SourceImageRegion
-------------           -----------------
ami-0abcdef1234567890 us-west-2
```

使用 AMI 水印来跟踪和识别 AMI

AMI 水印是向私有 AMI 附加的用于跟踪来源和执行治理策略的标识符。水印在整个 AMI 生命周期中持续存在：

- 如果基于从带水印的 AMI 启动的正在运行的实例创建新的 AMI，则新 AMI 会继承该水印。
- 如果复制带水印的 AMI，则副本会带有该水印。
- 如果将带水印的 AMI 存储到 S3 并进行还原，则还原后的 AMI 将保留该水印。
- 如果与其他账户共享带水印的 AMI，则接收方仍能看到该水印。

使用 AMI 水印有助于您：

主要优势

- 跨账户和区域跟踪来源：识别哪些 AMI 源自已批准的基本图像。
- 跨账户筛选并查找相关的 AMI。
- 帮助 AMI 使用者发现并识别与项目或组织相关联的可信 AMI。

主题

- AMI 水印的工作原理
- 所需的权限
- 向 AMI 附加水印
- 从 AMI 分离水印
- 查看 AMI 水印
- 按水印筛选 AMI

AMI 水印的工作原理

AMI 水印是向 AMI 附加的结构化标识符。下面介绍了水印的主要特征：

- 持续存在：当向 AMI 附加水印时，它会延续到所有衍生 AMI。
- 仅所有者：仅 AMI 所有者可以向 AMI 附加水印。
- 对所有人可见：任何有权访问 AMI 的人均可查看其水印。
- 上限为 5：一个 AMI 最多可以有 5 个水印。
- 不适用于公用 AMI：无法向公用 AMI 附加水印，也无法将带水印的 AMI 设为公有。
- 可筛选：使用 `describe-images` 时可以按水印筛选 AMI。

水印格式

水印是包含以下字段的结构化对象：

- `WatermarkKey`：水印的唯一标识符，由 *`account-id`*`:`*`watermark-name`* 组成。账户 ID 部分是 AMI 所有者的 12 位 Amazon 账户 ID。水印名称部分是客户指定的名称。
- `SourceImageRegion`：最初向其附加水印的 AMI 的区域。
- `SourceImageId`：最初向其附加水印的 AMI。
- `SourceImageCreationDate`：最初向其附加水印的 AMI 的创建日期。

- `WatermarkCreationTime`：应用水印的时间戳。

水印名称的长度必须为 3 到 128 个字符，且可包含字母数字字符、圆括号 (())、方括号 ([])、空格、句点 (.)、斜杠 (/)、破折号 (-)、单引号 (')、@ 符号或下划线 (_)。

所需的权限

要使用 AMI 水印，您需要以下 IAM 权限：

- `ec2:AttachImageWatermark`：向 AMI 附加水印。
- `ec2:DetachImageWatermark`：从 AMI 分离水印。
- `ec2:DescribeImages`：查看 AMI 上的水印。

向 AMI 附加水印

您可以使用控制台、Amazon CLI 或 PowerShell 向 AMI 附加水印。

Console

向 AMI 附加水印

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡的水印部分，选择管理水印。
5. 输入水印名称，然后选择附加。

Amazon CLI

向 AMI 附加水印

使用 attach-image-watermark 命令。

```
aws ec2 attach-image-watermark \
    --image-id ami-1111111111EXAMPLE \
    --image-watermark-name "prod-baseline"
```

下面是示例输出。

```
{
    "WatermarkKey": "123456789012:prod-baseline"
}
```

PowerShell

向 AMI 附加水印

使用 Add-EC2ImageWatermark cmdlet。

```
Add-EC2ImageWatermark `
    -ImageId ami-1111111111EXAMPLE `
    -ImageWatermarkName "prod-baseline"
```

一个 AMI 最多可以附加 5 个水印。

从 AMI 分离水印

您可以使用控制台、Amazon CLI 或 PowerShell 从 AMI 分离水印。

Console

从 AMI 分离水印

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡的水印部分，选择管理水印。
5. 选择要移除的水印，然后选择移除。

Amazon CLI

从 AMI 分离水印

使用 detach-image-watermark 命令。

```
aws ec2 detach-image-watermark \
    --image-id ami-1111111111EXAMPLE \
```

```
    --image-watermark-key "111122223333:prod-baseline"
```

PowerShell

从 AMI 分离水印

使用 Remove-EC2ImageWatermark cmdlet。

```
Remove-EC2ImageWatermark `
    -ImageId ami-1111111111EXAMPLE `
    -ImageWatermarkKey "111122223333:prod-baseline"
```

> **Note**
>
> 从 AMI 分离水印不会将其从已带该水印的衍生 AMI 中移除。为确保水印持续存在，请仅向需要管理水印的可信管理员授予 `ec2:DetachImageWatermark` 权限。

查看 AMI 水印

您可以使用控制台、Amazon CLI 或 PowerShell 查看 AMI 的水印。

Console

查看 AMI 的水印

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡的水印部分查看水印。

Amazon CLI

查看 AMI 的水印

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-046863d776a820ccd \
```

```
    --region us-east-1
```

响应包含每个 AMI 的 ImageWatermarks 数组。

```
{
    "Images": [
        {
            "ImageId": "ami-046863d776a820ccd",
            "Public": false,
            "OwnerId": "123456789012",
            ...
            "ImageWatermarks": [
                {
                    "WatermarkKey": "111122223333:prod-baseline",
                    "Region": "us-east-1",
                    "SourceImageId": "ami-0b752bf1df193a6c4",
                    "SourceImageCreationDate": "2024-07-10T08:15:00",
                    "CreationDate": "2024-07-12T14:30:00"
                },
                {
                    "WatermarkKey": "222222222222:security-approved",
                    "Region": "eu-north-1",
                    "SourceImageId": "ami-12345678",
                    "SourceImageCreationDate": "2024-06-01T10:00:00",
                    "CreationDate": "2024-06-05T09:45:00"
                }
            ]
        }
    ]
}
```

PowerShell

查看 AMI 的水印

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-046863d776a820ccd).ImageWatermarks
```

按水印筛选 AMI

您可以使用控制台、Amazon CLI 或 PowerShell 按水印筛选 AMI。

Console

按水印筛选 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 在搜索栏中，选择水印键筛选条件并输入水印键值。

Amazon CLI

按水印筛选 AMI

使用 describe-images 命令和 `image-watermark-key` 筛选条件。

```
aws ec2 describe-images \
    --filters "Name=image-watermark-key,Values=111122223333:prod-baseline"
```

这将返回您有权访问且带有指定水印的所有 AMI，包括通过复制操作继承了该水印的衍生 AMI。

PowerShell

按水印筛选 AMI

使用带 `-Filter` 参数的 Get-EC2Image cmdlet。

```
Get-EC2Image `
    -Filter @{Name="image-watermark-key"; Values="111122223333:prod-baseline"}
```

这将返回您有权访问且带有指定水印的所有 AMI，包括通过复制操作继承了该水印的衍生 AMI。

管理和监控 AMI 使用情况

Amazon 提供了多种功能来帮助您有效地管理和监控 AMI 使用情况。您可以跟踪哪些账户正在使用共享 AMI，确定 AMI 的上次使用时间，以及发现 Amazon Web Services 账户中哪些资源正在引用特定的 AMI。

下表概括介绍了用于管理和监控 AMI 使用情况的功能：

功能	使用案例	主要优势
AMI 使用情况报告	了解哪些 Amazon Web Services 账户正在使用您的 AMI 以及每个 AMI 的使用量。	• 识别引用 AMI 的 Amazon Web Services 账户和资源类型，以便您可以安全地取消注册或禁用 AMI。 • 识别未使用的 AMI 以取消注册，从而降低存储成本。 • 识别最常用的 AMI。
上次使用的跟踪	检查上次使用 AMI 的时间。	• 识别未使用的 AMI，以便您可以安全地取消注册 AMI。 • 识别未使用的 AMI 以取消注册，从而降低存储成本。
AMI 引用检查	确保 Amazon 资源使用最新的合规 AMI。	• 审核您账户中 AMI 的使用情况。 • 检查引用特定 AMI 的位置。 • 通过更新资源以引用最新的 AMI 来保持合规性。

主题

- 查看 AMI 使用情况
- 检查上次使用 Amazon EC2 AMI 的时间
- 识别引用指定 AMI 的资源

查看 AMI 使用情况

如果与其他 Amazon Web Services 账户共享 Amazon 系统映像 (AMI)（无论是与特定 Amazon Web Services 账户、组织、组织单元 (OU) 还是公开），可以通过创建 AMI 使用情况报告来了解这些 AMI 的使用情况。这些报告可让您了解以下内容：

- 哪些 Amazon Web Services 账户正在 EC2 实例或启动模板中使用您的 AMI
- 引用每个 AMI 的 EC2 实例或启动模板数量

AMI 使用情况报告可帮助您更有效地管理 AMI，方式如下：

- 识别引用 AMI 的 Amazon Web Services 账户和资源类型，以便您可以安全地取消注册或禁用 AMI。
- 识别未使用的 AMI 以取消注册，从而降低存储成本。
- 识别最常用的 AMI。

内容

- AMI 使用情况报告的工作原理
- 创建 AMI 使用情况报告
- 查看 AMI 使用情况报告
- 删除 AMI 使用情况报告
- 报告配额

AMI 使用情况报告的工作原理

在创建 AMI 使用情况报告时，可以指定：

- 要报告的 AMI。
- 要检查的 Amazon Web Services 账户（特定账户或所有账户）。
- 要检查的资源类型（EC2 实例、启动模板或两者）。
- 对于启动模板，要检查的版本数（默认为最新 20 个版本）。

Amazon EC2 为每个 AMI 创建一份单独的报告。每份报告都提供：

- 使用 AMI 的 Amazon Web Services 账户列表。

- 按每个账户的资源类型划分的引用 AMI 的资源计数。请注意，对于启动模板，如果在启动模板的多个版本中引用了某个 AMI，则计数仅为 1。

> **Important**
>
> 当您生成 AMI 使用情况报告时，它可能不包含最新的活动。过去 24 小时的实例活动和过去几天的启动模板活动可能不会出现在报告中。

Amazon EC2 会在报告创建 30 天后自动删除该报告。您可以从 EC2 控制台下载报告以在本地保留。

创建 AMI 使用情况报告

要查看 AMI 的使用情况，您必须首先创建 AMI 使用情况报告，指定要报告的账户和资源类型。报告创建完成后，您可以查看报告的内容。您也可以从 EC2 控制台下载报告。

Console

创建 AMI 使用情况报告

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择一个 AMI，然后选择操作、AMI 使用情况、查看我的 AMI 使用情况。
4. 在创建我的 AMI 使用情况报告页面上，执行下列操作：
   a. 对于资源类型，选择一个或多个要报告的资源类型。
   b. 对于账户 ID，执行下列操作之一：
      - 选择指定账户 ID，然后为要报告的每个账户选择添加账户 ID。
      - 选择包括所有帐户以报告所有帐户。
   c. 选择创建我的 AMI 使用情况报告。
5. 在 AMI 页面上，选择我的 AMI 使用情况选项卡。
6. 选择一个报告 ID 来查看其详细信息。

Amazon CLI

为账户列表创建 AMI 使用情况报告

使用带以下必需参数的 create-image-usage-report 命令：

- `--image-id` – 要报告的 AMI 的 ID。
- `--resource-types` – 要检查的资源类型。在下面的示例中，要检查的资源类型是 EC2 实例和启动模板。此外，还指定要检查的启动模板版本的数量 (`version-depth=100`)。

要报告特定账户，请使用 `--account-ids` 参数指定要报告的每个账户的 ID。

```
aws ec2 create-image-usage-report \
    --image-id ami-0abcdef1234567890 \
    --account-ids 111122223333 444455556666 123456789012 \
    --resource-types ResourceType=ec2:Instance \
      'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=100}]'
```

创建所有账户的 AMI 使用情况报告

要使用指定 AMI 报告所有账户，请使用相同的命令，但省略 `--account-ids` 参数。

```
aws ec2 create-image-usage-report \
    --image-id ami-0abcdef1234567890 \
    --resource-types ResourceType=ec2:Instance \
      'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=100}]'
```

下面是示例输出。

```
{
    "ReportId": "amiur-00b877d192f6b02d0"
}
```

监控报告创建状态

使用 describe-image-usage-reports 命令并指定报告 ID。

```
aws ec2 describe-image-usage-reports --report-ids amiur-00b877d192f6b02d0
```

下面是示例输出。`State` 字段的初始值为 `pending`。为了能够查看报告条目，状态必须为 `available`。

```
{
    "ImageUsageReports": [
        {
            "ImageId": "ami-0e9ae3dc21c2b3a64",
            "ReportId": "amiur-abcae3dc21c2b3999",
            "ResourceTypes": [
                {"ResourceType": "ec2:Instance"}
            ],
            "State": "pending",
            "CreationTime": "2025-09-29T13:27:12.322000+00:00",
            "ExpirationTime": "2025-10-28T13:27:12.322000+00:00"
        }
    ]
}
```

PowerShell

为账户列表创建 AMI 使用情况报告

使用带以下必需参数的 New-EC2ImageUsageReport cmdlet：

- -ImageId – 要报告的 AMI 的 ID。
- -ResourceType – 要检查的资源类型。在下面的示例中，要检查的资源类型是 EC2 实例和启动模板。此外，还指定要检查的启动模板版本的数量 ('version-depth' = *100*)。

要报告特定账户，请使用 -AccountId 参数指定要报告的每个账户的 ID。

```
New-EC2ImageUsageReport `
    -ImageId ami-0abcdef1234567890 `
    -AccountId 111122223333 444455556666 123456789012 `
    -ResourceType @(
        @{ResourceType = 'ec2:Instance'},
        @{ResourceType = 'ec2:LaunchTemplate'ResourceTypeOptions = @{'version-depth'
 = 100}
        })
```

创建所有账户的 AMI 使用情况报告

要使用指定 AMI 报告所有账户，请使用相同的命令，但省略 -AccountId 参数。

```
New-EC2ImageUsageReport `
```

```
    -ImageId ami-0abcdef1234567890 `
    -ResourceType @(
        @{ResourceType = 'ec2:Instance'},
        @{ResourceType = 'ec2:LaunchTemplate'ResourceTypeOptions = @{'version-depth'
 = 100}
        })
```

下面是示例输出。

```
ReportId
--------
amiur-00b877d192f6b02d0
```

监控报告创建状态

使用 Get-EC2ImageUsageReport 命令并指定报告 ID。

```
Get-EC2ImageUsageReport -ReportId amiur-00b877d192f6b02d0
```

下面是示例输出。State 字段的初始值为 pending。为了能够查看报告条目，状态必须为 available。

```
ImageUsageReports
-----------------
{@{ImageId=ami-0e9ae3dc21c2b3a64; ReportId=amiur-abcae3dc21c2b3999;
 ResourceTypes=System.Object[]; State=pending; CreationTime=2025-09-29;
 ExpirationTime=2025-10-28}}
```

查看 AMI 使用情况报告

可以查看过去 30 天内为 AMI 创建的所有使用情况报告。Amazon EC2 会在报告创建 30 天后自动删除该报告。

对于每个报告，您可以查看正在使用 AMI 的 Amazon Web Services 账户，并且对于每个帐户，可以按资源类型查看引用 AMI 的资源计数。您还可以查看启动报告创建的时间。仅当报告处于完成（控制台）或 available (Amazon CLI) 状态时，此信息才可用。

> **Important**
>
> 当您生成 AMI 使用情况报告时，它可能不包含最新的活动。过去 24 小时的实例活动和过去几天的启动模板活动可能不会出现在报告中。

Console

查看 AMI 使用情况报告

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 选择我的使用情况报告选项卡。

   报告列表显示：

   - 过去 30 天内为选定 AMI 生成的所有报告。
   - 对于每个报告，报告启动时间列显示创建报告的日期。
5. 选择报告的 ID 以查看其内容。
6. 要返回 AMI 详细信息页面上的我的使用情况报告选项卡，请选择查看此 AMI 的所有报告。

Amazon CLI

列出指定 AMI 的所有 AMI 使用情况报告

使用 describe-image-usage-reports 命令并指定 AMI 的 ID 以获取其报告列表。

```
aws ec2 describe-image-usage-reports --image-ids ami-0abcdef1234567890
```

下面是示例输出。每个报告 ID 都与扫描的资源类型以及报告创建和到期日期一起列出。您可以使用此信息来识别您想要查看其条目的报告。

```
{
  "ImageUsageReports": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ReportId": "amiur-1111111111111111",
```

```
      "ResourceTypes": [
        {
          "ResourceType": "ec2:Instance"
        }
      ],
      "State": "available",
      "CreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ExpirationTime": "2025-10-28T13:27:12.322000+00:00",
      "Tags": []
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ReportId": "amiur-2222222222222222",
      "ResourceTypes": [
        {
          "ResourceType": "ec2:Instance"
        },
        {
          "ResourceType": "ec2:LaunchTemplate"
        }
      ],
      "State": "available",
      "CreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ExpirationTime": "2025-10-30T13:27:12.322000+00:00",
      "Tags": []
    }
  ],
  "NextToken": "opaque"
}
```

查看指定 AMI 的 AMI 使用情况报告的内容

使用 describe-image-usage-report-entries 命令并指定 AMI 的 ID。响应会返回指定 AMI 的所有报告，显示使用过 AMI 的账户及其资源计数。

```
aws ec2 describe-image-usage-report-entries --image-ids ami-0abcdef1234567890
```

下面是示例输出。

```
{
  "ImageUsageReportEntries": [
    {
      "ImageId": "ami-0abcdef1234567890",
```

```
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 15,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-1111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 2,
      "ReportCreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ReportId": "amiur-22222222222222222"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "001100110011",
      "UsageCount": 39,
      "ReportCreationTime": "2025-10-01T13:27:12.322000+00:00",
      "ReportId": "amiur-22222222222222222"
    }
  ],
  "NextToken": "opaque"
}
```

查看指定报告的 AMI 使用情况报告的内容

使用 describe-image-usage-report-entries 命令并指定报告的 ID。响应会返回指定报告的所有条目，显示使用过 AMI 的账户及其资源计数。

```
aws ec2 describe-image-usage-report-entries --report-ids amiur-1111111111111111
```

下面是示例输出。

```
{
  "ImageUsageReportEntries": [
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:Instance",
      "AccountId": "123412341234",
      "UsageCount": 15,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
```

```
      "ReportId": "amiur-11111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:LaunchTemplate",
      "AccountId": "123412341234",
      "UsageCount": 4,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-11111111111111111"
    },
    {
      "ImageId": "ami-0abcdef1234567890",
      "ResourceType": "ec2:LaunchTemplate",
      "AccountId": "001100110011",
      "UsageCount": 2,
      "ReportCreationTime": "2025-09-29T13:27:12.322000+00:00",
      "ReportId": "amiur-11111111111111111"
    }
  ],
  "NextToken": "opaque"
}
```

PowerShell

列出指定 AMI 的所有 AMI 使用情况报告

使用 Get-EC2ImageUsageReport cmdlet 并指定 AMI 的 ID 以获取其报告列表。

```
Get-EC2ImageUsageReport -ImageId ami-0abcdef1234567890
```

下面是示例输出。每个报告 ID 都与扫描的资源类型以及报告创建和到期日期一起列出。您可以使用此信息来识别您想要查看其条目的报告。

```
@{
    ImageUsageReports = @(
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-1111111111111111"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
```

```
            State = "available"
            CreationTime = "2025-09-29T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-28T13:27:12.322000+00:00"
        },
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-2222222222222222"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-09-30T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-29T13:27:12.322000+00:00"
        },
        @{
            ImageId = "ami-0abcdef1234567890"
            ReportId = "amiur-33333333333333333"
            ResourceTypes = @(
                @{
                    ResourceType = "ec2:Instance"
                }
            )
            State = "available"
            CreationTime = "2025-10-01T13:27:12.322000+00:00"
            ExpirationTime = "2025-10-30T13:27:12.322000+00:00"
        }
    )
    NextToken = "opaque"
}
```

查看指定 AMI 的 AMI 使用情况报告的内容

使用 Get-EC2ImageUsageReportEntry cmdlet 并指定 AMI 的 ID。响应会返回指定 AMI 的所有报告，显示使用过 AMI 的账户及其资源计数。

```
Get-EC2ImageUsageReportEntry -ImageId ami-0abcdef1234567890
```

下面是示例输出。

```
ImageUsageReportEntries : {@{
    ImageId = "ami-0abcdef1234567890"
```

```
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 15
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 7
    ReportCreationTime = "2025-09-30T13:27:12.322000+00:00"
    ReportId = "amiur-22222222222222222"
    }...}
NextToken : opaque
```

查看指定报告的 AMI 使用情况报告的内容

使用 Get-EC2ImageUsageReportEntry cmdlet 并指定报告的 ID。响应会返回指定报告的所有条目，显示使用过 AMI 的账户及其资源计数。

```
Get-EC2ImageUsageReportEntry -ReportId amiur-1111111111111111
```

下面是示例输出。

```
ImageUsageReportEntries : {@{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:Instance"
    AccountId = "123412341234"
    UsageCount = 15
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:LaunchTemplate"
    AccountId = "123412341234"
    UsageCount = 4
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }, @{
    ImageId = "ami-0abcdef1234567890"
    ResourceType = "ec2:LaunchTemplate"
    AccountId = "************"
    UsageCount = 2
```

```
    ReportCreationTime = "2025-09-29T13:27:12.322000+00:00"
    ReportId = "amiur-1111111111111111"
    }}
NextToken : opaque
```

删除 AMI 使用情况报告

Amazon EC2 会在报告创建 30 天后自动删除该报告。您可以在此之前手动将其删除。

Console

删除 AMI 使用情况报告

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 选择我的 AMI 使用情况选项卡。
5. 选择要删除的报告旁边的选项按钮，然后选择删除。

Amazon CLI

删除 AMI 使用情况报告

使用 delete-image-usage-report 命令并指定报告的 ID。

```
aws ec2 delete-image-usage-report --report-id amiur-0123456789abcdefg
```

PowerShell

删除 AMI 使用情况报告

使用 Remove-EC2ImageUsageReport cmdlet 并指定报告的 ID。

```
Remove-EC2ImageUsageReport -ReportId amiur-0123456789abcdefg
```

报告配额

以下配额适用于创建 AMI 使用情况报告。限额按 Amazon Web Services 区域应用。

说明	配额
每个 Amazon Web Services 账户正在进行 (pending) 的 AMI 使用情况报告	2000
每个 AMI 正在进行 (pending) 的 AMI 使用情况报告	1

检查上次使用 Amazon EC2 AMI 的时间

Amazon EC2 会自动跟踪 AMI 上次被用于启动实例的日期和时间。如果您的 AMI 已经很长时间没有用于启动实例，请考虑该 AMI 是否适合注销或弃用。

注意事项

- 在使用 AMI 启动实例时，使用情况报告会有 24 小时的延迟。
- 您必须是 AMI 的所有者才能获取上次启动的时间。
- AMI 使用情况数据自 2017 年 4 月起可用。

Console

查看 AMI 的上次启动时间

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 AMI。
3. 从筛选栏中选择 Owned by me（我拥有的）。
4. 选中该 AMI 对应的复选框。
5. 在详细信息选项卡上，找到上次启动时间。

Amazon CLI

通过描述 AMI 来查看上次启动时间

使用 describe-images 命令。如果输出中不存在 `LastLaunchedTime`，则请确认您拥有该 AMI。

```
aws ec2 describe-images \
```

```
    --image-id ami-0abcdef1234567890 \
    --query Images[].LastLaunchedTime \
    --output text
```

下面是示例输出。

```
2025-02-17T20:22:19Z
```

查看 AMI 的上次启动时间属性

使用 describe-image-attribute 命令。您必须是指定 AMI 的所有者。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute lastLaunchedTime \
    --query LastLaunchedTime.Value \
    --output text
```

下面是示例输出。

```
2025-02-17T20:22:19Z
```

PowerShell

通过描述 AMI 来查看上次启动时间

使用 Get-EC2Image cmdlet。如果输出中不存在 `LastLaunchedTime`，则请确认您拥有该 AMI。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).LastLaunchedTime
```

下面是示例输出。

```
2025-02-17T20:22:19Z
```

查看 AMI 的上次启动时间属性

使用 Get-EC2ImageAttribute cmdlet。您必须是指定 AMI 的所有者。

```
(Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
```

```
    -Attribute LastLaunchedTime).LastLaunchedTime
```

下面是示例输出。

```
2025-02-17T20:22:19Z
```

识别引用指定 AMI 的资源

您可以识别引用指定亚马逊机器映像（AMI）的 Amazon 资源，无论 AMI 是公共的还是私有的，也无论它们的所有者是谁。这种可见性可帮助您确保资源使用最新的合规 AMI。

主要优势

检查 AMI 引用可以帮助您：

- 审核您账户中 AMI 的使用情况。
- 检查引用特定 AMI 的位置。
- 通过更新资源以引用最新的 AMI 来保持合规性。

主题

- 支持的资源
- AMI 引用检查的工作原理
- 所需的 IAM 权限
- 检查 AMI 引用的步骤

支持的资源

可在以下位置检查 AMI 引用：

- EC2 实例
- 启动模板
- SSM 参数
- Image Builder 映像配方
- Image Builder 容器配方

AMI 引用检查的工作原理

基础操作

运行 AMI 引用调查时，您可以：

- 指定要检查的 AMI。
- 选择要扫描的资源类型。
- 接收引用指定 AMI 的资源列表。

资源类型选择

在控制台中，选择要扫描的资源类型。

在 CLI 中，使用以下一个或两个 CLI 参数指定要扫描的资源类型：

- `IncludeAllResourceTypes`：扫描所有支持的资源类型。
- `ResourceTypes`：扫描指定的资源类型。

响应范围界定

您可以使用 `ResourceTypes` 参数自定义 `ResourceTypeOptions` 值，从而限定 EC2 实例和启动模板的响应范围。控制台和 `IncludeAllResourceTypes` 参数都使用默认选项值。当 `ResourceTypes` 和 `IncludeAllResourceTypes` 一起使用时，`ResourceTypes` 选项值优先于默认值。

以下是默认值：

资源类型	范围界定选项 (**`OptionName`**)	用途	**`OptionValue`** 和控制台的默认值
EC2 实例	`state-name`	按实例状态筛选	`pending`、`running`、`shutti` `down` 、`terminate` `d` 、`stopping`、`stopped`（所有状态）
启动模板	`version-depth`	指定要检查的启动模板版本数（从最新版本开始）	`10`（最新版本）

所需的 IAM 权限

要使用 DescribeImageReferences API 来识别引用指定 AMI 的资源，您需要以下 IAM 权限来描述资源：

- ec2:DescribeInstances
- ec2:DescribeLaunchTemplates
- ec2:DescribeLaunchTemplateVersions
- ssm:DescribeParameters
- ssm:GetParameters
- imagebuilder:ListImageRecipes
- imagebuilder:ListContainerRecipes
- imagebuilder:GetContainerRecipe

使用 DescribeImageReferences API 的 IAM 策略示例

以下示例策略授予您使用 DescribeImageReferences API 的权限，其中包括描述 EC2 实例、启动模板、Systems Manager 参数、Image Builder 映像配方和 Image Builder 容器配方的权限。

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "ec2:DescribeImageReferences",
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeLaunchTemplateVersions",
    "ssm:DescribeParameters",
    "ssm:GetParameters",
    "imagebuilder:ListImageRecipes",
```

```
    "imagebuilder:ListContainerRecipes",
    "imagebuilder:GetContainerRecipe"
   ],
   "Resource": "*",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "ec2-images.amazonaws.com"
     ]
    }
   }
  }
 ]
}
```

Important

我们强烈建议您使用 Amazon 托管策略 AmazonEC2ImageReferencesAccessPolicy，而不是自行创建策略。创建仅提供所需权限的自定义 IAM 策略需要时间和专业知识，并且随着新资源类型推出，需要更新。

AmazonEC2ImageReferencesAccessPolicy 托管策略：

- 授予使用 DescribeImageReferences API 所需的所有权限（包括描述 EC2 实例、启动模板、Systems Manager 参数以及 Image Builder 容器和映像配方的权限）。
- 在新的资源类型可用时自动支持它们（在使用 IncludeAllResourceTypes 参数时尤其重要）。

您可以将 AmazonEC2ImageReferencesAccessPolicy 策略附加到 IAM 身份（用户、组和角色）。

要查看此策略中包含的权限，请参阅《Amazon 托管式策略参考》中的 AmazonEC2ImageReferencesAccessPolicy。

检查 AMI 引用的步骤

使用以下过程确定哪些 Amazon 资源正在引用指定的 AMI。

Console

识别引用指定 AMI 的资源

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择一个或多个 AMI 来检查引用。
4. 选择操作、AMI 使用情况、查看引用的资源。
5. 在查看引用所选 AMI 的资源页面上：
   a. 对于资源类型，选择一种或多种资源类型。
   b. 选择查看资源。
6. 此时将出现引用所选 AMI 的资源部分。该列表显示引用指定 AMI 的资源。每行提供以下信息：
   - AMI ID – 所引用 AMI 的 ID。
   - 资源类型 – 引用 AMI 的资源的资源类型。
   - 资源 ID – 引用 AMI 的资源的 ID。

Amazon CLI

检查特定资源类型的 AMI 引用

使用带有 `--resource-types` 参数的 describe-image-references cmdlet。以下示例检查 EC2 实例（按实例状态限定范围）、启动模板（限定范围为最新 20 个启动模板版本）以及其他特定资源类型。

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --resource-types \
        'ResourceType=ec2:Instance,ResourceTypeOptions=[{OptionName=state-
name,OptionValues=[running,pending]}]' \
        'ResourceType=ec2:LaunchTemplate,ResourceTypeOptions=[{OptionName=version-
depth,OptionValues=[20]}]' \
        'ResourceType=ssm:Parameter' \
        'ResourceType=imagebuilder:ImageRecipe' \
        'ResourceType=imagebuilder:ContainerRecipe'
```

下面是示例输出。

```
{
    "ImageReferences": [
        {
            "ImageId": "ami-0abcdef1234567890",
            "ResourceType": "ec2:Instance",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
        },
        {
            "ImageId": "ami-1234567890abcdef0",
            "ResourceType": "ec2:LaunchTemplate",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:launch-template/
lt-1234567890abcdef0"
        }
    ]
}
```

检查所有支持的资源类型的 AMI 引用

使用带 `--include-all-resource-types` 参数的 describe-image-references cmdlet。

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --include-all-resource-types
```

检查所有支持的资源类型和特定选项的 AMI 引用

使用带 `--include-all-resource-types` 和 `--resource-types` 参数的 describe-image-references cmdlet。此示例检查所有资源类型，同时将 EC2 实例的响应范围限定为正在运行或待处理的实例。

```
aws ec2 describe-image-references \
    --image-ids ami-0abcdef1234567890 ami-1234567890abcdef0 \
    --include-all-resource-types \
    --resource-types
 'ResourceType=ec2:Instance,ResourceTypeOptions=[{OptionName=state-
name,OptionValues=[running,pending]}]'
```

PowerShell

检查特定资源类型的 AMI 引用

使用带 -ResourceType 参数的 Get-EC2ImageReference cmdlet。以下示例检查 EC2 实例（按实例状态限定范围）、启动模板（限定范围为最新 20 个启动模板版本）以及其他特定资源类型。

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -ResourceType @(
        @{
            ResourceType = 'ec2:Instance'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'state-name'
                    OptionValues = @('running', 'pending')
                }
            )
        },
        @{
            ResourceType = 'ec2:LaunchTemplate'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'version-depth'
                    OptionValues = @('20')
                }
            )
        },
        @{
            ResourceType = 'ssm:Parameter'
        },
        @{
            ResourceType = 'imagebuilder:ImageRecipe'
        },
        @{
            ResourceType = 'imagebuilder:ContainerRecipe'
        }
    )
```

检查所有支持的资源类型的 AMI 引用

使用带 -IncludeAllResourceTypes 参数的 Get-EC2ImageReference cmdlet。

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -IncludeAllResourceTypes
```

检查所有支持的资源类型和特定选项的 AMI 引用

使用带 `-IncludeAllResourceTypes` 和 `-ResourceType` 参数的 Get-EC2ImageReference cmdlet。此示例检查所有资源类型，同时将 EC2 实例的响应范围限定为正在运行或待处理的实例。

```
Get-EC2ImageReference `
    -ImageId 'ami-0abcdef1234567890', 'ami-1234567890abcdef0' `
    -IncludeAllResourceTypes `
    -ResourceType @(
        @{
            ResourceType = 'ec2:Instance'
            ResourceTypeOptions = @(
                @{
                    OptionName = 'state-name'
                    OptionValues = @('running', 'pending')
                }
            )
        }
    )
```

弃用 Amazon EC2 AMI

您可以弃用 AMI，以表明该 AMI 已过期且不应使用。您还可以为 AMI 指定未来弃用日期，表明 AMI 何时过期。例如，您可能会弃用不再主动维护的 AMI，或是已被较新版本取代的 AMI。默认情况下，已弃用的 AMI 不会显示在 AMI 列表中，防止新用户使用过期的 AMI。但是，现有用户和启动服务（如启动模板和 Auto Scaling 组）可以通过指定其 ID 来继续使用已弃用的 AMI。要删除 AMI 以便使用户和服务无法使用，您必须取消注册。

AMI 弃用后：

- 对于 AMI 用户，已弃用的 AMI 不会显示在 DescribeImages API 调用中，除非您指定其 ID 或要求必须显示已弃用的 AMI。AMI 拥有者可继续在 DescribeImages API 调用中看到已弃用的 AMI。
- 对于 AMI 用户，已弃用的 AMI 无法通过 EC2 控制台进行选择。例如，已弃用的 AMI 不会出现在启动实例向导的 AMI 目录中。AMI 拥有者可继续在 EC2 控制台中看到已弃用的 AMI。
- 对于 AMI 用户，如果您知道已弃用的 AMI 的 ID，则可以通过 API、CLI 或开发工具包继续使用已弃用的 AMI 启动实例。
- 启动服务（如启动模板和 Auto Scaling 组）可以继续引用已弃用的 AMI。
- 使用随后被弃用的 AMI 启动的 EC2 实例不会受到影响，可以停止、启动和重新启动。

您可以弃用私有和公用 AMI。

内容

- 成本
- 注意事项
- 弃用 AMI
- 描述已弃用的 AMI
- 取消 AMI 弃用

成本

当您弃用 AMI 时，不会删除该 AMI。AMI 拥有者须继续为 AMI 的快照付费。要停止支付快照费用，AMI 拥有者必须通过取消注册删除 AMI。

注意事项

- 要弃用 AMI，您必须是 AMI 的拥有者。
- 最近未用于启动实例的 AMI 可能需要弃用或注销。有关更多信息，请参阅 the section called “检查上次使用 AMI 的时间”。
- 您可以创建 Amazon Data Lifecycle Manager EBS-backed AMI 策略，以自动执行 EBS-backed AMI 的弃用。有关更多信息，请参阅 Create AMI lifecycle policies。
- 默认情况下，所有公用 AMI 的弃用日期设置为自 AMI 创建日期起的两年。您可以将弃用日期设置为早于两年。要取消弃用日期，或将弃用移至未来某一日期，您必须通过仅将 AMI 与特定 Amazon 账户共享来将其设为私有。

弃用 AMI

您可以在特定日期和时间弃用 AMI。您必须是该 AMI 的所有者。

弃用日期的上限为从当前日期起 10 年，但公有 AMI 除外，其上限为自创建之日起 2 年。不能指定某个过去的日期。

Console

在特定日期弃用 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在左侧导航器中选择 AMI。
3. 从筛选栏中选择 Owned by me（我拥有的）。
4. 选取相应 AMI，然后选择 Actions（操作）、Manage AMI Deprecation（管理 AMI 弃用）。您可以选择多个 AMI，一次性为多个 AMI 设置同一弃用日期。
5. 选择启用复选框，然后输入弃用日期和时间。
6. 选择保存。

Amazon CLI

在特定日期弃用 AMI

使用 enable-image-deprecation 命令。如果指定以秒为单位的值，Amazon EC2 会将秒四舍五入到最近的分钟数。

```
aws ec2 enable-image-deprecation \
    --image-id ami-0abcdef1234567890 \
    --deprecate-at "2025-04-15T13:17:12.000Z"
```

PowerShell

在特定日期弃用 AMI

使用 Enable-EC2ImageDeprecation cmdlet。如果指定以秒为单位的值，Amazon EC2 会将秒四舍五入到最近的分钟数。

```
Enable-EC2ImageDeprecation `
    -ImageId ami-0abcdef1234567890 `
    -DeprecateAt 2025-04-15T13:17:12.000Z
```

描述已弃用的 AMI

您可以查看 AMI 的弃用日期和时间，以及按弃用日期筛选 AMI。

Console

查看 AMI 的弃用日期

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在左侧导航器中选择 AMIs，然后选择相应 AMI。
3. 检查弃用时间字段（如果选中 AMI 旁边的复选框，则它位于详细信息选项卡上）。该字段将显示 AMI 的弃用日期和时间。如果该字段为空，表示该 AMI 未遭弃用。

按弃用日期筛选 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航器中选择 AMI。
3. 从筛选栏中选择 Owned by me（我拥有的）或 Private images（私有映像）（私有映像包括与您共享的 AMI 以及您拥有的 AMI）。
4. 在搜索栏中输入 **`Deprecation time`**（输入字母时会出现 Deprecation time（弃用时间）筛选器），然后选择运算符以及日期和时间。

Amazon CLI

当您描述所有 AMI 时，结果会因您是 AMI 用户还是 AMI 拥有者而异。

- AMI 用户 – 默认情况下，当您描述所有 AMI 时，结果会排除与您共享但不属于您的已弃用 AMI。要在结果中包括已弃用的 AMI，请指定 `--include-deprecated` 选项。
- AMI 所有者 – 当您描述所有 AMI 时，结果将包含您拥有的所有 AMI，包括已弃用的 AMI。您不能使用 `--no-include-deprecated` 选项来排除自己拥有的已弃用 AMI。

在描述某个账户的所有 AMI 时包含已弃用 AMI

使用以下 describe-images 命令。

```
aws ec2 describe-images
    --owners 123456789012 \
    --include-deprecated
```

描述您账户中的已弃用 AMI

使用以下 describe-images 命令。

```
aws ec2 describe-images \
    --owners self \
    --query "Images[?DeprecationTime!=null].ImageId" \
```

```
    --output text
```

下面是示例输出。

```
ami-0abcdef1234567890
```

描述 AMI 的弃用日期

使用以下 describe-images 命令。如果输出中不存在 DeprecationTime，则表示该 AMI 未被弃用，也未设置为在某个未来日期弃用。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].DeprecationTime \
    --output text
```

下面是示例输出。

```
2025-05-01T00:00:00.000Z
```

PowerShell

列出您账户中的已弃用 AMI

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -Owner self | Where-Object {$_.DeprecationTime -ne $null}).ImageId
```

下面是示例输出。

```
ami-0abcdef1234567890
```

描述 AMI 的弃用日期

使用 Get-EC2Image cmdlet。如果输出中不存在 DeprecationTime，则表示该 AMI 未被弃用，也未设置为在某个未来日期弃用。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).DeprecationTime
```

下面是示例输出。

```
2025-05-01T00:00:00.000Z
```

取消 AMI 弃用

您可以取消弃用某个 AMI，这将移除弃用日期和时间。您必须是 AMI 拥有者才能执行此过程。

Console

取消弃用 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航器中选择 AMI。
3. 从筛选栏中选择 Owned by me（我拥有的）。
4. 选取相应 AMI，然后选择 Actions（操作）、Manage AMI Deprecation（管理 AMI 弃用）。您可以选择多个 AMI，一次性取消对多个 AMI 的弃用。
5. 清除启用复选框，然后选择保存。

Amazon CLI

取消弃用 AMI

使用以下 disable-image-deprecation 命令。

```
aws ec2 disable-image-deprecation --image-id ami-0abcdef1234567890
```

PowerShell

取消弃用 AMI

使用 Disable-EC2ImageDeprecation cmdlet。

```
Disable-EC2ImageDeprecation -ImageId ami-0abcdef1234567890
```

禁用 Amazon EC2 AMI

您可以禁用 AMI，以防止系统将其用于实例启动。您无法从已禁用的 AMI 启动新实例。您可以重新启用已禁用的 AMI，使其可以再次用于实例启动。

您可以禁用私有和公用 AMI。

禁用由 EBS 支持的 AMI 后，如果该 AMI 的使用频率很少，但需要长期留存，则可以将其关联的快照归档以降低其存储成本。有关更多信息，请参阅《Amazon EBS 用户指南》中的归档 Amazon EBS 快照。

内容

- AMI 禁用的工作原理
- 成本
- 先决条件
- 所需的 IAM 权限
- 禁用 AMI
- 描述已禁用的 AMI
- 重新启用已禁用的 AMI

AMI 禁用的工作原理

> **Warning**
>
> 禁用 AMI 会移除其所有启动权限。

当 AMI 被禁用时：

- AMI 的状态会更改为 `disabled`。
- 无法共享已禁用的 AMI。如果 AMI 为公用性质或以前共享过，则会将其设为私有。如果与 Amazon Web Services 账户、组织或组织单位共享了 AMI，则它们将无法访问已禁用的 AMI。
- 默认情况下，已禁用的 AMI 不会出现在 DescribeImages API 调用中。
- 禁用的 AMI 不会在我拥有的控制台筛选条件下显示。要查找已禁用的 AMI，请使用已禁用的映像控制台筛选条件。
- 已禁用的 AMI 不可在 EC2 控制台中选择用于实例启动。例如，已禁用的 AMI 不会出现在启动实例向导的 AMI 目录中，创建启动模板时也不会显示。
- 启动服务（如启动模板和自动扩缩组）可以继续引用已禁用的 AMI。从已禁用的 AMI 发起的后续实例启动将失败，因此我们建议更新启动模板和自动扩缩组，使其仅引用可用的 AMI。
- 之前使用随后被禁用的 AMI 启动的 EC2 实例不会受到影响，可以停止、启动和重新启动。

- 无法删除与已禁用的 AMI 关联的快照。尝试删除关联的快照会导致 `snapshot is currently in use` 错误。

重新启用 AMI 时：

- AMI 的状态将更改为 `available`，可用于启动实例。
- AMI 可以共享。
- 在 AMI 被禁用后失去对其的访问权限的 Amazon Web Services 账户、组织和组织单位不会自动重新获得访问权限，但可以再次与它们共享 AMI。

成本

当您禁用 AMI 时，不会删除该 AMI。如果 AMI 是 EBS-backed AMI，则您需要继续为 AMI 的 EBS 快照付费。如果您想保留此类 AMI，则可以通过归档快照来降低存储成本。有关更多信息，请参阅《Amazon EBS 用户指南》中的归档 Amazon EBS 快照。如果您不想保留此类 AMI 及其快照，则必须取消注册相关 AMI 并删除相应快照。有关更多信息，请参阅 取消注册 AMI。

先决条件

要禁用或重新启用 AMI，您必须是 AMI 的所有者。

所需的 IAM 权限

要禁用和重新启用 AMI，您必须拥有以下 IAM 权限：

- `ec2:DisableImage`
- `ec2:EnableImage`

禁用 AMI

您可以使用 EC2 控制台或 Amazon Command Line Interface（Amazon CLI）禁用 AMI。您必须是 AMI 拥有者才能执行此过程。

Console

要禁用 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在左侧导航窗格中，选择 AMI。
3. 从筛选栏中选择 Owned by me（我拥有的）。
4. 选择 AMI，然后依次选择操作和禁用 AMI。您可以同时选择多个 AMI，一并将其禁用。
5. 在禁用 AMI窗口中，选择禁用 AMI。

Amazon CLI

要禁用 AMI

使用以下 disable-image 命令。

```
aws ec2 disable-image --image-id ami-0abcdef1234567890
```

PowerShell

禁用 AMI

使用 Disable-EC2Image cmdlet。

```
Disable-EC2Image -ImageId ami-0abcdef1234567890
```

描述已禁用的 AMI

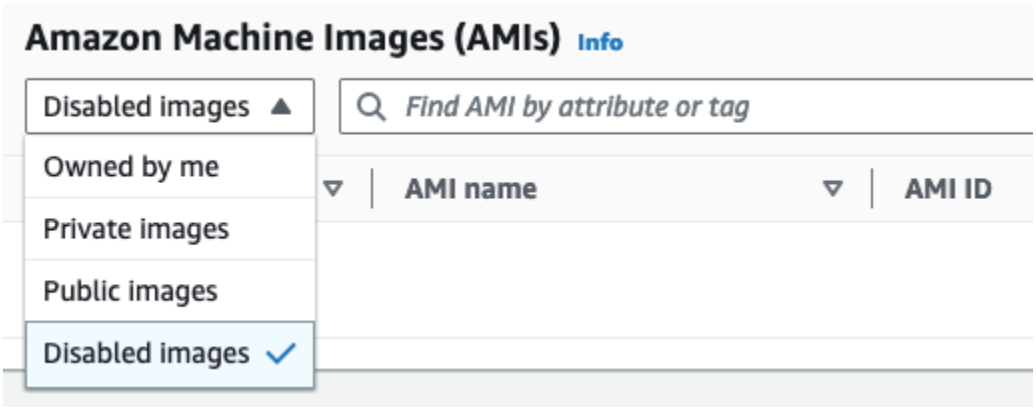
您可以在 EC2 控制台中或使用 Amazon CLI 查看已禁用的 AMI。

您必须是 AMI 所有者才能查看已禁用的 AMI。由于已禁用的 AMI 被设为私有，因此如果您不是所有者，则无法对其进行查看。

Console

要查看已禁用的 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 AMI。
3. 从筛选栏中选择已禁用的映像。



Amazon CLI

默认情况下，当您描述所有 AMI 时，已禁用的 AMI 不会包含在结果中。要在结果中包含已禁用的 AMI，请指定 `--include-disabled` 选项。如果 AMI 已被禁用，则该 AMI 的 `State` 字段将为 `disabled`。

在描述某个账户的所有 AMI 时包含已禁用 AMI

使用以下 describe-images 命令。

```
aws ec2 describe-images \
    --owners 123456789012 \
    --include-disabled
```

列出您账户中的已禁用 AMI

使用以下 describe-images 命令。

```
aws ec2 describe-images \
    --owners self \
    --include-disabled \
    --filters Name=state,Values=disabled \
    --query Images[].ImageId \
    --output text
```

下面是示例输出。

```
ami-0abcdef1234567890
```

描述 AMI 的状态

使用以下 describe-images 命令。如果输出中不存在 DeprecationTime，则表示该 AMI 未被弃用，也未设置为在某个未来日期弃用。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[].State \
    --output text
```

下面是示例输出。

```
disabled
```

PowerShell

默认情况下，当您描述所有 AMI 时，已禁用的 AMI 不会包含在结果中。要在结果中包含已禁用的 AMI，请指定 `-IncludeDisabled` 参数。如果 AMI 已被禁用，则该 AMI 的 `State` 字段将为 `disabled`。

列出您账户中的已禁用 AMI

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image `
    -Owner self `
    -IncludeDisabled $true | Where-Object {$_.State -eq "disabled"}).ImageId
```

下面是示例输出。

```
ami-0abcdef1234567890
```

描述 AMI 的状态

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).State.Value
```

下面是示例输出。

```
disabled
```

重新启用已禁用的 AMI

您可以重新启用已禁用的 AMI。您必须是 AMI 拥有者才能执行此过程。

Console

要重新启用已禁用的 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 AMI。
3. 从筛选栏中选择已禁用的映像。
4. 选择 AMI，然后依次选择操作、启用 AMI。您可以同时选择多个 AMI，一并将其重新启用。
5. 在启用 AMI窗口中，选择启用。

Amazon CLI

要重新启用已禁用的 AMI

使用以下 enable-image 命令。

```
aws ec2 enable-image --image-id ami-0abcdef1234567890
```

PowerShell

要重新启用已禁用的 AMI

使用 Enable-EC2Image cmdlet。

```
Enable-EC2Image -ImageId ami-0abcdef1234567890
```

取消注册 Amazon EC2 AMI

取消注册 AMI 时，Amazon EC2 会将其永久删除。取消注册 AMI 之后，便无法将其用于启动新实例。使用完 AMI 之后，可以考虑将其取消注册。

为防止意外或恶意取消注册 AMI，您可以开启取消注册保护。如果您意外取消注册了 EBS-backed AMI，则可以使用回收站将其还原，但仅限在允许的期限内可以还原，并且该期限届满后 AMI 将被永久删除。

注销 AMI 时，您可以选择同时删除其关联的快照。如果一个快照关联到多个 AMI，则即使指定删除，系统也不会将该快照删除，但该 AMI 仍会被注销。任何未删除的快照都将继续产生存储费用。

取消注册 AMI 不会影响从该 AMI 启动的任何实例。您可以继续使用这些实例。默认情况下，注销 AMI 也不会影响在 AMI 创建过程中创建的任何快照。您需要继续按这些实例和快照的存储成本支付使用费。因此，为避免产生不必要的成本，我们建议您终止任何不需要的实例并删除任何不需要的快照。您可以在注销过程中自动删除快照，也可以在注销后手动删除快照。有关更多信息，请参阅 避免因未使用的资源产生的成本。

对于从随后取消注册的 AMI 启动的实例，您仍然可以使用 `describe-instance-image-metadata` Amazon CLI 命令查看有关 AMI 的一些高级信息。有关更多信息，请参阅 describe-instance-image-metadata。

内容

- 注意事项
- 取消注册 AMI
- 避免因未使用的资源产生的成本
- Amazon EC2 AMI 取消注册保护

注意事项

- 您无法取消注册不属于您账户所有的 AMI。
- 您无法使用 Amazon EC2 注销由 Amazon Backup 服务管理的 AMI。而是应该使用 Amazon Backup 删除备份文件库中相应的恢复点。有关更多信息，请参阅 Amazon Backup 开发人员指南中的删除备份。

取消注册 AMI

您可以注销 EBS-backed AMI 和 Amazon S3 支持的 AMI。对于 EBS-backed AMI，您可以选择同时删除关联的快照。但如果一个快照还关联了其他 AMI，则即使指定删除，系统也不会将其删除。

Console

取消注册 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。

3. 在筛选条件栏中，选择我拥有的可列出可用的 AMI，选择已禁用的映像可列出已禁用的 AMI。
4. 选择要取消注册的 AMI。
5. 选择 Actions（操作）、Deregister AMI（取消注册 AMI）。
6. （可选）要在注销过程中删除关联的快照，请选中删除关联的快照复选框。

> **Note**
>
> 如果一个快照还关联了其他 AMI，则即使选中了该复选框，系统也不会将其删除。

7. 选择注销 AMI。

   控制台可能需要几分钟才能从列表中删除该 AMI。选择 Refresh 以刷新状态。

Amazon CLI

取消注册 AMI

使用以下 deregister-image 命令。

```
aws ec2 deregister-image --image-id ami-0abcdef1234567890
```

注销 AMI 并删除其关联的快照

使用以下 deregister-image 命令并指定 `--delete-associated-snapshots` 参数。请注意，如果一个快照还关联了其他 AMI，则即使指定了此参数，系统也不会将其删除。

```
aws ec2 deregister-image \
    --image-id ami-0abcdef1234567890 \
    --delete-associated-snapshots
```

PowerShell

取消注册 AMI

使用 Unregister-EC2Image cmdlet。

```
Unregister-EC2Image -ImageId ami-0abcdef1234567890
```

注销 AMI 并删除其关联的快照

使用 Unregister-EC2Image cmdlet 并指定 -DeleteAssociatedSnapshots 参数。请注意，如果一个快照还关联了其他 AMI，则即使指定了此参数，系统也不会将其删除。

```
Unregister-EC2Image `
    -ImageId ami-0abcdef1234567890 `
    -DeleteAssociatedSnapshots
```

避免因未使用的资源产生的成本

默认情况下，注销 AMI 不会删除与该 AMI 关联的所有资源。这些资源包括 EBS-backed AMI 的快照以及在 Amazon S3 中 Amazon S3 支持的 AMI 的文件。取消注册某个 AMI 时，您也不会终止或停止从该 AMI 启动的任何实例。

存储快照和文件将继续产生费用，任何正在运行的实例都将产生费用。

为避免产生此类不必要的费用，我们建议您删除任何不再需要的资源。

EBS 支持的 AMI

- 在注销 AMI 的同时删除关联的快照。有关更多信息，请参阅 取消注册 AMI。
- 如果注销 AMI 但未删除其关联的快照，则可以手动删除快照。创建 AMI 期间创建的实例根卷快照采用以下描述格式：

```
Created by CreateImage(i-1234567890abcdef0) for ami-0abcdef1234567890
```

- 如果您不再需要利用该 AMI 启动的实例，则可以将其停止或终止。要列出实例，请按 AMI 的 ID 进行筛选。

Amazon S3 支持的 AMI

- 使用 ec2-delete-bundle（AMI 工具）命令删除 Amazon S3 中的捆绑包。
- 如果删除捆绑包后 Amazon S3 存储桶为空，并且该存储桶不再有进一步的用途，您可以删除存储桶。
- 如果您不再需要利用该 AMI 启动的实例，则可以将其终止。要列出实例，请按 AMI 的 ID 进行筛选。

Amazon EC2 AMI 取消注册保护

您可以在 AMI 上开启取消注册保护，以防止意外或恶意删除。开启取消注册保护后，任何用户都无法取消注册该 AMI，无论其 IAM 权限如何。如果要取消注册该 AMI，则必须首先关闭其上的取消注册保护。

在 AMI 上开启取消注册保护时，您可以选择包含 24 小时的冷却时间。此冷却时间是取消注册保护在您关闭后仍然有效的时间。在此冷却时间内，该 AMI 无法取消注册。冷却时间结束后，可以取消注册该 AMI。

所有现有和新的 AMI 都会默认关闭取消注册保护。

开启取消注册保护

使用以下过程开启取消注册保护。

Console

开启注销保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 在筛选条件栏中，选择我拥有的可列出可用的 AMI，选择已禁用的映像可列出已禁用的 AMI。
4. 选择要开启取消注册保护的 AMI，然后选择操作、管理 AMI 注销保护。
5. 在管理 AMI 取消注册保护对话框中，您可以在带冷却时间或不带冷却时间的情况下开启取消注册保护。请选择以下选项之一：
   - 启用并包含 24 小时的冷却时间 – 有冷却时间，取消注册保护关闭后 24 小时内无法取消注册 AMI。
   - 启用但无冷却时间 – 无冷却时间，取消注册保护关闭后可以立即取消注册 AMI。
6. 选择保存。

Amazon CLI

开启注销保护

使用 enable-image-deregistration-protection 命令。要启用可选的冷却时间，请添加 `--with-cooldown` 选项。

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0abcdef1234567890 \
    --with-cooldown
```

PowerShell

开启注销保护

使用 Enable-EC2ImageDeregistrationProtection cmdlet。要启用可选的冷却时间，请将 -WithCooldown 参数设置为 true。

```
Enable-EC2ImageDeregistrationProtection `
    -ImageId ami-0abcdef1234567890 `
    -WithCooldown $true
```

关闭取消注册保护

使用以下过程关闭取消注册保护。

如果您在为 AMI 开启取消注册保护时选择了包含 24 小时冷却时间的选项，则在您关闭取消注册保护后，您将无法立即取消注册 AMI。此冷却时间为 24 小时，是取消注册保护在您关闭后仍然有效的时间。在此冷却时间内，该 AMI 无法取消注册。冷却时间结束后，可以取消注册该 AMI。

Console

关闭注销保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 在筛选条件栏中，选择我拥有的可列出可用的 AMI，选择已禁用的映像可列出已禁用的 AMI。
4. 选择要关闭取消注册保护的 AMI，然后选择操作、管理 AMI 取消注册保护。
5. 在管理 AMI 取消注册保护对话框中，选择禁用。
6. 选择保存。

Amazon CLI

关闭注销保护

使用 disable-image-deregistration-protection 命令。

```
aws ec2 disable-image-deregistration-protection --image-id ami-0abcdef1234567890
```

PowerShell

关闭注销保护

使用 Disable-EC2ImageDeregistrationProtection cmdlet。

```
Disable-EC2ImageDeregistrationProtection -ImageId ami-0abcdef1234567890
```

使用 Amazon EC2 启动模式的实例启动行为

电脑启动时，它运行的第一个软件负责初始化平台并为操作系统执行特定于平台的操作提供界面。

在 Amazon EC2 中，支持启动模式软件的两种变体包括：统一可扩展固件接口（UEFI）和传统 BIOS。

AMI 上可能的启动模式参数

AMI 可以具有以下任一启动模式参数值：`uefi`、`legacy-bios` 或 `uefi-preferred`。AMI 启动模式参数是可选项。对于不带启动模式参数的 AMI，从这些 AMI 启动的实例将使用实例类型的默认启动模式值。

AMI 启动模式参数的用途

AMI 启动模式参数向 Amazon EC2 发出信号，指明启动实例时要使用哪种启动模式。当启动模式参数设置为 `uefi` 时，EC2 将尝试在 UEFI 上启动实例。如果操作系统未配置为支持 UEFI，则实例启动可能失败。

UEFI Preferred 启动模式参数

您可以使用 `uefi-preferred` 启动模式参数创建同时支持 UEFI 和传统 BIOS 的 AMI。当启动模式参数设置为 `uefi-preferred` 并且实例类型支持 UEFI 时，实例会在 UEFI 上启动。如果实例类型不支持 UEFI，实例会在传统 BIOS 上启动。

> **Warning**
>
> UEFI 安全启动等部分功能仅适用于在 UEFI 上启动的实例。将 `uefi-preferred` AMI 启动模式参数与不支持 UEFI 的实例类型结合使用时，实例将以传统 BIOS 启动，并禁用 UEFI 相关功能。如果您依赖于 UEFI 相关功能的可用性，请将 AMI 启动模式参数设置为 `uefi`。

实例类型的默认启动模式

- Graviton 实例类型：UEFI
- Intel 和 AMD 实例类型：传统 BIOS

区域支持

Wavelength 区中不支持 UEFI 启动。

启动模式主题

- 在 UEFI 启动模式下启动 EC2 实例的要求
- 确定 Amazon EC2 AMI 的启动模式参数
- 确定 EC2 实例类型支持的启动模式
- 确定 EC2 实例的启动模式
- 确定 EC2 实例操作系统的启动模式
- 设置 Amazon EC2 AMI 的启动模式
- Amazon EC2 实例的 UEFI 变量
- 适用于 Amazon EC2 实例的 UEFI 安全启动

在 UEFI 启动模式下启动 EC2 实例的要求

实例的启动模式由 AMI 的配置、其中包含的操作系统和实例类型决定。要在 UEFI 启动模式下启动实例，必须满足以下要求。

AMI

AMI 必须按如下方式为 UEFI 配置：

- 操作系统 – AMI 中包含的操作系统必须配置为使用 UEFI；否则，实例启动将失败。有关更多信息，请参阅 确定 EC2 实例操作系统的启动模式。
- AMI 启动模式参数 – AMI 的启动模式参数必须设置为 `uefi` 或 `uefi-preferred`。有关更多信息，请参阅 确定 Amazon EC2 AMI 的启动模式参数。

Linux – 以下 Linux AMI 支持 UEFI：

- Amazon Linux 2023
- Amazon Linux 2（仅适用于 Graviton 实例类型）

至于其他 Linux AMI，则必须配置 AMI。

Windows：以下 Windows AMI 支持 UEFI：

- Windows_Server-2025-*（带有 BIOS- 名称前缀的 AMI 除外）
- TPM-Windows_Server-2022-English-Full-Base
- TPM-Windows_Server-2022-English-Core-Base
- TPM-Windows_Server-2019-English-Full-Base
- TPM-Windows_Server-2019-English-Core-Base
- TPM-Windows_Server-2016-English-Full-Base
- TPM-Windows_Server-2016-English-Core-Base

实例类型

所有基于 Amazon Nitro System 构建的实例都支持 UEFI 和传统 BIOS，下列实例除外：裸机实例、DL1、G4ad、P4、u-3tb1、u-6tb1、u-9tb1、u-12tb1、u-18tb1、u-24tb1 和 VT1。有关更多信息，请参阅 the section called “实例类型启动模式”。

下表显示，实例的启动模式（由生成的实例启动模式列表示）由 AMI 的启动模式参数（第 1 列）、AMI 中包含的操作系统的启动模式配置（第 2 列）和实例类型的启动模式支持（第 3 列）共同决定。

AMI 启动模式参数	操作系统启动模式配置	实例类型启动模式支持	生成的实例启动模式
UEFI	UEFI	UEFI	UEFI
传统 BIOS	传统 BIOS	传统 BIOS	传统 BIOS
UEFI Preferred	UEFI	UEFI	UEFI
UEFI Preferred	UEFI	UEFI 和传统 BIOS	UEFI
UEFI Preferred	传统 BIOS	传统 BIOS	传统 BIOS
UEFI Preferred	传统 BIOS	UEFI 和传统 BIOS	传统 BIOS
未指定启动模式 – ARM	UEFI	UEFI	UEFI

AMI 启动模式参数	操作系统启动模式配置	实例类型启动模式支持	生成的实例启动模式
未指定启动模式 – x86	传统 BIOS	UEFI 和传统 BIOS	传统 BIOS

确定 Amazon EC2 AMI 的启动模式参数

AMI 启动模式参数是可选项。AMI 可以具有以下任一启动模式参数值：`uefi`、`legacy-bios` 或 `uefi-preferred`。

某些 AMI 没有启动模式参数。如果 AMI 没有启动模式参数，从 AMI 启动的实例将使用实例类型的默认值，在 Graviton 上使用 `uefi`，并在 Intel 和 AMD 实例类型上使用 `legacy-bios`。

Console

确定 AMI 的启动模式参数

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI，然后选择 AMI。
3. 检查启动模式字段。
   - 值 uefi 表示 AMI 支持 UEFI。
   - 值 uefi-preferred 表示 AMI 同时支持 UEFI 和传统 BIOS。
   - 如果没有值，从 AMI 启动的实例将使用实例类型的默认值。

在启动实例时确定 AMI 的启动模式参数

使用启动实例向导启动实例时，请在选择 AMI 的步骤中检查启动模式字段。有关更多信息，请参阅应用程序和操作系统映像（亚马逊机器映像）。

Amazon CLI

确定 AMI 的启动模式参数

使用 describe-images 命令确定 AMI 的启动模式。

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-id ami-0abcdef1234567890 \
```

```
    --query Images[].BootMode \
    --output text
```

下面是示例输出。

```
uefi
```

输出中的值 `uefi` 表示 AMI 支持 UEFI。值 `uefi-preferred` 表示 AMI 同时支持 UEFI 和传统 BIOS。如果没有值，从 AMI 启动的实例将使用实例类型的默认值。

PowerShell

确定 AMI 的启动模式参数

使用 Get-EC2Image cmdlet 确定 AMI 的启动模式。

```
Get-EC2Image -Region us-east-1 `
    -ImageId ami-0abcdef1234567890 | Format-List Name, BootMode, TpmSupport
```

下面是示例输出。

```
Name       : TPM-Windows_Server-2016-English-Full-Base-2023.05.10
BootMode   : uefi
TpmSupport : v2.0
```

输出中的值 BootMode 表示 AMI 的启动模式。值 `uefi` 表示 AMI 支持 UEFI。值 `uefi-preferred` 表示 AMI 同时支持 UEFI 和传统 BIOS。如果没有值，从 AMI 启动的实例将使用实例类型的默认值。

确定 EC2 实例类型支持的启动模式

您可以确定实例类型支持的启动模式。

Amazon EC2 控制台不会显示实例类型支持的启动模式。

Amazon CLI

使用 describe-instance-types 命令确定实例类型支持的启动模式。`--query` 参数筛选输出以仅返回支持的启动模式。

以下示例显示，指定的实例类型同时支持 UEFI 和传统 BIOS 启动模式。

```
aws ec2 describe-instance-types \
    --instance-types m5.2xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

下面是示例输出。

```
[
    [
        "legacy-bios",
        "uefi"
    ]
]
```

以下示例显示 t2.xlarge 仅支持传统 BIOS。

```
aws ec2 describe-instance-types \
    --instance-types t2.xlarge \
    --query "InstanceTypes[*].SupportedBootModes"
```

下面是示例输出。

```
[
    [
        "legacy-bios"
    ]
]
```

PowerShell

使用 Get-EC2InstanceType cmdlet 确定实例类型支持的启动模式。

以下示例显示 m5.2xlarge 支持 UEFI 和传统 BIOS 启动模式。

```
Get-EC2InstanceType -InstanceType m5.2xlarge | Format-List InstanceType,
 SupportedBootModes
```

下面是示例输出。

```
InstanceType       : m5.2xlarge
```

```
SupportedBootModes : {legacy-bios, uefi}
```

以下示例显示 t2.xlarge 仅支持传统 BIOS。

```
Get-EC2InstanceType -InstanceType t2.xlarge | Format-List InstanceType,
 SupportedBootModes
```

下面是示例输出。

```
InstanceType       : t2.xlarge
SupportedBootModes : {legacy-bios}
```

确定支持 UEFI 的实例类型

您可以确定支持 UEFI 的实例类型。Amazon EC2 控制台不会显示实例类型是否支持 UEFI。

Amazon CLI

可用的实例类型因 Amazon Web Services 区域 而异。要查看某个区域中支持 UEFI 的可用实例类型，请使用 describe-instance-types 命令。包括 --filters 参数以将结果范围限定为支持 UEFI 的实例类型，并包括 --query 参数以将输出范围限定为 InstanceType 的值。

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

可用的实例类型因 Amazon Web Services 区域 而异。要查看某个区域中支持 UEFI 的可用实例类型，请使用 Get-EC2InstanceType cmdlet。

```
Get-EC2InstanceType -Filter @{Name="supported-boot-mode"; Values="uefi"} |
 Sort-Object @{E={$_.ProcessorInfo.SupportedArchitectures}}, InstanceType |
 Format-Table InstanceType, SupportedBootModes, `
      @{Name="SupportedArchitectures"; E={$_.ProcessorInfo.SupportedArchitectures}}
```

确定支持 UEFI 安全启动并保留非易失性变量的实例类型

裸机实例不支持 UEFI 安全启动和非易失性变量，因此这些示例将它们从输出中排除。

Amazon CLI

使用 describe-instance-types 命令，并从输出中排除裸机实例。

```
aws ec2 describe-instance-types \
    --filters Name=supported-boot-mode,Values=uefi Name=bare-metal,Values=false \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

使用 Get-EC2InstanceType cmdlet，并从输出中排除裸机实例。

```
Get-EC2InstanceType -Filter `
      @{Name="supported-boot-mode"; Values="uefi"}, `
      @{Name="bare-metal"; Values="false"} |
 Sort-Object @{E={$_.ProcessorInfo.SupportedArchitectures}}, InstanceType |
 Format-Table InstanceType, SupportedBootModes, `
      @{Name="SupportedArchitectures"; E={$_.ProcessorInfo.SupportedArchitectures}}
```

确定 EC2 实例的启动模式

实例的启动模式会在 Amazon EC2 控制台中的启动模式字段中，通过 Amazon CLI 中的 `currentInstanceBootMode` 参数显示。

启动实例时，其启动模式参数的值由启动该实例所用的 AMI 的启动模式参数的值决定，如下所示：

- 启动模式参数为 `uefi` 的 AMI 会创建 `currentInstanceBootMode` 参数为 `uefi` 的实例。
- 启动模式参数为 `legacy-bios` 的 AMI 会创建 `currentInstanceBootMode` 参数为 `legacy-bios` 的实例。
- 如果实例类型支持 UEFI，则启动模式参数为 `uefi-preferred` 的 AMI 会创建 `currentInstanceBootMode` 参数为 `uefi` 的实例；否则将创建 `currentInstanceBootMode` 参数为 `legacy-bios` 的实例。
- 没有启动模式参数值的 AMI 会创建参数值为 `currentInstanceBootMode` 的实例，该值取决于 AMI 架构是 ARM 还是 x86，以及实例类型支持哪种启动模式。Graviton 实例类型上的默认启动模式为 `uefi`，而 Intel 和 AMD 实例类型上的默认启动模式为 `legacy-bios`。

Console

确定实例的启动模式

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择您的实例。
3. 在详细信息选项卡上，检查启动模式字段。

Amazon CLI

确定实例的启动模式

使用 describe-instances 命令确定实例的启动模式。您还可以确定用于创建实例的 AMI 的启动模式。

```
aws ec2 describe-instances \
    --region us-east-1 \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BootMode \
    --output text
```

下面是示例输出。

```
uefi
```

PowerShell

确定实例的启动模式

使用 Get-EC2Image cmdlet 确定实例的启动模式。您还可以确定用于创建实例的 AMI 的启动模式。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | Format-List BootMode,
 CurrentInstanceBootMode, InstanceType, ImageId
```

下面是示例输出。

```
BootMode                : uefi
CurrentInstanceBootMode : uefi
InstanceType            : c5a.large
```

```
ImageId                 : ami-0abcdef1234567890
```

确定 EC2 实例操作系统的启动模式

AMI 的启动模式指导 Amazon EC2 使用哪种启动模式来启动实例。要查看实例的操作系统是否配置为 UEFI，需要通过 SSH（Linux 实例）或 RDP（Windows 实例）连接到实例。

请使用适用于您实例操作系统的说明。

Linux

要确定实例操作系统的启动模式

1. 使用 SSH 连接到 Linux 实例。
2. 要查看操作系统的启动模式，请尝试以下任一操作：

   - 运行如下命令。

     ```
     [ec2-user ~]$ sudo /usr/sbin/efibootmgr
     ```

     在 UEFI 启动模式下启动的实例的预期输出

     ```
     BootCurrent: 0001
     Timeout: 0 seconds
     BootOrder: 0000,0001
     Boot0000* UiApp
     Boot0001* UEFI Amazon Elastic Block Store vol-xyz
     ```

   - 运行以下命令以验证 /sys/firmware/efi 目录是否存在。仅当实例使用 UEFI 启动时，此目录才会存在。如果此目录不存在，该命令将返回 Legacy BIOS Boot Detected。

     ```
     [ec2-user ~]$ [ -d /sys/firmware/efi ] && echo "UEFI Boot Detected" || echo
      "Legacy BIOS Boot Detected"
     ```

     在 UEFI 启动模式下启动的实例的预期输出

     ```
     UEFI Boot Detected
     ```

     在传统 BIOS 启动模式下启动的实例的预期输出

```
Legacy BIOS Boot Detected
```

- 运行以下命令以验证 EFI 是否出现在 dmesg 输出中。

```
[ec2-user ~]$ dmesg | grep -i "EFI"
```

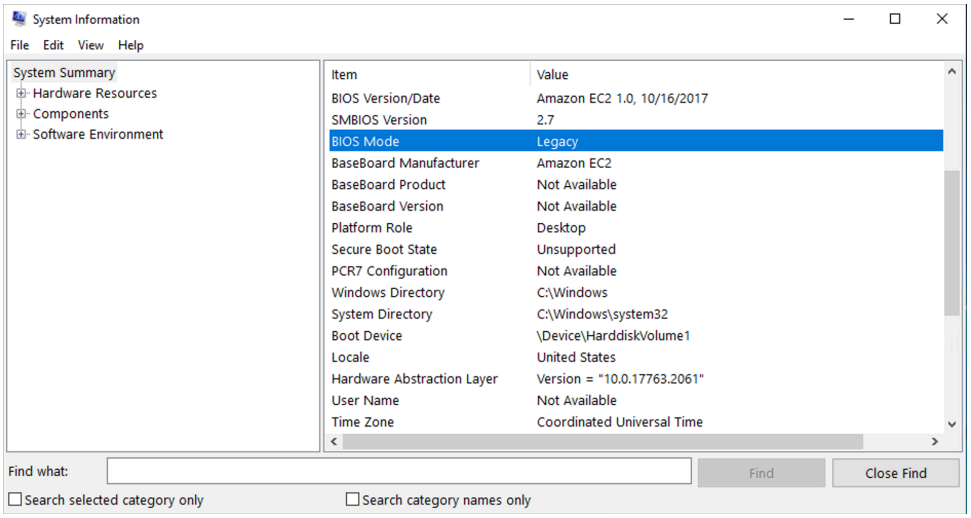
在 UEFI 启动模式下启动的实例的预期输出

```
[    0.000000] efi: Getting EFI parameters from FDT:
[    0.000000] efi: EFI v2.70 by EDK II
```

Windows

要确定实例操作系统的启动模式

1. 使用 RDP 连接到 Windows 实例。
2. 转到系统信息并检查 BIOS 模式行。



设置 Amazon EC2 AMI 的启动模式

默认情况下，AMI 会继承用于创建 AMI 的 EC2 实例的启动模式。例如，如果从传统 BIOS 上运行的 EC2 实例创建 AMI，则新 AMI 的启动模式为 `legacy-bios`。如果从启动模式为 `uefi-preferred` 的 EC2 实例创建 AMI，则新 AMI 的启动模式为 `uefi-preferred`。

注册 AMI 时，可以将 AMI 的启动模式设置为 `uefi`、`legacy-bios` 或 `uefi-preferred`。

当 AMI 启动模式设置为 `uefi-preferred` 时，实例将按如下方式启动：

- 对于同时支持 UEFI 和传统 BIOS 的实例类型（例如 `m5.large`），实例通过 UEFI 启动。
- 对于仅支持传统 BIOS 的实例类型（例如 `m4.large`），实例通过传统 BIOS 启动。

如果将 AMI 启动模式设置为 `uefi-preferred`，则操作系统必须支持同时启动 UEFI 和传统 BIOS 的功能。

要将现有的基于传统 BIOS 的实例转换为 UEFI，或将现有的基于 UEFI 的实例转换为传统 BIOS，必须首先修改实例的卷和操作系统以支持选定的启动模式。然后，创建该卷的快照。首先，从快照创建 AMI。

注意事项

- 设置 AMI 启动模式参数不会自动将操作系统配置为指定的启动模式。您必须首先对实例的卷和操作系统进行适当的修改，以支持通过选定的启动模式启动。否则，生成的 AMI 将不可用。例如，如果要将基于传统 BIOS 的 Windows 实例转换为 UEFI，则可以使用 Microsoft 的 MBR2GPT 工具将系统磁盘从 MBR 转换为 GPT。所需的修改是特定于操作系统的。有关更多信息，请参阅操作系统的手册。
- UEFI 安全启动等部分功能仅适用于在 UEFI 上启动的实例。将 `uefi-preferred` AMI 启动模式参数与不支持 UEFI 的实例类型结合使用时，实例将以传统 BIOS 启动，并禁用 UEFI 相关功能。如果您依赖于 UEFI 相关功能的可用性，请将 AMI 启动模式参数设置为 `uefi`。

Amazon CLI

设置 AMI 的启动模式

1. 对实例的卷和操作系统进行适当的修改，以支持通过选定的启动模式启动。所需的修改是特定于操作系统的。有关更多信息，请参阅操作系统的手册。

   > **Warning**
   >
   > 如果不执行此步骤，AMI 将无法使用。

2. 要查找实例的卷 ID，请使用 describe-instances 命令。您将在下一步中创建此卷的快照。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
```

```
    --query Reservations[].Instances[].BlockDeviceMappings
```

下面是示例输出。

```
[
    [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "AttachTime": "2024-07-11T01:05:51+00:00",
                "DeleteOnTermination": true,
                "Status": "attached",
                "VolumeId": "vol-1234567890abcdef0"
            }
        }
    ]
]
```

3. 要创建卷的快照，请使用 create-snapshot 命令。使用上一步中的卷 ID。

```
aws ec2 create-snapshot \
    --volume-id vol-01234567890abcdef \
    --description "my snapshot"
```

下面是示例输出。

```
{
    "Description": "my snapshot",
    "Encrypted": false,
    "OwnerId": "123456789012",
    "Progress": "",
    "SnapshotId": "snap-0abcdef1234567890",
    "StartTime": "",
    "State": "pending",
    "VolumeId": "vol-01234567890abcdef",
    "VolumeSize": 30,
    "Tags": []
}
```

4. 等到快照状态为 completed 后，继续执行下一步。要获取快照状态，请将 describe-snapshots 命令与上一步中的快照 ID 结合使用。

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-0abcdef1234567890 \
    --query Snapshots[].State \
    --output text
```

下面是示例输出。

```
completed
```

5. 要创建新的 AMI，请使用 register-image 命令。使用 CreateSnapshot 输出中 SnapshotId 的值。

   - 要将启动模式设置为 UEFI，请添加 --boot-mode 参数，并指定值 uefi。

   ```
   aws ec2 register-image \
      --description "my image" \
      --name "my-image" \
      --block-device-mappings "DeviceName=/dev/
   sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
      --root-device-name /dev/sda1 \
      --virtualization-type hvm \
      --ena-support \
      --boot-mode uefi
   ```

   - 要将启动模式设置为 uefi-preferred，请将 --boot-mode 的值设置为 uefi-preferred

   ```
   aws ec2 register-image \
      --description "my description" \
      --name "my-image" \
      --block-device-mappings "DeviceName=/dev/
   sda1,Ebs={SnapshotId=snap-0abcdef1234567890,DeleteOnTermination=true}" \
      --root-device-name /dev/sda1 \
      --virtualization-type hvm \
      --ena-support \
      --boot-mode uefi-preferred
   ```

6. （可选）要验证新创建的 AMI 是否具有您指定的启动模式，请使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-id ami-1234567890abcdef0 \
```

```
    --query Images[].BootMode \
    --output text
```

下面是示例输出。

```
uefi
```

PowerShell

设置 AMI 的启动模式

1. 对实例的卷和操作系统进行适当的修改，以支持通过选定的启动模式启动。所需的修改是特定于操作系统的。有关更多信息，请参阅操作系统的手册。

   > **Warning**
   >
   > 如果不执行此步骤，AMI 将无法使用。

2. 要查找实例的卷 ID，请使用 Get-EC2Instance cmdlet。

   ```
   (Get-EC2Instance `
       -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
   ```

   下面是示例输出。

   ```
   AssociatedResource  :
   AttachTime          : 7/11/2024 1:05:51 AM
   DeleteOnTermination : True
   Operator            :
   Status              : attached
   VolumeId            : vol-01234567890abcdef
   ```

3. 要创建卷的快照，请使用 New-EC2Snapshot cmdlet。使用上一步中的卷 ID。

   ```
   New-EC2Snapshot `
       -VolumeId vol-01234567890abcdef `
       -Description "my snapshot"
   ```

   下面是示例输出。

```
AvailabilityZone          :
Description               : my snapshot
Encrypted                 : False
FullSnapshotSizeInBytes   : 0
KmsKeyId                  :
OwnerId                   : 123456789012
RestoreExpiryTime         :
SnapshotId                : snap-0abcdef1234567890
SseType                   :
StartTime                 : 4/25/2025 6:08:59 PM
State                     : pending
StateMessage              :
VolumeId                  : vol-01234567890abcdef
VolumeSize                : 30
```

4. 等到快照状态为 completed 后，继续执行下一步。要获取快照状态，请将 Get-EC2Snapshot cmdlet 与上一步中的快照 ID 结合使用。

```
(Get-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890).State.Value
```

下面是示例输出。

```
completed
```

5. 要创建新的 AMI，请使用 Register-EC2Image cmdlet。使用 New-EC2Snapshot 输出中 SnapshotId 的值。

   - 要将启动模式设置为 UEFI，请添加 -BootMode 参数，并指定值 uefi。

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
   -Description "my image" `
   -Name "my-image" `
   -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
   -RootDeviceName /dev/xvda `
   -EnaSupport $true `
   -BootMode uefi
```

   - 要将启动模式设置为 uefi-preferred，请将 -BootMode 的值设置为 uefi-preferred

```
$block = @{SnapshotId=snap-0abcdef1234567890}
Register-EC2Image `
   -Description "my image" `
   -Name "my-image" `
   -BlockDeviceMapping @{DeviceName="/dev/xvda";Ebs=$block} `
   -RootDeviceName /dev/xvda `
   -EnaSupport $true `
   -BootMode uefi-preferred
```

6. （可选）要验证新创建的 AMI 是否具有您指定的启动模式，请使用 Get-EC2Image cmdlet。

```
(Get-EC2Image `
    -ImageId ami-1234567890abcdef0).BootMode.Value
```

下面是示例输出。

```
uefi
```

Amazon EC2 实例的 UEFI 变量

当您启动实例且其启动模式设置为 UEFI 时，将为变量创建键值存储。UEFI 和实例操作系统可以使用该键值存储来存储 UEFI 变量。

启动加载程序和操作系统使用 UEFI 变量来配置早期系统启动。它们允许操作系统管理启动过程的某些设置（例如启动顺序）或管理 UEFI 安全启动的密钥。

> **Warning**
>
> 可以连接到实例的任何人（以及可能在实例上运行的任何软件），或者有权在实例上使用 GetInstanceUefiData API 的任何人都可以读取变量。您绝不应将敏感数据（例如密码或个人身份信息）存储在 UEFI 变量存储中。

UEFI 变量持久性

- 对于在 2022 年 5 月 10 日或之前启动的实例，UEFI 变量会在重启或停止时擦除。
- 对于在 2022 年 5 月 11 日或之后启动的实例，标记为非易失性的 UEFI 变量将在重启和停止/启动时保留。

- 裸机实例不会在实例停止/启动操作中保留 UEFI 非易失性变量。

适用于 Amazon EC2 实例的 UEFI 安全启动

UEFI 安全启动构建在 Amazon EC2 长期以来的安全启动过程之上，并提供了额外的深度防御功能，可以帮助客户保护软件免受重新引导过程中持续存在的威胁。它确保实例只启动使用了加密密钥签名的软件。密钥存储在 UEFI 非易失性变量存储的密钥数据库中。UEFI 安全启动可防止未经授权地修改实例启动流程。

内容

- 如何将 UEFI 安全启动与 Amazon EC2 实例配合使用
- Amazon EC2 上 UEFI 安全启动的要求
- 验证 Amazon EC2 实例是否启用了 UEFI 安全启动
- 使用自定义 UEFI 安全启动密钥创建 Linux AMI
- 为 UEFI 安全启动创建 Amazon 二进制 blob

如何将 UEFI 安全启动与 Amazon EC2 实例配合使用

UEFI 安全启动是 UEFI 中指定的一项功能，它提供了有关启动链状态的验证。它旨在确保固件自初始化后只执行经过加密验证的 UEFI 二进制文件。这些二进制文件包括 UEFI 驱动程序和主启动加载程序以及链加载组件。

UEFI 安全启动指定了四个密钥数据库，这些数据库在信任链中使用。数据库存储在 UEFI 变量存储中。

信任链如下所示：

平台密钥（PK）数据库

PK 数据库是信任根。它包含一个公有 PK 密钥，该密钥在信任链中用于更新密钥交换密钥（KEK）数据库。

要更改 PK 数据库，必须拥有私有 PK 密钥才能签署更新请求。这包括通过写入空 PK 密钥来删除 PK 数据库。

密钥交换密钥（KEK）数据库

KEK 数据库是信任链中用于更新签名（db）和拒绝列表（dbx）数据库的公有 KEK 密钥的列表。

要更改公有 KEK 数据库，必须拥有私有 PK 密钥才能签署更新请求。

签名（db）数据库

db 数据库是信任链中用于验证所有 UEFI 启动二进制文件的公有密钥和哈希值的列表。

要更改 db 数据库，必须拥有私有 PK 密钥或私有 KEK 密钥中的任何一个才能签署更新请求。

签名拒绝列表（dbx）数据库

dbx 数据库是不受信任的公有密钥和二进制哈希的列表，在信任链中用作吊销文件。

dbx 数据库始终优先于所有其他密钥数据库。

要更改 dbx 数据库，必须拥有私有 PK 密钥或私有 KEK 密钥中的任何一个才能签署更新请求。

UEFI 论坛在 https://uefi.org/revocationlistfile 为许多已知不良的二进制文件和证书维护了公开可用的 dbx。

> **Important**
>
> UEFI 安全启动在任何 UEFI 二进制文件上强制执行签名验证。要允许在 UEFI 安全启动中执行 UEFI 二进制文件，请使用上述任何私有 db 密钥对其进行签名。

默认情况下，UEFI 安全启动处于禁用状态，且系统处于 `SetupMode`。当系统处于 `SetupMode` 时，所有密钥变量都可以在没有加密签名的情况下进行更新。设置 PK 后，启用 UEFI 安全启动并退出 SetupMode。

Amazon EC2 上 UEFI 安全启动的要求

当您使用支持的 AMI 和支持的实例类型启动 Amazon EC2 实例时，该实例将根据其 UEFI 安全启动数据库自动验证 UEFI 启动二进制文件。无需其他配置。您还可以在启动后在实例上配置 UEFI 安全启动。

> **Note**
>
> UEFI 安全启动可以保护您的实例及其操作系统免受启动流程修改的影响。如果从启用了 UEFI 安全引导的源 AMI 创建新的 AMI，并在复制过程中修改了某些参数，如更改了 AMI 中的 `UefiData`，则可以禁用 UEFI 安全引导。

内容

- 支持的 AMI
- 支持的实例类型

支持的 AMI

Linux AMI

要启动 Linux 实例，Linux AMI 必须启用 UEFI 安全引导。

从 AL2023 2023.1 版本开始，Amazon Linux 支持 UEFI 安全启动。但是，默认 AMI 中不会启用 UEFI 安全启动。有关更多信息，请参阅《AL2023 User Guide》中的 UEFI Secure Boot。旧版本的 Amazon Linux AMI 不会为 UEFI 安全启动启用。要使用受支持的 AMI，您必须在自己的 Linux AMI 上执行多个配置步骤。有关更多信息，请参阅 使用自定义 UEFI 安全启动密钥创建 Linux AMI。

目前，我们不支持使用 import-image 命令导入包含 UEFI 安全启动的 Windows。

> **Note**
>
> Amazon 提供了预配置用于 UEFI 安全引导的 Windows AMI。此类型的 AMI 在中国（北京）和中国（宁夏）区域不可用。

支持的实例类型

所有支持 UEFI 的虚拟化实例类型也支持 UEFI 安全启动。有关支持 UEFI 安全引导的实例类型，请参阅 UEFI 启动模式的要求。

> **Note**
>
> 裸机实例类型不支持 UEFI 安全启动。

验证 Amazon EC2 实例是否启用了 UEFI 安全启动

您可以使用以下过程确定 Amazon EC2 是否启用了 UEFI 安全启动。

Linux 实例

您可以使用 `mokutil` 实用程序验证 Linux 实例是否启用了 UEFI 安全启动。如果实例上未安装 `mokutil`，则必须进行安装。有关 Amazon Linux 2 的安装说明，请参阅 Find and install software packages on an Amazon Linux 2 instance。有关其他 Linux 发行版，请参阅特定于该版本的文档。

验证 Linux 实例是否启用了 UEFI 安全启动

连接到您的实例并在终端窗口中作为 root 运行以下命令。

```
mokutil --sb-state
```

下面是示例输出。

- 如果启用了 UEFI 安全启动，则输出包含 SecureBoot enabled。
- 如果未启用 UEFI 安全启动，则输出将包含 SecureBoot disabled 或 Failed to read SecureBoot。

使用自定义 UEFI 安全启动密钥创建 Linux AMI

这些说明向您展示如何使用 UEFI 安全启动和定制私有密钥创建 Linux AMI。从 AL2023 2023.1 版本开始，Amazon Linux 支持 UEFI 安全启动。有关更多信息，请参阅《Amazon Linux 2023 用户指南》中的 UEFI Secure Boot on AL2023。

> **Important**
>
> 以下过程仅供高级用户使用。您必须对 SSL 和 Linux 发行版启动流程有足够的了解才能使用这些步骤。

先决条件

- 将使用以下工具：
  - OpenSSL – https://www.openssl.org/
  - efivar – https://github.com/rhboot/efivar
  - efitools – https://git.kernel.org/pub/scm/linux/kernel/git/jejb/efitools.git/
  - get-instance-uefi-data 命令
- 您的 Linux 实例必须使用支持 UEFI 启动模式的 Linux AMI 启动，并且存在非易失性数据。

在 SetupMode 中创建了没有 UEFI 安全启动密钥的新创建的实例，从而使您能够注册自己的密钥。有些 AMI 预先配置了 UEFI 安全启动，您无法更改现有密钥。如果要更改密钥，您必须基于原始 AMI 创建新 AMI。

有两种方法可以在变量存储中传播密钥，这些方法在后面的选项 A 和选项 B 中进行说明。选项 A 描述了如何在实例内执行此操作，以模仿真实硬件的流程。选项 B 描述了如何创建二进制 Blob，然后在创建 AMI 时将其作为 base64 编码文件传递。对于这两种选项，您必须首先创建三个密钥对，它们将用于信任链。

要创建支持 UEFI 安全启动的 Linux AMI，请先创建三个密钥对，然后完成选项 A 或选项 B，但不能同时完成两者：

- 任务 1：创建密钥对
- 任务 2 - 选项 A：从实例内向变量存储添加密钥
- 任务 2 - 选项 B：创建包含预填充变量存储的二进制 blob

任务 1：创建密钥对

UEFI 安全启动基于在信任链中使用的以下三个密钥数据库：平台密钥（PK）、密钥交换密钥（KEK）和签名数据库（db）。[1]

在实例上创建每个密钥。要以对 UEFI 安全启动标准有效的格式准备公钥，请为每个密钥创建一个证书。DER 定义 SSL 格式（格式的二进制编码）。然后，您将每个证书转换为 UEFI 签名列表，这是 UEFI 安全启动所理解的二进制格式。最后，使用相关密钥签署每个证书。

任务

- 准备创建密钥对
- 密钥对 1：创建平台密钥（PK）
- 密钥对 2：创建密钥交换密钥（KEK）
- 密钥对 3：创建签名数据库（db）
- 用私有密钥签署启动映像（内核）

准备创建密钥对

在创建密钥对之前，请创建用于密钥生成的全局唯一标识符（GUID）。

1. 连接到实例。
2. 在 Shell 提示符中运行以下命令。

```
uuidgen --random > GUID.txt
```

密钥对 1：创建平台密钥（PK）

PK 是 UEFI 安全启动实例的信任根。私有 PK 用于更新 KEK，这反过来又可以用来将授权密钥添加到签名数据库（db）。

X.509 标准用于创建密钥对。有关标准的信息，请参阅 Wikipedia 上的 X.509。

要创建 PK

1. 创建密钥。您必须将变量命名为 PK。

```
openssl req -newkey rsa:4096 -nodes -keyout PK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Platform key/" -out PK.crt
```

   指定了以下参数：

   - `-keyout PK.key` – 私有密钥文件。
   - `-days 3650` – 证书的有效天数。
   - `-out PK.crt` – 用于创建 UEFI 变量的证书。
   - CN=*Platform key* – 密钥的公用名（CN）。您可以输入您自己企业的名称而不是*####*。
2. 创建证书。

```
openssl x509 -outform DER -in PK.crt -out PK.cer
```

3. 将证书转换为 UEFI 签名列表。

```
cert-to-efi-sig-list -g "$(< GUID.txt)" PK.crt PK.esl
```

4. 使用私有 PK（自签名）签署 UEFI 签名列表。

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt PK PK.esl PK.auth
```

密钥对 2：创建密钥交换密钥（KEK）

私有 KEK 用于向 db 添加密钥，该数据库是要在系统上启动的授权签名列表。

要创建 KEK

1. 创建密钥。

```
openssl req -newkey rsa:4096 -nodes -keyout KEK.key -new -x509 -sha256 -days 3650 -
subj "/CN=Key Exchange Key/" -out KEK.crt
```

2. 创建证书。

```
openssl x509 -outform DER -in KEK.crt -out KEK.cer
```

3. 将证书转换为 UEFI 签名列表。

```
cert-to-efi-sig-list -g "$(< GUID.txt)" KEK.crt KEK.esl
```

4. 使用私有 PK 签署签名列表。

```
sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt KEK KEK.esl KEK.auth
```

密钥对 3：创建签名数据库（db）

db 列表包含授权在系统上启动的授权密钥。要修改列表，需要私有 KEK。启动映像将使用此步骤中创建的私有密钥签名。

要创建 db

1. 创建密钥。

```
openssl req -newkey rsa:4096 -nodes -keyout db.key -new -x509 -sha256 -days 3650 -
subj "/CN=Signature Database key/" -out db.crt
```

2. 创建证书。

```
openssl x509 -outform DER -in db.crt -out db.cer
```

3. 将证书转换为 UEFI 签名列表。

```
cert-to-efi-sig-list -g "$(< GUID.txt)" db.crt db.esl
```

4. 用私有 KEK 签署签名列表。

```
sign-efi-sig-list -g "$(< GUID.txt)" -k KEK.key -c KEK.crt db db.esl db.auth
```

用私有密钥签署启动映像（内核）

对于 Ubuntu 22.04，以下映像需要签名。

```
/boot/efi/EFI/ubuntu/shimx64.efi
/boot/efi/EFI/ubuntu/mmx64.efi
/boot/efi/EFI/ubuntu/grubx64.efi
/boot/vmlinuz
```

要签署映像

使用以下语法签署映像。

```
sbsign --key db.key --cert db.crt --output /boot/vmlinuz /boot/vmlinuz
```

> **Note**
>
> 您必须签署所有新内核。*/boot/vmlinuz* 通常会将符号链接到上次安装的内核。

请参阅您的发行版的文档以了解启动链和所需映像。

[1] 感谢 ArchWiki 社区所做的所有工作。创建 PK、创建 KEK、创建数据库和签署映像的命令来自创建密钥，由 ArchWiki 维护团队和/或 ArchWiki 贡献者撰写。

任务 2 - 选项 A：从实例内向变量存储添加密钥

创建了三个密钥对后，您可以通过完成以下步骤连接到实例并从实例内将密钥添加到变量存储中。或者，完成the section called “任务 2 - 选项 B：创建包含预填充变量存储的二进制 blob”中的步骤。

选项 A 步骤：

- 步骤 1：启动将支持 UEFI 安全启动的实例
- 步骤 2：配置实例以支持 UEFI 安全启动
- 步骤 3：从实例创建 AMI

步骤 1：启动将支持 UEFI 安全启动的实例

当您使用以下先决条件启动实例时，该实例将准备好配置为支持 UEFI 安全启动。您只能在启动时在实例上启用对 UEFI 安全启动的支持；无法在以后启用它。

先决条件

- AMI – Linux AMI 必须支持 UEFI 启动模式。要验证 AMI 是否支持 UEFI 启动模式，AMI 启动模式参数必须为 uefi。有关更多信息，请参阅 确定 Amazon EC2 AMI 的启动模式参数。

  请注意，Amazon 仅提供配置为支持基于 Graviton 实例类型的 UEFI 的 Linux AMI。Amazon 当前不提供支持 UEFI 启动模式的 x86_64 Linux AMI。您可以配置自己的 AMI 以支持所有架构的 UEFI 启动模式。若要配置自己的 AMI 以支持 UEFI 启动模式，您必须在自己的 AMI 上执行多个配置步骤。有关更多信息，请参阅 设置 Amazon EC2 AMI 的启动模式。
- 实例类型 – 许多支持 UEFI 的虚拟化实例类型也支持 UEFI 安全启动。裸机实例类型不支持 UEFI 安全启动。有关支持 UEFI 安全引导的实例类型，请参阅 UEFI 启动模式的要求。
- 在 UEFI 安全启动发布后启动您的实例。只有在 2022 年 5 月 10 日（UEFI 安全启动发布时）之后启动的实例才能支持 UEFI 安全启动。

启动实例后，您可以通过检查是否存在 UEFI 数据来验证它是否已准备好配置为支持 UEFI 安全启动（换句话说，您可以继续步骤 2）。UEFI 数据的存在表明非易失性数据是持久的。

验证您的实例是否已为步骤 2 做好准备

使用 get-instance-uefi-data 命令并指定实例 ID。

```
aws ec2 get-instance-uefi-data --instance-id i-1234567890abcdef0
```

如果输出中存在 UEFI 数据，则实例已准备好进入步骤 2。如果输出为空，则无法将实例配置为支持 UEFI 安全启动。如果您的实例在 UEFI 安全启动支持可用之前启动，则可能会发生这种情况。启动新实例，然后重试。

步骤 2：配置实例以支持 UEFI 安全启动

在实例上的 UEFI 变量存储中注册密钥对

> **Warning**
>
> 您必须在注册密钥之后签署启动映像，否则您将无法启动实例。

创建签名的 UEFI 签名列表后（PK、KEK 和 db），必须将它们注册到 UEFI 固件中。

只有在以下情况下才能写入 PK 变量：

- 尚未注册 PK，当 SetupMode 变量为 1 时即表明这一点。使用以下命令检查此内容。输出为 1 或 0。

```
efivar -d -n 8be4df61-93ca-11d2-aa0d-00e098032b8c-SetupMode
```

- 新 PK 由现有 PK 的私有密钥签名。

在 UEFI 变量存储中注册密钥

以下命令必须在实例上运行。

如果启用了 SetupMode（值为 1），您可以通过在实例上运行以下命令注册密钥：

```
[ec2-user ~]$ efi-updatevar -f db.auth db
```

```
[ec2-user ~]$ efi-updatevar -f KEK.auth KEK
```

```
[ec2-user ~]$ efi-updatevar -f PK.auth PK
```

验证 UEFI 安全启动是否已启用

要验证 UEFI 安全启动是否已启用，请遵照 验证 Amazon EC2 实例是否启用了 UEFI 安全启动 中的步骤。

现在，您可以使用 get-instance-uefi-data CLI 命令导出 UEFI 变量存储，或者继续执行下一步并签署启动映像，以重新启动到启用了 UEFI 安全启动的实例中。

步骤 3：从实例创建 AMI

要从实例创建 AMI，您可以使用控制台或 CreateImage API、CLI 或开发工具包。有关控制台说明，请参阅 创建 Amazon EBS-backed AMI。有关 API 说明，请参阅 CreateImage。

> **Note**
>
> CreateImage API 会自动将实例的 UEFI 变量存储复制到 AMI。控制台会使用 CreateImage API。使用此 AMI 启动实例后，实例将具有相同的 UEFI 变量存储。

任务 2 - 选项 B：创建包含预填充变量存储的二进制 blob

创建了三个密钥对后，您可以创建一个二进制 blob，其中包含含有 UEFI 安全启动密钥的预填充变量存储。或者，完成the section called “任务 2 - 选项 A：从实例内向变量存储添加密钥”中的步骤。

> **Warning**
>
> 您必须先签署您的启动映像，然后才能注册密钥，否则您将无法启动实例。

选项 B 步骤：

- 步骤 1：创建新的变量存储或更新现有变量存储
- 步骤 2：在 AMI 创建时上传二进制 blob

步骤 1：创建新的变量存储或更新现有变量存储

您可以使用 python-uefivars 工具离线创建变量存储，而不需要正在运行的实例。该工具可以从您的密钥中创建一个新的变量存储。该脚本目前支持 EDK2 格式、Amazon 格式，以及使用更高级别的工具更容易编辑的 JSON 表示法。

在没有运行实例的情况下离线创建变量存储

1. 通过以下链接下载该工具。

   ```
   https://github.com/awslabs/python-uefivars
   ```

2. 通过运行以下命令从密钥创建新的变量存储。这将在 *your_binary_blob*.bin 中创建 base64 编码的二进制 blob。该工具还通过 `-I` 参数支持更新二进制 blob。

   ```
   ./uefivars.py -i none -o aws -O your_binary_blob.bin -P PK.esl -K KEK.esl --db
    db.esl --dbx dbx.esl
   ```

步骤 2：在 AMI 创建时上传二进制 blob

使用 register-image 传递 UEFI 变量存储数据。对于 `--uefi-data` 参数，指定您的二进制 blob，对于 `--boot-mode` 参数，请指定 `uefi`。

```
aws ec2 register-image \
```

```
    --name uefi_sb_tpm_register_image_test \
    --uefi-data $(cat your_binary_blob.bin) \
    --block-device-mappings "DeviceName=/dev/sda1,Ebs=
{SnapshotId=snap-0123456789example,DeleteOnTermination=true}" \
    --architecture x86_64 \
    --root-device-name /dev/sda1 \
    --virtualization-type hvm \
    --ena-support \
    --boot-mode uefi
```

为 UEFI 安全启动创建 Amazon 二进制 blob

您可以使用以下步骤在 AMI 创建期间自定义 UEFI 安全启动变量。这些步骤中使用的 KEK 截止到 2021 年 9 月。如果 Microsoft 更新 KEK，您必须使用最新的 KEK。

要创建 Amazon 二进制 blob

1. 创建一个空的 PK 签名列表。

```
touch empty_key.crt
cert-to-efi-sig-list empty_key.crt PK.esl
```

2. 下载 KEK 证书。

```
https://go.microsoft.com/fwlink/?LinkId=321185
```

3. 将 KEK 证书包装在 UEFI 签名列表中（`siglist`）。

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_KEK.esl MicCorKEKCA2011_2011-06-24.crt
```

4. 下载 Microsoft 的 db 证书。

```
https://www.microsoft.com/pkiops/certs/MicWinProPCA2011_2011-10-19.crt
https://www.microsoft.com/pkiops/certs/MicCorUEFCA2011_2011-06-27.crt
```

5. 生成 db 签名列表。

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_Win_db.esl MicWinProPCA2011_2011-10-19.crt
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
 MS_UEFI_db.esl MicCorUEFCA2011_2011-06-27.crt
```

```
cat MS_Win_db.esl MS_UEFI_db.esl > MS_db.esl
```

6. Unified Extensible Firmware Interface Forum 不再提供 DBX 文件。这些文件现在由 Microsoft 在 GitHub 上提供。请从 Microsoft 安全启动更新存储库下载最新的 DBX 更新，网址为 https://github.com/microsoft/secureboot_objects。
7. 解压已签名的更新二进制文件。

   使用下面的脚本内容创建 `SplitDbxContent.ps1`。或者，也可以使用 `Install-Script -Name SplitDbxContent` 从 PowerShell Gallery 安装脚本。

```
<#PSScriptInfo

.VERSION 1.0

.GUID ec45a3fc-5e87-4d90-b55e-bdea083f732d

.AUTHOR Microsoft Secure Boot Team

.COMPANYNAME Microsoft

.COPYRIGHT Microsoft

.TAGS Windows Security

.LICENSEURI

.PROJECTURI

.ICONURI

.EXTERNALMODULEDEPENDENCIES

.REQUIREDSCRIPTS

.EXTERNALSCRIPTDEPENDENCIES

.RELEASENOTES
Version 1.0: Original published version.

#>

<#
.DESCRIPTION
```

```
 Splits a DBX update package into the new DBX variable contents and the signature
 authorizing the change.
 To apply an update using the output files of this script, try:
 Set-SecureBootUefi -Name dbx -ContentFilePath .\content.bin -SignedFilePath .
\signature.p7 -Time 2010-03-06T19:17:21Z -AppendWrite'
.EXAMPLE
.\SplitDbxAuthInfo.ps1 DbxUpdate_x64.bin
#>


# Get file from script input
$file  = Get-Content -Encoding Byte $args[0]

# Identify file signature
$chop = $file[40..($file.Length - 1)]
if (($chop[0] -ne 0x30) -or ($chop[1] -ne 0x82 )) {
    Write-Error "Cannot find signature"
    exit 1
}

# Signature is known to be ASN size plus header of 4 bytes
$sig_length = ($chop[2] * 256) + $chop[3] + 4
$sig = $chop[0..($sig_length - 1)]

if ($sig_length -gt ($file.Length + 40)) {
    Write-Error "Signature longer than file size!"
    exit 1
}

# Content is everything else
$content = $file[0..39] + $chop[$sig_length..($chop.Length - 1)]

# Write signature and content to files
Set-Content -Encoding Byte signature.p7 $sig
Set-Content -Encoding Byte content.bin $content
```

使用该脚本解压缩已签名的 DBX 文件。

```
PS C:\Windows\system32> SplitDbxContent.ps1 .\dbx.bin
```

这会生成两个文件：signature.p7 和 content.bin。在下一步中使用 content.bin。

8. 使用 uefivars.py 脚本构建 UEFI 变量存储。

```
./uefivars.py -i none -o aws -O uefiblob-microsoft-keys-empty-pk.bin -P ~/PK.esl -K
 ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx ~/content.bin
```

9. 检查二进制 blob 和 UEFI 变量存储。

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o json | less
```

10. 您可以通过再次将 blob 传递给同一个工具来更新它。

```
./uefivars.py -i aws -I uefiblob-microsoft-keys-empty-pk.bin -o aws -O uefiblob-
microsoft-keys-empty-pk.bin -P ~/PK.esl -K ~/MS_Win_KEK.esl --db ~/MS_db.esl  --dbx
 ~/content.bin
```

预期输出

```
Replacing PK
Replacing KEK
Replacing db
Replacing dbx
```

将加密与 EBS 支持的 AMI 结合使用

由 Amazon EBS 快照支持的 AMI 可以利用 Amazon EBS 加密。可以将数据和根卷的快照加密并附加到 AMI。在启动实例和复制映像时，您可以包含 EBS 完全加密支持。在提供了 Amazon KMS 的所有区域中，支持在这些操作中使用加密参数。

从 AMIs 中启动带加密 EBS 卷的 EC2 实例的方式与其他实例相同。另外，从未加密 EBS 快照支持的 AMI 中启动实例时，您可以在启动过程中将部分或全部卷加密。

与 EBS 卷相似，可使用默认 Amazon KMS key 或您指定的客户托管密钥加密 AMI 中的快照。在所有情况下，您都必须拥有使用所选 KMS 密钥 的权限。

带加密快照的 AMI 可以跨 Amazon 账户共享。有关更多信息，请参阅了解 Amazon EC2 中共享 AMI 的使用情况。

“将加密与 EBS 支持的 AMI 结合使用”相关主题

- 启动实例场景
- 映像复制场景

启动实例场景

Amazon EC2 实例是通过 Amazon Web Services 管理控制台 或者直接使用 Amazon EC2 API 或 CLI，使用 RunInstances 操作以及通过块储存设备映射提供的参数在 AMI 中启动的。有关更多信息，请参阅 Amazon EC2 实例上卷的块设备映射。有关通过 Amazon CLI 控制数据块设备映射的示例，请参阅启动、列出和终止 EC2 实例。

如果不使用显式加密参数，在默认情况下，RunInstances 操作会在从 AMI 的源快照中还原 EBS 卷时保持这些快照的现有加密状态。如果启用默认加密，从 AMI 中（无论使用加密还是未加密的快照）创建的所有卷都会被加密。如果在默认情况下并未启用加密，则实例保持 AMI 的加密状态。

您也可以启动实例，同时通过提供加密参数来对结果卷应用新的加密状态。因此，会观察到以下行为：

启动时不指定加密参数

- 未加密快照会还原为未加密卷，除非已在默认情况下启用加密，那么所有新创建的所有卷都将加密。
- 您拥有的加密快照会还原为使用相同 KMS 密钥 加密的卷。
- 您未拥有的加密快照（例如，与您共享了 AMI）会还原到由您的 Amazon 账户的默认 KMS 密钥加密的卷。

可以通过提供加密参数覆盖默认行为。可用参数包括 Encrypted 和 KmsKeyId。仅设置 Encrypted 参数会得到以下结果：

已设置 **Encrypted**，但未指定 **KmsKeyId** 时的实例启动行为

- 未加密快照会还原到由您的 Amazon 账户的默认 KMS 密钥加密的 EBS 卷。
- 您拥有的加密快照会还原到由相同 KMS 密钥 加密的 EBS 卷。（也就是说，Encrypted 参数没有影响。）
- 您未拥有的加密快照（即，与您共享了 AMI）会还原到由您的 Amazon 账户的默认 KMS 密钥加密的卷。（也就是说，Encrypted 参数没有影响。）

如果同时设置 Encrypted 和 KmsKeyId 参数，可以为加密操作指定非默认 KMS 密钥。会实现以下行为：

同时设置 **Encrypted** 和 **KmsKeyId** 的实例

- 未加密快照会还原到由指定 KMS 密钥 加密的 EBS 卷。
- 加密快照还原为未使用原始 KMS 密钥 加密，而是使用指定 KMS 密钥 加密的 EBS 卷。

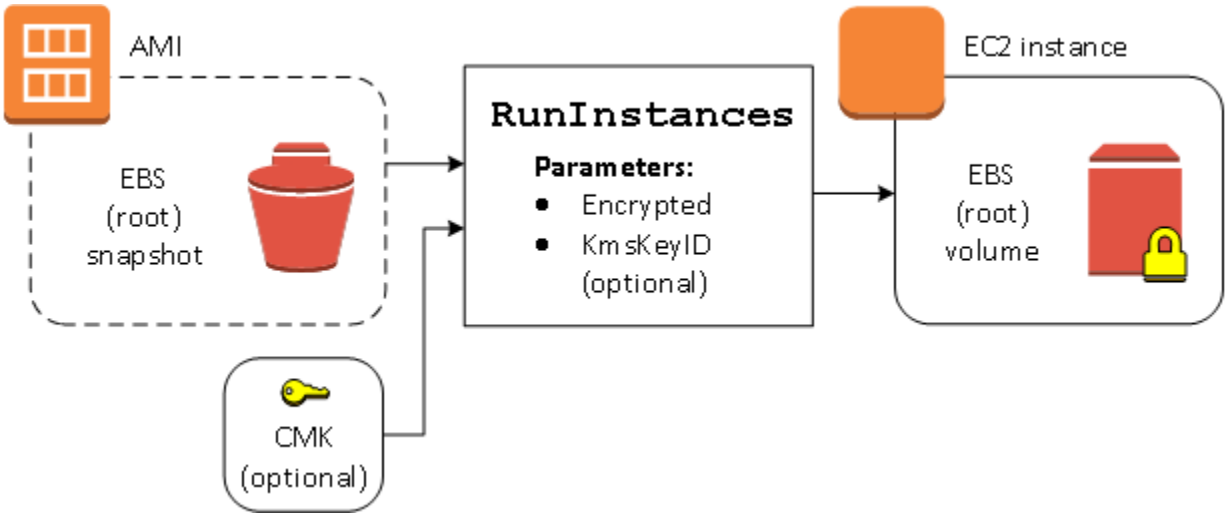
提交 KmsKeyId 但没有同时设置 Encrypted 参数会导致错误。

以下部分提供使用非默认加密参数从 AMI 中启动实例的示例。在以下每个场景中，提供给 RunInstances 操作的参数会导致在使用快照还原卷的过程中加密状态发生变化。

有关使用控制台从 AMI 启动实例的信息，请参阅 启动 Amazon EC2 实例。

在启动过程中加密卷

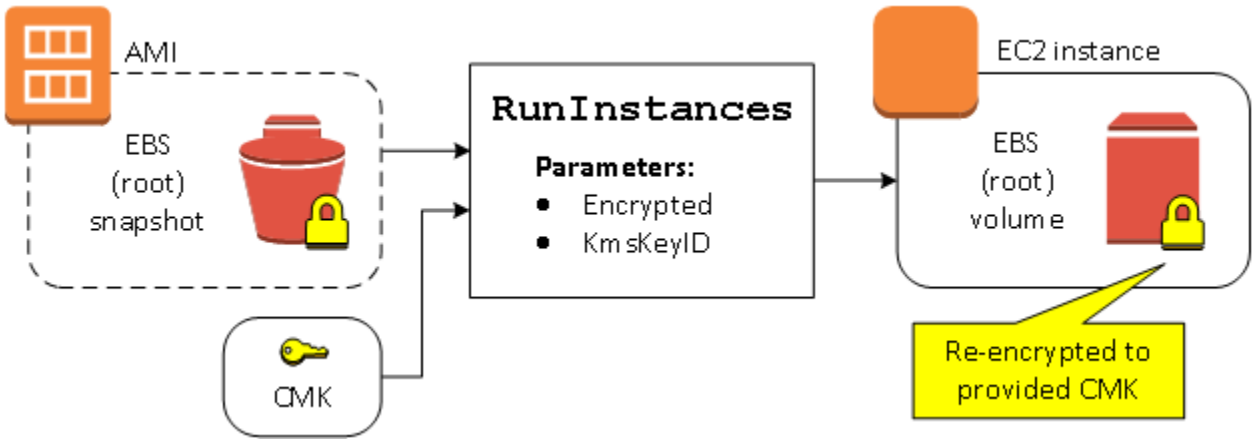
在该示例中，使用未加密快照支持的 AMI 启动带加密 EBS 卷的 EC2 实例。



仅使用 Encrypted 参数的结果是对该实例中的卷行加密。提供 KmsKeyId 参数是可选的。如果未指定 KMS 密钥 ID，会使用 Amazon 账户的默认 KMS 密钥加密卷。要使用您拥有的不同 KMS 密钥 加密卷，请提供 KmsKeyId 参数。

在启动过程中重新加密卷

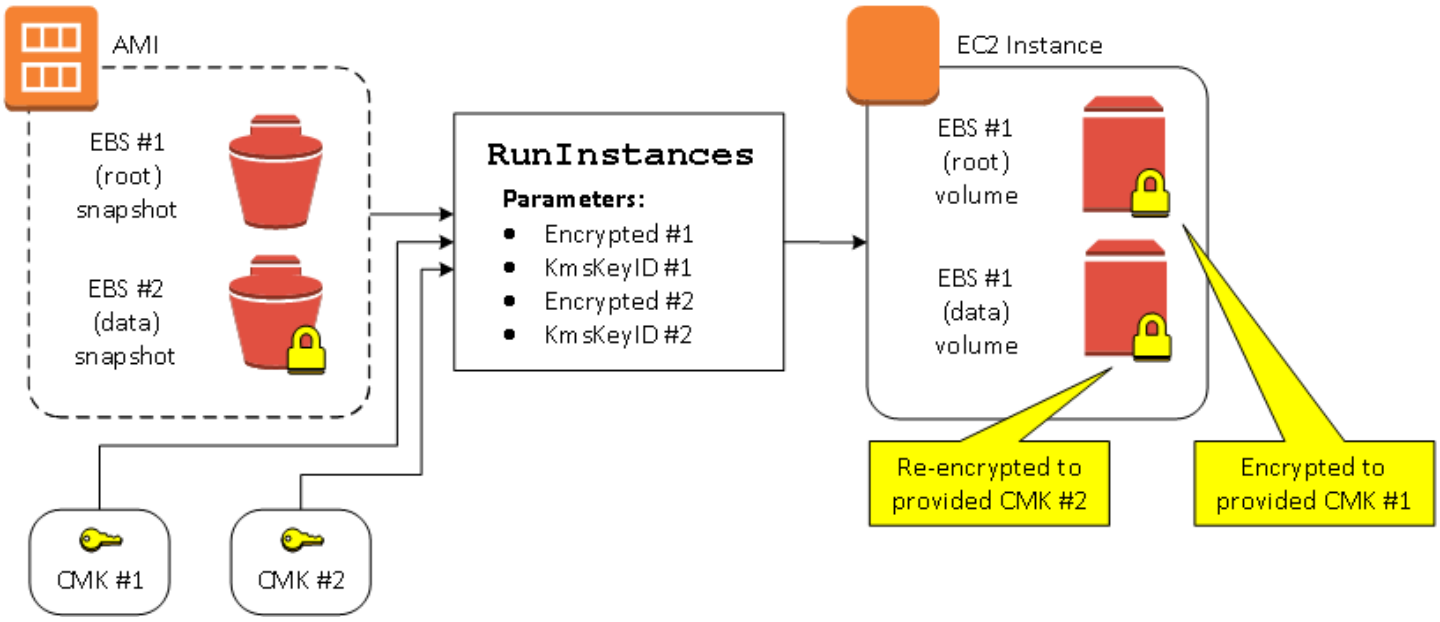
在该示例中，使用加密快照支持的 AMI 启动带有由新 KMS 密钥 加密的 EBS 卷的 EC2 实例。



如果您拥有 AMI 且未提供加密参数，则生成的实例具有由与快照相同的 KMS 密钥 加密的卷。如果 AMI 是与他人共享而不是由您拥有，且您未提供加密参数，则由您的默认 KMS 密钥 对卷进行加密。如果按所示提供加密参数，则会由指定 KMS 密钥 对卷进行加密。

在启动过程中更改多个卷的加密状态

在这一更为复杂的示例中，会使用多个快照（分别具有自己的加密状态）支持的 AMI 启动带有新加密卷和重新加密卷的 EC2 实例。



在这种情况下，会为 `RunInstances` 操作提供用于各个源快照的加密参数。在指定所有可用的加密参数后，无论您是否拥有 AMI，结果实例都相同。

映像复制场景

可通过 Amazon Web Services 管理控制台 或者直接使用 Amazon EC2 API 或 CLI，使用 `CopyImage` 操作复制 Amazon EC2 AMI。

如果不使用显式加密参数，在默认情况下，`CopyImage` 操作会在复制 AMI 的源快照时保持这些快照的现有加密状态。您也可以复制 AMI，同时通过提供加密参数来对其关联的 EBS 快照应用新的加密状态。因此，会观察到以下行为：

复制时不指定加密参数

- 未加密快照会复制为另一个未加密快照，除非已在默认情况下启用加密，那么所有新创建的快照都将加密。
- 您拥有的加密快照会复制为使用相同 KMS 密钥 加密的快照。

- 您未拥有的加密快照（例如，与您共享了 AMI）会复制为由您的 Amazon 账户的默认 KMS 密钥加密的快照。

可以通过提供加密参数覆盖以上所有默认行为。可用参数包括 Encrypted 和 KmsKeyId。仅设置 Encrypted 参数会得到以下结果：

已设置 **Encrypted**，但未指定 **KmsKeyId** 时的复制映像行为

- 未加密快照会复制为由 Amazon 账户的默认 KMS 密钥加密的快照。
- 加密快照会复制为由相同 KMS 密钥 加密的快照。（也就是说，Encrypted 参数没有影响。）
- 您未拥有的加密快照（即，与您共享了 AMI）会复制为由您的 Amazon 账户的默认 KMS 密钥加密的卷。（也就是说，Encrypted 参数没有影响。）

通过同时设置 Encrypted 和 KmsKeyId 参数，可以为加密操作指定客户托管的 KMS 密钥。会实现以下行为：

同时设置 **Encrypted** 和 **KmsKeyId** 时的复制映像行为

- 未加密快照会复制为由指定 KMS 密钥 加密的快照。
- 加密快照会复制为未使用原始 KMS 密钥 加密，而是使用指定 KMS 密钥 加密的快照。

提交 KmsKeyId 但没有同时设置 Encrypted 参数会导致错误。

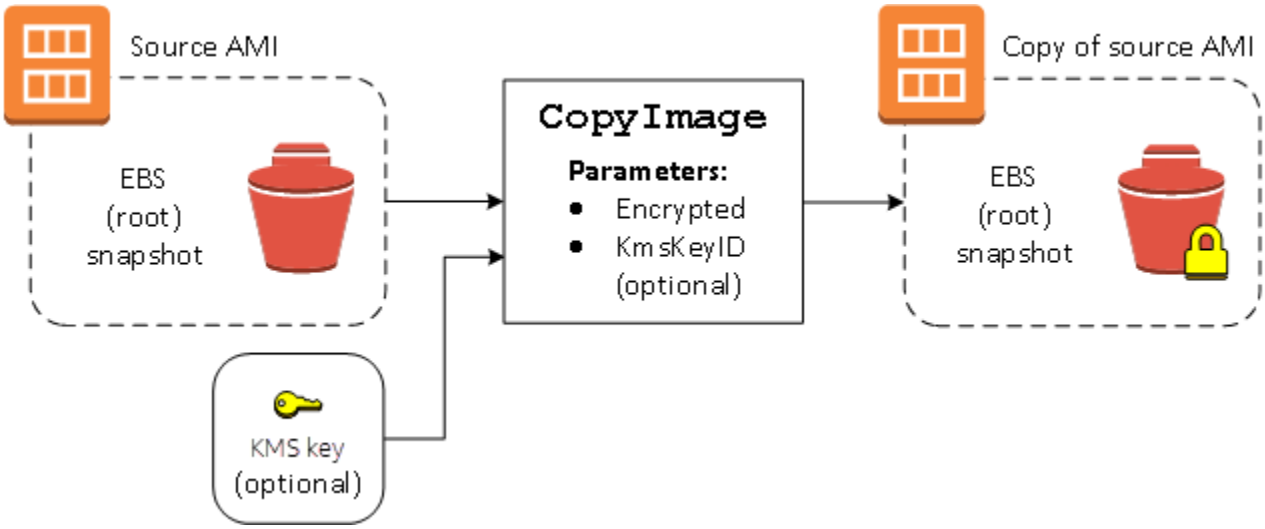
以下部分提供使用非默认加密参数复制 AMI，导致更改加密状态的示例。

有关使用控制台的说明，请参阅 复制 Amazon EC2 AMI。

在复制过程中将未加密映像加密

在这种情况下，由未加密根快照支持的 AMI 会复制到带加密根快照的 AMI。CopyImage 操作将使用两个加密参数（包括一个客户托管密钥）调用。因此，根快照的加密状态发生变化，这样目标 AMI 由与原快照具有相同数据的根快照支持，但使用指定的密钥进行加密。在两个 AMI 中，您都会产生快照的存储成本，以及从任一 AMI 启动的任何实例的费用。

> **Note**
>
> 启用默认加密与将 AMI 中所有快照的 Encrypted 参数设置为 true 具有相同的效果。



设置 `Encrypted` 参数会加密此实例的单个快照。如果您未指定 `KmsKeyId` 参数，会使用默认客户托管密钥来加密快照副本。

> **Note**
>
> 您也可以复制带多个快照的映像，并单独配置每个快照的加密状态。

了解 Amazon EC2 中共享 AMI 的使用情况

共享 AMI 是开发人员创建并可供其他人员使用的 AMI。要开始使用 Amazon EC2，最简单的方法之一是使用共享 AMI，您可以从中获得所需的组件，然后添加自定义内容。您还可以创建自己的 AMI 并与他人共享。

使用共享 AMI 需自行承担风险。Amazon 不保证其他 Amazon EC2 用户共享的 AMI 的完整性或安全性。因此，您应该像处理其他您可能会考虑在自己的数据中心部署的外来代码一样处理共享 AMI，对其执行适当的功能调查。我们建议您从可信来源（例如经过验证的提供者）获取 AMI。

经过验证的提供商

在 Amazon EC2 控制台中，将 Amazon 或经过验证的 Amazon 合作伙伴拥有的公有 AMI 标记为经过验证的提供商。

您也可以使用 describe-images Amazon CLI 命令标识来自经过验证的提供商的公共 AMI。Amazon 或经过验证的合作伙伴拥有的公共映像具有一个别名拥有者，即 `amazon`、`aws-backup-vault` 或 `aws-marketplace`。在 CLI 输出中，将为 `ImageOwnerAlias` 显示这些值。其他用户不能对其 AMI 使用别名。这使您可以轻松查找来自 Amazon 或经过验证的合作伙伴的 AMI。

要成为经过验证的提供商，您必须在 Amazon Web Services Marketplace 上注册成为卖方。在注册后，您可以在 Amazon Web Services Marketplace 上列出您的 AMI。有关更多信息，请参阅《Amazon Web Services Marketplace 卖方指南》中的以卖方身份开始使用和基于 AMI 的产品。

共享 AMI 主题

- 查找用于 Amazon EC2 实例的共享 AMI
- 准备使用适用于 Linux 的共享 AMI
- 使用“允许的 AMI”控制在 Amazon EC2 中对 AMI 的发现和使用
- 将您的 AMI 设为可在 Amazon EC2 中公开使用
- 了解 AMI 的屏蔽公共访问
- 与组织和组织单位共享 AMI
- 与特定 Amazon 账户共享 AMI
- 取消与您的 Amazon Web Services 账户 共享 AMI
- 创建共享 Linux AMI 的建议

如果您正在寻找有关其他主题的信息

- 有关创建 AMI 的信息，请参阅 the section called “创建 Amazon S3 支持的 AMI” 或 the section called “创建 AMI”。
- 有关在 Amazon Web Services Marketplace 中构建、交付和维护应用程序的信息，请参阅 Amazon Web Services Marketplace 文档。

查找用于 Amazon EC2 实例的共享 AMI

您可使用 Amazon EC2 控制台或命令行查找与您的 Amazon EC2 实例一起使用的共享 AMI。

AMI 是一种区域性资源。在搜索共享 AMI（公有或私有）时，必须在共享此 AMI 的同一区域中进行搜索。要使 AMI 能够在其他区域使用，请将该 AMI 复制到该区域，然后进行共享。有关更多信息，请参阅 复制 Amazon EC2 AMI。

Console

控制台提供了一个 AMI 筛选条件字段。您也可以使用搜索字段中提供的筛选条件来确定搜索范围。

查找共享 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 AMI。
3. 在第一个筛选条件中，选择以下选项之一：
   - 私有映像 – 列出共享给您的所有 AMI。
   - 公有映像 – 列出所有公有 AMI。
4. （可选）要仅显示 Amazon 提供的公有映像，请选择搜索字段，然后从菜单选项中选择所有者别名，再选择 =，然后选择 amazon。
5. （可选）添加筛选条件，将搜索范围限定为符合您要求的 AMI。

查找来自经过验证的提供商的共享公共 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI Catalog（AMI 目录）。
3. 选择社区 AMI。
4. 在优化结果窗格中，选择经过验证的提供商。经过验证的提供商标签指示相关 AMI 是由 Amazon 或经过验证的合作伙伴提供。

Amazon CLI

使用 describe-images 命令列出 AMI。可以将该列表范围确定为所需的 AMI 类型，如以下示例所示。

列出所有公用 AMI

以下命令将列出所有公用 AMI，包括您拥有的所有公用 AMI。

```
aws ec2 describe-images --executable-users all
```

列出拥有显式启动许可的 AMI

以下命令列出您对其拥有显式启动许可的 AMI。此列表不包括您拥有的任何 AMI。

```
aws ec2 describe-images --executable-users self
```

列出经过验证的提供商拥有的 AMI

以下命令将列出经过验证的提供商拥有的 AMI。经过验证的提供商（Amazon 或经过验证的合作伙伴）拥有的公共 AMI 具有一个别名拥有者，它将在账户字段中显示为 amazon、aws-backup-

`vault` 或 `aws-marketplace`。这可帮助您轻松查找来自经过验证的提供商的 AMI。其他用户不能对其 AMI 使用别名。

```
aws ec2 describe-images \
    --owners amazon aws-marketplace \
    --query 'Images[*].[ImageId]' \
    --output text
```

列出账户拥有的 AMI

以下命令列出指定 Amazon Web Services 账户 拥有的 AMI。

```
aws ec2 describe-images --owners 123456789012
```

使用筛选条件确定 AMI 的范围

要减少显示的 AMI 数量，请使用筛选条件只列出您感兴趣的 AMI 类型。例如，使用以下筛选条件可以只显示 EBS 支持的 AMI。

```
--filters "Name=root-device-type,Values=ebs"
```

PowerShell

使用 Get-EC2Image cmdlet（Tools for Windows PowerShell）列出 AMI。可以将该列表范围确定为所需的 AMI 类型，如以下示例所示。

列出所有公用 AMI

以下命令将列出所有公用 AMI，包括您拥有的所有公用 AMI。

```
Get-EC2Image -ExecutableUser all
```

列出拥有显式启动许可的 AMI

以下命令列出您对其拥有显式启动许可的 AMI。此列表不包括您拥有的任何 AMI。

```
Get-EC2Image -ExecutableUser self
```

列出经过验证的提供商拥有的 AMI

以下命令将列出经过验证的提供商拥有的 AMI。经过验证的提供商（Amazon 或经过验证的合作伙伴）拥有的公共 AMI 具有一个别名拥有者，它将在账户字段中显示为 amazon、aws-backup-vault 或 aws-marketplace。这可帮助您轻松查找来自经过验证的提供商的 AMI。其他用户不能对其 AMI 使用别名。

```
Get-EC2Image -Owner amazon aws-marketplace
```

列出账户拥有的 AMI

以下命令列出指定 Amazon Web Services 账户 拥有的 AMI。

```
Get-EC2Image -Owner 123456789012
```

使用筛选条件确定 AMI 的范围

要减少显示的 AMI 数量，请使用筛选条件只列出您感兴趣的 AMI 类型。例如，使用以下筛选条件可以只显示 EBS 支持的 AMI。

```
-Filter @{Name="root-device-type"; Values="ebs"}
```

准备使用适用于 Linux 的共享 AMI

使用适用于 Linux 的共享 AMI 之前，应执行以下步骤以确认没有预安装凭证允许第三方对您的实例进行不希望的访问，并且没有可能将敏感数据传输给第三方的预配置远程登录。查看 AMI 使用的 Linux 发行版的文档以了解有关提高系统安全性的信息。

为了确保您不会在无意中丢失对您的实例的访问，我们建议您启动两个 SSH 会话并将第二个会话保持为打开状态，直到您删除了无法识别的凭证并确认您仍可以使用 SSH 登录您的实例。

1. 标识并禁用任何未经授权的公有 SSH 密钥。该文件中的唯一密钥应是您用于启动 AMI 的密钥。以下命令查找 authorized_keys 文件：

   ```
   [ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
   ```

2. 对根用户禁用基于密码的身份验证。打开 sshd_configPermitRootLogin 文件并编辑 行，如下所示：

   ```
   PermitRootLogin without-password
   ```

或者，您可以禁用以根用户身份登录实例的功能：

```
PermitRootLogin No
```

重启 sshd 服务。

3. 检查是否有任何其他用户能够登录您的实例。具有超级用户权限的用户尤为危险。删除或锁定任何未知账户的密码。
4. 检查打开的端口以确认您未在使用和运行侦听传入连接的网络服务。
5. 要防止预配置的远程登录，应删除现有配置文件并重启 `rsyslog` 服务。例如：

```
[ec2-user ~]$ sudo rm /etc/rsyslog.conf
[ec2-user ~]$ sudo service rsyslog restart
```

6. 验证所有 cron 作业的合法性。

如果您发现了认为存在安全风险的公用 AMI，请联系 Amazon 安全团队。有关更多信息，请参阅 Amazon 安全中心。

使用“允许的 AMI”控制在 Amazon EC2 中对 AMI 的发现和使用

要控制 Amazon Web Services 账户中的用户对亚马逊机器映像（AMI）的发现和使用，可以使用允许的 AMI 功能。您可指定 AMI 要在账户中可见和可用而必须满足的标准。启用该标准后，启动实例的用户将只能看到并有权访问符合指定标准的 AMI。例如，您可以指定可信的 AMI 提供商列表作为标准，只有来自这些提供商的 AMI 才可见并可供使用。

在启用“允许的 AMI”设置之前，您可以启用审核模式来预览哪些 AMI 将可见或不可见、可供使用或不可供使用。这有助于您根据需要完善标准，确保只有预期的 AMI 才对账户中的用户可见和可用。此外，使用 describe-instance-image-metadata 命令可查找使用不符合指定标准的 AMI 启动的实例。这些信息可以指导您作出决策，选择是更新启动配置以使用合规的 AMI（例如在启动模板中指定不同的 AMI），还是调整标准来允许这些 AMI。

您可以在账户级别指定“允许的 AMI”设置，可以直接在账户中指定，也可以使用声明式策略进行指定。必须在想要在其中控制 AMI 使用的每个 Amazon Web Services 区域中配置这些设置。使用声明式策略允许同时将设置应用于多个区域，也可以同时应用于多个账户。当使用声明式策略时，您无法直接在账户中修改设置。本主题介绍如何直接在账户中配置设置。有关使用声明式策略的信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

> **Note**
>
> “允许的 AMI”功能仅控制公用 AMI 或与账户共享的 AMI 的发现和使用。它不会限制账户拥有的 AMI。无论设置的条件是什么，您账户中的用户始终可以发现和使用由您的账户创建的 AMI。

“允许的 AMI”功能的主要优势

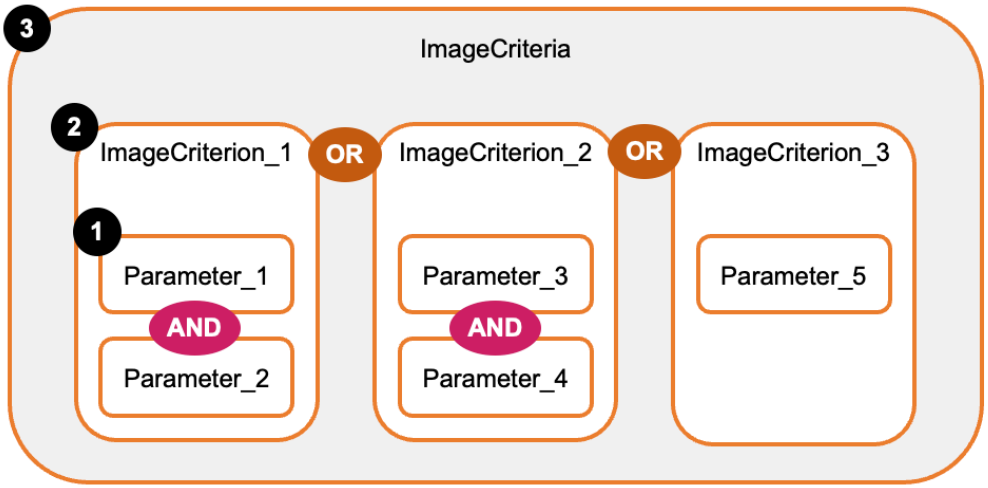
- 合规与安全：用户只能发现和使用符合指定标准的 AMI，从而降低使用不合规的 AMI 的风险。
- 高效管理：通过减少允许的 AMI 数量，管理其余 AMI 将变得更加轻松高效。
- 集中式账户级别实施：直接在账户内或通过声明式策略在账户级别配置“允许的 AMI”设置。这提供了一种集中而高效的方式来控制整个账户的 AMI 使用情况。

内容

- “允许的 AMI”的工作原理
- 实施“允许的 AMI”的最佳实践
- 所需的 IAM 权限
- 管理“允许的 AMI”设置

“允许的 AMI”的工作原理

为了控制哪些 AMI 可以在您的账户中发现和使用，您可以定义一组用于评估 AMI 的标准。这些标准由一个或多个 `ImageCriterion` 组成，如下图所示。图后附有说明。



配置分为三个级别：

- 1：参数值
  - 多值参数：
    - `ImageProviders`
    - `ImageNames`
    - `MarketplaceProductCodes`

      AMI 匹配参数中的任意值即获得允许。

      示例：`ImageProviders` = amazon OR 账户 111122223333 OR 账户 444455556666（参数值的评估逻辑未在图表中显示。）
  - 单值参数：
    - `CreationDateCondition`
    - `DeprecationTimeCondition`
- 2：`ImageCriterion`
  - 使用 AND 逻辑对多个参数进行分组。
  - AMI 必须匹配 `ImageCriterion` 中的所有参数才能获得允许。
  - 示例：`ImageProviders` = amazon AND `CreationDateCondition` = 300 天或更短
- 3：`ImageCriteria`
  - 使用 OR 逻辑将多个 `ImageCriterion` 组合在一起。
  - AMI 匹配任何 `ImageCriterion` 即获得允许。
  - 形成评估 AMI 所依据的完整配置。

主题

- 允许的 AMI 参数
- 允许的 AMI 配置
- 如何评估标准
- 限制
- “允许的 AMI”操作

允许的 AMI 参数

可配置以下参数用于创建 `ImageCriterion`：

`ImageProviders`

允许其 AMI 的 AMI 提供商。

有效值是由 Amazon 和 Amazon Web Services 账户 ID 定义的别名，如下所示：

- `amazon`：别名，用于标识由 Amazon 或经过验证的提供商创建的 AMI
- `aws-marketplace`：别名，用于标识由 Amazon Web Services Marketplace 中经过验证的提供商创建的 AMI
- `aws-backup-vault`：别名，用于标识位于逻辑上受物理隔离的 Amazon Backup 保管库账户中的备份 AMI。如果您使用 Amazon Backup 逻辑上受物理隔离的保管库功能，请确保将此别名作为 AMI 提供商包含在内。
- Amazon Web Services 账户 ID：一个或多个 12 位 Amazon Web Services 账户 ID
- `none`：表示只能发现和使用您的账户创建的 AMI。无法发现和使用公有或共享的 AMI。指定后将无法指定其他标准。

`ImageNames`

允许的 AMI 名称，使用精确匹配或通配符（？或 *）。

`MarketplaceProductCodes`

允许的 AMI Amazon Web Services Marketplace 的产品代码。

`CreationDateCondition`

允许的 AMI 的最大期限。

`DeprecationTimeCondition`

自弃用允许的 AMI 以来的最长期限。

有关每个标准的有效值和约束，请参阅《Amazon EC2 API Reference》中的 ImageCriterionRequest。

允许的 AMI 配置

“允许的 AMI”的核心配置是定义允许的 AMI 标准的 `ImageCriteria` 配置。以下 JSON 结构显示可以指定的参数：

```
{
    "State": "enabled" | "disabled" | "audit-mode",
    "ImageCriteria" : [
        {
```

```
            "ImageProviders": ["string",...],
            "MarketplaceProductCodes": ["string",...],
            "ImageNames":["string",...],
            "CreationDateCondition" : {
                "MaximumDaysSinceCreated": integer
            },
            "DeprecationTimeCondition" : {
                "MaximumDaysSinceDeprecated": integer
            }
         },
         ...
}
```

ImageCriteria 示例

以下 ImageCriteria 示例配置四个 ImageCriterion。如果 AMI 与其中任何一个 ImageCriterion 匹配，则允许使用。有关如何评估标准的信息，请参阅如何评估标准。

```
{
    "ImageCriteria": [
        // ImageCriterion 1: Allow Amazon Web Services Marketplace AMIs with product
 code "abcdefg1234567890"
        {
            "MarketplaceProductCodes": [
                "abcdefg1234567890"
            ]
        },
        // ImageCriterion 2: Allow AMIs from providers whose accounts are
        // "123456789012" OR "123456789013" AND AMI age is less than 300 days
        {
            "ImageProviders": [
                "123456789012",
                "123456789013"
            ],
            "CreationDateCondition": {
                "MaximumDaysSinceCreated": 300
            }
        },
        // ImageCriterion 3: Allow AMIs from provider whose account is "123456789014"
        // AND with names following the pattern "golden-ami-*"
        {
            "ImageProviders": [
                "123456789014"
```

```
            ],
            "ImageNames": [
                "golden-ami-*"
            ]
        },
        // ImageCriterion 4: Allow AMIs from Amazon or verified providers
        // AND which aren't deprecated
        {
            "ImageProviders": [
                "amazon"
            ],
            "DeprecationTimeCondition": {
                "MaximumDaysSinceDeprecated": 0
            }
        }
    ]
}
```

如何评估标准

下表说明确定是否允许 AMI 的评估规则，并显示在每个级别如何应用 AND 或 OR 运算符：

评估级别	运算符	成为允许的 AMI 的要求
ImageProviders 、ImageNames 和 MarketplaceProductCodes 的参数值	OR	AMI 必须匹配每个参数列表中至少一个值
ImageCriterion	AND	AMI 必须匹配每个 ImageCriterion 中的所有参数
ImageCriteria	OR	AMI 必须匹配任何一个 ImageCriterion

使用上述评估规则，让我们看看如何将其应用于 ImageCriteria 示例：

- ImageCriterion 1：允许具有 Amazon Web Services Marketplace 产品代码 abcdefg1234567890 的 AMI

OR

- ImageCriterion 2：允许同时符合以下两个标准的 AMI：
  - 由账户 123456789012 OR 123456789013 拥有
    - AND
  - 已在过去 300 天内创建

OR

- ImageCriterion 3：允许同时符合以下两个标准的 AMI：
  - 由账户 123456789014 拥有
    - AND
  - 以 golden-ami-* 模式命名

OR

- ImageCriterion 4：允许同时符合以下两个标准的 AMI：
  - 由 Amazon 发布或经过验证的提供商（通过 amazon 别名指定）
    - AND
  - 未弃用（自弃用以来的最长天数为 0）

限制

ImageCriteria 最多可以包含：

- 10 ImageCriterion

每个 ImageCriterion 最多可以包含：

- 200 个 ImageProviders 值
- 50 个 ImageNames 值
- 50 个 MarketplaceProductCodes 值

限制示例

使用前面的 ImageCriteria 示例：

- 有 4 个 ImageCriterion。最多可以向请求再添加 6 个，达到 10 个的上限。
- 在第一个 ImageCriterion 中，有 1 个 MarketplaceProductCodes 值。最多可以向此 ImageCriterion 再添加 49 个值，达到 50 个值的上限。
- 在第二个 ImageCriterion 中，有 2 个 ImageProviders 值。最多可以向此 ImageCriterion 再添加 198 个值，达到 200 个值的上限。
- 在第三个 ImageCriterion 中，有 1 个 ImageNames 值。最多可以向此 ImageCriterion 再添加 49 个值，达到 50 个值的上限。

“允许的 AMI”操作

“允许的 AMI”功能有三种用于管理映像标准的运行状态：启用、禁用和审核模式。您可借助此功能启用或禁用镜像标准，或根据需要对其进行审核。

已启用

启用“允许的 AMI”时：

- 将应用 ImageCriteria。
- 只有允许的 AMI 才能在 EC2 控制台中由使用镜像（例如描述、复制、存储或执行其他使用镜像的操作）的 API 发现。
- 只能使用允许的 AMI 启动实例。

已禁用

禁用“允许的 AMI”时：

- 不会应用 ImageCriteria。
- 对 AMI 的可发现性或使用不会施加任何限制。

审核模式

在审核模式下：

- 将应用 ImageCriteria，但对 AMI 的可发现性或使用不会施加任何限制。
- 在 EC2 控制台中，对于每个 AMI，允许的镜像字段会显示是或否，指示在启用“允许的 AMI”时，账户中的用户是否可以发现和使用该 AMI。

- 在命令行中，`describe-image` 操作的响应包括 `"ImageAllowed": true` 或 `"ImageAllowed": false`，指示在启用“允许的 AMI”时，账户中的用户是否可以发现和使用该 AMI。
- 在 EC2 控制台中，AMI 目录会在当启用“允许的 AMI”时账户中的用户无法发现或无法使用的 AMI 旁显示不允许。

实施“允许的 AMI”的最佳实践

在实施“允许的 AMI”时，请考虑这些最佳实践，确保平稳过渡并最大限度地减少对 Amazon 环境的潜在干扰。

1. 启用审核模式

   首先在审核模式下启用“允许的 AMI”。此状态允许您查看哪些 AMI 将受到标准的影响，而无需实际限制访问权限，从而提供无风险的评估期。

2. 设置“允许的 AMI”标准

   仔细确定哪些 AMI 提供商符合贵组织的安全政策、合规性要求和运营需求。

   > **Note**
   >
   > 在使用 Amazon ECS 或 Amazon EKS 或 Amazon Lambda 托管实例等 Amazon 托管服务时，我们建议指定 `amazon` 别名以允许 Amazon 创建的 AMI。这些服务依赖于 Amazon 发布的 AMI 来启动实例。
   >
   > 为任何 AMI 设置 `CreationDateCondition` 限制时均需谨慎。如果设置过于严格的日期条件（例如，AMI 必须少于 5 天），则 AMI（无论是来自 Amazon 还是其他提供商）未在指定的时间范围内更新可能会导致实例启动失败。
   >
   > 我们建议将 `ImageNames` 与 `ImageProviders` 搭配使用，以获得更好的控制和更高的精确度。单独使用 `ImageNames` 可能无法唯一标识 AMI。

3. 检查对预期业务流程的影响

   您可以通过控制台或 CLI 来识别任何使用不符合指定标准的 AMI 启动的实例。这些信息可以指导您作出决策，选择是更新启动配置以使用合规的 AMI（例如在启动模板中指定不同的 AMI），还是调整标准来允许这些 AMI。

   控制台：使用 ec2-instance-launched-with-allowed-ami Amazon Config 规则来检查正在运行或已停止的实例是否使用符合“允许的 AMI”标准的 AMI 启动。如果 AMI 不符合“允许的 AMI”标准，则规则

为 NON_COMPLIANT；如果符合，则规则为 COMPLIANT。仅当“允许的 AMI”设置为已启用或审核模式时，规则才会生效。

CLI：运行 describe-instance-image-metadata 命令并筛选响应，以识别任何使用不符合指定标准的 AMI 启动的实例。

有关控制台和 CLI 说明，请参阅 查找利用不允许的 AMI 启动的实例。

4. 启用“允许的 AMI”

一旦确认该标准不会对预期的业务流程产生不利影响，请启用“允许的 AMI”。

5. 监控实例启动情况

继续监控 AMI 在应用程序和您使用的 Amazon 托管服务（例如 Amazon EMR、Amazon ECR、Amazon EKS 和 Amazon Elastic Beanstalk）中启动实例的情况。检查是否存在任何意外问题，并对“允许的 AMI”标准进行必要的调整。

6. 试用新 AMI

要测试不符合您当前“允许的 AMI”设置的第三方 AMI，Amazon 建议采用以下方法：

- 使用单独的 Amazon Web Services 账户：创建一个无法访问您关键业务资源的账户。确保未在此账户中启用“允许的 AMI”设置，或明确允许您要测试的 AMI，以便您可以对其进行测试。
- 在其他 Amazon Web Services 区域进行测试：使用可使用第三方 AMI 但尚未启用“允许的 AMI”设置的区域。

这些方法有助于确保在测试新 AMI 时，您的关键业务资源仍然安全。

所需的 IAM 权限

要使用“允许的 AMI”功能，您需要以下 IAM 权限：

- `GetAllowedImagesSettings`
- `EnableAllowedImagesSettings`
- `DisableAllowedImagesSettings`
- `ReplaceImageCriteriaInAllowedImagesSettings`

管理“允许的 AMI”设置

您可以管理“允许的 AMI”设置 这些设置按区域和账户生效。

任务

- 启用“允许的 AMI”
- 设置“允许的 AMI”条件
- 禁用“允许的 AMI”
- 获取“允许的 AMI”条件
- 查找允许的 AMI
- 查找利用不允许的 AMI 启动的实例

启用“允许的 AMI”

您可以启用“允许的 AMI”并指定“允许的 AMI”条件。我们建议您首先在审计模式中进行此操作，因为此模式可在不实际限制访问权限的情况下显示会受到筛选条件影响的具体 AMI。

Console

启用“允许的 AMI”

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡的设置下，选择允许的 AMI。
4. 在允许的 AMI 选项卡上，选择管理。
5. 对于允许的 AMI 设置，选择审计模式或启用。我们建议您首先在审计模式中对筛选条件进行测试，然后再返回此步骤启用“允许的 AMI”。
6. （可选）对于 AMI 条件，请输入 JSON 格式的条件。
7. 选择更新。

Amazon CLI

启用“允许的 AMI”

使用 enable-allowed-images-settings 命令。

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state enabled
```

要改为启用审计模式，请指定 `audit-mode`，而不是 `enabled`。

```
aws ec2 enable-allowed-images-settings --allowed-images-settings-state audit-mode
```

PowerShell

启用“允许的 AMI”

使用 Enable-EC2AllowedImagesSetting cmdlet。

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState enabled
```

要改为启用审计模式，请指定 `audit-mode`，而不是 `enabled`。

```
Enable-EC2AllowedImagesSetting -AllowedImagesSettingsState audit-mode
```

设置“允许的 AMI”条件

启用“允许的 AMI”后，您可以设置或替换“允许的 AMI”条件。

有关正确的配置和有效值，请参阅允许的 AMI 配置和允许的 AMI 参数。

Console

设置“允许的 AMI”条件

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡的设置下，选择允许的 AMI。
4. 在允许的 AMI 选项卡上，选择管理。
5. 对于 AMI 条件，请输入 JSON 格式的条件。
6. 选择更新。

Amazon CLI

设置“允许的 AMI”条件

使用 replace-image-criteria-in-allowed-images-settings 命令并指定包含允许的 AMI 标准的 JSON 文件。

```
aws ec2 replace-image-criteria-in-allowed-images-settings --cli-input-json
 file://file_name.json
```

PowerShell

设置“允许的 AMI”条件

使用 Set-EC2ImageCriteriaInAllowedImagesSetting cmdlet 并指定包含允许的 AMI 标准的 JSON 文件。

```
$imageCriteria = Get-Content -Path .\file_name.json | ConvertFrom-Json
Set-EC2ImageCriteriaInAllowedImagesSetting -ImageCriterion $imageCriteria
```

禁用“允许的 AMI”

您可以按如下所示禁用“允许的 AMI”。

Console

禁用“允许的 AMI”

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡的设置下，选择允许的 AMI。
4. 在允许的 AMI 选项卡上，选择管理。
5. 对于“允许的 AMI”设置，选择禁用。
6. 选择更新。

Amazon CLI

禁用“允许的 AMI”

使用 disable-allowed-images-settings 命令。

```
aws ec2 disable-allowed-images-settings
```

PowerShell

禁用“允许的 AMI”

使用 Disable-EC2AllowedImagesSetting cmdlet。

```
Disable-EC2AllowedImagesSetting
```

获取“允许的 AMI”条件

您可以获取当前“允许的 AMI”设置状态和“允许的 AMI”条件。

Console

获取“允许的 AMI”状态和条件

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡的设置下，选择允许的 AMI。
4. 在允许的 AMI 选项卡上，将允许的 AMI 设置为启用、禁用或审计模式。
5. 如果“允许的 AMI”设置状态为启用或审计模式，则 AMI 条件将以 JSON 格式显示 AMI 条件。

Amazon CLI

获取“允许的 AMI”状态和条件

使用 get-allowed-images-settings 命令。

```
aws ec2 get-allowed-images-settings
```

在以下示例输出中，状态为 `audit-mode`，且已在账户中设置映像标准。

```
{
    "State": "audit-mode",
    "ImageCriteria": [
        {
            "MarketplaceProductCodes": [
                "abcdefg1234567890"
            ]
        },
        {
            "ImageProviders": [
                "123456789012",
```

```
                "123456789013"
            ],
            "CreationDateCondition": {
                "MaximumDaysSinceCreated": 300
            }
        },
        {
            "ImageProviders": [
                "123456789014"
            ],
            "ImageNames": [
                "golden-ami-*"
            ]
        },
        {
            "ImageProviders": [
                "amazon"
            ],
            "DeprecationTimeCondition": {
                "MaximumDaysSinceDeprecated": 0
            }
        }
    ],
    "ManagedBy": "account"
}
```

PowerShell

获取“允许的 AMI”状态和条件

使用 Get-EC2AllowedImagesSetting cmdlet。

```
Get-EC2AllowedImagesSetting | Select-Object `
    State, `
    ManagedBy, `
    @{Name='ImageProviders'; Expression={($_.ImageCriteria.ImageProviders)}}, `
    @{Name='MarketplaceProductCodes';
 Expression={($_.ImageCriteria.MarketplaceProductCodes)}}, `
    @{Name='ImageNames'; Expression={($_.ImageCriteria.ImageNames)}}, `
    @{Name='MaximumDaysSinceCreated';
 Expression={($_.ImageCriteria.CreationDateCondition.MaximumDaysSinceCreated)}}, `
    @{Name='MaximumDaysSinceDeprecated';
 Expression={($_.ImageCriteria.DeprecationTimeCondition.MaximumDaysSinceDeprecated)}}
```

在以下示例输出中，状态为 audit-mode，且已在账户中设置映像标准。

```
State      : audit-mode
ManagedBy  : account
ImageProviders            : {123456789012, 123456789013, 123456789014, amazon}
MarketplaceProductCodes   : {abcdefg1234567890}
ImageNames                : {golden-ami-*}
MaximumDaysSinceCreated   : 300
MaximumDaysSinceDeprecated: 0
```

查找允许的 AMI

您可以查找当前“允许的 AMI”条件允许或不允许的 AMI。

> **Note**
>
> “允许的 AMI”设置必须为“审计模式”。

Console

检查 AMI 是否符合“允许的 AMI”条件

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡（如果选中了复选框）或摘要区域（如果选择了 AMI ID）中，找到允许的镜像字段。
   - 是 – 此 AMI 符合“允许的 AMI”条件。启用“允许的 AMI”后，您账户中的用户将可使用此 AMI。
   - 否 – 此 AMI 不符合“允许的 AMI”条件。
5. 在导航窗格中，选择 AMI Catalog（AMI 目录）。

   标记为不允许的 AMI 表示 AMI 不符合“允许的 AMI”标准。禁用“允许的 AMI”时，此 AMI 将对您账户中的用户不可见且不可用。

Amazon CLI

检查 AMI 是否符合“允许的 AMI”条件

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[].ImageAllowed \
    --output text
```

下面是示例输出。

```
True
```

查找符合“允许的 AMI”条件的 AMI

使用 describe-images 命令。

```
aws ec2 describe-images \
    --filters "Name=image-allowed,Values=true" \
    --max-items 10 \
    --query Images[].ImageId
```

下面是示例输出。

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

PowerShell

检查 AMI 是否符合“允许的 AMI”条件

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).ImageAllowed
```

下面是示例输出。

```
True
```

查找符合“允许的 AMI”条件的 AMI

使用 Get-EC2Image cmdlet。

```
Get-EC2Image `
    -Filter @{Name="image-allows";Values="true"} `
    -MaxResult 10 | `
    Select ImageId
```

下面是示例输出。

```
ami-000eaaa8be2fd162a
ami-000f82db25e50de8e
ami-000fc21eb34c7a9a6
ami-0010b876f1287d7be
ami-0010b929226fe8eba
ami-0010957836340aead
ami-00112c992a47ba871
ami-00111759e194abcc1
ami-001112565ffcafa5e
ami-0011e45aaee9fba88
```

查找利用不允许的 AMI 启动的实例

您可以识别使用不符合“允许的 AMI”条件的 AMI 启动的实例。

Console

检查某个实例是否是使用不允许的 AMI 启动的

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。

4. 在详细信息选项卡的实例详细信息下，找到允许的映像字段。

   - 是 – 此 AMI 符合“允许的 AMI”条件。
   - 否 – 此 AMI 不符合“允许的 AMI”条件。

Amazon CLI

查找使用不允许的 AMI 启动的实例

使用 describe-instance-image-metadata 命令和 `image-allowed` 筛选条件。

```
aws ec2 describe-instance-image-metadata \
    --filters "Name=image-allowed,Values=false" \
    --query "InstanceImageMetadata[*].[InstanceId,ImageMetadata.ImageId]" \
    --output table
```

下面是示例输出。

```
-----------------------------------------------------
|          DescribeInstanceImageMetadata          |
+----------------------+--------------------------+
|  i-08fd74f3f1595fdbd |  ami-09245d5773578a1d6   |
|  i-0b1bf24fd4f297ab9 |  ami-07cccf2bd80ed467f   |
|  i-026a2eb590b4f7234 |  ami-0c0ec0a3a3a4c34c0   |
|  i-006a6a4e8870c828f |  ami-0a70b9d193ae8a799   |
|  i-0781e91cfeca3179d |  ami-00c257e12d6828491   |
|  i-02b631e2a6ae7c2d9 |  ami-0bfddf4206f1fa7b9   |
+----------------------+--------------------------+
```

PowerShell

查找使用不允许的 AMI 启动的实例

使用 Get-EC2InstanceImageMetadata cmdlet。

```
Get-EC2InstanceImageMetadata `
    -Filter @{Name="image-allowed";Values="false"} | `
    Select InstanceId, @{Name='ImageId'; Expression={($_.ImageMetadata.ImageId)}}
```

下面是示例输出。

```
InstanceId          ImageId
----------          -------
i-08fd74f3f1595fdbd ami-09245d5773578a1d6
i-0b1bf24fd4f297ab9 ami-07cccf2bd80ed467f
i-026a2eb590b4f7234 ami-0c0ec0a3a3a4c34c0
i-006a6a4e8870c828f ami-0a70b9d193ae8a799
i-0781e91cfeca3179d ami-00c257e12d6828491
i-02b631e2a6ae7c2d9 ami-0bfddf4206f1fa7b9
```

Amazon Config

您可以添加 ec2-instance-launched-with-allowed-ami Amazon Config 规则，根据自己的要求对其进行配置，然后用来评估您的实例。

有关更多信息，请参阅 Amazon Config 开发人员指南中的添加 Amazon Config 规则和 ec2-instance-launched-with-allowed-ami。

将您的 AMI 设为可在 Amazon EC2 中公开使用

您可以通过向所有 Amazon Web Services 账户 共享来公开您的 AMI。

如果您想阻止公开共享您的 AMI，可以启用阻止 AMI 的公开访问。这可以阻止任何公开 AMI 的尝试，从而有助于防止未经授权的访问和对 AMI 数据的潜在滥用。请注意，启用阻止公开访问不会影响已经公开提供的 AMI；它们仍然公开可用。有关更多信息，请参阅 了解 AMI 的屏蔽公共访问。

要仅允许特定账户使用您的 AMI 启动实例，请参阅 与特定 Amazon 账户共享 AMI。

内容

- 注意事项
- 与所有 Amazon 账户共享 AMI（公开共享）

注意事项

在将 AMI 设为公用之前，请考虑以下事项。

- 所有权 – 若要将 AMI 设为公有，您的 Amazon Web Services 账户 必须拥有 AMI。
- 区域 – AMI 是一种区域性资源。当您共享 AMI 时，则它只能在您共享该 AMI 的区域使用。要使 AMI 能够在其他区域使用，请将该 AMI 复制到该区域并进行共享。有关更多信息，请参阅 复制 Amazon EC2 AMI。

- 阻止公开访问 – 要公开共享 AMI，必须在 AMI 公开共享的每个区域禁用阻止 AMI 的公开访问。公开共享 AMI 后，您可以重新启用阻止 AMI 的公开访问功能，以防止进一步公开共享您的 AMI。
- 有些 AMI 无法设为公共 – 如果您的 AMI 包含以下组件之一，则无法将其设为公共（但您可以将 AMI 与特定 Amazon Web Services 账户 账户共享）：
  - 加密卷
  - 加密卷的快照
  - 产品代码
- 避免泄露敏感数据 – 要避免在共享 AMI 时泄露敏感数据，请阅读 创建共享 Linux AMI 的建议 中的安全注意事项并遵循建议的操作。
- 使用 – 当您共享 AMI 时，用户只能从该 AMI 启动实例。他们无法删除、共享或修改实例。但是，在他们使用您的 AMI 启动实例后，他们可以从其启动的实例创建 AMI。
- 自动弃用 – 默认情况下，所有公有 AMI 的弃用日期均设置为自 AMI 创建日期起 2 年。您可以将弃用日期设置为早于两年。要取消弃用日期，或将弃用移至未来某一日期，您必须通过仅将 AMI 与特定 Amazon Web Services 账户 共享来将其设为私有。
- 删除过时的 AMI – 在公共 AMI 达到其弃用日期后，如果在六个月或更长时间内没有从 AMI 启动任何新实例，则 Amazon 最终会删除该公共共享属性，这样过时的 AMI 就不会出现在公共 AMI 列表中。
- 账单 – 当其他 Amazon Web Services 账户 使用您的 AMI 启动实例时，您无需付费。使用 AMI 启动实例的账户将为它们启动的实例付费。

与所有 Amazon 账户共享 AMI（公开共享）

将 AMI 设为公开后，它将在控制台的社区 AMI 中可用，您可以从 EC2 控制台左侧导航器的 AMI 目录中访问此 AMI，或者在使用控制台启动实例时访问该 AMI。请注意，将某个 AMI 设置为公用之后，可能需要一点时间 Community AMIs 中才会显示该 AMI。

Console

将 AMI 设为公用

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 从列表中选择您的 AMI，然后选择 Actions（操作）、Edit AMI permissions（编辑 AMI 权限）。
4. 在 AMI 可用性下，选择公开。
5. 选择保存更改。

Amazon CLI

每个 AMI 都有一个 launchPermission 属性，用于控制允许哪些 Amazon Web Services 账户（包括拥有者账户在内）使用该 AMI 启动实例。通过修改 AMI 的 launchPermission 属性，可以将该 AMI 设为公共（这会向所有 Amazon Web Services 账户 授予启动权限），或仅将其与您指定的 Amazon Web Services 账户 共享。

可以在具有 AMI 启动许可的账户的列表中添加或删除账户 ID。要将 AMI 设为公有，请指定 all 组。公用和显式启动许可都可以指定。

将 AMI 设为公用

1. 使用 modify-image-attribute 命令可以将 all 组添加到指定 AMI 的 launchPermission 列表中，如下所示。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Add=[{Group=all}]"
```

2. 要验证 AMI 的启动权限，请使用 describe-image-attribute 命令。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

3. （可选）要再次将 AMI 设为私有，请从其启动许可中删除 all 组。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Remove=[{Group=all}]"
```

PowerShell

每个 AMI 都有一个 launchPermission 属性，用于控制允许哪些 Amazon Web Services 账户（包括拥有者账户在内）使用该 AMI 启动实例。通过修改 AMI 的 launchPermission 属性，可以将该 AMI 设为公共（这会向所有 Amazon Web Services 账户 授予启动权限），或仅将其与您指定的 Amazon Web Services 账户 共享。

可以在具有 AMI 启动许可的账户的列表中添加或删除账户 ID。要将 AMI 设为公有，请指定 `all` 组。公用和显式启动许可都可以指定。

将 AMI 设为公用

1. 使用 Edit-EC2ImageAttribute 命令可以将 `all` 组添加到指定 AMI 的 `launchPermission` 列表中，如下所示。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserGroup all
```

2. 要验证 AMI 的启动权限，请使用以下 Get-EC2ImageAttribute 命令。

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

3. （可选）要再次将 AMI 设为私有，请从其启动许可中删除 `all` 组。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -UserGroup all
```

了解 AMI 的屏蔽公共访问

要防止公开共享您的 AMI，请在账户级别启用屏蔽对 AMI 的公共访问。

启用阻止公开访问后，任何将 AMI 设为公开的尝试都会被自动阻止。但是，如果您已经有公有 AMI，则其将保持公开可用。

要公开共享 AMI，您必须禁用阻止公开访问。共享完成后，最佳实践是重新启用阻止公开访问功能，以防止意外地公开共享您的 AMI。

> **Note**
>
> 此设置是在账户级别配置，可以直接在账户中配置，也可以使用声明式策略进行配置。您必须在要防止公开共享 AMI 的每个 Amazon Web Services 区域中配置该功能。使用声明式策略可同时将设置应用于多个区域，也可以同时应用于多个账户。当使用声明式策略时，您无法直接在账户中修改设置。本主题介绍如何直接在账户中配置设置。有关使用声明式策略的信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

您可以将 IAM 权限限制为管理员用户，这样只有此类用户可启用或禁用阻止 AMI 的公开访问。

主题

- 默认设置
- 管理 AMI 的屏蔽公共访问设置

默认设置

根据您的账户是新账户还是现有账户，以及您是否拥有公用 AMI，默认启用或禁用阻止公开访问 AMI 设置。下表列出了默认设置：

Amazon 账户	“阻止公开访问 AMI”默认设置
新账户	已启用
没有公用 AMI 的现有账户[1]	已启用
有一个或多个公用 AMI 的现有账户[1]	已禁用

[1]如果您的账户在 2023 年 7 月 15 日当天或之后拥有一个或多个公用 AMI，那么即使您随后将所有 AMI 设为私有，系统仍会为您的账户默认禁用阻止公开访问 AMI。

管理 AMI 的屏蔽公共访问设置

您可以管理 AMI 的屏蔽公共访问设置，以控制其是否可以公开共享。您可以使用 Amazon EC2 控制台或 Amazon CLI 来启用、禁用或查看 AMI 的当前屏蔽公共访问状态。

查看 AMI 的阻止公共访问状态

要查看您的账户中是否禁止公开共享 AMI，您可以查看 AMI 的阻止公开访问状态。您必须在每个希望了解是否阻止 AMI 公开共享的 Amazon Web Services 区域 中查看此状态。

所需的权限

要获取 AMI 的当前屏蔽公共访问设置，您必须拥有 `GetImageBlockPublicAccessState` IAM 权限。

Console

查看指定区域中 AMI 的阻止公开访问状态

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航栏（位于屏幕顶部）中，选择查看 AMI 的阻止公开访问状态的区域。
3. 在导航窗格中，选择控制面板。
4. 在账户属性卡的设置下，选择数据保护和安全。
5. 在阻止公开访问 AMI 下，选中公开访问字段。值为已阻止新的公开共享或已允许新的公开共享。

Amazon CLI

要获取 AMI 的阻止公开访问状态

使用 get-image-block-public-access-state 命令。该值为 `block-new-sharing` 或 `unblocked`。

示例：对于特定区域

```
aws ec2 get-image-block-public-access-state --region us-east-1
```

`ManagedBy` 字段表示配置了该设置的实体。在本例中，`account` 表示是直接在账户中配置的设置。值为 `declarative-policy` 表示该设置是由声明式策略所配置。有关更多信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

```
{
    "ImageBlockPublicAccessState": "block-new-sharing",
    "ManagedBy": "account"
}
```

示例：对于账户中的所有区域

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-image-block-public-access-state \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

下面是示例输出。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       unblocked
eu-west-3        block-new-sharing
...
```

PowerShell

要获取 AMI 的阻止公开访问状态

使用 Get-EC2ImageBlockPublicAccessState cmdlet。该值为 `block-new-sharing` 或 `unblocked`。

示例：对于特定区域

```
Get-EC2ImageBlockPublicAccessState -Region us-east-1
```

下面是示例输出。

```
block-new-sharing
```

示例：对于账户中的所有区域

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
        [PSCustomObject]@{
            Region   = $_
            PublicAccessState = (Get-EC2ImageBlockPublicAccessState -Region $_)
        }
} | `
Format-Table -AutoSize
```

下面是示例输出。

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

启用阻止公开访问 AMI

要防止公开共享您的 AMI，请在账户级别启用阻止公开访问 AMI。您必须在要阻止公开共享 AMI 的每个 Amazon Web Services 区域 中启用阻止公开访问 AMI。如果您已经有公开 AMI，它们将保持公开状态。

所需的权限

要启用 AMI 的屏蔽公共访问设置，您必须拥有 `EnableImageBlockPublicAccess` IAM 权限。

注意事项

- 最多可能需要 10 分钟来配置此设置。在此期间，如果您描述了公共访问状态，则响应为 `unblocked`。配置完成后，响应将为 `block-new-sharing`。

Console

在指定区域中启用 AMI 的屏蔽公共访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航栏（位于屏幕顶部）中，选择启用阻止公开访问 AMI 的区域。
3. 在导航窗格中，选择控制面板。
4. 在账户属性卡的设置下，选择数据保护和安全。
5. 在阻止公开访问 AMI 下，选择管理。
6. 选择阻止新的公开共享复选框，然后选择更新。

Amazon CLI

要启用 AMI 的阻止公开访问

使用 enable-image-block-public-access 命令。

示例：对于特定区域

```
aws ec2 enable-image-block-public-access \
--region us-east-1 \
--image-block-public-access-state block-new-sharing
```

下面是示例输出。

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

示例：对于账户中的所有区域

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 enable-image-block-public-access \
            --region $region \
            --image-block-public-access-state block-new-sharing \
            --output text)
```

```
        echo -e "$region \t $output"
    );
done
```

下面是示例输出。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       block-new-sharing
eu-west-3        block-new-sharing
...
```

PowerShell

要启用 AMI 的阻止公开访问

使用 Enable-EC2ImageBlockPublicAccess 命令。

示例：对于特定区域

```
Enable-EC2ImageBlockPublicAccess `
    -Region us-east-1 `
    -ImageBlockPublicAccessState block-new-sharing
```

下面是示例输出。

```
Value
-----
block-new-sharing
```

示例：对于账户中的所有区域

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (
        Enable-EC2ImageBlockPublicAccess `
         -Region $_ `
```

```
          -ImageBlockPublicAccessState block-new-sharing)
    }
} | `
Format-Table -AutoSize
```

下面是示例输出。

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

禁用阻止公开访问 AMI

要允许您账户中的用户公开共享您的 AMI，请在账户级别禁用阻止公开访问。在要允许公开共享 AMI 的每个 Amazon Web Services 区域 中，您都必须禁用阻止公开访问 AMI。

所需的权限

要禁用 AMI 的屏蔽公共访问设置，您必须拥有 `DisableImageBlockPublicAccess` IAM 权限。

注意事项

- 最多可能需要 10 分钟来配置此设置。在此期间，如果您描述了公共访问状态，则响应为 `block-new-sharing`。配置完成后，响应将为 `unblocked`。

Console

在指定区域禁用 AMI 的屏蔽公共访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航栏（位于屏幕顶部）中，选择禁用阻止公开访问 AMI 的区域。
3. 在导航窗格中，选择控制面板。
4. 在账户属性卡的设置下，选择数据保护和安全。
5. 在阻止公开访问 AMI 下，选择管理。
6. 清除阻止新的公开共享复选框，然后选择更新。

7. 提示进行确认时，输入 **confirm**，然后选择允许公开共享。

Amazon CLI

要禁用 AMI 的阻止公开访问

使用 disable-image-block-public-access 命令。

示例：对于特定区域

```
aws ec2 disable-image-block-public-access --region us-east-1
```

下面是示例输出。

```
{
   "ImageBlockPublicAccessState": "unblocked"
}
```

示例：对于账户中的所有区域

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 disable-image-block-public-access \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

下面是示例输出。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
```

```
eu-north-1       unblocked
eu-west-3        unblocked
...
```

PowerShell

要禁用 AMI 的阻止公开访问

使用 Disable-EC2ImageBlockPublicAccess cmdlet。

示例：对于特定区域

```
Disable-EC2ImageBlockPublicAccess -Region us-east-1
```

下面是示例输出。

```
Value
-----
unblocked
```

示例：对于账户中的所有区域

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Disable-EC2ImageBlockPublicAccess -Region $_)
    }
} | `
Format-Table -AutoSize
```

下面是示例输出。

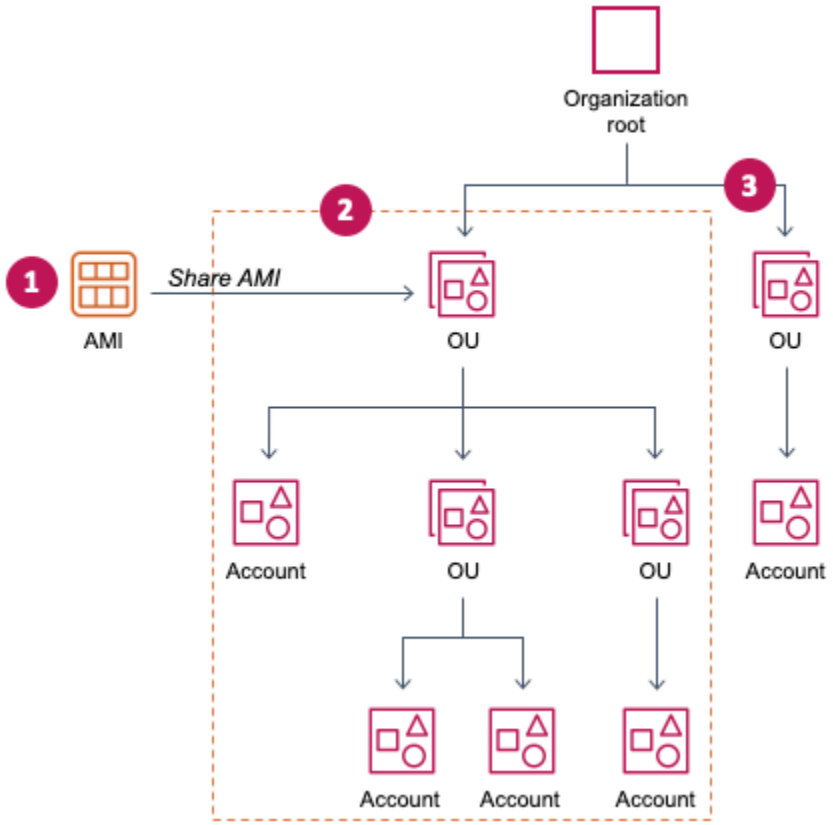
```
Region         PublicAccessState
------         -----------------
ap-south-1     unblocked
eu-north-1     unblocked
eu-west-3      unblocked
...
```

与组织和组织单位共享 AMI

Amazon Organizations 是一项账户管理服务，可让您将多个 Amazon Web Services 账户 整合到您创建并集中管理的组织中。除了与特定账户共享 AMI 之外，您还可以与您创建的企业或企业部门 (OU) 共享 AMI。

组织是指您创建用于整合和集中管理您的 Amazon Web Services 账户 的实体。您可以以分层树状结构来组织账户，将根放在树顶部，并将组织单位嵌套在组织根下。每个账户都可以直接添加到根中，也可以放在层次结构的其中一个 OU 中。有关更多信息，请参阅《Amazon 用户指南》https://docs.amazonaws.cn/organizations/latest/userguide/orgs_getting-started_concepts.html中的 Amazon Organizations Organizations 术语和概念。

当您与企业或 OU 共享 AMI 时，所有子账户都可以访问该 AMI。例如，在下图中，AMI 与顶层 OU 共享（由数字 1 处的箭头指示）。嵌套在该顶层 OU 下面的所有 OU 和账户（由数字 2 处的虚线指示）还可以访问 AMI。组织中的账户和 OU 在虚线之外的账户（由数字 3 指示）没有访问该 AMI 的权限，因为它们不是 AMI 共享的 OU 的子级账户。



主题

- 注意事项

- 获取组织或组织单位的 ARN
- 允许企业和 OU 使用 KMS 密钥
- 管理与组织或 OU 共享的 AMI

注意事项

在与特定企业或企业部门共享 AMI 时，请考虑以下事项。

- 所有权 – 若要共享 AMI，您的 Amazon Web Services 账户 必须拥有 AMI。
- 共享限制 – AMI 拥有者可以与任何组织或 OU 共享 AMI，包括他们不是其成员的组织和 OU。

  有关某一区域内可以共享 AMI 的最大实体数量，请参阅 Amazon EC2 服务限额。
- 标签 – 您无法共享用户定义的标签（附加到 AMI 的标签）。共享 AMI 时，与其共享 AMI 的组织或 OU 中的 Amazon Web Services 账户 的任何用户均无法使用用户定义的标签。
- ARN 格式 – 在命令中指定企业或 OU 时，请确保使用正确的 ARN 格式。如果只指定 ID，例如，如果您仅指定 `o-123example` 或 `ou-1234-5example`，则会出现错误。

  正确的 ARN 格式：

  - 企业 ARN：`arn:aws:organizations::`*`111122223333`*`:organization/`*`organization-id`*
  - OU ARN：`arn:aws:organizations::`*`111122223333`*`:ou/`*`organization-id`*`/`*`ou-id`*

  其中：

  - *`111122223333`* 举例说明管理账户的 12 位账户 ID。如果您不知道管理账号，您可以描述企业或企业部门以获取 ARN，其中包括管理账号。有关更多信息，请参阅 获取组织或组织单位的 ARN。
  - *`organization-id`* 是企业 ID，例如 `o-123example`。
  - *`ou-id`* 是企业部门 ID，例如 `ou-1234-5example`。

  有关 ARN 格式的更多信息，请参阅 IAM 用户指南中的 Amazon 资源名称(ARN)。
- 加密和密钥 – 您可以共享由未加密和加密快照支持的 AMI。
  - 加密快照必须使用客户托管式密钥加密。您无法共享由使用默认 Amazon 托管式密钥加密的快照支持的 AMI。
  - 如果您共享由加密快照支持的 AMI，则必须允许企业或 OU 使用用于加密快照的客户托管式密钥。有关更多信息，请参阅 允许企业和 OU 使用 KMS 密钥。

- 区域 – AMI 是一种区域性资源。当您共享 AMI 时，则它只能在您共享该 AMI 的区域使用。要使 AMI 能够在其他区域使用，请将该 AMI 复制到该区域并进行共享。有关更多信息，请参阅 复制 Amazon EC2 AMI。
- 使用 – 当您共享 AMI 时，用户只能从该 AMI 启动实例。他们无法删除、共享或修改实例。但是，在他们使用您的 AMI 启动实例后，他们可以从其启动的实例创建 AMI。
- 账单 – 当其他 Amazon Web Services 账户 使用您的 AMI 启动实例时，您无需付费。使用 AMI 启动实例的账户将为它们启动的实例付费。

获取组织或组织单位的 ARN

企业和企业部门 ARN 包含 12 位数的管理账号。如果您不知道管理账号，您可以描述企业或企业部门以获取每个账号的 ARN。在以下示例中，123456789012 是管理账户的账户 ID。

所需的权限

在获得 ARN 之前，您必须拥有描述企业和企业部门的权限。以下示例策略提供了必要权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "organizations:Describe*"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon CLI

获取企业的 ARN

使用 describe-organization 命令。添加 `--query` 选项将仅返回组织 ARN。

```
aws organizations describe-organization --query 'Organization.Arn'
```

下面是示例输出。

```
"arn:aws:organizations::123456789012:organization/o-1234567abc"
```

获取企业部门的 ARN

使用 describe-organizational-unit 命令。使用 `--query` 参数将仅返回组织单元 ARN。

```
aws organizations describe-organizational-unit \
    --organizational-unit-id ou-a123-b4567890 \
    --query 'OrganizationalUnit.Arn'
```

下面是示例输出。

```
"arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890"
```

PowerShell

获取企业的 ARN

使用 Get-ORGOrganization cmdlet。

```
(Get-ORGOrganization).Arn
```

下面是示例输出。

```
arn:aws:organizations::123456789012:organization/o-1234567abc
```

获取企业部门的 ARN

使用 Get-ORGOrganizationalUnit cmdlet。

```
(Get-ORGOrganizationalUnit -OrganizationalUnitId "ou-a123-b4567890").Arn
```

下面是示例输出。

```
arn:aws:organizations::123456789012:ou/o-1234567abc/ou-a123-b4567890
```

允许企业和 OU 使用 KMS 密钥

如果共享由加密快照支持的 AMI，则还须允许组织或组织单位（OU）使用用于加密快照的 KMS 密钥。

> **Note**
>
> 加密快照必须使用客户管理型密钥加密。您无法共享由使用默认 Amazon 托管式密钥加密的快照支持的 AMI。

要控制对 KMS 密钥的访问，可以在密钥策略中使用 `aws:PrincipalOrgID` 和 `aws:PrincipalOrgPaths` 条件键来仅允许特定主体执行指定操作。主体可以是用户、IAM 角色、联合用户或 Amazon Web Services 账户根用户。

条件键的使用方式如下：

- `aws:PrincipalOrgID`：允许属于由指定 ID 表示的组织的任何主体。
- `aws:PrincipalOrgPaths`：允许属于由指定路径表示的组织单位的任何主体。

要向组织（包括属于该组织的组织单位和账户）授予使用 KMS 密钥的权限，请向密钥策略添加以下语句。

```
{
    "Sid": "Allow access for organization root",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
```

```
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

要向特定组织单位（和属于该组织单位的账户）授予使用 KMS 密钥的权限，您可以使用类似于下列示例的策略。

```
{
        "Sid": "Allow access for specific OUs and their descendants",
        "Effect": "Allow",
        "Principal": "*",
        "Action": [
            "kms:Describe*",
            "kms:List*",
            "kms:Get*",
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:CreateGrant"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": "o-123example"
            },
            "ForAnyValue:StringLike": {
                "aws:PrincipalOrgPaths": [
                    "o-123example/r-ab12/ou-ab12-33333333/*",
                    "o-123example/r-ab12/ou-ab12-22222222/*"
                ]
            }
        }
}
```

有关更多示例条件语句，请参阅《IAM 用户指南》https://docs.amazonaws.cn/IAM/latest/UserGuide/reference_policies_condition-keys.html#condition-keys-principalorgid中的 aws:PrincipalOrgID 和 aws:PrincipalOrgPaths。

有关跨账户访问的更多信息，请参阅《Amazon Key Management Service Developer Guide》中的 Allowing users in other accounts to use a KMS key。

管理与组织或 OU 共享的 AMI

您可以使用组织和组织部门（OU）来管理 AMI 共享，从而控制其是否可以启动 Amazon EC2 实例。

查看共享 AMI 的企业和 OU

可以查找与 AMI 共享的组织和组织单元。

Console

检查您与哪些组织和组织单位共享了 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 在列表中选择您的 AMI，选择 Permissions（权限）选项卡，然后向下滚动到 Shared organizations/OUs（共享企业/OU）。

   要查找与您共享的 AMI，请参阅查找用于 Amazon EC2 实例的共享 AMI。

Amazon CLI

检查您与哪些组织和组织单位共享了 AMI

使用带 `launchPermission` 属性的 describe-image-attribute 命令。

```
aws ec2 describe-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

以下为响应示例。

```
{
    "ImageId": "ami-0abcdef1234567890",
    "LaunchPermissions": [
        {
            "OrganizationalUnitArn": "arn:aws:organizations::111122223333:ou/
o-123example/ou-1234-5example"
        }
    ]
}
```

PowerShell

检查您与哪些组织和组织单位共享了 AMI

使用 Get-EC2ImageAttribute cmdlet。

```
Get-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

与组织或 OU 共享 AMI

可以与组织或组织单元共享 AMI。

> **Note**
>
> 您不需要为了共享 AMI 而共享 AMI 引用的 Amazon EBS 快照。只需共享 AMI 本身，系统会自动为实例提供访问所引用 EBS 快照的权限，以便启动。不过，您确实需要共享用于对 AMI 引用的快照加密的 KMS 密钥。有关更多信息，请参阅 允许企业和 OU 使用 KMS 密钥。

Console

与企业或 OU 共享 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 在列表中选择您的 AMI，然后选择 Actions、然后选择 Edit AMI Permissions。
4. 在 AMI availability（AMI 可用性）下，选择 Private（私有）。
5. 在 Shared organizations/OUs（共享企业/OU）旁，选择 Add organization/OU ARN（添加企业/OU ARN）。
6. 对于 Organization/OU ARN（企业/OU ARN），输入要与之共享 AMI 的企业 ARN 或 OU ARN，然后选择 Share AMI（共享 AMI）。请注意，您必须指定完整 ARN，而不仅仅是 ID。

   要与多个企业或 OU 共享此 AMI，请重复此步骤，直至您添加完所需全部企业或 OU。
7. 完成后，选择 Save changes (保存更改)。

8. （可选）要查看您已共享 AMI 的企业或 OU，请在列表中选择此 AMI，然后选择 Permissions（权限）选项卡，然后向下滚动到 Shared organizations/OUs（共享企业/OU）。要查找与您共享的 AMI，请参阅查找用于 Amazon EC2 实例的共享 AMI。

Amazon CLI

与组织共享 AMI

使用 modify-image-attribute 命令，向指定组织授予指定 AMI 的启动权限。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

与组织单位共享 AMI

modify-image-attribute 命令向指定 OU 授予指定 AMI 的启动许可。请注意，您必须指定完整 ARN，而不仅仅是 ID。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Add=[{OrganizationalUnitArn=arn:aws:organizations::123456789012:ou/o-123example/
ou-1234-5example}]"
```

PowerShell

使用 Edit-EC2ImageAttribute 命令（适用于 Windows PowerShell 的工具）共享 AMI，如以下示例所示。

与企业或 OU 共享 AMI

以下命令向指定企业授予指定 AMI 的启动许可。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
```

```
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

停止与企业或 OU 共享 AMI

以下命令从指定企业中删除指定 AMI 的启动许可：

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

停止与所有组织、OU 和 Amazon Web Services 账户 共享 AMI

以下命令从指定 AMI 中删除所有公用和显式启动许可。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

停止与组织或 OU 共享 AMI

可以停止与组织或组织单元共享 AMI。

> **Note**
>
> 如果某个特定账户位于您与之共享 AMI 的企业或 OU 中，则您无法停止与该账户共享 AMI 如果您尝试通过删除账户的启动许可来停止共享 AMI，Amazon EC2 将返回成功消息。但是，AMI 将继续与账户共享。

Console

停止与企业或 OU 共享 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。

3. 在列表中选择您的 AMI，然后选择 Actions、然后选择 Edit AMI Permissions。
4. 在 Shared organizations/OUs（共享企业/OU）下，选择要停止共享 AMI 的企业或 OU，然后选择 Remove selected（删除所选）。
5. 完成后，选择 Save changes (保存更改)。
6. （可选）要确认您已停止与企业或 OU 共享 AMI，请在列表中选择此 AMI，然后选择 Permissions（权限）选项卡，然后向下滚动到 Shared organizations/OUs（共享企业/OU）。

Amazon CLI

停止与企业或 OU 共享 AMI

使用 modify-image-attribute 命令。此示例从指定组织中删除指定 AMI 的启动权限。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission
 "Remove=[{OrganizationArn=arn:aws:organizations::123456789012:organization/
o-123example}]"
```

停止与所有组织、OU 和 Amazon Web Services 账户 共享 AMI

使用 reset-image-attribute 命令。此示例从指定 AMI 中删除所有公有和显式启动权限。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
aws ec2 reset-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

停止与企业或 OU 共享 AMI

使用 Edit-EC2ImageAttribute cmdlet。此示例从指定组织中删除指定 AMI 的启动权限。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType remove `
```

```
    -OrganizationArn "arn:aws:organizations::123456789012:organization/o-123example"
```

停止与所有组织、OU 和 Amazon Web Services 账户 共享 AMI

使用 Reset-EC2ImageAttribute cmdlet。此示例从指定 AMI 中删除所有公有和显式启动权限。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute LaunchPermission
```

与特定 Amazon 账户共享 AMI

您可以在不将 AMI 设为公共的情况下，与特定 Amazon Web Services 账户 共享 AMI。您需要的只是 Amazon Web Services 账户 ID。

Amazon Web Services 账户 ID 是一个 12 位数字（如 012345678901），用于唯一标识 Amazon Web Services 账户。有关更多信息，请参阅《Amazon 账户管理 参考指南》中的 Viewing Amazon Web Services 账户 identifiers。

注意事项

在与特定 Amazon Web Services 账户 共享 AMI 时，请考虑以下事项。

- 所有权 – 若要共享 AMI，您的 Amazon Web Services 账户 必须拥有 AMI。
- 共享限制 - 有关某一区域内可以共享 AMI 的最大实体数量，请参阅 Amazon EC2 服务限额。
- 标签 – 您无法共享用户定义的标签（附加到 AMI 的标签）。共享 AMI 时，与其共享 AMI 的任何 Amazon Web Services 账户 均无法使用用户定义的标签。
- 快照 – 您不需要共享某个 AMI 为了共享该 AMI 而引用的 Amazon EBS 快照。您可以仅共享 AMI 本身；系统会自动为实例提供访问所引用 EBS 快照的权限来完成启动。不过，您必须共享用于对 AMI 引用的快照进行加密的所有 KMS 密钥。有关更多信息，请参阅《Amazon EBS 用户指南》中的共享 Amazon EBS 快照。
- 加密和密钥 – 您可以共享由未加密和加密快照支持的 AMI。
  - 加密快照必须使用 KMS 密钥加密。您无法共享由使用默认 Amazon 托管式密钥加密的快照支持的 AMI。

  - 如果您共享由加密快照支持的 AMI，则必须允许 Amazon Web Services 账户 使用用于加密快照的 KMS 密钥。有关更多信息，请参阅 允许企业和 OU 使用 KMS 密钥。要设置在使用客户托管密钥进行加密时启动 Auto Scaling 实例所需的密钥策略，请参阅《Amazon EC2 Auto Scaling 用户指南》中的加密卷所需的 Amazon KMS key 策略。
- 区域 – AMI 是一种区域性资源。如果您共享 AMI，则它只能在该区域使用。要使 AMI 能够在其他区域使用，请将该 AMI 复制到该区域并进行共享。有关更多信息，请参阅 复制 Amazon EC2 AMI。
- 使用 – 当您共享 AMI 时，用户只能从该 AMI 启动实例。他们无法删除、共享或修改实例。但是，在他们使用您的 AMI 启动实例后，他们可以从其实例创建 AMI。
- 复制共享 AMI – 如果另一个账户中的用户想复制共享 AMI，则必须向他们授予对支持 AMI 的存储的读取权限。有关更多信息，请参阅 跨账户复制。
- 账单 – 当其他 Amazon Web Services 账户 使用您的 AMI 启动实例时，您无需付费。使用 AMI 启动实例的账户将为它们启动的实例付费。

Console

要授予显式启动许可

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 在列表中选择您的 AMI，然后选择 Actions、然后选择 Edit AMI Permissions。
4. 选择私有。
5. 在 Shared accounts（共享账户）下，选择 Add account ID（添加账户 ID）。
6. 对于 Amazon Web Services 账户 ID，输入要与其共享 AMI 的 Amazon Web Services 账户 ID，然后选择 Share AMI（共享 AMI）。

   要与多账户共享此 AMI，请重复此步骤 5 和 6，直至您添加完所需全部账户 ID。
7. 完成后，选择 Save changes（保存更改）。
8. （可选）要查看您已与其共享 AMI 的 Amazon Web Services 账户 ID，请在列表中选择该 AMI，然后选择 Permissions（权限）选项卡。要查找与您共享的 AMI，请参阅查找用于 Amazon EC2 实例的共享 AMI。

Amazon CLI

使用 modify-image-attribute 命令共享 AMI，如以下示例所示。

要授予显式启动许可

以下示例向指定 Amazon Web Services 账户 授予指定 AMI 的启动权限。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Add=[{UserId=123456789012}]"
```

要删除账户的启动许可

以下示例从指定 Amazon Web Services 账户 中删除指定 AMI 的启动权限。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

要删除所有的启动许可

以下示例从指定 AMI 中删除所有公有和显式启动权限。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
aws ec2 reset-image-attribute \
    --image-id ami-0abcdef1234567890 \
    --attribute launchPermission
```

PowerShell

使用 Edit-EC2ImageAttribute 命令（适用于 Windows PowerShell 的工具）共享 AMI，如以下示例所示。

要授予显式启动许可

以下示例向指定 Amazon Web Services 账户 授予指定 AMI 的启动权限。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission `
    -OperationType add `
    -UserId "123456789012"
```

要删除账户的启动许可

以下示例从指定 Amazon Web Services 账户 中删除指定 AMI 的启动权限。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission -OperationType remove `
    -UserId "123456789012"
```

要删除所有的启动许可

以下示例从指定 AMI 中删除所有公有和显式启动权限。请注意，AMI 的拥有者始终具有启动许可，因此不受该命令影响。

```
Reset-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -Attribute launchPermission
```

取消与您的 Amazon Web Services 账户 共享 AMI

通过将账户添加到 AMI 的启动权限，可以与特定 Amazon Web Services 账户 共享亚马逊机器映像（AMI）。如果已与您的 Amazon Web Services 账户 共享 AMI，而您不想再与您的账户共享该 AMI，则可以将您的账户从 AMI 的启动权限中移除。您可以通过运行 `cancel-image-launch-permission` Amazon CLI 命令进行这项操作。在运行此命令时，将从指定 AMI 的启动权限中移除您的 Amazon Web Services 账户。若要查找与 Amazon Web Services 账户 共享的 AMI，请参阅 查找用于 Amazon EC2 实例的共享 AMI。

例如，您可以取消与您的账户共享 AMI，以降低使用已与您共享的未使用或已弃用的 AMI 启动实例的可能性。当您取消与您的账户共享 AMI 时，它将不再出现在 EC2 控制台的任何 AMI 列表中，也不会出现在 describe-images 的输出中。

主题

- 限制
- 取消与您的账户共享 AMI

限制

- 您可以将您的账户从仅与您的 Amazon Web Services 账户 共享的 AMI 的启动权限中移除。您不能使用 `cancel-image-launch-permission` 从 与组织或组织单位（OU）共享的 AMI 的启动权限中移除您的账户，也不能将其用于移除对公共 AMI 的访问权限。

- 您不能从 AMI 的启动权限中永久移除您的账户。AMI 拥有者可以再次与您的账户共享 AMI。
- AMI 是一种区域性资源。在运行 `cancel-image-launch-permission` 时，必须指定 AMI 所在的区域。可在命令中指定区域，或者使用 AWS_DEFAULT_REGION 环境变量。
- 只有 Amazon CLI 和 SDK 支持从 AMI 的启动权限中移除您的账户。EC2 控制台目前不支持此操作。

取消与您的账户共享 AMI

> **Note**
>
> 在您取消与您的账户共享 AMI 后，您将无法撤消该操作。要重新获得对 AMI 的访问权限，AMI 拥有者必须将其与您的账户共享。

Amazon CLI

取消与账户共享 AMI

使用 cancel-image-launch-permission 命令。

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0abcdef1234567890 \
    --region us-east-1
```

PowerShell

取消与账户共享 AMI

使用 Stop-EC2ImageLaunchPermission cmdlet。

```
Stop-EC2ImageLaunchPermission `
    -ImageId ami-0abcdef1234567890 `
    -Region us-east-1
```

创建共享 Linux AMI 的建议

使用以下指南可缩小攻击面并提高您创建的 AMI 的可靠性。

> **Important**
>
> 任何安全指南都不是详尽无遗的。请仔细构建您的共享 AMI，并花时间考虑可能导致暴露敏感数据的位置。

目录

- 对根用户禁用基于密码的远程登录
- 禁用本地根访问
- 删除 SSH 主机密钥对
- 安装公有密钥凭证
- 禁用 sshd DNS 检查（可选）
- 移除敏感数据

如果为 Amazon Web Services Marketplace 构建 AMI，请参阅 Amazon Web Services Marketplace 卖家指南中的构建 AMI 的最佳实践，以了解指导原则、策略和最佳实践。

对根用户禁用基于密码的远程登录

为公用 AMI 使用固定的根密码是一种很快为人知晓的安全风险。甚至于用户在第一次登录后更改密码都会给可能的滥用以可乘之机。

要解决此问题，请对根用户禁用基于密码的远程登录。

对根用户禁用基于密码的远程登录

1. 用文字编辑器打开 `/etc/ssh/sshd_config` 文件并查找以下行：

   ```
   #PermitRootLogin yes
   ```

2. 将行更改为：

   ```
   PermitRootLogin without-password
   ```

   若您的发行版不同或您未运行 OpenSSH，此配置文件的位置可能也会不同。若情况如此，请咨询相关文档。

禁用本地根访问

在使用共享 AMI 时，最佳做法是禁用直接根登录。为此，请登录到您正在运行的实例并发出以下命令：

```
[ec2-user ~]$ sudo passwd -l root
```

> **Note**
>
> 该命令不影响 sudo 的使用。

删除 SSH 主机密钥对

如果您计划共享源自公用 AMI 的 AMI，请删除 /etc/ssh 中的现有 SSH 主机密钥对。这会促使 SSH 在有人使用您的 AMI 启动实例时生成新的独特 SSH 密钥对，从而提高安全性并降低“中间人”攻击可能性。

删除系统上存在的以下所有密钥文件。

- ssh_host_dsa_key
- ssh_host_dsa_key.pub
- ssh_host_key
- ssh_host_key.pub
- ssh_host_rsa_key
- ssh_host_rsa_key.pub
- ssh_host_ecdsa_key
- ssh_host_ecdsa_key.pub
- ssh_host_ed25519_key
- ssh_host_ed25519_key.pub

您可以使用以下命令安全地删除所有这些文件。

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

> **Warning**
>
> 安全删除实用工具（例如 shred）可能不会移除存储介质中某个文件的所有副本。文件的隐藏副本可能是由日志文件系统 (包括 Amazon Linux 默认 ext4)、快照、备份、RAID 和临时缓存创建的。有关更多信息，请参阅 shred 文档。

> **Important**
>
> 如果您忘记从您的公用 AMI 中删除现有 SSH 主机密钥对，我们的例行审核过程会通知您和所有运行您的 AMI 实例的客户存在潜在安全风险。短暂的宽限期过后，我们会将 AMI 标记为私有。

安装公有密钥凭证

配置 AMI 以防止使用密码进行登录后，您必须确保用户能用另一种机制登录。

Amazon EC2 允许用户在启动实例时指定公用–私有密钥对名称。向 `RunInstances` API 调用提供有效的密钥对名称后（或通过命令行 API 工具），公用密钥（Amazon EC2 在至 `CreateKeyPair` 或 `ImportKeyPair` 的调用后在服务器上保留的密钥对的部分)通过针对实例元数据的 HTTP 查询供实例使用。

要通过 SSH 登录，您的 AMI 必须在启动时检索密钥值并将该值附加到 `/root/.ssh/authorized_keys` (或 AMI 上任何其他用户账户的等效密钥)。用户可使用密钥对启动您的 AMI 的实例，并在不需要根密码的情况下进行登录。

很多发行版（包括 Amazon Linux 和 Ubuntu）使用 `cloud-init` 软件包为配置的用户插入公有密钥凭证。如果您的发行版不支持 `cloud-init`，则可以将以下代码添加到系统启动脚本（如 `/etc/rc.local`），以提取您在启动时为根用户指定的公有密钥。

> **Note**
>
> 在以下示例中，IP 地址 http://169.254.169.254/ 是链路本地地址，仅从该实例有效。

IMDSv2

```
if [ ! -d /root/.ssh ] ; then
```

```
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
        cat /tmp/my-key >> /root/.ssh/authorized_keys
        chmod 700 /root/.ssh/authorized_keys
        rm /tmp/my-key
fi
```

IMDSv1

```
if [ ! -d /root/.ssh ] ; then
        mkdir -p /root/.ssh
        chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
        cat /tmp/my-key >> /root/.ssh/authorized_keys
        chmod 700 /root/.ssh/authorized_keys
        rm /tmp/my-key
fi
```

这一点适用于任何用户；无需将其限制为 `root` 用户。

> **Note**
>
> 根据此 AMI 进行的实例重新捆绑包括启动时所用的密钥。要防止密钥被包含，您必须清除 (或删除) `authorized_keys` 文件或将此文件排除在重新捆绑之外。

禁用 sshd DNS 检查（可选）

禁用 sshd DNS 检查会稍微减弱您的 sshd 安全性。但是，如果 DNS 解析失败，SSH 的登录仍然有效。如果您未禁用 sshd 检查，DNS 解析失败后会阻止所有的登录。

要禁用 sshd DNS 检查

1. 用文字编辑器打开 /etc/ssh/sshd_config 文件并查找以下行：

```
#UseDNS yes
```

2. 将行更改为：

```
UseDNS no
```

> **Note**
>
> 若您的发行版不同或您未运行 OpenSSH，此配置文件的位置也会不同。若情况如此，请咨询相关文档。

移除敏感数据

我们不建议您将敏感数据或软件存储在您共享的任何 AMI 上。启动共享 AMI 的用户可能能够重新捆绑 AMI 并能自行注册 AMI。遵循上述指南可助您避免一些容易被忽视的安全风险：

- 我们建议对 `--exclude` *`directory`* 使用 ec2-bundle-vol 选项，以跳过包含您不想在捆绑中包含的机密信息的所有目录和子目录。具体而言，在捆绑映像时，排除所有用户拥有的 SSH 公有/私有密钥对和 SSH authorized_keys 文件。Amazon 公有 AMI 会将其存储在 /root/.ssh（对于根用户）和 /home/*user_name*/.ssh/（对于常规用户）中。有关更多信息，请参阅 ec2-bundle-vol。
- 务必在捆绑前删除 Shell 程序历史记录。如果您在同一 AMI 中多次尝试捆绑上传，Shell 程序历史中将包含访问密钥。以下示例应为从实例内部捆绑前您运行的最后一个命令。

```
[ec2-user ~]$ shred -u ~/.*history
```

> **Warning**
>
> 以上警告中描述的 shred 的限制在此处也适用。
> 请注意，bash 在退出时会将当前会话的历史记录写入磁盘。如果您在删除 ~/.bash_history 后注销您的实例，然后重新登录，您将发现 ~/.bash_history 已重新创建且包含上一会话期间运行的所有命令。

Bash 以外的其他程序也会将历史记录写入磁盘，请谨慎使用并删除或排除不必要的点文件和点目录。

- 捆绑正在运行的实例需要您的私有密钥和 X.509 证书。将上述密钥和凭证以及其他证书放置到未予捆绑的位置 (如实例存储)。

使用 Amazon EventBridge 监控 AMI 事件

当亚马逊机器映像（AMI）状态发生变化时，Amazon EC2 将生成发送到 Amazon EventBridge（以前称为 Amazon CloudWatch Events）的事件。事件以 JSON 格式发送到默认的 EventBridge 事件总线。您可以使用 Amazon EventBridge 检测并响应这些事件。要做到这一点，您可以在 EventBridge 中创建触发操作以响应事件的规则。例如，您可以创建 EventBridge 规则，以检测 AMI 创建过程何时完成，然后调用 Amazon SNS 主题以向您发送电子邮件通知。

当 AMI 进入以下任何状态时，Amazon EC2 将生成 `EC2 AMI State Change` 事件：

- `available`
- `failed`
- `deregistered`
- `disabled`

事件将尽最大努力生成。

下表列出了 AMI 操作和 AMI 可能存在的状态。在表中，是表示相应操作运行时 AMI 可能存在的状态。

AMI 操作	available	failed	deregistered	disabled
CopyImage	支持	是		
CreateImage	是	是		
CreateRestoreImageTask	是	是		
DeregisterImage			是	
DisableImage				是

AMI 操作	available	failed	deregistered	disabled
EnableImage	是			
RegisterImage	是	是		

EC2 AMI State Change 事件

- 事件详细信息
- available 事件
- failed 事件
- deregistered 事件
- disabled 事件

事件详细信息

您可以使用事件中的以下字段创建触发操作的规则：

`"source": "aws.ec2"`

识别来自 Amazon EC2 的事件。

`"detail-type": "EC2 AMI State Change"`

识别事件名称。

`"detail": { "ImageId": "ami-0abcdef1234567890", "State": "available", }`

提供 AMI ID 和 AMI 的状态（`available`、`failed`、`deregistered` 或 `disabled`）。

有关更多信息，请参阅《Amazon EventBridge User Guide》中的以下内容：

- Amazon EventBridge 事件
- Amazon EventBridge 事件模式
- Amazon EventBridge 规则

有关如何创建 Lambda 函数和运行 Lambda 函数的 EventBridge 规则的教程，请参阅《Amazon Lambda Developer Guide》中的 Tutorial: Log the state of an Amazon EC2 instance using EventBridge。

available 事件

以下是 Amazon EC2 在 AMI 于 `CreateImage`、`CopyImage`、`RegisterImage`、`CreateRestoreImageTask` 或 `EnableImage` 操作成功后进入 `available` 状态时所生成的事件的示例。

`"State": "available"` 表明 操作成功。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "available",
        "ErrorMessage": ""
    }
}
```

failed 事件

以下是 Amazon EC2 在 AMI 于 `CreateImage`、`CopyImage`、`RegisterImage` 或 `CreateRestoreImageTask` 操作失败后进入 `failed` 状态时所生成的事件的示例。

以下字段提供相关信息：

- `"State": "failed"` – 指示操作失败。
- `"ErrorMessage": ""` – 提供操作失败的原因。

```
{
```

```
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "failed",
        "ErrorMessage": "Description of failure"
    }
}
```

deregistered 事件

以下是 Amazon EC2 在 AMI 于 DeregisterImage 操作成功后进入 deregistered 状态时所生成的事件的示例。如果操作失败，则不会生成任何事件。由于 DeregisterImage 是同步操作，任何故障都能被立即知晓。

"State": "deregistered" 表明 DeregisterImage 操作成功。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "deregistered",
        "ErrorMessage": ""
    }
}
```

disabled 事件

以下是 Amazon EC2 在 AMI 于 `DisableImage` 操作成功后进入 `disabled` 状态时所生成的事件的示例。如果操作失败，则不会生成任何事件。由于 `DisableImage` 是同步操作，任何故障都能被立即知晓。

`"State": "disabled"` 表明 `DisableImage` 操作成功。

```
{
    "version": "0",
    "id": "example-9f07-51db-246b-d8b8441bcdf0",
    "detail-type": "EC2 AMI State Change",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1::image/ami-0abcdef1234567890"],
    "detail": {
        "RequestId": "example-9dcc-40a6-aa77-7ce457d5442b",
        "ImageId": "ami-0abcdef1234567890",
        "State": "disabled",
        "ErrorMessage": ""
    }
}
```

了解 AMI 账单信息

启动实例时，有许多亚马逊机器映像（AMI）可供选择，它们支持各种操作系统平台和功能。要了解您在启动实例时选择的 AMI 如何影响 Amazon 账单的底线，您可以研究相关的操作系统平台和账单信息。在启动任何按需或竞价型实例之前执行此操作，或者购买预留实例。

下面是关于提前研究 AMI 如何帮助您选择最适合自己需求的 AMI 的两个示例：

- 对于竞价型实例，您可以使用 AMI Platform details (平台详细信息) 来确认对于竞价型实例支持 AMI。
- 购买时 Reserved Instance 时，您可以确保选择映射到 AMI Platform details (平台详细信息) 的操作系统平台（Platform [平台]）。

有关实例定价的更多信息，请参阅 Amazon EC2 定价。

目录

- AMI 账单信息字段
- 查找 AMI 账单和使用详细信息
- 验证账单上的 AMI 费用

AMI 账单信息字段

以下字段提供与 AMI 关联的账单信息：

平台详细信息

与 AMI 的账单代码关联的平台详细信息。例如：`Red Hat Enterprise Linux`。

使用情况操作

Amazon EC2 实例的操作以及与 AMI 关联的账单代码。例如：`RunInstances:0010`。Usage operation (使用情况操作) 对应于 Amazon 成本和使用情况报告 (CUR) 和 Amazon 价目表 API 中的 lineitem/Operation 列。

您可以在 Amazon EC2 控制台的实例或 AMI 页面上查看这些字段，或者在 describe-images 或 Get-EC2Image 命令返回的响应中查看这些字段。

示例数据：按平台划分的使用情况操作

下表列出了一些平台详细信息和使用情况操作值，这些信息显示在 Amazon EC2 控制台的实例或 AMI 页面上，或者显示在由 describe-images 或 Get-EC2Image 命令返回的响应中。

平台详细信息	使用情况操作 [2]
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0[3]
Red Hat Enterprise Linux	RunInstances:0010
Red Hat Enterprise Linux with HA	RunInstances:1010

平台详细信息	使用情况操作 [2]
Red Hat Enterprise Linux with SQL Server Standard and HA	RunInstances:1014
Red Hat Enterprise Linux with SQL Server Enterprise and HA	RunInstances:1110
Red Hat Enterprise Linux with SQL Server Standard	RunInstances:0014
Red Hat Enterprise Linux with SQL Server Web	RunInstances:0210
Red Hat Enterprise Linux with SQL Server Enterprise	RunInstances:0110
SQL Server Enterprise	RunInstances:0100
SQL Server Standard	RunInstances:0004
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Ubuntu Pro	RunInstances:0g00
Windows	RunInstances:0002
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise[1]	RunInstances:0102
Windows with SQL Server Standard[1]	RunInstances:0006
Windows with SQL Server Web[1]	RunInstances:0202

[1] 如果两个软件许可证与 AMI 关联，则平台详细信息字段将显示两者。

[2] 如果您正在运行竞价型实例，则 Amazon 成本和使用情况报告上的 lineitem/Operation 可能与此处列出的使用情况操作值不同。例如，如果 `lineitem/Operation` 显示 `RunInstances:0010:SV006`，则表示 Amazon EC2 正在区域 6 中的美国东部（弗吉尼亚州北部）运行 Red Hat Enterprise Linux 竞价型实例小时。

[3] 这在使用情况报告中显示为 RunInstances (Linux/UNIX)。

查找 AMI 账单和使用详细信息

以下属性可以帮助您验证账单上的 AMI 费用：

- 平台详细信息
- 使用情况操作
- AMI ID

Console

查找 AMI 的 AMI 账单信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡上，查找平台详细信息和使用情况操作。

查找实例的 AMI 账单信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 在详细信息选项卡上，展开实例详细信息，然后查找平台详细信息和使用情况操作。

Amazon CLI

查找 AMI 的 AMI 账单信息

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query "Images[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

以下是 Linux AMI 的示例输出。

```
[
    {
        "PlatformDetails": "Linux/UNIX",
        "UsageOperation": "RunInstances"
    }
]
```

查找实例的 AMI 账单信息

可以使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].
{PlatformDetails:PlatformDetails,UsageOperation:UsageOperation}"
```

以下是 Windows 实例的示例输出。

```
[
    {
        "PlatformDetails": "Windows",
        "UsageOperation": "RunInstances:0002"
    }
]
```

PowerShell

查找 AMI 的 AMI 账单信息

使用 Get-EC2Image cmdlet。

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Format-List PlatformDetails, UsageOperation
```

以下是 Linux AMI 的示例输出。

```
PlatformDetails : Linux/UNIX
UsageOperation  : RunInstances
```

查找实例的 AMI 账单信息

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances | `
    Format-List PlatformDetails, UsageOperation
```

以下是 Windows 实例的示例输出。

```
PlatformDetails : Windows
UsageOperation  : RunInstances:0002
```

验证账单上的 AMI 费用

为确保您不会产生计划外费用，您可以验证 Amazon 成本和使用情况报告 (CUR) 中某个实例的账单信息和与您用于启动实例的 AMI 关联的账单信息匹配。

要验证账单信息，请在 CUR 中找到实例 ID 并检查 `lineitem/Operation` 列中的相应值。该值应和与 AMI 关联的 Usage operation (使用情况操作) 的值相匹配。

例如，AMI `ami-0123456789EXAMPLE` 具有以下账单信息：

- Platform details (平台详细信息) = `Red Hat Enterprise Linux`
- Usage operation (使用情况操作) = `RunInstances:0010`

如果您使用此 AMI 启动了实例，您可以在 CUR 中找到实例 ID 并检查 `lineitem/Operation` 列中的相应值。在这个例子中，值应该是 `RunInstances:0010`。

Amazon EC2 中的 AMI 配额

以下限额适用于创建和共享 AMI。限额按 Amazon Web Services 区域应用。

限额名称	描述	每个区域的默认限额
AMI	每个区域允许的最大公有和私有 AMI 数量。这包括可用、待处理和已禁用的 AMI 以及回收站中的 AMI。	50000
公有 AMI	每个区域允许的最大公有 AMI 数量，包括回收站中的公有 AMI。	5
AMI 共享	某一区域内可以共享 AMI 的最大实体（组织、组织部门 (OU) 和账户）数量。请注意，如果您与某个组织或 OU 共享一个 AMI，则该组织或 OU 中的账户数量不计入该限额。	1000

如果您超出限额，但想创建或共享更多 AMI，可以执行以下操作：

- 如果您超出总 AMI 或公有 AMI 限额，请考虑取消注册未使用的映像。
- 如果您超出公有 AMI 限额，请考虑将一个或多个公有 AMI 设为私有。
- 如果您超出 AMI 共享限额，请考虑与某个组织或 OU（而不是单独的账户）共享您的 AMI。
- 请求增加 AMI 的限额。

请求增加 AMI 的限额

如果您需要的 AMI 超出其默认限额，您可以请求增加限额。

请求增加 AMI 的限额

1. 访问 https://console.aws.amazon.com/servicequotas/，打开服务限额控制台。
2. 在导航窗格中，选择 Amazon 服务。
3. 从列表中选择 Amazon Elastic Compute Cloud (Amazon EC2)，或在搜索框中键入服务的名称。
4. 选择 AMI 限额以请求增加限额。您可以选择的 AMI 限额包括：

- AMI
- 公有 AMI
- AMI 共享

5. 选择请求增加限额。
6. 对于 Change quota value（更改限额值），输入新的限额值，然后选择 Request（请求）。

要查看任何待处理或最近解决的请求，请从导航窗格选择 Dashboard (控制面板)。对于待处理的请求，请选择请求状态以打开收到的请求。请求的初始状态为 Pending（待处理）。状态更改为 Quota requested（已请求限额）后，您将在 Support Center case number（支持中心案例编号）下看到案例编号。选择案例编号以打开请求服务单。

解决请求后，配额的 Applied quota value (应用的配额值) 设置为新值。

有关更多信息，请参阅 Service Quotas 用户指南。

Amazon EC2 实例

Amazon EC2 实例是 Amazon 云环境中的虚拟服务器。从首次启动实例（称为启动实例）到将其删除（称为终止实例），您可以完全控制自己的实例。启动实例时，您可以从各种操作系统中进行选择。您可以连接到您的实例并对其进行自定义以满足您的需求。例如，您可以在实例上配置操作系统、安装操作系统更新和安装应用程序。

Amazon EC2 提供各种实例类型。您可以选择一种实例类型，以提供运行应用程序所需的计算资源、内存、存储和网络性。

使用 Amazon EC2，您可以按实际用量付费。当您启动实例并转换到运行状态时，实例计费开始。当您停止实例时，计费停止，并在您启动实例后恢复计费。当您终止实例时，计费在实例转换到关闭状态时停止。

Amazon EC2 提供可用于优化实例的性能和成本的功能。例如，您可以使用 Amazon EC2 Fleet 或 Amazon EC2 Auto Scaling 在实例使用率发生变化时向上或向下扩展容量。您可以使用竞价型实例或节省计划来降低实例的成本。

托管式实例是指由服务提供商（例如 Amazon EKS 自动模式）管理的实例。您不能直接修改托管式实例的设置。托管式实例由托管字段中的 true 进行标识。有关更多信息，请参阅 Amazon EC2 托管式实例。

功能和任务

- Amazon EC2 实例类型
- Amazon EC2 托管式实例
- 利用嵌套虚拟化在 Amazon EC2 实例中运行虚拟机监控程序
- Amazon EC2 账单和购买选项
- 在 Amazon EC2 启动模板中存储实例启动参数
- 启动 Amazon EC2 实例
- 连接到您的 EC2 实例
- Amazon EC2 实例状态更改
- 实例自动恢复
- 使用实例元数据管理 EC2 实例
- 检测主机是否为 EC2 实例
- Amazon EC2 实例的实例身份文档

- EC2 实例的 STIG 合规性
- 在 EC2 实例上实现精确时钟和时间同步
- 管理 EC2 实例的设备驱动程序
- 配置您的 Amazon EC2 Windows 实例
- 将 EC2 Windows 实例升级到更高版本的 Windows Server
- Tutorial: Connect an Amazon EC2 instance to an Amazon RDS database（教程：将 Amazon EC2 实例连接到 Amazon RDS 数据库）

Amazon EC2 实例类型

启动实例时，您指定的实例类型 决定了用于您的实例的主机硬件。每个实例类型提供不同的计算、内存和存储功能，并按照这些功能分组到实例系列。选择一种基于您打算在实例上运行的应用程序或软件的需求的实例类型。有关功能和使用案例的更多信息，请参阅 Amazon EC2 实例类型。

Amazon EC2 会将主机的一些资源（例如 CPU、内存和实例存储）专用于特定实例。Amazon EC2 在实例间共享主机的其他资源，例如网络和磁盘子系统。如果一个主机上的每个实例都试图尽可能多地使用这些共享的资源，那么每个实例都将获得该资源相等份额。但是，当某个资源利用不充分时，会有实例会在该资源可用时消耗其更多的份额。

每种实例类型均从共享资源提供更高或更低的起始性能。例如，高 I/O 性能的实例类型能获取共享资源的更高份额。分配更大份额的共享资源也降低了 I/O 性能的方差。对于大多数应用程序，中等 I/O 是绰绰有余的。然而，对于需要更大或一致性更高的 I/O 性能的应用程序，可考虑使用更高 I/O 性能的实例类型。

目录

- 可用实例类型
- 硬件规格
- 虚拟机监控程序类型
- AMI 虚拟化类型
- 处理器
- 查找 Amazon EC2 实例类型
- 从 EC2 实例类型查找器获取建议
- 从 Compute Optimizer 获取 EC2 实例建议
- Amazon EC2 实例类型更改

- 可突增性能实例
- 使用 GPU 实例加速性能
- Amazon EBS 优化的实例类型
- Amazon EC2 实例适用的 CPU 选项
- Amazon EC2 实例的 AMD SEV-SNP
- Amazon EC2 Linux 实例的处理器状态控制

可用实例类型

Amazon EC2 提供各种不同的实例类型，这些实例类型经过优化，适合不同的使用案例。实例类型包括 CPU、内存、存储和网络容量的不同组合，便于您灵活选择适合应用程序的资源组合。每种实例类型都包含一个或多个实例大小，您可以根据目标工作负载的要求扩展资源。

实例类型命名约定

名称基于实例系列、代系、处理器系列、功能和型号。有关更多信息，请参阅《Amazon EC2 实例类型指南》中的命名约定。

查找实例类型

要确定哪些实例类型满足您的要求（例如支持的区域、计算资源或存储资源），请参阅 查找 Amazon EC2 实例类型 和《Amazon EC2 Instance Types Guide》中的 Amazon EC2 instance type specifications。

硬件规格

有关详细的实例类型规格，请参阅《Amazon EC2 实例类型指南》中的规格。有关定价信息，请参阅 Amazon EC2 按需定价。

要确定最适合您的需求的实例类型，我们建议启动一个实例，并使用自己的基准测试应用程序。由于您是按实例秒付费的，因此在做出决策前测试多个实例类型将会既方便又经济。如果您的需求有变化，甚至是在做出决策后，您可以在以后更改您的实例类型。有关更多信息，请参阅 Amazon EC2 实例类型更改。

虚拟机监控程序类型

Amazon EC2 支持以下虚拟机监控程序：Xen 和 Nitro。

基于 Nitro 的实例

- 通用型：M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M7i-flex | M8a | M8azn | M8g | M8gb | M8gd | M8gn | M8i | M8id | M8i-flex | M8in | M8idn | M8ine | M8ib | M8idb | M9g | M9gd | T3 | T3a | T4g
- 计算优化型：C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i | C7i-flex | C8a | C8g | C8gb | C8gd | C8gn | C8i | C8id | C8i-flex | C8in | C8ine | C8ib
- 内存优化型：R5 | R5a | R5ad | R5b | R5d | R5dn | R5n | R6a | R6g | R6gd | R6i | R6id | R6idn | R6in | R7a | R7g | R7gd | R7i | R7iz | R8a | R8g | R8gb | R8gd | R8gn | R8i | R8id | R8i-flex | R8in | R8idn | R8ib | R8idb | U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | U-24tb1 | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X2gd | X2idn | X2iedn | X2iezn | X8g | X8aedz | X8i | z1d
- 存储优化型：D3 | D3en | I3en | I4g | I4i | I7i | I7ie | I8g | I8ge | Im4gn | Is4gen
- 加速计算型：DL1 | DL2q | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | G6f | Gr6 | Gr6f | G7e | Inf1 | Inf2 | P4d | P4de | P5 | P5e | P5en | P6-B200 | P6-B300 | P6e-GB200 | Trn1 | Trn1n | Trn2 | Trn2u | VT1
- 高性能计算：Hpc6a | Hpc6id | Hpc7a | Hpc7g | Hpc8a
- 上一代实例：A1 | P3dn

要详细了解支持的 Nitro 虚拟机监控器版本，请参阅 Amazon EC2 实例类型指南中的网络功能支持。

基于 Xen 的实例

- 通用型：M1 | M2 | M3 | M4 | T1 | T2
- 计算优化型：C1 | C3 | C4
- 内存优化型：R3 | R4 | X1 | X1e
- 存储优化型：D2 | H1 | I2 | I3
- 加速计算型：G3 | P3

AMI 虚拟化类型

实例的虚拟化类型由用于启动该实例的 AMI 决定。最新一代实例类型仅支持硬件虚拟机 (HVM)。某些上一代实例类型支持半虚拟化 (PV)，某些 Amazon 区域支持半虚拟化实例。有关更多信息，请参阅虚拟化类型。

为获得最佳性能，我们建议您使用 HVM AMI。此外，HVM AMI 还需要利用增强联网。HVM 虚拟化使用 Amazon 平台提供的硬件辅助技术。借助 HVM 虚拟化，访客虚拟机如同在本地硬件平台上运行一样，除了仍然使用半虚拟 (PV) 网络和存储驱动程序以提高性能。

处理器

EC2 实例支持多种处理器。

处理器

- 英特尔处理器
- AMD 处理器
- Amazon Graviton 处理器
- Amazon Trainium
- Amazon Inferentia

英特尔处理器

在 Intel 处理器上运行的 Amazon EC2 实例可能包括以下处理器功能。并非所有在 Intel 处理器上运行的实例都支持所有这些处理器功能。有关每种实例类型有哪些功能可用的信息，请参阅 Amazon EC2 实例类型。

- Intel AES New Instructions (AES-NI) — Intel AES-NI 加密指令集改进了原先 Advanced Encryption Standard (AES) 的算法，可以提供更快的数据保护和更好的安全性。所有最新一代 EC2 实例都支持此处理器功能。
- Intel Advanced Vector Extensions（Intel AVX、Intel AVX2 和 Intel AVX-512）— ：Intel AVX/Intel AVX2 和 Intel AVX-512 分别是 256 位和 512 位指令集扩展，专为浮点 (FP) 密集型应用程序而设计。Intel AVX 指令可以提升诸如图像和音频/视频处理、科学模拟、财务分析、3D 建模与分析等应用的性能。这些功能仅在使用 HVM AMI 启动的实例上可用。
- Intel 睿频加速技术 — Intel 睿频加速技术处理器以比基本操作频率快的速度自动运行核心。
- Intel Deep Learning Boost (Intel DL Boost) — 加速 AI 深度学习使用案例。第二代 Intel Xeon 可扩展处理器能够扩展采用全新向量神经网络指令 (VNNI/INT8) 的 Intel AVX-512，与上一代 Intel Xeon 可扩展处理器（配有 FP32）相比，深度学习推理性能显著提高，适用于图像识别/分割、对象检测、语音识别、语言翻译、推荐系统和强化学习等。VNNI 可能不会兼容所有 Linux 发行版。

  以下实例支持 VNNI：`M5n`、`R5n`、`M5dn`、`M5zn`、`R5b`、`R5dn`、`D3`、`D3en` 和 `C6i`。`C5` 和 `C5d` 实例仅支持 `12xlarge`、`24xlarge` 和 `metal` 实例的 VNNI。

64 位 CPU 的行业命名惯例可能会导致混淆。芯片制造商 Advanced Micro Devices (AMD) 成功引入了第一款基于 Intel x86 指令集的商用 64 位架构。因此，不论芯片制造商是谁，这一架构被普遍称为 AMD64。Windows 和多个 Linux 发行版遵循这一实践。这说明了为什么实例即使运行在 Intel 硬件上，但 Ubuntu 或 Windows 上的内部系统信息仍将 CPU 架构显示为 AMD64。

AMD 处理器

在 AMD EPYC 处理器上运行的 Amazon EC2 实例有助您优化工作负载的成本和性能。此类实例可能支持以下处理器功能。并非所有在 AMD 处理器上运行的实例都支持所有这些处理器功能。有关每种实例类型有哪些功能可用的信息，请参阅 Amazon EC2 实例类型。

- AMD 安全内存加密（SME）
- AMD 透明单密钥内存加密（TSME）
- AMD 高级向量扩展（AVX）
- AMD 安全加密虚拟化安全嵌套分页（SEV-SNP）
- 向量神经网络指令（VNNI）
- BFloat16

Amazon Graviton 处理器

Amazon Graviton 是一系列处理器，旨在为在 Amazon EC2 实例上运行的工作负载提供最佳的性价比。

有关更多信息，请参阅 Getting started with Graviton。

Amazon Trainium

由 Amazon Trainium 提供支持的实例专为高性能、经济实惠的深度学习训练而构建。您可以使用这些实例来训练在语音识别、推荐、欺诈检测以及图像和视频分类等各种应用程序中使用的自然语言处理、计算机视觉和推荐器模型。在流行的机器学习（ML）框架（如 PyTorch 和 TensorFlow）中使用您的现有工作流程。

Amazon Inferentia

由 Amazon Inferentia 提供支持的实例旨在加速机器学习。其提供高性能、低延迟的机器学习推理。这些实例经过优化，专用于部署各种应用领域的深度学习 (DL) 模型，例如自然语言处理、对象检测和分类、内容个性化和过滤以及语音识别。

您可以通过多种方式开始使用这些模型：

- 使用 SageMaker AI（这是一种完全托管式服务）是开始使用机器学习模型的最简单方法。有关更多信息，请参阅《Amazon SageMaker AI Developer Guide》中的 Get Started with SageMaker AI。
- 使用 Deep Learning AMI 启动 Inf1 或 Inf2 实例。有关更多信息，请参阅 Amazon Deep Learning AMIs 开发人员指南中的使用 DLAMI 的 Amazon Inferentia。
- 使用您自己的 AMI 启动 Inf1 或 Inf2 实例并安装 Amazon Neuron 开发工具包，您可以利用此工具包为 Amazon Inferentia 编译、运行和分析深度学习模型。
- 结合使用 Inf1 或 Inf2 实例和经过 Amazon ECS 优化的 AMI 启动容器实例。有关更多信息，请参阅《Amazon Elastic Container Service 开发人员指南》中的 Amazon Linux 2（Inferentia）AMI。
- 创建包含运行 Inf1 实例的节点的 Amazon EKS 集群。有关更多信息，请参阅 Amazon EKS 用户指南中的 Inferentia 支持。

查找 Amazon EC2 实例类型

您必须先选择要使用的实例类型，然后才能启动实例。您选择的实例类型可能取决于您的工作负载所需的资源，例如计算、内存或存储资源。确定几种可能适合您的工作负载的实例类型，并在测试环境中评估它们的性能可能很有帮助。没有方法可以替代衡量应用程序在负载下的性能。

您可以使用 EC2 实例类型查找器获取有关 EC2 实例类型的建议和指导。有关更多信息，请参阅 the section called “EC2 实例类型查找器”。

如果您已在运行 EC2 实例，可以使用 Amazon Compute Optimizer 获取有关您应该用来提高性能、节省资金或两者兼顾的实例类型的建议。有关更多信息，请参阅 the section called “Compute Optimizer 建议”。

任务

- 使用控制台查找实例类型
- 使用 Amazon CLI 描述实例类型
- 使用 Amazon CLI 查找实例类型
- 使用 Tools for PowerShell 来查找实例类型

使用控制台查找实例类型

您可以使用 Amazon EC2 控制台查找满足您的需求的实例类型。

使用控制台查找实例类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 从导航栏，选择您在其中启动实例的区域。您可以选择向您提供的任何区域，无需理会您身处的位置。
3. 在导航窗格中，选择 Instance Types (实例类型)。
4. （可选）选择首选项（齿轮）图标以选择要显示的实例类型属性（例如 On-Demand Linux pricing (按需 Linux 定价)），然后选择 Confirm (确认)。或者，选择实例类型的名称以打开其详细信息页面，并查看通过控制台提供的所有属性。控制台不会显示通过 API 或命令行提供的所有属性。
5. 使用实例类型属性筛选显示的实例类型列表，以仅显示满足您的需求的实例类型。例如，您可以筛选以下属性：

   - Availability zones（可用区）– 可用区、本地区域或 Wavelength 区域的名称。有关更多信息，请参阅 the section called “区域和可用区”。
   - vCPUs 或 Cores（内核）– vCPU 或内核的数量。
   - Memory (GiB) [内存（GiB)] - 内存大小，以 GiB 为单位。
   - Network performance（网络性能）– 网络性能，以千兆为单位。
   - Local instance storage（本地实例存储）– 指示实例类型是否具有本地实例存储（`true` | `false`）。
6. （可选）要并排查看比较结果，请选中多个实例类型的复选框。比较结果显示在屏幕底部。
7. （可选）要将实例类型列表保存到逗号分隔值（.csv）文件以进一步查看，请依次选择 Actions（操作）、Download list CSV（下载列表 CSV）。该文件包括与您设置的筛选条件匹配的所有实例类型。
8. （可选）要使用符合您需求的实例类型启动实例，请选中该实例类型的复选框并依次选择 Actions（操作）、Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

使用 Amazon CLI 描述实例类型

您可以使用 describe-instance-types 命令来描述特定的实例类型。

完整描述实例类型

以下命令显示了指定实例类型的所有可用详细信息。输出很长，所以此处进行了省略。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2
```

描述实例类型并筛选输出

以下命令显示了指定实例类型的网络详细信息。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].NetworkInfo"
```

下面是示例输出。

```
[
    {
        "NetworkPerformance": "Low to Moderate",
        "MaximumNetworkInterfaces": 2,
        "MaximumNetworkCards": 1,
        "DefaultNetworkCardIndex": 0,
        "NetworkCards": [
            {
                "NetworkCardIndex": 0,
                "NetworkPerformance": "Low to Moderate",
                "MaximumNetworkInterfaces": 2,
                "BaselineBandwidthInGbps": 0.064,
                "PeakBandwidthInGbps": 1.024
            }
        ],
        "Ipv4AddressesPerInterface": 2,
        "Ipv6AddressesPerInterface": 2,
        "Ipv6Supported": true,
        "EnaSupport": "unsupported",
        "EfaSupported": false,
        "EncryptionInTransitSupported": false,
        "EnaSrdSupported": false
    }
]
```

以下命令显示了指定实例类型的可用内存。

```
aws ec2 describe-instance-types \
    --instance-types t2.micro \
    --region us-east-2 \
    --query "InstanceTypes[].MemoryInfo"
```

下面是示例输出。

```
[
    {
        "SizeInMiB": 1024
    }
]
```

使用 Amazon CLI 查找实例类型

您可以使用 describe-instance-types 和 describe-instance-type-offerings 命令来查找满足需求的实例类型。

示例

- 示例 1：按可用区查找实例类型
- 示例 2：按可用内存大小查找实例类型
- 示例 3：按可用实例存储查找实例类型
- 示例 4：查找支持休眠的实例类型

示例 1：按可用区查找实例类型

以下示例仅显示了在指定可用区中提供的实例类型。

```
aws ec2 describe-instance-type-offerings \
    --location-type "availability-zone" \
    --filters "Name=location,Values=us-east-2a" \
    --region us-east-2 \
    --query "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

输出是按字母顺序排序的实例类型列表。下面只是输出的开始部分。

```
a1.2xlarge
a1.4xlarge
a1.large
a1.medium
a1.metal
a1.xlarge
c4.2xlarge
   ...
```

示例 2：按可用内存大小查找实例类型

以下示例仅显示了当前一代中具有 64 GiB（65536 MiB）内存的实例类型。

```
aws ec2 describe-instance-types \
    --filters "Name=current-generation,Values=true" "Name=memory-info.size-in-
mib,Values=65536" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

输出是按字母顺序排序的实例类型列表。下面只是输出的开始部分。

```
c5a.8xlarge
c5ad.8xlarge
c6a.8xlarge
c6g.8xlarge
c6gd.8xlarge
c6gn.8xlarge
c6i.8xlarge
c6id.8xlarge
c6in.8xlarge
   ...
```

示例 3：按可用实例存储查找实例类型

以下示例显示了包含实例存储卷的所有 R7 实例的实例存储总大小。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r7*" "Name=instance-storage-
supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[].[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
    --output table
```

下面是示例输出。

```
---------------------------
|  DescribeInstanceTypes  |
+----------------+--------+
|  r7gd.xlarge   |  237   |
|  r7gd.8xlarge  |  1900  |
|  r7gd.16xlarge |  3800  |
```

```
|  r7gd.medium   |  59    |
|  r7gd.4xlarge  |  950   |
|  r7gd.2xlarge  |  474   |
|  r7gd.metal    |  3800  |
|  r7gd.large    |  118   |
|  r7gd.12xlarge |  2850  |
+----------------+--------+
```

示例 4：查找支持休眠的实例类型

以下示例显示支持休眠的实例类型。

```
aws ec2 describe-instance-types \
    --filters "Name=hibernation-supported,Values=true" \
    --region us-east-2 \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

输出是按字母顺序排序的实例类型列表。下面只是输出的开始部分。

```
c4.2xlarge
c4.4xlarge
c4.8xlarge
c4.large
c4.xlarge
c5.12xlarge
c5.18xlarge
c5.2xlarge
c5.4xlarge
c5.9xlarge
...
```

使用 Tools for PowerShell 来查找实例类型

您可以使用 Get-EC2InstanceType 和 Get-EC2InstanceTypeOffering cmdlet 来查找满足您需求的实例类型。

示例

- 按可用区查找实例类型
- 按可用内存大小查找实例类型
- 按可用实例存储查找实例类型

- 查找支持休眠的实例类型

按可用区查找实例类型

以下示例仅显示了在指定可用区中提供的实例类型。

```
(Get-EC2InstanceTypeOffering `
    -LocationType "availability-zone" `
    -Region us-east-2 `
    -Filter @{Name="location"; Values="us-east-2a"}).InstanceType | Sort-Object `
```

按可用内存大小查找实例类型

以下示例仅显示了当前一代中具有 64 GiB（65536 MiB）内存的实例类型。

```
(Get-EC2InstanceType `
    -Filter @{Name="current-generation"; Values="true"},
            @{Name="memory-info.size-in-mib"; Values="65536"}).InstanceType | Sort-
Object
```

按可用实例存储查找实例类型

以下示例显示了包含实例存储卷的所有 R7 实例的实例存储总大小。

```
Get-EC2InstanceType `
    -Filter @{Name="instance-type"; Values="r7*"},
            @{Name="instance-storage-supported"; Values="true"} | `
     Select InstanceType, @{Name="TotalSizeInGB";
 Expression={($_.InstanceStorageInfo.TotalSizeInGB)}}
```

下面是示例输出。

```
InstanceType  TotalSizeInGB
------------  -------------
r7gd.8xlarge           1900
r7gd.16xlarge          3800
r7gd.xlarge             237
r7gd.4xlarge            950
r7gd.medium              59
r7gd.2xlarge            474
r7gd.large              118
r7gd.metal             3800
```

```
r7gd.12xlarge          2850
```

查找支持休眠的实例类型

以下示例显示支持休眠的实例类型。

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-Object
```

从 EC2 实例类型查找器获取建议

EC2 实例类型查找器会考虑您的使用案例、工作负载类型和 CPU 制造商偏好，您优先考虑价格还是性能，以及您可以指定的其他参数。随后，它利用这些数据提供有关最适合新工作负载的 Amazon EC2 实例类型的建议和指导。

有如此多的实例类型可供使用，因此找到适用于您工作负载的实例类型可能非常耗时和复杂。使用 EC2 实例类型查找器，您可以随时获得最新的实例类型，并为您的工作负载实现最佳性价比。

您可以使用 Amazon EC2 控制台获取有关 EC2 实例类型的建议和指导。您也可以直接转到 Amazon Q，以寻求实例类型建议。有关更多信息，请参阅《Amazon Q 开发者版用户指南》https://docs.amazonaws.cn/amazonq/latest/qdeveloper-ug/what-is.html。

如果您正在为现有的工作负载寻找实例类型建议，请使用 Amazon Compute Optimizer。有关更多信息，请参阅 从 Compute Optimizer 获取 EC2 实例建议。

使用 EC2 实例类型查找器

在 Amazon EC2 控制台中，您可以在启动实例向导中、创建启动模板时或在实例类型页面上，从 EC2 实例类型查找器获取实例类型建议。

请按照以下说明，使用 Amazon EC2 控制台中的 EC2 实例类型查找器获得 EC2 实例类型建议和指导。要观看这些步骤的动画，请参阅 观看动画：使用 EC2 实例类型查找器获取实例类型建议。

使用 EC2 实例类型查找器获取实例类型建议

1. 使用以下任一方法开始您的流程：
   - 按照程序启动实例。在实例类型的旁边，选择获取建议链接。
   - 按照该程序创建启动模板。在实例类型的旁边，选择获取建议链接。
   - 在导航窗格中，选择实例类型，然后选择实例类型查找器按钮。
2. 在获取有关实例类型选择的建议屏幕中，执行以下操作：

a. 通过选择工作负载类型、使用案例、优先级和 CPU 制造商选项来指定您的实例类型要求。

b. （可选）要为您的工作负载指定更详细的要求，请执行以下操作：

   i. 展开高级参数。

   ii. 要添加参数，请选择一个参数，选择添加，然后为该参数指定一个值。对每个想要添加的附加参数重复上述操作。要表示没有最小值或最大值，请将该字段留空。

   iii. 要在添加参数后将其删除，请选择该参数旁边的 X。

c. 选择获取实例类型建议。

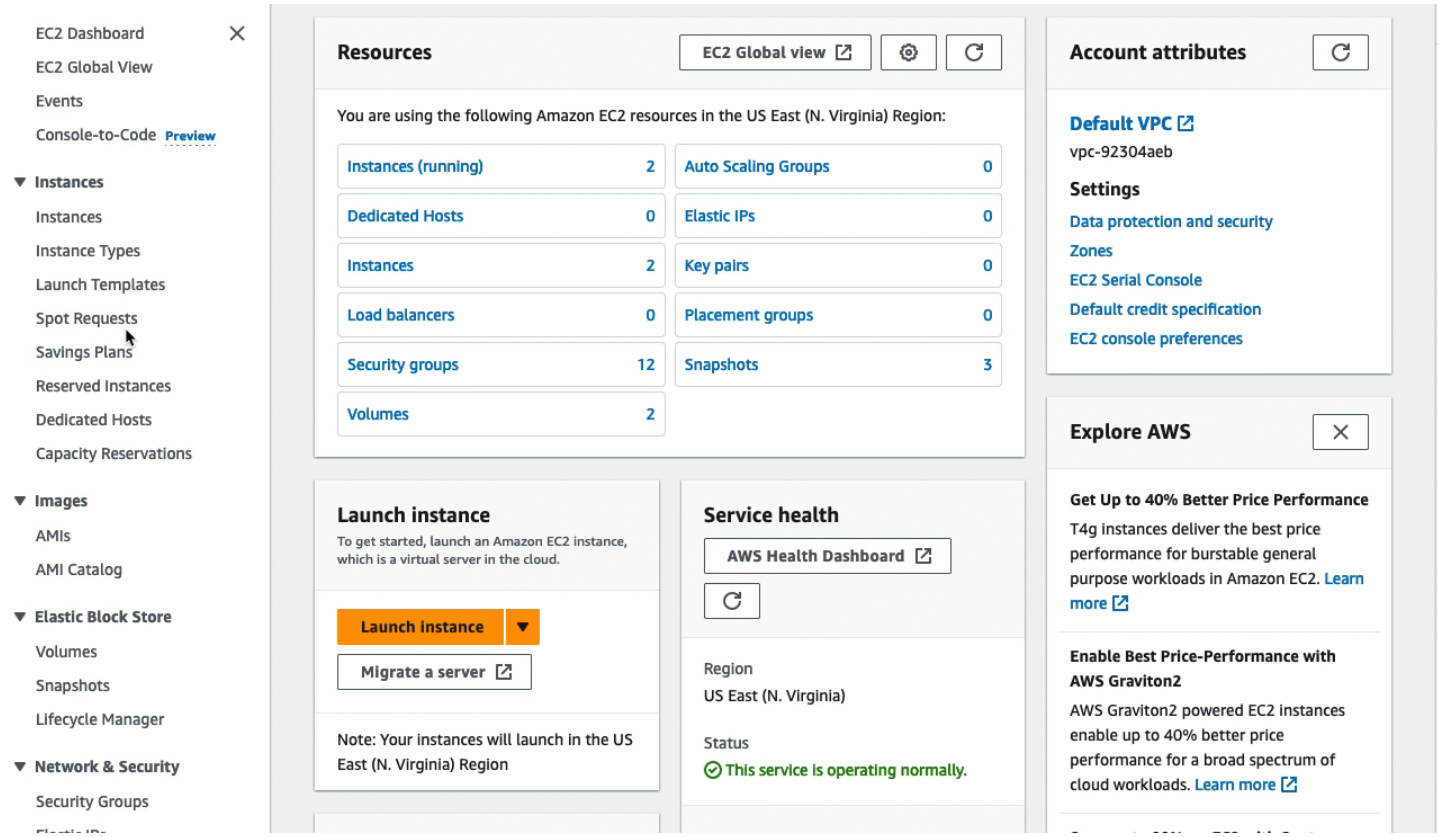
   Amazon EC2 为您提供符合您指定要求的实例系列的建议。

3. 要查看建议的实例系列中每种实例类型的详细信息，请选择查看推荐的实例系列详细信息。

4. 选择符合您要求的实例类型，然后选择操作、启动实例或操作、创建启动模板。

或者，如果您在启动实例向导或启动模板页面中启动了该流程，并且希望返回到原来的流程，则请记下您要使用的实例类型。然后，在启动实例向导或启动模板中，为实例类型选择该实例类型，然后完成启动实例或创建启动模板的过程。

观看动画：使用 EC2 实例类型查找器获取实例类型建议



从 Compute Optimizer 获取 EC2 实例建议

Amazon Compute Optimizer 提供了 Amazon EC2 建议，以帮助您提高性能和/或节省资金。您可以根据这些建议来决定是否更改为新的实例类型。

为了生成建议，Compute Optimizer 会分析现有实例规范和利用率指标。然后，利用已编译数据来建议哪些 Amazon EC2 实例类型能够最好地处理现有工作负载。建议随每小时实例定价一起返回。有关更多信息，请参阅《Amazon Compute Optimizer User Guide》中的 Amazon EC2 instance metrics。

内容

- 要求
- 结果分类
- 查看建议
- 评估建议时的注意事项

要求

要从 Compute Optimizer 中获取建议，您必须首先选择加入 Compute Optimizer。有关更多信息，请参阅 Amazon Compute Optimizer 用户指南中的 Amazon Compute Optimizer 入门。

Compute Optimizer 会为某些实例类型生成建议，但不会为所有实例类型生成建议。如果您使用不支持的实例类型，Compute Optimizer 将不会生成建议。有关支持的实例类型的列表，请参阅《Amazon Compute Optimizer User Guide》中的 Amazon EC2 instance requirements。

结果分类

Compute Optimizer 将其对 EC2 实例的调查结果分类为：

- 预配置不足 – 当您的实例的至少一个规格（如 CPU、内存或网络）没有满足工作负载的性能要求时，将 EC2 实例视为预配置不足。预配置不足的 EC2 实例可能会导致应用程序性能较差。
- 过度预配置 – 当您的实例的至少一个规格（如 CPU、内存或网络）可缩小但仍能满足工作负载的性能要求时，并且没有任何规格处于预配置不足状态时，将 EC2 实例视为过度预配置。过度预配置的 EC2 实例可能会导致不必要的基础设施成本。
- 已优化 – 当您的实例的所有规格（如 CPU、内存和网络）满足工作负载的性能要求且实例未处于过度预配置状态时，将 EC2 实例视为已优化。已优化的 EC2 实例以最佳的性能和基础设施成本运行您的工作负载。对于已优化的实例，Compute Optimizer 有时可能会建议新一代实例类型。
- 无 – 没有对此实例的建议。如果您选择加入 Compute Optimizer 的时间少于 12 小时、实例的运行时间少于 30 小时，或者 Compute Optimizer 不支持实例类型，则可能会发生这种情况。

查看建议

在选择加入 Compute Optimizer 后，您可以在 Amazon EC2 控制台中查看 Compute Optimizer 为 EC2 实例生成的结果。然后，您可以访问 Compute Optimizer 控制台来查看建议。如果您是最近选择加入的，EC2 控制台可能在长达 12 小时内不会反映调查结果。

使用 Amazon EC2 控制台查看实例的建议

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例 ID 以打开实例详细信息页面。
4. 在实例详细信息页面的顶部摘要部分，找到 Amazon Compute Optimizer 调查发现。如果有调查发现，我们会显示调查发现分类和查看详细信息的链接。否则，我们将显示此实例没有可用的建议。
5. 如果有调查发现，请选择查看详细信息。这将打开 Compute Optimizer 控制台中的 EC2 实例建议页面。当前实例类型标记为当前。还有最多三个实例类型建议，分别标记为选项 1、选项 2 和选项 3。此页面还显示了该实例的最新 CloudWatch 指标数据。

查看对所有区域中所有实例的建议

您可以使用 Compute Optimizer 控制台查看所有区域中所有 Amazon EC2 实例的建议。有关更多信息，请参阅《Amazon Compute Optimizer User Guide》中的 Viewing EC2 instances recommendations 和 Viewing EC2 instance details。

评估建议时的注意事项

在收到建议时，您必须决定是否执行操作。在更改实例类型之前，请考虑以下事项：

- 这些建议不会预测您的使用情况。建议基于您在最近 14 天时间段内的历史使用情况。请务必选择一种预计能够满足您的未来资源需求的实例类型。
- 关注图表指标以确定实际使用量是否低于实例容量。您还可以在 CloudWatch 中查看指标数据（平均值、峰值、百分比），以进一步评估 EC2 实例建议。例如，观察当天 CPU 百分比指标如何变化，以及是否有需要满足的峰值。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的查看可用指标。
- Compute Optimizer 可能会为可突增性能实例（即 T3、T3a 和 T2 实例）提供建议。如果您定期突增至基线之上，请确保您可以基于新实例类型的 vCPU 继续如此。有关更多信息，请参阅可突增性能实例的关键概念。

- 如果您购买了 Reserved Instance，您的个按需型实例可能会作为 Reserved Instance 进行计费。在更改当前实例类型之前，请首先评估对Reserved Instance使用率和覆盖率的影响。
- 尽可能考虑转换为较新一代实例。
- 在迁移到其他实例系列时，请确保当前实例类型和新实例类型在虚拟化、架构或网络类型等方面兼容。有关更多信息，请参阅更改实例类型的兼容性。
- 最后，请考虑为每个建议提供的性能风险评级。性能风险指示您为了验证建议的实例类型是否满足工作负载的性能要求而可能需要执行的工作量。我们还建议在进行任何更改前后进行严格的负载和性能测试。

Amazon EC2 实例类型更改

随着您的需求变化，您可能会发现您的实例过度使用 (实例类型过小) 或利用不足 (实例类型过大)。如果是这种情况，您可以通过更改实例类型来调整实例的大小。例如，如果您 `t2.micro` 实例对于其工作负载来说太小，您可以通过将其更改为更大的 T2 实例类型来增加其大小，例如 `t2.large`。或者，您可以将其更改其它实例类型，如 `m5.large`。您可能还想将实例类型从上一代更改为最新一代以利用某些功能，例如 IPv6 支持。

如果您想要针对能够最好地处理现有工作负载的实例类型获得建议，则可以使用 Amazon Compute Optimizer。有关更多信息，请参阅 从 Compute Optimizer 获取 EC2 实例建议。

更改实例类型时，您将开始支付新实例类型的费率。有关所有实例类型的按需费率，请参阅 Amazon EC2 按需定价。

要在不更改实例类型的情况下，向实例添加额外的存储空间，请向实例添加一个 EBS 卷。有关更多信息，请参阅《Amazon EBS 用户指南》中的将 Amazon EBS 卷挂载到实例。

要遵循哪些说明？

对于更改实例类型，我们有不同的说明。应该使用哪些说明取决于实例的根卷，以及实例类型是否与实例的当前配置兼容。有关如何确定兼容性的信息，请参阅 更改实例类型的兼容性。

使用下表确定需要遵守的说明。

根卷	兼容性	使用以下说明进行操作
EBS	兼容	更改实例类型
EBS	不兼容	迁移到新实例类型

根卷	兼容性	使用以下说明进行操作
实例存储	不适用	迁移到新实例类型

更改实例类型的兼容性

仅当实例的当前配置与您所需的实例类型兼容时，才能更改实例类型。如果您所需的实例类型与实例的当前配置不兼容，则您必须启动一个具有与新实例类型兼容的配置的新实例，并将应用程序迁移到新实例。

可通过以下方式确定兼容性：

虚拟化类型

Linux AMI 使用两种虚拟化之一：半虚拟化 (PV) 或硬件虚拟机 (HVM)。如果是从 PV AMI 启动的实例，则您无法将其更改为仅限 HVM 的实例类型。有关更多信息，请参阅 虚拟化类型。要查看实例的虚拟化类型，请在 Amazon EC2 控制台中查看 Instances（实例）屏幕的详细信息窗格中的 Virtualization（虚拟化）值。

架构

AMI 特定于处理器的架构，因此您必须选择与当前实例类型具有相同处理器架构的实例类型。例如：

- 如果当前实例类型处理器是基于 Arm 架构的，则仅限于支持基于 Arm 架构的处理器的实例类型，例如 C6g 和 M6g。
- 只有以下实例类型支持 32 位 AMIs：`t2.nano`、`t2.micro`、`t2.small`、`t2.medium`、`c3.large`、`t1.micro`、`m1.small`、`m1.` 和 `c1.medium`。如需更改 32 位实例的实例类型，则仅限使用以下实例类型。

网络适配器

如果从一个网络适配器的驱动程序切换到另一个网络适配器，则在操作系统创建新的适配器时将重置网络适配器设置。要重新配置设置，您可能需要使用管理员权限访问本地账户。以下是从一个网络适配器移到另一个网络适配器的示例：

- Amazon PV（T2 实例）到 Intel 82599 VF（M4 实例）
- Intel 82599 VF（大多数 M4 实例）到 ENA（M5 实例）
- ENA（M5 实例）到高带宽 ENA（M5n 实例）

增强联网

支持增强联网的实例类型需要安装必要的驱动程序。例如，基于 Nitro 的实例需要安装了弹性网络适配器（ENA）驱动程序的 EBS-backed AMI。要将实例的实例类型从不支持增强联网的类型更改为支持增强联网的实例类型，必须根据需要在实例上安装 ENA 驱动程序或 ixgbevf 驱动程序。

> **Note**
>
> 在启用 ENA Express 的情况下调整实例大小时，新实例类型必须同时支持 ENA Express。有关支持 ENA Express 的实例类型列表，请参阅 ENA Express 支持的实例类型。
> 要将实例类型从支持 ENA Express 更改为不支持 ENA Express，请确保在调整实例大小之前尚未启用 ENA Express。

NVMe

在基于 Nitro 的实例上，EBS 卷显示为 NVMe 块储存设备。如果将实例类型不支持 NVMe 的实例的实例类型更改为支持 NVMe 的实例类型，您必须先在实例上安装 NVMe 驱动程序。此外，您在块设备映射中所指定设备的设备名称将使用 NVMe 设备名称（`/dev/nvme[0-26]n1`）进行重命名。

[Linux 实例] 因此，要使用 `/etc/fstab` 在启动时挂载文件系统，必须使用 UUID/标签而非设备名称。

卷限制

您可以挂载到实例的最大 Amazon EBS 卷数取决于实例类型和实例规模。有关更多信息，请参阅 Amazon EC2 实例的 Amazon EBS 卷限制。

您只能更改为支持的卷数与当前附加到实例的卷数相同或更大的实例类型或实例大小。如果您更改为不支持当前附加卷数量的实例类型或实例大小，则请求会失败。例如，如果您从具有 32 个附加卷的 `m7i.4xlarge` 实例更改为最多支持 27 个卷的 `m6i.4xlarge`，请求将失败。

更改 Amazon EC2 实例的实例类型

如果您需要的实例类型与实例的当前配置兼容，请按照以下说明来更改 Amazon EBS 支持的实例的实例类型。有关更多信息，请参阅 the section called “兼容性”。

注意事项

- 您必须先停止实例，然后才能更改其实例类型。当实例停止时，请确保您已计划停机时间。停止实例并更改其实例类型可能需要几分钟时间，重新启动实例所用的时间则由应用程序的启动脚本决定。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。
- 当您停止并启动实例时，我们会将该实例移动到新硬件。如果您的实例具有公有 IPv4 地址，这并非弹性 IP，我们会释放该地址并向实例提供一个新的公有 IPv4 地址。有关实例在整个生命周期中的 IP 地址行为的更多信息，请参阅实例状态之间的区别。
- 您无法更改竞价型实例的实例类型。
- [Windows 实例] 建议您在更改实例类型之前，更新 Amazon PV 驱动程序包。有关更多信息，请参阅 the section called “升级半虚拟化驱动程序”。
- 如果您的实例处于自动扩缩组中，则 Amazon EC2 Auto Scaling 服务会将已停止的实例标记为运行状况不佳，可能会终止该实例并启动替换实例。为防止出现此情况，您可以在更改实例类型时，为组暂停扩展流程。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的暂停和恢复 Auto Scaling 组的进程。
- 当您更改具有 NVMe 实例存储卷实例的实例类型时，更新后的实例可能会有更多实例存储卷，因为即使没有在 AMI 或实例块储存设备映射中指定，所有 NVMe 实例存储卷都可用。否则，已更新的实例通常具有您在启动原始实例时指定的相同实例存储卷数。
- 您可以挂载到实例的最大 Amazon EBS 卷数取决于实例类型和实例规模。您不能更改为不支持已附加到您实例的卷数量的实例类型或实例大小。有关更多信息，请参阅 Amazon EC2 实例的 Amazon EBS 卷限制。
- [Linux 实例] 您可以使用 `AWSSupport-MigrateXenToNitroLinux` 运行手册将兼容 Linux 实例从 Xen 实例类型迁移到 Nitro 实例类型。有关更多信息，请参阅《Amazon Systems Manager Automation Runbook Reference》中的 AWSSupport-MigrateXenToNitroLinux runbook。
- [Windows 实例] 有关将兼容的 Windows 实例从 Xen 实例类型迁移到 Nitro 实例类型的其他指南，请参阅迁移到最新一代的实例类型。

更改由 Amazon EBS 支持的实例的实例类型

1. （可选）如果新实例类型需要现有实例上未安装的驱动程序，您必须先连接到您的实例并安装驱动程序。有关更多信息，请参阅 更改实例类型的兼容性。
2. [Windows 实例] 如果您将 Windows 实例配置为使用静态 IP 寻址，并将不支持增强联网的实例类型更改为支持增强联网的实例类型，则当您在重新配置静态 IP 寻址时，可能会收到有关潜在 IP 地址冲突的警告。要防止此情况出现，在更改实例类型之前，请在网络接口上为实例启用 DHCP。从您的实例中，打开 Network and Sharing Center（网络和共享中心），打开网络接口的 Internet

Protocol Version 4 (TCP/IPv4) Properties（互联网协议版本 4 (TCP/IPv4) 属性），并选择 Obtain an IP address automatically（自动获取 IP 地址）。更改实例类型并在网络接口上重新配置静态 IP 寻址。

3. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
4. 在导航窗格中，选择实例。
5. 选择所需实例，然后依次选择实例状态、停止实例。当系统提示您确认时，选择 Stop。停止实例可能需要几分钟时间。
6. 在实例处于选中状态时，依次选择操作、实例设置、更改实例类型。若实例状态不是 `stopped`，则此操作会显示为灰色。
7. 请在 Change instance type（更改实例类型）页面上执行以下操作：

   a. 对于 Instance type（实例类型），选择您所需的实例类型。

      如果列表中未包含该实例类型，则说明其与您的实例配置不兼容。请改为使用以下说明：通过启动新的 EC2 实例迁移到新实例类型。

   b. （可选）如果您选择的实例类型支持 EBS 优化，则选择 EBS-optimized（EBS 优化）以启用 EBS 优化，或取消选择 EBS-optimized（EBS 优化）以禁用 EBS 优化。

      如果您选择的实例类型默认情况下已经过 EBS 优化，则 EBS-optimized（EBS 优化）已选中，您无法取消选择。

   c. （可选）配置新实例类型的 vCPU 选项。

      当您对现有实例更改实例类型时，Amazon EC2 会尽可能将现有实例的 CPU 选项设置应用到新实例。如果新的实例类型不支持这些设置，则 CPU 选项将重置为无。对于新实例类型，此选项会使用默认 vCPU 数。

      如果您选择的实例类型支持 vCPU 配置，请在高级详细信息面板中选择指定 CPU 选项，从而为您的新实例类型配置 vCPU。

   d. 选择更改以接受新设置。

8. 要启动实例，请选择该实例，然后依次选择 Instance state（实例状态）、Start instance（启动实例）。实例进入 `running` 状态可能需要几分钟时间。如果您的实例无法启动，请参阅 实例类型更改的问题排查。
9. [Windows 实例] 如果您的实例运行 Windows Server 2016 或 Windows Server 2019（带 EC2Launch v1），则会连接到您的 Windows 实例并运行以下 EC2Launch PowerShell 脚本，以便在更改实例类型后对其进行配置。

> **Important**
>
> 当您启用初始化实例 EC2 Launch 脚本时，管理员密码将重置。您可以修改配置文件，通过在初始化任务的设置中指定禁用管理员密码重置来禁用它。有关如何禁用密码重置的步骤，请参阅配置初始化任务（EC2Launch）或更改设置（EC2Launch v2）。

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

通过启动新的 EC2 实例迁移到新实例类型

仅当 EC2 实例是 EBS 支持的实例，并且其配置与您想要的新实例类型兼容时，您才能更改该实例的实例类型。否则，如果配置或实例与新实例类型不兼容，或者它是基于实例存储的实例，则您必须启动与所需实例类型兼容的替代实例。有关如何确定兼容性的更多信息，请参阅更改实例类型的兼容性。

迁移过程概述

- 备份原始实例上的数据。
- 启动一个具有与您所需的新实例类型兼容的配置的新实例，附加已附加到原始实例的任何 EBS 卷。
- 在新实例上安装应用程序。
- 恢复所有数据。
- 如果原始实例有弹性 IP 地址，则必须将其与新实例关联，以确保您的用户可以继续使用您的应用程序而不会中断。

将实例迁移到新实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 备份仍然需要的任何数据，如下所示：
   - 连接到您的实例，并将实例存储卷上的数据复制到持久性存储中。
   - 创建 EBS 卷的快照，以便您可以使用相同数据创建新卷；或者将卷从原始实例分离，以便您可以将其附加到新实例。
3. 在导航窗格中，选择 Instances (实例)。
4. 选择 Launch instances。配置实例时，执行以下操作：

a. 选择支持您所需实例类型的 AMI。例如，您必须选择支持新实例类型的处理器类型的 AMI。此外，当前一代的实例类型需要 HVM AMI。

b. 选择所需的新实例类型。如果您所需的实例类型不可用，则说明其与您所选 AMI 的配置不兼容。

c. 如果您要允许相同的流量到达新实例，请选择与原始实例使用相同的 VPC 和安全组。

d. 完成新实例的配置后，请完成以下步骤以选择密钥对并启动实例。实例进入 `running` 状态可能需要几分钟时间。

5. 如果您将数据备份到 EBS 快照，请从快照创建卷，然后将该卷附加到新实例。

   要将 EBS 卷从原始实例移动到新实例，请先从原始实例分离该卷，然后将该卷附加到新实例。

6. 在新实例上安装应用程序和所有必需软件。

7. 还原您在原始实例的实例存储卷中备份的所有数据。

8. 如果原始实例有弹性 IP 地址，请将其按如下方式分配给新实例：

   a. 在导航窗格中，选择 Elastic IPs。

   b. 选择与原始实例关联的弹性 IP 地址，然后依次选择操作、取消关联弹性 IP 地址。当系统提示进行确认时，选择取消关联。

   c. 在弹性 IP 地址仍处于选中状态的情况下，依次选择操作、关联弹性 IP 地址。

   d. 对于 Resource type (资源类型)，选择 Instance (实例)。

   e. 对于实例，选择新实例。

   f. （可选）对于 Private IP address (私有 IP 地址)，请指定要将弹性 IP 地址关联到的私有 IP 地址。

   g. 选择 Associate。

9. (可选) 如果不再需要原始实例，您可以将其终止。选择实例，确认您将要终止原始实例而不是新实例（例如，查看名称或启动时间），然后依次选择 Instance state（实例状态）、Terminate instance（终止实例）。

实例类型更改的问题排查

使用以下信息可帮助您诊断和修复在更改实例类型时可能遇到的常见问题。

更改实例类型后，实例无法启动

可能的原因：未满足新实例类型的要求

如果您的实例未启动，则可能是新实例类型的某一要求未满足。有关更多信息，请参阅为什么我的 Linux 实例在更改其类型后无法启动？

可能的原因：AMI 不支持实例类型

若使用 EC2 控制台更改实例类型，则仅限所选 AMI 支持的实例类型可用。若使用 Amazon CLI 启动实例，则可以指定不兼容的 AMI 和实例类型。如果 AMI 和实例类型不兼容，则实例无法启动。有关更多信息，请参阅 更改实例类型的兼容性。

可能的原因：实例位于集群置放群组中

如果您的实例位于集群置放群组中，并且更改实例类型后实例无法启动，请尝试以下操作：

1. 停止集群置放群组中的所有实例。
2. 更改受影响的实例的实例类型。
3. 启动集群置放群组中的所有实例。

更改实例类型后，无法从互联网访问应用程序或网站

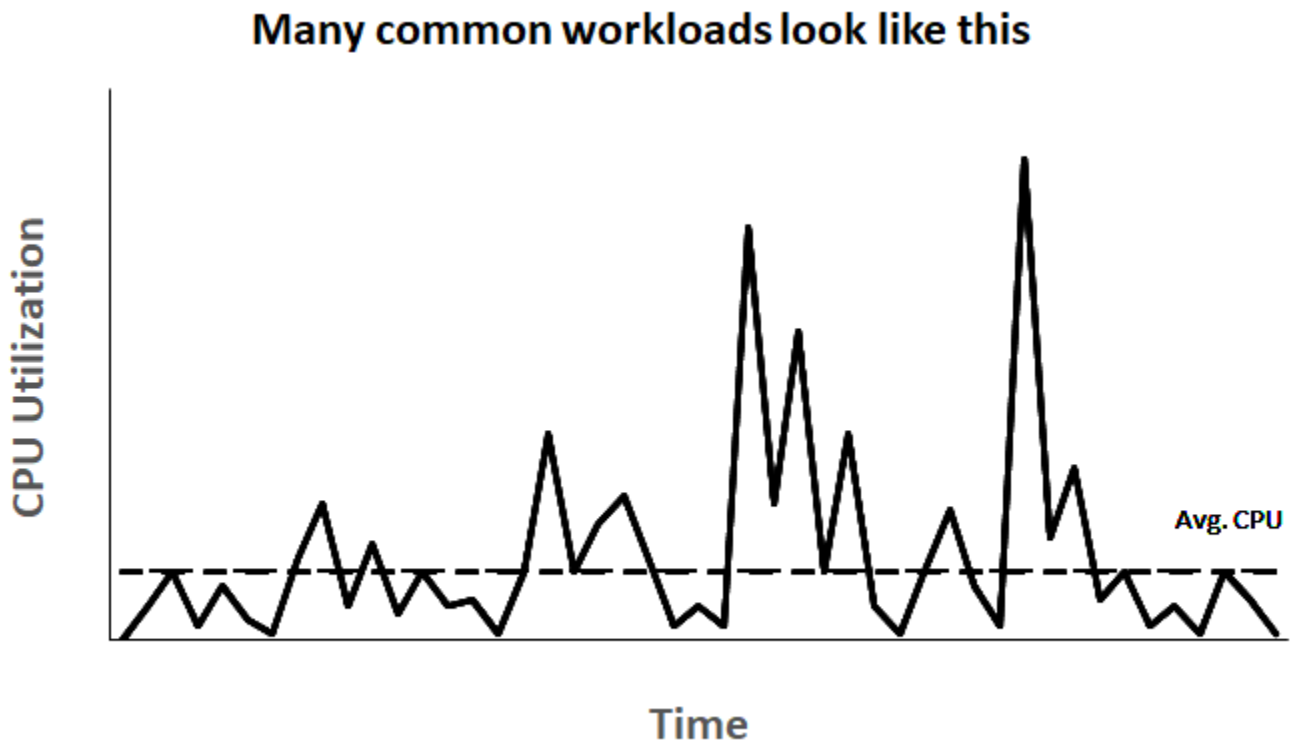
可能的原因：公有 IPv4 地址被释放

更改实例类型时，您必须先停止该实例。停止实例时，我们会释放公有 IPv4 地址并向该实例提供一个新的公有 IPv4 地址。

为了在实例停止和启动之间保留公有 IPv4 地址，我们建议您使用弹性 IP 地址，若您的实例处于运行中，则无需额外费用。有关更多信息，请参阅 弹性 IP 地址。

可突增性能实例

许多通用型工作负载通常不繁忙，并且不需要高水平的持续 CPU 性能。下图展示了当前客户在 Amazon 云中运行的许多常见工作负载的 CPU 利用率。



这些低至中等 CPU 利用率工作负载会导致 CPU 周期浪费，因此，您需要支付的费用超过使用量。为了解决这个问题，您可以利用低成本可突增通用型实例，即 T 实例。

T 实例系列提供基准 CPU 性能，而且只要有需要，就能随时突增到基准线以上。基准 CPU 旨在满足大多数通用型工作负载的需求，包括大型微服务、Web 服务器、中小型数据库、数据记录、代码存储库、虚拟桌面、开发和测试环境以及业务关键型应用程序。T 实例平衡了计算、内存和网络资源，并为您提供最经济高效的方式来运行各种具有低至中等 CPU 利用率的通用型应用程序。与 M 实例相比，它们可以为您节省高达 15% 的成本，并且可以通过更小、更经济的实例大小节约更多成本，提供低至 2 个 vCPU 和 0.5GiB 的内存。较小的 T 实例大小（例如纳型、微型、小型和中型）非常适合需要少量内存且无需高 CPU 利用率的工作负载。

> **Note**
>
> 本主题介绍可突增 CPU。有关可突增网络性能的信息，请参阅 Amazon EC2 实例网络带宽。

EC2 可突增实例类型

EC2 可突增实例包括 T4g、T3a 和 T3 实例类型，以及上一代 T2 实例类型。

T4g 实例类型是最新一代可突增实例。它们提供最好的性价比，并为您提供所有 EC2 实例类型中最低的成本。T4g 实例类型由基于 Arm 的 Amazon Graviton2 处理器提供动力，并且具有操作系统供应商、独立软件供应商以及流行的 Amazon 服务和应用程序广泛的生态系统支持。

下表总结了可突增实例类型之间的主要区别。

类型	描述	处理器系列
最新一代		
T4g	成本最低的 EC2 实例类型，与 T3 相比，性价比高 40%，成本低 20%	采用 Arm Neoverse N1 核心的 Amazon Graviton2 处理器
T3a	成本最低的基于 x86 的实例，与 T3 实例相比，成本低 10%	AMD 第一代 EPYC 处理器
T3	适用于 x86 工作负载的最佳峰值性价比，同等性能的价格比上一代 T2 实例低 30%	英特尔至强可扩展（Skylake、Cascade Lake 处理器）
上一代		
T2	上一代可突增实例	英特尔至强处理器

有关实例定价以及其他规格的信息，请参阅 Amazon EC2 定价和 Amazon EC2 实例类型。有关可突增网络性能的信息，请参阅 Amazon EC2 实例网络带宽。

T 实例支持的购买选项

- On-Demand Instances
- Reserved Instances
- 专用实例（仅限 T3）
- 专属主机（仅限 T3，仅处于 `standard` 模式）
- 竞价型实例

有关更多信息，请参阅 Amazon EC2 账单和购买选项。

目录

- 最佳实践
- 可突增性能实例的关键概念
- 可突增性能实例的无限模式
- 可突增性能实例的标准模式
- 配置可突增性能实例
- 监控可突增实例的 CPU 积分

最佳实践

按照这些最佳实践可以从可突增性能实例获得最大的好处。

- 确保您选择的实例大小达到您的操作系统和应用程序的最低内存要求。在许多使用案例中，带有消耗大量内存和 CPU 资源的图形用户界面的操作系统（例如，Windows）可能需要 `t3.micro` 或更大的实例。随着您的工作负载对内存和 CPU 的需求随时间增加，您可以通过 T 实例灵活地扩展到相同实例类型更大实例大小，或者选择其他实例类型。
- 为您的账户启用 Amazon Compute Optimizer 并审查针对您的工作负载的 Compute Optimizer 推荐。Compute Optimizer 可以帮助评测是应该扩大实例规模以提高性能，还是应该缩小实例规模以节省成本。Compute Optimizer 还可以根据您的场景推荐不同的实例类型。有关更多信息，请参阅《Amazon Compute Optimizer 用户指南》中的查看 EC2 实例建议。

可突增性能实例的关键概念

传统 Amazon EC2 实例类型提供固定的 CPU 资源，而可突增性能实例提供基准水平的 CPU 利用率并能够将 CPU 利用率突增到基准水平之上。这样确保您只需为基准 CPU 以及任何额外的突增 CPU 使用量付费，从而降低计算成本。基准利用率和突增能力由 CPU 积分控制。可突增性能实例是唯一将积分用于 CPU 使用的实例类型。

当每个可突增性能实例保持在 CPU 基准以下时，它会持续获得积分，并且在突增超过基准时持续花费积分。获得或花费的积分数量取决于实例的 CPU 利用率：

- 如果 CPU 利用率低于基准，则所获得的积分将会大于所花费的积分。
- 如果 CPU 利用率等于基准，则所获得的积分将会等于所花费的积分。
- 如果 CPU 利用率高于基准，则所花费的积分将会高于所获得的积分。

当获得的积分大于花费的积分时，差额称为“累积积分”，这些积分以后可以用于突增超过基准的 CPU 利用率。同样，当花费的积分超过获得的积分时，实例行为取决于信用配置模式 – 标准模式或无限模式。

在标准模式下，当花费的积分超过获得的积分时，实例将使用累积积分来突增到基准 CPU 利用率之上。如果没有剩余的累积积分，则实例将逐渐下降到基准 CPU 利用率，并且在累积更多积分之前不能突增到基准之上。

在无限模式下，如果实例突发超过基准 CPU 利用率，则实例首先使用累积积分进行突增。如果没有剩余的累积积分，则实例将花费剩余积分进行突增。在该实例的 CPU 使用率低于基准时，实例会使用它获得的 CPU 积分支付以前花费的超额积分。凭借获得 CPU 积分来支付超额积分的能力，Amazon EC2 可以在 24 小时周期内将实例的 CPU 使用率保持在平均水平。如果 24 小时的平均 CPU 使用率超过基准，则会按每 vCPU 小时的固定额外费率对实例收取额外的使用费用。

内容

- 关键概念和定义
- 赢取 CPU 积分
- CPU 积分获得率
- CPU 积分累积限制
- 累积的 CPU 积分生命期
- 基准利用率

关键概念和定义

以下关键概念和定义适用于可突增性能实例。

CPU 使用率

CPU 利用率是当前正在实例上使用的已分配 EC2 计算单位的百分率。此指标衡量实例上正在利用的已分配 CPU 周期的百分比。CPU 利用率 CloudWatch 指标显示每个实例的 CPU 利用率，而不是每个核心的 CPU 利用率。实例的基准 CPU 规范还基于每个实例的 CPU 利用率。要测量 CPU 利用率，请使用 Amazon Web Services 管理控制台 或 Amazon CLI，请参阅 获取特定实例的统计数据。

CPU 积分

vCPU 时间的单位。

示例：

1 个 CPU 积分 = 1 个 vCPU * 100% 利用率 * 1 分钟。

1 个 CPU 积分 = 1 个 vCPU * 50% 利用率 * 2 分钟

1 个 CPU 积分 = 2 个 vCPU * 25% 利用率 * 2 分钟

基准利用率

基准利用率 是当获得的 CPU 积分数与所使用的 CPU 积分数相等时，CPU 在净积分余额为零时的利用率水平。基准利用率也称为基准。基准利用率表示为 vCPU 利用率的百分比，计算方式如下：基准利用率 % =（获得的积分数/vCPU 数）/60 分钟。

有关每种可突增性能实例类型的基准利用率，请参阅积分表。

获得的积分

实例在运行时连续获得的积分。

每小时获得的积分数 = % 基准利用率 * vCPU 数 * 60 分钟

例如：

具有 2 个 vCPU 且基准利用率为 5% 的 T3.nano 每小时可获得 6 积分，计算方式如下：

2 个 vCPU * 5% 基准 * 60 分钟 = 每小时 6 积分

已花费或使用的积分

实例在运行时连续使用的积分。

每分钟花费的 CPU 积分 = vCPU 数 * CPU 利用率 * 1 分钟

累积积分

当实例使用的积分少于基准利用率所需的积分时，未使用的 CPU 积分。换句话说，累积积分 = 低于基准的（获得的积分 - 已使用的积分）。

例如：

如果 t3.nano 以 2% 的 CPU 利用率运行，在一小时内低于 5% 的基准，则累积积分的计算方式如下：

累积 CPU 积分 =（每小时获得的积分 - 每小时使用的积分）= 6 - 2 个 vCPU * 2% CPU 利用率 * 60 分钟 = 6 - 2.4 = 每小时 3.6 积分

积分累积限制

取决于实例大小，但通常等于 24 小时内获得的最大积分数。

例如：

对于 t3.nano，积分累积限额 = 24 * 6 = 144 积分

启动积分

仅适用于为标准模式配置的 T2 实例。启动积分是分配给新 T2 实例的有限数量的 CPU 积分，这样在标准模式下启动时，它可以突增到基准以上。

剩余积分

实例在耗尽其累积积分余额后花费的积分。剩余积分专为可突增实例而设计，以便在较长时间内保持高性能，并且仅在无限模式下使用。剩余积分余额用于确定实例在无限模式下使用了多少积分。

标准模式

积分配置模式，允许实例通过支出积分余额中累积的积分突增到基准以上。

无限模式

积分配置模式，允许实例通过承受所需的任何时段的高 CPU 利用率来突增到基准以上。如果在滚动 24 小时或实例生命周期（以较短者为准）内实例的平均 CPU 使用率等于或低于基准，实例的每小时价格自动涵盖所有 CPU 使用峰值。如果实例长时间以较高的 CPU 使用率运行，可能会按每 vCPU 小时的固定费率收取额外的费用。

下表总结了可突增实例类型之间的主要积分区别。

类型	支持的 CPU 积分类型	积分配置模式	实例启动和停止之间的累积的 CPU 积分生命周期
最新一代			
T4g	获得的积分、累积积分、已使用的积分、剩余积分（仅无限模式）	标准、无限（默认值）	7 天（积分在实例停止后持续 7 天）

类型	支持的 CPU 积分类型	积分配置模式	实例启动和停止之间的累积的 CPU 积分生命周期
T3a	获得的积分、累积积分、已使用的积分、剩余积分（仅无限模式）	标准、无限（默认值）	7 天（积分在实例停止后持续 7 天）
T3	获得的积分、累积积分、已使用的积分、剩余积分（仅无限模式）	标准、无限（默认值）	7 天（积分在实例停止后持续 7 天）
上一代			
T2	获得的积分、累积积分、已使用的积分、启动积分（标准模式）、剩余积分（仅无限模式）	标准（默认值）、无限	0 天（实例停止时积分会丢失）

> **Note**
>
> 在专属主机上启动的 T3 实例不支持无限制模式。

赢取 CPU 积分

每个可突增性能实例以设定的每小时速率（以毫秒级精度）持续获得 CPU 积分，具体取决于实例大小。用于加减积分的核算过程也以毫秒级精度进行，因此您不必担心 CPU 积分超支；CPU 的短时间突增只消耗少量 CPU 积分。

如果可突增性能实例使用的 CPU 资源少于基准利用率所需的数量（例如，处于空闲状态时），则未使用的 CPU 积分将累积到 CPU 积分余额中。如果可突增性能实例需要突增至基准利用率水平以上，它将花费累积积分。可突增性能实例累积的积分越多，在需要更高 CPU 利用率时，它突增到基准以上的时间就越长。

下表列出了可突增性能实例类型、每小时获得 CPU 积分的速率、实例可以累积获得的最大 CPU 积分数、每个实例的 vCPU 数以及以完整核心百分比形式表示的基准利用率（使用单个 vCPU）。

实例类型	每小时获得的 CPU 积分	可累积获得的最大积分数*	vCPU***	每个 vCPU 的基准利用率
T2				
t2.nano	3	72	1	5%
t2.micro	6	144	1	10%
t2.small	12	288	1	20%
t2.medium	24	576	2	20%**
t2.large	36	864	2	30%**
t2.xlarge	54	1296	4	22.5%**
t2.2xlarge	81.6	1958.4	8	17%**
T3				
t3.nano	6	144	2	5%**
t3.micro	12	288	2	10%**
t3.small	24	576	2	20%**
t3.medium	24	576	2	20%**
t3.large	36	864	2	30%**
t3.xlarge	96	2304	4	40%**
t3.2xlarge	192	4608	8	40%**
T3a				
t3a.nano	6	144	2	5%**

实例类型	每小时获得的 CPU 积分	可累积获得的最大积分数*	vCPU***	每个 vCPU 的基准利用率
t3a.micro	12	288	2	10%**
t3a.small	24	576	2	20%**
t3a.medium	24	576	2	20%**
t3a.large	36	864	2	30%**
t3a.xlarge	96	2304	4	40%**
t3a.2xlarge	192	4608	8	40%**
T4g				
t4g.nano	6	144	2	5%**
t4g.micro	12	288	2	10%**
t4g.small	24	576	2	20%**
t4g.medium	24	576	2	20%**
t4g.large	36	864	2	30%**
t4g.xlarge	96	2304	4	40%**
t4g.2xlarge	192	4608	8	40%**

* 可累积的积分数等于可在 24 小时周期内获得的积分数。

** 表中的基准利用率百分比按每个 vCPU 计算。在 CloudWatch 中，CPU 使用率按各个 vCPU 显示。例如，以基准水平运行的 t3.large 实例的 CPU 利用率在 CloudWatch CPU 指标中显示为 30%。有关如何计算基准利用率的信息，请参阅基准利用率。

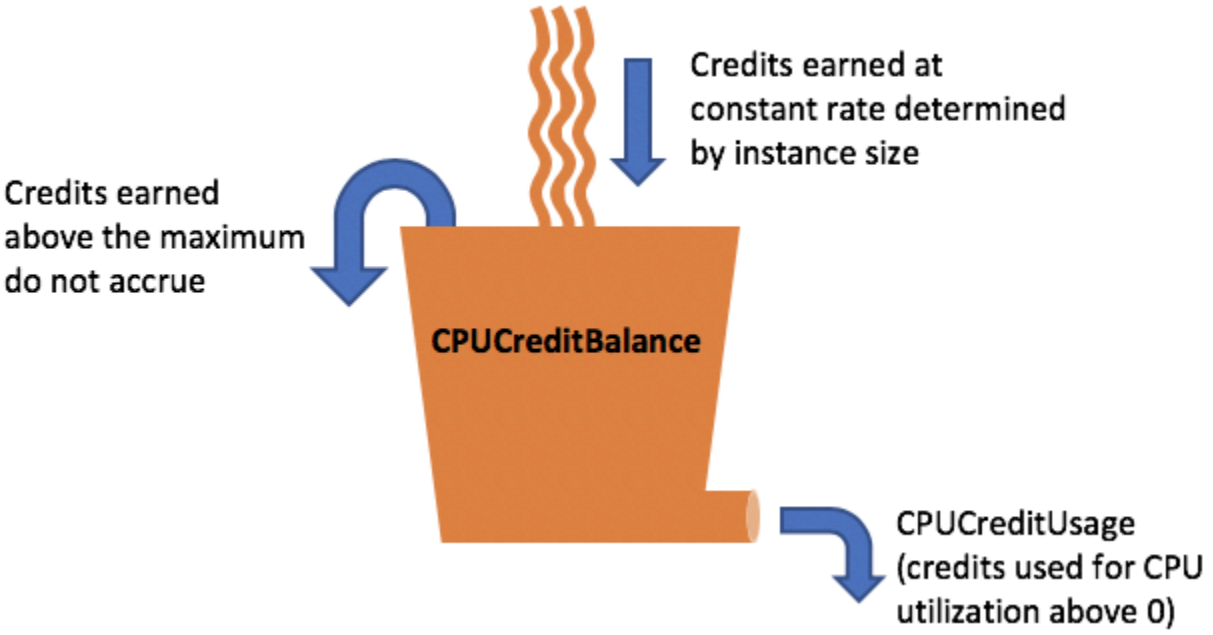
*** 每个 vCPU 都是英特尔至强核心或 AMD EPYC 核心的线程，但 T2 和 T4g 实例除外。

CPU 积分获得率

每小时获得的 CPU 积分数是由实例大小决定的。例如，`t3.nano` 每小时获得 6 积分，而 `t3.small` 每小时获得 24 积分。上表列出了所有实例的积分获得率。

CPU 积分累积限制

虽然获得的积分在运行的实例上从不过期，但实例可累积获得的积分数存在限制。该限制由 CPU 积分余额限制决定。在到达限制后，获得的任何新积分都会被丢弃，如下图所示。存储桶已满表示达到 CPU 积分余额限制，而溢出指示超出限制的新获得积分。



对于每种 实例大小，CPU 积分余额限制是不同的。例如，`t3.micro` 实例可在 CPU 积分余额中累积最多 288 个获得的 CPU 积分。上表列出了每个 实例可以累积获得的最大积分数。

T2 标准实例也获得启动积分。启动积分不计入 CPU 积分余额限制。如果 T2 实例尚未使用其启动积分，并保持闲置状态 24 小时，同时累积获得的积分，则其 CPU 积分余额将超过限制。有关更多信息，请参阅 启动积分。

T4g、T3a 和 T3 实例不会获得启动积分。默认情况下，这些实例以 `unlimited` 模式启动，因此可以在启动时立即突增，无需任何启动积分。预设情况下 T3 实例在专属主机启动时启动为 `standard`；`unlimited` 模式不支持专属主机上的 T3 实例。

累积的 CPU 积分生命期

运行的实例上的 CPU 积分不会过期。

对于 T2，CPU 积分余额在实例停止与启动之间不保留。如果您停止 T2 实例，实例将失去其所有累积积分。

对于 T4g、T3a 和 T3，CPU 积分余额将在实例停止后保留七天，而后不复存在。如果在七天内启动实例，则不会丢失积分。

有关更多信息，请参阅 CloudWatch 指标表中的 CPUCreditBalance。

基准利用率

基准利用率 是当获得的 CPU 积分数与所使用的 CPU 积分数相等时，CPU 在净积分余额为零时的利用率水平。基准利用率也称为基准。

基准利用率表示为 vCPU 利用率的百分比，计算方法如下：

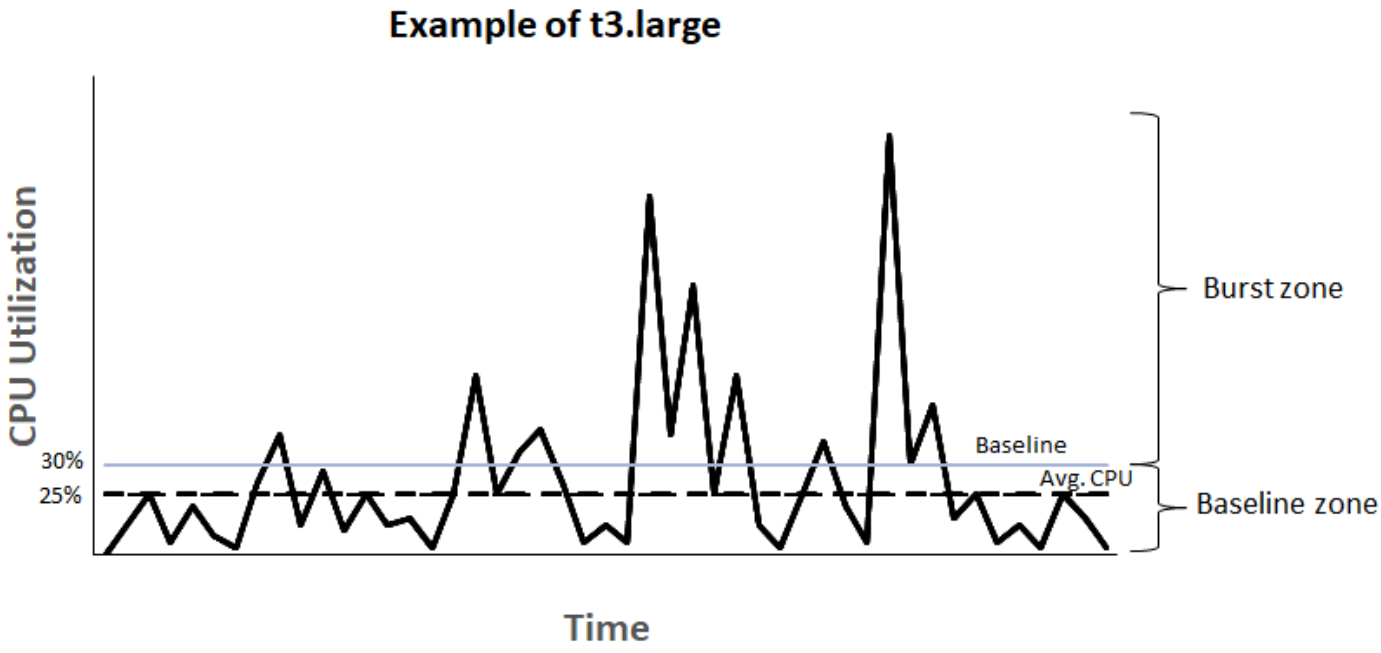
```
(number of credits earned/number of vCPUs)/60 minutes = % baseline
utilization
```

例如，一个 t3.nano 实例具有 2 个 vCPU，每小时获得 6 积分，得到的基准利用率为 5%，计算方法如下：

```
(6 credits earned/2 vCPUs)/60 minutes = 5% baseline utilization
```

一个 t3.large 实例具有 2 个 vCPU，每小时获得 36 积分，得到的基准利用率为 30%（(36/2)/60）。

下图提供了平均 CPU 利用率低于基准的 t3.large 示例。



可突增性能实例的无限模式

配置为 `unlimited` 的可突增性能实例可以承受所需的任何时段的高 CPU 利用率。如果在滚动 24 小时或实例生命周期（以较短者为准）内实例的平均 CPU 使用率等于或低于基准，实例的每小时价格自动涵盖所有 CPU 使用峰值。

对于绝大多数通用型工作负载，配置为 `unlimited` 的实例可提供足够高的性能，而不会收取任何额外的费用。如果实例长时间以较高的 CPU 利用率运行，可能会按每 vCPU 小时的固定费率收取额外的费用。有关定价的信息，请参阅 Amazon EC2 定价和 T2/T3 无限模式定价。

T4g、T3a 和 T3 实例默认情况下以 `unlimited` 模式启动（除非您更改默认值）。如果滚动 24 小时内的平均 CPU 使用率超过基准，将会产生超额积分费用。如果您以 `unlimited` 模式启动竞价型实例并计划立即短时间使用它们（没有空闲时间来累积 CPU 积分），则会产生超额积分费用。建议您以标准模式启动竞价型实例，以避免支付更高的费用。有关更多信息，请参阅 超额积分会产生费用 和 启动可突增性能实例。

> **Note**
>
> 预设情况下 T3 实例在专属主机启动时启动为 `standard`；`unlimited` 模式不支持专属主机上的 T3 实例。

内容

- 可突增实例的无限模式概念
  - 无限可突增性能实例的工作原理
  - 何时使用无限模式与固定 CPU
  - 超额积分会产生费用
  - 无限突增性能的成本是多少？
  - T2 无限实例没有启动积分
  - 启用无限模式
  - 在无限模式和标准模式之间切换时，积分会出现什么情况
  - 监控积分使用情况
- 可突增实例的无限模式示例
  - 示例 1：介绍 T3 无限版的积分使用情况
  - 示例 2：介绍 T2 无限版的积分使用情况

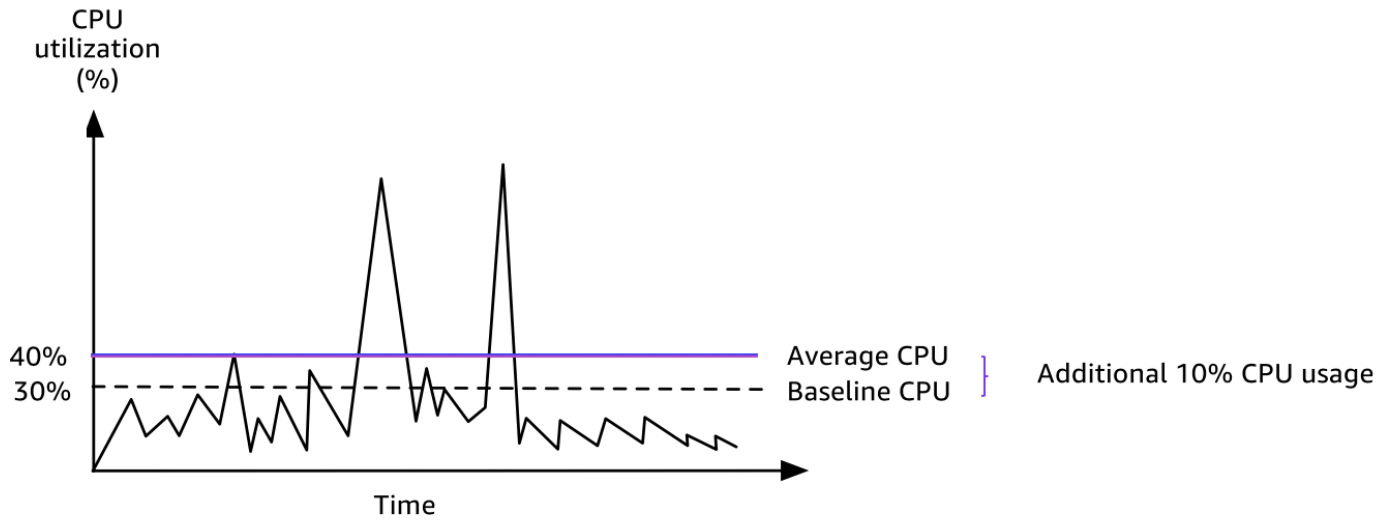
可突增实例的无限模式概念

unlimited 是用于可突增性能实例的积分配置选项。可以随时对正在运行或已停止的实例启用或禁用它。您可以在每个 Amazon 区域的账户级别将 unlimited 设置为每个可突增性能实例系列的默认积分选项，以便账户中所有新的可突增性能实例都使用默认积分选项启动。

无限可突增性能实例的工作原理

如果配置为 unlimited 的可突增性能实例用完其 CPU 积分余额，它可能会花费超额积分以突增到基准以上。在该实例的 CPU 使用率低于基准时，实例会使用它获得的 CPU 积分支付以前花费的超额积分。凭借获得 CPU 积分来支付超额积分的能力，Amazon EC2 可以在 24 小时周期内将实例的 CPU 使用率保持在平均水平。如果 24 小时的平均 CPU 使用率超过基准，则会按每 vCPU 小时的固定额外费率对实例收取额外的使用费用。

下图显示 t3.large 的 CPU 使用率。t3.large 的基准 CPU 使用率为 30%。如果实例在 24 小时内以平均 30% CPU 使用率或更低运行，则没有额外费用，因为费用已由实例每小时价格所涵盖。但是，如果实例在 24 小时内以平均 40% 的 CPU 使用率运行（如图中所示），则会按每 vCPU 小时的固定额外费率对实例收取额外的 10% CPU 使用率费用。



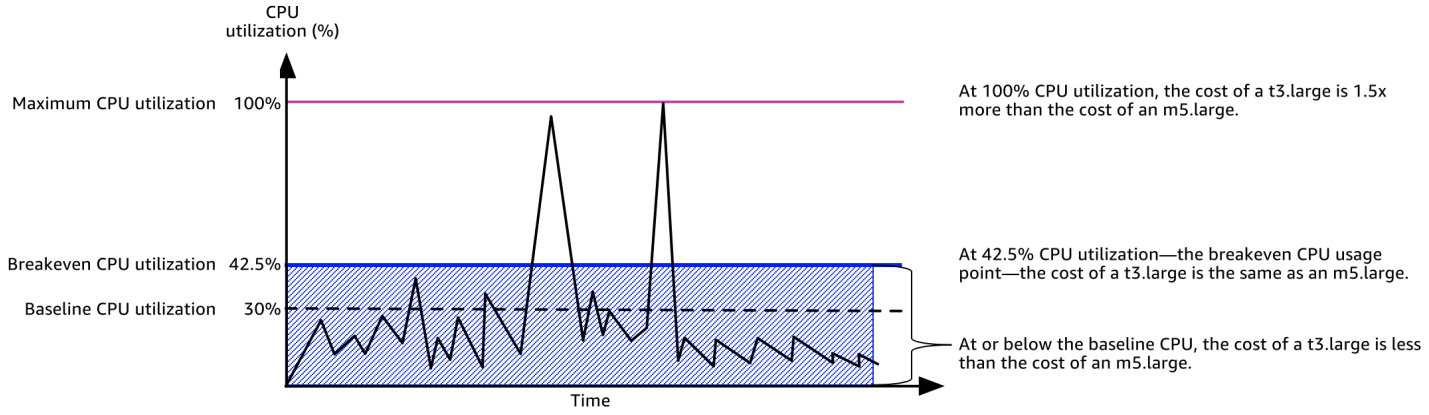
有关每个实例类型的每个 vCPU 的基准利用率以及每个实例类型可获得的积分数的更多信息，请参阅积分表。

何时使用无限模式与固定 CPU

当确定您在 unlimited 模式下是应使用可突增性能实例（如 T3）还是固定性能实例（如 M5）时，您需要确定收支平衡 CPU 使用率。可突增性能实例的收支平衡 CPU 使用率是可突增性能实例与固定性能实例的费用相同的点。收支平衡 CPU 使用率可帮助您确定以下内容：

- 如果 24 小时内的平均 CPU 使用率等于或低于收支平衡 CPU 使用率，请在 unlimited 模式下使用可突增性能实例，以便您可以受益于可突增性能实例的较低价格，同时获得与固定性能实例相同的性能。
- 如果 24 小时内的平均 CPU 使用率高于收支平衡 CPU 使用率，可突增性能实例将花费比同等大小的固定性能实例更多的费用。如果 T3 实例以 100% CPU 持续突增，则您最终要支付的价格约为同等大小 M5 实例的价格的 1.5 倍。

下图显示了其中 t3.large 花费与 m5.large 花费相同的收支平衡 CPU 使用率点。t3.large 的收支平衡 CPU 使用率点为 42.5%。如果平均 CPU 使用率为 42.5%，则运行 t3.large 的费用与 m5.large 的运行费用相同，如果平均 CPU 使用率高于 42.5%，则前者费用更高。如果工作负载需要低于 42.5% 的平均 CPU 使用率，您可以受益于 t3.large 的较低价格，同时获得与 m5.large 相同的性能。



下表显示了如何计算收支平衡的 CPU 使用率阈值，以便您可以确定何时使用 unlimited 模式的可突增性能实例或固定性能实例将更为便宜。表中的列标记为 A 到 K。

实例类型	vCPU	T3 价格*/小时	M5 价格*/小时	价格差异	T3 的每个 vCPU 的基准利用率 (%)	向超额积分每 vCPU 小时收取费用	每 vCPU 分钟收费	每 vCPU 可用的额外突增分钟数	可用的额外 CPU %	收支平衡 CPU %
A	B	C	D	E = D - C	F	G	H = G / 60	I = E / H	J = (I / 60) / B	K = F + J
t3.large	2	0.0835 美元	0.096 美元	0.0125 USD	30%	$0.05	0.000833 美元	15	12.5%	42.5%

* 价格基于 us-east-1 和 Linux OS。

该表提供以下信息：

- 列 A 显示实例类型 t3.large。
- 列 B 显示 t3.large 的 vCPU 数。

- 列 C 显示每小时 t3.large 的价格。
- 列 D 显示每小时 m5.large 的价格。
- 列 E 显示 t3.large 和 m5.large 之间的价格差异。
- F 列显示的是 t3.large 的每个 vCPU 的基准利用率，即 30%。在基准时，实例的每小时成本涵盖 CPU 使用率的成本。
- 列 G 显示向实例收取的每 vCPU 小时的固定额外费率（如果实例在耗尽其获得的积分后以 100% CPU 突增）。
- 列 H 显示向实例收取的每 vCPU 分钟的固定额外费率（如果实例在耗尽其获得的积分后以 100% CPU 突增）。
- 列 I 显示 t3.large 可每小时以 100% CPU 突增的同时支付与 m5.large 相同的每小时价格的额外分钟数。
- 列 J 显示该实例可突增的同时支付与 m5.large 相同的每小时价格的超过基准的额外 CPU 使用率（单位为 %）。
- 列 K 显示 t3.large 在支付不超过 m5.large 的费用的情况下可突增的收支平衡 CPU 使用率（单位为 %）。除此之外，t3.large 的费用超过 m5.large。

下表显示了与类似大小的 M5 实例类型相比 T3 实例类型的收支平衡 CPU 使用率（单位为 %）。

T3 实例类型	T3 的收支平衡 CPU 使用率（单位为 %）（与 M5 相比）
t3.large	42.5%
t3.xlarge	52.5%
t3.2xlarge	52.5%

超额积分会产生费用

如果实例的平均 CPU 使用率等于或低于基准，则实例不会产生额外的费用。由于实例在 24 小时周期内可获得最大数量的积分（例如，t3.micro 实例可在 24 小时周期内获得最多 288 积分）的原因，因此在花费的超额积分不超过最大积分数时，不会立即向您收费。

但是，如果 CPU 利用率保持在基准以上，则实例无法获得足够的积分来支付已花费的超额积分。对于未支付的超额积分，按每 vCPU 小时的固定费率收取额外的费用。有关费率的信息，请参阅 T2/T3 无限模式定价。

在出现以下任一情况时，将对之前花费的超额积分收费：

- 花费的超额积分超出实例可在 24 小时周期内获得的最大积分数。对于超出最大积分数的所花费超额积分，将在该小时结束时向您收费。
- 实例已停止或终止。
- 实例从 `unlimited` 切换为 `standard`。

花费的超额积分是通过 CloudWatch 指标 `CPUSurplusCreditBalance` 跟踪的。通过 CloudWatch 指标 `CPUSurplusCreditsCharged` 来跟踪收费的超额积分。有关更多信息，请参阅 可突增性能实例的其他 CloudWatch 指标。

无限突增性能的成本是多少？

如果您使用了剩余的服务抵扣金，但这些服务抵扣金未被已获得的服务抵扣金予以清偿（请参阅 超额积分会产生费用），则需按照每 vCPU 小时的固定额外费率来支付剩余的服务抵扣金费用。费率列在 Amazon EC2 按需型定价页面上的 T2/T3 无限模式定价部分中。

T2 无限实例没有启动积分

T2 标准实例可收到启动积分，但 T2 无限实例不会收到启动积分。T2 无限实例可以随时突增到基准以上，而不会收取额外的费用，但前提是在滚动 24 小时时间段或其生命周期（以较短者为准）内实例的平均 CPU 使用率等于或低于基准。因此，T2 无限实例不需要启动积分，即可在启动后立即达到较高的性能。

如果 T2 实例从 `standard` 切换到 `unlimited`，则将从 `CPUCreditBalance` 中扣除所有累积的启动积分，然后再结转剩余的 `CPUCreditBalance`。

T4g、T3a 和 T3 实例永远不会收到启动积分，因为这些实例默认以无限模式启动，所以才可以在启动时立即突增。无限模式积分配置让 T4g、T3a 和 T3 实例能够根据需要使用尽可能多的 CPU，以便在需要的时间内突增超出基准。

启用无限模式

您可以随时在正在运行或停止的实例上从 `unlimited` 切换到 `standard` 以及从 `standard` 切换到 `unlimited`。有关更多信息，请参阅在启动时配置积分规范 和 管理可突增性能实例的积分规范。

您可以在每个Amazon区域的账户级别将 `unlimited` 设置为每个可突增性能实例系列的默认积分选项，以便账户中所有新的可突增性能实例都使用默认积分选项启动。有关更多信息，请参阅管理账户的默认积分规范。

您可以使用 Amazon EC2 控制台或 Amazon CLI，检查可突增性能实例已配置为 unlimited 还是 standard。有关更多信息，请参阅 配置可突增性能实例。

在无限模式和标准模式之间切换时，积分会出现什么情况

CPUCreditBalance 是跟踪实例产生的积分数的 CloudWatch 指标。CPUSurplusCreditBalance 是跟踪实例所用超额积分数的 CloudWatch 指标。

当您将配置为 unlimited 的实例更改为 standard 时，会出现以下情况：

- CPUCreditBalance 值保持不变并进行结转。
- 立即针对 CPUSurplusCreditBalance 值进行收费。

在 standard 实例切换到 unlimited 时，会出现以下情况：

- 将结转包含已累积获得的积分的 CPUCreditBalance 值。
- 对于 T2 标准实例，将从 CPUCreditBalance 值中扣除所有启动积分，并且将结转包含已累积获得的积分的剩余 CPUCreditBalance 值。

监控积分使用情况

要了解您实例花费的积分是否超过基准提供的积分，您可以使用 CloudWatch 指标来跟踪使用情况，并且可以设置小时警报，以便获得积分使用情况通知。有关更多信息，请参阅 监控可突增实例的 CPU 积分。

可突增实例的无限模式示例

以下示例介绍当实例配置为 unlimited 时的积分使用情况。

示例

- 示例 1：介绍 T3 无限版的积分使用情况
- 示例 2：介绍 T2 无限版的积分使用情况

示例 1：介绍 T3 无限版的积分使用情况

在此示例中，您可以查看作为 t3.nano 启动的 unlimited 实例的 CPU 利用率，以及它如何花费获得 的积分和超额 积分来保持 CPU 利用率。

t3.nano 实例在 24 小时滚动周期内获得 144 个 CPU 积分，这些积分可兑换 144 分钟 vCPU 使用时间。在实例用完 CPU 积分余额（由 CloudWatch 指标 CPUCreditBalance表示）时，它会花费超额 CPU 积分—（尚未获得的 积分）—以突增所需的时间。由于 t3.nano 实例在 24 小时周期内最多可获得 144 积分，因此，在花费的超额积分不超过该最大积分数时，不会立即向您收费。如果花费 144 个以上的 CPU 积分，则会在该小时结束时对超出的部分进行收费。

下图所示的示例旨在说明实例如何使用超额积分突增到基准以上，甚至在用完 CPUCreditBalance 后。以下工作流程引用图中的编号数据点：

P1 – 在图表中的 0 小时处，实例以 unlimited 模式启动并立即开始获得积分。实例自启动后保持闲置状态（CPU 利用率为 0%），不使用任何积分。所有未使用的积分都累积到积分余额中。对于前 24 小时，CPUCreditUsage 为 0，而 CPUCreditBalance 值达到其最大值 144。

P2 – 对于接下来的 12 小时，CPU 利用率为 2.5%，这低于 5% 基准。实例获得的积分多于花费的积分，但 CPUCreditBalance 值不能超过其最大值 144 积分。

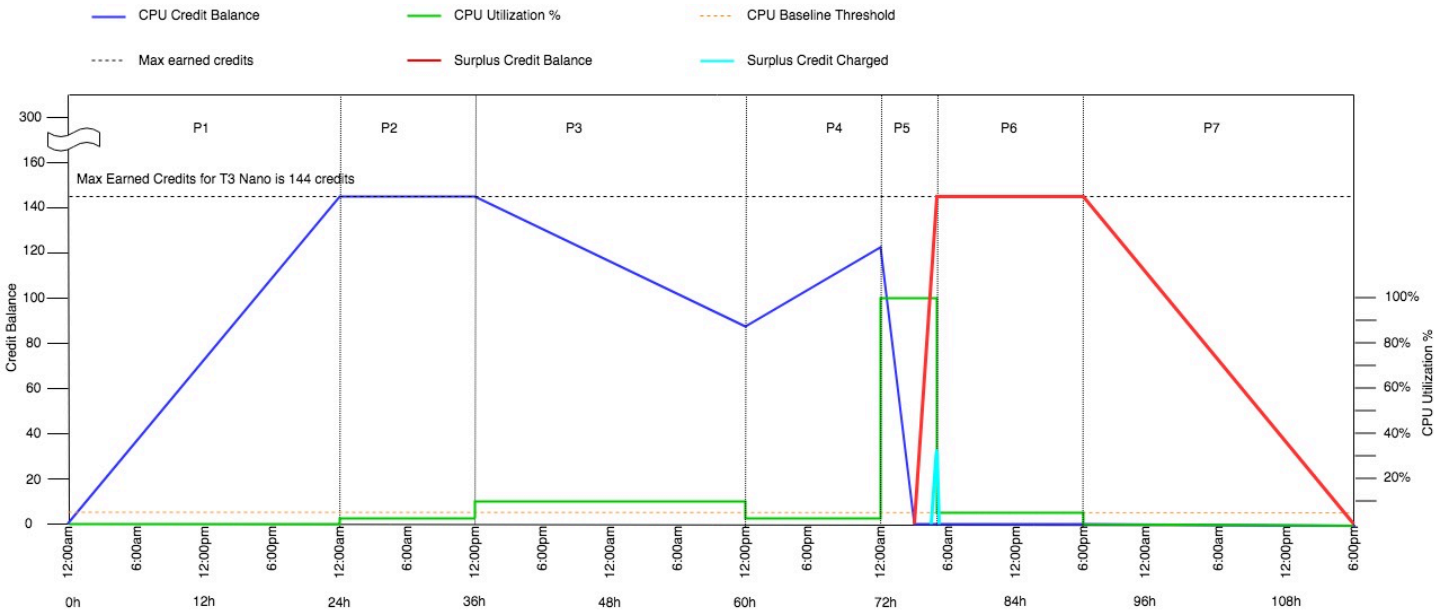
P3 – 对于接下来的 24 小时，CPU 利用率为 7%（高于基准），这要求花费 57.6 积分。实例花费的积分多于获得的积分，CPUCreditBalance 值降至 86.4 积分。

P4 – 对于接下来的 12 小时，CPU 利用率降至 2.5%（低于基准），这要求花费 36 积分。同时，实例获得 72 积分。实例获得的积分多于花费的积分，CPUCreditBalance 值增至 122 积分。

P5 – 对于接下来的 5 小时，实例突增至 100% CPU 利用率，并花费总计 570 积分来持续突增。在进入此期间的大约一小时内，此实例用完其整个 CPUCreditBalance 122 积分，并开始花费超额积分来维持高的 CPU 利用率，在此期间总共花费 448 个超额积分 (570-122=448)。当 CPUSurplusCreditBalance 值达到 144 个 CPU 积分（t3.nano 实例在 24 小时内可获得的最大值）时，之后任何花费的超额积分都无法由获得的积分抵消。之后花费的超额积分总计为 304 积分 (448-144=304)，这会导致这一小时结束后对于这 304 积分收取很小的一笔附加费。

P6 – 对于接下来的 13 小时，CPU 利用率为 5%（基准）。实例获得的积分与花费的积分一样多，而无需额外支付 CPUSurplusCreditBalance 的费用。CPUSurplusCreditBalance 值保持为 144 积分。

P7 – 对于本例中的最后 24 小时，实例空闲，CPU 利用率为 0%。在此期间，实例获得 144 积分，用于支付 CPUSurplusCreditBalance 的费用。



示例 2：介绍 T2 无限版的积分使用情况

在此示例中，您可以查看作为 t2.nano 启动的 unlimited 实例的 CPU 利用率，以及它如何花费获得 的积分和超额 积分来保持 CPU 利用率。

t2.nano 实例在 24 小时滚动周期内获得 72 个 CPU 积分，这些积分可兑换 72 分钟 vCPU 使用时间。在实例用完 CPU 积分余额（由 CloudWatch 指标 CPUCreditBalance表示）时，它会花费超额 CPU 积分—（尚未获得的 积分）—以突增所需的时间。由于 t2.nano 实例在 24 小时周期内最多可获得 72 积分，因此，在花费的超额积分不超过该最大积分数时，不会立即向您收费。如果花费 72 个以上的 CPU 积分，则会在该小时结束时对超出的部分进行收费。

下图所示的示例旨在说明实例如何使用超额积分突增到基准以上，甚至在用完 CPUCreditBalance 后。您可以假定，在图表的时间线开始时，实例累积的积分余额等于它可在 24 小时内获得的最大积分数。以下工作流程引用图中的编号数据点：

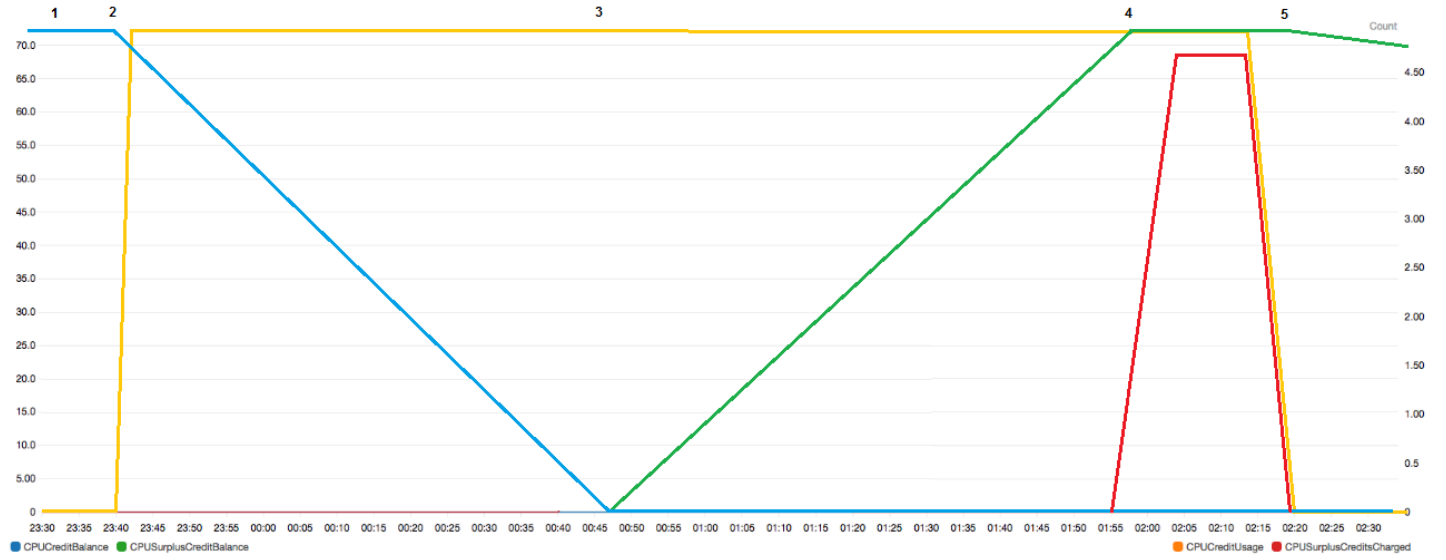
1 – 在前 10 分钟内，CPUCreditUsage 设置为 0 并且 CPUCreditBalance 值始终为最大值 72。

2 – 在 23:40，随着 CPU 使用率增加，实例花费 CPU 积分并且 CPUCreditBalance 值减少。

3 – 在大约 00:47，实例用完全部 CPUCreditBalance，并开始花费超额积分以保持较高的 CPU 利用率。

4 – 一直花费超额积分，直到 1:55，此时 CPUSurplusCreditBalance 值达到 72 个 CPU 积分。这等于 t2.nano 实例在 24 小时周期内可获得的最大积分数。以后花费的任何超额积分无法由 24 小时周期内获得的积分抵消，这会导致在该小时结束时收取少量的额外费用。

5 – 在大约 2:20，实例继续花费超额积分。此时，CPU 使用率低于基准并且实例开始获得积分，每小时 3 积分 (或每 5 分钟 0.25 积分)，它使用这些积分来支付 CPUSurplusCreditBalance。在 CPUSurplusCreditBalance 值减少到 0 后，实例开始在其 CPUCreditBalance 中累积获得积分（每 5 分钟 0.25 积分）。



All metrics | Graphed metrics (4) | Graph options

Label	Details	Statistic	Period	Y Axis	Actions
CPUCreditBalance	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUCreditBalance	Maximum	5 Minutes		
CPUCreditUsage	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUCreditUsage	Maximum	5 Minutes		
CPUSurplusCreditBalance	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUSurplusCreditBalance	Maximum	5 Minutes		
CPUSurplusCreditsCharged	EC2 • InstanceId:i-0aa4b948d7eb37d6b • CPUSurplusCreditsCharged	Maximum	5 Minutes		

计算账单（Linux 实例）

超额积分每 vCPU 小时收取 0.05 美元。在 1:55 和 2:20 之间，实例大约花费 25 个超额积分，这相当于 0.42 个 vCPU 小时。该实例产生的额外费用为 0.42 vCPU 小时 x 0.05 美元/vCPU 小时 = 0.021 美元，舍入到 0.02 美元。下面是该 T2 无限实例的月末账单：

Amazon Elastic Compute Cloud running Linux/UNIX		
$0.0058 per On Demand Linux t2.nano Instance Hour	720.000 Hrs	$4.18

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.05 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.02

计算账单（Windows 实例）

超额积分每 vCPU 小时收取 0.096 美元。在 1:55 和 2:20 之间，实例大约花费 25 个超额积分，这相当于 0.42 个 vCPU 小时。该实例产生的额外费用为 0.42 vCPU 小时 x 0.096 美元/vCPU 小时 = 0.04032 美元，舍入到 0.04 美元。下面是该 T2 无限实例的月末账单：

Amazon Elastic Compute Cloud running Windows		
$0.0081 per On Demand Windows t2.nano Instance Hour	720.000 Hrs	$5.83

Amazon Elastic Compute Cloud T2 CPU Credits		
$0.096 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	$0.04

您可以设置账单提醒以每小时通知一次产生的任何费用，并在必要时采取相应的措施。

可突增性能实例的标准模式

配置为 `standard` 的可突增性能实例适用于具有平均 CPU 利用率的工作负载，它始终低于实例的基准 CPU 利用率。为了突增到基准以上，实例会花费在其 CPU 积分余额中累积的积分。如果实例累积的积分较少，CPU 利用率将逐渐下降到基准水平，因此，在累积的 CPU 积分余额用完时，实例的性能不会急剧下降。有关更多信息，请参阅可突增性能实例的关键概念。

内容

- 可突增实例的标准模式概念
  - 标准可突增性能实例的工作原理
  - 启动积分
  - 启动积分限制
  - 启动积分和获得的积分之间的区别
- 可突增实例的标准模式示例
  - 示例 1：介绍 T3 标准版的积分使用情况
  - 示例 2：介绍 T2 标准版的积分使用情况
    - 第 1 个时段：1 – 24 小时
    - 第 2 个时段：25 – 36 小时
    - 第 3 个时段：37 – 61 小时
    - 第 4 个时段：62 – 72 小时
    - 第 5 个时段：73 – 75 小时
    - 第 6 个时段：76 – 90 小时
    - 第 7 个时段：91 – 96 小时

可突增实例的标准模式概念

standard 是用于可突增性能实例的配置选项。可以随时对正在运行或已停止的实例启用或禁用它。您可以在每个 Amazon 区域的账户级别将 standard 设置为每个可突增性能实例系列的默认积分选项，以便账户中所有新的可突增性能实例都使用默认积分选项启动。

标准可突增性能实例的工作原理

当配置为 standard 的可突增性能实例处于运行状态时，它会以设定的每小时速率（以毫秒级精度）持续获得积分。对于 T2 标准模式，在实例停止后，会丢失积累的全部积分，积分余额会重置为零。在它重新启动后，会接受一组新的启动积分，并开始累积获得积分。对于 T4g、T3a 和 T3 标准实例，CPU 积分余额将在实例停止后保留七天，而后不复存在。如果在七天内启动实例，则不会丢失积分。

T2 标准实例会收到两种类型的 CPU 积分：获得的积分以及启动积分。在 T2 标准实例处于运行状态时，它会以固定的每小时速率 (以毫秒级精度) 持续获得积分。在一开始，该实例尚未获得积分来提供良好的初始体验；因此为了提供良好的初始体验，一开始会收到启动积分，可以先花费，同时累积获得积分。

T4g、T3a 和 T3 实例不会收到启动积分，因为这些实例支持无限模式。无限模式积分配置让 T4g、T3a 和 T3 实例能够根据需要使用尽可能多的 CPU，以便在需要的时间内突增超出基准。

启动积分

在启动时，T2 标准实例的每个 vCPU 会获得 30 个启动积分，而 T1 标准实例会获得 15 个启动积分。例如，t2.micro 实例具有一个 vCPU 并获得 30 个启动积分，而 t2.xlarge 实例具有 4 个 vCPU 并获得 120 个启动积分。启动积分旨在提供良好的初始体验，以使实例能够在启动后 (没有累积获得积分之前) 立即突增到更高的性能。

首先花费启动积分，再使用获得的积分。未花费的启动积分将累积到 CPU 积分余额中，但不会计入 CPU 积分余额限制。例如，t2.micro 实例的 CPU 积分余额限制为 144 个获得的积分。如果实例启动并保持空闲状态 24 小时，其 CPU 积分余额将达到 174 (30 个启动积分 + 144 个获得的积分)，这已超过限制。不过，在实例花费 30 个启动积分后，积分余额就不能超过 144 个。有关每种实例大小的 CPU 积分余额限制的更多信息，请参阅积分表。

下表列出了在启动时分配的初始 CPU 积分以及 vCPU 数。

实例类型	启动积分	vCPU
t1.micro	15	1

实例类型	启动积分	vCPU
t2.nano	30	1
t2.micro	30	1
t2.small	30	1
t2.medium	60	2
t2.large	60	2
t2.xlarge	120	4
t2.2xlarge	240	8

启动积分限制

T2 标准实例接收启动积分的次数存在限制。在每个区域，每 24 个小时的滚动周期内，每个账户中所有 T2 标准实例组合的默认限制是 100 次启动。例如，当一个实例在 24 小时周期内停止并启动 100 次时，或当 100 个实例在 24 小时周期内启动时，或者其他组合等同于 100 次启动时，将达到此限制。新账户可能具有较低的限制，该限制随着时间根据您的使用情况而增加。

> **Tip**
>
> 要确保您的工作负载始终获得所需的性能，请切换到可突增性能实例的无限模式或考虑使用更大的实例。

启动积分和获得的积分之间的区别

下表列出了启动积分和获得的积分之间的区别。

	启动积分	获得的积分
积分获得率	在启动时，T2 标准实例的每个 vCPU 获得 30 个启动积分。	每个 T2 实例以固定的每小时速率 (以毫秒级精度) 持续获得 CPU 积分，具体取决于实例大小。有关每种实例大小获

	启动积分	获得的积分
	如果 T2 实例从 `unlimited` 切换到 `standard`，则在切换时不会获得启动积分。	得的 CPU 积分数量的更多信息，请参阅积分表。
积分获得限制	对于每个区域，在每 24 个小时的滚动周期内，每个账户中所有 T2 标准实例组合的启动积分接收限制是 100 次启动。新账户可能具有较低的限制，该限制随着时间根据您的使用情况而增加。	T2 实例累积的积分数不能超过 CPU 积分余额限制。如果 CPU 积分余额已达到其限制，则将丢弃在达到限制后获得的任何积分。启动积分不计入限制。有关每种 T2 实例大小的 CPU 积分余额限制的更多信息，请参阅积分表。
积分使用	首先花费启动积分，再使用获得的积分。	只有花完所有启动积分后才能花费获得的积分。
过期积分	在 T2 标准实例运行过程中，启动积分不会过期。当 T2 标准实例停止或切换至“T2 无限”时，所有启动积分都将丢失。	在 T2 实例运行过程中，已累积获得的积分不会过期。T2 实例停止后，将丢失所有已累积获得的积分。

CloudWatch 指标 `CPUCreditBalance` 可跟踪已累积的启动积分和已累积获得的积分数。有关更多信息，请参阅 CloudWatch 指标表中的 `CPUCreditBalance`。

可突增实例的标准模式示例

以下示例介绍当实例配置为 `standard` 时的积分使用情况。

示例

- 示例 1：介绍 T3 标准版的积分使用情况
- 示例 2：介绍 T2 标准版的积分使用情况

示例 1：介绍 T3 标准版的积分使用情况

在本示例中，您将了解作为 `t3.nano` 启动的 `standard` 实例如何获得、累积和使用获得 的积分。您可以看到积分余额如何反映累积的获得 的积分。

运行的 t3.nano 实例每 24 小时获得 144 积分。其积分余额限制为 144 个获得的积分。达到该限制后，将丢弃获得的任何新积分。有关可获得和可累积的积分数的更多信息，请参阅积分表。

您可启动 T3 标准实例并立即使用它。或者，您可能在启动 T3 标准实例后让其闲置几天，再在该实例上运行应用程序。实例是正在被使用还是闲置决定积分是消耗还是累积。如果实例从启动时开始保持闲置状态 24 小时，则积分余额将达到其限制，这是可以累积的获得积分的最大数目。

本示例介绍启动后闲置 24 小时的实例，并向您分析 96 小时内共 7 个时段的积分情况，演示获得、累积、使用和丢弃积分的速率以及每个时段结束时的积分余额值。

以下工作流程引用图中的编号数据点：

P1 – 在图表中的 0 小时处，实例以 standard 模式启动并立即开始获得积分。实例自启动后保持闲置状态（CPU 利用率为 0%），不使用任何积分。所有未使用的积分都累积到积分余额中。对于前 24 小时，CPUCreditUsage 为 0，而 CPUCreditBalance 值达到其最大值 144。

P2 – 对于接下来的 12 小时，CPU 利用率为 2.5%，这低于 5% 基准。实例获得的积分多于花费的积分，但 CPUCreditBalance 值不能超过其最大值 144 积分。所获得的超过限制的所有积分都会被丢弃。

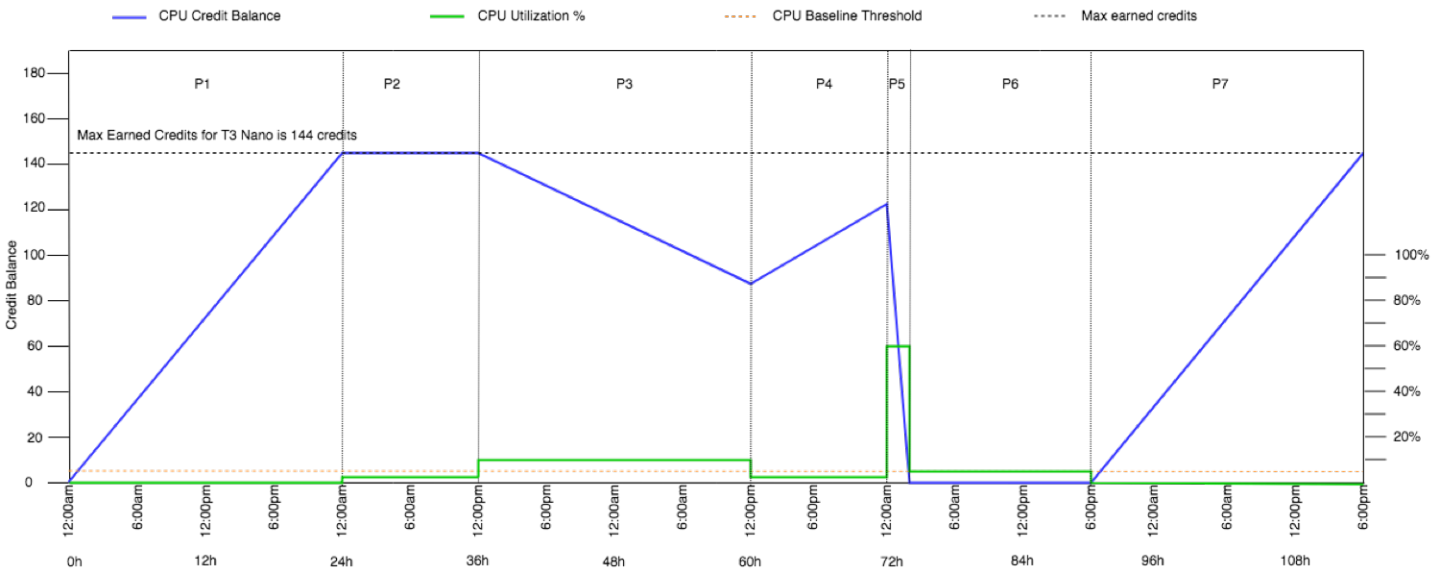
P3 – 对于接下来的 24 小时，CPU 利用率为 7%（高于基准），这要求花费 57.6 积分。实例花费的积分多于获得的积分，CPUCreditBalance 值降至 86.4 积分。

P4 – 对于接下来的 12 小时，CPU 利用率降至 2.5%（低于基准），这要求花费 36 积分。同时，实例获得 72 积分。实例获得的积分多于花费的积分，CPUCreditBalance 值增至 122 积分。

P5 – 对于接下来的两个小时，实例突增至 60% CPU 利用率，并耗尽其整个 CPUCreditBalance 值的 122 积分。在此期间结束时，CPUCreditBalance 为零，CPU 利用率会被强制降低到基准利用率级别 5%。在基准时，实例获得的积分与花费的积分一样多。

P6 – 对于接下来的 14 小时，CPU 利用率为 5%（基准）。实例获得的积分与花费的积分一样多。CPUCreditBalance 值保持为 0。

P7 – 对于本例中的最后 24 小时，实例空闲，CPU 利用率为 0%。在此期间，实例获得 144 积分，这些积分将累积到其 CPUCreditBalance 中。



示例 2：介绍 T2 标准版的积分使用情况

在本示例中，您将了解作为 `t2.nano` 启动的 `standard` 实例如何获得、累积和使用启动积分和获得的积分。您还可以了解积分余额如何反映累积获得的积分和累积启动积分。

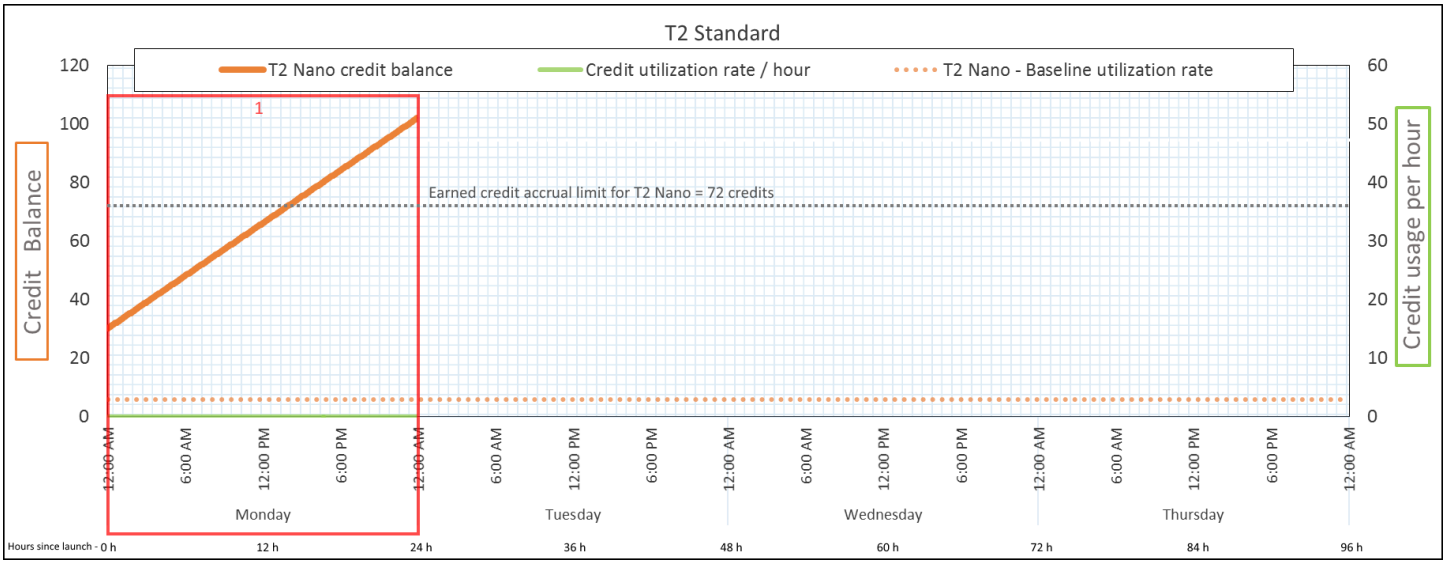
启动时，`t2.nano` 实例获得 30 启动积分，之后每 24 小时获得 72 积分。其积分余额限制是获得的 72 积分；启动积分不计入该限制。达到该限制后，将丢弃获得的任何新积分。有关可获得和可累积的积分数的更多信息，请参阅积分表。有关 限制的更多信息，请参阅 启动积分限制。

您可启动 T2 标准实例并立即使用它。或者，您可能在启动 T2 标准实例后让其闲置几天，再在该实例上运行应用程序。实例是正在被使用还是闲置决定积分是消耗还是累积。如果实例自启动后闲置 24 小时，积分余额将超过其限制，因为积分余额同时反映累积获得的积分和累积启动积分。不过，使用 CPU 后，会先使用启动积分。此后，积分余额限制始终反映可累积获得的最大积分。

本示例介绍启动后闲置 24 小时的实例，并向您分析 96 小时内共 7 个时段的积分情况，演示获得、累积、使用和丢弃积分的速率以及每个时段结束时的积分余额值。

第 1 个时段：1 – 24 小时

在图上的第 0 小时，T2 实例作为 `standard` 启动并立即获得 30 启动积分。当它处于运行状态时，会获得积分。实例自启动后保持闲置状态（CPU 利用率为 0%），不使用任何积分。所有未使用的积分都累积到积分余额中。在启动后大约 14 小时，积分余额为 72 (30 启动积分 + 获得的 42 积分)，这与实例在 24 小时内获得的积分相同。在启动后 24 小时，积分余额超过 72，因为未使用的启动积分累积到了积分余额中 (积分余额为 —102 积分：30 启动积分 + 72 获得积分)。



积分使用率	每 24 小时 0 积分 (0% CPU 利用率)
积分获得率	每 24 小时 72 积分
积分丢弃率	每 24 小时 0 积分
积分余额	102 积分 (30 启动积分 + 获得的 72 积分)

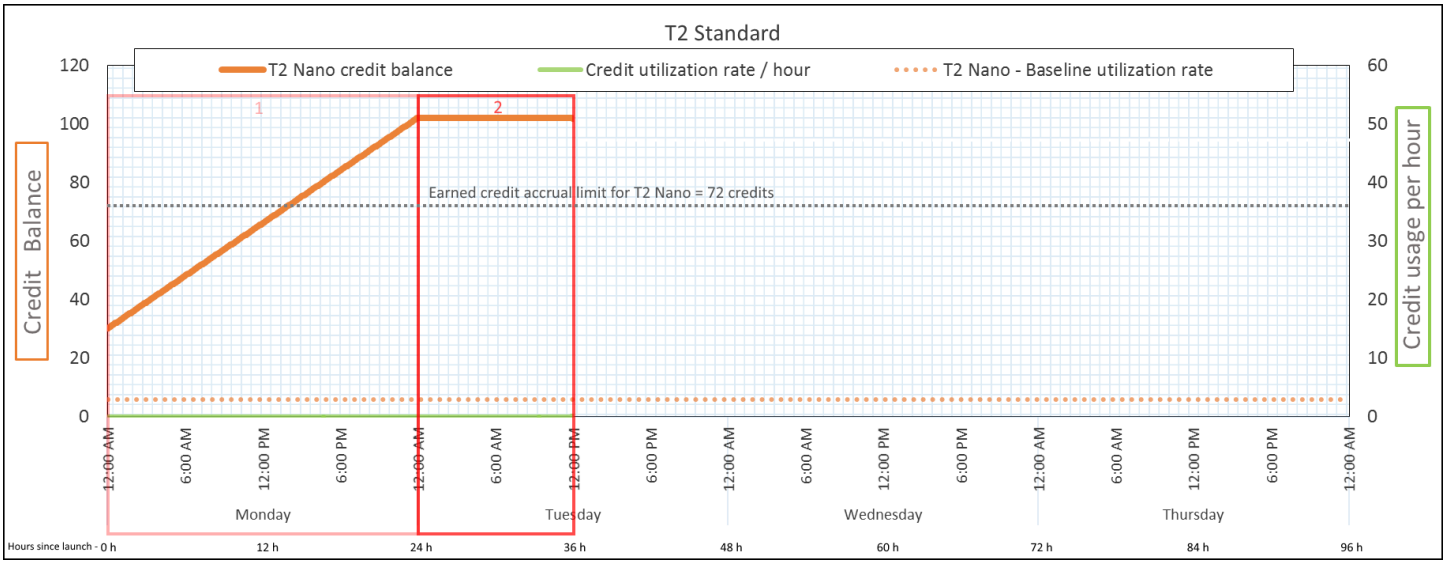
结论

如果启动后没有 CPU 利用率，实例累积的积分将超过其在 24 小时内获得的积分 (30 启动积分 + 获得的 72 积分 = 102 积分)。

在真实场景中，EC2 实例在启动和运行时会使用少量积分，以防止积分余额达到本实例中的最大理论值。

第 2 个时段：25 – 36 小时

在接下来 12 小时，实例继续保持闲置状态并获得积分，但积分余额不会增加。积分余额保持在 102 (30 启动积分 + 获得的 72 积分)。积分余额已达到 72 累积获得的积分限制，因此会丢弃新获得的积分。



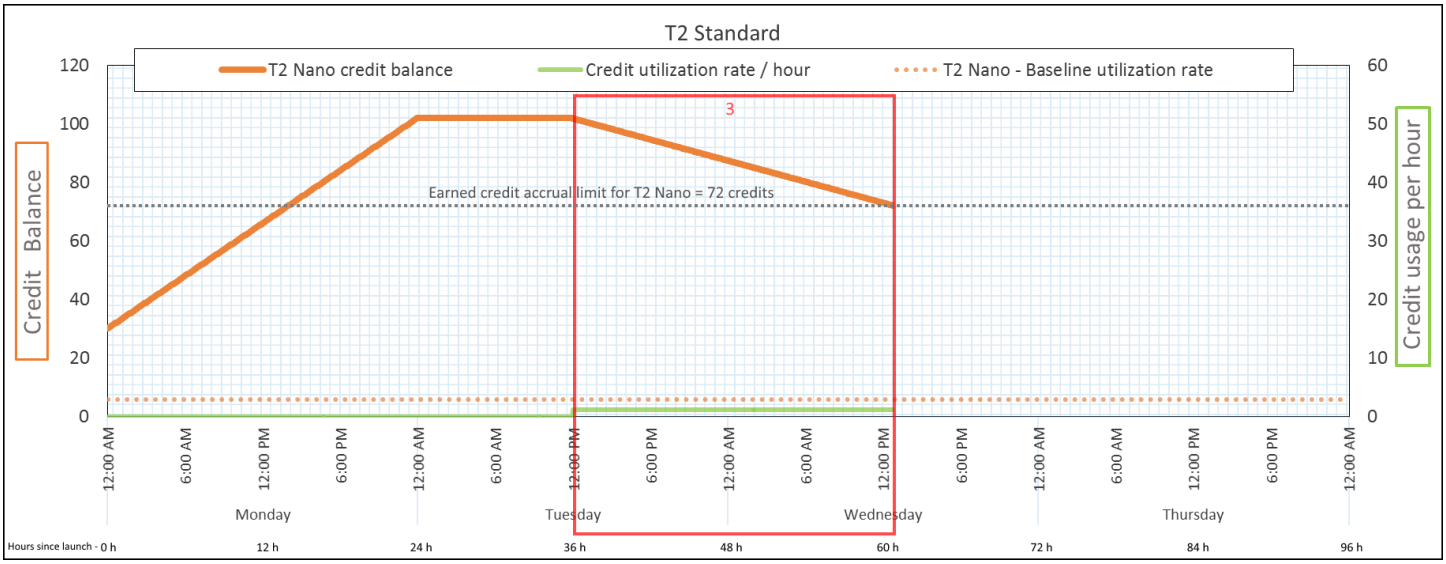
积分使用率	每 24 小时 0 积分 (0% CPU 利用率)
积分获得率	每 24 小时 72 积分 (每小时 3 积分)
积分丢弃率	每 24 小时 72 积分 (100% 积分获得率)
积分余额	102 积分 (30 启动积分 + 72 获得积分) — 余额保持不变

结论

如果积分余额已达到其限制，实例会继续获得积分，但不会累积更多获得的积分。达到该限制后，会丢弃新获得的积分。启动积分不计入 积分余额限制。如果余额包含累积的启动积分，余额将超过该限制。

第 3 个时段：37 – 61 小时

在接下来 25 小时，实例使用 2% CPU，需要 30 积分。在同一周期，它获得 75 积分，但积分余额减少。余额减少的原因是先使用累积的启动 积分，并且由于积分余额已达到其获得的 72 积分限制，因此丢弃了新获得的积分。



积分使用率	24 小时 28.8 积分 (每小时 1.2 积分，2% CPU 利用率，40% 积分获得率) — 25 小时 30 积分
积分获得率	每 24 小时 72 积分
积分丢弃率	每 24 小时 72 积分 (100% 积分获得率)
积分余额	72 积分 (使用了 30 启动积分；剩余获得的 72 积分未使用)

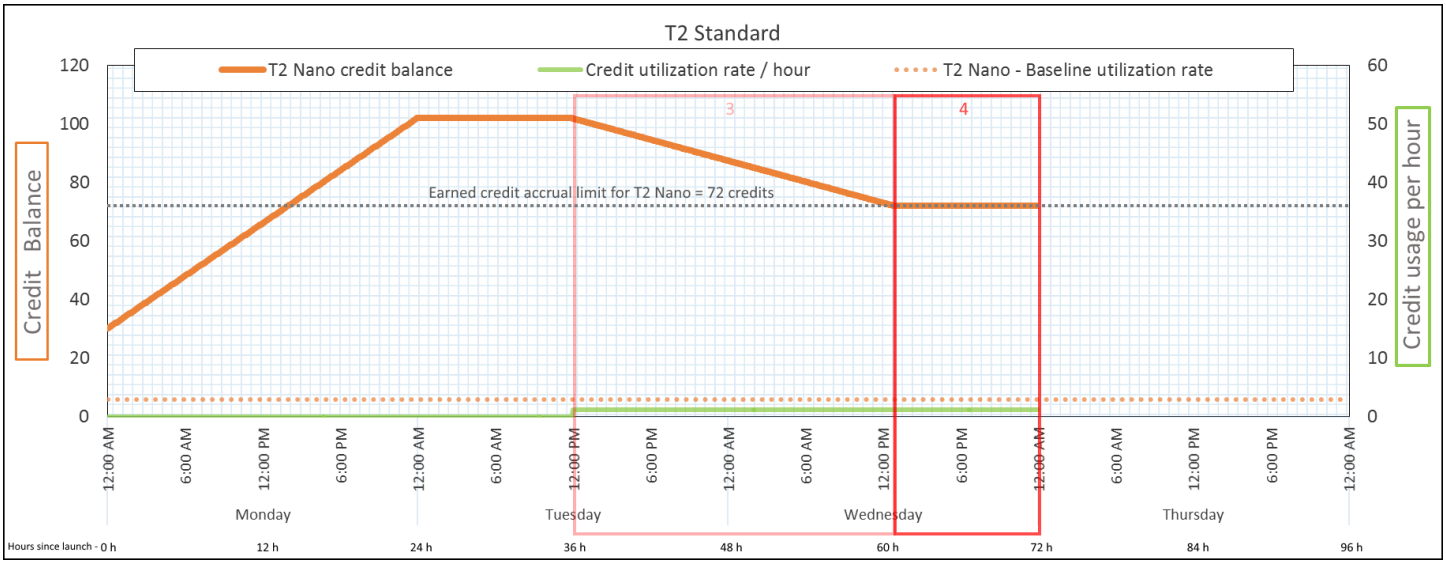
结论

实例先使用启动积分，再使用获得的积分。启动积分不计入积分限制。使用启动积分后，积分余额永远不会超过在 24 小时内可获得的积分。此外，实例运行时，不会获得更多启动积分。

第 4 个时段：62 – 72 小时

在接下来 11 小时，实例使用 2% CPU，需要 13.2 积分。这与上一周期的 CPU 利用率相同，但积分余额不会减少。它保持在 72 积分。

积分余额不减少的原因是积分获得率高于积分使用率。实例使用 13.2 积分的同时，获得 33 积分。不过，由于余额限制是 72 积分，因此会丢弃获得的超过该限制的任何积分。积分余额保持在 72 积分，这与第 2 个时段保持在 102 积分不同，因为没有累积的启动积分。



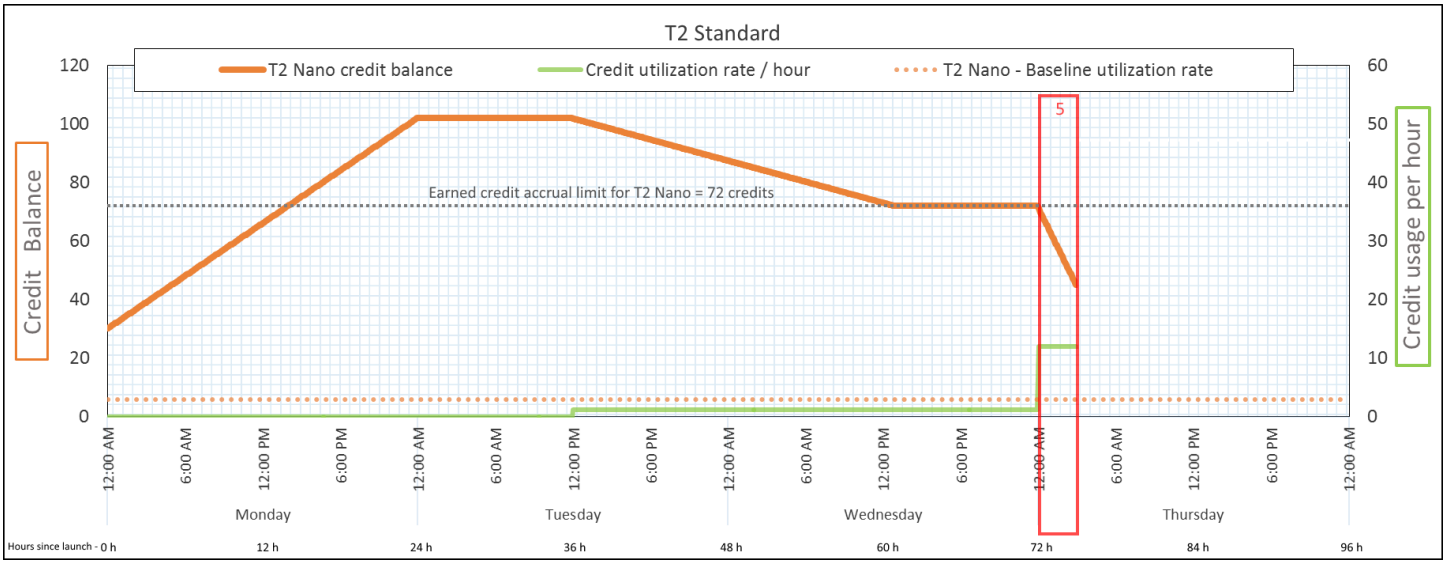
积分使用率	24 小时 28.8 积分 (每小时 1.2 积分，2% CPU 利用率，40% 积分获得率) — 11 小时 13.2 积分
积分获得率	每 24 小时 72 积分
积分丢弃率	每 24 小时 43.2 积分 (60% 积分获得率)
积分余额	72 积分 (0 启动积分，获得的 72 积分) — 余额达到其限制

结论

使用启动积分后，积分余额限制由实例在 24 小时内可获得的积分数决定。如果实例获得的积分多于使用的积分，则会丢弃新获得的超过限制的积分。

第 5 个时段：73 – 75 小时

在接下来 3 小时，实例的 CPU 利用率激增至 20%，需要 36 积分。在相同的 3 小时内，实例获得 9 积分，导致净余额减少 27 积分。在这 3 小时结束时，积分余额为累积获得的 45 积分。



积分使用率	24 小时 288 积分 (每小时 12 积分，20% CPU 利用率，400% 积分获得率) — 3 小时 36 积分
积分获得率	每 24 小时 72 积分 (3 小时 9 积分)
积分丢弃率	每 24 小时 0 积分
积分余额	45 积分 (以前的余额 (72) - 使用的积分 (36) + 获得的积分 (9)) — 余额按每 24 小时 216 积分的速率减少 (使用率 288/24 + 获得率 72/24 = 余额减少率 216/24)

结论

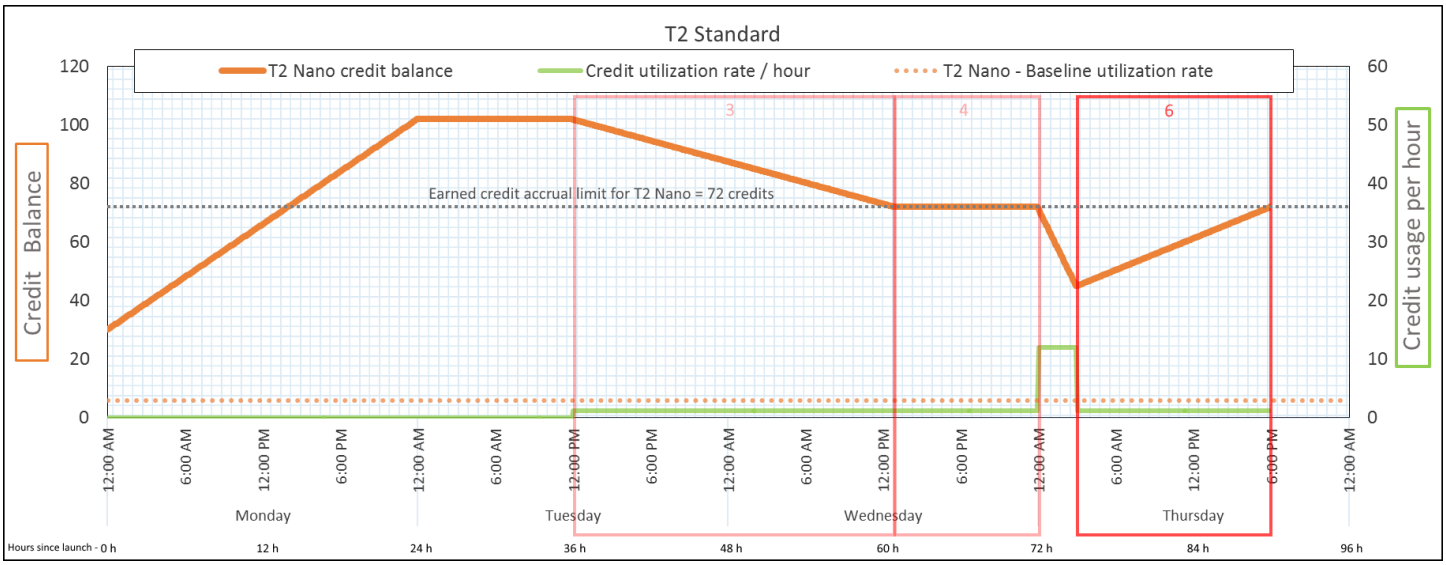
如果实例使用的积分多于获得的积分，则其积分余额将减少。

第 6 个时段：76 – 90 小时

在接下来 15 小时，实例使用 2% CPU，需要 18 积分。这与第 3 个和第 4 个时段的 CPU 利用率相同。不过，此周期的积分余额增加，而第 3 个时段的积分余额减少，第 4 个时段的保持不变。

在第 3 个时段，使用累积的启动积分，并会丢弃获得的超过积分限制的任何积分，导致积分余额减少。在第 4 个时段，实例发挥的积分数少于其获得的积分数。所获得的任何超出限制的积分将丢弃，因此余额保持在其最大值 72 积分。

在本周期，没有累积的启动积分，余额中累积获得的积分数低于限制。不会丢弃获得的任何积分。此外，实例获得的积分多于使用的积分，导致积分余额增加。



积分使用率	24 小时 28.8 积分 (每小时 1.2 积分，2% CPU 利用率，40% 积分获得率) — 15 小时 18 积分
积分获得率	每 24 小时 72 积分 (15 小时 45 积分)
积分丢弃率	每 24 小时 0 积分
积分余额	72 积分 (余额按每 24 小时 43.2 积分的速率增加 — 更改率 = 使用率 28.8/24 + 获得率 72/24)

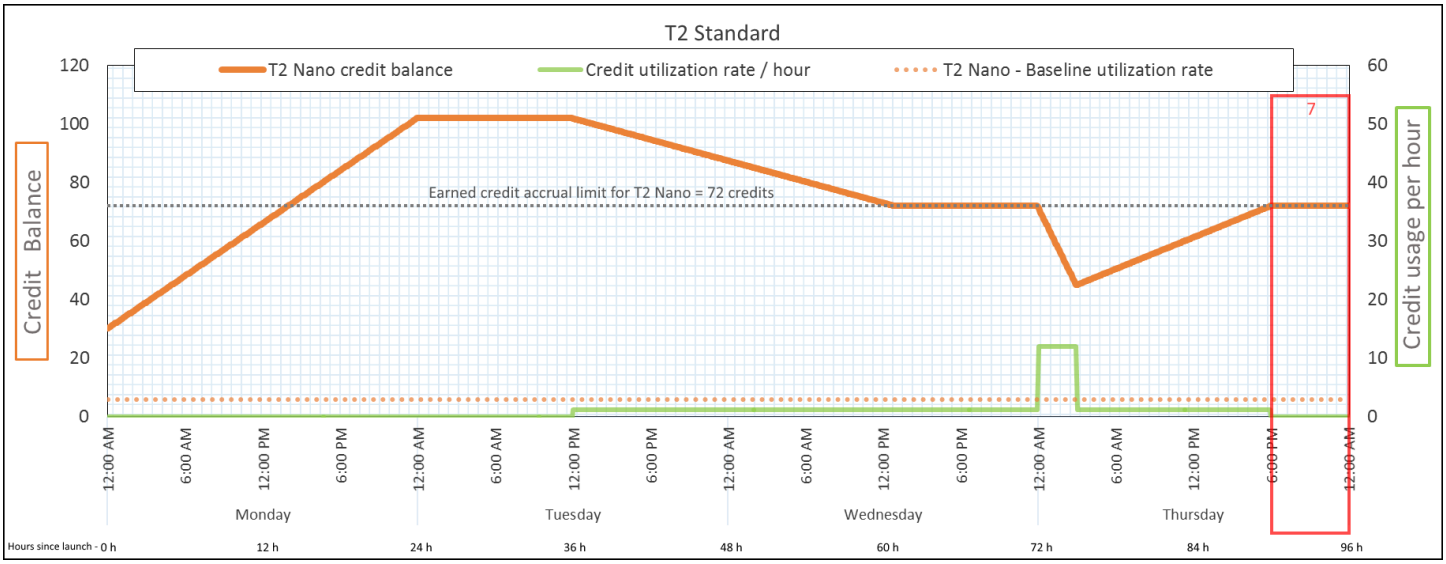
结论

如果实例使用的积分少于获得的积分，则其积分余额将增加。

第 7 个时段：91 – 96 小时

在接下来六小时，实例保持闲置状态— (CPU 利用率为 0%) ，—不使用任何积分。这与第 2 个时段的 CPU 利用率相同，但积分余额不保持在 102 积分，而保持在 72 积分，—这是实例的积分余额限制。

在第 2 个时段，积分余额中包含累积的 30 启动积分。启动积分在第 3 个时段使用。正在运行的实例无法获得更多启动积分。达到积分余额限制后，会丢弃获得的超过限制的任何积分。



积分使用率	每 24 小时 0 积分 (0% CPU 利用率)
积分获得率	每 24 小时 72 积分
积分丢弃率	每 24 小时 72 积分 (100% 积分获得率)
积分余额	72 积分 (0 启动积分，获得的 72 积分)

结论

如果已达到积分余额限制，实例会继续获得积分，但不会累积更多获得的积分。达到该限制后，会丢弃新获得的积分。积分余额限制由实例在 24 小时内可获得的积分数决定。有关积分余额限制的更多信息，请参阅积分表。

配置可突增性能实例

用于启动、监控和修改可突增性能实例(T 实例)的步骤是类似的。主要差别在于它们启动时的默认积分规范。

每个 T 实例系列都具有以下默认积分规范：

- T4g、T3a 和 T3 实例以 `unlimited` 模式启动
- 专属主机上的 T3 实例仅可以作为 `standard` 启动
- T2 实例以 `standard` 模式启动

您可以更改账户的默认积分规范。

任务

- 在启动时配置积分规范
- 配置自动扩缩组以将积分规范设置为 unlimited
- 管理可突增性能实例的积分规范
- 管理账户的默认积分规范

在启动时配置积分规范

您可以启动积分规范为 `unlimited` 或 `standard` 的 T 实例。

以下过程介绍了如何使用 EC2 控制台或 Amazon CLI。有关如何使用自动扩缩组的信息，请参阅 配置自动扩缩组以将积分规范设置为 unlimited。

Console

在启动时配置实例的积分规范

1. 按照程序启动实例。
2. 在 Instance type（实例类型）下，选择一个 T 实例类型。
3. 展开高级详细信息。在积分规范中选择积分规范。
4. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。

Amazon CLI

在启动时设置实例的积分规范

使用 run-instances 命令和 `--credit-specification` 选项。

```
--credit-specification CpuCredits=unlimited
```

PowerShell

在启动时设置实例的积分规范

将 New-EC2Instance cmdlet 与 `-CreditSpecification_CpuCredit` 参数结合使用。

```
-CreditSpecification_CpuCredit unlimited
```

配置自动扩缩组以将积分规范设置为 unlimited

在启动 T 实例时，它们需要具有 CPU 积分才能获得良好的引导体验。如果您使用自动扩缩组启动实例，建议您将实例配置为 unlimited。如果这样做，实例会在自动启动或者由自动扩缩组重新启动时使用超额积分。使用超额积分可以防止受到性能限制。

创建启动模板

在自动扩缩组中以 unlimited 模式启动实例时，您必须使用启动模板。启动配置不支持以 unlimited 模式启动实例。

Console

创建可用于设置积分规范的启动模板

1. 请遵照《Amazon EC2 Auto Scaling 用户指南》中的使用高级设置创建启动模板。
2. 在启动模板内容中，对于实例类型，请选择实例大小。
3. 要在自动扩缩组中以 unlimited 模式启动实例，请在高级详细信息下，对于积分规范选择无限。
4. 在您完成后，定义启动模板参数，选择创建启动模板。

Amazon CLI

创建可用于设置积分规范的启动模板

使用 create-launch-template 命令。

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --version-description FirstVersion \
    --launch-template-data CreditSpecification={CpuCredits=unlimited}
```

PowerShell

创建可用于设置积分规范的启动模板

使用 New-EC2LaunchTemplate cmdlet。按如下方式定义启动模板数据的积分规范。

```
$creditSpec = New-Object Amazon.EC2.Model.CreditSpecificationRequest
$creditSpec.CpuCredits = "unlimited"
$launchTemplateData = New-Object Amazon.EC2.Model.RequestLaunchTemplateData
```

```
$launchTemplateData.CreditSpecification = $creditSpec
```

关联自动扩缩组与启动模板

要将启动模板与一个自动扩缩组相关联，请使用启动模板创建自动扩缩组，或者将启动模板添加到现有自动扩缩组中。

Console

使用启动模板创建自动扩缩组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，选择在创建启动模板时使用的同一区域。
3. 在导航窗格中，依次选择自动扩缩组和创建自动扩缩组。
4. 选择启动模板，选择您的启动模板，然后选择下一步。
5. 填写自动扩缩组的各个字段。当您在审核页面上完成审核配置设置时，选择创建自动扩缩组。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的使用启动模板创建自动扩缩组。

添加启动模板到现有自动扩缩组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，选择在创建启动模板时使用的同一区域。
3. 在导航窗格中，选择 Auto Scaling Groups。
4. 从自动扩缩组列表中选择一个自动扩缩组，然后依次选择操作和编辑。
5. 在详细信息选项卡上，对于启动模板，选择一个启动模板，然后选择保存。

Amazon CLI

使用启动模板创建自动扩缩组

使用 create-auto-scaling-group 命令并指定 `--launch-template` 参数。

添加启动模板到现有自动扩缩组

使用 update-auto-scaling-group 命令并指定 `--launch-template` 参数。

PowerShell

使用启动模板创建自动扩缩组

使用 New-ASAutoScalingGroup cmdlet 并指定 -LaunchTemplate_LaunchTemplateId 或 -LaunchTemplate_LaunchTemplateName 参数。

添加启动模板到现有自动扩缩组

使用 Update-ASAutoScalingGroup cmdlet 并指定 -LaunchTemplate_LaunchTemplateId 或 -LaunchTemplate_LaunchTemplateName 参数。

管理可突增性能实例的积分规范

您可以随时将正在运行或停止的 T 实例的积分规范在 unlimited 与 standard 之间切换。

请注意，在 unlimited 模式中，实例可能会花费超额积分，从而产生额外费用。有关更多信息，请参阅 超额积分会产生费用。

Console

管理实例的积分规范

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. （可选）选择实例。在详细信息选项卡上，找到积分规范。该值为 unlimited 或 standard。
4. （可选）要一次修改多个实例的积分规范，请选择所有实例。
5. 依次选择操作、实例设置和更改积分规范。仅当您选择了 T 实例时，此选项才会启用。
6. 如要启用或停用无限模式，选中或清除每个实例 ID 旁边的复选框。

Amazon CLI

获取实例的积分规范

使用 describe-instance-credit-specifications 命令。如果您不指定实例 ID，则将返回积分规范为 unlimited 的所有实例。输出还将包括先前配置了 unlimited 积分规范的实例。例如，如果您将 T3 实例大小调整为 M4 实例，而该实例配置为 unlimited，Amazon EC2 将返回 M4 实例。

```
aws ec2 describe-instance-credit-specifications \
    --instance-id i-1234567890abcdef0 \
    --query InstanceCreditSpecifications[].CpuCredits \
    --output text
```

下面是示例输出。

```
unlimited
```

设置实例的积分规范

使用 modify-instance-credit-specification 命令。

```
aws ec2 modify-instance-credit-specification \
    --region us-east-1 \
    --instance-credit-specification
 "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

PowerShell

获取实例的积分规范

使用 Get-EC2CreditSpecification cmdlet。

```
(Get-EC2CreditSpecification `
    -InstanceId i-1234567890abcdef0).CpuCredits
```

下面是示例输出。

```
unlimited
```

设置实例的积分规范

使用 Edit-EC2InstanceCreditSpecification cmdlet。

```
Edit-EC2InstanceCreditSpecification `
    -Region us-east-1 `
    -InstanceCreditSpecification @({InstanceId="i-1234567890abcdef0"
 CpuCredits="unlimited"})
```

管理账户的默认积分规范

每个 T 实例系列都有默认积分规范。您可以在每个 Amazon 区域的账户级别更改每个 T 实例系列的默认积分规范。默认积分规范的有效值为 `unlimited` 和 `standard`。

如果您使用 EC2 控制台中的启动实例向导来启动实例，则您为积分规范选择的值会覆盖账户级别的默认积分规范。如果您使用 Amazon CLI 启动实例，则账户中所有新的 T 实例都使用默认积分规范启动。现有正在运行或已停止的实例的积分规范不受影响。

考虑因素

实例系列的默认积分规范在 5 分钟滚动周期内只能修改一次，在 24 小时滚动周期内最多可修改四次。

Console

管理默认积分规范

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择控制面板。
4. 在账户属性卡的设置下，选择默认服务抵扣金规格。
5. 选择管理。
6. 对于每个实例系列，选择 Unlimited (无限) 或 Standard (标准)，然后选择 Update (更新)。

Amazon CLI

获取默认积分规范

使用 get-default-credit-specification 命令。

```
aws ec2 get-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --query InstanceFamilyCreditSpecifications[].CpuCredits \
    --output text
```

下面是示例输出。

```
standard
```

设置默认积分规范

使用 modify-default-credit-specification 命令。以下示例将该值设置为 unlimited。

```
aws ec2 modify-default-credit-specification \
    --region us-east-1 \
    --instance-family t2 \
    --cpu-credits unlimited
```

PowerShell

获取默认积分规范

使用 Get-EC2DefaultCreditSpecification cmdlet。

```
(Get-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2).CpuCredits
```

下面是示例输出。

```
standard
```

设置默认积分规范

使用 Edit-EC2DefaultCreditSpecification cmdlet。以下示例将该值设置为 unlimited。

```
Edit-EC2DefaultCreditSpecification `
    -Region us-east-1 `
    -InstanceFamily t2 `
    -CpuCredit unlimited
```

监控可突增实例的 CPU 积分

EC2 将指标发送到 Amazon CloudWatch。您可以在 CloudWatch 控制台的 Amazon EC2 每个实例指标中查看 CPU 积分指标，也可以使用 Amazon CLI 列出每个实例的指标。有关更多信息，请参阅 可用于实例的 CloudWatch 指标。

内容

- 可突增性能实例的其他 CloudWatch 指标

- 计算使用的 CPU 积分

可突增性能实例的其他 CloudWatch 指标

可突增性能实例具有这些额外的 CloudWatch 指标，将每 5 分钟更新一次这些指标：

- `CPUCreditUsage` – 在测量周期内花费的 CPU 积分数。
- `CPUCreditBalance` – 实例产生的 CPU 积分数量。在 CPU 突增以及 CPU 积分的花费速度比获得速度快时，该余额将用完。
- `CPUSurplusCreditBalance` – 在 `CPUCreditBalance` 值为零时，用于保持 CPU 利用率而花费的超额 CPU 积分数。
- `CPUSurplusCreditsCharged` – 超过可在 24 小时内获得的 CPU 积分数上限的超额 CPU 积分数，因而会产生额外的费用。

最后两个指标仅适用于配置为 `unlimited` 的实例。

下表描述了可突增性能实例的 CloudWatch 指标。有关更多信息，请参阅可用于实例的 CloudWatch 指标。

指标	描述
`CPUCreditUsage`	实例为保持 CPU 使用率而花费的 CPU 积分数。一个 CPU 积分等于一个 vCPU 按 100% 利用率运行一分钟，或者 vCPU、利用率和时间的等效组合（例如，一个 vCPU 按 50% 利用率运行两分钟，或者两个 vCPU 按 25% 利用率运行两分钟）。 CPU 积分指标仅每 5 分钟提供一次。如果您指定一个大于五分钟的时间段，请使用`Sum` 统计数据，而非 `Average` 统计数据。 单位：积分 (vCPU 分钟)
`CPUCreditBalance`	实例自启动后已累积获得的 CPU 积分数。对于 T2 标准，`CPUCreditBalance` 还包含已累积的启动积分数。 在获得积分后，积分将在积分余额中累积；在花费积分后，将从积分余额中扣除积分。积分余额具有最大值限制，这是由实例大小决定的。在达到限制后，将丢弃获得的任何新积分。对于 T2 标准，启动积分不计入限制。

指标	描述
	实例可以花费 CPUCreditBalance 中的积分，以便突增到基准 CPU 使用率以上。 在实例运行过程中，CPUCreditBalance 中的积分不会过期。在 T4g、T3a 或 T3 实例停止时，CPUCreditBalance 值将保留七天。之后，所有累积的积分都将丢失。在 T2 实例停止时，CPUCreditBalance 值不会保留，并且所有累积的积分都将丢失。 CPU 积分指标仅每 5 分钟提供一次。 单位：积分 (vCPU 分钟)
CPUSurplusCreditBalance	在 unlimited 值为零时，CPUCreditBalance 实例花费的超额积分数。 CPUSurplusCreditBalance 值由获得的 CPU 积分支付。如果超额积分数超出实例可在 24 小时周期内获得的最大积分数，则超出最大积分数的已花费超额积分将产生额外费用。 单位：积分 (vCPU 分钟)
CPUSurplusCreditsCharged	未由获得的 CPU 积分支付并且会产生额外费用的已花费超额积分数。 在出现以下任一情况时，将对花费的超额积分收费： • 花费的超额积分超出实例可在 24 小时周期内获得的最大积分数。对于超出最大积分数的所花费超额积分，将在该小时结束时向您收费。 • 实例已停止或终止。 • 实例从 unlimited 切换为 standard。 单位：积分 (vCPU 分钟)

计算使用的 CPU 积分

实例使用的 CPU 积分使用上表中所述的实例 CloudWatch 指标计算。

Amazon EC2 每 5 分钟向 CloudWatch 发送一次指标。在任何时间点引用的以前 指标值是指 5 分钟前发送的以前指标值。

计算标准实例使用的 CPU 积分

- 如果 CPU 使用率低于基准，此时花费的积分低于前 5 分钟间隔获得的积分，CPU 积分余额将增加。
- 如果 CPU 使用率高于基准，此时花费的积分超过前 5 分钟间隔获得的积分，CPU 积分余额将减少。

从数学上讲，这是使用以下公式得出的：

Example

```
CPUCreditBalance = prior CPUCreditBalance + [Credits earned per hour * (5/60) -
 CPUCreditUsage]
```

实例大小确定实例每小时可获得的积分数以及实例可在积分余额中累积获得的积分数。有关每小时获得的积分数的信息，以及每个实例大小的积分余额限制，请参阅积分表。

示例

该示例使用 `t3.nano` 实例。要计算实例的 `CPUCreditBalance` 值，请按以下方式使用前面的公式：

- `CPUCreditBalance` – 要计算的当前积分余额。
- `prior CPUCreditBalance` – 5 分钟前的积分余额。在该示例中，实例累积了两积分。
- `Credits earned per hour` – `t3.nano` 实例每小时获得 6 积分。
- `5/60` – 表示 CloudWatch 指标发布的 5 分钟间隔。将每小时获得的积分乘以 5/60（5 分钟）以计算实例在过去 5 分钟获得的积分数。`t3.nano` 实例每 5 分钟获得 0.5 积分。
- `CPUCreditUsage` – 实例在过去 5 分钟内花费的积分数。在该示例中，实例在过去 5 分钟内花费 1 积分。

您可以使用这些值计算 `CPUCreditBalance` 值：

Example

```
CPUCreditBalance = 2 + [0.5 - 1] = 1.5
```

计算无限实例使用的 CPU 积分

在可突增性能实例需要突增到基准以上时，它始终先花费累积的积分，然后再花费超额积分。在用完累积的 CPU 积分余额时，它会花费超额积分以将 CPU 突增所需的时间。在 CPU 使用率低于基准时，在实例累积获得的积分之前始终先支付超额积分。

我们在以下公式中使用 Adjusted balance 项以反映在该 5 分钟间隔内发生的活动。我们使用该值计算 CPUCreditBalance 和 CPUSurplusCreditBalance CloudWatch 指标的值。

Example

```
Adjusted balance = [prior CPUCreditBalance - prior CPUSurplusCreditBalance] + [Credits
 earned per hour * (5/60) - CPUCreditUsage]
```

如果 0 的值为 Adjusted balance，表示实例花费获得的所有积分来进行突增，而未花费任何超额积分。因此，CPUCreditBalance 和 CPUSurplusCreditBalance 均设置为 0。

正的 Adjusted balance 值表示实例累积获得了积分，并支付了以前的超额积分 (如果有)。因此，将 Adjusted balance 值分配给 CPUCreditBalance，并将 CPUSurplusCreditBalance 设置为 0。实例大小决定了可累积的最大积分数。

Example

```
CPUCreditBalance = min [max earned credit balance, Adjusted balance]
CPUSurplusCreditBalance = 0
```

负的 Adjusted balance 值表示实例花费了其累积获得的所有积分，并且还花费了超额积分来进行突增。因此，将 Adjusted balance 值分配给 CPUSurplusCreditBalance，并将 CPUCreditBalance 设置为 0。此外，实例大小决定了它可累积的最大积分数。

Example

```
CPUSurplusCreditBalance = min [max earned credit balance, -Adjusted balance]
CPUCreditBalance = 0
```

如果花费的超额积分超过了实例可累积的最大积分，超额积分余额将设置为最大值，如前面的公式中所示。将对剩余的超额积分收费，如 CPUSurplusCreditsCharged 指标表示。

Example

```
CPUSurplusCreditsCharged = max [-Adjusted balance - max earned credit balance, 0]
```

最后，在实例终止时，将对由 CPUSurplusCreditBalance 跟踪的任何超额积分收费。如果实例从 unlimited 切换到 standard，则还会对任何剩余的 CPUSurplusCreditBalance 收费。

使用 GPU 实例加速性能

基于 GPU 的实例能让您访问具有数千个计算内核的 NVIDIA GPU。您可以通过这些实例利用 CUDA 或开放计算语言 (OpenCL) 并行计算框架，为科学、工程和渲染应用程序加速。还可以将这些实例用于图形应用程序，包括游戏流式处理、3-D 应用流式处理和其他图形工作负载。

在激活或优化基于 GPU 的实例之前，您必须安装相应的驱动程序，如下所示：

- 要在挂载 NVIDIA GPU 的实例（例如，P3 或 G4dn 实例）上安装 NVIDIA 驱动程序，请参阅 NVIDIA 驱动程序。
- 要在挂载 AMD GPU 的实例（例如，G4ad 实例）上安装 AMD 驱动程序，请参阅 AMD 驱动程序。

内容

- 在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序
- 优化 Amazon EC2 实例上的 GPU 设置
- 在 G4ad Linux 实例上设置双 4K 显示屏
- 开始使用 GPU 加速实例

在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序

要激活基于 GPU 的实例上的 GRID 虚拟应用程序（默认情况下启用 NVIDIA GRID 虚拟工作站），您必须为驱动程序定义产品类型。您使用的过程取决于实例的操作系统。

Linux 实例

激活 Linux 实例上的 GRID 虚拟应用程序

1. 从提供的模板文件创建 /etc/nvidia/gridd.conf 文件。

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. 在您常用的文本编辑器中打开 /etc/nvidia/gridd.conf 文件。
3. 找到 FeatureType 行，并将其设置为 0。然后，添加包含 IgnoreSP=TRUE 的行。

```
FeatureType=0 IgnoreSP=TRUE
```

4. 保存文件并退出。
5. 重启实例以接受新配置。

```
[ec2-user ~]$ sudo reboot
```

Windows 实例

激活 Windows 实例上的 GRID 虚拟应用程序

1. 运行 regedit.exe 以打开注册表编辑器。
2. 导航到 HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global \GridLicensing。
3. 打开右侧窗格上的上下文 (右键单击) 菜单，选择新建，然后选择 DWORD。
4. 对于 Name (名称)，输入 FeatureType，然后按 Enter。
5. 打开 FeatureType 上的上下文 (右键单击) 菜单，然后选择修改。
6. 对于值数据，请输入 NVIDIA GRID 虚拟应用的 0 并选择 OK。
7. 打开右侧窗格上的上下文 (右键单击) 菜单，选择新建，然后选择 DWORD。
8. 对于名称，输入 IgnoreSP，然后按 Enter。
9. 打开 IgnoreSP 上的上下文 (右键单击) 菜单，然后选择修改。
10. 对于值数据，键入 1，然后选择确定。
11. 关闭注册表编辑器。

优化 Amazon EC2 实例上的 GPU 设置

您可以执行多个 GPU 设置优化，以实现 NVIDIA GPU 实例的最佳性能。对于其中一些实例类型，NVIDIA 驱动程序使用 autoboost 功能，该功能会改变 GPU 时钟速度。通过禁用 autoboost 并将 GPU 时钟速度设置为其最大频率，您可以始终获得 GPU 实例的最大性能。

在 Linux 上优化 GPU 设置

1. 将 GPU 设置配置为永久。该命令可能需要几分钟才能运行完毕。

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. [仅限 G3 实例] 禁用实例上所有 GPU 的 autoboost 功能。

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
```

3. 将所有 GPU 时钟速度设置为其最大频率。使用以下命令中指定的内存和图形时钟速度。

   有些版本的 NVIDIA 驱动程序不支持设置应用程序时钟速度，并会显示错误 "Setting applications clocks is not supported for GPU..."，您可以忽略该错误。

   - G3 实例：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
     ```

   - G4dn 实例：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
     ```

   - G5 实例：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 6250,1710
     ```

   - G6、Gr6f、Gr6 和 Gr6f 实例：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 6251,2040
     ```

   - G6e 实例：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 9001,2520
     ```

   - G7e 实例：

     ```
     [ec2-user ~]$ sudo nvidia-smi -ac 12481,2430
     ```

   - P3 和 P3dn 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 877,1530
```

- P4d 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 1215,1410
```

- P4de 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 1593,1410
```

- P5 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 2619,1980
```

- P5e 和 P5en 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 3201,1980
```

- P6-B200 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 3996,1965
```

- P6-B300 实例：

```
[ec2-user ~]$ sudo nvidia-smi -ac 3996,2032
```

在 Windows 上优化 GPU 设置

1. 打开 PowerShell 窗口并导航到 NVIDIA 安装文件夹。

```
PS C:\> cd "C:\Windows\System32\DriverStore\FileRepository\nvgridsw_aws.inf_*\"
```

2. [仅限 G3 实例] 禁用实例上所有 GPU 的 autoboost 功能。

```
PS C:\> .\nvidia-smi --auto-boost-default=0
```

3. 将所有 GPU 时钟速度设置为其最大频率。使用以下命令中指定的内存和图形时钟速度。

有些版本的 NVIDIA 驱动程序不支持设置应用程序时钟速度，并会显示错误 "Setting applications clocks is not supported for GPU..."，您可以忽略该错误。

- G3 实例：

```
PS C:\> .\nvidia-smi -ac "2505,1177"
```

- G4dn 实例：

```
PS C:\> .\nvidia-smi -ac "5001,1590"
```

- G5 实例：

```
PS C:\> .\nvidia-smi -ac "6250,1710"
```

- G6、Gr6f、Gr6 和 Gr6f 实例：

```
PS C:\> .\nvidia-smi -ac "6251,2040"
```

- G6e 实例：

```
PS C:\> .\nvidia-smi -ac "9001,2520"
```

- P3 和 P3dn 实例：

```
PS C:\> .\nvidia-smi -ac "877,1530"
```

在 G4ad Linux 实例上设置双 4K 显示屏

启动 G4ad 实例后，您可以设置双 4K 显示屏。

安装 AMD 驱动程序并配置双屏

1. 连接到您的 Linux 实例以获取相关 GPU 的 PCI 总线地址，以用作双 4K（2x4k）的目标：

```
lspci -vv | grep -i amd
```

您将获得与下内容类似的输出：

```
00:1e.0 Display controller: Advanced Micro Devices, Inc. [*AMD*/ATI] Device 7362 (rev
 c3)
Subsystem: Advanced Micro Devices, Inc. [AMD/ATI] Device 0a34
```

2. 请注意，上述输出中的 PCI 总线地址为 00:1e.0。创建一个名为 /etc/modprobe.d/amdgpu.conf 的文件然后添加：

```
options amdgpu virtual_display=0000:00:1e.0,2
```

3. 要在 Linux 上安装 AMD 驱动程序，请参阅 EC2 实例的 AMD 驱动程序。如果您已经安装了 AMD GPU 驱动程序，则需要通过 dkms 重建 amdgpu 内核模块。

4. 使用下面的 xorg.conf 文件定义双（2x4K）屏幕拓扑并将文件保存到 /etc/X11/xorg.conf：

```
~$ cat /etc/X11/xorg.conf
Section "ServerLayout"
    Identifier     "Layout0"
    Screen          0 "Screen0"
    Screen        1 "Screen1"
    InputDevice     "Keyboard0" "CoreKeyboard"
    InputDevice     "Mouse0" "CorePointer"
    Option          "Xinerama" "1"
EndSection
Section "Files"
    ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
    ModulePath "/opt/amdgpu/lib/xorg/modules"
    ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
    ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
    ModulePath "/usr/lib64/xorg/modules"
    ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
    # generated from default
    Identifier     "Mouse0"
    Driver         "mouse"
    Option         "Protocol" "auto"
    Option         "Device" "/dev/psaux"
    Option         "Emulate3Buttons" "no"
    Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
    # generated from default
```

```
    Identifier     "Keyboard0"
    Driver         "kbd"
EndSection

Section "Monitor"
    Identifier     "Virtual"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "Primary" "true"
EndSection

Section "Monitor"
    Identifier     "Virtual-1"
    VendorName     "Unknown"
    ModelName      "Unknown"
    Option         "RightOf" "Virtual"
EndSection

Section "Device"
    Identifier     "Device0"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection

Section "Device"
    Identifier     "Device1"
    Driver         "amdgpu"
    VendorName     "AMD"
    BoardName      "Radeon MxGPU V520"
    BusID          "PCI:0:30:0"
EndSection

Section "Extensions"
    Option         "DPMS" "Disable"
EndSection

Section "Screen"
    Identifier     "Screen0"
    Device         "Device0"
    Monitor        "Virtual"
    DefaultDepth   24
    Option         "AllowEmptyInitialConfiguration" "True"
```

```
    SubSection "Display"
        Virtual    3840 2160
        Depth      32
    EndSubSection
EndSection

Section "Screen"
    Identifier     "Screen1"
    Device         "Device1"
    Monitor        "Virtual"
    DefaultDepth   24
    Option         "AllowEmptyInitialConfiguration" "True"
    SubSection "Display"
        Virtual    3840 2160
        Depth      32
    EndSubSection
EndSection
```

5. 按照设置交互式桌面中的说明设置 DCV。
6. 完成 DCV 设置后，重新启动。
7. 确认驱动程序运行正常：

```
dmesg | grep amdgpu
```

该响应应当与以下内容相似：

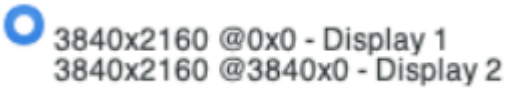
```
Initialized amdgpu
```

8. 您应会在 DISPLAY=:0 xrandr -q 的输出中看到您已连接了 2 个虚拟显示器：

```
~$ DISPLAY=:0 xrandr -q
Screen 0: minimum 320 x 200, current 3840 x 1080, maximum 16384 x 16384
Virtual connected primary 1920x1080+0+0 (normal left inverted right x axis y axis)
 0mm x 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
```

```
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
 1280x800 59.96
 1280x720 59.97
 1152x768 59.95
 1024x768 60.00 59.95
 800x600  60.32 59.96 56.25
 848x480  60.00 59.94
 720x480  59.94
 640x480  59.94 59.94
Virtual-1 connected 1920x1080+1920+0 (normal left inverted right x axis y axis) 0mm x
 0mm
 4096x3112  60.00
 3656x2664  59.99
 4096x2160  60.00
 3840x2160  60.00
 1920x1200  59.95
 1920x1080  60.00
 1600x1200  59.95
 1680x1050  60.00
 1400x1050  60.00
 1280x1024  59.95
 1440x900 59.99
 1280x960 59.99
 1280x854 59.95
 1280x800 59.96
 1280x720 59.97
 1152x768 59.95
 1024x768 60.00 59.95
 800x600  60.32 59.96 56.25
 848x480  60.00 59.94
 720x480  59.94
640x480  59.94 59.94
```

9. 连接到 DCV 后，将分辨率更改为 2x4K，确认 DCV 已经注册了双显示器支持。



为 Linux 设置交互式桌面

确认 Linux 实例已安装 AMD GPU 驱动程序并且 amdgpu 正在使用中之后，您可以安装交互式桌面管理器。我们建议使用 MATE 桌面环境，以获得最佳兼容性和性能。

先决条件

打开文本编辑器并将以下内容另存为名为 xorg.conf 的文件。您的实例上需要此文件。

```
Section "ServerLayout"
Identifier     "Layout0"
Screen          0 "Screen0"
InputDevice     "Keyboard0" "CoreKeyboard"
InputDevice     "Mouse0" "CorePointer"
EndSection
Section "Files"
ModulePath "/opt/amdgpu/lib64/xorg/modules/drivers"
ModulePath "/opt/amdgpu/lib/xorg/modules"
ModulePath "/opt/amdgpu-pro/lib/xorg/modules/extensions"
ModulePath "/opt/amdgpu-pro/lib64/xorg/modules/extensions"
ModulePath "/usr/lib64/xorg/modules"
ModulePath "/usr/lib/xorg/modules"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Mouse0"
Driver         "mouse"
Option         "Protocol" "auto"
Option         "Device" "/dev/psaux"
Option         "Emulate3Buttons" "no"
Option         "ZAxisMapping" "4 5"
EndSection
Section "InputDevice"
# generated from default
Identifier     "Keyboard0"
Driver         "kbd"
EndSection
Section "Monitor"
Identifier     "Monitor0"
VendorName     "Unknown"
ModelName      "Unknown"
EndSection
Section "Device"
Identifier     "Device0"
```

```
Driver         "amdgpu"
VendorName     "AMD"
BoardName      "Radeon MxGPU V520"
BusID          "PCI:0:30:0"
EndSection
Section "Extensions"
Option         "DPMS" "Disable"
EndSection
Section "Screen"
Identifier     "Screen0"
Device         "Device0"
Monitor        "Monitor0"
DefaultDepth   24
Option         "AllowEmptyInitialConfiguration" "True"
SubSection "Display"
    Virtual    3840 2160
    Depth      32
EndSubSection
EndSection
```

在 Amazon Linux 2 上设置交互式桌面

1. 安装 EPEL 存储库。

```
[ec2-user ~]$ sudo amazon-linux-extras install epel -y
```

2. 安装 MATE 桌面。

```
[ec2-user ~]$ sudo amazon-linux-extras install mate-desktop1.x -y
[ec2-user ~]$ sudo yum groupinstall "MATE Desktop" -y
[ec2-user ~]$ sudo systemctl disable firewalld
```

3. 将 `xorg.conf` 文件复制到 `/etc/X11/xorg.conf`。
4. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

5. （可选）安装 Amazon DCV 服务器以将 Amazon DCV 作为高性能显示协议，然后使用您偏好的客户端连接到 Amazon DCV 会话。

在 Ubuntu 上设置交互式桌面

1. 安装 MATE 桌面。

```
$ sudo apt install xorg-dev ubuntu-mate-desktop -y
$ sudo apt purge ifupdown -y
```

2. 将 xorg.conf 文件复制到 /etc/X11/xorg.conf。
3. 重启实例。

```
$ sudo reboot
```

4. 为适当版本的 Ubuntu 安装 AMF 编码器。

```
$ sudo apt install ./amdgpu-pro-20.20-*/amf-amdgpu-pro_20.20-*_amd64.deb
```

5. （可选）安装 Amazon DCV 服务器以将 Amazon DCV 作为高性能显示协议，然后使用您偏好的客户端连接到 Amazon DCV 会话。
6. DCV 安装完成后，授予 DCV 用户视频权限：

```
$ sudo usermod -aG video dcv
```

在 CentOS 上设置交互式桌面

1. 安装 EPEL 存储库。

```
$ sudo yum update -y
$ sudo yum install epel-release -y
```

2. 安装 MATE 桌面。

```
$ sudo yum groupinstall "MATE Desktop" -y
$ sudo systemctl disable firewalld
```

3. 将 xorg.conf 文件复制到 /etc/X11/xorg.conf。
4. 重启实例。

```
$ sudo reboot
```

5. （可选）安装 Amazon DCV 服务器以将 Amazon DCV 作为高性能显示协议，然后使用您偏好的客户端连接到 Amazon DCV 会话。

开始使用 GPU 加速实例

最新一代 GPU 加速实例类型（如以下列表所示的类型）为深度学习和高性能计算 (HPC) 应用程序提供了最高性能的功能。选择实例类型链接以了解有关其功能的更多信息。

- P6 系列
- P6 系列
- P5 系列

有关加速实例类型的实例类型规范的完整列表，请参阅 Amazon EC2 实例类型参考中的加速计算。

软件配置

开始使用最新一代 GPU 加速实例类型的最简单方法是从预先配置了所有必需软件的 Amazon 深度学习 AMI 启动实例。有关与 GPU 加速实例类型配合使用的最新 Amazon Deep Learning AMIs，请参阅《Amazon Deep Learning AMIs 开发人员指南》中的 P6 Supported DLAMI。

如果需要构建自定义 AMI 来启动托管深度学习或 HPC 应用程序的实例，则建议在基础映像上安装以下最低软件版本。

实例类型	NVIDIA 驱动程序	CUDA	NVIDIA GDRCopy	EFA 安装程序	NCCL	EFA K8s
G7e	575	12.9	2.5	1.45.0	2.28.3	0.5.10
P5	530	12.1	2.3	1.24.1	2.18.3	0.4.4
p5.4xLarge	530	12.1	2.3	1.43.1 [2]	2.18.3	0.4.4
P5e	550	12.1	2.3	1.24.1	2.18.3	0.5.5
P5en	550	12.1	2.3	1.24.1	2.18.3	0.5.6
P6-B200	570	12.8	2.5	1.41.0	2.26.2-1	0.5.10

实例类型	NVIDIA 驱动程序	CUDA	NVIDIA GDRCopy	EFA 安装程序	NCCL	EFA K8s
P6e-GB200	570	12.8	2.5	1.41.0	2.26.2-1	0.5.10
P6-B300	580	13.0	2.5	1.44.0	2.28.3	0.5.10

[1] EFA K8s 列包含 `aws-efa-k8s-device-plugin` 的最低推荐版本。

[2] 当 GPU 到 GPU 通信使用 Elastic Fabric Adapter (EFA) 和 NVIDIA 集体通信库 (NCCL) 时，存在影响 `P5.4xlarge` 实例的兼容性问题。要缓解此问题，需将环境变量 `FI_HMEM_DISABLE_P2P` 设置为 `1`，并确保安装了 1.43.1 或更高版本的 EFA。

> **Note**
>
> 如果您使用 EFA 安装程序 1.41.0 版本，则 `aws-ofi-nccl plugin` 会附带它。对于早期版本的 EFA 安装程序，请使用 `aws-ofi-nccl plugin` 版本 `1.7.2-aws` 或更高版本。

还会建议您将实例配置为不使用深层 C 状态。有关更多信息，请参阅《Amazon Linux 2 User Guide》中的 High performance and low latency by limiting deeper C-states。最新的 Amazon 深度学习基础 GPU AMI 已预先配置为不使用深层 C 状态。

有关网络和 Elastic Fabric Adapter（EFA）配置，请参阅 使用多网卡最大化 Amazon EC2 实例上的网络带宽。

Amazon EBS 优化的实例类型

Amazon EBS 优化的实例使用经过优化的配置堆栈，并为 Amazon EBS I/O 提供额外的专用带宽。这种优化通过最小化 Amazon EBS I/O 与来自您实例的其他流量之间的争用，为您的 EBS 卷提供最佳性能。

当附加到 EBS 优化的实例时，通用型 SSD（`gp2` 和 `gp3`）卷可在给定年度 99% 的时间内提供至少 90% 的预调配 IOPS 性能，而预调配 IOPS SSD（`io1` 和 `io2`）卷可在给定年度 99.9% 的时间内提供至少 90% 的预调配 IOPS 性能。吞吐量优化型 HDD（`st1`）和冷 HDD（`sc1`）在给定年度 99% 的时间内提供至少 90% 的预期吞吐量性能。不合规时间段大致均匀分布，目标是达到 99% 的每小时预计总吞吐量。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 卷类型。

某些实例类型默认为 EBS 优化，无需启用，尝试禁用也不会有任何影响。其他实例类型可选支持 EBS 优化，您可以在启动期间或之后启用该功能，但需额外支付每小时费用。某些实例类型不支持 EBS 优化。

有关详细的实例类型规范和功能，请参阅 Amazon EC2 实例类型指南。

注意事项

- 实例的 EBS 性能受实例类型的性能限制或其附加卷的聚合性能的限制，以较小者为准。要实现最大的 EBS 性能，实例必须附加卷以提供等于或大于最大实例性能的总性能。例如，要实现 `r6i.16xlarge` 的 `80,000` IOPS，实例必须至少具有预置了 `16,000` IOPS 的 5 个 `gp2` 卷（5 个卷 x `16,000` IOPS = `80,000` IOPS），或具有预置了 `80,000` IOPS 的 1 个 `gp3` 卷。建议您选择一种实例类型，其提供的专用 Amazon EBS 吞吐量高于您的应用程序需求；否则，Amazon EBS 与 Amazon EC2 的连接将成为性能瓶颈。
- 您可以挂载到实例的最大 Amazon EBS 卷数取决于实例类型和实例规模。有关更多信息，请参阅 Amazon EC2 实例的 Amazon EBS 卷限制。
- 最大 IOPS 和吞吐量限制是相互依赖的。根据您的 I/O 大小，您可能会先达到一个限制，然后再达到另一个限制，这可能会影响整体性能。为获得最佳结果，请在计划工作负载时同时考虑这两个限制。

默认为 EBS 优化

以下实例类型在默认情况下会进行 EBS 优化。无需启用 EBS 优化，禁用 EBS 优化也没有效果。

实例

- 通用型
- 计算优化型
- 内存优化型
- 存储优化
- 加速计算型
- 高性能计算

通用型

> **Note**
>
> M8a、M8g、M8gd、M8i、M8id、M8i-flex、M9g、M9gd 实例类型支持可配置的带宽权重。使用这些实例类型，您可以针对网络性能或 Amazon EBS 性能优化实例的带宽。下表显示了这些实例类型的默认 Amazon EBS 带宽性能。有关更多信息，请参阅 EC2 实例带宽权重配置。

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
a1.medium [1]	300	3500	37.50	437.50	2500	20000
a1.large [1]	525	3500	65.62	437.50	4000	20000
a1.xlarge [1]	800	3500	100.00	437.50	6000	20000
a1.2xlarge [1]	1750	3500	218.75	437.50	10000	20000
a1.4xlarge [2]	3500		437.5		20000	
a1.metal [2]	3500		437.5		20000	
m4.large [2]	450		56.25		3600	
m4.xlarge [2]	750		93.75		6000	
m4.2xlarge [2]	1000		125.0		8000	
m4.4xlarge [2]	2000		250.0		16000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m4.10xlarge [2]	4000		500.0		32000	
m4.16xlarge [2]	10000		1250.0		65000	
m5.large [1]	650	4750	81.25	593.75	3600	18750
m5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5.4xlarge [2]	4750		593.75		18750	
m5.8xlarge [2]	6800		850.0		30000	
m5.12xlarge [2]	9500		1187.5		40000	
m5.16xlarge [2]	13600		1700.0		60000	
m5.24xlarge [2]	19000		2375.0		80000	
m5.metal [2]	19000		2375.0		80000	
m5a.large [1]	650	2880	81.25	360.00	3600	16000
m5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5a.4xlarge [2]	2880		360.0		16000	
m5a.8xlarge [2]	4750		593.75		20000	
m5a.12xlarge [2]	6780		847.5		30000	
m5a.16xlarge [2]	9500		1187.5		40000	
m5a.24xlarge [2]	13750		1718.75		60000	
m5ad.large [1]	650	2880	81.25	360.00	3600	16000
m5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
m5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
m5ad.4xlarge [2]	2880		360.0		16000	
m5ad.8xlarge [2]	4750		593.75		20000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m5ad.12xlarge [2]	6780		847.5		30000	
m5ad.16xlarge [2]	9500		1187.5		40000	
m5ad.24xlarge [2]	13750		1718.75		60000	
m5d.large [1]	650	4750	81.25	593.75	3600	18750
m5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5d.4xlarge [2]	4750		593.75		18750	
m5d.8xlarge [2]	6800		850.0		30000	
m5d.12xlarge [2]	9500		1187.5		40000	
m5d.16xlarge [2]	13600		1700.0		60000	
m5d.24xlarge [2]	19000		2375.0		80000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m5d.metal [2]	19000		2375.0		80000	
m5dn.large [1]	650	4750	81.25	593.75	3600	18750
m5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5dn.4xlarge [2]	4750		593.75		18750	
m5dn.8xlarge [2]	6800		850.0		30000	
m5dn.12xlarge [2]	9500		1187.5		40000	
m5dn.16xlarge [2]	13600		1700.0		60000	
m5dn.24xlarge [2]	19000		2375.0		80000	
m5dn.metal [2]	19000		2375.0		80000	
m5n.large [1]	650	4750	81.25	593.75	3600	18750

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
m5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
m5n.4xlarge [2]	4750		593.75		18750	
m5n.8xlarge [2]	6800		850.0		30000	
m5n.12xlarge [2]	9500		1187.5		40000	
m5n.16xlarge [2]	13600		1700.0		60000	
m5n.24xlarge [2]	19000		2375.0		80000	
m5n.metal [2]	19000		2375.0		80000	
m5zn.large [1]	800	3170	100.00	396.25	3333	13333
m5zn.xlarge [1]	1564	3170	195.50	396.25	6667	13333
m5zn.2xlarge [2]	3170		396.25		13333	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m5zn.3xlarge [2]	4750		593.75		20000	
m5zn.6xlarge [2]	9500		1187.5		40000	
m5zn.12xlarge [2]	19000		2375.0		80000	
m5zn.metal [2]	19000		2375.0		80000	
m6a.large [1]	650	10000	81.25	1250.00	3600	40000
m6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6a.8xlarge [2]	10000		1250.0		40000	
m6a.12xlarge [2]	15000		1875.0		60000	
m6a.16xlarge [2]	20000		2500.0		80000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m6a.24xlarge [2]	30000		3750.0		120000	
m6a.32xlarge [2]	40000		5000.0		160000	
m6a.48xlarge [2]	40000		5000.0		240000	
m6a.metal [2]	40000		5000.0		240000	
m6g.medium [1]	315	4750	39.38	593.75	2500	20000
m6g.large [1]	630	4750	78.75	593.75	3600	20000
m6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6g.4xlarge [2]	4750		593.75		20000	
m6g.8xlarge [2]	9500		1187.5		40000	
m6g.12xlarge [2]	14250		1781.25		50000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m6g.16xlarge [2]	19000		2375.0		80000	
m6g.metal [2]	19000		2375.0		80000	
m6gd.medium [1]	315	4750	39.38	593.75	2500	20000
m6gd.large [1]	630	4750	78.75	593.75	3600	20000
m6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
m6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
m6gd.4xlarge [2]	4750		593.75		20000	
m6gd.8xlarge [2]	9500		1187.5		40000	
m6gd.12xlarge [2]	14250		1781.25		50000	
m6gd.16xlarge [2]	19000		2375.0		80000	
m6gd.metal [2]	19000		2375.0		80000	
m6i.large [1]	650	10000	81.25	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6i.8xlarge [2]	10000		1250.0		40000	
m6i.12xlarge [2]	15000		1875.0		60000	
m6i.16xlarge [2]	20000		2500.0		80000	
m6i.24xlarge [2]	30000		3750.0		120000	
m6i.32xlarge [2]	40000		5000.0		160000	
m6i.metal [2]	40000		5000.0		160000	
m6id.large [1]	650	10000	81.25	1250.00	3600	40000
m6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m6id.8xlarge [2]	10000		1250.0		40000	
m6id.12xlarge [2]	15000		1875.0		60000	
m6id.16xlarge [2]	20000		2500.0		80000	
m6id.24xlarge [2]	30000		3750.0		120000	
m6id.32xlarge [2]	40000		5000.0		160000	
m6id.metal [2]	40000		5000.0		160000	
m6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
m6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m6idn.8xlarge [2]	25000		3125.0		100000	
m6idn.12xlarge [2]	37500		4687.5		150000	
m6idn.16xlarge [2]	50000		6250.0		200000	
m6idn.24xlarge [2]	75000		9375.0		300000	
m6idn.32xlarge [2]	100000		12500.0		400000	
m6idn.metal [2]	100000		12500.0		400000	
m6in.large [1]	1562	25000	195.31	3125.00	6250	100000
m6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
m6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
m6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
m6in.8xlarge [2]	25000		3125.0		100000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m6in.12xlarge [2]	37500		4687.5		150000	
m6in.16xlarge [2]	50000		6250.0		200000	
m6in.24xlarge [2]	75000		9375.0		300000	
m6in.32xlarge [2]	100000		12500.0		400000	
m6in.metal [2]	100000		12500.0		400000	
m7a.medium [1]	325	10000	40.62	1250.00	2500	40000
m7a.large [1]	650	10000	81.25	1250.00	3600	40000
m7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7a.8xlarge [2]	10000		1250.0		40000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m7a.12xlarge [2]	15000		1875.0		60000	
m7a.16xlarge [2]	20000		2500.0		80000	
m7a.24xlarge [2]	30000		3750.0		120000	
m7a.32xlarge [2]	40000		5000.0		160000	
m7a.48xlarge [2]	40000		5000.0		240000	
m7a.metal-48xl [2]	40000		5000.0		240000	
m7g.medium [1]	315	10000	39.38	1250.00	2500	40000
m7g.large [1]	630	10000	78.75	1250.00	3600	40000
m7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m7g.8xlarge [2]	10000		1250.0		40000	
m7g.12xlarge [2]	15000		1875.0		60000	
m7g.16xlarge [2]	20000		2500.0		80000	
m7g.metal [2]	20000		2500.0		80000	
m7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m7gd.large [1]	630	10000	78.75	1250.00	3600	40000
m7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7gd.8xlarge [2]	10000		1250.0		40000	
m7gd.12xlarge [2]	15000		1875.0		60000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m7gd.16xlarge [2]	20000		2500.0		80000	
m7gd.metal [2]	20000		2500.0		80000	
m7i.large [1]	650	10000	81.25	1250.00	3600	40000
m7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i.8xlarge [2]	10000		1250.0		40000	
m7i.12xlarge [2]	15000		1875.0		60000	
m7i.16xlarge [2]	20000		2500.0		80000	
m7i.24xlarge [2]	30000		3750.0		120000	
m7i.48xlarge [2]	40000		5000.0		240000	
m7i.metal-24xl [2]	30000		3750.0		120000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m7i.metal-48xl [2]	40000		5000.0		240000	
m7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
m7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
m7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
m8a.medium [1]	325	10000	40.62	1250.00	2500	40000
m8a.large [1]	650	10000	81.25	1250.00	3600	40000
m8a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8a.8xlarge [2]	10000		1250.0		40000	
m8a.12xlarge [2]	15000		1875.0		60000	
m8a.16xlarge [2]	20000		2500.0		80000	
m8a.24xlarge [2]	30000		3750.0		120000	
m8a.48xlarge [2]	60000		7500.0		240000	
m8a.metal-24xl [2]	30000		3750.0		120000	
m8a.metal-48xl [2]	60000		7500.0		240000	
m8azn.medium [1]	625	15000	78.12	1875.00	2500	60000
m8azn.large [1]	1250	15000	156.25	1875.00	5000	60000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8azn.xlarge [1]	2500	15000	312.50	1875.00	10000	60000
m8azn.3xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m8azn.6xlarge [2]	15000		1875.0		60000	
m8azn.12xlarge [2]	30000		3750.0		120000	
m8azn.24xlarge [2]	60000		7500.0		240000	
m8azn.metal-12xl [2]	30000		3750.0		120000	
m8azn.metal-24xl [2]	60000		7500.0		240000	
m8g.medium [1]	315	10000	39.38	1250.00	2500	40000
m8g.large [1]	630	10000	78.75	1250.00	3600	40000
m8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8g.8xlarge [2]	10000		1250.0		40000	
m8g.12xlarge [2]	15000		1875.0		60000	
m8g.16xlarge [2]	20000		2500.0		80000	
m8g.24xlarge [2]	30000		3750.0		120000	
m8g.48xlarge [2]	40000		5000.0		240000	
m8g.metal-24xl [2]	30000		3750.0		120000	
m8g.metal-48xl [2]	40000		5000.0		240000	
m8gb.medium [1]	1562	25000	195.31	3125.00	7500	120000
m8gb.large [1]	3125	25000	390.62	3125.00	15000	120000
m8gb.xlarge [1]	6250	25000	781.25	3125.00	30000	120000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8gb.2xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
m8gb.4xlarge [2]	25000		3125.0		120000	
m8gb.8xlarge [2]	50000		6250.0		240000	
m8gb.12xlarge [2]	75000		9375.0		360000	
m8gb.16xlarge [2]	100000		12500.0		480000	
m8gb.24xlarge [2]	150000		18750.0		720000	
m8gb.48xlarge [2]	300000		37500.0		1440000	
m8gb.metal-24xl [2]	150000		18750.0		720000	
m8gb.metal-48xl [2]	300000		37500.0		1440000	
m8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
m8gd.large [1]	630	10000	78.75	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8gd.8xlarge [2]	10000		1250.0		40000	
m8gd.12xlarge [2]	15000		1875.0		60000	
m8gd.16xlarge [2]	20000		2500.0		80000	
m8gd.24xlarge [2]	30000		3750.0		120000	
m8gd.48xlarge [2]	40000		5000.0		240000	
m8gd.metal-24xl [2]	30000		3750.0		120000	
m8gd.metal-48xl [2]	40000		5000.0		240000	
m8gn.medium [1]	760	10000	95.00	1250.00	2500	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8gn.large [1]	1250	10000	156.25	1250.00	5000	40000
m8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
m8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8gn.4xlarge [2]	10000		1250.0		40000	
m8gn.8xlarge [2]	20000		2500.0		80000	
m8gn.12xlarge [2]	30000		3750.0		120000	
m8gn.16xlarge [2]	40000		5000.0		160000	
m8gn.24xlarge [2]	60000		7500.0		240000	
m8gn.48xlarge [2]	120000		15000.0		480000	
m8gn.metal-24xl [2]	60000		7500.0		240000	
m8gn.metal-48xl [2]	120000		15000.0		480000	
m8i.large [1]	650	10000	81.25	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8i.8xlarge [2]	10000		1250.0		40000	
m8i.12xlarge [2]	15000		1875.0		60000	
m8i.16xlarge [2]	20000		2500.0		80000	
m8i.24xlarge [2]	30000		3750.0		120000	
m8i.32xlarge [2]	40000		5000.0		160000	
m8i.48xlarge [2]	60000		7500.0		240000	
m8i.96xlarge [2]	80000		10000.0		480000	
m8i.metal-48xl [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8i.metal-96xl [2]	80000		10000.0		480000	
m8id.large [1]	650	10000	81.25	1250.00	3600	40000
m8id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8id.8xlarge [2]	10000		1250.0		40000	
m8id.12xlarge [2]	15000		1875.0		60000	
m8id.16xlarge [2]	20000		2500.0		80000	
m8id.24xlarge [2]	30000		3750.0		120000	
m8id.32xlarge [2]	40000		5000.0		160000	
m8id.48xlarge [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8id.96xlarge [2]	80000		10000.0		480000	
m8id.metal-48xl [2]	60000		7500.0		240000	
m8id.metal-96xl [2]	80000		10000.0		480000	
m8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
m8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
m8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
m8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
m8in.large [1]	650	10000	81.25	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8in.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8in.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8in.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8in.8xlarge [2]	10000		1250.0		40000	
m8in.12xlarge [2]	15000		1875.0		60000	
m8in.16xlarge [2]	20000		2500.0		80000	
m8in.24xlarge [2]	30000		3750.0		120000	
m8in.32xlarge [2]	40000		5000.0		160000	
m8in.48xlarge [2]	60000		7500.0		240000	
m8in.96xlarge [2]	120000		15000.0		480000	
m8in.metal-48xl [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8in.metal-96xl [2]	120000		15000.0		480000	
m8idn.large [1]	650	10000	81.25	1250.00	3600	40000
m8idn.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8idn.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8idn.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8idn.8xlarge [2]	10000		1250.0		40000	
m8idn.12xlarge [2]	15000		1875.0		60000	
m8idn.16xlarge [2]	20000		2500.0		80000	
m8idn.24xlarge [2]	30000		3750.0		120000	
m8idn.32xlarge [2]	40000		5000.0		160000	
m8idn.48xlarge [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8idn.96xlarge [2]	120000		15000.0		480000	
m8idn.metal-48xl [2]	60000		7500.0		240000	
m8idn.metal-96xl [2]	120000		15000.0		480000	
m8ine.large [1]	650	10000	81.25	1250.00	3600	40000
m8ine.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
m8ine.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
m8ine.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
m8ine.8xlarge [2]	10000		1250.0		40000	
m8ine.12xlarge [2]	15000		1875.0		60000	
m8ib.large [1]	1563	25000	195.38	3125.00	7500	120000
m8ib.xlarge [1]	3125	25000	390.62	3125.00	15000	120000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8ib.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
m8ib.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
m8ib.8xlarge [2]	25000		3125.0		120000	
m8ib.12xlarge [2]	37500		4687.5		180000	
m8ib.16xlarge [2]	50000		6250.0		240000	
m8ib.24xlarge [2]	75000		9375.0		360000	
m8ib.32xlarge [2]	100000		12500.0		480000	
m8ib.48xlarge [2]	150000		18750.0		720000	
m8ib.96xlarge [2]	300000		37500.0		1440000	
m8ib.metal-48xl [2]	150000		18750.0		720000	
m8ib.metal-96xl [2]	300000		37500.0		1440000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8idb.large [1]	1563	25000	195.38	3125.00	7500	120000
m8idb.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
m8idb.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
m8idb.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
m8idb.8xlarge [2]	25000		3125.0		120000	
m8idb.12xlarge [2]	37500		4687.5		180000	
m8idb.16xlarge [2]	50000		6250.0		240000	
m8idb.24xlarge [2]	75000		9375.0		360000	
m8idb.32xlarge [2]	100000		12500.0		480000	
m8idb.48xlarge [2]	150000		18750.0		720000	
m8idb.96xlarge [2]	300000		37500.0		1440000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m8idb.metal-48xl [2]	150000		18750.0		720000	
m8idb.metal-96xl [2]	300000		37500.0		1440000	
m9g.medium [1]	380	12000	47.50	1500.00	2500	48000
m9g.large [1]	760	12000	95.00	1500.00	3600	48000
m9g.xlarge [1]	1500	12000	187.50	1500.00	6000	48000
m9g.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
m9g.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
m9g.8xlarge [2]	12000		1500.0		48000	
m9g.12xlarge [2]	18000		2250.0		72000	
m9g.16xlarge [2]	24000		3000.0		96000	
m9g.24xlarge [2]	36000		4500.0		144000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m9g.48xlarge [2]	72000		9000.0		288000	
m9g.metal-48xl [2]	72000		9000.0		288000	
m9gd.medium [1]	380	12000	47.50	1500.00	2500	48000
m9gd.large [1]	760	12000	95.00	1500.00	3600	48000
m9gd.xlarge [1]	1500	12000	187.50	1500.00	6000	48000
m9gd.2xlarge [1]	3000	12000	375.00	1500.00	12000	48000
m9gd.4xlarge [1]	6000	12000	750.00	1500.00	24000	48000
m9gd.8xlarge [2]	12000		1500.0		48000	
m9gd.12xlarge [2]	18000		2250.0		72000	
m9gd.16xlarge [2]	24000		3000.0		96000	
m9gd.24xlarge [2]	36000		4500.0		144000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
m9gd.48xlarge [2]	72000		9000.0		288000	
m9gd.metal-48xl [2]	72000		9000.0		288000	
mac1.metal [2]	14000		1750.0		80000	
mac2.metal [2]	10000		1250.0		55000	
mac2-m1ultra.metal [2]	10000		1250.0		55000	
mac2-m2.metal [2]	8000		1000.0		55000	
mac2-m2pro.metal [2]	8000		1000.0		55000	
mac-m4.metal [2]	8000		1000.0		55000	
mac-m4pro.metal [2]	8000		1000.0		55000	
mac-m4max.metal [2]	10000		1250.0		55000	
t3.nano [1]	43	2085	5.38	260.62	250	11800
t3.micro [1]	87	2085	10.88	260.62	500	11800

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
t3.small [1]	174	2085	21.75	260.62	1000	11800
t3.medium [1]	347	2085	43.38	260.62	2000	11800
t3.large [1]	695	2780	86.88	347.50	4000	15700
t3.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.nano [1]	45	2085	5.62	260.62	250	11800
t3a.micro [1]	90	2085	11.25	260.62	500	11800
t3a.small [1]	175	2085	21.88	260.62	1000	11800
t3a.medium [1]	350	2085	43.75	260.62	2000	11800
t3a.large [1]	695	2780	86.88	347.50	4000	15700
t3a.xlarge [1]	695	2780	86.88	347.50	4000	15700
t3a.2xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.nano [1]	43	2085	5.38	260.62	250	11800
t4g.micro [1]	87	2085	10.88	260.62	500	11800
t4g.small [1]	174	2085	21.75	260.62	1000	11800
t4g.medium [1]	347	2085	43.38	260.62	2000	11800

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
t4g.large [1]	695	2780	86.88	347.50	4000	15700
t4g.xlarge [1]	695	2780	86.88	347.50	4000	15700
t4g.2xlarge [1]	695	2780	86.88	347.50	4000	15700

[1] 这些实例可以至少每 24 小时维持一次 30 分钟的最大性能，之后会恢复到基线性能。

[2] 这些实例可以无限期地维持其既定性能。如果工作负载需要在超过 30 分钟的时长内保持最大性能，请选择以下任一实例。

计算优化型

> **Note**
>
> C8a、C8g、C8gd、C8i、C8id、C8i-flex 实例类型支持可配置的带宽权重。使用这些实例类型，您可以针对网络性能或 Amazon EBS 性能优化实例的带宽。下表显示了这些实例类型的默认 Amazon EBS 带宽性能。有关更多信息，请参阅 EC2 实例带宽权重配置。

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c4.large [2]	500		62.5		4000	
c4.xlarge [2]	750		93.75		6000	
c4.2xlarge [2]	1000		125.0		8000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c4.4xlarge [2]	2000		250.0		16000	
c4.8xlarge [2]	4000		500.0		32000	
c5.large [1]	650	4750	81.25	593.75	4000	20000
c5.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5.4xlarge [2]	4750		593.75		20000	
c5.9xlarge [2]	9500		1187.5		40000	
c5.12xlarge [2]	9500		1187.5		40000	
c5.18xlarge [2]	19000		2375.0		80000	
c5.24xlarge [2]	19000		2375.0		80000	
c5.metal [2]	19000		2375.0		80000	
c5a.large [1]	200	3170	25.00	396.25	800	13300
c5a.xlarge [1]	400	3170	50.00	396.25	1600	13300

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c5a.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5a.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5a.8xlarge [2]	3170		396.25		13300	
c5a.12xlarge [2]	4750		593.75		20000	
c5a.16xlarge [2]	6300		787.5		26700	
c5a.24xlarge [2]	9500		1187.5		40000	
c5ad.large [1]	200	3170	25.00	396.25	800	13300
c5ad.xlarge [1]	400	3170	50.00	396.25	1600	13300
c5ad.2xlarge [1]	800	3170	100.00	396.25	3200	13300
c5ad.4xlarge [1]	1580	3170	197.50	396.25	6600	13300
c5ad.8xlarge [2]	3170		396.25		13300	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c5ad.12xlarge [2]	4750		593.75		20000	
c5ad.16xlarge [2]	6300		787.5		26700	
c5ad.24xlarge [2]	9500		1187.5		40000	
c5d.large [1]	650	4750	81.25	593.75	4000	20000
c5d.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5d.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5d.4xlarge [2]	4750		593.75		20000	
c5d.9xlarge [2]	9500		1187.5		40000	
c5d.12xlarge [2]	9500		1187.5		40000	
c5d.18xlarge [2]	19000		2375.0		80000	
c5d.24xlarge [2]	19000		2375.0		80000	
c5d.metal [2]	19000		2375.0		80000	
c5n.large [1]	650	4750	81.25	593.75	4000	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c5n.xlarge [1]	1150	4750	143.75	593.75	6000	20000
c5n.2xlarge [1]	2300	4750	287.50	593.75	10000	20000
c5n.4xlarge [2]	4750		593.75		20000	
c5n.9xlarge [2]	9500		1187.5		40000	
c5n.18xlarge [2]	19000		2375.0		80000	
c5n.metal [2]	19000		2375.0		80000	
c6a.large [1]	650	10000	81.25	1250.00	3600	40000
c6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6a.8xlarge [2]	10000		1250.0		40000	
c6a.12xlarge [2]	15000		1875.0		60000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c6a.16xlarge [2]	20000		2500.0		80000	
c6a.24xlarge [2]	30000		3750.0		120000	
c6a.32xlarge [2]	40000		5000.0		160000	
c6a.48xlarge [2]	40000		5000.0		240000	
c6a.metal [2]	40000		5000.0		240000	
c6g.medium [1]	315	4750	39.38	593.75	2500	20000
c6g.large [1]	630	4750	78.75	593.75	3600	20000
c6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
c6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6g.4xlarge [2]	4750		593.75		20000	
c6g.8xlarge [2]	9500		1187.5		40000	
c6g.12xlarge [2]	14250		1781.25		50000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c6g.16xlarge [2]	19000		2375.0		80000	
c6g.metal [2]	19000		2375.0		80000	
c6gd.medium [1]	315	4750	39.38	593.75	2500	20000
c6gd.large [1]	630	4750	78.75	593.75	3600	20000
c6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
c6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
c6gd.4xlarge [2]	4750		593.75		20000	
c6gd.8xlarge [2]	9500		1187.5		40000	
c6gd.12xlarge [2]	14250		1781.25		50000	
c6gd.16xlarge [2]	19000		2375.0		80000	
c6gd.metal [2]	19000		2375.0		80000	
c6gn.medium [1]	760	9500	95.00	1187.50	2500	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c6gn.large [1]	1235	9500	154.38	1187.50	5000	40000
c6gn.xlarge [1]	2375	9500	296.88	1187.50	10000	40000
c6gn.2xlarge [1]	4750	9500	593.75	1187.50	20000	40000
c6gn.4xlarge [2]	9500		1187.5		40000	
c6gn.8xlarge [2]	19000		2375.0		80000	
c6gn.12xlarge [2]	28500		3562.5		120000	
c6gn.16xlarge [2]	38000		4750.0		160000	
c6i.large [1]	650	10000	81.25	1250.00	3600	40000
c6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6i.8xlarge [2]	10000		1250.0		40000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c6i.12xlarge [2]	15000		1875.0		60000	
c6i.16xlarge [2]	20000		2500.0		80000	
c6i.24xlarge [2]	30000		3750.0		120000	
c6i.32xlarge [2]	40000		5000.0		160000	
c6i.metal [2]	40000		5000.0		160000	
c6id.large [1]	650	10000	81.25	1250.00	3600	40000
c6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c6id.8xlarge [2]	10000		1250.0		40000	
c6id.12xlarge [2]	15000		1875.0		60000	
c6id.16xlarge [2]	20000		2500.0		80000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c6id.24xlarge [2]	30000		3750.0		120000	
c6id.32xlarge [2]	40000		5000.0		160000	
c6id.metal [2]	40000		5000.0		160000	
c6in.large [1]	1562	25000	195.31	3125.00	6250	100000
c6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
c6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
c6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
c6in.8xlarge [2]	25000		3125.0		100000	
c6in.12xlarge [2]	37500		4687.5		150000	
c6in.16xlarge [2]	50000		6250.0		200000	
c6in.24xlarge [2]	75000		9375.0		300000	
c6in.32xlarge [2]	100000		12500.0		400000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c6in.metal [2]	100000		12500.0		400000	
c7a.medium [1]	325	10000	40.62	1250.00	2500	40000
c7a.large [1]	650	10000	81.25	1250.00	3600	40000
c7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7a.8xlarge [2]	10000		1250.0		40000	
c7a.12xlarge [2]	15000		1875.0		60000	
c7a.16xlarge [2]	20000		2500.0		80000	
c7a.24xlarge [2]	30000		3750.0		120000	
c7a.32xlarge [2]	40000		5000.0		160000	
c7a.48xlarge [2]	40000		5000.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c7a.metal-48xl [2]	40000		5000.0		240000	
c7g.medium [1]	315	10000	39.38	1250.00	2500	40000
c7g.large [1]	630	10000	78.75	1250.00	3600	40000
c7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7g.8xlarge [2]	10000		1250.0		40000	
c7g.12xlarge [2]	15000		1875.0		60000	
c7g.16xlarge [2]	20000		2500.0		80000	
c7g.metal [2]	20000		2500.0		80000	
c7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c7gd.large [1]	630	10000	78.75	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7gd.8xlarge [2]	10000		1250.0		40000	
c7gd.12xlarge [2]	15000		1875.0		60000	
c7gd.16xlarge [2]	20000		2500.0		80000	
c7gd.metal [2]	20000		2500.0		80000	
c7gn.medium [1]	521	10000	65.12	1250.00	2083	40000
c7gn.large [1]	1042	10000	130.25	1250.00	4167	40000
c7gn.xlarge [1]	2083	10000	260.38	1250.00	8333	40000
c7gn.2xlarge [1]	4167	10000	520.88	1250.00	16667	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c7gn.4xlarge [1]	8333	10000	1041.62	1250.00	33333	40000
c7gn.8xlarge [1]	16667	20000	2083.38	2500.00	66667	80000
c7gn.12xlarge [1]	25000	30000	3125.00	3750.00	100000	120000
c7gn.16xlarge [1]	33333	40000	4166.62	5000.00	133333	160000
c7gn.metal [1]	33333	40000	4166.62	5000.00	133333	160000
c7i.large [1]	650	10000	81.25	1250.00	3600	40000
c7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i.8xlarge [2]	10000		1250.0		40000	
c7i.12xlarge [2]	15000		1875.0		60000	
c7i.16xlarge [2]	20000		2500.0		80000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c7i.24xlarge [2]	30000		3750.0		120000	
c7i.48xlarge [2]	40000		5000.0		240000	
c7i.metal-24xl [2]	30000		3750.0		120000	
c7i.metal-48xl [2]	40000		5000.0		240000	
c7i-flex.large [1]	312	10000	39.06	1250.00	2500	40000
c7i-flex.xlarge [1]	625	10000	78.12	1250.00	3600	40000
c7i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c7i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c7i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c7i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
c7i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8a.medium [1]	325	10000	40.62	1250.00	2500	40000
c8a.large [1]	650	10000	81.25	1250.00	3600	40000
c8a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8a.8xlarge [2]	10000		1250.0		40000	
c8a.12xlarge [2]	15000		1875.0		60000	
c8a.16xlarge [2]	20000		2500.0		80000	
c8a.24xlarge [2]	30000		3750.0		120000	
c8a.48xlarge [2]	60000		7500.0		240000	
c8a.metal-24xl [2]	30000		3750.0		120000	
c8a.metal-48xl [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8g.medium [1]	315	10000	39.38	1250.00	2500	40000
c8g.large [1]	630	10000	78.75	1250.00	3600	40000
c8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8g.8xlarge [2]	10000		1250.0		40000	
c8g.12xlarge [2]	15000		1875.0		60000	
c8g.16xlarge [2]	20000		2500.0		80000	
c8g.24xlarge [2]	30000		3750.0		120000	
c8g.48xlarge [2]	40000		5000.0		240000	
c8g.metal-24xl [2]	30000		3750.0		120000	
c8g.metal-48xl [2]	40000		5000.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8gb.medium [1]	1562	25000	195.31	3125.00	7500	120000
c8gb.large [1]	3125	25000	390.62	3125.00	15000	120000
c8gb.xlarge [1]	6250	25000	781.25	3125.00	30000	120000
c8gb.2xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
c8gb.4xlarge [2]	25000		3125.0		120000	
c8gb.8xlarge [2]	50000		6250.0		240000	
c8gb.12xlarge [2]	75000		9375.0		360000	
c8gb.16xlarge [2]	100000		12500.0		480000	
c8gb.24xlarge [2]	150000		18750.0		720000	
c8gb.48xlarge [2]	300000		37500.0		1440000	
c8gb.metal-24xl [2]	150000		18750.0		720000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8gb.metal-48xl [2]	300000		37500.0		1440000	
c8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
c8gd.large [1]	630	10000	78.75	1250.00	3600	40000
c8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8gd.8xlarge [2]	10000		1250.0		40000	
c8gd.12xlarge [2]	15000		1875.0		60000	
c8gd.16xlarge [2]	20000		2500.0		80000	
c8gd.24xlarge [2]	30000		3750.0		120000	
c8gd.48xlarge [2]	40000		5000.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8gd.metal-24xl [2]	30000		3750.0		120000	
c8gd.metal-48xl [2]	40000		5000.0		240000	
c8gn.medium [1]	760	10000	95.00	1250.00	2500	40000
c8gn.large [1]	1250	10000	156.25	1250.00	5000	40000
c8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
c8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8gn.4xlarge [2]	10000		1250.0		40000	
c8gn.8xlarge [2]	20000		2500.0		80000	
c8gn.12xlarge [2]	30000		3750.0		120000	
c8gn.16xlarge [2]	40000		5000.0		160000	
c8gn.24xlarge [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8gn.48xlarge [2]	120000		15000.0		480000	
c8gn.metal-24xl [2]	60000		7500.0		240000	
c8gn.metal-48xl [2]	120000		15000.0		480000	
c8i.large [1]	650	10000	81.25	1250.00	3600	40000
c8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8i.8xlarge [2]	10000		1250.0		40000	
c8i.12xlarge [2]	15000		1875.0		60000	
c8i.16xlarge [2]	20000		2500.0		80000	
c8i.24xlarge [2]	30000		3750.0		120000	
c8i.32xlarge [2]	40000		5000.0		160000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8i.48xlarge [2]	60000		7500.0		240000	
c8i.96xlarge [2]	80000		10000.0		480000	
c8i.metal-48xl [2]	60000		7500.0		240000	
c8i.metal-96xl [2]	80000		10000.0		480000	
c8id.large [1]	650	10000	81.25	1250.00	3600	40000
c8id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8id.8xlarge [2]	10000		1250.0		40000	
c8id.12xlarge [2]	15000		1875.0		60000	
c8id.16xlarge [2]	20000		2500.0		80000	
c8id.24xlarge [2]	30000		3750.0		120000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8id.32xlarge [2]	40000		5000.0		160000	
c8id.48xlarge [2]	60000		7500.0		240000	
c8id.96xlarge [2]	80000		10000.0		480000	
c8id.metal-48xl [2]	60000		7500.0		240000	
c8id.metal-96xl [2]	80000		10000.0		480000	
c8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
c8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
c8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
c8in.large [1]	650	10000	81.25	1250.00	3600	40000
c8in.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8in.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8in.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8in.8xlarge [2]	10000		1250.0		40000	
c8in.12xlarge [2]	15000		1875.0		60000	
c8in.16xlarge [2]	20000		2500.0		80000	
c8in.24xlarge [2]	30000		3750.0		120000	
c8in.32xlarge [2]	40000		5000.0		160000	
c8in.48xlarge [2]	60000		7500.0		240000	
c8in.96xlarge [2]	120000		15000.0		480000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8in.metal-48xl [2]	60000		7500.0		240000	
c8in.metal-96xl [2]	120000		15000.0		480000	
c8ine.large [1]	650	10000	81.25	1250.00	3600	40000
c8ine.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
c8ine.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
c8ine.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
c8ine.8xlarge [2]	10000		1250.0		40000	
c8ine.12xlarge [2]	15000		1875.0		60000	
c8ib.large [1]	1563	25000	195.38	3125.00	7500	120000
c8ib.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
c8ib.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
c8ib.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
c8ib.8xlarge [2]	25000		3125.0		120000	
c8ib.12xlarge [2]	37500		4687.5		180000	
c8ib.16xlarge [2]	50000		6250.0		240000	
c8ib.24xlarge [2]	75000		9375.0		360000	
c8ib.32xlarge [2]	100000		12500.0		480000	
c8ib.48xlarge [2]	150000		18750.0		720000	
c8ib.96xlarge [2]	300000		37500.0		1440000	
c8ib.metal-48xl [2]	150000		18750.0		720000	
c8ib.metal-96xl [2]	300000		37500.0		1440000	

[1] 这些实例可以至少每 24 小时维持一次 30 分钟的最大性能，之后会恢复到基线性能。

[2] 这些实例可以无限期地维持其既定性能。如果工作负载需要在超过 30 分钟的时长内保持最大性能，请选择以下任一实例。

内存优化型

> **Note**
>
> - R8a、R8g、R8gd、R8i、R8id、R8i-flex、X8g、X8aedz、X8i 实例类型支持可配置的带宽权重。使用这些实例类型，您可以针对网络性能或 Amazon EBS 性能优化实例的带宽。下表显示了这些实例类型的默认 Amazon EBS 带宽性能。有关更多信息，请参阅 EC2 实例带宽权重配置。
> - 为了实现 U7i 实例的最佳 IOPS 性能，我们建议使用 io2 BlockExpress 卷。

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r4.large [2]	425		53.125		3000	
r4.xlarge [2]	850		106.25		6000	
r4.2xlarge [2]	1700		212.5		12000	
r4.4xlarge [2]	3500		437.5		18750	
r4.8xlarge [2]	7000		875.0		37500	
r4.16xlarge [2]	14000		1750.0		75000	
r5.large [1]	650	4750	81.25	593.75	3600	18750
r5.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5.2xlarge [1]	2300	4750	287.50	593.75	12000	18750

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r5.4xlarge [2]	4750		593.75		18750	
r5.8xlarge [2]	6800		850.0		30000	
r5.12xlarge [2]	9500		1187.5		40000	
r5.16xlarge [2]	13600		1700.0		60000	
r5.24xlarge [2]	19000		2375.0		80000	
r5.metal [2]	19000		2375.0		80000	
r5a.large [1]	650	2880	81.25	360.00	3600	16000
r5a.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5a.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5a.4xlarge [2]	2880		360.0		16000	
r5a.8xlarge [2]	4750		593.75		20000	
r5a.12xlarge [2]	6780		847.5		30000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r5a.16xlarge [2]	9500		1187.5		40000	
r5a.24xlarge [2]	13570		1696.25		60000	
r5ad.large [1]	650	2880	81.25	360.00	3600	16000
r5ad.xlarge [1]	1085	2880	135.62	360.00	6000	16000
r5ad.2xlarge [1]	1580	2880	197.50	360.00	8333	16000
r5ad.4xlarge [2]	2880		360.0		16000	
r5ad.8xlarge [2]	4750		593.75		20000	
r5ad.12xlarge [2]	6780		847.5		30000	
r5ad.16xlarge [2]	9500		1187.5		40000	
r5ad.24xlarge [2]	13570		1696.25		60000	
r5b.large [1]	1250	10000	156.25	1250.00	5417	43333
r5b.xlarge [1]	2500	10000	312.50	1250.00	10833	43333

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r5b.2xlarge [1]	5000	10000	625.00	1250.00	21667	43333
r5b.4xlarge [2]	10000		1250.0		43333	
r5b.8xlarge [2]	20000		2500.0		86667	
r5b.12xlarge [2]	30000		3750.0		130000	
r5b.16xlarge [2]	40000		5000.0		173333	
r5b.24xlarge [2]	60000		7500.0		260000	
r5b.metal [2]	60000		7500.0		260000	
r5d.large [1]	650	4750	81.25	593.75	3600	18750
r5d.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5d.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5d.4xlarge [2]	4750		593.75		18750	
r5d.8xlarge [2]	6800		850.0		30000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r5d.12xlarge [2]	9500		1187.5		40000	
r5d.16xlarge [2]	13600		1700.0		60000	
r5d.24xlarge [2]	19000		2375.0		80000	
r5d.metal [2]	19000		2375.0		80000	
r5dn.large [1]	650	4750	81.25	593.75	3600	18750
r5dn.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5dn.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5dn.4xlarge [2]	4750		593.75		18750	
r5dn.8xlarge [2]	6800		850.0		30000	
r5dn.12xlarge [2]	9500		1187.5		40000	
r5dn.16xlarge [2]	13600		1700.0		60000	
r5dn.24xlarge [2]	19000		2375.0		80000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r5dn.metal [2]	19000		2375.0		80000	
r5n.large [1]	650	4750	81.25	593.75	3600	18750
r5n.xlarge [1]	1150	4750	143.75	593.75	6000	18750
r5n.2xlarge [1]	2300	4750	287.50	593.75	12000	18750
r5n.4xlarge [2]	4750		593.75		18750	
r5n.8xlarge [2]	6800		850.0		30000	
r5n.12xlarge [2]	9500		1187.5		40000	
r5n.16xlarge [2]	13600		1700.0		60000	
r5n.24xlarge [2]	19000		2375.0		80000	
r5n.metal [2]	19000		2375.0		80000	
r6a.large [1]	650	10000	81.25	1250.00	3600	40000
r6a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r6a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6a.8xlarge [2]	10000		1250.0		40000	
r6a.12xlarge [2]	15000		1875.0		60000	
r6a.16xlarge [2]	20000		2500.0		80000	
r6a.24xlarge [2]	30000		3750.0		120000	
r6a.32xlarge [2]	40000		5000.0		160000	
r6a.48xlarge [2]	40000		5000.0		240000	
r6a.metal [2]	40000		5000.0		240000	
r6g.medium [1]	315	4750	39.38	593.75	2500	20000
r6g.large [1]	630	4750	78.75	593.75	3600	20000
r6g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r6g.4xlarge [2]	4750		593.75		20000	
r6g.8xlarge [2]	9500		1187.5		40000	
r6g.12xlarge [2]	14250		1781.25		50000	
r6g.16xlarge [2]	19000		2375.0		80000	
r6g.metal [2]	19000		2375.0		80000	
r6gd.medium [1]	315	4750	39.38	593.75	2500	20000
r6gd.large [1]	630	4750	78.75	593.75	3600	20000
r6gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
r6gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
r6gd.4xlarge [2]	4750		593.75		20000	
r6gd.8xlarge [2]	9500		1187.5		40000	
r6gd.12xlarge [2]	14250		1781.25		50000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r6gd.16xlarge [2]	19000		2375.0		80000	
r6gd.metal [2]	19000		2375.0		80000	
r6i.large [1]	650	10000	81.25	1250.00	3600	40000
r6i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6i.8xlarge [2]	10000		1250.0		40000	
r6i.12xlarge [2]	15000		1875.0		60000	
r6i.16xlarge [2]	20000		2500.0		80000	
r6i.24xlarge [2]	30000		3750.0		120000	
r6i.32xlarge [2]	40000		5000.0		160000	
r6i.metal [2]	40000		5000.0		160000	
r6id.large [1]	650	10000	81.25	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r6id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r6id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r6id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r6id.8xlarge [2]	10000		1250.0		40000	
r6id.12xlarge [2]	15000		1875.0		60000	
r6id.16xlarge [2]	20000		2500.0		80000	
r6id.24xlarge [2]	30000		3750.0		120000	
r6id.32xlarge [2]	40000		5000.0		160000	
r6id.metal [2]	40000		5000.0		160000	
r6idn.large [1]	1562	25000	195.31	3125.00	6250	100000
r6idn.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6idn.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r6idn.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6idn.8xlarge [2]	25000		3125.0		100000	
r6idn.12xlarge [2]	37500		4687.5		150000	
r6idn.16xlarge [2]	50000		6250.0		200000	
r6idn.24xlarge [2]	75000		9375.0		300000	
r6idn.32xlarge [2]	100000		12500.0		400000	
r6idn.metal [2]	100000		12500.0		400000	
r6in.large [1]	1562	25000	195.31	3125.00	6250	100000
r6in.xlarge [1]	3125	25000	390.62	3125.00	12500	100000
r6in.2xlarge [1]	6250	25000	781.25	3125.00	25000	100000
r6in.4xlarge [1]	12500	25000	1562.50	3125.00	50000	100000
r6in.8xlarge [2]	25000		3125.0		100000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r6in.12xlarge [2]	37500		4687.5		150000	
r6in.16xlarge [2]	50000		6250.0		200000	
r6in.24xlarge [2]	75000		9375.0		300000	
r6in.32xlarge [2]	100000		12500.0		400000	
r6in.metal [2]	100000		12500.0		400000	
r7a.medium [1]	325	10000	40.62	1250.00	2500	40000
r7a.large [1]	650	10000	81.25	1250.00	3600	40000
r7a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7a.8xlarge [2]	10000		1250.0		40000	
r7a.12xlarge [2]	15000		1875.0		60000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r7a.16xlarge [2]	20000		2500.0		80000	
r7a.24xlarge [2]	30000		3750.0		120000	
r7a.32xlarge [2]	40000		5000.0		160000	
r7a.48xlarge [2]	40000		5000.0		240000	
r7a.metal-48xl [2]	40000		5000.0		240000	
r7g.medium [1]	315	10000	39.38	1250.00	2500	40000
r7g.large [1]	630	10000	78.75	1250.00	3600	40000
r7g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7g.8xlarge [2]	10000		1250.0		40000	
r7g.12xlarge [2]	15000		1875.0		60000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r7g.16xlarge [2]	20000		2500.0		80000	
r7g.metal [2]	20000		2500.0		80000	
r7gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r7gd.large [1]	630	10000	78.75	1250.00	3600	40000
r7gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r7gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7gd.8xlarge [2]	10000		1250.0		40000	
r7gd.12xlarge [2]	15000		1875.0		60000	
r7gd.16xlarge [2]	20000		2500.0		80000	
r7gd.metal [2]	20000		2500.0		80000	
r7i.large [1]	650	10000	81.25	1250.00	3600	40000
r7i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r7i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7i.8xlarge [2]	10000		1250.0		40000	
r7i.12xlarge [2]	15000		1875.0		60000	
r7i.16xlarge [2]	20000		2500.0		80000	
r7i.24xlarge [2]	30000		3750.0		120000	
r7i.48xlarge [2]	40000		5000.0		240000	
r7i.metal-24xl [2]	30000		3750.0		120000	
r7i.metal-48xl [2]	40000		5000.0		240000	
r7iz.large [1]	792	10000	99.00	1250.00	3600	40000
r7iz.xlarge [1]	1584	10000	198.00	1250.00	6667	40000
r7iz.2xlarge [1]	3168	10000	396.00	1250.00	13333	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r7iz.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r7iz.8xlarge [2]	10000		1250.0		40000	
r7iz.12xlarge [2]	19000		2375.0		76000	
r7iz.16xlarge [2]	20000		2500.0		80000	
r7iz.32xlarge [2]	40000		5000.0		160000	
r7iz.metal-16xl [2]	20000		2500.0		80000	
r7iz.metal-32xl [2]	40000		5000.0		160000	
r8a.medium [1]	325	10000	40.62	1250.00	2500	40000
r8a.large [1]	650	10000	81.25	1250.00	3600	40000
r8a.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8a.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8a.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8a.8xlarge [2]	10000		1250.0		40000	
r8a.12xlarge [2]	15000		1875.0		60000	
r8a.16xlarge [2]	20000		2500.0		80000	
r8a.24xlarge [2]	30000		3750.0		120000	
r8a.48xlarge [2]	60000		7500.0		240000	
r8a.metal-24xl [2]	30000		3750.0		120000	
r8a.metal-48xl [2]	60000		7500.0		240000	
r8g.medium [1]	315	10000	39.38	1250.00	2500	40000
r8g.large [1]	630	10000	78.75	1250.00	3600	40000
r8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8g.8xlarge [2]	10000		1250.0		40000	
r8g.12xlarge [2]	15000		1875.0		60000	
r8g.16xlarge [2]	20000		2500.0		80000	
r8g.24xlarge [2]	30000		3750.0		120000	
r8g.48xlarge [2]	40000		5000.0		240000	
r8g.metal-24xl [2]	30000		3750.0		120000	
r8g.metal-48xl [2]	40000		5000.0		240000	
r8gb.medium [1]	1562	25000	195.31	3125.00	7500	120000
r8gb.large [1]	3125	25000	390.62	3125.00	15000	120000
r8gb.xlarge [1]	6250	25000	781.25	3125.00	30000	120000
r8gb.2xlarge [1]	12500	25000	1562.50	3125.00	60000	120000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8gb.4xlarge [2]	25000		3125.0		120000	
r8gb.8xlarge [2]	50000		6250.0		240000	
r8gb.12xlarge [2]	75000		9375.0		360000	
r8gb.16xlarge [2]	100000		12500.0		480000	
r8gb.24xlarge [2]	150000		18750.0		720000	
r8gb.48xlarge [2]	300000		37500.0		1440000	
r8gb.metal-24xl [2]	150000		18750.0		720000	
r8gb.metal-48xl [2]	300000		37500.0		1440000	
r8gd.medium [1]	315	10000	39.38	1250.00	2500	40000
r8gd.large [1]	630	10000	78.75	1250.00	3600	40000
r8gd.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8gd.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8gd.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8gd.8xlarge [2]	10000		1250.0		40000	
r8gd.12xlarge [2]	15000		1875.0		60000	
r8gd.16xlarge [2]	20000		2500.0		80000	
r8gd.24xlarge [2]	30000		3750.0		120000	
r8gd.48xlarge [2]	40000		5000.0		240000	
r8gd.metal-24xl [2]	30000		3750.0		120000	
r8gd.metal-48xl [2]	40000		5000.0		240000	
r8gn.medium [1]	760	10000	95.00	1250.00	2500	40000
r8gn.large [1]	1250	10000	156.25	1250.00	5000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
r8gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8gn.4xlarge [2]	10000		1250.0		40000	
r8gn.8xlarge [2]	20000		2500.0		80000	
r8gn.12xlarge [2]	30000		3750.0		120000	
r8gn.16xlarge [2]	40000		5000.0		160000	
r8gn.24xlarge [2]	60000		7500.0		240000	
r8gn.48xlarge [2]	120000		15000.0		480000	
r8gn.metal-24xl [2]	60000		7500.0		240000	
r8gn.metal-48xl [2]	120000		15000.0		480000	
r8i.large [1]	650	10000	81.25	1250.00	3600	40000
r8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8i.8xlarge [2]	10000		1250.0		40000	
r8i.12xlarge [2]	15000		1875.0		60000	
r8i.16xlarge [2]	20000		2500.0		80000	
r8i.24xlarge [2]	30000		3750.0		120000	
r8i.32xlarge [2]	40000		5000.0		160000	
r8i.48xlarge [2]	60000		7500.0		240000	
r8i.96xlarge [2]	80000		10000.0		480000	
r8i.metal-48xl [2]	60000		7500.0		240000	
r8i.metal-96xl [2]	80000		10000.0		480000	
r8id.large [1]	650	10000	81.25	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8id.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8id.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8id.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8id.8xlarge [2]	10000		1250.0		40000	
r8id.12xlarge [2]	15000		1875.0		60000	
r8id.16xlarge [2]	20000		2500.0		80000	
r8id.24xlarge [2]	30000		3750.0		120000	
r8id.32xlarge [2]	40000		5000.0		160000	
r8id.48xlarge [2]	60000		7500.0		240000	
r8id.96xlarge [2]	80000		10000.0		480000	
r8id.metal-48xl [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8id.metal-96xl [2]	80000		10000.0		480000	
r8i-flex.large [1]	315	10000	39.38	1250.00	2500	40000
r8i-flex.xlarge [1]	630	10000	78.75	1250.00	3600	40000
r8i-flex.2xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8i-flex.4xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8i-flex.8xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8i-flex.12xlarge [1]	7500	15000	937.50	1875.00	30000	60000
r8i-flex.16xlarge [1]	10000	20000	1250.00	2500.00	40000	80000
r8in.large [1]	650	10000	81.25	1250.00	3600	40000
r8in.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
r8in.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8in.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8in.8xlarge [2]	10000		1250.0		40000	
r8in.12xlarge [2]	15000		1875.0		60000	
r8in.16xlarge [2]	20000		2500.0		80000	
r8in.24xlarge [2]	30000		3750.0		120000	
r8in.32xlarge [2]	40000		5000.0		160000	
r8in.48xlarge [2]	60000		7500.0		240000	
r8in.96xlarge [2]	120000		15000.0		480000	
r8in.metal-48xl [2]	60000		7500.0		240000	
r8in.metal-96xl [2]	120000		15000.0		480000	
r8idn.large [1]	650	10000	81.25	1250.00	3600	40000
r8idn.xlarge [1]	1250	10000	156.25	1250.00	6000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8idn.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
r8idn.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
r8idn.8xlarge [2]	10000		1250.0		40000	
r8idn.12xlarge [2]	15000		1875.0		60000	
r8idn.16xlarge [2]	20000		2500.0		80000	
r8idn.24xlarge [2]	30000		3750.0		120000	
r8idn.32xlarge [2]	40000		5000.0		160000	
r8idn.48xlarge [2]	60000		7500.0		240000	
r8idn.96xlarge [2]	120000		15000.0		480000	
r8idn.metal-48xl [2]	60000		7500.0		240000	
r8idn.metal-96xl [2]	120000		15000.0		480000	
r8ib.large [1]	1563	25000	195.38	3125.00	7500	120000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8ib.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
r8ib.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
r8ib.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
r8ib.8xlarge [2]	25000		3125.0		120000	
r8ib.12xlarge [2]	37500		4687.5		180000	
r8ib.16xlarge [2]	50000		6250.0		240000	
r8ib.24xlarge [2]	75000		9375.0		360000	
r8ib.32xlarge [2]	100000		12500.0		480000	
r8ib.48xlarge [2]	150000		18750.0		720000	
r8ib.96xlarge [2]	300000		37500.0		1440000	
r8ib.metal-48xl [2]	150000		18750.0		720000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8ib.metal-96xl [2]	300000		37500.0		1440000	
r8idb.large [1]	1563	25000	195.38	3125.00	7500	120000
r8idb.xlarge [1]	3125	25000	390.62	3125.00	15000	120000
r8idb.2xlarge [1]	6250	25000	781.25	3125.00	30000	120000
r8idb.4xlarge [1]	12500	25000	1562.50	3125.00	60000	120000
r8idb.8xlarge [2]	25000		3125.0		120000	
r8idb.12xlarge [2]	37500		4687.5		180000	
r8idb.16xlarge [2]	50000		6250.0		240000	
r8idb.24xlarge [2]	75000		9375.0		360000	
r8idb.32xlarge [2]	100000		12500.0		480000	
r8idb.48xlarge [2]	150000		18750.0		720000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
r8idb.96xlarge [2]	300000		37500.0		1440000	
r8idb.metal-48xl [2]	150000		18750.0		720000	
r8idb.metal-96xl [2]	300000		37500.0		1440000	
u-3tb1.56xlarge [2]	19000		2375.0		80000	
u-6tb1.56xlarge [2]	38000		4750.0		160000	
u-6tb1.112xlarge [2]	38000		4750.0		160000	
u-6tb1.metal [2]	38000		4750.0		160000	
u-9tb1.112xlarge [2]	38000		4750.0		160000	
u-9tb1.metal [2]	38000		4750.0		160000	
u-12tb1.112xlarge [2]	38000		4750.0		160000	
u-12tb1.metal [2]	38000		4750.0		160000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
u-18tb1.112xlarge [2]	38000		4750.0		160000	
u-18tb1.metal [2]	38000		4750.0		160000	
u-24tb1.112xlarge [2]	38000		4750.0		160000	
u-24tb1.metal [2]	38000		4750.0		160000	
u7i-6tb.112xlarge [2]	100000		12500.0		560000	
u7i-8tb.112xlarge [2]	100000		12500.0		560000	
u7i-12tb.224xlarge [2]	100000		12500.0		560000	
u7in-16tb.224xlarge [2]	100000		12500.0		560000	
u7in-24tb.224xlarge [2]	100000		12500.0		560000	
u7in-32tb.224xlarge [2]	100000		12500.0		560000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
u7inh-32tb.480xlarge [2]	160000		20000.0		840000	
x1.16xlarge [2]	7000		875.0		40000	
x1.32xlarge [2]	14000		1750.0		80000	
x1e.xlarge [2]	500		62.5		3700	
x1e.2xlarge [2]	1000		125.0		7400	
x1e.4xlarge [2]	1750		218.75		10000	
x1e.8xlarge [2]	3500		437.5		20000	
x1e.16xlarge [2]	7000		875.0		40000	
x1e.32xlarge [2]	14000		1750.0		80000	
x2gd.medium [1]	315	4750	39.38	593.75	2500	20000
x2gd.large [1]	630	4750	78.75	593.75	3600	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
x2gd.xlarge [1]	1188	4750	148.50	593.75	6000	20000
x2gd.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
x2gd.4xlarge [2]	4750		593.75		20000	
x2gd.8xlarge [2]	9500		1187.5		40000	
x2gd.12xlarge [2]	14250		1781.25		60000	
x2gd.16xlarge [2]	19000		2375.0		80000	
x2gd.metal [2]	19000		2375.0		80000	
x2idn.16xlarge [2]	40000		5000.0		173333	
x2idn.24xlarge [2]	60000		7500.0		260000	
x2idn.32xlarge [2]	80000		10000.0		260000	
x2idn.metal [2]	80000		10000.0		260000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
x2iedn.xlarge [1]	2500	20000	312.50	2500.00	8125	65000
x2iedn.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000
x2iedn.4xlarge [1]	10000	20000	1250.00	2500.00	32500	65000
x2iedn.8xlarge [2]	20000		2500.0		65000	
x2iedn.16xlarge [2]	40000		5000.0		130000	
x2iedn.24xlarge [2]	60000		7500.0		195000	
x2iedn.32xlarge [2]	80000		10000.0		260000	
x2iedn.metal [2]	80000		10000.0		260000	
x2iezn.2xlarge [2]	3170		396.25		13333	
x2iezn.4xlarge [2]	4750		593.75		20000	
x2iezn.6xlarge [2]	9500		1187.5		40000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
x2iezn.8xlarge [2]	12000		1500.0		55000	
x2iezn.12xlarge [2]	19000		2375.0		80000	
x2iezn.metal [2]	19000		2375.0		80000	
x8g.medium [1]	315	10000	39.38	1250.00	2500	40000
x8g.large [1]	630	10000	78.75	1250.00	3600	40000
x8g.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
x8g.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
x8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
x8g.8xlarge [2]	10000		1250.0		40000	
x8g.12xlarge [2]	15000		1875.0		60000	
x8g.16xlarge [2]	20000		2500.0		80000	
x8g.24xlarge [2]	30000		3750.0		120000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
x8g.48xlarge [2]	40000		5000.0		240000	
x8g.metal-24xl [2]	30000		3750.0		120000	
x8g.metal-48xl [2]	40000		5000.0		240000	
x8aedz.large [1]	1250	15000	156.25	1875.00	5000	60000
x8aedz.xlarge [1]	2500	15000	312.50	1875.00	10000	60000
x8aedz.3xlarge [1]	7500	15000	937.50	1875.00	30000	60000
x8aedz.6xlarge [2]	15000		1875.0		60000	
x8aedz.12xlarge [2]	30000		3750.0		120000	
x8aedz.24xlarge [2]	60000		7500.0		240000	
x8aedz.metal-12xl [2]	30000		3750.0		120000	
x8aedz.metal-24xl [2]	60000		7500.0		240000	
x8i.large [1]	650	10000	81.25	1250.00	3600	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
x8i.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
x8i.2xlarge [1]	2500	10000	312.50	1250.00	12000	40000
x8i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
x8i.8xlarge [2]	10000		1250.0		40000	
x8i.12xlarge [2]	15000		1875.0		60000	
x8i.16xlarge [2]	20000		2500.0		80000	
x8i.24xlarge [2]	30000		3750.0		120000	
x8i.32xlarge [2]	40000		5000.0		160000	
x8i.48xlarge [2]	60000		7500.0		240000	
x8i.64xlarge [2]	70000		8750.0		320000	
x8i.96xlarge [2]	80000		10000.0		480000	
x8i.metal-48xl [2]	60000		7500.0		240000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
x8i.metal-96xl [2]	80000		10000.0		480000	
z1d.large [1]	800	3170	100.00	396.25	3333	13333
z1d.xlarge [1]	1580	3170	197.50	396.25	6667	13333
z1d.2xlarge [2]	3170		396.25		13333	
z1d.3xlarge [2]	4750		593.75		20000	
z1d.6xlarge [2]	9500		1187.5		40000	
z1d.12xlarge [2]	19000		2375.0		80000	
z1d.metal [2]	19000		2375.0		80000	

[1] 这些实例可以至少每 24 小时维持一次 30 分钟的最大性能，之后会恢复到基线性能。

[2] 这些实例可以无限期地维持其既定性能。如果工作负载需要在超过 30 分钟的时长内保持最大性能，请选择以下任一实例。

存储优化

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
d2.xlarge [2]	750		93.75		6000	
d2.2xlarge [2]	1000		125.0		8000	
d2.4xlarge [2]	2000		250.0		16000	
d2.8xlarge [2]	4000		500.0		32000	
d3.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3.4xlarge [2]	2800		350.0		15000	
d3.8xlarge [2]	5000		625.0		30000	
d3en.xlarge [1]	850	2800	106.25	350.00	5000	15000
d3en.2xlarge [1]	1700	2800	212.50	350.00	10000	15000
d3en.4xlarge [2]	2800		350.0		15000	
d3en.6xlarge [2]	4000		500.0		25000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
d3en.8xlarge [2]	5000		625.0		30000	
d3en.12xlarge [2]	7000		875.0		40000	
h1.2xlarge [2]	1750		218.75		12000	
h1.4xlarge [2]	3500		437.5		20000	
h1.8xlarge [2]	7000		875.0		40000	
h1.16xlarge [2]	14000		1750.0		80000	
i3.large [2]	425		53.125		3000	
i3.xlarge [2]	850		106.25		6000	
i3.2xlarge [2]	1700		212.5		12000	
i3.4xlarge [2]	3500		437.5		16000	
i3.8xlarge [2]	7000		875.0		32500	
i3.16xlarge [2]	14000		1750.0		65000	
i3.metal [2]	19000		2375.0		80000	
i3en.large [1]	576	4750	72.10	593.75	3000	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
i3en.xlarge [1]	1153	4750	144.20	593.75	6000	20000
i3en.2xlarge [1]	2307	4750	288.39	593.75	12000	20000
i3en.3xlarge [1]	3800	4750	475.00	593.75	15000	20000
i3en.6xlarge [2]	4750		593.75		20000	
i3en.12xlarge [2]	9500		1187.5		40000	
i3en.24xlarge [2]	19000		2375.0		80000	
i3en.metal [2]	19000		2375.0		80000	
i4g.large [1]	625	10000	78.12	1250.00	2500	40000
i4g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4g.8xlarge [2]	10000		1250.0		40000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
i4g.16xlarge [2]	20000		2500.0		80000	
i4i.large [1]	625	10000	78.12	1250.00	2500	40000
i4i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i4i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i4i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i4i.8xlarge [2]	10000		1250.0		40000	
i4i.12xlarge [2]	15000		1875.0		60000	
i4i.16xlarge [2]	20000		2500.0		80000	
i4i.24xlarge [2]	30000		3750.0		120000	
i4i.32xlarge [2]	40000		5000.0		160000	
i4i.metal [2]	40000		5000.0		160000	
i7i.large [1]	625	10000	78.12	1250.00	2500	40000
i7i.xlarge [1]	1250	10000	156.25	1250.00	5000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
i7i.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i7i.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000
i7i.8xlarge [2]	10000		1250.0		40000	
i7i.12xlarge [2]	15000		1875.0		60000	
i7i.16xlarge [2]	20000		2500.0		80000	
i7i.24xlarge [2]	30000		3750.0		120000	
i7i.48xlarge [2]	60000		7500.0		240000	
i7i.metal-24xl [2]	30000		3750.0		120000	
i7i.metal-48xl [2]	60000		7500.0		240000	
i7ie.large [1]	625	10000	78.12	1250.00	2500	40000
i7ie.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i7ie.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
i7ie.3xlarge [1]	3750	10000	468.75	1250.00	15000	40000
i7ie.6xlarge [1]	7500	10000	937.50	1250.00	30000	40000
i7ie.12xlarge [2]	15000		1875.0		60000	
i7ie.18xlarge [2]	22500		2812.5		90000	
i7ie.24xlarge [2]	30000		3750.0		120000	
i7ie.48xlarge [2]	60000		7500.0		240000	
i7ie.metal-24xl [2]	30000		3750.0		120000	
i7ie.metal-48xl [2]	60000		7500.0		240000	
i8g.large [1]	625	10000	78.12	1250.00	2500	40000
i8g.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8g.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8g.4xlarge [1]	5000	10000	625.00	1250.00	20000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
i8g.8xlarge [2]	10000		1250.0		40000	
i8g.12xlarge [2]	15000		1875.0		60000	
i8g.16xlarge [2]	20000		2500.0		80000	
i8g.24xlarge [2]	30000		3750.0		120000	
i8g.48xlarge [2]	60000		7500.0		240000	
i8g.metal-24xl [2]	30000		3750.0		120000	
i8g.metal-48xl [2]	60000		7500.0		240000	
i8ge.large [1]	625	10000	78.12	1250.00	2500	40000
i8ge.xlarge [1]	1250	10000	156.25	1250.00	5000	40000
i8ge.2xlarge [1]	2500	10000	312.50	1250.00	10000	40000
i8ge.3xlarge [1]	3750	10000	468.75	1250.00	15000	40000
i8ge.6xlarge [1]	7500	10000	937.50	1250.00	30000	40000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
i8ge.12xlarge [2]	15000		1875.0		60000	
i8ge.18xlarge [2]	22500		2812.5		90000	
i8ge.24xlarge [2]	30000		3750.0		120000	
i8ge.48xlarge [2]	60000		7500.0		240000	
i8ge.metal-24xl [2]	30000		3750.0		120000	
i8ge.metal-48xl [2]	60000		7500.0		240000	
im4gn.large [1]	1250	10000	156.25	1250.00	5000	40000
im4gn.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
im4gn.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
im4gn.4xlarge [2]	10000		1250.0		40000	
im4gn.8xlarge [2]	20000		2500.0		80000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
im4gn.16xlarge [2]	40000		5000.0		160000	
is4gen.medium [1]	625	10000	78.12	1250.00	2500	40000
is4gen.large [1]	1250	10000	156.25	1250.00	5000	40000
is4gen.xlarge [1]	2500	10000	312.50	1250.00	10000	40000
is4gen.2xlarge [1]	5000	10000	625.00	1250.00	20000	40000
is4gen.4xlarge [2]	10000		1250.0		40000	
is4gen.8xlarge [2]	20000		2500.0		80000	

[1] 这些实例可以至少每 24 小时维持一次 30 分钟的最大性能，之后会恢复到基线性能。

[2] 这些实例可以无限期地维持其既定性能。如果工作负载需要在超过 30 分钟的时长内保持最大性能，请选择以下任一实例。

加速计算型

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
dl1.24xlarge [2]	19000		2375.0		80000	
dl2q.24xlarge [2]	19000		2375.0		80000	
f2.6xlarge [2]	7500		937.5		30000	
f2.12xlarge [2]	15000		1875.0		60000	
f2.48xlarge [2]	60000		7500.0		240000	
g3.4xlarge [2]	3500		437.5		20000	
g3.8xlarge [2]	7000		875.0		40000	
g3.16xlarge [2]	14000		1750.0		80000	
g4ad.xlarge [1]	400	3170	50.00	396.25	1700	13333
g4ad.2xlarge [1]	800	3170	100.00	396.25	3400	13333
g4ad.4xlarge [1]	1580	3170	197.50	396.25	6700	13333

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
g4ad.8xlarge [2]	3170		396.25		13333	
g4ad.16xlarge [2]	6300		787.5		26667	
g4dn.xlarge [1]	950	3500	118.75	437.50	3000	20000
g4dn.2xlarge [1]	1150	3500	143.75	437.50	6000	20000
g4dn.4xlarge [2]	4750		593.75		20000	
g4dn.8xlarge [2]	9500		1187.5		40000	
g4dn.12xlarge [2]	9500		1187.5		40000	
g4dn.16xlarge [2]	9500		1187.5		40000	
g4dn.metal [2]	19000		2375.0		80000	
g5.xlarge [1]	700	3500	87.50	437.50	3000	15000
g5.2xlarge [1]	850	3500	106.25	437.50	3500	15000
g5.4xlarge [2]	4750		593.75		20000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
g5.8xlarge [2]	16000		2000.0		65000	
g5.12xlarge [2]	16000		2000.0		65000	
g5.16xlarge [2]	16000		2000.0		65000	
g5.24xlarge [2]	19000		2375.0		80000	
g5.48xlarge [2]	19000		2375.0		80000	
g5g.xlarge [1]	1188	4750	148.50	593.75	6000	20000
g5g.2xlarge [1]	2375	4750	296.88	593.75	12000	20000
g5g.4xlarge [2]	4750		593.75		20000	
g5g.8xlarge [2]	9500		1187.5		40000	
g5g.16xlarge [2]	19000		2375.0		80000	
g5g.metal [2]	19000		2375.0		80000	
g6.xlarge [1]	1000	5000	125.00	625.00	4000	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
g6.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6.4xlarge [2]	8000		1000.0		32000	
g6.8xlarge [2]	16000		2000.0		64000	
g6.12xlarge [2]	20000		2500.0		80000	
g6.16xlarge [2]	20000		2500.0		80000	
g6.24xlarge [2]	30000		3750.0		120000	
g6.48xlarge [2]	60000		7500.0		240000	
g6e.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6e.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6e.4xlarge [2]	8000		1000.0		32000	
g6e.8xlarge [2]	16000		2000.0		64000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
g6e.12xlarge [2]	20000		2500.0		80000	
g6e.16xlarge [2]	20000		2500.0		80000	
g6e.24xlarge [2]	30000		3750.0		120000	
g6e.48xlarge [2]	60000		7500.0		240000	
g6f.large [1]	936	5000	117.00	625.00	3750	20000
g6f.xlarge [1]	1000	5000	125.00	625.00	4000	20000
g6f.2xlarge [1]	2000	5000	250.00	625.00	8000	20000
g6f.4xlarge [2]	6000		750.0		24000	
gr6.4xlarge [2]	8000		1000.0		32000	
gr6.8xlarge [2]	16000		2000.0		64000	
gr6f.4xlarge [2]	8000		1000.0		32000	
g7e.2xlarge [1]	2000	5000	250.00	625.00	8000	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
g7e.4xlarge [2]	8000		1000.0		32000	
g7e.8xlarge [2]	16000		2000.0		64000	
g7e.12xlarge [2]	25000		3125.0		100000	
g7e.24xlarge [2]	50000		6250.0		200000	
g7e.48xlarge [2]	100000		12500.0		400000	
inf1.xlarge [1]	1190	4750	148.75	593.75	4000	20000
inf1.2xlarge [1]	1190	4750	148.75	593.75	6000	20000
inf1.6xlarge [2]	4750		593.75		20000	
inf1.24xlarge [2]	19000		2375.0		80000	
inf2.xlarge [1]	1250	10000	156.25	1250.00	6000	40000
inf2.8xlarge [2]	10000		1250.0		40000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
inf2.24xlarge [2]	30000		3750.0		120000	
inf2.48xlarge [2]	60000		7500.0		240000	
p3.2xlarge [2]	1750		218.75		10000	
p3.8xlarge [2]	7000		875.0		40000	
p3.16xlarge [2]	14000		1750.0		80000	
p3dn.24xlarge [2]	19000		2375.0		80000	
p4d.24xlarge [2]	19000		2375.0		80000	
p4de.24xlarge [2]	19000		2375.0		80000	
p5.4xlarge [2]	10000		1250.0		32500	
p5.48xlarge [2]	80000		10000.0		260000	
p5e.48xlarge [2]	80000		10000.0		260000	

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
p5en.48xlarge [2]	100000		12500.0		400000	
p6-b200.48xlarge [2]	100000		12500.0		400000	
p6-b300.48xlarge [2]	100000		12500.0		400000	
p6e-gb200.36xlarge [2]	60000		7500.0		240000	
trn1.2xlarge [1]	5000	20000	625.00	2500.00	16250	65000
trn1.32xlarge [2]	80000		10000.0		260000	
trn1n.32xlarge [2]	80000		10000.0		260000	
trn2.3xlarge [2]	5000		625.0		16250	
trn2.48xlarge [2]	80000		10000.0		260000	
trn2u.48xlarge [2]	80000		10000.0		260000	
vt1.3xlarge [1]	2375	4750	296.88	593.75	10000	20000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
vt1.6xlarge [2]	4750		593.75		20000	
vt1.24xlarge [2]	19000		2375.0		80000	

[1] 这些实例可以至少每 24 小时维持一次 30 分钟的最大性能，之后会恢复到基线性能。

[2] 这些实例可以无限期地维持其既定性能。如果工作负载需要在超过 30 分钟的时长内保持最大性能，请选择以下任一实例。

高性能计算

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
hpc6a.48xlarge [1]	87	2085	10.88	260.62	500	11000
hpc6id.32xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.12xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.24xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7a.48xlarge [1]	87	2085	10.88	260.62	500	11000

实例大小	基准带宽（Mbps）	最大带宽（Mbps）	基准吞吐量（MB/s，128 KiB I/O）	最大吞吐量（MB/s，128 KiB I/O）	基准 IOPS（16KiB I/O）	最大 IOPS（16KiB I/O）
hpc7a.96xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.4xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.8xlarge [1]	87	2085	10.88	260.62	500	11000
hpc7g.16xlarge [1]	87	2085	10.88	260.62	500	11000
hpc8a.96xlarge [1]	87	2085	10.88	260.62	500	11000

[1] 这些实例可以至少每 24 小时维持一次 30 分钟的最大性能，之后会恢复到基线性能。

[2] 这些实例可以无限期地维持其既定性能。如果工作负载需要在超过 30 分钟的时长内保持最大性能，请选择以下任一实例。

支持的 EBS 优化

以下实例类型支持 EBS 优化，但默认情况下不启用 EBS 优化。您必须在启动期间或之后启用 EBS 优化，以达到所述的 EBS 性能水平，但需支付额外的小时费用。

实例大小	最大带宽 (Mbps)	最大吞吐量（MB/s，128 KiB I/O）	最大 IOPS (16 KiB I/O)
c1.xlarge	1000	125.0	8000
c3.xlarge	500	62.5	4000
c3.2xlarge	1000	125.0	8000

实例大小	最大带宽 (Mbps)	最大吞吐量（MB/s，128 KiB I/O）	最大 IOPS (16 KiB I/O)
c3.4xlarge	2000	250.0	16000
i2.xlarge	500	62.5	4000
i2.2xlarge	1000	125.0	8000
i2.4xlarge	2000	250.0	16000
m1.large	500	62.5	4000
m1.xlarge	1000	125.0	8000
m2.2xlarge	500	62.5	4000
m2.4xlarge	1000	125.0	8000
m3.xlarge	500	62.5	4000
m3.2xlarge	1000	125.0	8000
r3.xlarge	500	62.5	4000
r3.2xlarge	1000	125.0	8000
r3.4xlarge	2000	250.0	16000

> **Note**
>
> `i2.8xlarge`、`c3.8xlarge` 和 `r3.8xlarge` 实例没有专用 EBS 带宽，因此不提供 EBS 优化。在这些实例上，网络流量和 Amazon EBS 流量将共用同一 10 Gb 网络接口。

获得最佳的 Amazon EBS 优化性能

实例的 EBS 性能受实例类型的性能限制或其附加卷的聚合性能的限制，以较小者为准。要实现最大的 EBS 性能，实例必须附加卷以提供等于或大于最大实例性能的总性能。例如，要实现 `r6i.16xlarge` 的 `80,000` IOPS，实例必须至少具有预置了 `16,000` IOPS 的 5 个 gp2 卷（5 个卷 x `16,000` IOPS =

80,000 IOPS），或具有预置了 80,000 IOPS 的 1 个 gp3 卷。建议您选择一种实例类型，其提供的专用 Amazon EBS 吞吐量高于您的应用程序需求；否则，Amazon EBS 与 Amazon EC2 的连接将成为性能瓶颈。

> **Important**
>
> 使用可配置的带宽权重时，实例的 EBS 带宽限制可能会改变。对于使用 VPC-1 加权配置的实例（可增加网络带宽），由于在 IOPS 限制之前先达到 EBS 带宽限制，因此 EBS 卷的 IOPS 可能会低于预期。对于较大的 I/O 大小，这一点尤为明显。请始终测试您的特定工作负载，确保其符合您选择的带宽权重的性能要求。有关更多信息，请参阅 EC2 实例带宽权重配置。

您可以使用 EBSIOBalance% 和 EBSByteBalance% 指标来帮助您确定您的实例是否正确调整了大小。您可以在 CloudWatch 控制台中查看这些指标，并设置将根据您指定的阈值触发的警报。这些指标以百分比形式表示。具有持续低余额百分比的实例是上调大小的候选对象。余额百分比从未低于 100% 的实例是下调大小的候选对象。有关更多信息，请参阅使用 CloudWatch 监控您的实例。

内存增强型实例适合在云中运行大型内存中数据库，包括 SAP HANA 内存数据库的生产部署。要最大限度提高 EBS 性能，请使用包含偶数个 io1 或 io2 卷（具有相同的预置性能）的内存增强型实例。例如，对于 IOPS 密集型工作负载，请使用 4 个具有 40,000 预调配 IOPS 的 io1 或 io2 卷以获得最高 160,000 的实例 IOPS。同样，对于吞吐量密集型工作负载，请使用 6 个具有 48,000 预调配 IOPS 的 io1 或 io2 卷以获得最高 4,750 MB/s 的吞吐量。有关其他建议，请参阅 SAP HANA 的存储配置。

注意事项

- 2020 年 2 月 26 日之后启动的 G4dn、I3en、Inf1、M5a、M5ad、R5a、R5ad、T3、T3a 和 Z1d 实例可提供最高的 EBS 优化性能。要从 2020 年 2 月 26 日之前启动的实例中获得最高性能，请停止该实例，然后重新启动。
- 2019 年 12 月 3 日之后启动的 C5、C5d、C5n、M5、M5d、M5n、M5dn、R5、R5d、R5n、R5dn 和 P3dn 实例可提供最高的 EBS 优化性能。要从 2019 年 12 月 3 日之前启动的实例中获得最高性能，请停止该实例，然后重新启动。
- 2020 年 3 月 12 日之后启动的 u-6tb1.metal、u-9tb1.metal 和 u-12tb1.metal 实例可提供最高的 EBS 优化性能。2020 年 3 月 12 日之前启动的这些类型的实例可能会提供较低的性能。要从 2020 年 3 月 12 日之前启动的实例中获得最高性能，请与客户团队联系以免费升级实例。

查找 EBS 优化的 EC2 实例类型

您可以查看每个区域中支持 EBS 优化的实例类型。

Console

查找默认进行 EBS 优化的实例类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instance Types (实例类型)。
3. 添加筛选条件 EBS 优化支持 = default。
4. （可选）点击首选项图标，然后打开相关列，例如 EBS 最大 IOPS 和 EBS 基准 IOPS。
5. （可选）添加筛选条件以进一步筛选相关的特定实例类型。

Amazon CLI

查找默认进行 EBS 优化的实例类型

使用以下 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
--filters Name=ebs-info.ebs-optimized-support,Values=default  \
--query 'InstanceTypes[].{InstanceType:InstanceType,
 "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,
 MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops, "MaxThroughput(MB/
s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
--output=table
```

查找可选支持 EBS 优化的实例类型

使用以下 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
--filters Name=ebs-info.ebs-optimized-support,Values=supported \
--query 'InstanceTypes[].{InstanceType:InstanceType,
 "MaxBandwidth(Mb/s)":EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps,
 MaxIOPS:EbsInfo.EbsOptimizedInfo.MaximumIops, "MaxThroughput(MB/
s)":EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps}' \
--output=table
```

下面是 eu-west-1 的示例输出。

```
-------------------------------------------------------------------------
|                        DescribeInstanceTypes                          |
+--------------+----------------------+----------+-----------------------+
```

```
| InstanceType | MaxBandwidth(Mb/s)   | MaxIOPS  |  MaxThroughput(MB/s)  |
+--------------+----------------------+----------+-----------------------+
|  i2.2xlarge  |  1000                |  8000    |  125.0                |
|  m2.4xlarge  |  1000                |  8000    |  125.0                |
|  m2.2xlarge  |  500                 |  4000    |  62.5                 |
|  c1.xlarge   |  1000                |  8000    |  125.0                |
|  i2.xlarge   |  500                 |  4000    |  62.5                 |
|  m3.xlarge   |  500                 |  4000    |  62.5                 |
|  m1.xlarge   |  1000                |  8000    |  125.0                |
|  r3.4xlarge  |  2000                |  16000   |  250.0                |
|  r3.2xlarge  |  1000                |  8000    |  125.0                |
|  c3.xlarge   |  500                 |  4000    |  62.5                 |
|  m3.2xlarge  |  1000                |  8000    |  125.0                |
|  r3.xlarge   |  500                 |  4000    |  62.5                 |
|  i2.4xlarge  |  2000                |  16000   |  250.0                |
|  c3.4xlarge  |  2000                |  16000   |  250.0                |
|  c3.2xlarge  |  1000                |  8000    |  125.0                |
|  m1.large    |  500                 |  4000    |  62.5                 |
+--------------+----------------------+----------+-----------------------+
```

PowerShell

查找默认进行 EBS 优化的实例类型

使用 Get-EC2InstanceType cmdlet。

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="default"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

查找可选支持 EBS 优化的实例类型

使用 Get-EC2InstanceType cmdlet。

```
Get-EC2InstanceType `
    -Filter @{Name="ebs-info.ebs-optimized-support"; Values="supported"} | `
    Select InstanceType, `
        @{Name="MaxBandwidth(Mb/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumBandwidthInMbps)}}, `
```

```
        @{Name="MaxIOPS"; Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumIops)}}, `
        @{Name="MaxThroughput (MB/s)";
 Expression={($_.EbsInfo.EbsOptimizedInfo.MaximumThroughputInMBps)}}
```

下面是 eu-west-1 的示例输出。

```
InstanceType MaxBandwidth(Mb/s) MaxIOPS MaxThroughput (MB/s)
------------ ------------------ ------- --------------------
m2.4xlarge                1000    8000              125.000
i2.2xlarge                1000    8000              125.000
c1.xlarge                 1000    8000              125.000
m2.2xlarge                 500    4000               62.500
r3.2xlarge                1000    8000              125.000
m3.xlarge                  500    4000               62.500
r3.4xlarge                2000   16000              250.000
m1.xlarge                 1000    8000              125.000
i2.xlarge                  500    4000               62.500
c3.xlarge                  500    4000               62.500
c3.4xlarge                2000   16000              250.000
c3.2xlarge                1000    8000              125.000
i2.4xlarge                2000   16000              250.000
r3.xlarge                  500    4000               62.500
m3.2xlarge                1000    8000              125.000
m1.large                   500    4000               62.500
```

为 Amazon EC2 实例启用 EBS 优化

对于可选支持 EBS 优化的上一代实例类型，您只能手动启用 EBS 优化。如果您为这些实例类型启用了 EBS 优化，则需要按小时费率额外付费

先决条件

- 确认该实例类型需要启用 EBS 优化。有关更多信息，请参阅 支持的 EBS 优化。
- 要在启动后启用 EBS 优化，则必须首先停止实例。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

Console

要在启动期间启用 Amazon EBS 优化

在启动实例向导中，选择所需的实例类型。展开高级详细信息部分，然后对于 EBS 优化的实例，选择启用。

如果所选实例类型不支持 Amazon EBS 优化，则下拉列表将被禁用。如果实例类型默认为 Amazon EBS 优化，则已选择启用。

要在启动之后启用 Amazon EBS 优化

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择实例。
3. 停止实例。依次选择操作、实例状态、停止实例。
4. 在实例处于选中状态时，依次选择操作、实例设置、更改实例类型。
5. 选择 EBS 优化，然后选择应用。

   如果实例类型默认为 Amazon EBS 优化，或者不支持 Amazon EBS 优化，则该复选框将被禁用。
6. 重新启动实例。依次选择实例状态、启动实例。

Amazon CLI

要在启动期间启用 Amazon EBS 优化

使用 run-instances 命令和 `--ebs-optimized` 选项。

要在启动之后启用 Amazon EBS 优化

1. 如果实例正在运行，请使用 stop-instances 命令将其停止。

   ```
   aws ec2 stop-instances --instance-ids i-1234567890abcdef0
   ```
2. 使用 modify-instance-attribute 命令和 `--ebs-optimized` 选项来启用 EBS 优化。

   ```
   aws ec2 modify-instance-attribute \
       --instance-id i-1234567890abcdef0 \
       --ebs-optimized
   ```

PowerShell

要在启动期间启用 Amazon EBS 优化

使用 New-EC2Instance cmdlet 和 `-EbsOptimized` 选项。

要在启动之后启用 Amazon EBS 优化

1. 如果实例正在运行，请使用 Stop-EC2Instance cmdlet 将其停止。

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

2. 使用 Edit-EC2InstanceAttribute cmdlet 和 `-EbsOptimized` 选项启用 EBS 优化。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -EbsOptimized $true
```

Amazon EC2 实例适用的 CPU 选项

许多 Amazon EC2 实例都支持同步多线程（SMT），该技术可让多个线程在一个 CPU 核心上并发运行。每个线程都表示为实例上的一个虚拟 CPU (vCPU)。实例具有默认数量的 CPU 核心，根据实例类型而异。例如，默认情况下，`m5.xlarge` 实例类型有两个 CPU 内核，每个内核有两个线程—，共四个 vCPU。

在大多数情况下，都有一个 Amazon EC2 实例类型，它具有适合您工作负载的内存和 vCPU 数量组合。但是，为了针对特定工作负载或业务需求优化实例，您可以在实例启动期间和之后指定以下 CPU 选项：

- CPU 核心数：您可以自定义实例的 CPU 核心数。也许可以通过这种方式让实例拥有适合内存密集型工作负载的充足 RAM，同时减少 CPU 核心数，从而优化您的软件的许可成本。
- 每核心线程数：您可以通过为每个 CPU 核心指定一个线程来禁用 SMT。也许可以为特定工作负载执行该操作，例如高性能计算（HPC）工作负载。

注意事项

- 您无法修改 T2、C7a、M7a、R7a 和 Apple 硅芯片 Mac 实例以及基于 Amazon Graviton 处理器的实例的每个核心线程数。

- 您可以运行的实例数量基于所用实例类型的默认 vCPU。我们计算实例所消耗 vCPU 数量的方式不受更改其 CPU 选项的影响。

定价

指定 CPU 选项不会增加费用。对于从包含许可证的 Windows 和 SQL Server AMI 中启动的 EC2 实例，您可以自定义 CPU 选项，以利用 EC2 优化 CPU 功能，从而根据为您的实例配置的 vCPU 数量来支付许可费用。对于其他 EC2 实例，收费标准与使用默认 CPU 选项启动的实例相同。

内容

- 用于为 Amazon EC2 实例指定 CPU 选项的规则
- Amazon EC2 实例类型受支持的 CPU 选项
- 为 Amazon EC2 实例指定 CPU 选项
- 查看 Amazon EC2 实例的 CPU 线程和核心
- 包含许可证的实例的优化 CPU 功能

用于为 Amazon EC2 实例指定 CPU 选项的规则

要为您的实例指定 CPU 选项，请注意以下规则：

- 您无法为裸机实例指定 CPU 选项。
- 在实例启动过程中以及之后都可以指定 CPU 选项。
- 配置 CPU 选项时，必须在请求中同时指定 CPU 核心数和每核心线程数。有关示例请求，请参阅为 Amazon EC2 实例指定 CPU 选项。
- 实例的 vCPU 总数等于 CPU 内核数乘以每内核线程数。要指定自定义数量的 vCPU，必须为实例类型指定有效的 CPU 核心数和每核心线程数。不能超出实例的默认 vCPU 数量。有关更多信息，请参阅 Amazon EC2 实例类型受支持的 CPU 选项。
- 要禁用同步多线程（SMT）（也称为超线程），请为每个核心指定一个线程。
- 在控制台中，当您对现有实例更改实例类型时，Amazon EC2 会尽可能将现有实例的 CPU 选项设置应用到新实例。如果新的实例类型不支持这些设置，则 CPU 选项将重置为使用默认的 CPU 选项。对于新实例类型，此选项会使用默认 vCPU 数。

  要更新有关新实例的设置，请在更改实例类型视图的高级详细信息下，选择指定 CPU 选项。
- 停止、启动或重启实例后，仍将保留指定的 CPU 选项。

- 如果您使用预留实例，则在针对来自同一付款账户中包含许可证的 Windows AMI 创建的实例配置“优化 CPU”功能时，折扣可能无法适用。我们建议您使用节省计划来降低基于 vCPU 的许可成本，并提供与计算成本相关的同等节省金额。
- 为了节省从 Windows 和 SQL Server 包含许可证的 AMI 创建的实例的许可费用，您必须至少配置四个 vCPU。如果您配置的 vCPU 少于四个，则采用默认计费。
- T3 实例类型不支持针对包含许可证的实例的优化 CPU 功能。

Amazon EC2 实例类型受支持的 CPU 选项

下表列出了支持指定 CPU 选项的实例类型。

内容

- 通用实例
- 计算优化型实例
- 内存优化型实例
- 存储优化型实例
- 加速计算实例
- 高性能计算实例

通用实例

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m2.xlarge	2	2	1	1、2	1
m2.2xlarge	4	4	1	1、2、3、4	1
m2.4xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m3.large	2	1	2	1	1、2
m3.xlarge	4	2	2	1、2	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m3.2xlarge	8	4	2	1、2、3、4	1、2
m4.large	2	1	2	1	1、2
m4.xlarge	4	2	2	1、2	1、2
m4.2xlarge	8	4	2	1、2、3、4	1、2
m4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m4.10xlarge	40	20	2	2、4、6、8、10、12、14、16、18、20	1、2
m4.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5.large	2	1	2	1	1、2
m5.xlarge	4	2	2	2	1、2
m5.2xlarge	8	4	2	2、4	1、2
m5.4xlarge	16	8	2	2、4、6、8	1、2
m5.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m5a.large	2	1	2	1	1、2
m5a.xlarge	4	2	2	2	1、2
m5a.2xlarge	8	4	2	2、4	1、2
m5a.4xlarge	16	8	2	2、4、6、8	1、2
m5a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m5a.12xlarge	48	24	2	6、12、18、24	1、2
m5a.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m5a.24xlarge	96	48	2	12、18、24、36、48	1、2
m5ad.large	2	1	2	1	1、2
m5ad.xlarge	4	2	2	2	1、2
m5ad.2xlarge	8	4	2	2、4	1、2
m5ad.4xlarge	16	8	2	2、4、6、8	1、2
m5ad.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
m5ad.12xlarge	48	24	2	6、12、18、24	1、2
m5ad.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m5ad.24xlarge	96	48	2	12、18、24、36、48	1、2
m5d.large	2	1	2	1	1、2
m5d.xlarge	4	2	2	2	1、2
m5d.2xlarge	8	4	2	2、4	1、2
m5d.4xlarge	16	8	2	2、4、6、8	1、2
m5d.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5d.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m5d.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m5dn.large	2	1	2	1	1、2
m5dn.xlarge	4	2	2	1、2	1、2
m5dn.2xlarge	8	4	2	2、4	1、2
m5dn.4xlarge	16	8	2	2、4、6、8	1、2
m5dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m5dn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m5n.large	2	1	2	1	1、2
m5n.xlarge	4	2	2	1、2	1、2
m5n.2xlarge	8	4	2	2、4	1、2
m5n.4xlarge	16	8	2	2、4、6、8	1、2
m5n.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m5n.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m5n.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m5n.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m5zn.large	2	1	2	1	1、2
m5zn.xlarge	4	2	2	1、2	1、2
m5zn.2xlarge	8	4	2	2、4	1、2
m5zn.3xlarge	12	6	2	2、4、6	1、2
m5zn.6xlarge	24	12	2	2、4、6、8、10、12	1、2
m5zn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m6a.large	2	1	2	1	1、2
m6a.xlarge	4	2	2	1、2	1、2
m6a.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6a.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m6a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
m6a.12xlarge	48	24	2	1、2、3、4、5、6、7、8、16、24	1、2
m6a.16xlarge	64	32	2	4、6、8、10、12、14、16、32	1、2
m6a.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
m6a.32xlarge	128	64	2	8、12、16、20、24、28、32、64	1、2
m6a.48xlarge	192	96	2	8、12、16、20、24、28、32、64、96	1、2
m6g.large	2	2	1	1、2	1
m6g.xlarge	4	4	1	1、2、3、4	1
m6g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m6g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m6g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m6gd.large	2	2	1	1、2	1
m6gd.xlarge	4	4	1	1、2、3、4	1
m6gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m6gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m6gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m6i.large	2	1	2	1	1、2
m6i.xlarge	4	2	2	1、2	1、2
m6i.2xlarge	8	4	2	2、4	1、2
m6i.4xlarge	16	8	2	2、4、6、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6i.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m6i.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
m6i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m6i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m6id.large	2	1	2	1	1、2
m6id.xlarge	4	2	2	1、2	1、2
m6id.2xlarge	8	4	2	2、4	1、2
m6id.4xlarge	16	8	2	2、4、6、8	1、2
m6id.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
m6id.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6id.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
m6id.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m6id.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m6idn.large	2	1	2	1	1、2
m6idn.xlarge	4	2	2	1、2	1、2
m6idn.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m6idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m6idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m6idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m6idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m6in.large	2	1	2	1	1、2
m6in.xlarge	4	2	2	1、2	1、2
m6in.2xlarge	8	4	2	1、2、3、4	1、2
m6in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m6in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m6in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m6in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m6in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m7a.large	2	2	1	1、2	1
m7a.xlarge	4	4	1	1、2、3、4	1
m7a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m7a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
m7a.8xlarge	32	32	1	1、2、3、4、8、12、16、20、24、28、32	1
m7a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
m7a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
m7a.32xlarge	128	128	1	4、6、8、10、12、14、16、32、48、64、80、96、112、128	1
m7a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
m7g.large	2	2	1	1、2	1
m7g.xlarge	4	4	1	1、2、3、4	1
m7g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m7g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m7g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m7gd.large	2	2	1	1、2	1
m7gd.xlarge	4	4	1	1、2、3、4	1
m7gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m7gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m7gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
m7i.large	2	1	2	1	1、2
m7i.xlarge	4	2	2	1、2	1、2
m7i.2xlarge	8	4	2	1、2、3、4	1、2
m7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
m7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7i-flex.large	2	1	2	1	1、2
m7i-flex.xlarge	4	2	2	1、2	1、2
m7i-flex.2xlarge	8	4	2	1、2、3、4	1、2
m7i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m7i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m7i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m7i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8a.medium	1	1	1	1	1
m8a.large	2	2	1	1、2	1
m8a.xlarge	4	4	1	1、2、3、4	1
m8a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
m8a.8xlarge	32	32	1	1、2、3、4、8、12、16、20、24、28、32	1
m8a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
m8a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
m8a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
m8azn.medium	1	1	1	1	1
m8azn.large	2	2	1	1、2	1
m8azn.xlarge	4	4	1	1、2、3、4	1
m8azn.3xlarge	12	12	1	1、2、4、6、8、10、12	1
m8azn.6xlarge	24	24	1	1、2、3、4、8、12、16、20、24	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8azn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、16、24、32、40、48	1
m8azn.24xlarge	96	96	1	4、6、8、10、12、14、16、32、48、64、80、96	1
m8g.large	2	2	1	1、2	1
m8g.xlarge	4	4	1	1、2、3、4	1
m8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
m8gb.large	2	2	1	1、2	1
m8gb.xlarge	4	4	1	1、2、3、4	1
m8gb.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8gb.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8gb.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gb.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gb.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gb.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gb.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
m8gd.large	2	2	1	1、2	1
m8gd.xlarge	4	4	1	1、2、3、4	1
m8gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
m8gn.large	2	2	1	1、2	1
m8gn.xlarge	4	4	1	1、2、3、4	1
m8gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m8gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m8gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gn.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8gn.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
m8i.large	2	1	2	1	1、2
m8i.xlarge	4	2	2	1、2	1、2
m8i.2xlarge	8	4	2	1、2、3、4	1、2
m8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
m8id.large	2	1	2	1	1、2
m8id.xlarge	4	2	2	1、2	1、2
m8id.2xlarge	8	4	2	1、2、3、4	1、2
m8id.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8id.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8id.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8id.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8id.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8id.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m8id.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8id.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8i-flex.large	2	1	2	1	1、2
m8i-flex.xlarge	4	2	2	1、2	1、2
m8i-flex.2xlarge	8	4	2	1、2、3、4	1、2
m8i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8in.large	2	1	2	1	1、2
m8in.xlarge	4	2	2	1、2	1、2
m8in.2xlarge	8	4	2	1、2、3、4	1、2
m8in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m8in.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8in.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8idn.large	2	1	2	1	1、2
m8idn.xlarge	4	2	2	1、2	1、2
m8idn.2xlarge	8	4	2	1、2、3、4	1、2
m8idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m8idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8idn.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8idn.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
m8ine.large	2	1	2	1	1、2
m8ine.xlarge	4	2	2	1、2	1、2
m8ine.2xlarge	8	4	2	1、2、3、4	1、2
m8ine.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8ine.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8ine.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8ib.large	2	1	2	1	1、2
m8ib.xlarge	4	2	2	1、2	1、2
m8ib.2xlarge	8	4	2	1、2、3、4	1、2
m8ib.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8ib.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8ib.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
m8ib.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8ib.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8ib.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
m8ib.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8ib.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8idb.large	2	1	2	1	1、2
m8idb.xlarge	4	2	2	1、2	1、2
m8idb.2xlarge	8	4	2	1、2、3、4	1、2
m8idb.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
m8idb.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
m8idb.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8idb.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
m8idb.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
m8idb.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m8idb.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
m8idb.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
m9g.large	2	2	1	1、2	1
m9g.xlarge	4	4	1	1、2、3、4	1
m9g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m9g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
m9g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9g.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
m9gd.large	2	2	1	1、2	1
m9gd.xlarge	4	4	1	1、2、3、4	1
m9gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
m9gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
m9gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1
m9gd.16xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
m9gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
t3.nano	2	1	2	1	1、2
t3.micro	2	1	2	1	1、2
t3.small	2	1	2	1	1、2
t3.medium	2	1	2	1	1、2
t3.large	2	1	2	1	1、2
t3.xlarge	4	2	2	2	1、2
t3.2xlarge	8	4	2	2、4	1、2
t3a.nano	2	1	2	1	1、2
t3a.micro	2	1	2	1	1、2
t3a.small	2	1	2	1	1、2
t3a.medium	2	1	2	1	1、2
t3a.large	2	1	2	1	1、2
t3a.xlarge	4	2	2	2	1、2
t3a.2xlarge	8	4	2	2、4	1、2
t4g.nano	2	2	1	1、2	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
t4g.micro	2	2	1	1、2	1
t4g.small	2	2	1	1、2	1
t4g.medium	2	2	1	1、2	1
t4g.large	2	2	1	1、2	1
t4g.xlarge	4	4	1	1、2、3、4	1
t4g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

计算优化型实例

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c3.large	2	1	2	1	1、2
c3.xlarge	4	2	2	1、2	1、2
c3.2xlarge	8	4	2	1、2、3、4	1、2
c3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c3.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
c4.large	2	1	2	1	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c4.xlarge	4	2	2	1、2	1、2
c4.2xlarge	8	4	2	1、2、3、4	1、2
c4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c4.8xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5.large	2	1	2	1	1、2
c5.xlarge	4	2	2	2	1、2
c5.2xlarge	8	4	2	2、4	1、2
c5.4xlarge	16	8	2	2、4、6、8	1、2
c5.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c5.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c5.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c5a.large	2	1	2	1	1、2
c5a.xlarge	4	2	2	1、2	1、2
c5a.2xlarge	8	4	2	1、2、3、4	1、2
c5a.4xlarge	16	8	2	1、2、3、4、8	1、2
c5a.8xlarge	32	16	2	1、2、3、4、8、12、16	1、2
c5a.12xlarge	48	24	2	1、2、3、4、8、12、16、20、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c5a.16xlarge	64	32	2	1、2、3、4、8、12、16、20、24、28、32	1、2
c5a.24xlarge	96	48	2	1、2、3、4、8、12、16、20、24、28、32、36、40、44、48	1、2
c5ad.large	2	1	2	1	1、2
c5ad.xlarge	4	2	2	1、2	1、2
c5ad.2xlarge	8	4	2	1、2、3、4	1、2
c5ad.4xlarge	16	8	2	1、2、3、4、8	1、2
c5ad.8xlarge	32	16	2	1、2、3、4、8、12、16	1、2
c5ad.12xlarge	48	24	2	1、2、3、4、8、12、16、20、24	1、2
c5ad.16xlarge	64	32	2	1、2、3、4、8、12、16、20、24、28、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c5ad.24xlarge	96	48	2	1、2、3、4、8、12、16、20、24、28、32、36、40、44、48	1、2
c5d.large	2	1	2	1	1、2
c5d.xlarge	4	2	2	2	1、2
c5d.2xlarge	8	4	2	2、4	1、2
c5d.4xlarge	16	8	2	2、4、6、8	1、2
c5d.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
c5d.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c5d.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c5n.large	2	1	2	1	1、2
c5n.xlarge	4	2	2	2	1、2
c5n.2xlarge	8	4	2	2、4	1、2
c5n.4xlarge	16	8	2	2、4、6、8	1、2
c5n.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5n.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c6a.large	2	1	2	1	1、2
c6a.xlarge	4	2	2	1、2	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6a.2xlarge	8	4	2	1、2、3、4	1、2
c6a.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c6a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
c6a.12xlarge	48	24	2	1、2、3、4、5、6、7、8、16、24	1、2
c6a.16xlarge	64	32	2	4、6、8、10、12、14、16、32	1、2
c6a.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
c6a.32xlarge	128	64	2	8、12、16、20、24、28、32、64	1、2
c6a.48xlarge	192	96	2	8、12、16、20、24、28、32、64、96	1、2
c6g.large	2	2	1	1、2	1
c6g.xlarge	4	4	1	1、2、3、4	1
c6g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c6g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c6g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c6gd.large	2	2	1	1、2	1
c6gd.xlarge	4	4	1	1、2、3、4	1
c6gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c6gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c6gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c6gn.medium	1	1	1	1	1
c6gn.large	2	2	1	1、2	1
c6gn.xlarge	4	4	1	1、2、3、4	1
c6gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c6gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c6gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c6i.large	2	1	2	1	1、2
c6i.xlarge	4	2	2	1、2	1、2
c6i.2xlarge	8	4	2	2、4	1、2
c6i.4xlarge	16	8	2	2、4、6、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6i.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
c6i.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
c6i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
c6i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c6id.large	2	1	2	1	1、2
c6id.xlarge	4	2	2	1、2	1、2
c6id.2xlarge	8	4	2	2、4	1、2
c6id.4xlarge	16	8	2	2、4、6、8	1、2
c6id.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
c6id.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6id.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
c6id.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c6id.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c6in.large	2	1	2	1	1、2
c6in.xlarge	4	2	2	1、2	1、2
c6in.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c6in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c6in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c6in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c6in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c6in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c7a.large	2	2	1	1、2	1
c7a.xlarge	4	4	1	1、2、3、4	1
c7a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c7a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
c7a.8xlarge	32	32	1	1、2、3、4、8、12、16、20、24、28、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1
c7a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
c7a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
c7a.32xlarge	128	128	1	4、6、8、10、12、14、16、32、48、64、80、96、112、128	1
c7a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
c7g.large	2	2	1	1、2	1
c7g.xlarge	4	4	1	1、2、3、4	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c7g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c7g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c7gd.large	2	2	1	1、2	1
c7gd.xlarge	4	4	1	1、2、3、4	1
c7gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c7gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c7gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c7gn.large	2	2	1	1、2	1
c7gn.xlarge	4	4	1	1、2、3、4	1
c7gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c7gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c7gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
c7i.large	2	1	2	1	1、2
c7i.xlarge	4	2	2	1、2	1、2
c7i.2xlarge	8	4	2	1、2、3、4	1、2
c7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
c7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7i-flex.large	2	1	2	1	1、2
c7i-flex.xlarge	4	2	2	1、2	1、2
c7i-flex.2xlarge	8	4	2	1、2、3、4	1、2
c7i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c7i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c7i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c7i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8a.medium	1	1	1	1	1
c8a.large	2	2	1	1、2	1
c8a.xlarge	4	4	1	1、2、3、4	1
c8a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
c8a.8xlarge	32	32	1	1、2、3、4、8、12、16、20、24、28、32	1
c8a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
c8a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
c8a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
c8g.large	2	2	1	1、2	1
c8g.xlarge	4	4	1	1、2、3、4	1
c8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
c8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
c8gb.large	2	2	1	1、2	1
c8gb.xlarge	4	4	1	1、2、3、4	1
c8gb.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8gb.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c8gb.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gb.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gb.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gb.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gb.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
c8gd.large	2	2	1	1、2	1
c8gd.xlarge	4	4	1	1、2、3、4	1
c8gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c8gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
c8gn.large	2	2	1	1、2	1
c8gn.xlarge	4	4	1	1、2、3、4	1
c8gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
c8gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
c8gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gn.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8gn.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
c8i.large	2	1	2	1	1、2
c8i.xlarge	4	2	2	1、2	1、2
c8i.2xlarge	8	4	2	1、2、3、4	1、2
c8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
c8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
c8id.large	2	1	2	1	1、2
c8id.xlarge	4	2	2	1、2	1、2
c8id.2xlarge	8	4	2	1、2、3、4	1、2
c8id.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8id.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8id.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8id.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8id.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8id.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c8id.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
c8id.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8i-flex.large	2	1	2	1	1、2
c8i-flex.xlarge	4	2	2	1、2	1、2
c8i-flex.2xlarge	8	4	2	1、2、3、4	1、2
c8i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8in.large	2	1	2	1	1、2
c8in.xlarge	4	2	2	1、2	1、2
c8in.2xlarge	8	4	2	1、2、3、4	1、2
c8in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
c8in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c8in.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
c8in.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8ine.large	2	1	2	1	1、2
c8ine.xlarge	4	2	2	1、2	1、2
c8ine.2xlarge	8	4	2	1、2、3、4	1、2
c8ine.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8ine.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8ine.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8ib.large	2	1	2	1	1、2
c8ib.xlarge	4	2	2	1、2	1、2
c8ib.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8ib.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c8ib.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
c8ib.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
c8ib.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8ib.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c8ib.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
c8ib.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
c8ib.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

内存优化型实例

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r3.large	2	1	2	1	1、2
r3.xlarge	4	2	2	1、2	1、2
r3.2xlarge	8	4	2	1、2、3、4	1、2
r3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r3.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r4.large	2	1	2	1	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r4.xlarge	4	2	2	1、2	1、2
r4.2xlarge	8	4	2	1、2、3、4	1、2
r4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r4.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r4.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5.large	2	1	2	1	1、2
r5.xlarge	4	2	2	2	1、2
r5.2xlarge	8	4	2	2、4	1、2
r5.4xlarge	16	8	2	2、4、6、8	1、2
r5.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5a.large	2	1	2	1	1、2
r5a.xlarge	4	2	2	2	1、2
r5a.2xlarge	8	4	2	2、4	1、2
r5a.4xlarge	16	8	2	2、4、6、8	1、2
r5a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5a.12xlarge	48	24	2	6、12、18、24	1、2
r5a.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r5a.24xlarge	96	48	2	12、18、24、36、48	1、2
r5ad.large	2	1	2	1	1、2
r5ad.xlarge	4	2	2	2	1、2
r5ad.2xlarge	8	4	2	2、4	1、2
r5ad.4xlarge	16	8	2	2、4、6、8	1、2
r5ad.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
r5ad.12xlarge	48	24	2	6、12、18、24	1、2
r5ad.16xlarge	64	32	2	8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5ad.24xlarge	96	48	2	12、18、24、36、48	1、2
r5b.large	2	1	2	1	1、2
r5b.xlarge	4	2	2	1、2	1、2
r5b.2xlarge	8	4	2	2、4	1、2
r5b.4xlarge	16	8	2	2、4、6、8	1、2
r5b.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5b.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5b.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5b.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5d.large	2	1	2	1	1、2
r5d.xlarge	4	2	2	2	1、2
r5d.2xlarge	8	4	2	2、4	1、2
r5d.4xlarge	16	8	2	2、4、6、8	1、2
r5d.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5d.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5d.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5dn.large	2	1	2	1	1、2
r5dn.xlarge	4	2	2	1、2	1、2
r5dn.2xlarge	8	4	2	2、4	1、2
r5dn.4xlarge	16	8	2	2、4、6、8	1、2
r5dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5dn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r5n.large	2	1	2	1	1、2
r5n.xlarge	4	2	2	1、2	1、2
r5n.2xlarge	8	4	2	2、4	1、2
r5n.4xlarge	16	8	2	2、4、6、8	1、2
r5n.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r5n.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r5n.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r5n.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r6a.large	2	1	2	1	1、2
r6a.xlarge	4	2	2	1、2	1、2
r6a.2xlarge	8	4	2	1、2、3、4	1、2
r6a.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r6a.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
r6a.12xlarge	48	24	2	1、2、3、4、5、6、7、8、16、24	1、2
r6a.16xlarge	64	32	2	4、6、8、10、12、14、16、32	1、2
r6a.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6a.32xlarge	128	64	2	8、12、16、20、24、28、32、64	1、2
r6a.48xlarge	192	96	2	8、12、16、20、24、28、32、64、96	1、2
r6g.large	2	2	1	1、2	1
r6g.xlarge	4	4	1	1、2、3、4	1
r6g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r6g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r6g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r6gd.large	2	2	1	1、2	1
r6gd.xlarge	4	4	1	1、2、3、4	1
r6gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r6gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r6gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r6i.large	2	1	2	1	1、2
r6i.xlarge	4	2	2	1、2	1、2
r6i.2xlarge	8	4	2	2、4	1、2
r6i.4xlarge	16	8	2	2、4、6、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6i.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r6i.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
r6i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r6i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r6id.large	2	1	2	1	1、2
r6id.xlarge	4	2	2	1、2	1、2
r6id.2xlarge	8	4	2	2、4	1、2
r6id.4xlarge	16	8	2	2、4、6、8	1、2
r6id.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
r6id.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6id.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
r6id.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r6id.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r6idn.large	2	1	2	1	1、2
r6idn.xlarge	4	2	2	1、2	1、2
r6idn.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r6idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r6idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r6idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r6idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r6in.large	2	1	2	1	1、2
r6in.xlarge	4	2	2	1、2	1、2
r6in.2xlarge	8	4	2	1、2、3、4	1、2
r6in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r6in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r6in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r6in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r6in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r7a.large	2	2	1	1、2	1
r7a.xlarge	4	4	1	1、2、3、4	1
r7a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r7a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
r7a.8xlarge	32	32	1	1、2、3、4、8、12、16、20、24、28、32	1
r7a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
r7a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
r7a.32xlarge	128	128	1	4、6、8、10、12、14、16、32、48、64、80、96、112、128	1
r7a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
r7g.large	2	2	1	1、2	1
r7g.xlarge	4	4	1	1、2、3、4	1
r7g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r7g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r7g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r7gd.large	2	2	1	1、2	1
r7gd.xlarge	4	4	1	1、2、3、4	1
r7gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r7gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1
r7gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
r7i.large	2	1	2	1	1、2
r7i.xlarge	4	2	2	1、2	1、2
r7i.2xlarge	8	4	2	1、2、3、4	1、2
r7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
r7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7iz.large	2	1	2	1	1、2
r7iz.xlarge	4	2	2	1、2	1、2
r7iz.2xlarge	8	4	2	1、2、3、4	1、2
r7iz.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r7iz.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r7iz.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r7iz.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r7iz.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8a.medium	1	1	1	1	1
r8a.large	2	2	1	1、2	1
r8a.xlarge	4	4	1	1、2、3、4	1
r8a.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8a.4xlarge	16	16	1	1、2、4、6、8、10、12、14、16	1
r8a.8xlarge	32	32	1	1、2、3、4、8、12、16、20、24、28、32	1
r8a.12xlarge	48	48	1	1、2、3、4、5、6、12、18、24、30、36、42、48	1
r8a.16xlarge	64	64	1	1、2、3、4、5、6、7、8、16、24、32、40、48、56、64	1
r8a.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、24、36、48、60、72、84、96	1
r8a.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、48、72、96、120、144、168、192	1
r8g.large	2	2	1	1、2	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8g.xlarge	4	4	1	1、2、3、4	1
r8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
r8gb.large	2	2	1	1、2	1
r8gb.xlarge	4	4	1	1、2、3、4	1
r8gb.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8gb.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8gb.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gb.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gb.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gb.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gb.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
r8gd.large	2	2	1	1、2	1
r8gd.xlarge	4	4	1	1、2、3、4	1
r8gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gd.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gd.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
r8gn.large	2	2	1	1、2	1
r8gn.xlarge	4	4	1	1、2、3、4	1
r8gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
r8gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
r8gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gn.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gn.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8gn.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
r8i.large	2	1	2	1	1、2
r8i.xlarge	4	2	2	1、2	1、2
r8i.2xlarge	8	4	2	1、2、3、4	1、2
r8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
r8id.large	2	1	2	1	1、2
r8id.xlarge	4	2	2	1、2	1、2
r8id.2xlarge	8	4	2	1、2、3、4	1、2
r8id.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8id.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8id.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r8id.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8id.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8id.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8id.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8id.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8i-flex.large	2	1	2	1	1、2
r8i-flex.xlarge	4	2	2	1、2	1、2
r8i-flex.2xlarge	8	4	2	1、2、3、4	1、2
r8i-flex.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8i-flex.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8i-flex.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8i-flex.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8in.large	2	1	2	1	1、2
r8in.xlarge	4	2	2	1、2	1、2
r8in.2xlarge	8	4	2	1、2、3、4	1、2
r8in.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8in.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8in.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r8in.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8in.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8in.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8in.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8in.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8idn.large	2	1	2	1	1、2
r8idn.xlarge	4	2	2	1、2	1、2
r8idn.2xlarge	8	4	2	1、2、3、4	1、2
r8idn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8idn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8idn.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8idn.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8idn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r8idn.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8idn.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8idn.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
r8ib.large	2	1	2	1	1、2
r8ib.xlarge	4	2	2	1、2	1、2
r8ib.2xlarge	8	4	2	1、2、3、4	1、2
r8ib.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8ib.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8ib.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
r8ib.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8ib.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8ib.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
r8ib.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8ib.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8idb.large	2	1	2	1	1、2
r8idb.xlarge	4	2	2	1、2	1、2
r8idb.2xlarge	8	4	2	1、2、3、4	1、2
r8idb.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
r8idb.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
r8idb.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8idb.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
r8idb.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
r8idb.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
r8idb.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2
r8idb.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
u-3tb1.56xlarge	224	112	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u-6tb1.56xlarge	224	224	1	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224	1
u-6tb1.112xlarge	448	224	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u-9tb1.112xlarge	448	224	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224	1、2
u-12tb1.112xlarge	448	224	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u-18tb1.112xlarge	448	224	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224	1、2
u-24tb1.112xlarge	448	224	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7i-6tb.112xlarge	448	224	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112、116、120、124、128、132、136、140、144、148、152、156、160、164、168、172、176、180、184、188、192、196、200、204、208、212、216、220、224	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7i-8tb.112xlarge	448	224	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112、116、120、124、128、132、136、140、144、148、152、156、160、164、168、172、176、180、184、188、192、196、200、204、208、212、216、220、224	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7i-12tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7in-16tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7in-24tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7in-32tb.224xlarge	896	448	2	16、24、32、40、48、56、64、72、80、88、96、104、112、120、128、136、144、152、160、168、176、184、192、200、208、216、224、232、240、248、256、264、272、280、288、296、304、312、320、328、336、344、352、360、368、376、384、392、400、408、416、424、432、440、448	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
u7inh-32tb.480xlarge	1920	960	2	32、48、64、80、96、112、128、144、160、176、192、208、224、240、256、272、288、304、320、336、352、368、384、400、416、432、448、464、480、496、512、528、544、560、576、592、608、624、640、656、672、688、704、720、736、752、768、784、800、816、832、848、864、880、896、912、928、944、960	1、2
x1.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x1.32xlarge	128	64	2	4、8、12、16、20、24、28、32、36、40、44、48、52、56、60、64	1、2
x1e.xlarge	4	2	2	1、2	1、2
x1e.2xlarge	8	4	2	1、2、3、4	1、2
x1e.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
x1e.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
x1e.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
x1e.32xlarge	128	64	2	4、8、12、16、20、24、28、32、36、40、44、48、52、56、60、64	1、2
x2gd.large	2	2	1	1、2	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x2gd.xlarge	4	4	1	1、2、3、4	1
x2gd.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
x2gd.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
x2gd.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x2gd.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x2gd.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
x2idn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x2idn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
x2idn.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
x2iedn.xlarge	4	2	2	1、2	1、2
x2iedn.2xlarge	8	4	2	2、4	1、2
x2iedn.4xlarge	16	8	2	2、4、6、8	1、2
x2iedn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x2iedn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
x2iedn.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
x2iedn.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
x2iezn.2xlarge	8	4	2	2、4	1、2
x2iezn.4xlarge	16	8	2	2、4、6、8	1、2
x2iezn.6xlarge	24	12	2	2、4、6、8、10、12	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x2iezn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
x2iezn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
x8g.large	2	2	1	1、2	1
x8g.xlarge	4	4	1	1、2、3、4	1
x8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
x8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
x8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
x8aedz.large	2	2	1	1、2	1
x8aedz.xlarge	4	4	1	1、2、3、4	1
x8aedz.3xlarge	12	12	1	1、2、4、6、8、10、12	1
x8aedz.6xlarge	24	24	1	1、2、3、4、8、12、16、20、24	1
x8aedz.12xlarge	48	48	1	1、2、3、4、5、6、7、8、16、24、32、40、48	1
x8aedz.24xlarge	96	96	1	4、6、8、10、12、14、16、32、48、64、80、96	1
x8i.large	2	1	2	1	1、2
x8i.xlarge	4	2	2	1、2	1、2
x8i.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
x8i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
x8i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
x8i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8i.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
x8i.32xlarge	128	64	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
x8i.48xlarge	192	96	2	6、9、12、15、18、21、24、27、30、33、36、39、42、45、48、51、54、57、60、63、66、69、72、75、78、81、84、87、90、93、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
x8i.64xlarge	256	128	2	8、12、16、20、24、28、32、36、40、44、48、52、56、60、64、68、72、76、80、84、88、92、96、100、104、108、112、116、120、124、128	1、2
x8i.96xlarge	384	192	2	12、18、24、30、36、42、48、54、60、66、72、78、84、90、96、102、108、114、120、126、132、138、144、150、156、162、168、174、180、186、192	1、2
z1d.large	2	1	2	1	1、2
z1d.xlarge	4	2	2	1、2	1、2
z1d.2xlarge	8	4	2	2、4	1、2
z1d.3xlarge	12	6	2	2、4、6	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
z1d.6xlarge	24	12	2	2、4、6、8、10、12	1、2
z1d.12xlarge	48	24	2	4、6、8、10、12、14、16、18、20、22、24	1、2

存储优化型实例

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
d2.xlarge	4	2	2	1、2	1、2
d2.2xlarge	8	4	2	1、2、3、4	1、2
d2.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
d2.8xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
d3.xlarge	4	2	2	1、2	1、2
d3.2xlarge	8	4	2	2、4	1、2
d3.4xlarge	16	8	2	2、4、6、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
d3.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
d3en.xlarge	4	2	2	1、2	1、2
d3en.2xlarge	8	4	2	2、4	1、2
d3en.4xlarge	16	8	2	2、4、6、8	1、2
d3en.6xlarge	24	12	2	2、4、6、8、10、12	1、2
d3en.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
d3en.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
h1.2xlarge	8	4	2	1、2、3、4	1、2
h1.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
h1.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
h1.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
i2.xlarge	4	2	2	1、2	1、2
i2.2xlarge	8	4	2	1、2、3、4	1、2
i2.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i2.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
i3.large	2	1	2	1	1、2
i3.xlarge	4	2	2	1、2	1、2
i3.2xlarge	8	4	2	1、2、3、4	1、2
i3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
i3en.large	2	1	2	1	1、2
i3en.xlarge	4	2	2	1、2	1、2
i3en.2xlarge	8	4	2	2、4	1、2
i3en.3xlarge	12	6	2	2、4、6	1、2
i3en.6xlarge	24	12	2	2、4、6、8、10、12	1、2
i3en.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
i3en.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i4g.large	2	2	1	1、2	1
i4g.xlarge	4	4	1	1、2、3、4	1
i4g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
i4g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
i4g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i4g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
i4i.large	2	1	2	1	1、2
i4i.xlarge	4	2	2	1、2	1、2
i4i.2xlarge	8	4	2	1、2、3、4	1、2
i4i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i4i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
i4i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
i4i.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
i4i.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i4i.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
i7i.large	2	1	2	1	1、2
i7i.xlarge	4	2	2	1、2	1、2
i7i.2xlarge	8	4	2	1、2、3、4	1、2
i7i.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i7i.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i7i.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
i7i.16xlarge	64	32	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1、2
i7i.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i7i.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
i7ie.large	2	1	2	1	1、2
i7ie.xlarge	4	2	2	1、2	1、2
i7ie.2xlarge	8	4	2	1、2、3、4	1、2
i7ie.3xlarge	12	6	2	1、2、3、4、5、6	1、2
i7ie.6xlarge	24	12	2	1、2、3、4、5、6、7、8、9、10、11、12	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i7ie.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
i7ie.18xlarge	72	36	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i7ie.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2
i7ie.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
i8g.large	2	2	1	1、2	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8g.xlarge	4	4	1	1、2、3、4	1
i8g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
i8g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
i8g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8g.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8g.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8g.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
i8ge.large	2	2	1	1、2	1
i8ge.xlarge	4	4	1	1、2、3、4	1
i8ge.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
i8ge.3xlarge	12	12	1	1、2、3、4、5、6、7、8、9、10、11、12	1
i8ge.6xlarge	24	24	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8ge.12xlarge	48	48	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8ge.18xlarge	72	72	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8ge.24xlarge	96	96	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
i8ge.48xlarge	192	192	1	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96、98、100、102、104、106、108、110、112、114、116、118、120、122、124、126、128、130、132、134、136、138、140、142、144、146、148、150、152、154、156、158、160、162、164、166、168、170、172、174、176、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				178、180、182、184、186、188、190、192	
im4gn.large	2	2	1	1、2	1
im4gn.xlarge	4	4	1	1、2、3、4	1
im4gn.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
im4gn.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
im4gn.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
im4gn.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
is4gen.medium	1	1	1	1	1
is4gen.large	2	2	1	1、2	1
is4gen.xlarge	4	4	1	1、2、3、4	1
is4gen.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
is4gen.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
is4gen.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

加速计算实例

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
dl1.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
dl2q.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、2	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				2、24、26、28、30、32、34、36、38、40、42、44、46、48	
f2.6xlarge	24	12	2	1、2、3、6、9、12	1、2
f2.12xlarge	48	24	2	1、2、3、6、9、12、15、18、21、24	1、2
f2.48xlarge	192	96	2	4、6、8、10、12、24、36、48、60、72、84、96	1、2
g3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
g3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2
g4ad.xlarge	4	2	2	2	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g4ad.2xlarge	8	4	2	2、4	1、2
g4ad.4xlarge	16	8	2	2、4、8	1、2
g4ad.8xlarge	32	16	2	2、4、8、16	1、2
g4ad.16xlarge	64	32	2	2、4、8、16、32	1、2
g4dn.xlarge	4	2	2	2	1、2
g4dn.2xlarge	8	4	2	2、4	1、2
g4dn.4xlarge	16	8	2	2、4、6、8	1、2
g4dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
g4dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
g4dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g5g.xlarge	4	4	1	1、2、3、4	1
g5g.2xlarge	8	8	1	1、2、3、4、5、6、7、8	1
g5g.4xlarge	16	16	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1
g5g.8xlarge	32	32	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g5g.16xlarge	64	64	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64	1
g6.xlarge	4	2	2	1、2	1、2
g6.2xlarge	8	4	2	1、2、3、4	1、2
g6.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g6.8xlarge	32	16	2	1、2、4、6、8、10、12、14、16	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g6.12xlarge	48	24	2	1、2、3、6、9、12、15、18、21、24	1、2
g6.16xlarge	64	32	2	1、2、3、4、8、12、16、20、24、28、32	1、2
g6.24xlarge	96	48	2	1、2、3、4、5、6、12、18、24、30、36、42、48	1、2
g6.48xlarge	192	96	2	4、6、8、10、12、24、36、48、60、72、84、96	1、2
g6e.xlarge	4	2	2	1、2	1、2
g6e.2xlarge	8	4	2	1、2、3、4	1、2
g6e.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g6e.8xlarge	32	16	2	2、4、6、8、10、12、14、16	1、2
g6e.12xlarge	48	24	2	3、6、9、12、15、18、21、24	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g6e.16xlarge	64	32	2	4、8、12、16、20、24、28、32	1、2
g6e.24xlarge	96	48	2	6、12、18、24、30、36、42、48	1、2
g6e.48xlarge	192	96	2	12、24、36、48、60、72、84、96	1、2
g6f.large	2	1	2	1	1、2
g6f.xlarge	4	2	2	1、2	1、2
g6f.2xlarge	8	4	2	1、2、3、4	1、2
g6f.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
gr6.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
gr6.8xlarge	32	16	2	1、2、4、6、8、10、12、14、16	1、2
gr6f.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g7e.2xlarge	8	4	2	1、2、3、4	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g7e.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g7e.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
g7e.12xlarge	48	24	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24	1、2
g7e.24xlarge	96	48	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
g7e.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
inf1.xlarge	4	2	2	2	1、2
inf1.2xlarge	8	4	2	2、4	1、2
inf1.6xlarge	24	12	2	2、4、6、8、10、12	1、2
inf1.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
inf2.xlarge	4	2	2	1、2	1、2
inf2.8xlarge	32	16	2	4、6、8、10、12、14、16	1、2
inf2.24xlarge	96	48	2	4、6、8、10、12、14、16、32、48	1、2
inf2.48xlarge	192	96	2	4、8、12、16、20、24、28、32、64、96	1、2
p3.2xlarge	8	4	2	1、2、3、4	1、2
p3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16	1、2
p3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
p3dn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
p4d.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
p4de.24xlarge	96	48	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
p5.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
p5.48xlarge	192	96	2	12、24、36、48、60、72、84、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
p5e.48xlarge	192	96	2	12、24、36、48、60、72、84、96	1、2
p5en.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
p6-b200.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
p6-b300.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
p6e-gb200.36xlarge	144	144	1	1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17、18、19、20、21、22、23、24、25、26、27、28、29、30、31、32、33、34、35、36、37、38、39、40、41、42、43、44、45、46、47、48、49、50、51、52、53、54、55、56、57、58、59、60、61、62、63、64、65、66、67、68、69、70、71、72、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、90、91、92、93、94、95、96、97、98、99、100、101、1	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				02、103、104、105、106、107、108、109、110、111、112、113、114、115、116、117、118、119、120、121、122、123、124、125、126、127、128、129、130、131、132、133、134、135、136、137、138、139、140、141、142、143、144	
trn1.2xlarge	8	4	2	2、4	1、2
trn1.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
trn1n.32xlarge	128	64	2	2、4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	1、2
trn2.3xlarge	12	6	2	1、2、3、4、5、6	1、2
trn2.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
trn2u.48xlarge	192	96	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64、66、68、70、72、74、76、78、80、82、84、86、88、90、92、94、96	1、2
vt1.3xlarge	12	6	2	6	1、2
vt1.6xlarge	24	12	2	6、12	1、2
vt1.24xlarge	96	48	2	6、12、48	1、2

高性能计算实例

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
hpc6id.32xlarge	64	64	1	4、6、8、10、12、14、16、18、20、22、	1

实例类型	默认 vCPU	默认 CPU 核心数	默认每核心线程数	有效的 CPU 核心	每个核心的有效线程
				24、26、28、30、32、34、36、38、40、42、44、46、48、50、52、54、56、58、60、62、64	
hpc8a.96xlarge	192	192	1	24、48、72、96、120、144、168、192	1

为 Amazon EC2 实例指定 CPU 选项

在实例启动过程中以及之后都可以指定 CPU 选项。

任务

- 禁用同步多线程
- 在启动时指定自定义 vCPU 数
- 在启动模板中指定自定义 vCPU 数
- 更改 EC2 实例的 CPU 选项

禁用同步多线程

要禁用同步多线程（SMT）（也称为超线程），请为每个核心指定 1 个线程。

Console

在启动实例期间禁用 SMT

1. 按照 使用控制台中的启动实例向导来启动 EC2 实例 程序进行操作并根据需要配置实例。
2. 展开高级详细信息，然后选中指定 CPU 选项复选框。
3. 对于 Core count (内核数)，选择所需的 CPU 内核数量。在此示例中，要为 r5.4xlarge 实例指定默认 CPU 内核数，请选择 8。

4. 要禁用 SMT，请为每核心线程数选择 1。
5. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

在启动实例期间禁用 SMT

使用 run-instances Amazon CLI 命令，并将 1 参数的 ThreadsPerCore 值指定为 --cpu-options。对于 CoreCount，请指定 CPU 内核的数量。在此示例中，要为 r7i.4xlarge 实例指定默认 CPU 内核数，请指定值 8。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=8,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

在启动实例期间禁用 SMT

使用 New-EC2Instance 命令，并将 -CpuOptions 参数的 ThreadsPerCore 值指定为 1。对于 CoreCount，请指定 CPU 内核的数量。在此示例中，要为 r7i.4xlarge 实例指定默认 CPU 内核数，请指定值 8。

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=8; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

> **Note**
>
> 要为现有实例禁用 SMT，请按照更改 EC2 实例的 CPU 选项中所示的流程操作，并将每个核心运行的线程数更改为 1。

在启动时指定自定义 vCPU 数

您可以在启动实例时，通过 EC2 控制台或 Amazon CLI 自定义 CPU 核心数和每核心线程数。本节中的示例使用的是 `r5.4xlarge` 实例类型，该实例类型的默认设置如下：

- CPU 核心数：8
- 每核心线程数：2

默认情况下，实例会使用该实例类型可用的最大 vCPU 数量启动。对于这种实例类型，则共有 16 个 vCPU（8 个核心，每个核心运行 2 个线程）。有关此实例类型的更多信息，请参阅 内存优化型实例。

以下示例启动一个具有 4 个 vCPU 的 `r5.4xlarge` 实例。

Console

在实例启动期间指定自定义 vCPU 数量

1. 按照 使用控制台中的启动实例向导来启动 EC2 实例 程序进行操作并根据需要配置实例。
2. 展开高级详细信息，然后选中指定 CPU 选项复选框。
3. 要获得 4 个 vCPU，请指定 2 个 CPU 内核并为每个内核指定 2 个线程，如下所示：
   - 对于内核数，选择 2。
   - 对于 Threads per core (每内核线程数)，选择 2。
4. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

在实例启动期间指定自定义 vCPU 数量

使用 run-instances Amazon CLI 命令，并在 `--cpu-options` 参数中指定 CPU 内核数和线程数。可以指定 2 个 CPU 核心并为每个核心指定 2 个线程，从而获得 4 个 vCPU。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=2,ThreadsPerCore=2" \
    --key-name my-key-pair
```

或者，也可以通过指定 4 个 CPU 核心并为每个核心指定 1 个线程（禁用 SMT）来获得 4 个 vCPU：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r7i.4xlarge \
    --cpu-options "CoreCount=4,ThreadsPerCore=1" \
    --key-name my-key-pair
```

PowerShell

在实例启动期间指定自定义 vCPU 数量

使用 New-EC2Instance 命令，并在 `-CpuOptions` 参数中指定 CPU 核心数和线程数。可以指定 2 个 CPU 核心并为每个核心指定 2 个线程，从而获得 4 个 vCPU。

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=2; ThreadsPerCore=2} `
    -KeyName 'my-key-pair'
```

或者，也可以通过指定 4 个 CPU 核心并为每个核心指定 1 个线程（禁用 SMT）来获得 4 个 vCPU：

```
New-EC2Instance `
    -ImageId 'ami-0abcdef1234567890' `
    -InstanceType 'r7i.4xlarge' `
    -CpuOptions @{CoreCount=4; ThreadsPerCore=1} `
    -KeyName 'my-key-pair'
```

在启动模板中指定自定义 vCPU 数

您可以在启动模板中自定义实例的 CPU 核心数和每个核心的线程数。本节中的示例使用的是 `r5.4xlarge` 实例类型，该实例类型的默认设置如下：

- CPU 核心数：8
- 每核心线程数：2

默认情况下，实例会使用该实例类型可用的最大 vCPU 数量启动。对于这种实例类型，则共有 16 个 vCPU（8 个核心，每个核心运行 2 个线程）。有关此实例类型的更多信息，请参阅 内存优化型实例。

以下示例创建了一个启动模板，该模板指定具有 4 个 vCPU 的 `r5.4xlarge` 实例的配置。

Console

要在启动模板中指定自定义 vCPU 数

1. 按照 通过指定参数创建启动模板 程序进行操作并根据需要配置启动模板。
2. 展开高级详细信息，然后选中指定 CPU 选项复选框。
3. 要获得 4 个 vCPU，请指定 2 个 CPU 内核并为每个内核指定 2 个线程，如下所示：
   - 对于内核数，选择 2。
   - 对于 Threads per core (每内核线程数)，选择 2。
4. 在摘要面板中查看实例配置，然后选择创建启动模板。有关更多信息，请参阅 在 Amazon EC2 启动模板中存储实例启动参数。

Amazon CLI

要在启动模板中指定自定义 vCPU 数

使用 create-launch-template Amazon CLI 命令，并在 `CpuOptions` 参数中指定 CPU 核心数和线程数。可以指定 2 个 CPU 核心并为每个核心指定 2 个线程，从而获得 4 个 vCPU。

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForCPUOptions \
    --version-description CPUOptionsVersion1 \
    --launch-template-data file://template-data.json
```

下面是一个示例 JSON 文件，其中包含启动模板数据，包括本示例的实例配置的 CPU 选项。

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
```

```
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":2,
        "ThreadsPerCore":2
    }
}
```

或者，也可以通过指定 4 个 CPU 核心并为每个核心指定 1 个线程（禁用 SMT）来获得 4 个 vCPU：

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":1
    }
}
```

PowerShell

要在启动模板中指定自定义 vCPU 数

使用 New-EC2LaunchTemplate。

```
New-EC2LaunchTemplate `
    -LaunchTemplateName 'TemplateForCPUOptions' `
    -VersionDescription 'CPUOptionsVersion1' `
    -LaunchTemplateData (Get-Content -Path 'template-data.json' | ConvertFrom-Json)
```

更改 EC2 实例的 CPU 选项

随着需求的逐渐变化，您可能需要更改现有实例的 CPU 选项配置。实例上运行的每个线程都称为一个虚拟 CPU（vCPU）。您可以通过 Amazon EC2 控制台、Amazon CLI、API 或 SDK 更改现有实例中运行的 vCPU 数量。实例状态必须为 Stopped，然后才能进行此更改。

要查看控制台或命令行步骤，请选择与环境相匹配的选项卡。有关 API 请求和响应信息，请参阅《Amazon EC2 API 参考》中的 ModifyInstanceCpuOptions。

Console

按照以下过程操作，通过 Amazon Web Services 管理控制台更改实例的活动 vCPU 数量。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。这将打开为当前 Amazon Web Services 区域定义的实例列表。
3. 从实例列表中选择实例。您也可以通过选择实例链接来打开实例详细信息页面。
4. 如果实例正在运行，则必须首先停止实例，然后才能继续操作。从实例状态菜单中，选择停止实例。
5. 要更改 vCPU 配置，请从操作菜单的实例设置中，选择更改 CPU 选项。这将打开更改 CPU 选项页面。
6. 选择以下 CPU 选项之一来更改实例的配置。

   使用默认的 CPU 选项

   此选项会将实例重置为实例类型的默认 vCPU 数量。默认情况下会运行所有 CPU 核心的所有线程。

   指定 CPU 选项

   借助此选项可以配置实例上运行的 vCPU 数量。
7. 如果选择指定 CPU 选项，则会显示活动 vCPU 字段。

- 使用第一个选择器来配置每个 CPU 核心的线程数。要禁用同步多线程，请选择 1。
- 使用第二个选择器来配置在实例上运行的 CPU 数量。

当您更改 CPU 选项选择器时，以下字段会动态更新。

- 活动 vCPU：CPU 核心数乘以每核心线程数，具体取决于您的选择。例如，假设您选择了 2 个线程和 4 个核心，则该值为 8 个 vCPU。
- 总 vCPU 数：该实例类型的最大 vCPU 数。例如，对于 `m6i.4xlarge` 实例类型，这将为 16 个 vCPU（8 个核心，每个核心运行 2 个线程）。

8. 要应用更新，请选择更改。

Amazon CLI

按照以下过程操作，通过 Amazon CLI更改实例的活动 vCPU 数量。

使用 modify-instance-cpu-options 命令，然后在 `--core-count` 参数中指定运行的 CPU 核心数，并在 `--threads-per-core` 参数中指定每个核心运行的线程数。

以下示例演示了对于 `m6i.4xlarge` 实例类型，用于在指定实例上运行 8 个 vCPU 的两种可能配置。这种实例类型默认为 16 个 vCPU（8 个核心，每个核心运行 2 个线程）。

示例 1：运行 4 个 CPU 核心，每个核心 2 个线程，总计 8 个 vCPU。

```
aws ec2 modify-instance-cpu-options \
    --instance-id i-1234567890abcdef0 \

    --core-count=4 \
    --threads-per-core=2
```

示例 2：通过将每个核心运行的线程数更改为 1 来禁用同步多线程。得出的配置同样总计运行 8 个 vCPU（8 个 CPU 核心，每个核心 1 个线程）。

```
aws ec2 modify-instance-cpu-options \
    --instance-id 1234567890abcdef0 \
    --core-count=8 \
    --threads-per-core=1
```

PowerShell

更改实例的活动 vCPU 数量

使用 Edit-EC2InstanceCpuOption cmdlet，在 `-CoreCount` 参数中指定运行的 CPU 核心数，并且在 `ThreadsPerCore` 参数中指定每个核心运行的线程数。

示例 1：运行 4 个 CPU 核心，每个核心 2 个线程，总计 8 个 vCPU。

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 4 `
    -ThreadsPerCore 2
```

示例 2：通过将每个核心运行的线程数更改为 1 来禁用同步多线程。得出的配置同样总计运行 8 个 vCPU（8 个 CPU 核心，每个核心 1 个线程）。

```
Edit-EC2InstanceCpuOption `
    -InstanceId 'i-1234567890abcdef0' `
    -CoreCount 8 `
    -ThreadsPerCore 1
```

查看 Amazon EC2 实例的 CPU 线程和核心

您可以通过描述实例来查看现有实例的 CPU 选项。

Console

查看实例的 CPU 选项

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择实例，然后选择实例。
3. 在详细信息选项卡中的主机和置放群组下，找到 vCPU 数量。

Amazon CLI

查看实例的 CPU 选项

可以使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].CpuOptions
```

下面是示例输出。CoreCount 字段指示实例的核心数。ThreadsPerCore 字段指示每核心线程数。

```
[
    {
        "CoreCount": 24,
        "ThreadsPerCore": 2
    },
]
```

PowerShell

查看实例的 CPU 选项

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId 'i-1234567890abcdef0').Instances.CpuOptions
```

下面是示例输出。

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
          24        2
```

或者，要查看 CPU 信息，您可以连接到实例并使用以下系统工具之一：

- Windows 实例上的 Windows Task Manager
- Linux 实例上的 lscpu 命令

可以使用 Amazon Config 记录、评测、审计实例的配置更改，包括终止的实例。有关更多信息，请参阅 Amazon Config 开发人员指南中的 Amazon Config 入门

包含许可证的实例的优化 CPU 功能

Microsoft SQL Server 之类的工作负载通常需要大量的内存和 IOPS，但同时又需要较低的 vCPU 数量。Amazon 提供了种类繁多的实例类型，能够满足您大部分的基础设施需求。然而，为了降低 Windows 和 Microsoft SQL Server 基于 vCPU 的许可费用，您可以自定义在您的 EC2 实例上运行的 vCPU 数量，同时保持相同的内存、存储和网络规格。这种方法可以节省基于 vCPU 的许可费用（包括包含许可证工作负载和自带许可（BYOL）工作负载）。您可以使用控制台或 Amazon EC2 API 在启动或修改实例时指定 CPU 选项。有关说明，请参阅为 Amazon EC2 实例指定 CPU 选项。

有关更多信息，请参阅这篇关于优化 SQL Server 工作负载的 CPU 的最佳实践的博客文章。

支持的许可证类型

对于从包含许可证的 AMI 启动的实例，“优化 CPU”功能支持根据活跃 CPU 的数量对以下类型的许可证配置进行计费。有关许可证类型的更多信息，请参阅 AMI 账单信息字段。

含许可证的 AMI 实例账单

包含许可证	使用情况操作	每 vCPU 小时价格
Windows Server	RunInstances:0002	0.046 美元
含有 SQL Server Enterprise 的 Windows Server	RunInstances:0102	0.421 美元
含有 SQL Server Standard 的 Windows Server	RunInstances:0006	0.166 美元
包含 SQL Server Web 的 Windows Server	RunInstances:0202	0.063 美元

支持的购买选项

“优化 CPU”功能为包含许可证的实例提供以下购买选项：

- 按需型
- 节省计划

> **Warning**
>
> 如果您使用预留实例，则在针对同一付款账户中包含许可证的实例配置“优化 CPU”功能时，折扣可能无法适用。我们建议您使用节省计划来降低基于 vCPU 的许可成本，并提供与计算成本相关的同等节省金额。
>
> 在 2025 年 10 月 15 日之前，那些在同一实例类型上为 Windows 和 SQL Server 同时使用了“优化 CPU”和“预留实例”功能的账户已被列入退出列表，以维持其现有的计费体验。要利用优化 CPU 许可证节省的优势，请联系 Amazon Web Services 支持 中心将其从选择退出列表中移除。

“优化 CPU”功能如何节省许可费用

以下示例有助于说明在配置 CPU 使用率时所能实现的成本节省情况。

示例 1：默认账单此示例展示了通过包含许可证的 Windows 和 SQL Server Enterprise AMI 启动的 r7i.8xlarge 实例。该实例运行了 100 小时，其实例类型默认的 CPU 配置为 32 个 vCPU（总计 3200 个 vCPU 小时）。

该账单中有一个行项目列出了综合费率，其中既包含了使用费用，也包含了许可费用。

For illustration purposes only **Amazon Web Services, Inc. charges by service** **Total services: 1** **Description**	**Usage Quantity**	**Total charges** **$1,558.88** **Amount in USD**
Elastic Compute Cloud		$1,558.88
|- US East (N. Virginia)		$1,558.88
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$1,558.88
| |- $15.5888 per On Demand Windows with SQL Server Enterprise r7i.8xlarge Instance Hour	100 Hrs	$1,558.88

示例 2：优化 CPU 账单此示例展示了通过使用包含许可证的 Windows 和 SQL Server Enterprise AMI 来启动的 r7i.8xlarge 实例。为了节省许可证费用，处于活动状态的 CPU 数量减少到 16 个 vCPU。然后，该实例使用新配置运行了 100 个小时。

账单显示以下行项目。

计费说明：Elastic Compute Cloud

第一个行项目显示了运行 100 小时的 Windows 和 SQL Server 实例的基准成本（211.68 美元）。

计费说明：Amazon EC2 优化 CPU 包含许可证的第三方费用

第二个行项目涵盖了基于计费期间内处于活动状态的 vCPU 数量而产生的许可费用（673.60 美元）。

For illustration purposes only Amazon Web Services, Inc. charges by service Total services: 2 Description	Usage Quantity	Total charges $885.28 Amount in USD
Elastic Compute Cloud		$211.68
|- US East (N. Virginia)		$211.68
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$211.68
| |- $2.1168 per On Demand Windows and SQL Server Enterprise without licenses r7i.8xlarge Instance Hr	100 Hrs	$211.68
Amazon EC2 Optimize CPU License Included Third Party Fees		$673.60
|- US East (N. Virginia)		$673.60
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$673.60
| |- $0.421 per vCPU-Hour of On Demand Windows with SQL Server licenses on r7i Instance	1,600 vCPU-Hrs	$673.60

示例 3：采用节省计划的优化 CPU 账单此示例展示了通过使用包含许可证的 Windows 和 SQL Server Enterprise AMI 来启动的 r7i.8xlarge 实例。为了节省许可证费用，处于活动状态的 CPU 数量减少到 16 个 vCPU。然后，该实例使用新配置运行了 100 个小时。

一项为期一年的无预付费用计算类节省计划，每小时承诺费用为 1.60 美元（四舍五入），可进一步节省成本，从而降低 Windows 和 SQL Server 实例的基准成本。节省计划承诺涵盖了 r7i.8xlarge 实例的全部 100 小时使用时长，其节省计划的费率为每小时 1.53362 美元。

账单显示以下行项目。

计费说明：计算类使用量的节省计划

第一个行项目显示了整整 100 小时使用量的节省计划承诺（160.00 美元）。

计费说明：Elastic Compute Cloud

第二个行项目包含两个条目。第一个条目显示了运行 100 小时的 Windows 和 SQL Server 实例的基准成本在未采用节省计划时原本为（211.68 美元）。第二个条目显示，全部基准成本已由计算类节省计划承担（费用为 -211.68 美元），因此该行项目的净成本为零。

计费说明：Amazon EC2 优化 CPU 包含许可证的第三方费用

第三个行项目涵盖了基于计费期间内处于活动状态的 vCPU 数量而产生的许可费用（673.60 美元）。

For illustration purposes only Amazon Web Services, Inc. charges by service Total services: 3 Description	Usage Quantity	Total charges $833.60 Amount in USD
Savings Plans for Compute usage		$160.00
|- Any		$160.00
|- Compute Savings Plans		$160.00
| |- 1 year No Upfront Compute Savings Plan	100 Hrs	$160.00
Elastic Compute Cloud		$0.00
|- US East (N. Virginia)		$0.00
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$0.00
| |- $2.1168 per On Demand Windows and SQL Server Enterprise without licenses r7i.8xlarge Instance Hr	100 Hrs	$211.68
| |- r7i.8xlarge Windows with SQL Server Enterprise without licenses instance usage covered by CSP	100 Hrs	($211.68)
Amazon EC2 Optimize CPU License Included Third Party Fees		$673.60
|- US East (N. Virginia)		$673.60
|- Amazon Elastic Compute Cloud running Windows and SQL Server Enterprise		$673.60
| |- $0.421 per vCPU-Hour of On Demand Windows with SQL Server licenses on r7i Instance	1,600 vCPU-Hrs	$673.60

Amazon EC2 实例的 AMD SEV-SNP

AMD 安全加密虚拟化–安全嵌套分页（AMD SEV-SNP）CPU 功能具有以下属性：

- 证明 – AMD SEV-SNP 使您能够检索已签名的证明报告，其中包含可用于验证实例状态和身份并证明实例在正版 AMD 硬件上运行的加密措施。有关更多信息，请参阅 使用 AMD SEV-SNP 证明 Amazon EC2 实例。
- 内存加密 – 从 AMD EPYC（米兰）、Amazon Graviton2 和英特尔至强可扩展（Ice Lake）处理器开始，实例内存始终是加密的。启用 AMD SEV-SNP 的实例使用实例特定的密钥进行内存加密。

目录

- 概念和术语
- 要求
- 注意事项
- 定价
- 查找支持 AMD SEV-SNP 的 EC2 实例类型
- 为 EC2 实例启用 AMD SEV-SNP
- 使用 AMD SEV-SNP 证明 Amazon EC2 实例

概念和术语

在开始使用 AMD SEV-SNP 之前，请先熟悉以下概念和术语。

AMD SEV-SNP 证明报告

AMD SEV-SNP 证明报告是实例可以请求 CPU 提供的一种文档。AMD SEV-SNP 证明报告可用于验证实例的状态和身份，以及验证实例是否在认可的 AMD 环境中运行。证明报告包括一项启动指标，该指标是实例初始引导状态的加密哈希值，其中包括其初始实例内存内容和 vCPU 的初始状态。AMD SEV-SNP 证明报告采用 VLEK 签名签署，该签名可回链至 AMD 信任根。

VLEK

版本控制加载认可密钥（VLEK）是 AMD 认证的一种版本控制签名密钥，由 AMD CPU 用于签署 AMD SEV-SNP 证明报告。VLEK 签名可通过 AMD 提供的证书来进行验证。

OVMF 二进制文件

开放虚拟机固件（OVMF）是用于为实例提供 UEFI 环境的早期启动代码。早期启动代码将早于 AMI 中的代码启动运行。OVMF 还将查找并运行 AMI 中提供的引导加载程序。有关更多信息，请参阅 OVMF 存储库。

要求

要使用 AMD SEV-SNP，您必须执行以下操作：

- 使用以下任何一种支持的实例类型：
  - 通用型：`m6a.large` |`m6a.xlarge` |`m6a.2xlarge` |`m6a.4xlarge` |`m6a.8xlarge`
  - 计算优化型：`c6a.large` |`c6a.xlarge` |`c6a.2xlarge` |`c6a.4xlarge` |`c6a.8xlarge` |`c6a.12xlarge` |`c6a.16xlarge`
  - 内存优化型：`r6a.large`|`r6a.xlarge` |`r6a.2xlarge` |`r6a.4xlarge`
- 在支持的 Amazon Web Services 区域中启动实例。目前仅支持美国东部（俄亥俄州）和欧洲地区（爱尔兰）区域。
- 使用带有 `uefi` 或 `uefi-preferred` 启动模式的 AMI 以及支持 AMD SEV-SNP 的操作系统。要详细了解操作系统对 AMD SEV-SNP 的支持，请参阅相应操作系统的文档。对于 Amazon，AL2023、RHEL 9.3 SLES 15 SP4 以及 Ubuntu 23.04 和更高版本都支持 AMD SEV-SNP。

注意事项

您只能在启动实例时启用 AMD SEV-SNP。如果在实例启动时启用 AMD SEV-SNP，则以下规则适用。

- AMD SEV-SNP 启用后便无法禁用。会在整个实例生命周期中保持启用状态。

- 您只能将实例类型更改为支持 AMD SEV-SNP 的其他实例类型。
- 不支持休眠和 Nitro Enclaves。
- 不支持专属主机。
- 如果实例的底层主机计划进行维护，您将在该事件发生前 14 天收到计划事件通知。您必须手动停止或重启实例，然后才能将其迁移到新主机。

定价

在启用了 AMD SEV-SNP 的情况下启动 Amazon EC2 实例时，您需要按小时支付额外的使用费，费率为所选实例类型按需小时费率 的 10%。

此 AMD SEV-SNP 使用费将按您的 Amazon EC2 实例使用量单独收取。预留实例、Savings Plans 和操作系统使用情况不会影响此费用。

如果将竞价型实例配置为在启用了 AMD SEV-SNP 的情况下启动，您需要按小时支付额外的使用费，费率为所选实例类型按需小时费率的 10%。如果分配策略使用价格作为输入，则竞价型实例集不包括这笔额外费用；只使用 Spot 价格。

查找支持 AMD SEV-SNP 的 EC2 实例类型

您可以查找支持 AMD SEV-SNP 的实例类型。Amazon EC2 控制台不显示实例类型的此信息。

Amazon CLI

查找支持 AMD SEV-SNP 的实例类型

使用以下 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
    --filters Name=processor-info.supported-features,Values=amd-sev-snp \
    --query 'InstanceTypes[*].[InstanceType]' \
    --output text | sort
```

下面是示例输出。

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
```

```
c6a.4xlarge
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

PowerShell

查找支持 AMD SEV-SNP 的实例类型

使用 Get-EC2InstanceType cmdlet。

```
(Get-EC2InstanceType `
    -Filter @{Name="processor-info.supported-features"; Values="amd-sev-
snp"}).InstanceType.Value | Sort-Object
```

下面是示例输出。

```
c6a.12xlarge
c6a.16xlarge
c6a.2xlarge
c6a.4xlarge
c6a.8xlarge
c6a.large
c6a.xlarge
m6a.2xlarge
m6a.4xlarge
m6a.8xlarge
m6a.large
m6a.xlarge
r6a.2xlarge
r6a.4xlarge
r6a.large
r6a.xlarge
```

为 EC2 实例启用 AMD SEV-SNP

您可以在启用了 AMD SEV-SNP 的情况下启动实例。启动实例后便无法启用 AMD SEV-SNP。

启动已启用 AMD SEV-SNP 的实例

您不能使用 Amazon EC2 控制台启用 AMD SEV-SNP。

Amazon CLI

在启用了 AMD SEV-SNP 的情况下启动实例

使用带 `--cpu-options` 选项的 run-instances 命令。有关其他要求，请参阅 AMD SEV-SNP 要求。

```
--cpu-options AmdSevSnp=enabled
```

PowerShell

在启用了 AMD SEV-SNP 的情况下启动实例

将 New-EC2Instance cmdlet 与 `-CpuOption` 参数结合使用。

```
-CpuOption @{AmdSevSnp="enabled"}
```

检查 EC2 实例是否启用了 AMD SEV-SNP

您可以查找已启用 AMD SEV-SNP 的实例。Amazon EC2 控制台不显示此信息。

Amazon CLI

检查实例是否启用了 AMD SEV-SNP。

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].CpuOptions
```

下面是示例输出。如果 `AmdSevSnp` 中不存在 `CpuOptions`，则禁用 AMD SEV-SNP。

```
[
    {
        "AmdSevSnp": "enabled",
        "CoreCount": 1,
        "ThreadsPerCore": 2
    }
]
```

PowerShell

检查实例是否启用了 AMD SEV-SNP。

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.CpuOptions
```

下面是示例输出。如果不存在的 AmdSevSnp 值，则会禁用 AMD SEV-SNP。

```
AmdSevSnp CoreCount ThreadsPerCore
--------- --------- --------------
enabled   1         2
```

Amazon CloudTrail

实例启动请求出现 Amazon CloudTrail 事件时，以下属性将指示该实例已启用 AMD SEV-SNP。

```
"cpuOptions": {"AmdSevSnp": "enabled"}
```

使用 AMD SEV-SNP 证明 Amazon EC2 实例

证明是可让实例证明其状态和身份的过程。为实例启用 AMD SEV-SNP 后，您可以请求底层处理器提供 AMD SEV-SNP 证明报告。AMD SEV-SNP 证明报告包含初始客户机内存内容和初始 vCPU 状态的加密哈希值，称为启动指标。证明报告采用 VLEK 签名签署，该签名可回链至 AMD 信任根。您可以使用证明报告中包含的启动指标来验证实例是否在正版 AMD 环境中运行，并验证用于启动实例的初始启动代码。

先决条件

启动已启用 AMD SEV-SNP 的实例。有关更多信息，请参阅 为 EC2 实例启用 AMD SEV-SNP。

步数

- 第 1 步：获取证明报告
- 第 2 步：验证证明报告签名

第 1 步：获取证明报告

在这一步中，您需要安装和构建 snpguest 实用程序，然后使用其请求 AMD SEV-SNP 证明报告和证书。

1. 连接到您的实例。
2. 运行以下命令从 snpguest repository 构建 snpguest 实用程序。

```
$ git clone https://github.com/virtee/snpguest.git
$ cd snpguest
$ cargo build -r
$ cd target/release
```

3. 生成证明报告的请求。该实用程序从主机请求证明报告，并将其与提供的请求数据一起写入二进制文件。

   以下示例创建了一个随机请求字符串，并将其用作请求文件（request-file.txt）。命令返回证明报告后，该报告将存储在您指定的文件路径（report.bin）中。在这种情况下，实用程序会将报告存储在当前目录中。

```
$ ./snpguest report report.bin request-file.txt --random
```

4. 从主机内存请求证书，并将其存储为 PEM 文件。以下示例将文件存储在与 snpguest 实用程序相同的目录中。如果指定目录中已存在证书，则这些证书将被覆盖。

```
$ ./snpguest certificates PEM ./
```

第 2 步：验证证明报告签名

证明报告使用名为版本控制加载认可密钥（VLEK）的证书签署，该证书由 AMD 为 Amazon 颁发。在这一步中，您可以验证 VLEK 证书是否由 AMD 颁发，以及证明报告是否是用该 VLEK 证书签署的。

1. 从 AMD 官方网站将 VLEK 根信任证书下载到当前目录中。

```
$ sudo curl --proto '=https' --tlsv1.2 -sSf https://kdsintf.amd.com/vlek/v1/Milan/
cert_chain -o ./cert_chain.pem
```

2. 使用 openssl 验证 VLEK 证书是否是使用 AMD 根信任证书签署的。

```
$ sudo openssl verify --CAfile ./cert_chain.pem vlek.pem
```

下面是示例输出。

```
vlek.pem: OK
```

3. 使用 snpguest 实用程序来验证证明报告是否是用 VLEK 证书签署的。

```
$ ./snpguest verify attestation ./ report.bin
```

下面是示例输出。

```
Reported TCB Boot Loader from certificate matches the attestation report.
Reported TCB TEE from certificate matches the attestation report.
Reported TCB SNP from certificate matches the attestation report.
Reported TCB Microcode from certificate matches the attestation report.
VEK signed the Attestation Report!
```

Amazon EC2 Linux 实例的处理器状态控制

C 状态控制核心处于空闲状态时可以进入的睡眠级别。C 状态从 C0 (最浅空闲状态，此时核心完全唤醒并在执行指令) 开始编号，一直增进到 C6 (最深空闲状态，此时核心关闭)。

P 状态控制核心的所需性能（以 CPU 频率表示）。P 状态从 P0 (最高性能设置，此时核心可以使用 Intel 睿频加速技术提高频率) 开始编号，然后从 P1 (请求最大基准频率的 P 状态) 一直增加到 P15 (可能最低的频率)。

> **Note**
>
> Amazon Graviton 处理器具有内置的节能模式，并以固定频率运行。因此，它们不提供操作系统控制 C 状态和 P 状态的能力。

C 状态和 P 状态

以下实例类型为操作系统提供了控制 C 状态和 P 状态的功能：

- 通用型：m4.10xlarge | m4.16xlarge
- 计算优化型：c4.8xlarge
- 内存优化型：r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge
- 存储优化型：d2.8xlarge | i3.8xlarge | i3.16xlarge | h1.8xlarge | h1.16xlarge
- 加速计算型：g3.16xlarge | p3.16xlarge
- 裸机：所有搭载了 Intel 和 AMD 处理器的裸机实例

仅限 C 状态

以下实例类型为操作系统提供了控制 C 状态的功能：

- 通用型：m5.12xlarge | m5.24xlarge | m5d.12xlarge | m5d.24xlarge | m5n.12xlarge | m5n.24xlarge | m5dn.12xlarge | m5dn.24xlarge | m5zn.6xlarge | m5zn.12xlarge | m6a.24xlarge | m6a.48xlarge | m6i.16xlarge | m6i.32xlarge | m6id.16xlarge | m6id.32xlarge | m6idn.16xlarge | m6in.16xlarge | m6in.32xlarge | m7a.medium | m7a.large | m7a.xlarge | m7a.2xlarge | m7a.4xlarge | m7a.8xlarge | m7a.12xlarge | m7a.16xlarge | m7a.24xlarge | m7a.32xlarge | m7a.48xlarge | m7i.large | m7i.xlarge | m7i.2xlarge | m7i.4xlarge | m7i.8xlarge | m7i.12xlarge | m7i.16xlarge | m7i.24xlarge | m7i.48xlarge | m8a.medium | m8a.large | m8a.xlarge | m8a.2xlarge | m8a.4xlarge | m8a.8xlarge | m8a.12xlarge | m8a.16xlarge | m8a.24xlarge | m8a.48xlarge | m8azn.medium | m8azn.large | m8azn.xlarge | m8azn.3xlarge | m8azn.6xlarge | m8azn.12xlarge | m8azn.24xlarge | m8i.large | m8i.xlarge | m8i.2xlarge | m8i.4xlarge | m8i.8xlarge | m8i.12xlarge | m8i.16xlarge | m8i.24xlarge | m8i.32xlarge | m8i.48xlarge | m8i.96xlarge | m8id.large | m8id.xlarge | m8id.2xlarge | m8id.4xlarge | m8id.8xlarge | m8id.12xlarge | m8id.16xlarge | m8id.24xlarge | m8id.32xlarge | m8id.48xlarge | m8id.96xlarge | m8in.large | m8in.xlarge | m8in.2xlarge | m8in.4xlarge | m8in.8xlarge | m8in.12xlarge | m8in.16xlarge | m8in.24xlarge | m8in.32xlarge | m8in.48xlarge | m8in.96xlarge | m8idn.large | m8idn.xlarge | m8idn.2xlarge | m8idn.4xlarge | m8idn.8xlarge | m8idn.12xlarge | m8idn.16xlarge | m8idn.24xlarge | m8idn.32xlarge | m8idn.48xlarge | m8idn.96xlarge | m8ib.large

| m8ib.xlarge | m8ib.2xlarge | m8ib.4xlarge | m8ib.8xlarge | m8ib.12xlarge | m8ib.16xlarge | m8ib.24xlarge | m8ib.32xlarge | m8ib.48xlarge | m8ib.96xlarge | m8idb.large | m8idb.xlarge | m8idb.2xlarge | m8idb.4xlarge | m8idb.8xlarge | m8idb.12xlarge | m8idb.16xlarge | m8idb.24xlarge | m8idb.32xlarge | m8idb.48xlarge | m8idb.96xlarge | m8ine.large | m8ine.xlarge | m8ine.2xlarge | m8ine.4xlarge | m8ine.8xlarge | m8ine.12xlarge

- 计算优化型：c5.9xlarge | c5.12xlarge | c5.18xlarge | c5.24xlarge | c5a.24xlarge | c5ad.24xlarge | c5d.9xlarge | c5d.12xlarge | c5d.18xlarge | c5d.24xlarge | c5n.9xlarge | c5n.18xlarge | c6a.24xlarge | c6a.32xlarge | c6a.48xlarge | c6i.16xlarge | c6i.32xlarge | c6id.24xlarge | c6id.32xlarge | c6in.32xlarge | c7a.medium | c7a.large | c7a.xlarge | c7a.2xlarge | c7a.4xlarge | c7a.8xlarge | c7a.12xlarge | c7a.16xlarge | c7a.24xlarge | c7a.32xlarge | c7a.48xlarge | c7i.large | c7i.xlarge | c7i.2xlarge | c7i.4xlarge | c7i.8xlarge | c7i.12xlarge | c7i.16xlarge | c7i.24xlarge | c7i.48xlarge | c8a.medium | c8a.large | c8a.xlarge | c8a.2xlarge | c8a.4xlarge | c8a.8xlarge | c8a.12xlarge | c8a.16xlarge | c8a.24xlarge | c8a.48xlarge | c8i.large | c8i.xlarge | c8i.2xlarge | c8i.4xlarge | c8i.8xlarge | c8i.12xlarge | c8i.16xlarge | c8i.24xlarge | c8i.32xlarge | c8i.48xlarge | c8i.96xlarge | c8id.large | c8id.xlarge | c8id.2xlarge | c8id.4xlarge | c8id.8xlarge | c8id.12xlarge | c8id.16xlarge | c8id.24xlarge | c8id.32xlarge | c8id.48xlarge | c8id.96xlarge | c8in.large |c8in.xlarge | c8in.2xlarge | c8in.4xlarge |c8in.8xlarge | c8in.12xlarge | c8in.16xlarge | c8in.24xlarge | c8in.32xlarge | c8in.48xlarge |c8in.96xlarge | c8ib.large |c8ib.xlarge | c8ib.2xlarge | c8ib.4xlarge | c8ib.8xlarge | c8ib.12xlarge | c8ib.16xlarge |c8ib.24xlarge | c8ib.32xlarge | c8ib.48xlarge |c8ib.96xlarge | c8in.large |c8in.xlarge | c8in.2xlarge | c8in.4xlarge |c8in.8xlarge | c8in.12xlarge | x8i.large | x8i.xlarge | x8i.2xlarge | x8i.4xlarge | x8i.8xlarge | x8i.12xlarge | x8i.16xlarge | x8i.24xlarge | x8i.32xlarge | x8i.48xlarge | x8i.48xlarge | x8i.96xlarge
- 内存优化型：r5.12xlarge | r5.24xlarge | r5b.12xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge | r5dn.24xlarge | r6a.24xlarge | r6a.48xlarge | r6i.16xlarge | r6i.32xlarge | r6id.16xlarge | r6id.32xlarge | r6in.16xlarge | r6in.32xlarge | r7a.medium | r7a.large | r7a.xlarge | r7a.2xlarge | r7a.4xlarge | r7a.8xlarge | r7a.12xlarge | r7a.16xlarge | r7a.24xlarge | r7a.32xlarge | r7a.48xlarge | r7i.large | r7i.xlarge | r7i.2xlarge | r7i.4xlarge | r7i.8xlarge | r7i.12xlarge | r7i.16xlarge | r7i.24xlarge | r7i.48xlarge | r7iz.large | r7iz.xlarge |

r7iz.2xlarge | r7iz.4xlarge | r7iz.8xlarge | r7iz.12xlarge | r7iz.16xlarge | r7iz.32xlarge | r8a.medium | r8a.large | r8a.xlarge | r8a.2xlarge | r8a.4xlarge | r8a.8xlarge | r8a.12xlarge | r8a.16xlarge | r8a.24xlarge | r8a.48xlarge | r8i.large | r8i.xlarge | r8i.2xlarge | r8i.4xlarge | r8i.8xlarge | r8i.12xlarge | r8i.16xlarge | r8i.24xlarge | r8i.32xlarge | r8i.48xlarge | r8i.96xlarge | r8id.large | r8id.xlarge | r8id.2xlarge | r8id.4xlarge | r8id.8xlarge | r8id.12xlarge | r8id.16xlarge | r8id.24xlarge | r8id.32xlarge | r8id.48xlarge | r8id.96xlarge | r8in.large | r8in.xlarge | r8in.2xlarge | r8in.4xlarge | r8in.8xlarge | r8in.12xlarge | r8in.16xlarge | r8in.24xlarge | r8in.32xlarge | r8in.48xlarge | r8in.96xlarge | r8idn.large | r8idn.xlarge | r8idn.2xlarge | r8idn.4xlarge | r8idn.8xlarge | r8idn.12xlarge | r8idn.16xlarge | r8idn.24xlarge | r8idn.32xlarge | r8idn.48xlarge | r8idn.96xlarge | r8ib.large | r8ib.xlarge | r8ib.2xlarge | r8ib.4xlarge | r8ib.8xlarge | r8ib.12xlarge | r8ib.16xlarge | r8ib.24xlarge | r8ib.32xlarge | r8ib.48xlarge | r8ib.96xlarge | r8idb.large | r8idb.xlarge | r8idb.2xlarge | r8idb.4xlarge | r8idb.8xlarge | r8idb.12xlarge | r8idb.16xlarge | r8idb.24xlarge | r8idb.32xlarge | r8idb.48xlarge | r8idb.96xlarge | u-3tb1.56xlarge | u-6tb1.56xlarge | u-6tb1.112xlarge | u-9tb1.112xlarge | u-12tb1.112xlarge | u-18tb1.112xlarge | u-24tb1.112xlarge | u7i-6tb.112xlarge | u7i-8tb.112xlarge | u7i-12tb.224xlarge | u7in-16tb.224xlarge | u7in-24tb.224xlarge | u7in-32tb.224xlarge | u7inh-32tb.480xlarge | x2idn.32xlarge | x2iedn.16xlarge | x2iedn.32xlarge | x2iezn.12xlarge | x8aedz.large | x8aedz.xlarge | x8aedz.3xlarge | x8aedz.6xlarge | x8aedz.12xlarge | x8aedz.24xlarge | z1d.6xlarge | z1d.12xlarge

- 存储优化型：d3en.12xlarge | dl1.24xlarge | i3en.12xlarge | i3en.24xlarge | i4i.16xlarge | i7i.large | i7i.xlarge | i7i.2xlarge | i7i.4xlarge | i7i.8xlarge | i7i.12xlarge | i7i.16xlarge | i7i.24xlarge | i7i.48xlarge | i7ie.large | i7ie.xlarge | i7ie.2xlarge | i7ie.3xlarge | i7ie.6xlarge | i7ie.12xlarge | i7ie.18xlarge | i7ie.24xlarge | i7ie.48xlarge | r5b.12xlarge | r5b.24xlarge
- 加速计算型：dl1.24xlarge | f2.6xlarge | f2.12xlarge | f2.48xlarge | g5.24xlarge | g5.48xlarge | g6.24xlarge | g6.48xlarge | g6e.12xlarge | g6e.24xlarge | g6e.48xlarge | g7e.2xlarge | g7e.4xlarge | g7e.8xlarge | g7e.12xlarge | g7e.24xlarge | g7e.48xlarge | inf1.24xlarge | p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | p5.48xlarge | p5e.48xlarge | p5en.48xlarge | p6-b200.48xlarge | p6-b300.48xlarge | trn1.32xlarge | trn2.3xlarge | trn2.48xlarge | trn2a.3xlarge | trn2a.48xlarge | trn2n.3xlarge | trn2n.48xlarge | trn2p.48xlarge | trn2u.48xlarge | vt1.24xlarge

- 高性能计算：hpc7a.12xlarge | hpc7a.24xlarge | hpc7a.48xlarge | hpc7a.96xlarge | hpc8a.96xlarge

改变 C 状态或 P 状态设置可以增加处理器性能一致性，减少延迟，还可以针对特定工作负载对实例进行调校。默认 C 状态和 P 状态设置可提供最大性能，是大多数工作负载的最佳选择。但是，如果您的应用程序更适合以牺牲较高的单核或双核频率的方式来降低延迟，或需要在较低频率下保持稳定性能(而不适合使用突发式睿频加速频率)，那么可以考虑运用对这些实例可用的 C 状态或 P 状态设置。

有关不同处理器配置以及如何监控 Amazon Linux 配置影响的信息，请参阅《Amazon Linux 2 User Guide》中的 Processor state control for Amazon EC2 Amazon Linux instance。这些步骤专为 Amazon Linux 编写并供其使用，但也适用于搭载 Linux 内核 3.9 及更高版本的其他 Linux 发行版。有关其他 Linux 发行版和处理器状态控制的更多信息，请参阅您系统的特定文档。

Amazon EC2 托管式实例

Amazon EC2 托管式实例是指由指定服务提供商（例如 Amazon EKS 通过 EKS 自动模式）预置和管理的 EC2 实例。托管式实例允许将实例的操作控制权委派给服务提供商，由此简化在 Amazon EC2 上运行计算工作负载的方式。

委派控制权是为托管式实例引入的唯一变更。技术规格和计费方面与非托管 EC2 实例保持一致。由于托管式实例允许将控制权委派给服务提供商，您可以从服务提供商的运营专业知识和最佳实践中受益。对于托管式实例，服务提供商会负责执行预置实例、配置软件、扩缩容量、处理实例故障和替换以及终止实例等任务。

不能直接修改托管式实例的设置或将其终止。服务及具体操作由您与服务提供商之间的协议决定。不过，您可以在托管式实例中添加、修改或删除标签，以便在自己的 Amazon 环境中对实例进行分类。

内容

- 托管式实例的账单
- 确定托管式实例
- 托管资源可见性设置
- 开始使用托管式实例

托管式实例的账单

Amazon EC2 托管式实例的基本费用与非托管 Amazon EC2 实例的相同，但需另外向服务提供商支付一笔费用。这笔额外费用单独计费，由管理您实例的服务提供商收取。范围涵盖了托管式实例操作和维护服务所需的费用。

所有 Amazon EC2 购买选项都适用于托管式实例，包括按需型实例、预留实例、竞价型实例和节省计划。通过将从 EC2 中直接获取的计算实例提供给服务提供商，您可以从适用于自己账户的任何现有预留实例或节省计划中受益，确保自己使用的是最具成本效益的可用计算容量。

例如，在使用 Amazon EKS 自动模式时，您需要为基础实例支付一笔标准 EC2 实例费用，还要就 Amazon EKS 代为管理实例支付一笔额外费用。如果之后决定注册节省计划，EC2 实例费用会因节省计划而减少，但 Amazon EKS 的额外费用保持不变。

确定托管式实例

托管式实例由托管字段中的 true 进行标识。服务提供商由（控制台中的）运营商字段或（CLI 中的）`Principal` 字段标识。

按照以下过程确定托管式实例。

Console

确定托管式实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择要检查的实例。
4. 在详细信息选项卡（如果选中了复选框）或摘要区域（如果选择了实例 ID）中，找到托管字段。
   - 值为 true 表示托管式实例。
   - 值为 false 表示非托管式实例。
5. 如果将托管设置为 true，则运营商字段会显示一个值，用于标识负责管理实例的服务提供商。例如，值为 eks.amazonaws.com 表示将 Amazon EKS 标识为服务提供商。

Amazon CLI

确定托管式实例

使用 describe-instances 命令并指定实例 ID。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].Operator
```

下面是示例输出。如果 Managed 为 true，表示该实例属于托管式实例并且包含一个 Principal。主体即为管理实例的服务提供商。例如，值为 eks.amazonaws.com 表示将 Amazon EKS 标识为服务提供商。

```
[
    {
        "Managed": true,
        "Principal": "eks.amazonaws.com"
    }
]
```

查找托管式实例

使用 describe-instances 命令并指定值为 true 的 operator.managed 筛选条件。--query 选项用于仅显示托管式实例的 ID。

```
aws ec2 describe-instances \
    --filters "Name=operator.managed,Values=true" \
    --query Reservations[*].Instances[].InstanceId
```

PowerShell

确定托管式实例

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.Operator
```

下面是示例输出。

```
Managed Principal
------- ---------
True    eks.amazonaws.com
```

查找托管式实例

使用 Get-EC2Instance cmdlet。此示例仅显示托管式实例的 ID。

```
(Get-EC2Instance -Filter @{Name="operator.managed";
 Values="true"}).Instances.InstanceId
```

托管资源可见性设置

您可以控制 Amazon 服务代表您预置的资源是否显示在 Amazon EC2 控制台视图及 API 列表操作中。

什么是托管资源可见性？

诸如 Amazon EKS、Amazon ECS、WorkSpaces Core和 Amazon Lambda 等 Amazon 服务可在您账户内直接预置和运行 Amazon EC2 实例。这些服务负责扩展、操作系统补丁、安全更新以及生命周期管理。生成的 Amazon EC2 实例、Amazon EC2 启动模板、Amazon EBS 卷和网络接口（ENI）将与您的客户自主管理型资源一同显示在 Amazon EC2 控制台和 API 中。托管资源可见性设置可让您控制这些托管资源是否会在资源视图中显示。

受影响的资源类型

资源类型	预置这些资源的服务	说明
Amazon EC2 实例	Amazon EKS 工作节点、Amazon ECS 容器实例、Amazon Lambda 执行环境、Amazon WorkSpaces Core	受可见性设置影响的主要资源类型
Amazon EC2 启动模板	Amazon EKS、Amazon ECS、Amazon WorkSpaces Core	启动由托管服务创建的模板
Amazon EBS 卷	Amazon EKS、Amazon ECS、Amazon WorkSpaces Core	附加到托管实例的卷
网络接口（ENI）	Amazon EKS、Amazon ECS、Amazon Lambda、Amazon WorkSpaces Core	针对托管工作负载预置的网络接口

> **Note**
>
> 默认情况下，对于在可见性设置可用之前未拥有托管资源的账户，Amazon EC2 会隐藏其托管资源。对于已拥有托管资源的账户，Amazon EC2 会将可见性设置为可见，以保留现有工作流程。可见性设置适用于一个区域中的所有托管资源，无论其创建时间为何。您可以随时更改可见性设置。

为何要配置可见性设置

通过配置可见性设置，您可以自定义托管资源在操作工具中的显示方式。常见使用案例包括：

- 将合规性仪表板中的资源数量缩减至仅包含客户自主管理型资源，从而简化治理。
- 减少可观测性工具中的噪音，这些工具汇总账户内所有实例的 Amazon EC2 指标。
- 防止云安全态势管理（CSPM）扫描器（例如 Qualys）将托管资源误判为客户配置错误，从而避免误报。
- 对于托管实例，Amazon 负责 Amazon EC2 实例的配置、补丁更新和运行状况。通过控制可见性，您可以更清晰地向最终用户阐明责任共担模式。

> **Note**
>
> 可见性设置控制资源在 Amazon 控制台视图和 API 列表操作中的显示。这些设置不会影响计费、资源操作或实际访问权限。隐藏资源仍可正常运行并计费。

配置托管资源可见性

您可以通过 Amazon EC2 控制台或 Amazon CLI 来配置托管资源的可见性。

Console

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡片的设置下，选择托管资源。
4. 选择 Modify(修改)。
5. 在默认可见性下，选择下列选项之一：

- 隐藏（默认）：隐藏所有托管资源。
- 可见：显示所有托管资源。

6. 选择修改可见性。

Amazon CLI

获取当前的可见性设置

使用 get-managed-resource-visibility 命令检索当前的可见性配置：

```
aws ec2 get-managed-resource-visibility
```

示例响应：

```
{
    "visibility": {
        "defaultVisibility": "hidden"
    }
}
```

隐藏所有托管资源

使用 modify-managed-resource-visibility 命令隐藏所有托管资源，无论操作员是谁：

```
aws ec2 modify-managed-resource-visibility \
    --default-visibility "hidden"
```

发现隐藏的托管资源

关闭可见性后，仍可访问托管资源。以下方法可按需显示这些资源：

1. 特定服务的控制台：导航到相应的 Amazon 服务控制台（例如，Amazon EKS 控制台），查看为该服务预置的实例。服务控制台提供该服务在您账户中管理的所有资源的完整详细信息。
2. 直接 API 查询：使用带有特定 `instance-id` 参数的 `describe-instances` API。无论可见性设置如何，带已知实例 ID 的直接查询都会返回结果。可见性设置仅影响列表和筛选操作。您还可以使用带 `include-managed-resources` 参数的 `describe-instances`，以发现托管实例。

> **Note**
>
> 相同的 direct-query-by-ID 行为也适用于所有受影响的资源类型。您可以使用 `describe-volumes`、`describe-launch-templates` 以及带特定资源 ID 的 `describe-network-interfaces`，以访问这些类型的隐藏托管资源。

计费注意事项

托管资源可见性设置不会影响计费。隐藏的托管实例仍会出现在计费数据中，因为这些实例是在您账户内运行、代表您进行预置的资源，且无论可见性配置如何，均需要保持全额计费。

隐藏资源在以下情况下仍可见：

- Amazon 账单
- Amazon 成本和使用情况报告

> **Important**
>
> 托管实例会在您的账户中进行预置，并消耗计算资源。将这些实例隐藏在控制台视图中并不能降低成本。有关托管实例费用的详细信息，请查看特定服务的计费文档（例如 Amazon EKS 定价、Amazon ECS 定价）。

限制

- 可见性设置适用于整个账户，并统一影响所有 IAM 主体。
- 您无法按资源类型或创建托管资源的服务而选择性地显示或隐藏托管资源。例如，您无法选择显示由 Amazon EKS 创建的托管实例，同时隐藏由 Lambda、Amazon ECS 或 Amazon WorkSpaces 创建的托管实例。

开始使用托管式实例

有关使用托管式实例的指南，请参阅《Amazon EKS User Guide》中的 Automate cluster infrastructure with EKS Auto Mode。

利用嵌套虚拟化在 Amazon EC2 实例中运行虚拟机监控程序

使用嵌套虚拟化，您可以在虚拟 Amazon EC2 实例内运行虚拟机监控程序，例如 Hyper-V 和 KVM。虚拟的 EC2 实例是非裸机实例。此功能通过为虚拟的 EC2 实例添加处理器级虚拟化支持，增强了虚拟化的灵活性，使得运行在实例中的虚拟机监控程序能够创建和管理虚拟机。

当您在开发工作流程中运行 Docker Desktop、Windows Subsystem for Linux 2（WSL2）、Android Studio 模拟器或 QEMU 等开发工具时，采用嵌套虚拟化技术会非常有用，因为它能让您从众多符合您特定性能和价格要求的标准 Amazon EC2 虚拟实例类型中进行选择。

使用嵌套虚拟化技术无需额外付费。

内容

- 工作原理
- 注意事项
- 在启用了嵌套虚拟化的情况下启动新实例
- 将现有实例配置为使用嵌套虚拟化

工作原理

虚拟 EC2 实例在装有 Nitro 虚拟机监控器的物理主机上运行。为了支持嵌套虚拟化，Nitro System 会将处理器扩展程序（例如 Intel VT-x）传递给各个实例，以方便运行嵌套虚拟机。嵌套虚拟化架构由三层组成：物理 Amazon 基础设施和 Nitro 虚拟机监控器（L0）、运行虚拟机监控程序的 EC2 实例（L1），以及在该实例中创建的一台或多台虚拟机（L2）。

注意事项

在开始使用嵌套虚拟化之前，请考虑以下几点：

- 支持的实例类型：C8i、M8i 和 R8i 实例目前支持嵌套虚拟化。
- 支持的虚拟机监控程序：目前，KVM 和 Hyper-V 是所支持的 L1 虚拟化监控程序。
- Windows 实例：在 Windows 实例上启用嵌套虚拟化时：
  - 休眠：不支持实例休眠和恢复。
  - CPU 限制：在 CPU 超过 192 的 Windows 实例上不受支持，例如 `m8i.96xl`。
- 安全责任：在 EC2 实例上使用嵌套虚拟化时，Amazon 负责“云的安全”、保护底层基础设施并维护 Amazon Nitro System 提供的 EC2 实例之间的坚固隔离边界。客户负责“云中安全”，包括保护 EC2 实例中的操作系统、虚拟机监控程序、嵌套虚拟机、访客操作系统、应用程序和数据的安全。

- 性能：Amazon 建议那些希望运行需要使用硬件虚拟化扩展程序的工作负载、并且对性能有较高要求或者对延迟有严格限制的客户，来评估裸机实例。

在启用了嵌套虚拟化的情况下启动新实例

启动新实例时，您可以开启嵌套虚拟化功能，以便在此实例上运行虚拟机监控程序和虚拟机。

先决条件

您必须拥有启动 Amazon EC2 实例所需的 IAM 权限。

Console

要在实例启动期间启用嵌套虚拟化

1. 按照 the section called “使用启动实例向导启动” 程序进行操作并根据需要配置实例。
2. 确保选择了支持的实例类型。
3. 展开高级详细信息，对于嵌套虚拟化，请选择启用。
4. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。

Amazon CLI

要在启用嵌套虚拟化的情况下启动实例

可以使用 run-instances 命令。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type r8i.4xlarge \
    --cpu-options "NestedVirtualization=enabled" \
    --key-name my-key-pair
```

PowerShell

要在启用嵌套虚拟化的情况下启动实例

使用 New-EC2Instance 命令。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType r8i.4xlarge `
```

```
    -CpuOption @{NestedVirtualization='enabled'} `
    -KeyName my-key-pair
```

将现有实例配置为使用嵌套虚拟化

您可以在现有 Amazon EC2 实例上开启嵌套虚拟化。

先决条件

- 该实例必须处于 stopped 状态。
- 该实例类型必须支持嵌套虚拟化。

Console

要在现有实例上启用嵌套虚拟化

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 从实例表中选择要修改的实例。
4. 选择操作、实例设置、更改 CPU 选项。
5. 在更改 CPU 选项页面上，对于嵌套虚拟化，选择以下选项之一：
   - 启用：为实例开启嵌套虚拟化
   - 禁用：为实例关闭嵌套虚拟化
6. 查看您的更改，然后选择更改，以应用新的 CPU 选项。

Amazon CLI

要在现有实例上启用嵌套虚拟化

首先停止实例，然后使用 modify-instance-cpu-options 命令。

```
aws ec2 modify-instance-cpu-options \
    --instance-id i-1234567890abcdef0 \
    --core-count 4 \
    --threads-per-core 2 \
    --nested-virtualization enabled
```

PowerShell

要在现有实例上启用嵌套虚拟化

首先停止实例，然后使用 Edit-EC2InstanceCpuOption 命令。

```
Edit-EC2InstanceCpuOption `
    -InstanceId i-1234567890abcdef0 `
    -CoreCount 4 `
    -ThreadsPerCore 2 `
    -NestedVirtualization enabled
```

Amazon EC2 账单和购买选项

您可以使用以下选项优化 Amazon EC2 的成本：

- 按需型实例：按秒为启动的实例付费。
- 预留实例：通过承诺在 1 年或 3 年期限内提供一致的实例配置（包括实例类型和区域）来降低您的 Amazon EC2 成本。
- 竞价型实例：请求未使用的 EC2 实例，这可能会显著降低您的 Amazon EC2 成本。
- 专属主机：为完全专用于运行您的实例的物理主机付费，让您现有的按插槽、按内核或按 VM 计费的软件许可证降低成本。
- 专用实例：为在单一租户硬件上运行的实例按小时付费。
- 容量预留 – 可在特定可用区中为 EC2 实例预留容量。

如果您需要预留容量，请为特定可用区购买预留实例或预留容量。可使用容量块预留 GPU 实例集群。如果能灵活控制应用程序的运行时间并且您的应用程序可以中断，竞价型实例就是经济实惠之选。专用主机或专用实例可以通过使用现有服务器端绑定软件许可证来满足合规性要求和降低成本。

有关更多信息，请参阅 Amazon EC2 定价和 Amazon EC2 托管式实例。

为 Amazon EC2 购买按需型实例

使用按需型实例，您按秒为计算容量支付费用，无需长期订阅。实例的生命周期完全由您控制 – 您决定何时启动、停止、休眠、开机、重启或终止实例。

购买按需型实例没有长期承诺。您只需要为处于 `running` 状态的按需型实例的秒数付费，最少 60 秒。运行中的个按需型实例的每秒价格是固定的，上列出了此价格。

我们建议您为短期的不规则且不能中断的应用程序使用按需型实例。

目录

- 按需型实例限额
  - 监控按需型实例限额和使用情况
  - 请求提高限额
- 查询按需型实例的价格

按需型实例限额

每个 Amazon Web Services 账户可在每个区域运行的按需型实例数量存在限额。按需型实例的限额根据运行的按需型实例所用虚拟中央处理器（vCPU）数进行管理，而无论为哪种实例类型。每种限额类型都规定了一个或多个实例系列的最大 vCPUs 数。

您的账户包含以下按需型实例配额。处于待处理、正在停止、已停止和已休眠状态的实例不计入按需型实例限额。容量预留会计入按需型实例限额，即使未使用也是如此。

名称	默认值	可调整
正在运行的按需 DL 实例	0	是
正在运行的按需 F 实例	0	是
正在运行的按需 G 和 VT 实例	0	是
正在运行的按需型 HPC 实例	0	是
正在运行按需内存增强型实例	0	是
正在运行的按需 Inf 实例	0	是
正在运行的按需 P 实例	0	是
正在运行的按需标准（A、C、D、H、I、M、R、T、Z）实例	5	是
正在运行的按需型 Trn 实例	0	是
正在运行的按需 X 实例	0	是

有关不同实例系列、实例代和大小的信息，请参阅《Amazon EC2 Instance Types Guide》https://docs.amazonaws.cn/ec2/latest/instancetypes/instance-types.html。

只要 vCPU 数不超过账户的限额，您就可以启动任意实例类型组合来满足您不断变化的应用程序需求。例如，对于 256 个 vCPU 的标准实例限额，您可以启动 32 个 `m5.2xlarge` 实例（32 x 8 vCPU）或 16 个 `c5.4xlarge` 实例（16 x 16 vCPU）。有关更多信息，请参阅 EC2 个按需实例限制。

任务

- 监控按需型实例限额和使用情况
- 请求提高限额

监控按需型实例限额和使用情况

您可以使用以下方法查看和管理您在每个区域的按需型实例限额。

使用服务限额控制台查看当前限额

1. 访问 https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/，打开 Service Quotas 控制台。
2. 从导航栏中选择一个区域。
3. 在筛选条件字段中输入 **`On-Demand`**。
4. 应用的限额值列会显示账户中每种按需型实例限额类型的最大 vCPU 数量。

使用 Amazon Trusted Advisor 控制台查看当前限额

打开 Amazon Trusted Advisor 控制台中的服务限制页面。

配置 CloudWatch 告警

借助 Amazon CloudWatch 指标集成，您可以根据限额监控 EC2 使用情况。您还可以配置警报以在即将达到限额时发出警告。有关更多信息，请参阅《服务限额用户指南》中的 服务限额与 Amazon CloudWatch 警报。

请求提高限额

尽管 Amazon EC2 会根据您的使用情况自动提高按需型实例限额，您仍可以在必要时请求提高限额。例如，假设您计划启动超过当前限额允许数量的实例，则可以按照 Amazon EC2 Service Quotas 中的描述，在服务限额控制台中使用请求提高限额。

查询按需型实例的价格

可以使用价目表服务 API 或 Amazon 价目表 API 查询按需型实例的价格。有关更多信息，请参阅 Amazon Billing 用户指南中的使用Amazon价目表 API。

Amazon EC2 的预留实例概览

> **Important**
>
> 我们建议使用节省计划，而不是预留实例。节省计划是节省 Amazon 计算成本的最简单、最灵活方式，与预留实例一样，提供更低的价格（相比按需定价最高可省 72%）。但是，节省计划与预留实例不同。使用预留实例时，您将承诺使用特定的实例配置，而使用节省计划时，您可以灵活地使用最能满足您需求的实例配置。使用节省计划时，您将承诺保持一致的使用量（以美元/小时为单位）。有关更多信息，请参阅 Amazon Savings Plans 用户指南。

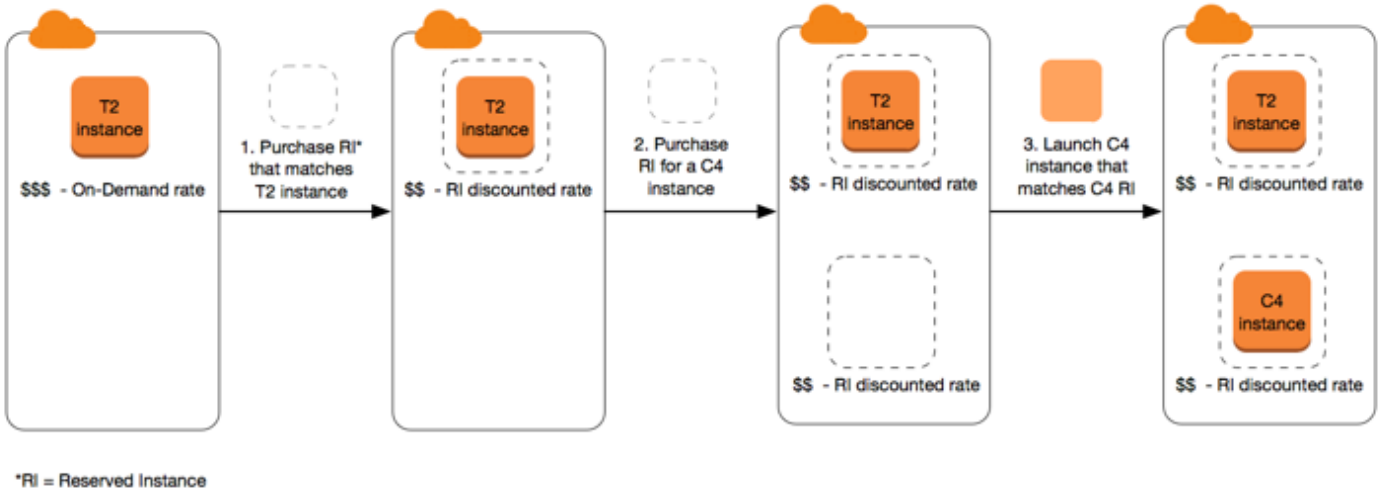
相比按需型实例定价，预留实例可大幅节约您的 Amazon EC2 成本。预留实例不是物理实例，而是对账户中使用的按需型实例所应用的账单折扣。这些按需型实例必须与特定属性（例如实例类型和区域）匹配才能享受账单折扣。

预留实例主题

- 预留实例示例场景
- 决定Reserved Instance定价的关键变量
- 区域和可用区预留实例（范围）
- 预留实例的类型（提供的类别）
- 如何应用预留实例折扣
- 使用您的预留实例
- 预留实例的计费方式
- 购买 Amazon EC2 的预留实例
- 在预留实例市场中销售 Amazon EC2 的预留实例
- 修改 预留实例
- 交换 可转换预留实例
- 预留实例限额

预留实例示例场景

下图是购买和使用预留实例的基本场景。



在此场景中，您的账户中有一个正在运行的个按需型实例 (T2)，当前您按照按需费率支付。您购买了一个与您正在运行的实例的属性相匹配的Reserved Instance，账单优势立即体现。接下来，您为 C4 实例购买一个Reserved Instance。您的账户中没有任何正在运行的实例与此Reserved Instance的属性相匹配。在最后的步骤中，您启动了一个与 C4 Reserved Instance的属性相匹配的实例，账单优势立即体现。

决定Reserved Instance定价的关键变量

Reserved Instance 定价由以下关键变量决定。

实例属性

预留实例有四个决定其价格的实例属性。

- 实例类型：例如，`m4.large`。这由实例系列（例如，`m4`）和实例大小（例如，`large`）组成。
- 区域：购买Reserved Instance的区域。
- 租赁：您的实例在共享 (默认) 还是单租户 (专用) 硬件上运行。有关更多信息，请参阅Amazon EC2 专用实例。
- 平台：操作系统；例如，Windows 或 Linux/Unix。有关更多信息，请参阅选择平台。

期限承诺

您可以承诺购买一年或三年的 Reserved Instance，三年承诺可以获得更大的折扣。

- 一年：一年定义为 31536000 秒（365 天）。
- 三年：三年定义为 94608000 秒 (1095 天)。

预留实例不会自动续订；当它们过期时，可以继续使用 EC2 实例而不会中断，但要支付按需费率。在上面的示例中，当涵盖 T2 和 C4 实例的 预留实例 过期后，将改为以按需费率支付，直至终止这些实例或者购买与实例属性相匹配的新 预留实例。

> **Important**
>
> 购买Reserved Instance后，您将不能取消您的购买。但是，如果您需要更改，则可以修改、交换或出售您的 Reserved Instance。

付款选项

针对 预留实例 可使用以下付款选项：

- 预付全费：所有款项于期限开始时支付，无论使用了多少小时数，剩余期限不会再产生其他任何费用或额外按小时计算的费用。
- 预付部分费用：必须预付部分费用，无论是否使用了 Reserved Instance，期限内剩余的小时数都将按照打折小时费率计费。
- 无预付费用：无论是否使用 Reserved Instance，您都将按照期限内的小时数，采用打折小时费率进行付费。无需预付款。

> **Note**
>
> 在整个预留期限内，“无预付费用”预留实例需要根据合同义务每月支付费用。因此，账户需要具有成功的账单历史记录才能购买“无预付”预留实例。

一般而言，通过为 预留实例 支付选较高的预付款，可以节省更多成本。您也可以在预留实例 Marketplace 中找到由第三方卖家以短期低价提供的预留实例。有关更多信息，请参阅在预留实例市场中销售 Amazon EC2 的预留实例。

优惠类别

在计算需求发生变化时，您可以根据产品类别修改或交换 Reserved Instance。

- 标准：这些提供最大力度的折扣，但只可以修改。标准 预留实例 无法交换。
- 可转换：这些相较于标准 预留实例 提供较低的折扣，但可以与具有不同实例属性的可转换 Reserved Instance 进行交换。可转换 预留实例 也可修改。

有关更多信息，请参阅预留实例的类型（提供的类别）。

> **Important**
>
> 购买Reserved Instance后，您将不能取消您的购买。但是，如果您需要更改，则可以修改、交换或出售您的 Reserved Instance。

有关更多信息，请参阅。

区域和可用区预留实例（范围）

当您购买 Reserved Instance 时，可决定 Reserved Instance 的范围。范围可以是区域或可用区。

- 区域：当您购买某个区域的 Reserved Instance，该实例称为区域性 Reserved Instance。
- 可用区：当您购买特定可用区的 Reserved Instance 时，该实例称为可用区 Reserved Instance。

范围不影响价格。您为区域或可用区 Reserved Instance 支付相同的价格。有关 Reserved Instance 定价的更多信息，请参阅决定Reserved Instance定价的关键变量和Amazon EC2 预留实例定价。

有关如何指定预留实例范围的更多信息，请参阅 RI 属性，特别是可用区小点。

区域性或可用区预留实例之间的差异

下表重点介绍了区域性 预留实例 和可用区 预留实例 之间的一些主要区别：

	区域性预留实例	可用区预留实例
是否能够预留容量	区域性Reserved Instance无法预留容量。	可用区Reserved Instance可在指定可用区中预留容量。
可用区灵活性		

	区域性预留实例	可用区预留实例
	Reserved Instance 折扣适用于指定区域的任何可用区中的实例使用。	无可用区灵活性 — Reserved Instance 折扣仅适用于指定可用区中的实例使用。
实例大小灵活性	Reserved Instance 折扣适用于实例系列中的实例使用，无论实例大小如何。 只在具有默认租期的 Amazon Linux/Unix 预留实例 上受支持。有关更多信息，请参阅实例大小灵活性由标准化因子决定。	无实例大小灵活性 — Reserved Instance 折扣仅适用于指定实例类型和大小的实例使用。
排队购买	您可以排队购买区域预留实例。	您不能将可用区预留实例的购买排队。

有关更多信息以及示例，请参阅 如何应用预留实例折扣。

预留实例的类型（提供的类别）

Reserved Instance 的产品类别是标准或可转换。标准 Reserved Instance 提供的折扣比可转换 Reserved Instance 更多，但是您无法交换标准 Reserved Instance。您可以交换可转换 预留实例。您可以修改标准版和可转换 预留实例。

Reserved Instance 的配置包括单个实例类型、平台、范围和租期。在计算需求发生变化时，您可以修改或交换 Reserved Instance。

标准和可转换 预留实例 的差异

下面是标准和可转换 预留实例 之间的差别。

	标准Reserved Instance	Convertible Reserved Instance
修改 预留实例	有些属性可以修改。有关更多信息，请参阅修改 预留实例。	有些属性可以修改。有关更多信息，请参阅 修改 预留实例。

	标准Reserved Instance	Convertible Reserved Instance
交换预留实例	无法交换。	在期限内可以与具有新属性（包括实例系列、实例类型、平台、范围或租期）的其他 可转换预留实例 进行交换。有关更多信息，请参阅 交换 可转换预留实例。
在预留实例 Marketplace 中销售	可以在预留实例 Marketplace 中销售。	无法在预留实例 Marketplace 中销售。
在预留实例 Marketplace 中购买	可以在预留实例 Marketplace 中购买。	无法在预留实例 Marketplace 中购买。

如何应用预留实例折扣

预留实例不是物理实例，而是对账户中正在运行的按需型实例所应用的账单折扣。这些按需型实例必须与预留实例的特定规格匹配，才能享受账单折扣。

如果您购买了预留实例并且已经有正在运行的按需型实例与预留实例的规范匹配，则账单折扣将立即自动应用。您不必重启您的实例。若没有符合条件的正在运行的按需型实例，则请启动与您的预留实例具有相同规格的按需型实例。有关更多信息，请参阅 使用您的预留实例。

预留实例的优惠类别（标准或可转换）不会影响账单折扣的应用方式。

主题

- 如何应用可用区预留实例
- 如何应用区域性预留实例
- 实例大小灵活性
- 应用预留实例的示例

如何应用可用区预留实例

为特定可用区中的预留容量而购买的预留实例称为地区性预留实例。

- 预留实例折扣适用于该可用区中匹配的实例使用。

- 正在运行的实例的属性 (租期、平台、可用区、实例类型和实例大小) 必须与预留实例的属性匹配。

例如，如果购买可用区 us-east-1a 中的两个 `c4.xlarge` 默认租期 Linux/Unix 标准预留实例，则可用区 us-east-1a 中最多两个正在运行的 `c4.xlarge` 默认租赁 Linux/Unix 实例可享受预留实例折扣。

如何应用区域性预留实例

为区域购买的预留实例称为区域性预留实例，可提供可用区和实例大小的灵活性。

- Reserved Instance 折扣适用于该区域的任何可用区中的实例使用。
- 预留实例折扣适用于实例系列中的实例使用，无论实例大小如何，这称为实例大小灵活性。

实例大小灵活性

凭借实例大小的灵活性，预留实例折扣适用于对同一系列实例的使用量。预留实例基于标准化因子应用于实例系列中各种规模的实例（从最小到最大）。有关如何应用预留实例折扣的示例，请参阅 场景 2：使用标准化因素在单个账户中使用预留实例。

限制

- 支持：仅区域性预留实例支持实例大小灵活性。
- 不支持：以下预留实例不支持实例大小灵活性：
  - 针对特定可用区购买的 预留实例（可用区 预留实例）
  - G4ad、G4dn、G5、G5g、G6、G6e、G6f、Gr6、Gr6f、hpc7a、P5、Inf1、Inf2、u7i-6tb 和 u7i-8tb 实例的预留实例
  - 适用于 Windows Server、装有 SQL Standard 的 Windows Server、装有 SQL Server Enterprise 的 Windows Server、装有 SQL Server Web 的 Windows Server、RHEL 和 SUSE Linux Enterprise Server 的 预留实例
  - 使用专用租赁的预留实例

实例大小灵活性由标准化因子决定

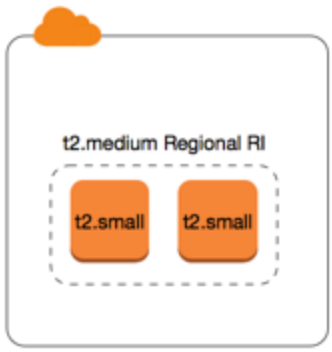
实例大小灵活性取决于实例大小的标准化因子。根据预留的实例大小，区域中的任何可用区中的相同实例系列的运行实例将享受全部或部分折扣。必须匹配的属性仅为实例系列、租期和平台。

下表列出了实例系列中的各种大小以及相应的标准化因子。这种比例用于将 预留实例 的折扣费率应用于实例系列的标准化使用。

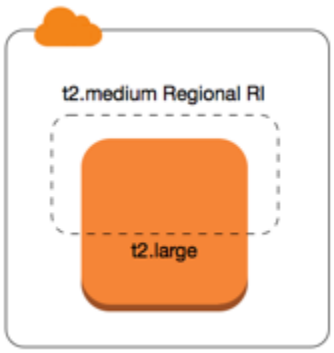
实例大小	标准化因子
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448

实例大小	标准化因子
112xlarge	896

例如，t2.medium 实例具有标准化因子 2。如果您在 US East (N. Virginia) 中购买了 t2.medium 默认租期 Amazon Linux/Unix Reserved Instance，并且您的账户在该区域中有两个正在运行的 t2.small 实例，则账单优势应用于全部两个实例。



或者，如果您的账户在 US East (N. Virginia) 区域有一个 t2.large 实例，则账单优势应用到 50% 的实例使用。



在修改预留实例时，标准化因子也适用。有关更多信息，请参阅修改 预留实例。

裸机实例的标准化因子

实例大小灵活性也适用于实例系列中的裸机实例。如果您具有区域性 Amazon Linux/Unix 预留实例 并对裸机实例使用共享租期，则可以获得在相同实例系列中节省 Reserved Instance 的好处。反过来也是如此：如果您具有区域性 Amazon Linux/Unix 预留实例 并对与裸机实例相同的系列中的实例使用共享租期，则可以获得在裸机实例中节省 Reserved Instance 的好处。

metal 实例大小的标准化因子不是单一的。裸机实例的标准化因子与同一实例系列中的等效虚拟化实例大小相同。例如，i3.metal 实例与 i3.16xlarge 实例具有相同的标准化因子。

实例大小	标准化因子
a1.metal	32
m5zn.metal | x2iezn.metal z1d.metal	96
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

例如，i3.metal 实例的标准化因子为 128。如果您购买 US East (N. Virginia) 中的 i3.metal 默认租期 Amazon Linux/Unix Reserved Instance，则可以获得如下的账单优惠：

- 如果您在该区域的账户中有一个正在运行的 i3.16xlarge，则账单优惠全部应用于 i3.16xlarge 实例（i3.16xlarge 标准化因子 = 128）。
- 或者，如果您在该区域的账户中有两个正在运行的 i3.8xlarge 实例，则账单优惠全部应用于这两个 i3.8xlarge 实例（i3.8xlarge 标准化因子 = 64）。
- 或者，如果您在该区域的账户中有四个正在运行的 i3.4xlarge 实例，则账单优惠全部应用于所有四个 i3.4xlarge 实例（i3.4xlarge 标准化因子 = 32）。

反之亦然。例如，如果您购买 US East (N. Virginia) 中的两个 i3.8xlarge 默认租期 Amazon Linux/Unix 预留实例，并且您在该区域中有一个正在运行的 i3.metal 实例，则账单优惠全部应用于 i3.metal 实例。

应用预留实例的示例

以下方案涵盖了各种应用预留实例的方式。

- 方案 1：单个账户中的预留实例
- 场景 2：使用标准化因素在单个账户中使用预留实例
- 场景 3：关联账户中的区域性预留实例
- 场景 4：关联账户中的可用区预留实例

方案 1：单个账户中的预留实例

您在账户 A 中运行以下按需型实例：

- 4 x `m3.large` Linux，可用区 us-east-1a 中的默认租期实例
- 2 x `m4.xlarge` Amazon Linux，可用区 us-east-1b 中的默认租期实例
- 1 x `c4.xlarge` Amazon Linux，可用区 us-east-1c 中的默认租期实例

您在账户 A 中购买以下预留实例：

- 4 x `m3.large`，可用区 us-east-1a 中的默认租期 预留实例（容量为预留）
- 4 x `m4.large` Amazon Linux，区域 us-east-1 中的默认租期 预留实例
- 1 x `c4.large` Amazon Linux，区域 us-east-1 中的默认租期 预留实例

Reserved Instance优惠以下面方式应用：

- 四个 `m3.large` 地区性 预留实例 的折扣和容量预留将由四个 `m3.large` 实例使用，因为它们之间的属性（实例大小、区域、平台、租期）相匹配。
- `m4.large` 区域性 预留实例 具备可用区和实例大小灵活性，因为它们是带默认租期的区域性 Amazon Linux 预留实例。

  `m4.large` 等效于 4 个标准化单位/小时。

  您已购买四个 `m4.large` 区域性 预留实例，它们加起来等效于 16 个标准化单位/小时 (4x4)。账户 A 具有两个正在运行的 `m4.xlarge` 实例，等效于 16 个标准化单位/小时 (2x8)。这种情况下，四个 `m4.large` 区域性预留实例可享受使用两个 `m4.xlarge` 实例的全部成本优势。

- us-east-1 中的 c4.large 区域 Reserved Instance 提供了可用区和实例大小灵活性，因为它是带默认租期的区域 Amazon Linux Reserved Instance，并且将应用于 c4.xlarge 实例。c4.large 实例等效于 4 个标准化单位/小时，c4.xlarge 等效于 8 个标准化单位/小时。

  在这种情况下，c4.large 区域 Reserved Instance 提供了针对 c4.xlarge 用量的部分优势。这是因为 c4.large Reserved Instance 等效于 4 个标准化单位/小时的用量，而 c4.xlarge 实例需要 8 个标准化单位/小时。因此，c4.large Reserved Instance 账单折扣应用于 50% 的 c4.xlarge 用量。剩余的 c4.xlarge 用量按照按需费率收费。

场景 2：使用标准化因素在单个账户中使用预留实例

您在账户 A 中运行以下按需型实例：

- 2 x m3.xlarge Amazon Linux，可用区 us-east-1a 中的默认租期实例
- 2 x m3.large Amazon Linux，可用区 us-east-1b 中的默认租期实例

您在账户 A 中购买以下预留实例：

- 1 x m3.2xlarge Amazon Linux，区域 us-east-1 中的默认租期预留实例

Reserved Instance优惠以下面方式应用：

- us-east-1 中的 m3.2xlarge 区域预留实例提供了可用区和实例大小灵活性，因为它是带默认租期的区域 Amazon Linux 预留实例。它首先适用于 m3.large 实例，然后适用于 m3.xlarge 实例，因为它根据标准化因子应用于实例系列中从最小到最大的实例大小。

  m3.large 实例等效于 4 个标准化单位/小时。

  m3.xlarge 实例等效于 8 个标准化单位/小时。

  m3.2xlarge 实例等效于 16 个标准化单位/小时。

  该益处应用如下：

  m3.2xlarge 区域性预留实例为 2 x m3.large 使用提供充分的益处，因为这些实例加起来占 8 个标准化单位/小时。这会使 8 个标准化单位/小时适用于 m3.xlarge 实例。

对于剩余 8 个标准化单位/小时，m3.2xlarge 区域性预留实例为 1 x m3.xlarge 使用提供充分益处，因为每个 m3.xlarge 实例等效于 8 个标准化单位/小时。剩余的 m3.xlarge 用量按照按需费率收费。

场景 3：关联账户中的区域性预留实例

预留实例首先供购买它们的账户使用，然后供组织中符合条件的任何其他账户使用。有关更多信息，请参阅预留实例和整合账单。对于具备大小灵活性的区域性预留实例，这种优势适用于实例系列中各种规模的实例 (从最小到最大)。

您在账户 A (购买账户) 中运行以下按需型实例：

- 2 x m4.xlarge Linux，可用区 us-east-1a 中的默认租期实例
- 1 x m4.2xlarge Linux，可用区 us-east-1b 中的默认租期实例
- 2 x c4.xlarge Linux，可用区 us-east-1a 中的默认租期实例
- 1 x c4.2xlarge Linux，可用区 us-east-1b 中的默认租期实例

另一个客户在账户 B（链接账户）中运行以下 按需型实例：

- 2 x m4.xlarge Linux，可用区 us-east-1a 中的默认租期实例

您在账户 A 中购买以下区域性预留实例：

- 4 x m4.xlarge Linux，区域 us-east-1 中的默认租期 预留实例
- 2 x c4.xlarge Linux，区域 us-east-1 中的默认租期 预留实例

区域Reserved Instance优惠以下面方式应用：

- 四个 m4.xlarge 预留实例 的折扣将由两个 m4.xlarge 实例和账户 A（购买账户）中的单个 m4.2xlarge 实例使用。所有三个实例均与这些属性相匹配（实例系列、区域、平台和租期）。折扣将首先应用于购买账户（账户 A）中的实例，即使账户 B（链接账户）具有两个也匹配 预留实例 的 m4.xlarge。由于 预留实例 是区域性 预留实例，因此没有容量预留。
- 两个 c4.xlarge 预留实例 的折扣适用于两个 c4.xlarge，因为它们比 c4.2xlarge 实例小。由于 预留实例 是区域性 预留实例，因此没有容量预留。

场景 4：关联账户中的可用区预留实例

通常，某个账户拥有的预留实例首先供该账户自用。不过，如果组织的其他账户中有适用于特定可用区、符合条件的未使用 预留实例（地区性 预留实例），这些实例将先于账户拥有的区域性 预留实例应用于账户。这样做是为了确保实现最大Reserved Instance使用率和较低的费用。出于记账目的，组织中的所有账户将被视为一个账户。以下示例可能有助于您理解。

您在账户 A (购买账户) 中运行以下个按需型实例：

- 1 x `m4.xlarge` Linux，可用区 us-east-1a 中的默认租期实例

某个客户在关联账户 B 中运行了以下个按需型实例：

- 1 x `m4.xlarge` Linux，可用区 us-east-1b 中的默认租期实例

您在账户 A 中购买以下区域性预留实例：

- 1 x `m4.xlarge` Linux，区域 us-east-1 中的默认租期 Reserved Instance

客户还在关联账户 C 中购买了以下区预留实例：

- 1 x `m4.xlarge` Linux，可用区 us-east-1a 中的默认租期 预留实例

Reserved Instance优惠以下面方式应用：

- 账户 C 拥有的 `m4.xlarge` 地区性 Reserved Instance 的折扣应用于账户 A 中的 `m4.xlarge` 用量。
- 账户 A 拥有的 `m4.xlarge` 区域性 Reserved Instance 的折扣应用于账户 B 中的 `m4.xlarge` 用量。
- 如果账户 A 拥有的区域性 Reserved Instance 先应用于账户 A 中的用量，则账户 C 拥有的地区性 Reserved Instance 将保持未使用状态，而账户 B 中的用量将按照按需费率收费。

有关更多信息，请参阅 Amazon 成本和使用情况报告中的 Understanding your reservations。

> **Note**
>
> 区域预留实例仅为所属账户预留容量，不能与其他 Amazon Web Services 账户 账户共享。如果您需要与其他 Amazon Web Services 账户 共享容量，请使用 使用 EC2 按需容量预留来预留计算容量。

使用您的预留实例

预留实例 将自动应用于正在运行的 按需型实例（前提匹配规范）。如果正在运行的 按需型实例 都与 Reserved Instance 的规范不匹配，则不会使用 Reserved Instance，直到您启动具有指定规范的实例。

如果您要启动按需型实例以利用预留实例的账单收益，请确保您在配置按需型实例时指定了以下信息：

平台

您必须指定与您预留实例的平台（产品说明）相匹配的亚马逊机器映像（AMI）。例如，如果您为预留实例指定 Linux/UNIX，则可以从 Amazon Linux AMI 或 Ubuntu AMI 启动实例。

实例类型

如果您购买了可用区预留实例，则必须指定与您的预留实例相同的实例类型，例如：t3.large。有关更多信息，请参阅 如何应用可用区预留实例。

如果您购买了区域性预留实例，则必须指定与预留实例的实例类型相同的实例系列中的实例类型。例如，如果您为预留实例指定 t3.xlarge 作为实例类型，则您必须从 T3 系列启动实例，但可指定其为任意大小，例如 t3.medium。有关更多信息，请参阅 如何应用区域性预留实例。

可用区

如果您为特定可用区购买了可用区预留实例，则必须在相同的可用区中启动实例。

如果您购买了区域性预留实例，则可以在任何您为该预留实例指定的区域的可用区中启动实例。

租赁

实例的租赁必须与预留实例的租赁匹配；例如 dedicated 或 shared。有关更多信息，请参阅 Amazon EC2 专用实例。

有关如何将预留实例应用于正在运行的按需型实例的示例，请参阅 如何应用预留实例折扣。有关更多信息，请参阅为什么我的 Amazon EC2 预留实例没有以我期望的方式应用于我的 Amazon 账单？

您可以使用多种方法来启动使用您的预留实例折扣的按需型实例。有关不同启动方法的信息，请参阅启动 Amazon EC2 实例。您也可以使用 Amazon EC2 Auto Scaling 来启动实例。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南。

预留实例的计费方式

与按需定价不同，所有预留实例都提供折扣。使用预留实例时，无论实际使用情况如何，都需要为整个期限付费。您可以根据为 Reserved Instance 指定的付款选项，为 Reserved Instance 选择预付、部分预付或按月付费。

预留实例过期后，需要根据按需费率支付 EC2 实例使用费用。您最早可以提前三年排队购买 Reserved Instance。这样可以帮助您确保获得不中断的服务。有关更多信息，请参阅 排队购买。

内容

- 使用情况计费
- 查看您的账单
- 预留实例和整合账单
- Reserved Instance折扣定价套餐

使用情况计费

在选择的预留实例期限内，无论实例是否运行，预留实例均按小时计费。每时钟小时从标准 24 小时制的整点（该小时经过了零分零秒）开始。例如，1:00:00 到 1:59:59 是一个时钟小时。有关实例状态的更多信息，请参阅 Amazon EC2 实例状态更改。

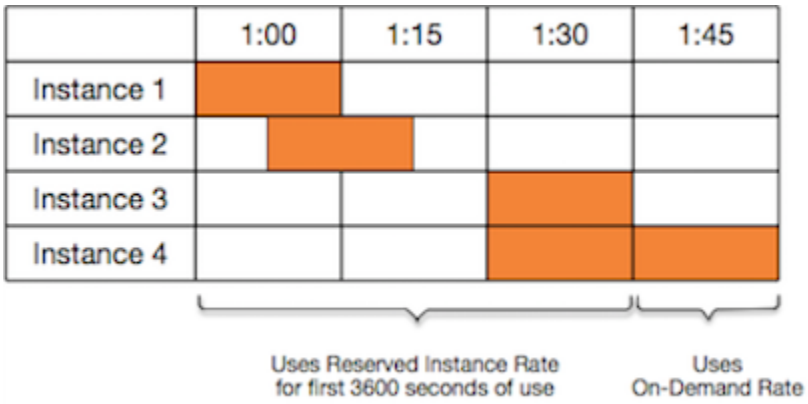
Reserved Instance账单优势适用于按秒收费的运行实例。每秒计费适用于使用开源 Linux 发行版的实例，例如 Amazon Linux 和 Ubuntu。每小时计费用于商业 Linux 发行版，例如 Red Hat Enterprise Linux 和 SUSE Linux Enterprise Server。

Reserved Instance账单优势适用于每小时最多 3600 秒 (一小时) 的实例使用。您可同时运行多个实例，但每小时只能获得总计 3600 秒的Reserved Instance折扣优惠；每小时超出 3600 秒的实例使用将根据按需费率计费。

例如，如果您购买了一个 `m4.xlarge` Reserved Instance，同时运行 4 个 `m4.xlarge` 实例 1 小时，则一个实例将按 1 小时的Reserved Instance使用收费，其他三个实例将按 3 小时的按需使用收费。

但是，如果您购买一个 `m4.xlarge` Reserved Instance，在同一小时内运行 4 个 `m4.xlarge` 实例各 15 分钟（900 秒），那么实例的总运行时间为 1 小时，这将产生 1 小时的Reserved Instance使用和 0 小时的按需使用。

如果多个合格实例同时运行，Reserved Instance账单优势将在一小时内（最多 3600 秒）同时适用于所有实例；在该时间后，根据按需费率收取费用。



使用 Billing and Cost Management 控制台上的 Cost Explorer (成本管理器) 可以分析运行 按需实例 所节省的成本。预留实例 常见问题包括标价计算的示例。

如果您关闭 Amazon 账户，则您资源的按需计费会停止。不过，如果账户中有任何预留实例，则会继续收到这些实例的账单，直至实例过期。

查看您的账单

您可通过查看 Amazon 账单与成本管理 控制台来了解您的账户的费用情况。

- 控制面板显示了您的账户的花费汇总。
- 在 Bills (账单) 页面的 Details (详细信息) 下，展开 Elastic Compute Cloud 部分及区域，以了解有关您的 预留实例 的账单信息。

您可以在线查看费用，也可以下载 CSV 文件。

您还可以使用Amazon成本和使用情况报告来跟踪您的预留实例使用情况。有关更多信息，请参阅 Understanding your reservations。

预留实例和整合账单

如果购买者账户是在一个整合账单付款人账户之下计费的一组账户中的其中之一，则可以共享预留实例定价优惠。每月将在付款人账户中汇总所有成员账户的实例使用量。这通常对具有不同职能团队或团体的公司很有用；然后，将应用正常的Reserved Instance逻辑来计算账单。有关更多信息，请参阅 Amazon Organizations 的整合账单。

如果您关闭购买预留实例的账户，会继续向付款人账户收取预留实例的费用，直到预留实例过期为止。已关闭的账户永久删除后，成员账户不再享受预留实例账单折扣。

> **Note**
>
> 区域预留实例仅为所属账户预留容量，不能与其他 Amazon Web Services 账户 账户共享。如果您需要与其他 Amazon Web Services 账户 共享容量，请使用 使用 EC2 按需容量预留来预留计算容量。

Reserved Instance折扣定价套餐

如果您的账户有资格获得折扣定价套餐，那么自您取得该资格时起，您在该套餐等级内购买的Reserved Instance的预付费和实例使用费均自动享受折扣。要取得折扣资格，在该区域内的 预留实例的标价必须达到 500000 美元或更高。

以下规则适用：

- 定价套餐和相关折扣仅适用于购买 Amazon EC2 标准 预留实例。
- 定价套餐不适用于面向带 SQL Server Standard、SQL Server Web 和 SQL Server Enterprise 的 Windows 的 预留实例。
- 定价套餐不适用于任何类型的含有 SQL Server 的 Linux 的预留实例。
- 定价套餐折扣仅适用于通过 Amazon 进行的购买。这些折扣不适用于第三方预留实例购买。
- 折扣定价套餐当前不适用于可转换预留实例购买。

主题

- 计算 Reserved Instance 定价折扣
- 以折扣套餐价格购买
- 跨越定价套餐

- 定价套餐的整合账单

计算 Reserved Instance 定价折扣

通过计算在区域中的所有 预留实例 的标价，可以确定账户所适用的定价套餐。将每个预留的每小时费用乘以期限的总小时数，再加上购买时的未打折预付价格（也称为固定价格）。因为价目表值基于未打折 (公开) 定价，是否有资格获得批量折扣或者购买预留实例后是否降价均不影响价目表值。

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

例如，对于一年期部分预付 `t2.small` Reserved Instance，假定预付价格是 60.00 美元，每小时费率为 0.007 美元。这将提供 121.32 美元的标价。

```
121.32 = 60.00 + (0.007 * 8760)
```

使用 Amazon EC2 控制台查看 预留实例 的固定价格

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances。
3. 要显示预付价格列，请选择右上角的设置（ ），开启预付价格，然后选择确认。

使用命令行查看 预留实例 的固定价格

- describe-reserved-instances (Amazon CLI)
- Get-EC2ReservedInstance (Amazon Tools for Windows PowerShell)

以折扣套餐价格购买

购买 预留实例 时，Amazon EC2 自动将所有折扣应用于所购产品处于折扣定价套餐范围内的部分。您无需执行任何其他操作，而且可以使用任何 Amazon EC2 工具购买 预留实例。有关更多信息，请参阅购买 Amazon EC2 的预留实例。

在某区域的活动 预留实例 的标价达到某一折扣定价套餐范围后，以后在该区域购买任何 预留实例 都将按打折费率计费。如果在某区域的 预留实例 单项购买额超过折扣套餐阈值，则该项购买超出价格阈

值的部分将按打折费率计费。有关在购买过程中创建的临时 Reserved Instance ID 的更多信息，请参阅跨越定价套餐。

如果标价降至低于折扣定价套餐价格点（例如，如果部分 预留实例 到期），之后在该区域购买 预留实例 将不享受折扣。不过，原来在折扣定价套餐范围内购买的所有预留实例将继续享受折扣。

购买预留实例时，可能出现以下四种情况之一：

- 没有折扣 — 您在某区域内的购买仍然低于折扣阈值。
- 部分折扣 — 您在某区域内的购买跨越了第一折扣套餐的阈值。没有折扣将应用于一个或多个预留，而折扣费率将应用于剩余的预留。
- 全部折扣 — 您在某区域内的购买全部在一个折扣套餐之内并且获得了相应的折扣。
- 两种折扣率 — 您在某区域内的购买从较低折扣套餐跨入较高的折扣套餐。您将按两种费率付费：一个或多个预留采用较低的折扣费率，剩余的预留采用较高的折扣费率。

跨越定价套餐

如果您的购买跨入某个折扣定价套餐范围，您将看到该项购买有多个条目：一个条目显示购买中将按常规价格收费的部分，另一个条目显示购买中将按适用的打折费率收费的部分。

Reserved Instance 服务会生成多个 Reserved Instance ID，因为您的购买从未打折套餐跨入到打折套餐，或从一个打折套餐跨入到另一个打折套餐。套餐中的每组预留都有一个 ID。因此，由购买 CLI 命令或 API 操作返回的 ID 不同于新预留实例的实际 ID。

定价套餐的整合账单

整合账单账户汇总了某个区域内所有成员账户的标价。当整合账单账户的所有活动 预留实例 的标价达到折扣定价套餐时，整合账单账户的任何账户成员在此后购买任何 预留实例 都将享受打折费率（前提是整合账单账户的标价始终高于折扣定价套餐阈值）。有关更多信息，请参阅 预留实例和整合账单。

购买 Amazon EC2 的预留实例

要购买 Amazon EC2 的预留实例，您可以使用 Amazon EC2 控制台、命令行工具或 SDK 搜索 Amazon 和第三方卖家的预留实例产品，调整搜索参数，直到找到与您的目标完全相符的对象。

在搜索要购买的预留实例时，您将收到一个关于退还产品的成本报价。当您继续购买时，Amazon 将自动对购买价格设定一个限定价格。预留实例的总成本不会超过报价金额。

如果价格由于任何原因上升或变动，将不会完成购买。当您从 Amazon EC2 预留实例市场购买第三方卖家的预留实例时，如果有与您的选择类似但预付价格较低的产品，Amazon 将以较低的预付价格向您出售这些产品。

在确认购买之前，请检查您计划购买的Reserved Instance的详细信息，并确保所有参数都是准确的。在您购买预留实例之后（无论是从预留实例 Marketplace 中的第三方卖家购买还是从 Amazon 购买），将无法取消您的购买。您可以将购买排队到将来的某个日期，也可以在预定时间之前取消已排队的购买。

要购买和修改预留实例，请确保您的用户具有相应的权限，例如描述可用区的能力。有关信息，请参阅 the section called “使用预留实例”（API）或 the section called “使用预留实例”（控制台）。

任务

- 选择平台
- 排队购买
- 购买标准 预留实例
- 购买 可转换预留实例
- 从预留实例 Marketplace 中购买
- 取消已排队的购买
- 续订 Reserved Instance

选择平台

Amazon EC2 支持以下平台的预留实例：

- Linux/UNIX
- 含有 SQL Server Standard 的 Linux
- 含有 SQL Server Web 的 Linux
- 含有 SQL Server Enterprise 的 Linux
- SUSE Linux
- Red Hat Enterprise Linux
- Red Hat Enterprise Linux with HA
- Windows
- 含有 SQL Server Standard 的 Windows
- 含有 SQL Server Web 的 Windows

- 含有 SQL Server Enterprise 的 Windows

注意事项

- 如果您对 Red Hat Enterprise Linux、SUSE Linux 或 Ubuntu Pro 使用现有订阅（BYOS），则必须选择适用于 Linux/Unix 平台的服务产品。
- 运行 macOS 或 Ubuntu Pro（包括 EC2 订阅，即不包括 BYOS）的实例不支持预留实例。如需通过按需型实例定价节省费用，建议您使用配套节省计划的 macOS 和 Ubuntu Pro（包括 EC2 订阅）。有关更多信息，请参阅《Savings Plans User Guide》。

为确保实例在特定的预留实例中运行，预留实例的平台必须与用于启动实例的 AMI 的平台相匹配。对于 Linux AMI，请务必检查 AMI 平台是使用常规值 Linux/UNIX 还是具体值（如 SUSE Linux）。

Console

检查 AMI 平台

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡上，记下平台详细信息的值。

Amazon CLI

检查 AMI 平台

使用 describe-images 命令并检查 `PlatformDetails` 的值。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

下面是示例输出。

```
[
    "Linux/UNIX"
]
```

PowerShell

检查 AMI 平台

使用 Get-EC2Image cmdlet 并检查 PlatformDetails 的值。

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

下面是示例输出。

```
PlatformDetails
---------------
Linux/UNIX
```

排队购买

默认情况下，当您购买时 Reserved Instance，购买立即完成。或者，您也可以排队预约在将来的某个日期和时间购买。例如，您可以排队预约在现有 Reserved Instance 到期的时间购买。这样可以帮助您确保获得不中断的服务。

您可以排队购买区域 预留实例，但不能排队购买其他卖家的区域 预留实例 或 预留实例。您最早可以提前三年排队购买。在指定日期和时间，将使用默认支付方式进行购买。支付成功后，将体现账单优势。

您可以在 Amazon EC2 控制台中为已排队的购买设置日期，购买将排队等候至该日期的 00:00 UTC。要为已排队的购买指定不同的时间，请使用 Amazon SDK 或命令行工具。

您可以在 Amazon EC2 控制台中查看已排队的购买。已排队的购买的状态为已排队。在指定时间之前，您随时可以取消已排队的购买。有关详细信息，请参阅 取消已排队的购买。

购买标准 预留实例

您可以购买特定可用区中的标准预留实例从而获得容量预留。或者，您也可以放弃容量预留并购买区域性标准Reserved Instance。

购买完成后，如果已有正在运行的与 Reserved Instance 规范匹配的实例，账单优势将立即体现。您不需要重启实例。如果您没有合适的正在运行的实例，请启动实例并确保符合您为预留实例指定的相同标准。有关更多信息，请参阅 使用您的预留实例。

有关如何将 预留实例 应用于正在运行的实例的示例，请参阅如何应用预留实例折扣。

Console

购买标准预留实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances (预留实例)，然后选择 Purchase 预留实例 (购买 预留实例)。
3. 对于 Offering class (服务类别)，选择 Standard (标准) 以显示标准 预留实例。
4. 要购买容量预留，请在购买屏幕的右上角开启 Only show offerings that reserve capacity (只显示预留容量的产品)。开启此设置后，Availability Zone (可用区) 字段将会出现。

   要购买区域 Reserved Instance，请关闭此设置。关闭此设置后，Availability Zone (可用区) 字段将会消失。
5. 根据需要选择其他配置，然后选择 Search (搜索)。
6. 对于您要购买的每个 Reserved Instance，输入所需数量，然后选择 Add to cart (添加到购物车)。

   要从预留实例市场购买标准预留实例，请在搜索结果的 Seller (卖家) 列中查找 3rd party (第三方)。Term 列会显示非标准期限。有关更多信息，请参阅从预留实例 Marketplace 中购买。
7. 要查看已选择的 预留实例 的汇总，请选择 View cart (查看购物车)。
8. 如果 Order on (订购时间) 为 Now (现在)，则在您选择 Order all (全部订购) 后，购买将立即完成。要排队购买，请选择 Now (现在) 并选择一个日期。您可以为购物车中每个符合条件的产品选择不同的日期。购买已排队，一直到选定日期 UTC 00:00 之前。
9. 要完成订单，请选择 Order all (全部订购)。

   如果在下订单时有与您的选择类似的低价位产品，Amazon 将为您提供价格更低的产品。
10. 选择 Close。

    您的订单状态将在 State (状态) 列中列出。当您的订单完成时，State (状态) 值将从 `Payment-pending` 变为 `Active`。当 Reserved Instance 的状态为 `Active` 时即可使用。

    如果状态转为 `Retired`，Amazon 可能未收到您的付款。

Amazon CLI

购买标准预留实例

1. 使用 describe-reserved-instances-offerings 命令查找可用 预留实例。为 `--offering-class` 选项指定 `standard` 以仅返回标准预留实例。可以应用更多标准来缩小结果范围。例如，使用以下命令为 `Linux/UNIX` 购买具有默认租赁的仅一年期的区域性 `t2.large` 预留实例。

```
aws ec2 describe-reserved-instances-offerings \
    --instance-type t2.large \
    --offering-class standard \
    --product-description "Linux/UNIX" \
    --instance-tenancy default \
    --filters Name=duration,Values=31536000 \
              Name=scope,Values=Region
```

要仅在预留实例 Marketplace 上查找预留实例，请使用 `marketplace` 筛选条件并在请求中不指定持续时间，因为期限可能会短于 1 年期或 3 年期。

```
aws ec2 describe-reserved-instances-offerings \
    --instance-type t2.large \
    --offering-class standard \
    --product-description "Linux/UNIX" \
    --instance-tenancy default \
    --filters Name=marketplace,Values=true
```

当您找到符合需求的 Reserved Instance 时，请记下产品 ID。例如：

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. 使用 purchase-reserved-instances-offering 命令购买您的 Reserved Instance。您必须指定在上一步中获取的Reserved Instance产品 ID，并且必须为预留指定实例数量。

```
aws ec2 purchase-reserved-instances-offering \
    --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
    --instance-count 1
```

默认情况下，会立即完成购买。或者，若要排队购买，请在之前的调用中添加以下选项。

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. 使用 describe-reserved-instances 命令获取您的 Reserved Instance 的状态。

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

购买标准预留实例

1. 使用 Get-EC2ReservedInstancesOffering cmdlet 查找可用的预留实例。为 `standard` 参数指定 `-OfferingClass` 以仅返回标准 预留实例。可以应用更多标准来缩小结果范围。例如，使用以下命令为 `Linux/UNIX` 购买具有默认租赁的仅一年期的区域性 `t2.large` 预留实例。

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="duration"; Values="31536000"} `
             @{Name="scope"; Values="Region"
```

要仅在预留实例 Marketplace 上查找预留实例，请使用 `marketplace` 筛选条件并在请求中不指定持续时间，因为期限可能会短于 1 年期或 3 年期。

```
Get-EC2ReservedInstancesOffering `
    -InstanceType t2.large `
    -OfferingClass "standard" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy default `
    -Filters @{Name="marketplace"; Values="true"}
```

当您找到符合需求的 Reserved Instance 时，请记下产品 ID。例如：

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. 使用 New-EC2ReservedInstance cmdlet 购买预留实例。您必须指定在上一步中获取的 Reserved Instance产品 ID，并且必须为预留指定实例数量。

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
    -InstanceCount 1
```

默认情况下，会立即完成购买。或者，若要排队购买，请在之前的调用中增加以下参数。

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. 使用 Get-EC2ReservedInstance cmdlet 获取预留实例的状态。

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

购买 可转换预留实例

您可以购买特定可用区中的可转换预留实例从而获得容量预留。或者，您也可以放弃容量预留并购买区域性可转换预留实例。

如果已经有与Reserved Instance的规格匹配的运行实例，则将立即体现账单收益。您不必重启您的实例。如果您没有合适的运行实例，请启动实例并确保符合您为Reserved Instance指定的相同标准。有关更多信息，请参阅使用您的预留实例。

有关如何将 预留实例 应用于正在运行的实例的示例，请参阅如何应用预留实例折扣。

Console

购买可转换预留实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances (预留实例)，然后选择 Purchase 预留实例 (购买 预留实例)。
3. 对于 Offering class (服务类别)，选择 Convertible (可转换) 以显示 可转换预留实例。
4. 要购买容量预留，请在购买屏幕的右上角开启 Only show offerings that reserve capacity (只显示预留容量的产品)。开启此设置后，Availability Zone (可用区) 字段将会出现。

   要购买区域 Reserved Instance，请关闭此设置。关闭此设置后，Availability Zone (可用区) 字段将会消失。

5. 根据需要选择其他配置并选择 搜索。
6. 对于您要购买的每个 可转换预留实例，输入数量，然后选择 Add to cart (添加到购物车)。
7. 要查看您的选择的摘要，请选择 View cart (查看购物车)。
8. 如果 Order on (订购时间) 为 Now (现在)，则在您选择 Order all (全部订购) 后，购买将立即完成。要排队购买，请选择 Now (现在) 并选择一个日期。您可以为购物车中每个符合条件的产品选择不同的日期。购买已排队，一直到选定日期 UTC 00:00 之前。
9. 要完成订单，请选择 Order all (全部订购)。

   如果在下订单时有与您的选择类似的低价位产品，Amazon 将为您提供价格更低的产品。
10. 选择 Close。

    您的订单状态将在 State (状态) 列中列出。当您的订单完成时，State (状态) 值将从 `Payment-pending` 变为 `Active`。当 Reserved Instance 的状态为 `Active` 时即可使用。

    如果状态转为 `Retired`，Amazon 可能未收到您的付款。

Amazon CLI

购买可转换预留实例

1. 使用 describe-reserved-instances-offerings 命令查找可用 预留实例。为 `--offering-class` 选项指定 `convertible` 以仅返回可转换预留实例。可以应用更多标准来缩小结果范围。例如，使用以下命令为 `Linux/UNIX` 购买具有默认租赁的区域性 `t2.large` 预留实例。

   ```
   aws ec2 describe-reserved-instances-offerings \
       --instance-type t2.large \
       --offering-class convertible \
       --product-description "Linux/UNIX" \
       --instance-tenancy default \
       --filters Name=scope,Values=Region
   ```

   当您找到符合需求的 Reserved Instance 时，请记下产品 ID。例如：

   ```
   "ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
   ```
2. 使用 purchase-reserved-instances-offering 命令购买您的 Reserved Instance。您必须指定在上一步中获取的Reserved Instance产品 ID，并且必须为预留指定实例数量。

```
aws ec2 purchase-reserved-instances-offering \
    --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
    --instance-count 1
```

默认情况下，会立即完成购买。或者，若要排队购买，请在之前的调用中添加以下选项。

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. 使用 describe-reserved-instances 命令获取您的 Reserved Instance 的状态。

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327 \
    --query ReservedInstances[].State
```

PowerShell

购买可转换预留实例

1. 使用 Get-EC2ReservedInstancesOffering cmdlet 查找可用的预留实例。为 `convertible` 参数指定 `-OfferingClass` 以仅返回 可转换预留实例。可以应用更多标准来缩小结果范围。例如，使用以下命令为 Linux/UNIX 购买具有默认租赁的区域性 `t2.large` 预留实例。

```
Get-EC2ReservedInstancesOffering `
    -InstanceType "t2.large" `
    -OfferingClass "convertible" `
    -ProductDescription "Linux/UNIX" `
    -InstanceTenancy "default" `
    -Filters @{Name="scope"; Values="Region"
```

当您找到符合需求的 Reserved Instance 时，请记下产品 ID。例如：

```
bec624df-a8cc-4aad-a72f-4f8abc34caf2
```

2. 使用 New-EC2ReservedInstance cmdlet 购买预留实例。您必须指定在上一步中获取的预留实例产品 ID，并且必须为预留指定实例数量。

```
New-EC2ReservedInstance `
    -ReservedInstancesOfferingId "bec624df-a8cc-4aad-a72f-4f8abc34caf2" `
```

```
    -InstanceCount 1
```

默认情况下，会立即完成购买。或者，若要排队购买，请在之前的调用中增加以下参数。

```
-PurchaseTime "2020-12-01T00:00:00Z"
```

3. 使用 Get-EC2ReservedInstance cmdlet 获取预留实例的状态。

```
Get-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327 | `
    Select State
```

从预留实例 Marketplace 中购买

您可以从预留实例 Marketplace 向不再需要其预留实例的第三方卖家购买预留实例。您可以使用 Amazon EC2 控制台或命令行工具执行此操作。该过程类似于从Amazon购买预留实例。有关更多信息，请参阅购买标准 预留实例。

在预留实例 Marketplace 中购买的预留实例与直接从 Amazon 购买的预留实例有一些区别：

- 有效期 – 从第三方卖方购买的预留实例具有的剩余期限短于完整标准期限。从 Amazon 获得的完整标准期限为一年或三年。
- 预付价格 – 第三方预留实例可以不同的预付价格出售。使用费或周期性费用与最初从Amazon购买预留实例时设定的费用一致。
- 预留实例类型 – 只能从预留实例市场购买 Amazon EC2 标准预留实例。可转换预留实例、Amazon RDS 和 Amazon ElastiCache 预留实例不能在预留实例 Marketplace 上购买。

有关您的基本信息将与卖方进行共享，如您的邮政编码和国家/地区信息。

此信息使卖方能够计算他们必须向政府缴纳并且采用支付报告形式提供的任何必需的交易税 (如销售税或增值税)。在极少数情况下，Amazon 可能必须向卖方提供您的电子邮件地址，这样卖方才能就与销售相关的问题 (例如税务问题) 与您联系。

出于相似的原因，Amazon 将在买方的购货发票上共享卖方的法律实体名称。如果您出于税务或相关原因需要关于卖方的额外信息，请联系 Amazon Web Services 支持。

取消已排队的购买

您最早可以提前三年排队购买。在指定时间之前，您随时可以取消已排队的购买。

Console

取消已排队的购买

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances。
3. 选择一个或多个 预留实例。
4. 依次选择 Actions (操作)、Delete queued Reserved Instances (删除已排队的预留实例)。
5. 提示进行确认时，选择 Delete (删除)，然后选择 Close (关闭)。

Amazon CLI

取消已排队的购买

使用 delete-queued-reserved-instances 命令。

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342fec06327
```

PowerShell

取消已排队的购买

使用 Remove-EC2QueuedReservedInstance cmdlet。

```
Remove-EC2QueuedReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342fec06327
```

续订 Reserved Instance

您可以在 Reserved Instance 按计划过期之前续订它。续订 Reserved Instance 将对购买 Reserved Instance（具有相同配置）进行排队，直到当前 Reserved Instance 过期。

您必须使用 Amazon EC2 控制台续订预留实例。

使用排队的购买续订预留实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances。

3. 选择要续订的预留实例。
4. 依次选择 Actions (操作)、Renew Reserved Instances (续订预留实例)。
5. 要完成订单，请选择 Order all (全部订购)，然后选择 Close (关闭)。

在预留实例市场中销售 Amazon EC2 的预留实例

Amazon EC2 预留实例市场是一个便于 Amazon 客户和第三方卖家销售未使用标准预留实例的平台。这些预留实例的期限长短和定价选项可能有所不同。当您不再需要预留实例时，例如将实例移动到新的 Amazon Web Services 区域、更改为其他实例类型、在预留实例期限到期之前完成项目、业务需求发生变化或容量过剩时，您可能希望出售预留实例。

只要在预留实例 Marketplace 中列出预留实例，便可供潜在的买方找到。所有预留实例将根据剩余期限及小时价格进行分组。

为了满足买家通过预留实例市场购买第三方卖家预留实例的请求，Amazon 首先以指定分组中最低的预付价格出售预留实例。然后，Amazon再出售下一个最低价格的预留实例，直到买方的整个订单完成为止。Amazon随后处理这些交易，并将预留实例的所有权转移给买方。

在您的Reserved Instance出售之前，它将归您所有。出售之后，您便放弃了容量预留和打折的周期性费用。如果继续使用您的实例，Amazon将从您的预留实例出售的时间开始以按需价格向您收费。

在预留实例 Marketplace 上销售未使用的预留实例必须满足特定资格条件。

有关在预留实例 Marketplace 上购买预留实例的更多信息，请参阅 从预留实例 Marketplace 中购买。

目录

- 限制和局限性
- 注册成为卖家
- 银行支付账户
- 税务信息
- 您的 预留实例 的价格
- 列出您的 预留实例
- Reserved Instance列表状态
- 列表的生命周期
- 在Reserved Instance出售后
- 收款

- 与买方共享的信息

限制和局限性

您必须先注册为预留实例 Marketplace 中的卖家，然后才能出售未使用的预留实例。有关信息，请参阅注册成为卖家。

以下限制在出售预留实例时适用：

- 只有 Amazon EC2 标准区域和分区预留实例可在预留实例 Marketplace 出售。
- Amazon EC2 可转换预留实例不可在预留实例市场出售。
- 其他 Amazon 服务（比如 Amazon RDS 和 Amazon ElastiCache）的预留实例无法在预留实例 Marketplace 出售。
- 标准Reserved Instance的剩余期限必须至少为一个月。
- 您不能在默认禁用的区域中销售标准预留实例。
- 您不能在中国区域（北京和宁夏）中销售标准预留实例。
- 预留实例 Marketplace 中允许的最低价格为 0.00 USD。
- 只要实例在账户中处于激活状态至少 30 天，您就可以在预留实例市场中销售“无预付费用”“预付部分费用”或“预付全部费用”的预留实例。此外，如果预留实例上有预付款，则只能在 Amazon 收到预付款之后将其出售。
- 如果您使用批量折扣购买预留实例，则无法在预留实例市场中销售该预留实例。
- 您无法在预留实例 Marketplace 中直接修改您的列示内容。然而，您可通过先取消它然后再用新参数创建另一个实例出售清单来改变您的实例出售清单。有关信息，请参阅 您的 预留实例 的价格。您也可以在列出预留实例时对其进行修改。有关信息，请参阅 修改 预留实例。
- Amazon会向您收取您在预留实例市场中出售的每个标准预留实例的总预付价格 12% 的服务费。预付价格是卖方对标准Reserved Instance收取的费用。
- 当您注册为卖家时，您指定的银行必须有一个美国地址。有关更多信息，请参阅 Amazon Web Services Marketplace 销售指南中的对付费商品的其他卖家要求。
- 即使 Amazon Web Services India Private Limited（Amazon India）的客户拥有美国银行账户，也无法在 EC2 预留实例市场中注册为卖家，且无法在 EC2 预留实例市场中上架或出售预留实例。有关更多信息，请参阅Amazon Web Services 账户 和 Amazon 印度账户之间有何区别？
- 如果您记录的卖家更改为 Amazon Web Services India Private Limited (Amazon India)，则您的卖家身份将从 EC2 预留实例市场注销，并且将移除您在 EC2 预留实例市场中的所有现有在售商品。要恢复卖家身份，您必须将账户所在地更改为印度以外的国家/地区，然后重新完成卖家注册流程。

注册成为卖家

> **Note**
>
> 只有 Amazon Web Services 账户根用户 才可以将账户注册为卖家。

要在预留实例 Marketplace 中销售，您必须先注册为卖家。在注册过程中，您应提供以下信息：

- 银行信息 –Amazon为了支付您出售预留实例时收取的资金，必须获得您的银行信息。您指定的银行必须有一个美国地址。有关更多信息，请参阅银行支付账户。
- 税务信息 — 所有卖方都需要完成税务信息审查以确定任何必要的税务报告义务。有关更多信息，请参阅税务信息。

在 Amazon 收到您已完成的卖家注册后，您会收到对您的注册进行确认并告知您可以开始在预留实例 Marketplace 中出售实例的电子邮件。

银行支付账户

为了支付您出售预留实例时收取的资金，Amazon必须获得您的银行信息。您指定的银行必须有一个美国地址。有关更多信息，请参阅 Amazon Web Services Marketplace 销售指南中的对付费商品的其他卖家要求。

注册付款的默认银行账户

1. 打开预留实例市场卖家注册页面并使用您的 Amazon 凭证登录。
2. 在 Manage Bank Account 页面上，提供有关您的收款行的以下信息：
   - 银行账户持有人姓名
   - 路由号码
   - 账号
   - 银行账户类型

> **Note**
>
> 如果您正在使用一个公司银行账户，则系统将提示您通过传真 (1-206-765-3424) 发送关于该银行账户的信息。

注册后，将提供的银行账户设置为默认账户，等待银行进行验证。验证新的银行账户可能需要两周时间，在此期间，您无法收到付款。对于已建立的账户，付款的完成通常需要两天左右的时间。

更改付款的默认银行账户

1. 在预留实例市场卖家注册页面上，使用您注册时所用的账户登录。
2. 在 Manage Bank Account 页面上，根据需要添加新的银行账户或修改默认银行账户。

税务信息

出售预留实例可能需要交纳交易税，例如销售税或增值税。您应与您的企业的税务、法律、财务或会计部门沟通，以确定是否适用于基于交易的税种。您负责向相关税务机构收集并交纳基于交易的税款。

作为卖家注册的一部分，您必须在卖家注册门户中完成税务审查。此审查将收集税务信息并填充 IRS 表 W-9、W-8BEN 或 W-8BEN-E，后者用于确定任何必要的税务报告义务。

您在税务审查中输入的税务信息可能不同，具体取决于您是作为个人还是企业运营，以及您是否为美国人，您的企业是否为美国实体。当您填写税务资料时，请记住以下事项：

- Amazon 提供的信息 (包括本主题中的信息) 不构成税务、法律或其他专业建议。查明 IRS 报告要求将如何影响您的企业，或者如果您有其他问题，请联系您的税务、法律或其他专业顾问。
- 为了尽可能高效地满足 IRS 报告要求，在会见过程中回答所有的问题并输入所有要求的信息。
- 检查您的回答。避免拼写错误或输入了不正确的税务识别号，它们会导致纳税申报表格无效。

根据您的税务审查响应和 IRS 报告阈值，Amazon 可能对表格 1099-K 归档。 Amazon 会在您的账户达到阈值级别的那一年的后一年的 1 月 31 日或之前通过电子邮件发送表格 1099-K 的副本。例如，如果您的账户在 2018 年达到阈值，则将在 2019 年 1 月 31 日或之前通过电子邮件发送您的表 1099-K。

有关 IRS 要求和表 1099-K 的更多信息，请参阅 IRS 网站上的 Form 1099-K FAQs。

您的 预留实例 的价格

在为预留实例设置价格时，请注意以下方面：

- 预付价格 – 预付价格是您可为要出售的预留实例指定的唯一费用。预付价格是买方在购买每个预留实例时支付的一次性费用。

  因为预留实例的价值随时间的推移而降低，所以，默认情况下，Amazon可设定以同样的变化量逐月降低的价格。但是，您可根据预留实例出售的时间设置不同的预付价格。例如，如果您的 Reserved

Instance 剩余期限为九个月，您可以指定客户如需购买这个剩余九个月的 Reserved Instance，您愿意接受的价格。您还可以分别设置剩余期限为五个月、一个月的价格。

预留实例 Marketplace 中允许的最低价格为 0.00 USD。

- 限制 – 以下预留实例出售限制适用于您的 Amazon Web Services 账户的有效期。这种限制不是年度限制，无法增加。
  - 对于预留实例，您最多可以销售 50,000 USD。
  - 您最多可以销售 5,000 个预留实例。
- 不能修改 – 您无法直接修改上架项目。然而，您可通过先取消它然后再用新参数创建另一个实例出售清单来改变您的实例出售清单。
- 可以取消 – 您可以随时取消处于 `active` 状态的上架项目。您无法取消已经匹配或正在为销售进行处理的实例出售清单。如果您的实例出售清单中的某些实例已匹配且您取消了实例出售清单，则仅剩余的未匹配的实例将从实例出售清单中删除。

列出您的 预留实例

注册卖家可以选择销售一个或多个预留实例。您可以选择在一次列出中销售所有实例，或分成多个部分销售。此外，您可以列出任意实例类型、平台和范围配置的预留实例。

控制台将确定建议的价格。它会检查与您的 Reserved Instance 匹配的产品，并与价格最低的产品匹配。否则，它会根据剩余时间的 Reserved Instance 成本计算建议价格。如果计算出的价值小于 1.01 美元，则建议的价格为 1.01 美元。

如果您取消出售清单，且出售清单的一部分已经售出，则取消不会在已售出的部分生效。仅实例出售清单中未售出的部分在预留实例 Marketplace 中将不再可用。

Console

在预留实例 Marketplace 中列出预留实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances。
3. 选择要列出的预留实例，然后依次选择操作、销售预留实例。
4. 在 Configure Your Reserved Instance Listing (配置您的预留实例列表) 页面上，在相关列中设置要出售的实例数并为剩余期限设定预付价格。单击 Months Remaining 列旁边的箭头，了解您的预留的价值是如何随着剩余期限的变化而变化的。

5. 如果您是高级用户且想对定价进行自定义，那么您可为后续月输入一个不同的值。要返回默认的线性价格降低，请选择 Reset。
6. 当您完成列表配置后，请选择 Continue (继续)。
7. 在 Confirm Your Reserved Instance Listing (确认您的预留实例列表) 页面上确认您的列表详细信息；如果对此类信息感到满意，请选择 List Reserved Instance (列出预留实例)。

在 Amazon EC2 控制台中查看您的列表

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances。
3. 选择您已列出的Reserved Instance，然后选择页面底部附近的我的列表选项卡。

Amazon CLI

在预留实例 Marketplace 中管理预留实例

1. 使用 describe-reserved-instances 命令获取 预留实例 的列表。请注意您要列出的预留实例的 ID。
2. 使用 create-reserved-instances-listing 命令。您必须指定Reserved Instance的 ID、实例数以及价格表。
3. 要查看列表，请使用 describe-reserved-instances-listings 命令。
4. 要取消列表，请使用 cancel-reserved-instances-listing 命令。

PowerShell

在预留实例 Marketplace 中管理预留实例

1. 使用 Get-EC2ReservedInstance cmdlet 获取您的预留实例列表。请注意您要列出的预留实例的 ID。
2. 使用 New-EC2ReservedInstancesListing cmdlet。您必须指定Reserved Instance的 ID、实例数以及价格表。
3. 要查看您的列表，请使用 Get-EC2ReservedInstancesListing cmdlet。
4. 要取消您的列表，请使用 Stop-EC2ReservedInstancesListing cmdlet。

Reserved Instance列表状态

预留实例 页面的 My Listings (我的列表) 选项卡上的 Listing State (列表状态) 显示了列表的当前状态：

Listing State (列表状态) 显示的信息与您在预留实例市场中的列表的状态有关。它与 Reserved Instances (预留实例) 页面中的 State (状态) 列显示的状态信息不同。此 State 信息是关于您的预留的。

- active (已激活) — 列表可供购买。
- canceled (已取消) –列表已取消，并且在预留实例市场中不再可供购买。
- closed (已关闭) — Reserved Instance 未列出。Reserved Instance 可能因列表已完成销售而处于 `closed` 状态。

列表的生命周期

当实例出售清单中的所有实例都匹配且售出时，My Listings 选项卡将指示 Total instance count 匹配 Sold 下方列出的计数。此外，实例出售清单中没有 Available 实例，并且其 Status 为 `closed`。

当列表中只有一部分售出时，Amazon将停用列表中的预留实例并创建与剩余预留实例数量相等的预留实例。因此，实例出售清单 ID 及其代表的实例出售清单 (现在具有较少的待售预留) 仍处于激活状态。

将以此方式处理此列表中任何未来预留实例销售。当列表中的所有预留实例售出后，Amazon将列表标记为 `closed`。

例如，您创建一个列表数量为 5 的列表：预留实例 listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample。

Reserved Instance 控制台页中的 My Listings 选项卡将按以下所示显示实例出售清单：

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 总预留计数= 5
- 已售 = 0
- 可用 = 5
- 状态 = 已激活

某个买家购买了其中两个预留，这使得三个预留的计数依然可供销售。由于此部分销售，Amazon 创建了一个实例计数为 3 的新预留，以表示剩下的三个预留依然可供销售。

这是您的实例出售清单在 My Listings 选项卡中的显示方式：

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- 总预留计数= 5
- 已售 = 2
- 可用 = 3
- 状态 = 已激活

如果您取消您的出售清单，且出售清单的一部分已经售出，取消不会在已售出的部分生效。仅实例出售清单中未售出的部分在预留实例 Marketplace 中将不再可用。

在Reserved Instance出售后

当您的预留实例售出后，Amazon会向您发送一条电子邮件通知。每天如有任何类型的活动，您会收到一封电子邮件通知，其中包含当天的所有活动。活动可能包括您创建或销售列表，或者 Amazon 向您的账户发送资金。

Console

跟踪预留实例列表的状态

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航页面中，选择预留实例。
3. 在我的列表选项卡上，查找列表状态值。该选项卡还包含有关期限和标价的信息，以及列表中可用、待处理、售出和取消的实例数量明细。

Amazon CLI

跟踪预留实例列表的状态

使用 describe-reserved-instances-listings 命令，借助合适的筛选条件来获取有关您的列表的信息。

```
aws ec2 describe-reserved-instances-listings
```

PowerShell

跟踪预留实例列表的状态

使用 Get-EC2ReservedInstancesListing cmdlet。

```
Get-EC2ReservedInstancesListing
```

收款

Amazon从买方收到资金后，会向已售预留实例的注册所有者账户发送一封电子邮件。

Amazon 将自动清算所 (ACH) 电汇发送至您的指定银行账户。通常，此电汇在您的Reserved Instance已售出后的一天到三天内发生。支付每天只发生一次。在发放资金后，您将收到包含支付报告的电子邮件。请记住，在 Amazon 从您的银行收到确认信息后，您才能收到付款。这可能需要长达两周的时间。

在查看 预留实例 时，仍会显示已销售的 Reserved Instance。

预留实例的现金付款通过电汇转账直接进入您的银行账户。Amazon 向您收取您在预留实例 Marketplace 中出售的每个预留实例的总预付价格 12% 的服务费。

与买方共享的信息

当您在预留实例 Marketplace 中出售时，Amazon 将按照美国的规章在买方声明上分享您的公司法律名称。此外，如果买家因发票或其他税务相关的原因需要联系您而致电 Amazon Web Services 支持，那么 Amazon 可能需要向买家提供您的电子邮件地址，这样买家就能与您直接联系。

出于同样的原因，买方的邮政编码和国家/地区信息将在支付报告中提供给卖方。作为卖家，您可能需要在汇给政府任何必要的交易税 (例如销售税和增值税) 时附带此信息。

Amazon 不能提供税务建议，但如果您的税务专家确定您另外需要特定的信息，请联系 Amazon Web Services 支持。

修改 预留实例

当需求改变时，可以修改标准或可转换预留实例并继续利用账单优势。您可以修改可用区、实例大小（在相同的实例系列和世代中），以及预留实例的范围等属性。

> **Note**
>
> 您还可以将可转换预留实例交换为具有不同配置的其他可转换预留实例。有关更多信息，请参阅交换 可转换预留实例。

可以修改全部或部分预留实例。可以将原始 预留实例 分为两个或更多新的 预留实例。例如，如果您在 us-east-1a 中有 10 个实例的预留，并决定将其中 5 个实例移至 us-east-1b，则修改请求会生成两个新的预留实例 - 一个用于 us-east-1a 中的 5 个实例，另一个用于 us-east-1b 中的 5 个实例。

还可以将两个或更多 预留实例 合并 成单个 Reserved Instance。例如，如果有四个 t2.small 均为预留实例，则可以将其合并以创建单个 t2.large Reserved Instance。有关更多信息，请参阅对于修改实例大小的支持。

修改之后，预留实例的定价权益仅适用于与新参数匹配的实例。例如，如果您更改预留的可用区，则容量预留和定价优势自动应用到在新可用区中使用的实例。除非您的账户有其他适用的预留，否则将按照按需费率对不再符合新参数的实例收费。

如果您的修改请求成功：

- 修改的预留会立即生效，并且定价优惠将于进行修改请求时这一小时的开始应用于新实例。例如，如果您在晚上 9:15 成功修改了预留，则定价优惠将在晚上 9:00 转移到新实例。您可以使用 describe-reserved-instances 命令，获取修改后的预留实例的生效日期。
- 原始预留将停用。其结束日期是新预留的开始日期，而新预留的结束日期与原始Reserved Instance 的结束日期相同。如果您修改一个剩余期限为 16 个月的三年期预留，则修改后得到的预留是为期 16 个月的预留，其结束日期与原始预留相同。
- 已修改的预留将列出 0 美元固定价格，而不是原始预留的固定价格。
- 已修改的预留实例的固定价格不影响您的账户的折扣定价套餐计算，后者基于原始预留的固定价格。

如果修改请求失败，预留实例会保持其原始配置，并立即对其他修改请求可用。

修改不会产生任何费用，因此您不会收到任何新账单或发票。

您可以根据自己的需要随时修改预留，但是不能在提交之后更改或取消挂起的修改请求。修改成功完成后，如果需要，您可以提交另一个修改请求，以回滚您所做的任何更改。

目录

- 修改的要求和限制
- 对于修改实例大小的支持
- 提交修改请求
- 修改请求故障排除

修改的要求和限制

您可以按如下方式修改这些属性。

可修改的属性	支持的平台	限制和注意事项
在相同区域内更改可用区	Linux 和 Windows	-
将范围从可用区更改到区域以及反之	Linux 和 Windows	可用区预留实例的范围限定为可用区，并在该可用区中预留容量。如果您将范围从可用区更改为区域（即从可用区范围到区域性），则会失去容量预留优势。 区域性预留实例的范围限定为区域。预留实例折扣适用于在该区域任何可用区中运行的实例。此外，预留实例折扣适用于所选实例系列中所有大小的实例用量。如果您将范围从区域更改为可用区（即从区域性到可用区范围），则会失去可用区灵活性和实例大小灵活性（如果适用）。 有关更多信息，请参阅 如何应用预留实例折扣。
更改相同实例系列和世代中的实例大小	仅限 Linux/UNIX 实例大小灵活性不适用于其他平台上的 预留实例，这些平台包括含有 SQL Server Standard 的 Linux、含有 SQL Server Web 的 Linux、含有 SQL Server Enterprise 的	预留必须使用默认租赁。某些实例系列不受支持，因为没有其他大小可用。有关更多信息，请参阅 对于修改实例大小的支持。

可修改的属性	支持的平台	限制和注意事项
	Linux、Red Hat Enterprise Linux、SUSE Linux、Windows、含有 SQL Standard 的 Windows、含有 SQL Server Enterprise 的 Windows 和含有 SQL Server Web 的 Windows。	

要求

Amazon EC2 处理您的修改请求的前提是，对于您的新配置（如果适用），有足够的容量，同时满足以下条件：

- 在您购买 Reserved Instance 时或在此之前，无法对其进行修改
- Reserved Instance 必须是活动的
- 不能有待处理的修改请求
- 预留实例未列在预留实例 Marketplace 中
- 原始预留的实例大小占用空间必须与新配置匹配。有关更多信息，请参阅对于修改实例大小的支持。
- 原始 预留实例 要么全部是标准 预留实例，要么全部是 可转换预留实例，不能每种类型都有一些
- 原始 预留实例 如果是标准 预留实例，则必须在相同的时间过期
- 要修改实例大小，预留实例必须支持实例大小灵活性。关于不支持实例大小灵活性的预留实例列表，请参阅实例大小灵活性。

对于修改实例大小的支持

在满足以下要求时可以修改Reserved Instance的实例大小。

要求

- 平台是 Linux/UNIX。
- 您必须在相同 实例系列（用字母表示，例如 T）和 世代（用数字指示，例如 2）中选择其他实例大小。

例如，您可以将预留实例从 t2.small 修改为 t2.large，因为它们属于同一 T2 系列和世代。但您不能将预留实例从 T2 修改为 M2 或从 T2 修改为 T3，因为在这两个示例中，目标实例系列和世代与原始预留实例不同。

- 只有在预留实例支持实例大小灵活性时，才可以修改预留实例的实例大小。关于不支持实例大小灵活性的预留实例列表，请参阅实例大小灵活性。
- 您不能修改 t1.micro 实例的预留实例的实例大小，因为 t1.micro 仅有一个大小。
- 原始和新 Reserved Instance 必须具有相同的实例大小占用空间。

目录

- 实例大小占用空间
- 裸机实例的标准化因子

实例大小占用空间

每个Reserved Instance都有实例大小占用空间，该空间由预留中实例大小的标准化因子和实例数量决定。修改 Reserved Instance 中的实例大小时，新配置的占用空间必须与原始配置相匹配，否则不会处理修改请求。

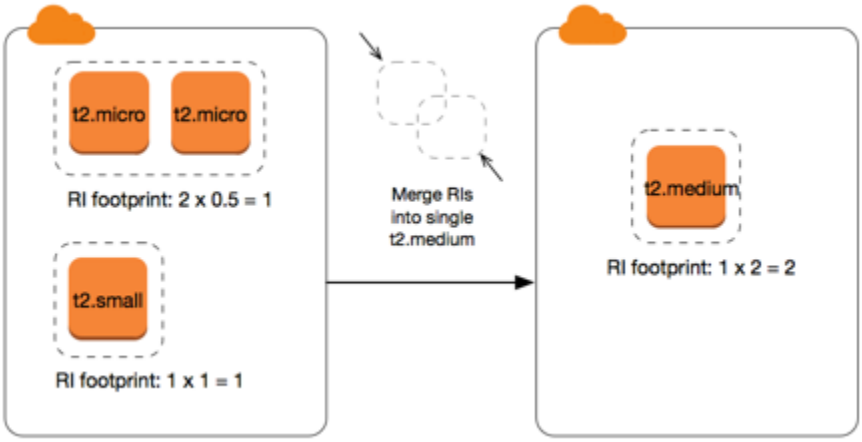
要计算Reserved Instance的实例占用空间大小，请将实例数量乘以标准化因子。在 Amazon EC2 控制台中，标准化因子用单位数来度量。下表描述了一个实例系列中实例大小的标准化因子。例如，t2.medium 的标准化因子为 2，因此 4 个 t2.medium 实例的预留具有 8 个单位的占用空间。

实例大小	标准化因子
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16

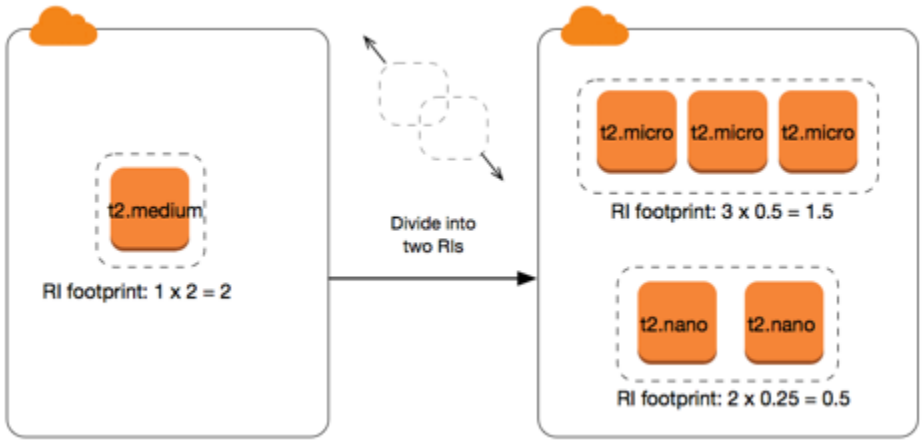
实例大小	标准化因子
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256
48xlarge	384
56xlarge	448
112xlarge	896

只要预留的实例大小占用空间保持不变，您就可以将预留分配给相同实例系列中的不同实例大小。例如，您可以将 1 个 t2.large（1 个 4 单位）实例的预留划分为 4 个 t2.small（4 个 1 单位）实例。同样，您可以将 4 个 t2.small 实例的预留合并为 1 个 t2.large 实例。但是，您不能将两个 t2.small 实例的预留更改为一个 t2.large 实例，因为新预留的占用空间（4 个单位）大于原始预留的占用空间（2 个单位）。

在以下示例中，您有 1 个具有 2 个 t2.micro 实例（1 个单位）的预留和 1 个具有 1 个 t2.small 实例（1 个单位）的预留。如果您将这些预留合并到具有 1 个 t2.medium 实例（2 个单位）的单个预留中，则新预留的占用空间等于合并预留的占用空间。



您还可以修改预留以将其拆分为多个预留。在以下示例中，您有 1 个具有 t2.medium 实例（2 个单位）的预留。您可以将该预留划分为两个预留，其中之一具有 2 个 t2.nano 实例（.5 个单位），另一个具有 3 个 t2.micro 实例（1.5 个单位）。



裸机实例的标准化因子

您可以使用相同实例系列中的其他大小，通过 metal 实例修改预留。同样，您可以使用相同实例系列中的 metal 大小，通过裸机实例以外的实例修改预留。通常，裸机实例的大小与相同实例系列中可用的最大实例大小相同。例如，i3.metal 实例的大小与 i3.16xlarge 实例的大小相同，因此它们具有相同的标准化因子。

下表描述了具有裸机实例的实例系列中裸机实例大小的标准化因子。metal 实例的标准化因子取决于实例系列，与其他实例大小不同。

实例大小	标准化因子
a1.metal	32

实例大小	标准化因子
m5zn.metal | x2iezn.metal z1d.metal	96
c6g.metal | c6gd.metal | i3.metal | m6g.metal | m6gd.metal | r6g.metal | r6gd.metal | x2gd.metal	128
c5n.metal	144
c5.metal | c5d.metal | i3en.metal | m5.metal | m5d.metal | m5dn.metal | m5n.metal | r5.metal | r5b.metal | r5d.metal | r5dn.metal | r5n.metal	192
c6i.metal | c6id.metal | m6i.metal | m6id.metal | r6d.metal | r6id.metal	256
u-18tb1.metal | u-24tb1.metal	448
u-6tb1.metal | u-9tb1.metal | u-12tb1.metal	896

例如，i3.metal 实例的标准化因子为 128。如果您购买 i3.metal 默认租期 Amazon Linux/Unix Reserved Instance，则可以按照如下方式划分预留：

- i3.16xlarge 的大小与 i3.metal 实例的大小相同，因此其标准化因子为 128 (128/1)。一个 i3.metal 实例的预留可以修改到一个 i3.16xlarge 实例中。
- i3.8xlarge 的大小是 i3.metal 实例大小的一半，因此其标准化因子为 64 (128/2)。一个 i3.metal 实例的预留可以划分到两个 i3.8xlarge 实例中。
- i3.4xlarge 的大小是 i3.metal 实例大小的四分之一，因此其标准化因子为 32 (128/4)。一个 i3.metal 实例的预留可以划分到四个 i3.4xlarge 实例中。

提交修改请求

在修改预留实例之前，请确保已阅读适用的限制。在您修改实例大小之前，请计算所要修改的原始预留的总实例大小占用空间，并确保该值与新配置的总实例大小占用空间相匹配。

Console

修改预留实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在 Reserved Instances (预留实例) 页面上，选择一个或多个要修改的预留实例，然后依次选择 Actions (操作) 和 Modify Reserved Instances (修改预留实例)。

   如果 预留实例 不处于活动状态或无法修改，则 Modify 预留实例 (修改预留实例) 处于禁用状态。
3. 修改表中的第一个条目显示选定 预留实例 的属性，而且其下方至少有一个目标配置。Units 列显示总实例大小占用空间。单击各个新配置的 Add 以添加。根据需要修改每个配置的属性。

   - Scope (范围)：选择配置是应用于可用区还是整个区域。
   - Availability Zone：选择所需的可用区。不适用于区域性预留实例。
   - 实例类型：选择所需的实例类型。组合配置必须等于原始配置的实例大小占用空间。
   - Count (数量)：指定实例数。要将预留实例拆分为多个配置，请减少数量，选择 Add (添加)，然后为其他配置指定数量。例如，如果单个配置的数量为 10，则可以将其数量更改为 6，并添加数量为 4 的配置。此过程在激活新的预留实例后会停用原始的Reserved Instance。
4. 选择 Continue (继续)。
5. 指定好目标配置之后，要确认您的修改选择，请选择 Submit modifications (提交修改)。
6. 您可以在 预留实例 屏幕中通过查看 State (状态) 列来确定修改请求的状态。有以下可能状态。

   - active (活动)（等待修改）— 原始预留实例的转换状态
   - 停用（等待修改）— 创建新预留实例时原始预留实例的转换状态。
   - 停用 — 已成功修改和替换预留实例
   - 活动 — 以下选项之一：
     - 从成功的修改请求创建的新预留实例
     - 修改请求失败后的原始预留实例

Amazon CLI

修改预留实例

使用 modify-reserved-instances 命令。您可以在 JSON 文件中提供配置详细信息。

```
aws ec2 modify-reserved-instances \
    --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 \
    --target-configurations file://configuration.json
```

获取修改请求的状态

使用 describe-reserved-instances-modifications 命令。状态为 processing、fulfilled 或 failed。

```
aws ec2 describe-reserved-instances-modifications \
    --reserved-instances-modification-ids rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687
 \
    --query ReservedInstancesModifications[].Status
```

PowerShell

修改预留实例

使用 Edit-EC2ReservedInstance cmdlet。您可以在类型为 Amazon.EC2.Model.ReservedInstancesConfiguration 的对象中提供配置详细信息。

```
Edit-EC2ReservedInstance `
    -ReservedInstancesId b847fa93-e282-4f55-b59a-1342f5bd7c02 `
    -TargetConfiguration $configuration
```

获取修改请求的状态

使用 Get-EC2ReservedInstancesModification cmdlet。状态为 processing、fulfilled 或 failed。

```
Get-EC2ReservedInstancesModification `
    -ReservedInstancesModificationId rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687 | `
    Select Status
```

修改请求故障排除

如果您请求的目标配置设置是唯一的，则您会收到正在处理该请求的消息。此时，Amazon EC2 仅确定了修改请求的参数有效。在处理过程中，您的修改请求仍然可能因无可用容量而失败。

在某些情况下，您可能会收到一个指示修改请求未完成或失败的消息而不是确认。使用此类消息中的信息作为重新提交另一个更改请求的起点。提交请求前，请确保您已阅读适用的限制。

并非所有选择的预留实例都可以进行修改处理

Amazon EC2 会确定并列出无法修改的 预留实例。如果收到与此类似的消息，请转到 Amazon EC2 控制台中的 Reserved Instances (预留实例) 页面，查看 预留实例 的信息。

处理修改请求时出错

您提交了一个或多个预留实例进行修改，但无法处理您的任何请求。根据您修改的预留数量，您可以获取不同版本的消息。

Amazon EC2 会显示无法处理请求的原因。举例来说，您可能已经为想要修改的 预留实例 的一个或更多子集指定了相同的目标配置（可用区和平台的组合）。尝试重新提交修改请求，但确保预留的实例详细信息是匹配的，并确保修改的所有子集的目标配置是唯一的。

交换 可转换预留实例

您可以将一个或多个 可转换预留实例 与具有不同配置的其他 可转换预留实例（包括实例系列、操作系统和租期）进行交换。执行交换的次数没有限制，前提是新的可转换预留实例的值等于或高于要交换的可转换预留实例的值。

在交换可转换预留实例时，您当前预留的实例数与新可转换预留实例的配置的实例数（其涵盖的值相等或更高）进行交换。Amazon EC2 计算您可以通过交换获得的预留实例数量。

您不能交换标准 预留实例，但可对其进行修改。有关更多信息，请参阅修改 预留实例。

目录

- 交换可转换预留实例的要求
- 计算 可转换预留实例 交换
- 合并 可转换预留实例
- 交换部分 可转换预留实例
- 提出交换请求

交换可转换预留实例的要求

如果满足以下条件，Amazon EC2 将处理您的交换请求。您可转换预留实例必须：

- 处于活动状态
- 没有以前等待处理的交换请求
- 距离到期至少还剩 24 小时

以下规则适用：

- 可转换预留实例必须与当前由 Amazon 提供的其他可转换预留实例交换。
- 可转换预留实例 与特定区域关联，在预留期限内是固定的。您不能将可转换预留实例与其他区域中的可转换预留实例进行交换。
- 要交换区域可转换预留实例，Amazon 必须为该区域中的新实例类型提供足够的容量。
- 一次只能将一个或多个 可转换预留实例 与一个 可转换预留实例 交换。
- 要交换部分 可转换预留实例，您可以修改它以将其拆分为两个或更多预留，然后将一个或多个预留与新 可转换预留实例 交换。有关更多信息，请参阅交换部分 可转换预留实例。有关修改 预留实例的更多信息，请参阅修改 预留实例。
- 预付全费的 可转换预留实例 可用来交换预付部分费用的 可转换预留实例，反之亦然。

> **Note**
>
> 如果交换所需的总预付款（调整费用）少于 0.00 USD，Amazon会自动向您提供可转换预留实例 中可确保调整费用大于等于 0.00 USD 的实例数。

> **Note**
>
> 如果新可转换预留实例的总价值（预付价格 + 每小时价格 * 剩余小时数）少于交换的可转换预留实例的总价值，Amazon会自动向您提供可转换预留实例中可确保总价值大于等于交换的可转换预留实例的总价值的实例数。

- 要享受更优惠的定价，您可以将无预付费用的 可转换预留实例 交换为预付全费或预付部分费用的 可转换预留实例。
- 不能将预付全部费用和预付部分费用的可转换预留实例交换为无预付费用的可转换预留实例。
- 仅当新 可转换预留实例 的每小时价格大于等于交换的 可转换预留实例 的每小时价格时，才能将一个无预付费用的 可转换预留实例 交换为另一个无预付费用的 可转换预留实例。

> **Note**
>
> 如果新可转换预留实例的总价值（预付价格 + 每小时价格 * 剩余小时数）少于交换的可转换预留实例的总价值，Amazon会自动向您提供可转换预留实例中可确保总价值大于等于交换的可转换预留实例的总价值的实例数。

- 如果交换到期日期不同的多个 可转换预留实例，则新 可转换预留实例 的到期日期是将来最晚的日期。
- 如果您交换了单个 可转换预留实例，则它必须与新 可转换预留实例 具有相同的期限（1 年或 3 年）。如果合并期限不同的多个 可转换预留实例，则新 可转换预留实例 期限为 3 年。有关更多信息，请参阅 合并 可转换预留实例。
- 当 Amazon EC2 交换可转换预留实例时，它会停用相关的预留，并将结束日期传输到新预留。交换后，Amazon EC2 会将旧预留的结束日期和新预留的开始日期设置为等于交换日期。例如，如果您交换一个剩余期限为 16 个月的三年期预留，则新预留是为期 16 个月的预留，其结束日期与您交换的可转换预留实例相同。

计算 可转换预留实例 交换

交换可转换预留实例是免费的。但是，您可能需要支付调整费用，即您拥有的可转换预留实例与通过交换收到的新可转换预留实例之间差额的按比例预付费用。

每个可转换预留实例都具有标价。此价目表值与您想要的可转换预留实例的价目表值比较，用于确定您可通过交换收到的实例预留数。

例如：您有 1 个 35 美金标价的可转换预留实例，您希望交换为标价为 10 美金的全新实例类型。

```
$35/$10 = 3.5
```

您可以将 可转换预留实例 交换为三个 10 美元的 可转换预留实例。无法购买半预留；因此必须购买额外的可转换预留实例才能涵盖剩余部分：

```
3.5 = 3 whole Convertible Reserved Instances + 1 additional Convertible Reserved
 Instance
```

第四个可转换预留实例与其他三个具有相同的结束日期。如果要交换部分或全部预付可转换预留实例，则需要支付第四个预留的调整费用。如果 可转换预留实例 的剩余预付费用为 500 USD，新预留通常按比例分摊为 600 USD，则需要支付 100 USD。

```
$600 prorated upfront cost of new reservations - $500 remaining upfront cost of old
 reservations = $100 difference
```

合并 可转换预留实例

如果合并两个或更多可转换预留实例，则新可转换预留实例的期限必须与旧可转换预留实例的期限相同，或者与可转换预留实例中的最高期限相同。新可转换预留实例的到期日期是未来有效时间最长的到期日期。

例如，您的账户中有以下可转换预留实例：

Reserved Instance ID	租期	到期日期
aaaa1111	1 年	2018-12-31
bbbb2222	1 年	2018-07-31
cccc3333	3 年	2018-06-30
dddd4444	3 年	2019-12-31

- 您可以合并 aaaa1111 和 bbbb2222 并将它们与 1 年期 可转换预留实例 交换。您无法将它们与 3 年期可转换预留实例交换。新可转换预留实例的到期日期为 2018-12-31。
- 您可以合并 bbbb2222 和 cccc3333 并将它们与 3 年期 可转换预留实例 交换。您无法将它们与 1 年期可转换预留实例交换。新可转换预留实例的到期日期为 2018-07-31。
- 您可以合并 cccc3333 和 dddd4444 并将它们与 3 年期 可转换预留实例 交换。您无法将它们与 1 年期可转换预留实例交换。新可转换预留实例的到期日期为 2019-12-31。

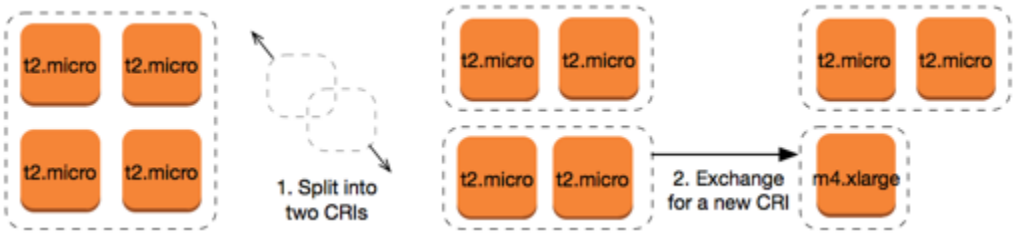
交换部分 可转换预留实例

您可以使用修改过程将 可转换预留实例 拆分为较小的预留，然后将一个或多个新预留与新 可转换预留实例 交换。以下示例演示了如何执行此操作。

Example示例：包含多个实例的可转换预留实例

在本示例中，您有一个在预留中有四个实例的 t2.micro 可转换预留实例。将两个 t2.micro 实例与一个 m4.xlarge 实例交换：

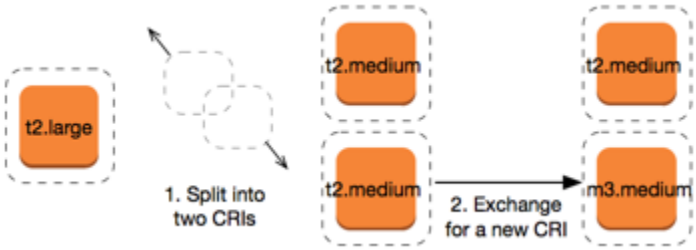
1. 修改 t2.micro 可转换预留实例，方法为将其拆分为两个 t2.micro 可转换预留实例，每一个都包含两个实例。
2. 将其中一个新 t2.micro 可转换预留实例 与一个 m4.xlarge 可转换预留实例 交换。



Example示例：包含单个实例的可转换预留实例

在本示例中，您拥有一个 t2.large 可转换预留实例。将其更改为一个较小的 t2.medium 实例和一个 m3.medium 实例：

1. 修改 t2.large 可转换预留实例，方法为将其拆分为两个 t2.medium 可转换预留实例。单个 t2.large 实例具有两个 t2.medium 实例相同的实例大小占用空间。
2. 将其中一个新 t2.medium 可转换预留实例 与一个 m3.medium 可转换预留实例 交换。



有关更多信息，请参阅对于修改实例大小的支持和提出交换请求。

提出交换请求

您可以交换可转换预留实例。交换的预留实例已停用。

Console

交换可转换预留实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 Reserved Instances (预留实例)，选择要交换的 可转换预留实例，然后依次选择 Actions (操作) 和 Exchange Reserved Instance (交换预留实例)。
3. 选择所需配置的属性，然后选择 Find offering (查找产品)。
4. 选择一个新的 可转换预留实例。在屏幕底部，您可以查看通过交换收到的 预留实例 数和任何额外费用。

5. 当您选择了符合需求的 可转换预留实例 时，请选择 Review (审核)。
6. 选择 Exchange (交换)，然后选择 Close (关闭)。

Amazon CLI

交换可转换预留实例

1. 使用 describe-reserved-instances-offerings 命令查找符合需求的新可转换预留实例。
2. 使用 get-reserved-instances-exchange-quote 命令获取交换的报价。这包括通过交换获得的预留实例数以及交换的调整费用：
3. 使用 accept-reserved-instances-exchange-quote 命令执行交换。

PowerShell

交换可转换预留实例

1. 使用 Get-EC2ReservedInstancesOffering cmdlet 查找符合需求的新可转换预留实例。
2. 使用 GetEC2-ReservedInstancesExchangeQuote cmdlet 获取交换的报价。这包括通过交换获得的预留实例数以及交换的调整费用：
3. 使用 Approve-EC2ReservedInstancesExchangeQuote cmdlet 进行交换

预留实例限额

您可以每月购买新的预留实例。您每月可购买的新预留实例数由每月配额决定，如下所示：

配额描述	默认配额
新的区域性预留实例	每个区域每月 20 个
新的可用区预留实例	每个可用区每月 20 个

例如，在带有三个可用区的区域中，默认配额为每月 80 个新预留实例，计算方法如下：

- 该区域有 20 个区域预留实例
- 加 60 个可用区预留实例（三个可用区各有 20 个）

处于 `running` 状态的实例将计入您的限额。处于 `pending`、`stopping`、`stopped` 和 `hibernated` 状态的实例不会计入您的限额。

查看您已购买的预留实例数量

您购买的预留实例数量通过 Instance count（实例计数）字段（控制台）或 `InstanceCount` 参数（Amazon CLI）表示。当您购买新的预留实例时，配额会根据实例总数进行衡量。例如，如果您购买了实例数量为 10 的单个预留实例配置，则计入配额的购买数量为 10，而不是 1。

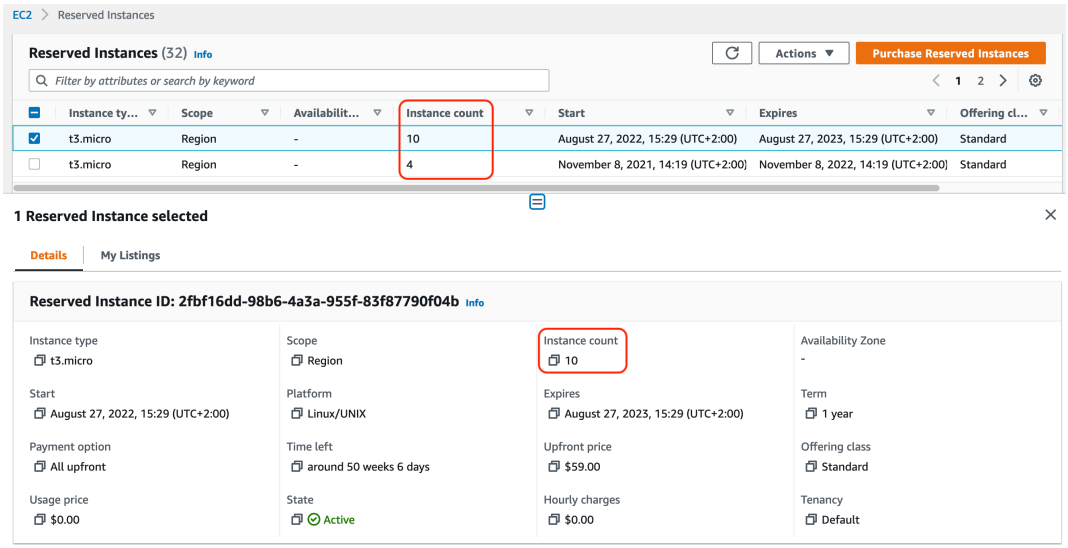
您可以使用 Amazon EC2 或 Amazon CLI 查看已购买的预留实例数量。

Console

查看您已购买的预留实例数量

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Reserved Instances。
3. 从表中选择预留实例配置，然后查看 Instance count（实例计数）字段。

   在下列屏幕截图中，选定的行表示 `t3.micro` 实例类型的单个预留实例配置。表视图中的 Instance count（实例计数）列和详细信息视图中的 Instance count（实例计数）字段（在屏幕截图中划出）表示此配置有 10 个预留实例。



Amazon CLI

查看您已购买的预留实例数量

使用 describe-reserved-instances 命令并指定预留实例配置的 ID。

```
aws ec2 describe-reserved-instances \
    --reserved-instances-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --output table
```

下面是示例输出。InstanceCount 字段表示此配置有 10 个预留实例。

```
-------------------------------------------------------------------
|                    DescribeReservedInstances                    |
+-----------------------------------------------------------------+
||                        ReservedInstances                      ||
|+----------------------+----------------------------------------+|
||  CurrencyCode        |  USD                                   ||
||  Duration            |  31536000                              ||
||  End                 |  2023-08-27T13:29:44+00:00             ||
||  FixedPrice          |  59.0                                  ||
||  InstanceCount       |  10                                    ||
||  InstanceTenancy     |  default                               ||
||  InstanceType        |  t3.micro                              ||
||  OfferingClass       |  standard                              ||
||  OfferingType        |  All Upfront                           ||
||  ProductDescription  |  Linux/UNIX                            ||
||  ReservedInstancesId |  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ||
||  Scope               |  Region                                ||
||  Start               |  2022-08-27T13:29:45.938000+00:00      ||
||  State               |  active                                ||
||  UsagePrice          |  0.0                                   ||
|+----------------------+----------------------------------------+|
|||                       RecurringCharges                      |||
||+---------------------------------+--------------------------+||
|||  Amount                         |  0.0                     |||
|||  Frequency                      |  Hourly                  |||
||+---------------------------------+--------------------------+||
```

PowerShell

查看您已购买的预留实例数量

使用 Get-EC2ReservedInstance Cmdlet 并指定预留实例配置的 ID。

```
Get-EC2ReservedInstance -ReservedInstancesId a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

下面是示例输出。InstanceCount 字段表示此配置有 10 个预留实例。

```
AvailabilityZone    :
CurrencyCode        : USD
Duration            : 31536000
End                 : 1/12/2017 8:57:08 PM
FixedPrice          : 0
InstanceCount       : 10
InstanceTenancy     : default
InstanceType        : t3.medium
OfferingClass       : standard
OfferingType        : All Upfront
ProductDescription  : Windows
RecurringCharges    : {}
ReservedInstancesId : a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
Scope               : Region
Start               : 10/12/2016 4:00:00 PM
State               : active
Tags                : {}
UsagePrice          : 0
```

注意事项

区域性预留实例账单折扣适用于正在运行的按需型实例。默认按需型实例 限制为 20。购买区域性预留实例时，不能超出正在运行的按需型实例限制。例如，如果您已有 20 个正在运行的按需型实例，并且购买了 20 个区域性预留实例，则使用 20 个区域性预留实例将折扣应用于 20 个正在运行的按需型实例。即使您购买了更多的区域性预留实例，也无法启动更多的实例，因为已达到按需型实例限制。

在购买区域性预留实例 之前，请确保 个按需型实例 限制匹配或超出您打算拥有的区域性预留实例 的数量。如果需要，请确保在购买更多区域性 预留实例 之前，请求增加 个按需实例 限制。

可用区预留实例（为特定可用区域购买的预留实例）提供容量预留以及折扣。购买地区性 预留实例时，可以超出 正在运行的 个按需实例 限制。例如，如果您已有 20 个正在运行的 按需型实例，并且购买了 20 个地区性 预留实例，则可以启动另外的 20 个 按需型实例，以区配地区性 预留实例 为您提供总共 40 个正在运行的实例的规范。

查看您的预留实例配额并请求增加配额

Amazon EC2 控制台提供了配额信息。您也可以请求增加配额。有关更多信息，请参阅查看当前限额和请求提高。

Spot Instances

竞价型实例是一种使用备用 EC2 容量的实例，以低于按需价格提供。由于竞价型实例允许您以极低的折扣请求未使用的 EC2 实例，这可能会显著降低您的 Amazon EC2 成本。竞价型实例的每小时价格称为 Spot 价格。每个可用区中的每种实例类型的 Spot 价格是由 Amazon EC2 设置的，并根据竞价型实例的长期供求趋势逐步调整。您的竞价型实例在容量可用时运行。

如果能灵活控制应用程序的运行时间并且应用程序可以中断，竞价型实例就是经济实惠之选。例如，竞价型实例非常适合数据分析、批处理作业、后台处理和可选的任务。有关更多信息，请参阅 Amazon EC2 Spot 实例。

有关 EC2 实例不同购买选项的比较，请参阅 Amazon EC2 账单和购买选项。

概念

在开始使用竞价型实例之前，应该熟悉以下概念：

- Spot 容量池 – 一组具有相同实例类型（如 `m5.large`）和可用区的未使用 EC2 实例。
- Spot 价格 – Spot 实例的当前每小时价格。
- Spot 实例请求 – 请求 Spot 实例。当有容量可用时，Amazon EC2 将满足您的请求。Spot 实例请求是一次性或者持续性请求。在与请求关联的竞价型实例中断之后，Amazon EC2 会自动重新提交持续性竞价型实例请求。
- EC2 实例再平衡建议 – Amazon EC2 发出实例再平衡建议信号，以通知您竞价型实例处于中断高风险。此信号使您有机会在现有或新的竞价型实例间主动再平衡工作负载，而无需等待两分钟的竞价型实例中断通知。
- 竞价型实例中断 – 如果 Amazon EC2 需要回收容量，Amazon EC2 会将您的竞价型实例停止、停止或休眠。Amazon EC2 将提供竞价型实例中断通知，这会在实例中断之前为其提供两分钟的警告。

竞价型实例与按需型实例的区别

下表列出了竞价型实例与按需型实例之间的主要区别。

	竞价型实例	On-Demand Instances
启动时间	只有竞价型实例请求处于活动状态并且有可用容量时才能立即启动。	只有发出手动启动请求并且有可用容量时才能立即启动。

	竞价型实例	On-Demand Instances
可用容量	如果没有可用容量，则竞价型实例请求会继续自动发起启动请求，直到有可用容量为止。	如果在发出启动请求时没有可用容量，您会收到容量不足错误 (ICE)。
每小时价格	竞价型实例的每小时价格根据长期的供需情况而有所调整。	按需型实例的每小时价格是静态的。
再平衡建议	当实例处于较高的中断风险时，Amazon EC2 为正在运行的竞价型实例发出信号。	您可以决定何时中断个按需型实例（停止、休眠或终止）。
实例中断	您可以停止和启动由 Amazon EBS 支持的竞价型实例。此外，如果容量不再可用，Amazon EC2 可能会中断单个竞价型实例。	您可以决定何时中断个按需型实例（停止、休眠或终止）。

定价和节省

您可以为竞价型实例支付 Spot 价格，该价格由 Amazon EC2 设置，并根据竞价型实例的长期供求趋势逐步进行调整。您的竞价型实例将持续运行，直到您将其终止、容量不再可用或您的 Amazon EC2 Auto Scaling 组在横向缩减期间将其终止为止。

如果您或 Amazon EC2 中断正在运行的竞价型实例，您将按使用的秒数或整个小时付费，或者您不收取任何费用，具体取决于所使用的操作系统以及竞价型实例的中断方。有关更多信息，请参阅 中断的竞价型实例的计费。

节省计划不涵盖竞价型实例。如果您使用了节省计划，那么除了使用竞价型实例已经节省的费用外，此计划不会节省额外的费用。此外，您在竞价型实例上的支出并不适用于计算类节省计划中的承诺。

查看价格

要查看各个 Amazon Web Services 区域和实例类型的当前最低竞价型实例价格（每 5 分钟更新一次），请参阅 Amazon EC2 竞价型实例定价 页面。

要查看过去三个月竞价型实例价格的历史记录，请使用 Amazon EC2 控制台或 describe-spot-price-history 命令。有关更多信息，请参阅 查看竞价型实例定价历史记录。

我们将可用区独立地映射到每个 Amazon Web Services 账户的代码。因此，不同账户的相同可用区代码 (如 `us-west-2a`) 可能会返回不同的结果。

查看节省

您可以查看使用单个 竞价型实例集 或所有竞价型实例时通过使用竞价型实例所节省的成本。您可以查看过去一小时或过去三天的节省，还可以查看每 vCPU 小时和每内存 (GiB) 小时的平均费用。节省是估算的，因为未算入您使用期间的计费调整，所以可能与实际的节省有所差异。有关查看节省信息的更多信息，请参阅通过购买竞价型实例实现节省。

查看账单

您的账单提供了服务使用量的详细信息。有关更多信息，请参阅 Amazon Billing 用户指南中的查看您的账单。

Amazon EC2 竞价型实例的最佳实践

Amazon EC2 通过竞价型实例提供 Amazon Web Services 云 中的备用 EC2 计算容量，与按需型实例价格相比，这种实例最高可以节省 90% 的成本。按需型实例和竞价型实例之间的唯一区别是，当 Amazon EC2 需要回收容量时，Amazon EC2 可以中断竞价型实例并提前两分钟发出通知。要确保在使用竞价型实例时获得最佳体验，必须了解并运用最佳使用实践。

竞价型实例建议用于无状态、容错且灵活的应用程序。例如，竞价型实例非常适合大数据、容器化工作负载、CI/CD、无状态 Web 服务器、高性能计算 (HPC) 和渲染工作负载。

在运行时，竞价型实例与按需型实例完全相同。不过，Spot 不能保证您可以将运行的实例保持足够长的时间以完成工作负载。Spot 也不能保证您可以立即使用查找的实例，或者不能保证您始终可以获得请求的总容量。此外，竞价型实例中断和容量可能会随着时间的推移而发生变化，因为竞价型实例可用性根据供需关系而发生变化，过去的性能并不能保证将来的结果。

竞价型实例不适用于不灵活、有状态、非容错或在实例节点之间紧密耦合的工作负载。对于不能容忍整个目标容量偶尔不完全可用的工作负载，也不建议使用竞价型实例。虽然遵循竞价型最佳实践来灵活处理实例类型和可用区可以实现高可用性，但这不能保证始终有可用的容量，因为对按需型实例的需求激增可能会中断竞价型实例上的工作负载。

强烈建议不要在这些工作负载中使用竞价型实例，也不要尝试失效转移到按需型实例，来处理中断或一段时间内不可用的问题。失效转移到按需型实例可能会无意中导致其他竞价型实例发生中断。此外，如果实例类型和可用区组合的竞价型实例发生中断，可能会很难获得具有相同组合的按需型实例。

无论您是经验丰富的 Spot 用户还是竞价型实例新用户，如果您当前遇到竞价型实例中断或可用性问题，我们建议您按照以下最佳实践以获得使用 Spot 服务的最佳体验。

Spot 最佳实践

- 准备各个实例以处理中断
- 灵活地选择实例类型和可用区
- 使用基于属性的实例类型选择
- 使用竞价放置分数来确定最佳区域和可用区
- 使用 EC2 自动扩缩组或 EC2 实例集管理总容量
- 使用价格和容量优化分配策略
- 使用集成的 Amazon 服务以管理竞价型实例
- 使用哪种竞价型请求方法最好？

准备各个实例以处理中断

正常处理竞价型实例中断的最佳方法是，设计应用程序以提供容错能力。为此，您可以利用 EC2 实例再平衡建议和竞价型实例中断通知。

EC2 实例再平衡建议是一个信号，可在竞价型实例处于较高的中断风险时通知您。该信号使您有机会在两分钟的竞价型实例中断通知之前主动管理竞价型实例。您可以决定将工作负载再平衡到不处于较高中断风险的新的或现有的竞价型实例。借助自动扩缩组和 EC2 实例集中的容量再平衡功能，您可以轻松使用这个新信号。

竞价型实例中断通知是在 Amazon EC2 中断竞价型实例之前的两分钟发出的警告。如果工作负载为“时间灵活型”负载，您可以将竞价型实例配置为在中断时停止或休眠，而不是终止。Amazon EC2 在中断时使竞价型实例自动停止或休眠，并在我们具有可用容量时自动恢复这些实例。

我们建议您在 Amazon EventBridge 中创建一个规则以捕获再平衡建议和中断通知，然后触发工作负载进度的检查点或正常处理中断。有关更多信息，请参阅监控再平衡建议信号。有关指导您如何创建和使用事件规则的详细示例，请参阅利用 Amazon EC2 Spot 实例中断通知。

有关更多信息，请参阅EC2 实例再平衡建议 和 竞价型实例中断。

灵活地选择实例类型和可用区

竞价型容量池是具有相同实例类型（如 `m5.large`）和可用区（如 us-east-1a）的未使用 EC2 实例的集合。您应该灵活地选择请求哪些实例类型，以及可以在哪些可用区中部署工作负载。这为 Spot 提供了更好的机会以查找和分配所需数量的计算容量。例如，如果您希望使用 c4、m5 和 m4 系列中的大型实例，则不要仅请求 `c5.large`。

根据您的特定需求，您可以评估可灵活地选择哪些实例类型以满足您的计算要求。如果可以纵向扩展工作负载，您应该在请求中包括更大的实例类型（更多的 vCPU 和内存）。如果您只能横向扩展，您应该包括较旧一代的实例类型，因为按需客户很少需要使用它们。

一条很好的经验法则是，对于每种工作负载，灵活地在至少 10 种实例类型之间进行选择。此外，还要确保所有可用区配置为在 VPC 中使用，并为工作负载选择了这些可用区。

使用基于属性的实例类型选择

使用基于属性的实例类型选择后，可以为要运行的工作负载指定实例属性，例如 vCPU、内存和存储。之后，EC2 Auto Scaling 或 EC2 实例集会自动识别并启动与指定属性匹配的实例。这省去了手动选择特定实例类型所需的工作，免于深入了解每种实例类型的产品。

此外，基于属性的实例类型选择支持在新发布的实例类型可用时自动使用这些实例类型。此举可确保无缝访问范围越来越广泛的竞价型实例容量。

基于属性的实例类型选择适用于可以灵活确定所运行实例类型的工作负载和框架，例如高性能计算（HPC）和大数据工作负载。

有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Create mixed instances group using attribute-based instance type selection，以及本指南中的指定 EC2 实例集或竞价型实例集的实例类型选择属性。

使用竞价放置分数来确定最佳区域和可用区

竞价型实例属于未使用的 EC2 容量，此类容量会因 EC2 的供需情况产生波动。因此，想在特定时间于特定地点获得所需的确切竞价型容量，并不总是可行。您可以使用竞价放置分数功能来应对这种不可预测性。此功能为更有可能拥有足够容量来满足竞价型容量需求的区域或可用区提供建议，不要求先在这些位置启动竞价型实例。

竞价放置分数适用于可以灵活确定实例类型及其可使用的区域或可用区的工作负载。您要做的就是指定所需的竞价型容量、实例类型要求，以及是否需要针对区域或可用区的建议。随后，每个区域或可用区会获得 1 到 10 的分数，表明在该位置成功预置请求的竞价型容量的可能性。10 分数表示竞价请求极有可能成功。

务必注意，竞价放置分数属于时间点建议，因为容量可能会随时间发生变化。其既不能保证可用容量，也不能预测中断风险。

您可以通过 Amazon EC2 控制台、Amazon CLI 或软件开发工具包使用竞价放置分数功能。有关更多信息，请参阅 竞价放置分数。

使用 EC2 自动扩缩组或 EC2 实例集管理总容量

Spot 允许您考虑总容量（包括 vCPU、内存、存储或网络吞吐量），而不是考虑单个实例。通过使用自动扩缩组和 EC2 实例集，您可以启动和维持目标容量，自动请求资源以替换任何中断或手动终止的资源。在配置自动扩缩组或 EC2 实例集时，您仅需要根据应用程序需求指定实例类型和目标容量。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南 中的 Auto Scaling 组和本用户指南中的创建 EC2 实例集。

使用价格和容量优化分配策略

Auto Scaling 组中的分配策略可以帮助您预置目标容量，而无需手动查找具有备用容量的 Spot 容量池。我们建议使用 `price-capacity-optimized` 策略，因为该策略自动从可用性最佳的竞价型容量池中预置实例，同时也具有最低价格。您也可以在 EC2 实例集中利用 `price-capacity-optimized` 分配策略。由于竞价型实例容量来自于具有最佳容量的池，因此，这会降低回收竞价型实例的可能性。有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Allocation strategies for multiple instance types 和本用户指南中的当工作负载拥有较高中断成本时。

使用集成的 Amazon 服务以管理竞价型实例

其他 Amazon 服务与 Spot 集成在一起以降低总体计算成本，而无需管理各个实例或队列。我们建议您为适用的工作负载考虑以下解决方案：Amazon EMR、Amazon Elastic Container Service、Amazon Batch、Amazon Elastic Kubernetes Service、Amazon SageMaker AI、Amazon Elastic Beanstalk 和 Amazon GameLift Servers。要了解这些服务的 Spot 最佳实践的更多信息，请参阅 Amazon EC2 竞价型实例 Workshops 网站。

使用哪种竞价型请求方法最好？

使用下表确定在请求竞价型实例时使用哪个 API。

API	何时使用？	使用案例	我应该使用这个 API 吗？
CreateAutoScalingGroup	• 您需要多个采用单一配置或混合配置的实例。 •	创建一个 Auto Scaling 组，以管理实例的生命周期，同时保持所需的实例数量。支持在指定的最小和最大限值之间的水平扩展（添加更多实例）。	是

API	何时使用？	使用案例	我应该使用这个 API 吗？
	您希望通过可配置的 API 自动执行生命周期管理。		
CreateFleet	• 您需要多个采用单一配置或混合配置的实例。 • 您希望自行管理实例生命周期。 • 如果不需要弹性伸缩，我们建议您使用 `instant` 类型的实例集。	在单个请求中创建一个由按需型实例和竞价型实例组成的实例集，并具有多个因实例类型、AMI、可用区或子网而异的启动规范。竞价型实例分配策略默认为每单位 `lowest-price` ，但您可以将其更改为 `price-capacity-optimized` 、`capacity-optimized` 或 `diversified` 。	是 - 如果不需要自动扩缩，则在 `instant` 模式下
RunInstances	• 您已经在使用 RunInstances API 启动按需型实例，而且只想通过更改单个参数来更改为启动竞价型实例。 • 您不需要多个实例类型不同的实例。	使用 AMI 和一个实例类型启动指定数量的实例。	否 - 因为 RunInstances 不允许在单个请求中混合多个实例类型

API	何时使用？	使用案例	我应该使用这个 API 吗？
RequestSpotFleet	• 我们强烈反对使用 RequestSpotFleet API，因为它是不具有计划投资的旧式 API。 • 如果想要管理实例生命周期，请使用 CreateFleet API。 • 如果不想管理实例生命周期，请使用 CreateAutoScalingGroup API。	请勿使用。RequestSpotFleet 是不具有计划投资的旧式 API。	否
RequestSpotInstances	• 我们强烈反对使用 RequestSpotInstances API，因为它是不具有计划投资的旧式 API。	请勿使用。RequestSpotInstances 是不具有计划投资的旧式 API。	否

竞价型实例的工作原理

要启动 Spot 实例，您可以创建 Spot 实例请求，或者 Amazon EC2 代表您创建 Spot 实例请求。竞价型实例在满足竞价型实例请求时启动。

您可以使用多种不同的服务启动竞价型实例。有关更多信息，请参阅 Amazon EC2 Spot 实例入门。在本用户指南中，我们介绍了以下使用 EC2 启动竞价型实例的方法：

- 您可以使用 Amazon EC2 控制台中的启动实例向导或 run-instances 命令创建竞价型实例请求。有关更多信息，请参阅 管理您的竞价型实例。
- 您可以创建 EC2 实例集，在其中指定所需数量的竞价型实例。Amazon EC2 代表您为 EC2 实例集中指定的每个竞价型实例创建一个竞价型实例请求。有关更多信息，请参阅创建 EC2 实例集。
- 您可以创建竞价型实例集请求，在其中指定所需数量的竞价型实例。Amazon EC2 代表您为竞价型实例集中指定的每个竞价型实例创建一个竞价型实例请求。有关更多信息，请参阅 创建竞价型实例集。

> **Note**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

如果有可用容量，则您的竞价型实例将启动。您的 Spot 实例会一直运行，直到您停止或终止它，或者 Amazon EC2 中断它（称为 Spot 实例中断）。Amazon EC2 可以在中断竞价型实例时使实例停止、终止或休眠。

当您使用竞价型实例时，您必须为中断做好准备。在竞价型实例需求增加或供应减少时，Amazon EC2 可能会中断您的竞价型实例。在 Amazon EC2 中断竞价型实例时，将提供竞价型实例中断通知，这会在 Amazon EC2 终止该实例之前为其提供两分钟的警告。无法为竞价型实例启用终止保护。有关更多信息，请参阅 竞价型实例中断。

内容

- 竞价型实例请求状态
- 在启动组中启动竞价型实例
- 在可用区组中启动竞价型实例
- 在 VPC 中启动竞价型实例
- 启动可突增性能实例
- 在单租户硬件上启动

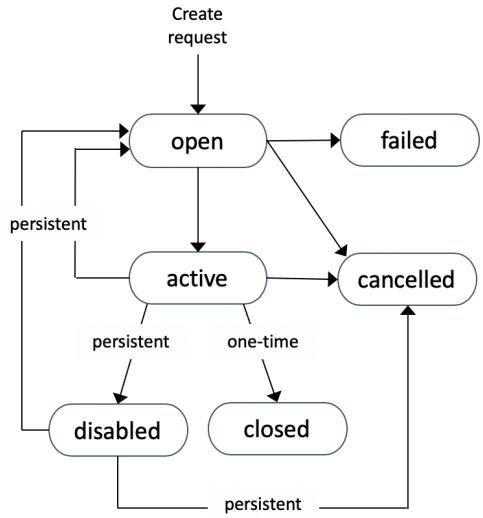
竞价型实例请求状态

竞价型实例请求可以处于以下某种状态：

- open – 请求正在等待执行。

- `active` – 请求已执行并有关联的竞价型实例。
- `failed` – 请求的一个或多个参数错误。
- `closed` – 竞价型实例被中断或终止。
- `disabled` – 您停止了竞价型实例。
- `cancelled` – 您取消了请求或请求已过期。

以下显示了请求状态之间的转换。请注意，转换取决于请求类型 (一次性还是持久性)。



一次性竞价型实例请求在 Amazon EC2 启动竞价型实例、请求过期前或者您取消请求前保持有效。当没有容量可用时，将终止您的竞价型实例并关闭竞价型实例请求。

持久性竞价型实例请求在过期或您取消它之前保持有效，即使该请求已完成也如此。当没有容量可用时，您的竞价型实例将会中断。在您的实例中断后，在再次具有可用的容量时，将会启动竞价型实例（如果已停止）或将其恢复（如果已休眠）。您可以停止竞价型实例，然后在有容量可用是重新启动实例。如果竞价型实例终止（无论竞价型实例处于停止状态还是正在运行状态），则会重新打开竞价型实例请求，并且 Amazon EC2 启动一个新的竞价型实例。有关更多信息，请参阅停止竞价型实例、启动竞价型实例和终止竞价型实例。

您可以跟踪竞价型实例请求的状态以及通过该状态启动的竞价型实例的状态。有关更多信息，请参阅获取竞价型实例请求的状态。

在启动组中启动竞价型实例

在竞价型实例请求中指定启动组，可以通知 Amazon EC2 只有在可以全部启动一组竞价型实例时才启动该组。此外，如果 Spot 服务必须终止启动组中的某个实例，它必须终止所有实例。不过，如果由您终止启动组中的一个或多个实例，Amazon EC2 不会终止该启动组中的剩余实例。

尽管此选项有用处，但是添加此约束会减少完成竞价型实例请求的几率并且增加竞价型实例被终止的几率。例如，启动组包括多个可用区中的实例。如果其中一个可用区中的容量减少且不再可用，则 Amazon EC2 会终止启动组的所有实例。

如果您创建了另一个成功的竞价型实例请求并指定与之前成功请求相同（现有）的启动组，则新实例将添加到该启动组中。以后，在该启动组的一个实例终止时，启动组中的所有实例均会终止，这包括第一次请求和第二次请求启动的实例。

在可用区组中启动竞价型实例

在竞价型实例请求中指定可用区组，可以通知 Amazon EC2 在同一可用区中启动一组竞价型实例。Amazon EC2 无需同时中断可用区组中的所有实例。如果 Amazon EC2 必须中断可用区组中的某个实例，剩余的实例仍保持运行。

虽然此选项非常有用，但添加此约束会减少完成您的竞价型实例请求的几率。

如果您指定了可用区组，但未在竞价型实例请求中指定可用区，则具体结果将取决于您所指定的网络。

默认 VPC

Amazon EC2 使用指定子网的可用区。如果您未指定子网，它会为您选择一个可用区及其默认子网，但不一定是价格最低的可用区。如果您删除了可用区的默认子网，则必须指定其他子网。

非默认 VPC

Amazon EC2 使用指定子网的可用区。

在 VPC 中启动竞价型实例

按照为按需型实例指定子网的相同方法，为您的竞价型实例指定子网。

- [默认 VPC] 如果希望在特定的低价格可用区中启动您的竞价型实例，您必须在竞价型实例请求中指定对应的子网。如果您没有指定子网，则 Amazon EC2 将为您选择一个子网，而该子网的可用区中的 Spot 价格不一定是最低的。
- [非默认 VPC] 您必须为您的竞价型实例指定子网。

启动可突增性能实例

T 实例类型是可突增性能实例。如果您使用可突增性能的实例类型启动竞价型实例，并计划立即短时间使用可突增性能的竞价型实例，且没有空闲时间来累积 CPU 积分，建议您以标准模式启动实例，以避免支付更高的费用。如果您以无限模式启动可突增性能的 Spot 实例并立即突增 CPU，您将会为突增花

费超额积分。如果使用实例的时间很短，使得实例没有时间积累 CPU 积分来支付超额积分，则您将在终止实例时为超额积分付费。

仅当实例的运行时间较长，足以积累进行突增的 CPU 积分时，针对可突增性能的竞价型实例的无限模式才适用。否则，支付剩余积分会使可突增性能的竞价型实例比使用其他实例更加昂贵。有关更多信息，请参阅 何时使用无限模式与固定 CPU。

以标准模式配置的 T2 实例可获得启动积分。T2 实例是唯一可获得启动积分的可突增性能实例。通过提供足够的计算资源来配置实例，启动积分旨在为 T2 实例提供有成效的初始启动体验。不允许重复启动 T2 实例以访问新的启动积分。如果您需要持续的 CPU，您可以赚取积分（通过空转一段时间），将无限模式用于 T2 Spot 实例，或将实例类型和专用 CPU 一起使用。

在单租户硬件上启动

您可以在单租户硬件上运行竞价型实例。专用竞价型实例与属于其他 Amazon 账户的实例物理隔离。有关更多信息，请参阅 Amazon EC2 专用实例和 Amazon EC2 专用实例。

要运行专用竞价型实例，请执行以下操作之一：

- 在创建竞价型实例请求时，请指定租期 `dedicated`。有关更多信息，请参阅管理您的竞价型实例。
- 在 VPC 中请求实例租期为 `dedicated` 的竞价型实例。有关更多信息，请参阅 在具有默认租赁的 VPC 中启动专用实例。如果您在 VPC 中请求实例租赁为 `dedicated` 的竞价型实例，则无法请求租赁为 `default` 的竞价型实例。

所有实例系列都支持除 T 实例外的专用竞价型实例。对于每个受支持的实例系列，只有最大的实例或设备支持专用竞价型实例。

查看竞价型实例定价历史记录

竞价型实例的价格由 Amazon EC2 设置，并根据竞价型实例容量的长期供求趋势逐步调整。

当您的 Spot 请求得到满足时，您的竞价型实例将以当前 Spot 价格启动，并且不超过按需价格。您可以查看最近 90 天的 Spot 价格历史记录，并按照实例类型、操作系统和可用区筛选。

要查看当前 Spot 实例价格，请参阅 Amazon EC2 Spot 实例定价。

Console

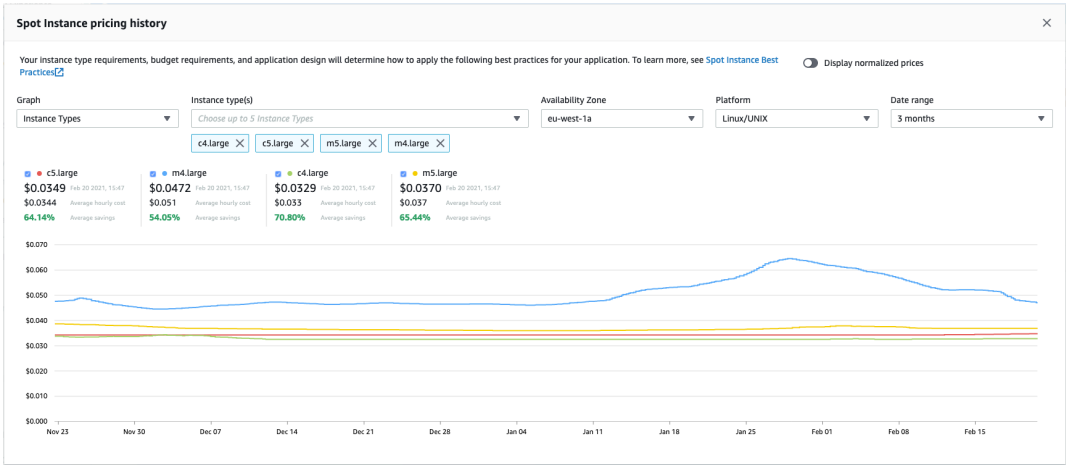
查看竞价型实例价格历史记录

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，请选择 Spot Requests。
3. 选择定价历史记录。
4. 对于 Graph (图表)，选择按 Availability Zones (可用区) 或 Instance Types (实例类型) 来比较价格历史记录。

   - 如果选择可用区，则选择实例类型、操作系统（平台）以及要查看价格历史记录的日期范围。
   - 如果选择实例类型，则最多选择五个实例类型、可用区、操作系统（平台）以及要查看价格历史记录的日期范围。

   以下屏幕截图显示了不同实例类型的价格比较。



5. 将指针移动到图形上并悬停，可显示选定日期范围内的特定时间的价格。价格显示在图表上方的信息块中。顶行中显示的价格显示了特定日期的价格。第二行中显示的价格显示了选定日期范围内的平均价格。
6. 要显示每个 vCPU 的价格，请开启 Display normalized prices (显示标准化价格)。要显示实例类型的价格，请关闭 Display normalized prices (显示标准化价格)。

Amazon CLI

查看竞价型实例价格历史记录

使用以下 describe-spot-price-history 命令。

```
aws ec2 describe-spot-price-history \
    --instance-types c6i.xlarge \
    --product-descriptions "Linux/UNIX" \
    --start-time 2025-04-01T00:00:00 \
```

```
    --end-time 2025-04-02T00:00:0
```

PowerShell

查看竞价型实例价格历史记录

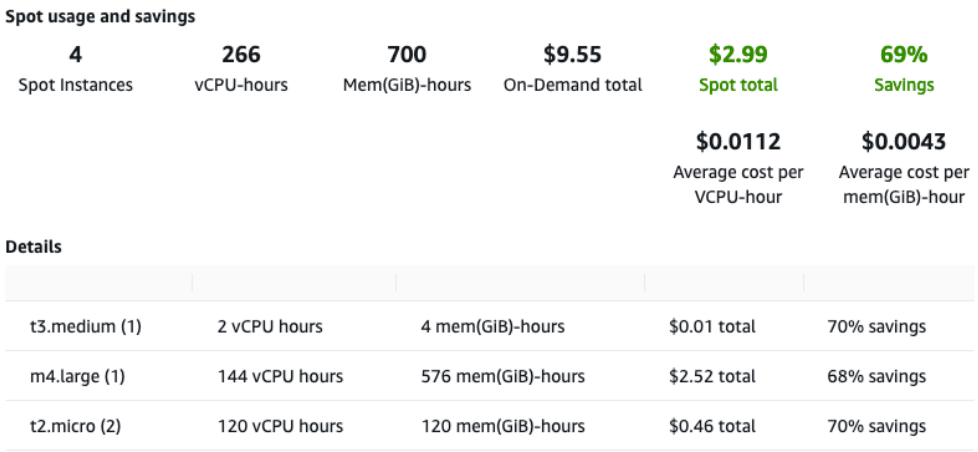
使用 Get-EC2SpotPriceHistory cmdlet。

```
Get-EC2SpotPriceHistory `
    -InstanceType c6i.xlarge `
    -ProductDescription "Linux/UNIX" `
    -UtcStartTime 2025-04-01T00:00:00 `
    -UtcEndTime 2025-04-02T00:00:0
```

通过购买竞价型实例实现节省

您可以在每个队列级别或针对所有正在运行的竞价型实例，查看竞价型实例的用量和节省信息。在每个队列级别，用量和节省信息包括该队列启动和终止的所有实例。您可以查看过去一小时或过去三天的此信息。

来自 Savings (节省成本) 部分的以下屏幕截图显示了 Spot 实例集的 Spot 用量和节省信息。



您可查看以下用量和节省信息：

- Spot Instances (Spot 实例) – Spot 实例集启动和终止的 Spot 实例数量。在查看节省摘要时，该数字表示您的所有正在运行的竞价型实例。
- vCPU-hours (vCPU 小时数) – 在所选时间范围内所有 Spot 实例 使用的 vCPU 小时数。
- Mem(GiB)-hours (内存 (GiB) 小时数) – 在所选时间范围内所有 Spot 实例 使用的 GiB 小时数。

- On-Demand total (按需总额) – 您在将这些实例作为 按需实例 启动后，在所选时间范围内支付的总额。
- Spot total (Spot 总额) – 您在所选时间范围内支付的总额。
- Savings (节省) – 您通过未支付按需价格而节省的百分比。
- Average cost per vCPU-hour (每 vCPU 小时的平均费用) – 在所选时间范围内所有 Spot 实例 使用 vCPU 的平均小时费用，其计算方式如下：每 vCPU 小时的平均费用 = Spot 总额 / vCPU 小时数。
- Average cost per mem(GiB)-hour (每内存 (GiB) 小时的平均费用) – 在所选时间范围内所有 Spot 实例 使用 GiB 的平均小时费用，其计算方式如下：每内存 (GiB) 小时的平均费用 = Spot 总额 / 内存 (GiB) 小时数。
- Details (详细信息) 表 – 构成 Spot 实例集的各种实例类型（每个实例类型的实例数括在圆括号中）。在查看节省摘要时，这些数字涵盖了您的所有正在运行的竞价型实例。

节省信息只能使用 Amazon EC2 控制台查看。

> **Note**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

查看竞价型实例集的节省信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择 Spot 队列请求的 ID，然后滚动到 Savings (节省成本) 部分。

   或者，选中竞价型实例集请求 ID 旁边的复选框，然后选择节省成本选项卡。
4. 默认情况下，该页面显示过去三天的用量和节省信息。您可以选择 last hour (过去一小时) 或 last three days (过去三天)。对于不到一小时之前启动的 Spot 队列，该页面显示这一小时的预计节省。

查看所有正在运行的竞价型实例的节省信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。

3. 选择 Savings Summary (节省成本摘要)。

创建竞价型实例请求

要使用竞价型实例，您需要创建竞价型实例请求，其中包括所需实例数量、实例类型以及可用区。当有容量可用时，Amazon EC2 将立即满足您的请求。否则，Amazon EC2 将等待直至可以完成您的请求，或者直至您取消请求。

您可以使用 Amazon EC2 控制台中的启动实例向导或 run-instances 命令以与启动按需型实例相同的方式请求竞价型实例。建议使用此方法的原因如下：

- 您已经在使用启动实例向导或 run-instances 命令启动按需型实例，而且只想通过更改单个参数来更改为启动竞价型实例。
- 您不需要多个实例类型不同的实例。

通常不建议使用此方法来启动竞价型实例，因为您无法指定多种实例类型，也无法在同一请求中启动竞价型实例和按需型实例。有关启动竞价型实例的首选方法（包括启动实例集，其中包含具有多种实例类型的竞价型实例和按需型实例类型），请参阅 使用哪种竞价型请求方法最好？

如果您一次请求了多个竞价型实例，Amazon EC2 将创建单独的竞价型实例，这样您可以分别跟踪各个请求的状态。有关跟踪竞价型实例请求的更多信息，请参阅 获取竞价型实例请求的状态。

Console

创建竞价型实例请求

步骤 1-9 与启动按需型实例时使用的步骤相同。在步骤 10 中，您配置竞价型实例请求。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，选择一个区域。
3. 从 Amazon EC2 控制台控制面板中，选择启动实例。
4. （可选）在Name and tags（名称和标签）下，您可以为实例命名，并为竞价型实例请求、实例、卷和弹性图形添加标签。有关标签的信息，请参阅 标记 Amazon EC2 资源。

   a. 对于 Name（名称），为您的实例输入一个描述性名称。

      实例名称是一个标签，其中密钥为名称，而值为您指定的名称。如果您没有指定名称，则可以通过其 ID 标识实例，该 ID 将在您启动实例时自动生成。

b. 要为竞价型实例请求、实例、卷和弹性图形添加标签，请选择 Add additional tags（添加其他标签）。选择添加标签，然后输入密钥和值，然后选择要标记的资源类型。为每个要添加的其它标签选择添加标签。

5. 在应用程序和操作系统镜像（亚马逊机器映像）下，为您的实例选择操作系统（OS），然后选择 AMI。有关更多信息，请参阅 应用程序和操作系统映像（亚马逊机器映像）。

6. 在 Instance type（实例类型）下，请选择符合您对实例硬件配置和大小要求的实例类型。有关更多信息，请参阅 实例类型。

7. 在 Key pair（login）（密钥对（登录））下，选择一个现有密钥对，或选择 Create new key pair（创建新密钥对）来新建一个密钥对。有关更多信息，请参阅 Amazon EC2 密钥对和 Amazon EC2 实例。

> **Important**
>
> 如果您选择 Proceed without key pair (Not recommended)（在没有密钥对的情况下继续（不推荐））选项，则将无法连接到此实例，除非您选择配置为允许用户以其它方式登录的 AMI。

8. 在 Network settings（网络设置）下，使用默认设置，或选择 Edit（编辑）根据需要配置网络设置。

   安全组构成了网络设置的一部分，并为实例定义防火墙规则。这些规则指定哪些传入的网络流量可传输到您的实例。

   有关更多信息，请参阅 网络设置。

9. 您选择的 AMI 包含一个或多个存储卷，包括根设备卷。在 Configure storage（配置存储）下，您可以选择 Add new volume（添加新卷）来指定要附加到实例的其他卷。有关更多信息，请参阅 配置存储。

10. 在 Advanced details（高级详细信息）下，按如下方式配置竞价型实例请求：

    a. 在购买选项下，选择请求竞价型实例复选框。

    b. 您可以保留竞价型实例请求的默认配置，也可以选择（位于右侧的）Customize（自定义）为竞价型实例请求指定自定义设置。

       选择 Customize（自定义）后，将显示以下字段。

       i. Maximum price（最高价）：您可以以 Spot 价格请求竞价型实例，上限为按需价格，也可以指定您愿意支付的最高金额。

> **Warning**
>
> 如果您指定了某个最高价，则实例被中断的频率将比您选择 No maximum price（无最高价）时更高。
> 如果您指定最高价格，则该价格必须高于 0.001 美元。指定低于 0.001 美元的值将导致启动失败。

- No maximum price（无最高价）：您的竞价型实例将以当前的 Spot 价格启动。价格永远不会超过按需价格。（建议）
- Set your maximum price (per instance/hour)（设置最高价（每实例/小时））：您可以指定愿意支付的最高金额。
  - 如果您指定的最高价低于 Spot 价格，则不会启动您的竞价型实例。
  - 如果您指定的最高价高于当前 Spot 价格，则您的竞价型实例将按当前 Spot 价格启动和计费。竞价型实例运行后，如果 Spot 价格上涨到超过您的最高价格，Amazon EC2 将中断您的竞价型实例。
  - 无论您指定哪种最高价，都将始终按当前的 Spot 价格向您收取费用。

  要查看 Spot 价格趋势，请参阅 查看竞价型实例定价历史记录。

ii. Request type（请求类型）：您选择的竞价型实例请求类型决定了竞价型实例中断时会发生什么。

- One-time（一次性）：Amazon EC2 为您的竞价型实例发出一次性请求。如果您的竞价型实例被中断，不会重新提交请求。
- Persistent request（持久性请求）：Amazon EC2 对您的竞价型实例发出持久请求。如果您的竞价型实例中断，则将重新提交请求以补充中断的竞价型实例。

如果您未指定值，则默认值为一次性请求。

iii. Valid to（有效截止时间）：持久性竞价型实例请求的到期日期。

一次性请求不支持此字段。一次性请求在请求中的所有实例启动或者您取消请求前保持有效。

- No request expiry date（没有请求到期日期）：请求在您将其取消之前保持有效。

- Set your request expiry date（设置请求到期日期）：持久性请求将保持活动状态，直到您指定的日期或取消该请求为止。

iv. Interruption behavior（中断行为）：您选择的行为决定了竞价型实例中断时会发生什么。

- 对于持久性请求，有效值为 Stop（停止）和 Hibernate（休眠）。实例停止后，将收取 EBS 卷存储费用。

> **Note**
>
> 竞价型实例现在使用与按需型实例相同的休眠功能。要启用休眠，您可以在此处选择休眠，也可以在启动实例向导下方显示的停止 - 休眠行为字段中选择启用。有关休眠的先决条件，请参阅 EC2 实例休眠的先决条件。

- 对于一次性请求，只有 Terminate（终止）有效。

如果您未指定值，则默认值为 Terminate（终止），这对持久性竞价型实例请求无效。如果您保持默认设置并尝试启动持久性竞价型实例请求，则会收到错误消息。

有关更多信息，请参阅 竞价型实例中断行为。

11. 在存储库的 Summary（摘要）面板，对于 Number of instances（实例数量），输入要启动的实例数量。

> **Note**
>
> Amazon EC2 为每个竞价型实例创建一个单独的请求。

12. 在 Summary（摘要）面板，查看实例的详细信息并进行必要的更改。在提交竞价型实例请求后，您无法更改该请求的参数。您可以在 Summary（摘要）面板选择启动实例向导中某部分的链接以直接导航到该部分。有关更多信息，请参阅 摘要。
13. 当您准备好启动您的实例时，请选择 Launch instance（启动实例）。

如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

Amazon CLI

使用 run-instances 创建竞价型实例请求

按如下所示，使用 run-instances 命令并在 `--instance-market-options` 参数中指定竞价型实例选项。

```
--instance-market-options file://spot-options.json
```

以下是要在 JSON 文件中指定的数据结构。您还可以指定 `ValidUntil` 和 `InstanceInterruptionBehavior`。如果未在数据结构中指定字段，则将使用默认值。

以下示例将创建一个 `persistent` 请求。

```
{
  "MarketType": "spot",
  "SpotOptions": {
    "SpotInstanceType": "persistent"
  }
}
```

使用 request-spot-instances 创建竞价型实例请求

> **Note**
>
> 我们强烈反对使用 request-spot-instances 命令来请求竞价型实例，因为它是不具有计划投资的旧式 API。有关更多信息，请参阅 使用哪种竞价型请求方法最好？

使用 request-spot-instances 命令可创建一次性请求。

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json
```

使用 request-spot-instances 命令可创建持久性请求。

```
aws ec2 request-spot-instances \
    --instance-count 5 \
```

```
    --type "persistent" \
    --launch-specification file://specification.json
```

有关要用于这些命令的启动规范文件的示例，请参阅竞价型实例请求示例启动规范。如果您从竞价型请求控制台下载启动规范文件，必须改为使用 request-spot-fleet 命令（竞价型请求控制台使用竞价型实例集指定竞价型实例请求）。

PowerShell

创建竞价型实例请求

使用 New-EC2Instance cmdlet 并使用 `-InstanceMarketOption` 参数指定竞价型实例选项。

```
-InstanceMarketOptions $marketOptions
```

按如下所示创建竞价型实例选项的数据结构。

```
$spotOptions = New-Object Amazon.EC2.Model.SpotMarketOptions
$spotOptions.SpotInstanceType="persistent"
$marketOptions = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketOptions.MarketType = "spot"
$marketOptions.SpotOptions = $spotOptions
```

竞价型实例请求示例启动规范

以下示例显示了可与 request-spot-instances 命令结合使用来创建 Spot 实例请求的启动配置。有关更多信息，请参阅 管理您的竞价型实例。

> **Important**
>
> 我们强烈反对使用 request-spot-instances 命令来请求竞价型实例，因为它是不具有计划投资的旧式 API。有关更多信息，请参阅 使用哪种竞价型请求方法最好？

示例

- 示例 1：启动竞价型实例
- 示例 2：在指定的可用区中启动竞价型实例
- 示例 3：在指定的子网中启动竞价型实例

- 示例 4：启动专用竞价型实例

示例 1：启动竞价型实例

以下示例不包含可用区或子网。Amazon EC2 会为您选择可用区。Amazon EC2 会在所选可用区的默认子网中启动实例。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

示例 2：在指定的可用区中启动竞价型实例

以下示例包括一个可用区。Amazon EC2 会在指定可用区的默认子网中启动实例。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "m5.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

示例 3：在指定的子网中启动竞价型实例

以下示例包括子网。Amazon EC2 会在指定子网中启动实例。如果 VPC 是一个非默认 VPC，则默认情况下，该实例不会收到公有 IPv4 地址。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
```

```
  "InstanceType": "m5.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

要将公有 IPv4 地址分配给非默认 VPC 中的实例，请指定 AssociatePublicIpAddress 字段，如以下示例所示。指定网络接口时，您必须包含使用网络接口 (而不是使用之前的代码块中所示的 SubnetId 和 SecurityGroupIds 字段) 的子网 ID 和安全组 ID。

```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "InstanceType": "m5.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
      "Groups": [ "sg-1a2b3c4d5e6f7g8h9" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

示例 4：启动专用竞价型实例

以下示例请求租期为 dedicated 的竞价型实例。专用竞价型实例必须在 VPC 中启动。

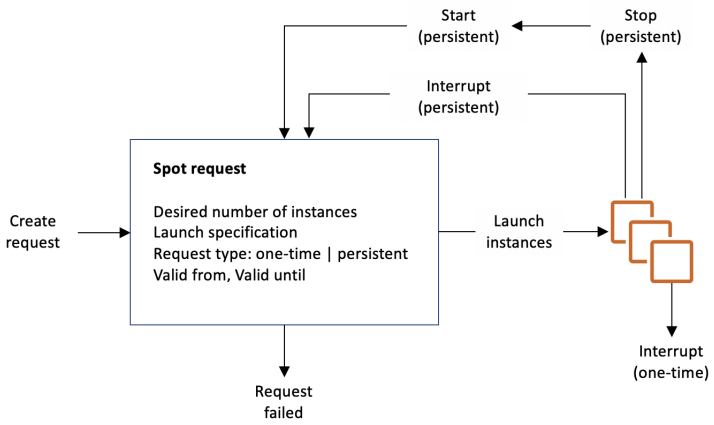
```
{
  "ImageId": "ami-0abcdef1234567890",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d5e6f7g8h9" ],
  "InstanceType": "c5.8xlarge",
  "SubnetId": "subnet-1a2b3c4d5e6f7g8h9",
  "Placement": {
    "Tenancy": "dedicated"
  }
}
```

获取竞价型实例请求的状态

要帮助跟踪您的竞价型实例请求和规划您对竞价型实例的使用，请使用 Amazon EC2 提供的请求状态。例如，请求状态可以提供尚未完成 Spot 请求的原因，或者列出妨碍完成 Spot 请求的限制。

在此过程（也称为 Spot 请求生命周期）中的每一步，都有特定事件确定连续的请求状态。

以下演示了竞价型实例请求的运行方式。请注意，请求类型（一次性或持久性）确定在 Amazon EC2 中断竞价型实例时或者在您停止竞价型实例时是否重新打开请求。如果请求是持久性请求，则在中断竞价型实例之后将重新打开请求。如果请求是持久性的，并且您停止了竞价型实例，则请求仅在您启动竞价型实例后打开。



内容

- 获取请求状态信息
- Spot 请求状态代码
- EC2 竞价型实例请求履行事件
- Spot 请求的状态更改

获取请求状态信息

您可以获取竞价型实例的状态信息。

Console

获取请求状态信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests (Spot 请求)，然后选择所需的 Spot 请求。

3. 要检查状态，请在 Description (描述) 选项卡上检查 Status (状态) 字段。

Amazon CLI

获取请求状态信息

使用以下 describe-spot-instance-requests 命令。

```
aws ec2 describe-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

获取请求状态信息

使用 Get-EC2SpotInstanceRequest cmdlet。

```
Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

Spot 请求状态代码

Spot 请求状态信息包含状态代码、更新时间以及状态消息。这些信息帮助您确定如何处理 Spot 请求。

下面是 Spot 请求状态代码：

`az-group-constraint`

Amazon EC2 无法在同一可用区中启动您请求的所有实例。

`bad-parameters`

您的 Spot 请求的一个或多个参数无效 (例如，您指定的 AMI 不存在)。状态消息指示哪个参数无效。

`canceled-before-fulfillment`

在完成前，用户取消了竞价请求。

`capacity-not-available`

您请求的实例没有足够的容量可用。

`constraint-not-fulfillable`

由于一个或多个限制无效 (例如，可用区不存在)，Spot 请求无法完成。状态消息指示哪个限制无效。

`fulfilled`

Spot 请求处于 `active` 状态，Amazon EC2 正在启动您的 Spot 实例。

`instance-stopped-by-price`

您的实例已停止，因为 Spot 价格超出了您的最高价格。

`instance-stopped-by-user`

由于用户停止了实例或从实例运行了 shutdown 命令，您的实例已停止。

`instance-stopped-no-capacity`

由于 EC2 容量管理需要，您的实例已停止。

`instance-terminated-by-price`

您的实例已终止，因为 Spot 价格超出了您的最高价格。如果您的请求是持久性的，该过程将重新开始，因此，您的请求将等待进行评估。

`instance-terminated-by-schedule`

您的竞价型实例在其计划持续时间结束时终止。

`instance-terminated-by-service`

您的实例从停止状态终止。

`instance-terminated-by-user` 或者 `spot-instance-terminated-by-user**`

您终止了已完成的竞价型实例，因此，请求状态为（`closed`除非这是持久性请求），实例状态为 `terminated`。

`instance-terminated-launch-group-constraint`

您的启动组中的一个或多个实例已终止，因此不再满足启动组的限制。

`instance-terminated-no-capacity`

由于标准容量管理流程，您的实例已终止。

`launch-group-constraint`

Amazon EC2 无法同时启动您请求的所有实例。启动组内的所有实例都一起启动和终止。

`limit-exceeded`

超过了 EBS 卷数量或总卷存储的限制。有关更多信息，请参阅《Amazon EBS User Guide》中的 Quotas for Amazon EBS。

`marked-for-stop`

竞价型实例被标记为停止。

`marked-for-termination`

竞价型实例被标记为终止。

`not-scheduled-yet`

您的 Spot 请求在指定日期之前不会被评估。

`pending-evaluation`

当您提交竞价型实例请求之后，该请求会进入 `pending-evaluation` 状态，同时系统会评估您的请求中的参数。

`pending-fulfillment`

Amazon EC2 正尝试预置竞价型实例。

`placement-group-constraint`

因为竞价型实例目前不能添加到置放群组中，因此尚无法完成 Spot 请求。

`price-too-low`

由于您的最高价低于 Spot 价格，无法完成请求。在这种情况下，不会启动任何实例，并且您的请求保持 `open` 状态。

`request-canceled-and-instance-running`

在竞价型实例仍在运行时，您取消了 Spot 请求。请求为 `cancelled`，但是，实例保持为 `running`。

`schedule-expired`

由于没有在指定日期前完成，Spot 请求已过期。

`system-error`

出现意外系统错误。如果这是反复出现的问题，请联系 Amazon Web Services 支持 以获取帮助。

EC2 竞价型实例请求履行事件

当竞价型实例请求履行后，Amazon EC2 会向 Amazon EventBridge 发送一个 EC2 竞价型实例请求履行事件。您可以创建一条规则，以在此事件发生时执行操作，例如调用某个 Lambda 函数或通知某个 Amazon SNS 主题。

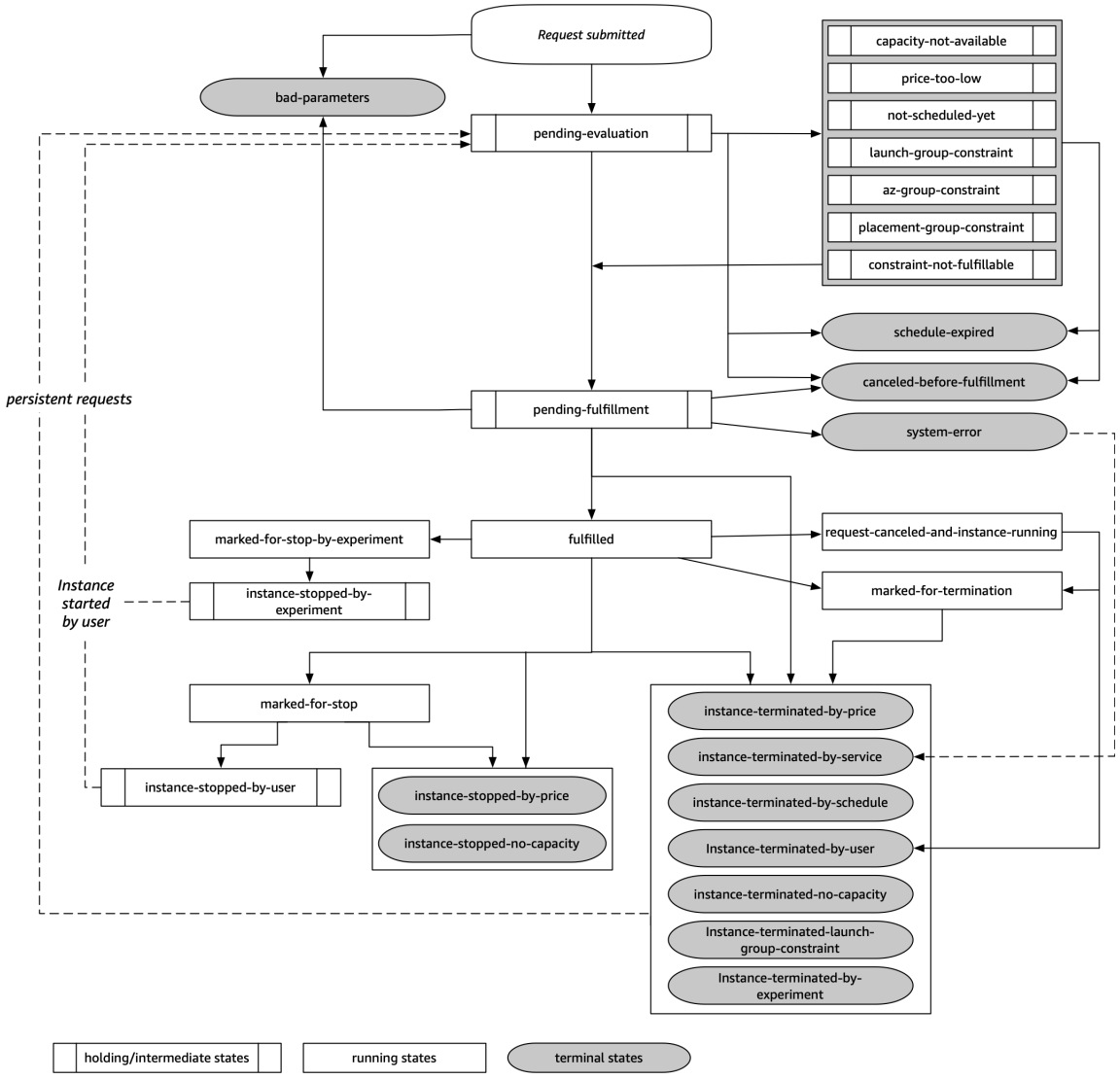
以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "01234567-1234-0123-1234-012345678901",
    "detail-type": "EC2 Spot Instance Request Fulfillment",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "spot-instance-request-id": "sir-0e54a519c9EXAMPLE",
        "instance-id": "i-1234567890abcdef0"
    }
}
```

有关更多信息，请参阅 Amazon EventBridge 用户指南。

Spot 请求的状态更改

以下图表显示您的 Spot 请求在其整个生命周期 (从提交到终止) 所遵循的路径。每个步骤用节点表示，每个节点状态代码描述您的 Spot 请求和竞价型实例的状态。



待评估

当您创建竞价型实例请求之后，除非一个或多个请求参数无效（bad-parameters），否则该请求就会进入 pending-evaluation 状态。

状态代码	请求状态	实例状态
pending-evaluation	open	不适用
bad-parameters	closed	不适用

暂停

如果一个或多个请求限制有效但目前无法满足，或者如果没有足够的容量，那么请求将进入暂挂状态，等待满足限制。请求选项影响请求完成的可能性。例如，如果没有容量可用，您的请求将会保持暂挂状态，直到有容量可用为止。如果您指定了可用区组，则该请求将保持为暂挂状态，直至满足可用区的限制。

如果其中一个可用区中断，则可能会影响其他可用区中可用于竞价型实例请求的备用 EC2 容量。

状态代码	请求状态	实例状态
capacity-not-available	open	不适用
price-too-low	open	不适用
not-scheduled-yet	open	不适用
launch-group-constraint	open	不适用
az-group-constraint	open	不适用
placement-group-constraint	open	不适用
constraint-not-fulfillable	open	不适用

等待评估/最终执行

如果您创建的请求仅在特定时段内有效，但该时段在您的请求到达等待执行阶段之前过期，则您的竞价型实例请求可能会进入 terminal 状态。如果您取消请求，或者出现系统错误，请求也可能会进入该状态。

状态代码	请求状态	实例状态
schedule-expired	cancelled	

状态代码	请求状态	实例状态
		不适用
canceled-before-fulfillment [1]	cancelled	不适用
bad-parameters	failed	不适用
system-error	closed	不适用

[1] 如果您取消请求。

等待履行

如果满足指定的限制（如有），您的 Spot 请求将会进入 pending-fulfillment 状态。

此时，Amazon EC2 已经准备好为您预置您请求的实例。如果该过程在此时停止，则可能是因为用户在启动竞价型实例之前取消了请求。也可能是因为出现了意外的系统错误。

状态代码	请求状态	实例状态
pending-fulfillment	open	不适用

已完成

当您的竞价型实例的所有规范都得到满足时，您的竞价型实例就会执行。Amazon EC2 会启动竞价型实例，这可能需要几分钟的时间。如果竞价型实例在中断时休眠或停止，它将保持该状态，直到可以再次完成该请求或取消该请求。

状态代码	请求状态	实例状态
fulfilled	active	pending → running
fulfilled	active	stopped → running

如果您停止竞价型实例，则 Spot 请求将进入 marked-for-stop 或 instance-stopped-by-user 状态，直到竞价型实例可以重新启动或者取消了请求。

状态代码	请求状态	实例状态
marked-for-stop	active	stopping
instance-stopped-by-user [1]	disabled 或 cancelled [2]	stopped

[1] 如果您停止实例或从实例运行 shutdown 命令，则竞价型实例将进入 instance-stopped-by-user 状态。停止实例后，您可以重新启动它。重新启动时，竞价型实例请求将返回 pending-evaluation 状态，然后 Amazon EC2 在满足约束时启动一个新的竞价型实例。

[2] 如果您停止竞价型实例但不取消请求，则 Spot 请求状态为 disabled。如果您的竞价型实例已停止并且请求已过期，则请求状态为 cancelled。

执行的最终

只要实例类型具有可用的容量，并且您未终止竞价型实例，这些实例就会继续运行。如果 Amazon EC2 必须终止竞价型实例，竞价型请求将会进入终止状态。如果取消 Spot 请求或终止竞价型实例，请求也将进入终止状态。

状态代码	请求状态	实例状态
request-canceled-and-instance-running	cancelled	running
marked-for-stop	active	running
marked-for-termination	active	running
instance-stopped-by-price	disabled	stopped
instance-stopped-by-user	disabled	stopped

状态代码	请求状态	实例状态
instance-stopped-no-capacity	disabled	stopped
instance-terminated-by-price	closed（一次性），open（持久性）	terminated
instance-terminated-by-schedule	closed	terminated
instance-terminated-by-service	cancelled	terminated
instance-terminated-by-user	closed 或 cancelled [1]	terminated
instance-terminated-no-capacity	closed（一次性），open（持久性）	running †
instance-terminated-no-capacity	closed（一次性），open（持久性）	terminated
instance-terminated-launch-group-constraint	closed（一次性），open（持久性）	terminated

[1] 如果您终止实例但未取消请求，则请求状态为 closed。如果您终止实例并取消请求，则请求状态为 cancelled。即使您在取消实例请求之前终止了竞价型实例，Amazon EC2 检测您的竞价型实例已终止的过程可能会有延迟。在这种情况下，请求状态可能是 closed 或 cancelled。

† Amazon EC2 中断竞价型实例时（如果其需要恢复容量且该实例配置为在中断时终止），状态将立即设置为 instance-terminated-no-capacity（如果未设置为 marked-for-termination）。但是，实例会保持 2 分钟 running 状态，以反映实例收到竞价型实例中断通知的 2 分钟时间段。2 分钟后，实例状态设置为 terminated。

中断实验

您可以使用 Amazon Fault Injection Service 启动竞价型实例中断，从而测试竞价型实例上的应用程序如何响应中断。如果 Amazon FIS 停止竞价型实例，则竞价型实例请求会先进入 `marked-for-stop-by-experiment` 状态，然后进入 `instance-stopped-by-experiment` 状态。如果 Amazon FIS 终止竞价型实例，则竞价型实例请求会进入 `instance-terminated-by-experiment` 状态。有关更多信息，请参阅 the section called “启动中断”。

状态代码	请求状态	实例状态
`marked-for-stop-by-experiment`	`active`	`running`
`instance-stopped-by-experiment`	`disabled`	`stopped`
`instance-terminated-by-experiment`	`closed`	`terminated`

持久性请求：

当您的竞价型实例终止（由您或由 Amazon EC2）时，如果 Spot 请求为持久性请求，则该请求返回 `pending-evaluation` 状态，并且在满足约束时，Amazon EC2 可以启动新的竞价型实例。

标记竞价型实例请求

要对您的竞价型实例请求进行分类和管理，您可使用自定义元数据标记它们。您可以在创建竞价型实例请求时或之后为其分配标签。您可以使用 Amazon EC2 控制台或命令行工具分配标签。

在标记竞价型实例请求时，不会自动标记由竞价型实例请求启动的实例和卷。您需要明确标记由竞价型实例请求启动的实例和卷。您可以在启动期间或之后为竞价型实例和卷分配标签。

有关标签的工作原理的更多信息，请参阅标记 Amazon EC2 资源。

内容

- 前提条件
- 标记新的竞价型实例请求
- 标记现有竞价型实例请求
- 查看竞价型实例请求标记

前提条件

授予用户标记资源的权限。有关 IAM policies 和示例策略的更多信息，请参阅示例：标记资源。

您创建的 IAM policy 由您用于创建竞价型实例请求的方法决定。

- 如果您使用启动实例向导或 `run-instances` 请求 Spot 实例，请参阅To grant a user the permission to tag resources when using the launch instance wizard or run-instances。
- 如果您使用 `request-spot-instances` 命令请求竞价型实例，请参阅 To grant a user the permission to tag resources when using request-spot-instances。

在使用启动实例向导或 run-instances 时向用户授予资源标记权限

创建包含以下内容的 IAM policy：

- `ec2:RunInstances` 操作。这将授予用户授予启动实例的权限。
- 对于 `Resource`，请指定 `spot-instances-request`。这允许用户创建竞价型实例请求，以请求竞价型实例。
- `ec2:CreateTags` 操作。这将授予用户创建标签的权限。
- 对于 `Resource`，请指定 `*`。这将允许用户标记在实例启动期间创建的所有资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLaunchInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
```

```
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagSpotInstanceRequests",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

使用 RunInstances 操作创建竞价型实例请求，并在创建时标记竞价型实例请求时，您需要了解 Amazon EC2 如何评估 RunInstances 语句（在 IAM 策略中评估）中的 `spot-instances-request` 资源，如下所示：

- 如果您在创建时未标记竞价型实例请求，则 Amazon EC2 不会在 RunInstances 语句中评估 `spot-instances-request` 资源。
- 如果您在创建时标记竞价型实例请求，则 Amazon EC2 会在 RunInstances 语句中评估 `spot-instances-request` 资源。

因此，对于 `spot-instances-request` 资源，以下规则适用于 IAM 策略：

- 如果您使用 RunInstances 创建竞价型实例请求，并且您不打算在创建时标记竞价型实例请求，则无需明确允许 `spot-instances-request` 资源；调用将成功。
- 如果您使用 RunInstances 创建竞价型实例请求并打算在创建时标记竞价型实例请求，则必须在 RunInstances 允许语句中包含 `spot-instances-request` 资源，否则调用将失败。
- 如果您使用 RunInstances 创建竞价型实例请求并打算在创建时标记竞价型实例请求，则必须在 CreateTags 允许语句中指定 `spot-instances-request` 资源或包括 * 通配符，否则调用将失败。

有关示例 IAM policy（包括竞价型实例请求不支持的策略），请参阅 使用竞价型实例。

在使用 request-spot-instances 时向用户授予资源标记权限

创建包含以下内容的 IAM policy：

- ec2:RequestSpotInstances 操作。这将授予用户创建竞价型实例请求的权限。
- ec2:CreateTags 操作。这将授予用户创建标签的权限。
- 对于 Resource，请指定 spot-instances-request。这将允许用户仅标记竞价型实例请求。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotInstanceRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:RequestSpotInstances",
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-instances-
request/*"
        }
    ]
}
```

标记新的竞价型实例请求

在 Amazon CLI 和 PowerShell 示例中，按如下方式配置竞价型实例请求：

- 对于 ResourceType，请指定 spot-instances-request。如果指定其他值，则竞价型实例请求将失败。
- 对于 Tags，请指定键值对。您可以指定多个键值对。

Console

标记新竞价型实例请求

1. 按照管理您的竞价型实例过程操作。
2. 要添加标签，请在添加标签页上，选择添加标签，然后输入标签的键和值。为每个附加标签选择添加其他标签。

对于每个标签，您可以使用相同标签来标记竞价型实例请求、竞价型实例和卷。要标记所有这三个项，请确保已选定 Instances (实例)、Volumes (卷)和 Spot Instance Requests (Spot 实例请求)。要仅标记其中的一个或两个项，请确保已选定要标记的资源，并清除其他资源。

3. 填写必填字段以创建 Spot 实例请求，然后选择 Launch (启动)。有关更多信息，请参阅 管理您的竞价型实例。

Amazon CLI

标记新竞价型实例请求

使用 request-spot-instances 命令和 `--tag-specification` 选项。

标签规范向竞价型实例请求添加两个标签：`Environment=Production` 和 `Cost-Center=123`。

```
aws ec2 request-spot-instances \
    --instance-count 5 \
    --type "one-time" \
    --launch-specification file://specification.json \
    --tag-specification 'ResourceType=spot-instances-
request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]'
```

PowerShell

标记新竞价型实例请求

使用 Request-EC2SpotInstance cmdlet 和 `-TagSpecification` 参数。

```
-TagSpecification $tagspec
```

标签规范定义如下。它向竞价型实例请求添加两个标签：`Environment=Production` 和 `Cost-Center=123`。

```
$tag1 = @{Key="Environment"; Value="Production"}
$tag2 = @{Key="Cost-Center"; Value="123"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "spot-instances-request"
$tagspec.Tags = @($tag1,$tag2)
```

标记现有竞价型实例请求

Console

标记现有竞价型实例请求

创建竞价型实例请求后，您可以使用控制台向竞价型实例请求添加标签。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择您的竞价型实例请求。
4. 选择 Tags (标签) 选项卡，然后选择 Create Tag (创建标签)。

使用控制台标记现有的竞价型实例

在竞价型实例请求启动竞价型实例后，您可以使用控制台向实例添加标签。有关更多信息，请参阅使用控制台添加标签。

Amazon CLI

标记现有的竞价型实例请求或竞价型实例

使用 create-tags 命令标记现有的资源。在以下示例中，使用 `purpose=test` 标记现有竞价型实例请求和竞价型实例。

```
aws ec2 create-tags \
    --resources sir-0e54a519c9EXAMPLE i-1234567890abcdef0 \
    --tags Key=purpose,Value=test
```

PowerShell

标记现有的竞价型实例请求或竞价型实例

使用 New-EC2Tag cmdlet。以下示例将标签 `purpose=test` 添加到现有竞价型实例请求和竞价型实例。

```
New-EC2Tag `
    -Resource sir-0e54a519c9EXAMPLE, i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="test"}
```

查看竞价型实例请求标记

Console

查看竞价型实例请求标签

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择您的 Spot 实例请求并选择 Tags (标签) 选项卡。

Amazon CLI

描述竞价型实例请求标记

使用 describe-spot-instance-requests 命令可查看指定的 Spot 实例请求的配置，其中包括为请求指定的任何标签。

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].Tags"
```

下面是示例输出。

```
[
    [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Department",
            "Value": "101"
        }
    ]
]
```

PowerShell

描述竞价型实例请求标记

使用 Get-EC2SpotInstanceRequest cmdlet。

```
(Get-EC2SpotInstanceRequest `
    -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).Tags
```

下面是示例输出。

```
Key         Value
---         -----
Environment Production
Department  101
```

取消竞价型实例请求

如果您不再需要竞价型实例请求，您可以将其取消。您只能取消 open、active 或 disabled 的竞价型实例请求。

- 当您的请求未执行，且实例没有启动时，您的竞价型实例请求处于 open 状态。
- 当您的请求完成且竞价型实例因此已启动时，您的竞价型实例请求处于 active 状态。
- 当您停止竞价型实例时，您的竞价型实例请求处于 disabled 状态。

如果您的竞价型实例请求处于 active 状态，且关联的竞价型实例正在运行，那么取消请求不会终止该实例。有关终止竞价型实例的更多信息，请参阅 终止竞价型实例。

Console

取消竞价型实例请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例请求。
4. 依次选择操作和取消请求。
5. (可选) 如果不再使用关联的竞价型实例，可以将其终止。在取消竞价请求对话框中，选择终止实例，然后选择确认。

Amazon CLI

取消竞价型实例请求

使用以下 cancel-spot-instance-requests 命令。

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-
ids sir-0e54a519c9EXAMPLE
```

PowerShell

取消竞价型实例请求

使用 Stop-EC2SpotInstanceRequest cmdlet。

```
Stop-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE
```

管理您的竞价型实例

当有容量可用时，Amazon EC2 会启动竞价型实例。竞价型实例将一直运行，直到该实例中断，或者您自行终止该实例。

内容

- 查找竞价型实例
- 查找由特定请求启动的实例
- 停止竞价型实例
- 启动竞价型实例
- 终止竞价型实例

查找竞价型实例

竞价型实例与按需型实例一起显示在控制台的实例页面中。使用以下过程查找竞价型实例。

Console

查找竞价型实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 要查找所有竞价型实例，请在搜索窗格中选择实例生命周期 = 竞价型。

4. 要验证实例是否为竞价型实例，请选择该实例，选择详细信息选项卡，然后查看生命周期的值。竞价型实例的值为 spot，按需型实例的值为 normal。

Amazon CLI

查找竞价型实例

使用以下 describe-instances 命令。

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

确定实例是否为竞价型实例

使用以下 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[*].InstanceLifecycle" \
    --output text
```

如果值为 spot，则表示实例是竞价型实例。如果没有输出，则表示实例是按需型实例。

PowerShell

查找竞价型实例

使用 Get-EC2Instance cmdlet。

```
Get-EC2Instance -Filter @{Name="instance-lifecycle"; Values="spot"}
```

确定实例是否为竞价型实例

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.InstanceLifecycle
```

如果值为 Spot，则表示实例是竞价型实例。如果没有输出，则表示实例是按需型实例。

查找由特定请求启动的实例

使用以下过程查找通过特定竞价型实例或竞价型实例集请求启动的竞价型实例。

Console

查找请求的竞价型实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。列表包含了竞价型实例请求和竞价型实例集请求。
3. 如果竞价型实例请求已执行，那么容量就是竞价型实例的 ID。对于 Spot 实例集，Capacity (容量) 表示已执行的请求容量。要查看 Spot 实例集中的实例的 ID，请选择扩展箭头，或者选择队列，然后选择 Instances (实例)。
4. 对于竞价型实例集，容量表示已执行的请求容量。要查看竞价型实例集中的实例 ID，请选择实例集 ID 打开其详细信息页面，然后找到实例窗格。

Amazon CLI

查找请求的竞价型实例

使用以下 describe-spot-instance-requests 命令。

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-0e54a519c9EXAMPLE \
    --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

下面是示例输出：

```
[
    {
        "ID": "i-1234567890abcdef0"
    },
    {
        "ID": "i-0598c7d356eba48d7"
    }
]
```

PowerShell

查找请求的竞价型实例

使用 Get-EC2SpotInstanceRequest cmdlet。

```
(Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-0e54a519c9EXAMPLE).InstanceId
```

停止竞价型实例

如果现在不需要您的竞价型实例，但希望能够在以后将其重启且不丢失在 Amazon EBS 卷中持久保留的数据，则可以将其停止。停止竞价型实例的步骤与停止按需型实例的步骤类似。

> **Note**
>
> 在停止竞价型实例后，您可以修改其部分实例属性，但不能修改实例类型。
> 我们不会对已停止的竞价型实例收费，也不会收取数据传输费，但我们会对所有 Amazon EBS 卷的存储收费。

限制

- 只有当竞价型实例是从 `persistent` 竞价型实例请求启动时，您才能停止竞价型实例。
- 如果关联的竞价型实例请求被取消，则无法停止竞价型实例。当竞价型实例请求被取消后，您只能终止竞价型实例。
- 如果竞价型实例是某个队列或启动组或可用区组的一部分，则无法停止它。

Console

停止竞价型实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择竞价型实例。如果您没有保存竞价型实例的实例 ID，请参阅the section called “查找竞价型实例”。
4. 依次选择实例状态、停止实例。
5. 当系统提示您确认时，选择 Stop。

Amazon CLI

停止竞价型实例

使用 stop-instances 命令手动停止竞价型实例。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

停止竞价型实例

使用 Stop-EC2Instance cmdlet。

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

启动竞价型实例

您可以启动以前停止的竞价型实例。

先决条件

您只能在以下情况下启动竞价型实例：

- 您手动停止了竞价型实例。
- 竞价型实例是 EBS 支持的实例。
- 竞价型实例容量可用。
- Spot 价格低于您的最高价格。

限制

- 如果竞价型实例是某个队列或启动组或可用区组的一部分，则无法启动它。

启动竞价型实例的步骤与启动按需型实例的步骤类似。

Console

启动竞价型实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择竞价型实例。如果您没有保存竞价型实例的实例 ID，请参阅the section called “查找竞价型实例”。
4. 依次选择实例状态、启动实例。

Amazon CLI

启动竞价型实例

使用 start-instances 命令手动启动竞价型实例|。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

启动竞价型实例

使用 Start-EC2Instance cmdlet。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

终止竞价型实例

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

如果您终止的运行中或已停止竞价型实例是由持久性竞价型实例请求启动的，则竞价型实例请求会转换为 open 状态，这样就可以启动新的竞价型实例。要确保没有启动新的竞价型实例，您必须首先取消该竞价型实例请求。

如果您取消含有正在运行的竞价型实例的 active 竞价型实例请求，则正在运行的竞价型实例不会自动终止；您必须手动终止该竞价型实例。

如果您取消含有已停止的竞价型实例的 disabled 竞价型实例请求，则 Amazon EC2 Spot 服务将自动终止已停止的竞价型实例。取消竞价型实例请求与 Spot 服务终止竞价型实例之间可能存在短暂的滞后。

有关更多信息，请参阅 取消竞价型实例请求。

Console

手动终止竞价型实例

1. 在终止实例前，请确认您是否会丢失任何数据，方法是确认您的 Amazon EBS 卷是否会在终止时被删除，以及您是否已将所需数据从实例存储卷复制到持久性存储，例如 Amazon EBS 或 Amazon S3。
2. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
3. 在导航窗格中，选择 Instances (实例)。
4. 选择竞价型实例。如果您没有保存竞价型实例的实例 ID，请参阅the section called “查找竞价型实例”。
5. 依次选择实例状态、终止（删除）实例。
6. 当系统提示您确认时，选择终止（删除）。

Amazon CLI

手动终止竞价型实例

使用 terminate-instances 命令可手动终止竞价型实例。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

PowerShell

手动终止竞价型实例

使用 Remove-EC2Instance cmdlet。

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

竞价型实例中断

您可以在空闲的 EC2 容量上启动竞价型实例，以便在 Amazon EC2 需要收回这些容量时，就返还实例获得大幅折扣。我们将 Amazon EC2 回收 Spot 实例的情况称为 Spot 实例中断。

对竞价型实例的需求可能因时间不同而有显著的差异，竞价型实例 的可用性也会因为有多少未使用 EC2 实例可用而差别巨大。竞价型实例可能会中断。下面列出了 Amazon EC2 中断您的竞价型实例的可能原因：

容量

Amazon EC2 可以在需要收回时中断 Spot 实例。EC2 回收您的实例主要是为了重新调整容量用途，但也可能出于其他原因，如主机维护或硬件停用。

价格

Spot 价格高于您的最高价格。

您可以在竞价型请求中指定最高价格。但如果您指定了某个最高价，则实例被中断的频率将比您未指定时更高。

约束

如果您的竞价型请求包含约束（如启动组或可用区组），则当不再满足约束时，这些竞价型实例将成组终止。

当 Amazon EC2 中断竞价型实例时，它会终止、停止或休眠实例，具体取决于您在创建 Spot 请求时指定的中断行为。

内容

- 竞价型实例中断行为
- 为测试竞价型实例中断做好准备
- 启动竞价型实例中断
- 竞价型实例中断通知
- 查找中断的竞价型实例
- 确定 Amazon EC2 是否终止了竞价型实例
- 中断的竞价型实例的计费

竞价型实例中断行为

您可以在创建 Spot 请求时指定中断行为。以下是可能的中断行为：

- 停止
- Hibernate
- 终止

默认行为是 Amazon EC2 在竞价型实例中断时终止这些实例。

停止中断的竞价型实例

您可以指定 Amazon EC2 在竞价型实例中断时将其停止。竞价型实例请求类型必须为 `persistent`。您不能在竞价型实例请求中指定启动组。对于 EC2 实例集或竞价型实例集，请求类型必须为 `maintain`。

注意事项

- 只有 Amazon EC2 才能重新启动已停止的竞价型实例。
- 对于由 `persistent` 竞价型实例请求启动的竞价型实例：Amazon EC2 会在容量在同一可用区中可用且适用于已停止的实例的同一实例类型（必须使用相同的启动规范）时重新启动已停止的实例。
- 在停止竞价型实例后，您可以修改其部分实例属性，但不能修改实例类型。如果您分离或删除一个 EBS 卷，则在启动竞价型实例时不会附加该卷。如果您分离根卷并且 Amazon EC2 尝试启动竞价型实例，则该实例将无法启动，Amazon EC2 将终止已停止的实例。
- 当竞价型实例停止时，您可以将其终止。
- 如果取消竞价型实例请求、EC2 机群或竞价型实例集，Amazon EC2 将终止任何停止的关联竞价型实例。
- 在中断的竞价型实例停止后，您只需为保留的 EBS 卷付费。对于 EC2 实例集和 Spot 实例集，如果具有很多停止的实例，则可能会超出您的账户的 EBS 卷数限制。有关竞价型实例中断时如何收费的更多信息，请参阅 中断的竞价型实例的计费。
- 确保您熟知停止实例的影响。有关实例停止可能会发生情况的更多信息，请参阅 实例状态之间的区别。

休眠中断的竞价型实例

您可以指定 Amazon EC2 在竞价型实例中断时将其休眠。有关更多信息，请参阅 将您的 Amazon EC2 实例休眠。

Amazon EC2 现在为竞价型实例提供了与当前按需型实例相同的休眠体验。它提供了更广泛的支持，竞价型实例休眠现在支持：

- 更多支持的 AMI
- 更多支持的实例系列
- 用户启动的休眠

终止中断的竞价型实例

当 Amazon EC2 中断竞价型实例时，默认情况下会终止该实例，除非您指定了其他中断行为，例如停止或休眠。有关更多信息，请参阅 终止 Amazon EC2 实例。

为测试竞价型实例中断做好准备

对竞价型实例的需求可能因时间不同而有显著的差异，竞价型实例 的可用性也会因为有多少未使用 EC2 实例可用而差别巨大。竞价型实例可能会中断。因此，必须确保应用程序针对竞价型实例中断做好准备。

我们建议您遵循以下最佳实践，以便您为竞价型实例中断做好准备。

- 使用 Auto Scaling 组创建竞价型请求。如果您的竞价型实例被中断，Auto Scaling 组将自动启动替换实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的具有多个实例类型和购买选项的自动扩缩组。
- 使用包含所需软件配置的亚马逊机器映像（AMI），确保您的实例在请求完成时随时可以启动。您还可以使用用户数据在启动时运行命令。
- 停止或终止实例后，实例存储卷上的数据将丢失。将实例存储卷上的所有重要数据备份到持久性存储，例如 Amazon S3、Amazon EBS 或 Amazon DynamoDB。
- 在不会受竞价型实例终止影响的位置例行存储重要数据。例如，您可以使用 Amazon S3、Amazon EBS 或 DynamoDB。
- 将工作拆分为小的任务 (使用网格、Hadoop 或基于队列的架构) 或者使用检查点，以便您经常保存工作。
- 当实例处于较高的中断风险时，Amazon EC2 会向竞价型实例发出再平衡建议信号。您可以依靠再平衡建议来主动管理竞价型实例中断，而无需等待两分钟的竞价型实例中断通知。有关更多信息，请参阅EC2 实例再平衡建议。
- 使用两分钟的竞价型实例中断通知监控您的竞价型实例的状态。有关更多信息，请参阅竞价型实例中断通知。
- 虽然我们尽一切努力尽快提供这些警告，但您的竞价型实例可能会在我们提供此警告之前被中断。测试您的应用程序，确保它很好地处理了意外的实例终止，即使您正在监控再平衡建议信号和中断通知。您可以使用按需型实例来运行应用程序，然后自行终止该按需型实例，以便确认这一点。
- 使用 Amazon Fault Injection Service 运行受控的故障注入实验，以测试在竞价型实例中断时应用程序如何响应。有关更多信息，请参阅 Amazon Fault Injection Service 用户指南中的教程，使用 Amazon FIS 测试竞价型实例中断。

启动竞价型实例中断

您可以在 Amazon EC2 控制台中选择一个竞价型实例请求或竞价型实例集请求并启动竞价型实例中断，从而测试竞价型实例上的应用程序将如何处理中断。当您启动竞价型实例中断时，Amazon EC2 会向您发送通知，提示您的竞价型实例将在两分钟后中断，然后将在两分钟后中断该实例。

执行竞价型实例中断的底层服务是 Amazon Fault Injection Service（Amazon FIS）。有关 Amazon FIS 的信息，请参阅 Amazon Fault Injection Service。

> **Note**
>
> 中断行为包括 `terminate`、`stop` 和 `hibernate`。如果您将中断行为设置为 `hibernate`，则当您启动竞价型实例中断时，将会立即开始休眠过程。

除亚太地区（雅加达）、亚太地区（大阪）、中国（北京）、中国（宁夏）和中东地区（阿联酋）外，所有 Amazon Web Services 区域均支持启动竞价型实例中断。

内容

- 启动竞价型实例中断
- 验证竞价型实例中断
- 配额

启动竞价型实例中断

您可以使用 EC2 控制台快速启动竞价型实例中断。选择竞价型实例请求时，您可以启动一个竞价型实例的中断。选择竞价型实例集请求时，您可以同时启动多个竞价型实例的中断。

要进行更高级的实验来测试竞价型实例中断情况，您可以使用 Amazon FIS 控制台创建自己的实验。

使用 EC2 控制台启动竞价型实例请求中一个竞价型实例的中断

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests（竞价型实例请求）。
3. 选择一个竞价型实例请求，然后依次选择 Actions（操作）、Initiate interruption（启动中断）。不能选择多个竞价型实例请求来启动中断。

4. 在 Initiate Spot Instance interruption（启动竞价型实例中断）对话框的 Service access（服务访问权限）下，您可以使用默认角色，也可以选择一个现有的角色。要选择现有角色，请选择使用现有的服务角色，然后对于 IAM 角色，选择要使用的角色。
5. 准备好启动竞价型实例中断后，选择 Initiate interruption（启动中断）。

使用 EC2 控制台启动竞价型实例集请求中的一个或多个竞价型实例的中断

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests（竞价型实例请求）。
3. 选择一个竞价型实例集请求，然后依次选择操作、启动中断。您不能选择多个竞价型实例集请求来启动中断。
4. 在指定竞价型实例数量对话框中，对于要中断的实例数，输入要中断的竞价型实例数量，然后选择确认。

> **Note**
>
> 该数量不能超过实例集中竞价型实例的数量或每个实验 Amazon FIS 可以中断的竞价型实例数量的配额。

5. 在 Initiate Spot Instance interruption（启动竞价型实例中断）对话框的 Service access（服务访问权限）下，您可以使用默认角色，也可以选择一个现有的角色。要选择现有角色，请选择使用现有的服务角色，然后对于 IAM 角色，选择要使用的角色。
6. 准备好启动竞价型实例中断后，选择 Initiate interruption（启动中断）。

使用 Amazon FIS 控制台创建更高级的实验来测试竞价型实例中断

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests（竞价型实例请求）。
3. 依次选择 Actions（操作）、Create advanced experiments（创建高级实验）。

   Amazon FIS 控制台将打开。有关更多信息，请参阅《Amazon Fault Injection Service User Guide》中的 Tutorial: Test Spot Instance interruptions using Amazon FIS。

验证竞价型实例中断

启动中断后，将会发生以下情况：

- 竞价型实例收到一个 实例再平衡建议。
- 在 Amazon FIS 中断实例之前两分钟发出竞价型实例中断通知。
- 两分钟后，竞价型实例将会中断。
- 已被 Amazon FIS 停止的竞价型实例将在您将其重新启动之前一直处于停止状态。

验证实例在您启动中断后是否已中断

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航窗格中，在单独的浏览器选项卡或窗口中打开 Spot Requests（竞价型实例请求）和 Instances（实例）。
3. 对于竞价型请求，请选择竞价型实例请求或竞价型实例集请求。初始状态为 `fulfilled`。实例中断后，状态会根据中断行为出现以下变化：
   - `terminate` – 状态变为 `instance-terminated-by-experiment`。
   - `stop` – 实例的状态首先变为 `marked-for-stop-by-experiment`，然后变为 `instance-stopped-by-experiment`。
4. 对于 Instances（实例），选择该竞价型实例。初始状态为 `Running`。在收到竞价型实例中断通知后两分钟，状态会根据中断行为出现以下变化：
   - `stop` – 实例的状态首先变为 `Stopping`，然后变为 `Stopped`。
   - `terminate` – 实例的状态首先变为 `Shutting-down`，然后变为 `Terminated`。

配额

对于每个实验 Amazon FIS 可以中断的竞价型实例数量，您的 Amazon Web Services 账户 具有如下默认配额。

名称	默认值	可调整	说明
aws:ec2:send-spot-instance-interruptions 的目标竞价型实例	每个受支持的区域：5 个	是	在每个实验中，当您使用标签标识目标时 aws:ec2:send-spot-instance-interruptions 可以确定为目标的最大竞价型实例数量。

您可以请求提高配额。有关更多信息，请参阅《服务配额用户指南》中的 Requesting a quota increase。

要查看 Amazon FIS 的所有限额，请打开服务限额控制台。在导航窗格中，选择 Amazon services（ 服务 ），然后选择 Amazon Fault Injection Service。您还可以在Amazon Fault Injection Service用户指南Amazon Fault Injection Service中查看的所有配额。

竞价型实例中断通知

竞价型实例中断通知是在 Amazon EC2 停止或终止竞价型实例之前的两分钟发出的警告。如果您将休眠指定为中断行为，则会收到中断通知，但由于休眠过程立即开始，因此您不会提前两分钟收到警告。

正常处理竞价型实例中断的最佳方法是，设计应用程序以提供容错能力。为此，您可以利用竞价型实例中断通知。我们建议您每 5 秒检查一次这些中断通知。

中断通知将以 EventBridge 事件以及竞价型实例上的实例元数据项的形式提供。将尽最大努力发出中断通知。

EC2 Spot Instance Interruption Warning 事件

当 Amazon EC2 将要中断竞价型实例时，它在实际中断之前的两分钟发出一个事件（休眠除外，此时会收到中断通知，但不会提前两分钟，因为休眠是立即开始的）。Amazon EventBridge 可以检测该事件。有关 EventBridge 事件的更多信息，请参阅《Amazon EventBridge 用户指南》https://docs.amazonaws.cn/eventbridge/latest/userguide/。有关指导您如何创建和使用事件规则的详细示例，请参阅利用 Amazon EC2 Spot 实例中断通知。

以下是竞价型实例中断事件的示例。`instance-action` 的可能值为 `hibernate`、`stop` 或 `terminate`。

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Spot Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2a:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
```

```
    }
}
```

> **Note**
>
> 竞价型实例中断事件的 ARN 格式为 `arn:aws:ec2:availability-zone:instance/instance-id`。此格式不同于 EC2 资源的 ARN 格式。

instance-action

`instance-action` 项目指定操作及其大致执行时间 (采用 UTC 格式)。

如果竞价型实例标记为由 Amazon EC2 停止或终止，将在您的实例元数据中包含 `instance-action` 项。如果没有，则不显示。可以使用实例元数据服务版本 2（IMDSv2）检索 `instance-action`，如下所示。

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/spot/instance-action
```

Windows

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/instance-action
```

以下示例输出指示将停止此实例的时间。

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

以下示例输出指示将终止此实例的时间。

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

如果 Amazon EC2 没有准备停止或终止该实例，或者如果您自己终止了该实例，则实例元数据中不存在 `instance-action`，并且在尝试检索它时出现 HTTP 404 错误。

termination-time

`termination-time` 项目指定实例将收到关闭信号的大致时间（用 UTC 表示）。

> **Note**
>
> 为向后兼容而保留此项目；您应改为使用 `instance-action`。

如果您的竞价型实例被 Amazon EC2 标记为终止（由于中断行为设置为 `terminate` 的竞价型实例中断，或由于永久竞价型实例请求取消），则 `termination-time` 项目将出现在您的实例元数据中。如果没有，则不显示。可以使用 IMDSv2 检索 `termination-time`，如下所示。

Linux

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo termination_scheduled; fi
```

Windows

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/meta-data/spot/termination-time
```

下面是示例输出。

```
2015-01-05T18:02:00Z
```

如果 Amazon EC2 未准备终止实例（因为没有竞价型实例中断，或因为您的中断行为设置为 `stop` 或 `hibernate`），或者如果您自己终止了竞价型实例，则 `termination-time` 项目或者不存在于实例元数据中（这样您会收到 HTTP 404 错误），或者包含并非时间值的值。

如果 Amazon EC2 无法终止实例，请求状态将设置为 `fulfilled`。`termination-time` 值会将实例元数据保持原始大致时间（现已成为过去时间）。

查找中断的竞价型实例

在描述您的 EC2 实例时，结果将包括您的竞价型实例。竞价型实例的实例生命周期为 `spot`。竞价型实例的实例状态为 `stopped` 或 `terminated`，具体取决于您配置的中断行为。对于休眠的竞价型实例，实例状态为 `stopped`。

有关中断原因的更多详细信息，请查看 Spot 请求状态代码。有关更多信息，请参阅 the section called “获取竞价型实例请求的状态”。

Console

查找已中断的竞价型实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 应用以下筛选条件：实例生命周期 = 竞价型。
4. 应用实例状态 = 已停止或实例状态 = 已终止筛选条件，具体取决于您配置的中断行为。
5. 对于每个竞价型实例，在详细信息选项卡的实例详细信息下，找到状态转换消息。以下代码表明竞价型实例已中断。

   - `Server.SpotInstanceShutdown`
   - `Server.SpotInstanceTermination`

Amazon CLI

查找中断的竞价型实例

将 `--filters` 选项与 run-instances 命令结合使用。要仅在输出中列出实例 ID，请包括 `--query` 选项。

如果中断行为是终止竞价型实例，请使用以下示例：

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=terminated \
              Name=state-reason-code,Values=Server.SpotInstanceTermination \
    --query "Reservations[*].Instances[*].InstanceId"
```

如果中断行为是停止竞价型实例，请使用以下示例：

```
aws ec2 describe-instances \
    --filters Name=instance-lifecycle,Values=spot \
              Name=instance-state-name,Values=stopped \
              Name=state-reason-code,Values=Server.SpotInstanceShutdown \
    --query "Reservations[*].Instances[*].InstanceId"
```

PowerShell

查找中断的竞价型实例

使用 Get-EC2Instance cmdlet。

如果中断行为是终止竞价型实例，请使用以下示例：

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="terminated"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

如果中断行为是停止竞价型实例，请使用以下示例：

```
(Get-EC2Instance `
    -Filter @{Key="instance-lifecycle"; Values="spot"} `
            @{Key="instance-state-name"; Values="stopped"} `
            @{Key="state-reason-code";
 Values="Server.SpotInstanceTermination"}).Instances.InstanceId
```

确定 Amazon EC2 是否终止了竞价型实例

竞价型实例将一直运行，直到 Amazon EC2 因竞价型实例中断而将其终止，或者直到您自行终止该竞价型实例。有关更多信息，请参阅 the section called “中断行为”。

竞价型实例终止后，您可以使用 Amazon CloudTrail 查看 Amazon EC2 是否终止了该实例。如果 CloudTrail 日志包含 `BidEvictedEvent`，则表示 Amazon EC2 终止了竞价型实例。相反，如果您看到一个 `TerminateInstances` 事件，则表示用户终止了竞价型实例。

或者，如果您想收到 Amazon EC2 将中断您的竞价型实例的通知，请使用 Amazon EventBridge 来响应 EC2 竞价型实例中断警告事件。

在 CloudTrail 中查看 BidEvictedEvent 事件

1. 访问 https://console.aws.amazon.com/cloudtrail/，打开 CloudTrail 控制台。
2. 在导航窗格中，选择事件历史记录。
3. 从筛选器列表中，选择事件名称，然后在右侧的筛选字段中，输入 **BidEvictedEvent**。
4. （可选）选择一个时间范围。
5. 如果列表不为空，请从生成的条目中选择 BidEvictedEvent 以打开其详细信息页面。您可以在事件记录窗格中找到有关竞价型实例的信息，包括竞价型实例的 ID。以下是事件记录的示例。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "ec2.amazonaws.com"
    },
    "eventTime": "2016-08-16T22:30:00Z",
    "eventSource": "ec2.amazonaws.com",
    "userAgent": "ec2.amazonaws.com",
    "sourceIPAddress": "ec2.amazonaws.com",
    "eventName": "BidEvictedEvent",
    "awsRegion": "us-east-2",
    "eventID": "d27a6096-807b-4bd0-8c20-a33a83375054",
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123456789012",
    "RequestParameters": null,
    "ResponseElements": null,
    "serviceEventDetails": {
        "instanceIdSet": [
          "i-1eb2ac8eEXAMPLE"
        ]
    }
}
```

6. 如果您没有找到 BidEvictedEvent 事件的条目，请输入 **TerminateInstances** 作为事件名称。有关 TerminateInstances 事件记录的更多信息，请参阅the section called “Amazon EC2 API 事件示例”。

中断的竞价型实例的计费

当竞价型实例被中断时，将向您收取实例和 EBS 卷使用的费用，并且可能产生其他费用，如下所示。

实例使用情况

谁中断了竞价型实例	操作系统	在第一个小时内中断	在第一个小时后的任一小时内中断
如果您停止或终止 Spot 实例	Windows 和 Linux（不包括 SUSE）	按使用的秒数收费	按使用的秒数收费
	SUSE	按完整小时收费（即使您的使用时间是部分小时也是如此）	按使用的完整小时数收费，对于中断的部分完整小时，按完整小时收费
如果 Amazon EC2 中断竞价型实例	Windows 和 Linux（不包括 SUSE）	免费	按使用的秒数收费
	SUSE	免费	按使用的完整小时数收费，对于中断的部分小时，不收费

EBS 卷使用情况

在中断的竞价型实例停止后，您只需为保留的 EBS 卷付费。

对于 EC2 实例集和 Spot 实例集，如果具有很多停止的实例，则可能会超出您的账户的 EBS 卷数限制。

EC2 实例再平衡建议

EC2 实例再平衡建议是一个信号，可在竞价型实例处于较高的中断风险时通知您。信号可能比该两分钟的 Spot 实例中断通知更早到达，从而让您有机会主动管理 Spot 实例。您可以决定将工作负载再平衡到不处于较高中断风险的新的或现有的竞价型实例。

Amazon EC2 并不总能在两分钟的竞价型实例中断通知之前发送再平衡建议信号。因此，再平衡建议信号可能会随两分钟的中断通知一起到达。

再平衡建议是作为 EventBridge 事件以及竞价型实例上的实例元数据中的项目提供的。尽最大努力发出事件。

> **Note**
>
> 再平衡建议仅支持 2020 年 11 月 5 日 00:00 UTC 之后推出的竞价型实例。

内容

- 您可以采取的再平衡操作
- 监控再平衡建议信号
- 使用再平衡建议信号的服务

您可以采取的再平衡操作

以下是您可以采取的一些可能的再平衡操作：

正常关闭

当您收到竞价型实例的再平衡建议信号时，您可以启动实例关闭过程，其中可能包括确保在停止进程之前完成它们。例如，您可以将系统或应用程序日志上传到 Amazon Simple Storage Service (Amazon S3)、关闭 Amazon SQS 工作程序或从域名系统 (DNS) 中完成注销。您还可以将工作保存在外部存储器中，稍后再恢复工作。

防止计划新工作

当您收到竞价型实例的再平衡建议信号时，您可以阻止在实例上计划新工作，同时继续使用实例直到计划的工作完成。

主动启动新的替换实例

您可以配置 Auto Scaling 组、EC2 实例集或竞价型实例集，以在系统发出再平衡建议信号时自动启动替换竞价型实例集。有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Use Capacity Rebalancing to handle Amazon EC2 Spot interruptions，以及本用户指南中的在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例。

监控再平衡建议信号

您可以监控再平衡建议信号，以便在发出该信号时，可以执行上一节中指定的操作。再平衡建议信号作为发送到 Amazon EventBridge（以前称为 Amazon CloudWatch Events）的事件和竞价型实例上的实例元数据提供。

监控再平衡建议信号：

- 使用 Amazon EventBridge
- 使用实例元数据

使用 Amazon EventBridge

当发出竞价型实例的再平衡建议信号时，该信号的事件将被发送到 Amazon EventBridge。如果 EventBridge 检测到与规则中定义的模式匹配的事件模式，则 EventBridge 调用规则中指定的一个或多个目标。

以下是再平衡建议信号的示例事件。

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Instance Rebalance Recommendation",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0"
    }
}
```

下列字段构成规则中定义的事件模式：

`"detail-type": "EC2 Instance Rebalance Recommendation"`

确定该事件是再平衡建议事件

`"source": "aws.ec2"`

识别来自 Amazon EC2 的事件

创建 EventBridge 规则

当事件模式与规则匹配时，您可以编写 EventBridge 规则并自动执行要采取的操作。

下面的示例可创建 EventBridge 规则，以便在 Amazon EC2 每次发出再平衡建议信号时发送电子邮件、短信或移动推送通知。信号作为 `EC2 Instance Rebalance Recommendation` 事件发出，这将触发规则定义的操作。

创建 EventBridge 规则之前，您必须为电子邮件、短信或移动推送通知创建 Amazon SNS 主题。

要为再平衡建议事件创建 EventBridge 规则

1. 打开位于 https://console.aws.amazon.com/events/ 的 Amazon EventBridge 控制台。
2. 选择 Create rule（创建规则）。
3. 对于定义规则详细信息，请执行以下操作：

   a. 输入规则的 Name (名称) 和“Description (描述)”（可选）。

      规则不能与同一区域中的另一个规则和同一事件总线上的名称相同。

   b. 对于 Event bus（事件总线），选择 default（默认）。当您账户中的某个 Amazon 服务生成一个事件时，它始终会发送到您账户的默认事件总线。

   c. 对于 Rule type（规则类型），选择 Rule with an event pattern（具有事件模式的规则）。

   d. 选择下一步。

4. 对于 Build event pattern（构建事件模式），执行以下操作：

   a. 对于 Event source（事件源），选择 Amazon 事件或 EventBridge 合作伙伴事件。

   b. 对于 Event pattern（事件模式），在此示例中，您将指定以下事件模式以匹配 `EC2 Instance Rebalance Recommendation` 事件，然后选择 Save（保存）。

```
{
 "source": ["aws.ec2"],
 "detail-type": ["EC2 Instance Rebalance Recommendation"]
}
```

      要添加事件模式，您可以通过选择 Event pattern form（事件模式表）使用模板，或者通过选择 Custom pattern (JSON editor)（自定义模式（JSON 编辑器））指定您自己的模式，如下所示：

      i. 要使用模板创建事件模式，请执行以下操作：

         A. 选择 Event pattern form（事件模式表）。

         B. 对于 Event source（事件源），选择 Amazon services（服务）。

         C. 对于 Amazon 服务，选择 EC2 竞价型实例集。

         D. 对于 Event type（事件类型），选择 EC2 Instance Rebalance Recommendation（EC2 实例再平衡建议）。

E. 要自定义模板，请选择 Edit pattern（编辑模式），然后进行更改以匹配示例事件模式。

ii. （可选）要指定自定义事件模式，请执行以下操作：

A. 选择 Custom pattern (JSON editor)（自定义模式（JSON 编辑器））。

B. 在 Event pattern（事件模式）框中，为此示例添加事件模式。

c. 选择下一步。

5. 对于 Select target(s)（选择目标），请执行以下操作：

a. 对于 Target types（目标类型），选择 Amazon service（服务）。

b. 对于 Select a target（选择目标），选择 SNS topic（SNS 主题）以在事件发生时发送电子邮件、短信或移动推送通知。

c. 对于 Topic（主题），选择现有主题。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。有关更多信息，请参阅Amazon Simple Notification Service 开发人员指南中的使用 Amazon SNS 进行应用程序对人 (A2P) 消息传送。

d. （可选）在 Additional settings（其他设置）下，您可以选择配置其他设置。有关更多信息，请参阅《Amazon EventBridge User Guide》中的 Creating Amazon EventBridge rules that react to events（步骤 16）。

e. 选择下一步。

6. （可选）对于 Tags（标签），您可以选择向规则分配一个或多个标签，然后选择 Next（下一步）。

7. 对于 Review and create（查看与创建），执行以下操作：

a. 查看规则的详细信息并根据需要对其进行修改。

b. 选择 Create rule（创建规则）。

有关更多信息，请参阅 Amazon EventBridge 用户指南中的 Amazon EventBridge 规则和 Amazon EventBridge 事件模式。

使用实例元数据

实例元数据类别 `events/recommendations/rebalance` 提供为 Spot 实例发出再平衡建议信号的大致时间 (UTC)。

我们建议您每 5 秒检查一次再平衡建议信号，这样您就不会错过对再平衡建议采取行动的机会。

如果竞价型实例收到再平衡建议，则信号发出的时间将显示在实例元数据中。您可以按如下方式检索信号发出的时间。

IMDSv2

Linux

在 Linux 实例上运行以下命令。

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

在 Windows 实例上运行以下 cmdlet

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

IMDSv1

Linux

在 Linux 实例上运行以下命令。

```
curl http://169.254.169.254/latest/meta-data/events/recommendations/rebalance
```

Windows

在 Windows 实例上运行以下 cmdlet。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/events/
recommendations/rebalance
```

以下是示例输出，表示为竞价型实例发出的再平衡建议信号时间 (UTC)。

```
{"noticeTime": "2020-10-27T08:22:00Z"}
```

如果尚未为实例发出信号，则 `events/recommendations/rebalance` 不存在，您将在尝试检索该信号时收到 HTTP 404 错误。

使用再平衡建议信号的服务

Amazon EC2 Auto Scaling、EC2 实例集和竞价型实例集使用再平衡建议信号让您可以轻松地在运行中的实例收到两分钟的竞价型实例中断通知之前，主动使用新的竞价型实例扩展您的队列，从而帮助您维护工作负载的可用性。您可以让这些服务监控和主动响应影响您的竞价型实例可用性的更改。有关更多信息，请参阅下列内容：

- 《Amazon EC2 Auto Scaling User Guide》中的 Use Capacity Rebalancing to handle Amazon EC2 Spot interruptions
- 本用户指南中 EC2 实例集和竞价型实例集主题中的在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例

竞价放置分数

竞价放置分数功能可以根据您的竞价容量需求推荐一个Amazon区域或可用区。竞价型容量会发生波动，所以您无法确保始终获得所需的容量。竞价放置分数表示竞价请求在某个区域或可用区中获得成功的可能性。

> **Note**
>
> 在中国（北京）或中国（宁夏）区域使用竞价放置分数功能时，Amazon EC2 只会返回该区域的分数，或者该区域中发出调用的每个可用区的分数。

> **Note**
>
> 竞价放置分数不能提供在可用容量或中断风险方面的任何保证。竞价放置分数仅可作为建议使用。

使用案例

您可以使用竞价放置分数功能完成以下操作：

- 根据需要将竞价型计算容量重新定位和扩展到不同区域，以应对当前区域内的容量需求增加或可用容量减少。
- 确定在其中运行单个可用区工作负载的最佳可用区。
- 模拟未来的竞价型容量需求，以便您可以选择一个最佳区域来扩展基于竞价的工作负载。
- 确定实例类型的最佳组合以满足竞价容量需求。

内容

- 限制
- 成本
- 竞价放置分数的工作原理
- Spot 置放分数所需的权限
- 计算 Spot 置放分数

限制

- 目标容量限制 – 您的竞价放置分数目标容量限制取决于您最近的竞价用量，并同时考虑潜在的使用量增长。如果您最近没有产生竞价用量，那么我们会为您提供一个较低的默认限制，这个默认限制与您的竞价请求限制一致。
- 请求配置限制 – 如果我们检测到与竞价放置分数功能的预期用途无关的模式，则我们可以限制 24 小时内的新请求配置的数量。如果达到限制，则可以重试曾使用过的请求配置，但在下一个 24 小时期间之前，您无法指定新的请求配置。
- 实例类型的最小数量 – 如果您指定实例类型，则必须指定至少三种不同的实例类型，否则 Amazon EC2 将返回较低的竞价放置分数。同样，如果您指定实例属性，则其必须解析为至少三种不同的实例类型。如果实例类型的名称不同，则被视为不同的实例类型。例如，m5.8xlarge、m5a.8xlarge 和 m5.12xlarge 都会被视作不同的实例类型。

成本

使用竞价放置分数功能不会收取任何额外费用。

竞价放置分数的工作原理

在使用竞价放置分数功能时，您先指定竞价型实例的计算要求，然后 Amazon EC2 会返回竞价请求可能成功的区域的分数或可用区。每个区域或可用区的评分范围为 1 到 10，其中 10 表示竞价请求的成功性较高，1 则表示竞价请求的成功性较低。

请按照以下步骤使用竞价放置分数功能：

- 步骤 1：指定竞价需求
- 步骤 2：筛选竞价放置分数响应
- 步骤 3：审查建议
- 步骤 4：采纳建议

步骤 1：指定竞价需求

首先，请指定所需的目标竞价容量和计算要求，如下所示：

1. 指定目标竞价容量，也可以选择目标容量单位。

   您可以根据实例数或 vCPU 数，或者根据以 MiB 为单位的内存量来指定所需的目标竞价容量。若根据 vCPU 数或内存量来指定目标容量，则必须将目标容量单位指定为 `vcpu` 或 `memory-mib`。否则即默认为实例数。

   若根据 vCPU 数或内存量指定目标容量，则您可以在计算总容量时使用这些单位。例如，若要混合使用不同大小的实例，则可以将目标容量指定为 vCPU 的总数。然后竞价放置分数功能会根据 vCPU 数来考虑请求中的每种实例类型，并计算 vCPU 总数，而不是在合计目标容量时计算实例总数。

   例如，假设您指定的总目标容量为 30 个 vCPU，并且您的实例类型列表中包括 c5.xlarge（4 个 vCPU）、m5.2xlarge（8 个 vCPU）和 r5.large（2 个 vCPU）。若要实现总计 30 个 vCPU，则您可以获得 2 个 c5.xlarge（2*4 vCPU）、2 个 m5.2xlarge（2*8 个 vCPU）和 3 个 r5.large（3*2 个 vCPU）的组成。
2. 指定实例类型或实例属性。

   您可以指定所用的实例类型，也可以指定计算要求所需的实例属性，然后由 Amazon EC2 来确定具有这些属性的实例类型。这称为基于属性的实例类型选择。

您无法在同一竞价放置分数请求中同时指定实例类型和实例属性。

如果您指定实例类型，则必须至少指定三种不同的实例类型，否则 Amazon EC2 将返回较低的竞价放置分数。同样，如果您指定实例属性，则其必须解析为至少三种不同的实例类型。

有关指定竞价需求的不同方法的示例，请参阅 示例配置。

步骤 2：筛选竞价放置分数响应

Amazon EC2 会计算区域或区域内每个可用区的竞价放置分数，并返回您的竞价请求可能成功的区域或可用区的分数。默认为返回区域的分数。如果您计划在单个可用区中启动所有竞价容量，那么请求一个已评分的可用区列表会非常有帮助。

步骤 3：审查建议

每个区域或可用区的竞价放置分数是根据目标容量、实例类型的组成、历史和当前竞价使用趋势以及请求的时间来计算的。由于竞价容量不断波动，因此在不同时间计算同一竞价放置分数请求时，可能会产生不同的分数。

区域和可用区的评分范围为 1 至 10。分数为 10 表示您的竞价请求具有较高的成功性（但不能保证）。分数为 1 则表示您的竞价请求几乎没有成功的可能。即使是不同的区域或可用区，也可能会返回相同的分数。

如果返回的分数较低，则您可以编辑计算要求并重新计算分数。您还可以在一天中的不同时间为同一个计算要求请求竞价放置分数建议。

步骤 4：采纳建议

只有在竞价请求具有与竞价放置分数配置（目标容量、目标容量单位和实例类型或实例属性）完全相同的配置时，竞价放置分数才是相关的，并且被配置为使用 `capacity-optimized` 分配策略。否则，获得可用竞价型容量的可能性将与分数不一致。

尽管竞价放置分数可作为指导，但无法保证完全或部分满足您的竞价请求，所以您可以使用以下信息来获得最佳结果：

- 使用相同的配置 – 仅当 Auto Scaling 组、EC2 机群或竞价型实例集中的竞价请求配置（目标容量、目标容量单位、实例类型或实例属性）与您为了获取竞价放置分数而输入的配置相同时，竞价放置分数才是相关的。

如果您在竞价放置分数请求中使用了基于属性的实例类型选择，就可以使用基于属性的实例类型选择来配置 Auto Scaling 组、EC2 机群或竞价型实例集。有关更多信息，请参阅 使用基于属性的实例类型选择创建一个混合实例组 和 指定 EC2 实例集或竞价型实例集的实例类型选择属性。

> **Note**
>
> 如果您根据 vCPU 数或内存量指定了目标容量，并且在竞价放置分数配置中指定了实例类型，那么请注意，您当前无法在 Auto Scaling 组、EC2 机群或竞价型实例集中创建此配置。相反，您必须使用 `WeightedCapacity` 参数来手动设置实例权重。

- 使用 **`capacity-optimized`** 分配策略 – 任何分数都假定您的实例集请求将被配置为使用所有可用区（用于跨区域请求容量）或单个可用区（若在一个可用区中请求容量）和用于成功请求竞价容量的 `capacity-optimized` 竞价型分配策略。如果您使用其它分配策略，例如 `lowest-price`，则获得竞价容量的可能性会与分数不一致。
- 立即根据分数采取行动 – 竞价放置分数建议反映了请求时的可用竞价容量，由于竞价容量会发生波动，所以在不同时间计算相同的配置可能会得到不同的分数。尽管 10 分意味着竞价型容量请求具有较高的成功性，但不能保证成功，所以为了获得最佳效果，我们建议您立即根据分数采取行动。我们还建议您在每次尝试请求容量时都获取新的分数。

Spot 置放分数所需的权限

默认情况下，IAM 身份（用户、角色或组）不具备使用 the section called “竞价放置分数”的权限。要允许 IAM 身份使用 Spot 置放分数，则您必须创建一个 IAM 策略，以授予其使用 `ec2:GetSpotPlacementScores` EC2 API 操作的权限。然后，将策略附加到需要此权限的 IAM 身份。

以下为授予使用 `ec2:GetSpotPlacementScores` EC2 API 操作权限的 IAM policy 的示例。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:GetSpotPlacementScores",
            "Resource": "*"
        }
```

```
    ]
}
```

有关编辑 IAM policy 的信息，请参阅 IAM 用户指南中的编辑 IAM policy。

要提供访问权限，请为您的用户、组或角色添加权限：

- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。
- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

计算 Spot 置放分数

您可以根据目标容量和计算要求计算 Spot 置放分数。有关更多信息，请参阅 the section called “竞价放置分数的工作原理”。

所需的权限

确保拥有所需的权限。有关更多信息，请参阅 the section called “所需的权限”。

选项

- 使用实例属性计算
- 使用实例类型计算
- 使用 Amazon CLI 计算

需要一种自动化的解决方案？ 无需按照本用户指南中的手动步骤操作，只需构建一个竞价置放分数跟踪器控制面板，自动捕获分数并将其存储在 Amazon CloudWatch 中即可。有关更多信息，请参阅有关如何在 Amazon 上构建竞价置放评分跟踪器控制面板的指南。

使用实例属性计算

如需通过指定实例属性来计算竞价放置分数

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 从顶部导航栏中，选择要在其中评估竞价放置分数的区域。
4. 选择请求竞价型实例旁边的向下箭头，然后选择计算竞价放置分数。
5. 选择 Enter requirements（输入要求）。
6. 对于 Target capacity（目标容量），请根据 instances（实例）或 vCPUs (vCPU) 的数量或者根据 memory (MiB)（内存 (MiB)）的数量来输入所需的容量。
7. 对于 Instance type requirements（实例类型要求），如需指定您的计算要求并使 Amazon EC2 识别符合这些要求的最佳实例类型，请选择 Specify instance attributes that match your compute requirements（指定符合计算要求的实例属性）。
8. 对于 vCPU，输入所需的最小和最大 vCPU 数。要指定没有限制，请选择 No minimum（没有最小值）和/或 No maximum（没有最大值）。
9. 对于 Memory (GiB)（内存 (GiB)），输入所需的最小和最大内存量。要指定没有限制，请选择 No minimum（没有最小值）和/或 No maximum（没有最大值）。
10. 对于 CPU architecture（CPU 架构），选择所需的实例架构。
11. （可选）对于 Additional instance attributes（其它实例属性），您可以选择指定一个或多个属性以更详细地表达计算要求。每个额外属性都会进一步增加对您的请求的限制。您可以省略其它属性；如果省略，则使用默认值。有关每个属性及其默认值的描述，请参阅 get-spot-placement-scores。
12. （可选）要查看具有指定属性的实例类型，请展开 Preview matching instance types（预览匹配的实例类型）。要排除在安置性评量中使用的实例类型，请选择实例，然后选择 Exclude selected instance types（排除选定的实例类型）。
13. 选择 Load placement scores（加载放置分数），并审查结果。
14. （可选）要在显示的区域中显示可用区的 Spot 置放分数，请选择提供每个可用区的置放分数复选框。如果您希望在单个可用区中启动所有竞价容量，那么一个已评分的可用区列表会非常有帮助。
15. （可选）如需编辑计算要求并获得新的放置分数，请选择 Edit（编辑）以进行必要的调整，然后选择 Calculate placement scores（计算放置分数）。

使用实例类型计算

如需通过指定实例类型来计算竞价放置分数

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 从顶部导航栏中，选择要在其中评估竞价放置分数的区域。
4. 选择请求竞价型实例旁边的向下箭头，然后选择计算竞价放置分数。
5. 选择 Enter requirements（输入要求）。
6. 对于 Target capacity（目标容量），请根据 instances（实例）或 vCPUs (vCPU) 的数量或者根据 memory (MiB)（内存 (MiB)）的数量来输入所需的容量。
7. 对于 Instance type requirements（实例类型要求），如需指定要使用的实例类型，请选择 Manually select instance types（手动选择实例类型）。
8. 选择 Select instance types（选择实例类型），选择要使用的实例类型，然后选择 Select（选择）。如需快速查找实例类型，您可以使用筛选条件来按不同属性筛选实例类型。
9. 选择 Load placement scores（加载放置分数），并审查结果。
10. （可选）要在显示的区域中显示可用区的 Spot 置放分数，请选择提供每个可用区的置放分数复选框。如果您希望在单个可用区中启动所有竞价容量，那么一个已评分的可用区列表会非常有帮助。
11. （可选）如需编辑实例类型列表并获得新的放置分数，请选择 Edit（编辑）以进行必要的调整，然后选择 Calculate placement scores（计算放置分数）。

使用 Amazon CLI 计算

如需计算竞价放置分数

1. （可选）如需生成所有可以指定为竞价放置分数配置参数，请使用 get-spot-placement-scores 命令和 `--generate-cli-skeleton` 参数。

```
aws ec2 get-spot-placement-scores \
    --region cn-north-1 \
    --generate-cli-skeleton
```

下面是示例输出。

```
{
    "InstanceTypes": [
```

```
        ""
    ],
    "TargetCapacity": 0,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "RegionNames": [
        ""
    ],
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": [
            "x86_64_mac"
        ],
        "VirtualizationTypes": [
            "hvm"
        ],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 0,
                "Max": 0
            },
            "MemoryMiB": {
                "Min": 0,
                "Max": 0
            },
            "CpuManufacturers": [
                "amd"
            ],
            "MemoryGiBPerVCpu": {
                "Min": 0.0,
                "Max": 0.0
            },
            "ExcludedInstanceTypes": [
                ""
            ],
            "InstanceGenerations": [
                "previous"
            ],
            "SpotMaxPricePercentageOverLowestPrice": 0,
            "OnDemandMaxPricePercentageOverLowestPrice": 0,
            "BareMetal": "excluded",
            "BurstablePerformance": "excluded",
            "RequireHibernateSupport": true,
            "NetworkInterfaceCount": {
                "Min": 0,
```

```
                "Max": 0
            },
            "LocalStorage": "included",
            "LocalStorageTypes": [
                "hdd"
            ],
            "TotalLocalStorageGB": {
                "Min": 0.0,
                "Max": 0.0
            },
            "BaselineEbsBandwidthMbps": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorTypes": [
                "fpga"
            ],
            "AcceleratorCount": {
                "Min": 0,
                "Max": 0
            },
            "AcceleratorManufacturers": [
                "amd"
            ],
            "AcceleratorNames": [
                "vu9p"
            ],
            "AcceleratorTotalMemoryMiB": {
                "Min": 0,
                "Max": 0
            }
        }
    },
    "DryRun": true,
    "MaxResults": 0,
    "NextToken": ""
}
```

2. 使用上一步的输出创建 JSON 配置文件，然后按如下方式进行配置：

   a. 对于 `TargetCapacity`（目标容量），请根据 instances（实例）或 vCPUs (vCPU) 的数量或者根据 memory (MiB)（内存 (MiB)）的数量来输入所需的容量。

b. 对于 TargetCapacityUnitType，请输入目标容量的单位。如果省略此参数，则默认为 units。

有效值：units（转换为实例数）| vcpu | memory-mib

c. 对于 SingleAvailabilityZone（单个可用区），请指定 true 以获得返回已评分的可用区列表的响应。如果您希望在单个可用区中启动所有竞价容量，那么一个已评分的可用区列表会非常有帮助。如果省略此参数，则默认为 false，然后响应会返回已评分的区域。

d. （可选）对于 RegionNames，请指定要用作筛选条件的区域。此处必须指定区域代码，例如 cn-north-1。

使用区域筛选条件后，响应将仅返回您指定的区域。若将 SingleAvailabilityZone 指定为 true，则响应仅返回指定区域中的可用区。

e. 配置中可以包括 InstanceTypes（实例类型）或 InstanceRequirements（实例要求），但两者不可同时包含于同一配置中。

在 JSON 配置中指定以下之一：

- 如需指定实例类型列表，请指定 InstanceTypes 参数中的实例类型。指定至少三种不同的实例类型。若只指定了一种或两种实例类型，则竞价放置分数将返回一个较低的分数。有关实例类型的列表，请参阅 Amazon EC2 实例类型。
- 如需指定实例属性以便 Amazon EC2 识别与这些属性匹配的实例类型，请指定位于 InstanceRequirements（实例要求）结构中的属性。

必须提供 VCpuCount、MemoryMiB 和 CpuManufacturers 的值。您可以省略其它属性；如果省略，则使用默认值。有关每个属性及其默认值的描述，请参阅 get-spot-placement-scores。

有关示例配置，请参阅 示例配置。

3. 若要获取您在 JSON 文件中指定的要求的竞价放置分数，请使用 get-spot-placement-scores 命令，然后使用 --cli-input-json 参数为 JSON 文件指定名称和路径。

```
aws ec2 get-spot-placement-scores \
    --region cn-north-1 \
    --cli-input-json file://file_name.json
```

如果 SingleAvailabilityZone 被设置为 false 或省略（省略即默认为 false）时的输出示例：返回已评分的区域。

```
{
    "SpotPlacementScores": [
        {
            "Region": "cn-north-1",
            "Score": 9
        }
    ]
}
```

如果 SingleAvailabilityZone 被设置为 true 时的输出示例：返回可用区的评分列表。

```
{
    "SpotPlacementScores": [
        {
            "Region": "cn-north-1",
            "AvailabilityZoneId": "cnn1-az2",
            "Score": 9
        },
        {
            "Region": "cn-north-1",
            "AvailabilityZoneId": "cnn1-az4",
            "Score": 9
        },
        {
            "Region": "cn-north-1",
            "AvailabilityZoneId": "cnn1-az1",
            "Score": 9
        }
    ]
}
```

示例配置

使用 Amazon CLI，您可以使用以下示例配置。

示例配置

- 示例：指定实例类型和目标容量
- 示例：根据内存指定实例类型和目标容量
- 示例：为基于属性的实例类型选择指定属性

- 示例：为基于属性的实例类型选择指定属性并返回可用区的评分列表

示例：指定实例类型和目标容量

以下示例配置指定了三种不同的实例类型和 500 个竞价型实例的目标竞价容量。

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
    ],
    "TargetCapacity": 500
}
```

示例：根据内存指定实例类型和目标容量

以下示例配置指定了三种不同的实例类型和 500000MiB 内存的目标竞价容量，其中要启动的竞价型实例数必须能够提供总计 500000MiB 的内存。

```
{
    "InstanceTypes": [
        "m5.4xlarge",
        "r5.2xlarge",
        "m4.4xlarge"
    ],
    "TargetCapacity": 500000,
    "TargetCapacityUnitType": "memory-mib"
}
```

示例：为基于属性的实例类型选择指定属性

以下示例配置是针对基于属性的实例类型选择而进行的配置，以及该示例配置的文本说明。

```
{
    "TargetCapacity": 5000,
    "TargetCapacityUnitType": "vcpu",
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
```

```
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

InstanceRequirementsWithMetadata

要使用基于属性的实例类型选择，您必须在配置中包括 InstanceRequirementsWithMetadata 结构，并为竞价型实例指定所需的属性。

在上述示例中，指定了以下必要实例属性：

- ArchitectureTypes – 实例类型的架构类型必须为 arm64。
- VirtualizationTypes – 实例类型的虚拟化类型必须为 hvm。
- VCpuCount – 实例类型的 vCPU 数量必须最少为 1 个，最多为 12 个。
- MemoryMiB – 实例类型的最小内存必须为 512MiB。若省略 Max 参数，则表示没有最大限制。

请注意，您可以指定其它几个可选属性。有关属性列表，请参阅 get-spot-placement-scores。

TargetCapacityUnitType

TargetCapacityUnitType 参数为目标容量指定单位。在该示例中，目标容量是 5000，目标容量单位类型是 vcpu，它们共同指定了所需的 5000 个 vCPU 目标容量，其中要启动的竞价型实例数必须能够提供总计 5000 个 vCPU。

示例：为基于属性的实例类型选择指定属性并返回可用区的评分列表

以下示例配置是针对基于属性的实例类型选择而进行的配置。若指定 "SingleAvailabilityZone": true，则响应将返回已评分的可用区列表。

```
{
    "TargetCapacity": 1000,
    "TargetCapacityUnitType": "vcpu",
    "SingleAvailabilityZone": true,
    "InstanceRequirementsWithMetadata": {
        "ArchitectureTypes": ["arm64"],
```

```
        "VirtualizationTypes": ["hvm"],
        "InstanceRequirements": {
            "VCpuCount": {
                "Min": 1,
                "Max": 12
            },
            "MemoryMiB": {
                "Min": 512
            }
        }
    }
}
```

竞价型实例请求的服务相关角色

Amazon EC2 使用服务相关角色获取代表您调用其他Amazon服务所需的权限。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Web Services 服务 直接相关。服务相关角色提供了一种将权限委托给 Amazon Web Services 服务 的安全方式，因为只有相关服务才能担任服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色。

Amazon EC2 使用名为 Amazon Web Services ServiceRoleForEC2Spot 的服务相关角色代表您启动和管理 Spot 实例。

AWSServiceRoleForEC2Spot 授予的权限

Amazon EC2 使用 Amazon Web Services ServiceRoleForEC2Spot 完成以下操作：

- `ec2:DescribeInstances` – 描述竞价型实例
- `ec2:StopInstances` – 停止竞价型实例
- `ec2:StartInstances` – 启动竞价型实例

创建服务相关角色

在大多数情况下，无需手动创建服务相关角色。Amazon EC2 在您首次使用控制台请求 Spot 实例时创建 Amazon Web Services ServiceRoleForEC2Spot 服务相关角色。

如果在 2017 年 10 月之前具有活动Spot 实例请求（此时 Amazon EC2 开始支持该服务相关角色），则 Amazon EC2 在您的Amazon账户中创建 Amazon Web Services ServiceRoleForEC2Spot 角色。有关更多信息，请参阅IAM 用户指南中的我的账户中出现新角色。

如果您使用 Amazon CLI 或 API 来请求竞价型实例，则必须确保此角色存在。

要使用控制台创建 Amazon Web Services ServiceRoleForEC2Spot

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）。
3. 选择创建角色。
4. 在 Select type of trusted entity (选择受信任实体的类型) 页面上，依次选择 EC2、EC2 - Spot Instances (EC2 - Spot 实例) 和 Next: Permissions (下一步: 权限)。
5. 在下一页上，选择 Next:Review（下一步：审核）。
6. 在 Review (审核) 页面上，选择 Create role (创建角色)。

要使用 Amazon CLI 创建 Amazon Web Services ServiceRoleForEC2Spot

如下所示使用 create-service-linked-role 命令。

```
aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

如果您不再需要使用 Spot 实例，我们建议您删除 Amazon Web Services ServiceRoleForEC2Spot 角色。从账户中删除该角色后，如果您请求竞价型实例，Amazon EC2 将再次创建该角色。

授予对用于加密的 AMI 和 EBS 快照的客户托管密钥的访问权限

如果为竞价型实例指定加密的 AMI 或加密的 Amazon EBS 快照，并且您使用客户托管密钥进行加密，则必须为 AWSServiceRoleForEC2Spot 角色授予使用客户托管密钥的权限，以便 Amazon EC2 可以代表您启动竞价型实例。为此，您必须在客户托管密钥中添加授权，如以下过程中所示。

在提供权限时，授权是密钥策略的替代方法。有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的使用授权和在 Amazon KMS 中使用密钥策略。

为 Amazon Web Services ServiceRoleForEC2Spot 角色授予使用客户托管密钥的权限

- 使用 create-grant 命令在客户托管密钥中添加授权，并指定授予权限的委托人（Amazon Web Services ServiceRoleForEC2Spot 服务相关角色），以执行授权允许的操作。客户托管密钥由 `key-id` 参数和客户托管密钥的 ARN 指定。委托人是由 `grantee-principal` 参数和 Amazon Web Services ServiceRoleForEC2Spot 服务相关角色的 ARN 指定的。

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
spot.amazonaws.com/AWSServiceRoleForEC2Spot \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

竞价型实例限额

每个 Amazon Web Services 账户 可在每个区域运行和待处理的竞价型实例数量存在限额。待处理竞价型实例请求完成后，该请求将不再计入限额，因为正在运行的实例将计入限额。

竞价型实例限额根据您正在运行的竞价型实例为了完成待处理的竞价型实例请求而正在使用或将会使用的虚拟中央处理器（vCPU）数进行管理。如果您终止了竞价型实例但未取消竞价型实例请求，则这些请求将会计入您的竞价型实例 vCPU 限额，直到 Amazon EC2 检测到竞价型实例已经终止并关闭请求为止。

我们为竞价型实例提供以下限额类型。

名称	默认值	可调整
所有 DL 竞价型实例请求	0	是
所有 F 竞价型实例请求	0	是
所有 G 和 VT 竞价型实例请求	0	是
所有 Inf 竞价型实例请求	0	是
所有 P 竞价型实例请求	0	是
所有标准（A、C、D、H、I、M、R、T、Z）竞价型实例请求	5	是
所有 Trn 竞价型实例请求	0	是
所有 X 竞价型实例请求	0	是

尽管 Amazon EC2 会根据您的使用情况自动调整竞价型实例限额，您仍可以在必要时请求提高限额。例如，假设您计划启动超过当前限额允许数量的竞价型实例，则可以请求提高限额。如果您提交竞价型

实例请求并收到 `Max spot instance count exceeded` 错误，则也可以请求提高限额。要请求提高限额，请根据 Amazon EC2 Service Quotas 中的描述使用服务限额控制台。

您可以启动任意实例类型组合以满足您不断变化的应用程序需求。例如，对于 256 个 vCPU 的全标准竞价型实例请求限额，您可以请求 32 个 `m5.2xlarge` 竞价型实例（32 x 8 vCPU）或 16 个 `c5.4xlarge` 竞价型实例（16 x 16 vCPU）。

借助 Amazon CloudWatch 指标集成，您可以根据限额监控 EC2 使用情况。您还可以配置警报以在即将达到限额时发出警告。有关更多信息，请参阅《服务限额用户指南》中的 服务限额与 Amazon CloudWatch 警报。

Amazon EC2 专属主机

Amazon EC2 专属主机是一种专供您使用的物理服务器。

专属主机提供对实例置放的可见性和控制力，并且支持主机关联性。这意味着您可以在特定主机上启动和运行实例，并确保实例仅在特定主机上运行。有关更多信息，请参阅 Amazon EC2 专属主机自动置放和主机关联。

专属主机提供全面的自带许可(BYOL)支持。它们允许您使用按插槽、按内核或按虚拟机授权的现有软件许可证，包括 Windows Server、SQL Server、SUSE Linux Enterprise Server、Red Hat Enterprise Linux，或其他绑定到虚拟机、插槽或物理内核的软件许可证(取决于您的许可证条款)。

如果您需要在专用硬件上运行实例，但不需要查看或控制实例的置放，也不需要使用按插槽或按内核授权的软件许可证，则可以考虑改用专用实例。专用实例和专属主机均可用于在专用物理服务器上启动 Amazon EC2 实例。专用实例与专属主机上的实例在性能、安全性或物理特性方面没有区别。但是，它们之间也存在一些关键差异。下表重点介绍专用实例和专属主机之间的一些重要区别：

	Dedicated Host	Dedicated Instance
专用的物理服务器	实例容量完全供您专用的物理服务器。	专供单个客户账户使用的物理服务器。
实例容量共享	可以与其他账户共享实例容量。	不支持
计费	按主机计费	按实例计费
套接字、内核和主机 ID 的可见性	提供套接字数和物理内核数的可见性	无可见性

	Dedicated Host	Dedicated Instance
主机和实例关联	允许您在一段时间内将您的实例一致地部署到同一物理服务器	不支持
定向实例置放	提供额外可见性以及对在物理服务器上放置实例的方式的控制	不支持
自动实例恢复	支持。有关更多信息，请参阅Amazon EC2 专属主机恢复。	支持
自带许可 (BYOL)	支持	部分支持*
容量预留	不支持	支持

* 通过软件保障计划提供许可证移动性的 Microsoft SQL Server 和 Windows 虚拟桌面访问 (VDA) 许可证可用于专用实例。

有关专用实例的更多信息，请参阅 Amazon EC2 专用实例。

专属主机限制

在分配专属主机之前，请注意以下限制：

- 要在专属主机上运行 RHEL 和 SUSE Linux，您必须自带 AMI。由 Amazon 提供或 Amazon Web Services Marketplace 上提供的 RHEL 和 SUSE Linux AMI 不可用于专属主机。有关如何创建自己的 AMI 的更多信息，请参阅 自带软件许可证到 Amazon EC2 专属主机。

  此限制不适用于分配给高内存实例（`u-6tb1.metal`、`u-9tb1.metal`、`u-12tb1.metal`、`u-18tb1.metal` 和 `u-24tb1.metal`）的主机。由 Amazon 提供或在 Amazon Web Services Marketplace 上可用的 RHEL 和 SUSE Linux AMI 可以用于这些主机。
- 每个区域每个 Amazon 账户的每个实例系列运行的专属主机数量具有一定的限制。限额仅适用于正在运行的实例。如果实例处于待处理、正在停止或已停止状态，则不会计入限额。要查看账户限额或请求增加限额，请使用服务限额控制台。
- 在使用指定主机资源组的启动模板时，支持自动扩缩组。有关更多信息，请参阅Amazon EC2 Auto Scaling 用户指南中的使用高级设置创建启动模板 。

- 不支持 Amazon RDS 实例。
- Amazon免费使用套餐不适用于专属主机。
- 实例置放控制是指管理专属主机中的实例启动。不能在置放群组中启动专属主机。
- 如果为虚拟化实例类型分配主机，在主机分配完成后，您无法将该实例类型修改为 `.metal` 实例类型。例如，如果您为 `m5.large` 实例类型分配主机，则无法将实例类型修改为 `m5.metal`。

  如果为 `.metal` 实例类型分配主机，在主机分配完成后，您无法将该实例类型修改为虚拟化实例类型。例如，如果您为 `m5.metal` 实例类型分配主机，则无法将实例类型修改为 `m5.large`。

内容

- Amazon EC2 专属主机定价和账单
- Amazon EC2 专属主机实例容量配置
- Amazon EC2 专属主机上的可突发 T3 实例
- 自带软件许可证到 Amazon EC2 专属主机
- Amazon EC2 专属主机自动置放和主机关联
- 分配一台 Amazon EC2 专属主机供您的账户使用
- 在 Amazon EC2 专属主机上启动 Amazon EC2 实例
- 在主机资源组中启动 Amazon EC2 实例
- 修改现有 Amazon EC2 专属主机的自动置放设置
- 修改现有 Amazon EC2 专属主机支持的实例类型
- 修改 Amazon EC2 实例的 Amazon EC2 专属主机租赁和关联
- 释放 Amazon EC2 专属主机
- 迁移到基于 Nitro 的 Amazon EC2 专属主机
- 购买专属主机预留，享受专属主机账单折扣
- Amazon EC2 专属主机恢复
- Amazon EC2 专属主机的主机维护
- 监控 Amazon EC2 专属主机的状态
- 使用 Amazon Config 跟踪 Amazon EC2 专属主机配置更改

Amazon EC2 专属主机定价和账单

专属主机的价格因付款选项而异。有关更多信息，请参阅 Amazon EC2 专属主机定价。

付款选项

- 按需专属主机
- Dedicated Host Reservations

按需专属主机

按需计费在您将专属主机分配到您的账户时自动激活。

专属主机的按需价格因实例系列和区域而异。您需要为处于活动状态的专属主机按每秒支付费用（至少 60 秒），无论您选择在专属主机上启动的实例的数量或大小如何。

您可以随时释放按需专属主机以停止产生费用。有关释放专属主机的信息，请参阅释放 Amazon EC2 专属主机。

Dedicated Host Reservations

与运行按需专属主机相比，专属主机预留可提供账单折扣。预留提供三种付款选项：

- 无预付费用：无预付费用预留为某个期限内的专属主机使用提供折扣，并且不需要预付款。可以选择一年或三年期限。只有一些实例系列支持三年期无预付预留。
- 预付部分费用 — 必须支付一部分预留费用，期限内的剩余时间享受折扣。可以选择一年或三年期限。
- 预付全费 — 提供最低的有效价格。提供一年和三年期限，覆盖整个前期费用，无需额外将来付费。

账户中必须有活动的专属主机才能购买预留。每个预留可以涵盖在单个可用区中支持同一实例系列的一个或多个主机。预留应用于主机上的实例系列，而不是实例大小。如果有三个不同实例大小的专属主机（`m4.xlarge`、`m4.medium` 和 `m4.large`），则可以将一个 `m4` 预留与所有这些专属主机关联。预留的实例系列和可用区必须与您希望与之关联的专属主机的实例系列和可用区相匹配。

当预留与专属主机关联后，将无法释放专属主机，直到预留期限结束。

Amazon EC2 专属主机实例容量配置

专属主机支持不同的配置（物理内核、套接字和 VCPU），允许您运行不同系列和规模的实例。

当您在账户中分配专属主机时，您可以选择支持同一个实例系列中单个实例类型或多种实例类型的配置。可以在主机上运行的实例数量取决于您选择的配置。

内容

- 单个实例类型支持
- 多种实例类型支持

单个实例类型支持

您可以分配仅支持一种实例类型的专属主机。使用此配置，您在专属主机上启动的每个实例都必须具有相同的实例类型，这是您在分配主机时指定的实例类型。

例如，您可以分配仅支持 m5.4xlarge 实例类型的主机。在这种情况下，您只能运行该主机上的 m5.4xlarge 实例。

可以在主机上启动的实例数量取决于主机提供的物理内核数以及指定实例类型消耗的内核数。例如，如果您为 m5.4xlarge 实例分配了一台主机，该主机提供 48 个物理内核，每个 m5.4xlarge 实例使用 8 个物理内核。这意味着您最多可以在该主机上启动 6 个实例（48 个物理内核/每个实例 8 个内核 = 6 个实例）。

多种实例类型支持

您可以分配支持同一实例系列中多种实例类型的专属主机。这允许您在同一主机上运行不同的实例类型，只要这些实例属于同一个实例系列并且主机有足够的实例容量即可。

例如，您可以分配支持 R5 实例系列中不同实例类型的主机。在这种情况下，您可以在该主机的物理核心容量内启动 R5 实例类型的某些组合，例如 r5.large、r5.xlarge、r5.2xlarge 和 r5.4xlarge。

以下实例系列支持带有多种实例类型的专属主机：

- 通用型：A1 | M5 | M5n | M6i | M7i | T3
- 计算优化型：C5 | C5n | C6i | C7i
- 内存优化型：R5 | R5b | R5n | R6i | R7i | R7iz | X2iedn

您可以在主机上运行的实例数量取决于主机提供的物理内核数，以及在主机上运行的每种实例类型所使用的内核数。例如，如果您分配 R5 主机，它提供 48 个物理内核，那么您运行两个 r5.2xlarge 实例（4 个内核 x 2 个实例）和三个 r5.4xlarge 实例（8 个内核 x 3 个实例），这些实例总共消耗 32 个核心，并且您可能能够运行某些 R5 实例组合，只要它们在剩余的 16 个核心之内即可。

但是，对于每个实例系列，每种实例类型可以运行的实例数量都有限制。例如，R5 专属主机支持最多 2 个 r5.8xlarge 实例，其中使用 32 个物理内核。在这种情况下，可以使用类型更小的其他 R5 实例将主机填充到内核容量。有关每个实例系列支持的实例类型数量，请参阅专属主机配置表。

下表显示了实例类型示例组合：

实例系列	实例类型组合示例
R5	• 示例 1：4 x r5.4xlarge + 4 x r5.2xlarge • 示例 2：1 x r5.12xlarge + 1 x r5.4xlarge + 1 x r5.2xlarge + 5 x r5.xlarge + 2 x r5.large
C5	• 示例 1：1 x c5.9xlarge + 2 x c5.4xlarge + 1 x c5.xlarge • 示例 2：4 x c5.4xlarge + 1 x c5.xlarge + 2 x c5.large
M5	• 示例 1：4 x m5.4xlarge + 4 x m5.2xlarge • 示例 2：1 x m5.12xlarge + 1 x m5.4xlarge + 1 x m5.2xlarge + 5 x m5.xlarge + 2 x m5.large

注意事项

使用支持多种实例类型的专属主机时，请记住以下几点：

- 仅在同一实例系列中才可以在同一主机上使用多种实例类型。
- 混合使用实例类型时，为了最大限度地提高主机利用率，我们建议先启动较大的实例类型，然后再启动较小的实例类型。
  - 根据专属主机上实例类型的组合和启动顺序，可能无法在物理上最大限度地提高主机的利用率。在主机上混合使用实例类型时，某些容量可能在主机上提供但不可使用。例如，您可能会看到 r5n 主机上有 16 个可用的 vCPU，但可能无法在主机上启动 4xlarge 实例，即使 r5n.4xlarge 在 16 个 vCPU 上运行。

> **Note**
>
> 如果您为多种实例类型启用 A1 专属主机，则只能在该主机上混合启动 a1.xlarge 和 a1.2xlarge 实例。如果您在该主机上启动了一个 a1.medium 或 a1.large 实例，则只能在该主机上启动同一实例类型的其他实例。单个 a1.4xlarge 实例会消耗主机上的所有容量。如果您需要为 a1.medium 或 a1.large 实例配置一个主机，我们建议您分别为这些实例类型分配单独的主机。

Amazon EC2 专属主机上的可突发 T3 实例

专属主机支持突增性能 T3 实例。T3 实例提供了在专用硬件上使用符合条件的 BYOL 许可证软件的经济高效的方式。T3 实例较小的 vCPU 占用空间使您能够在更少的主机上整合工作负载，并最大限度地提高每核心许可利用率。

T3 专属主机最适合运行具有低到中等 CPU 利用率的 BYOL 软件。这包括符合条件的按插槽、按内核或按虚拟机授权的软件许可证，例如 Windows Server、Windows 桌面、SQL Server、SUSE Enterprise Linux Server、Red Hat Enterprise Linux 和 Oracle 数据库。适用 T3 专属主机的工作负载示例包括中小型数据库、虚拟桌面、开发和测试环境、代码存储库以及产品原型。不建议使用 T3 专属主机用于持续高 CPU 利用率的工作负载或同时出现相关 CPU 突发的工作负载。

专属主机上的 T3 实例使用与共享租赁硬件上的 T3 实例相同的信用模型。但是，它们支持 standard 信用模式；不支持 unlimited 积分模式。在 standard 模式下，专属主机上的 T3 实例赚取、消耗和累积服务抵扣金，其方式与共享租赁硬件上的突增实例相同。突增实例具有在基准 CPU 性能，并能够在需要时提升到基准水平之上。为了突增到基准以上，实例会花费在其 CPU 积分余额中累积的积分。当累积积分耗尽时，CPU 利用率将降低到基准水平。有关 standard 模式的更多信息，请参阅[标准可突增性能实例的工作原理]。

T3 专属主机支持 Amazon EC2 专属主机提供的所有功能，包括单个主机上的多个实例大小、主机 Resource Groups 和 BYOL。

支持的 T3 实例大小和配置

T3 专属主机通过提供基准 CPU 性能和在需要时突发到更高级别的能力来运行通用的可爆发 T3 实例，这些实例共享主机的 CPU 资源。这使得拥有 48 个核心的 T3 专属主机能够支持每台主机最多 192 个实例。为了有效利用主机的资源并提供最佳实例性能，Amazon EC2 实例放置算法会自动计算支持的可在主机上启动的实例数和实例大小组合。

T3 专属主机支持同一主机上的多个实例类型。专属主机上不是支持所有 T3 实例。您可以运行不同的 T3 实例组合，直到主机的 CPU 限制为止。

下表列出了支持的实例类型，总结了每种实例类型的性能，并指出了每种大小可以启动的最大实例数。

实例类型	vCPU	内存（GiB）	每个 vCPU 的基准 CPU 利用率	网络突增带宽 (Gbps)	Amazon EBS 突增带宽 (Mbps)	每个专属主机的最大实例数
t3.na	2	0.5	5%	5	最多 2,085	192
t3.m	2	1	10%	5	最多 2,085	192
t3.sn	2	2	20%	5	最多 2,085	192
t3.m	2	4	20%	5	最多 2,085	192
t3.la	2	8	30%	5	2,780	96
t3.xl	4	16	40%	5	2,780	48
t3.2x e	8	32	40%	5	2,780	24

监控 T3 专属主机的 CPU 利用率

您可以使用 DedicatedHostCPUUtilization Amazon CloudWatch 指标，用于监控专属主机的 vCPU 利用率。EC2 命名空间和 Per-Host-Metrics 维度中可用指标。有关更多信息，请参阅 专属主机指标。

自带软件许可证到 Amazon EC2 专属主机

专属主机允许使用现有的按插槽、按内核或按虚拟机软件的许可证。如果自带许可，您有责任管理自己的许可证。不过，Amazon EC2 具有一些帮助您保持许可证合规性的功能，例如实例关联和定向置放。

要将您自己的卷许可的计算机镜像引入到 Amazon EC2 中，需要执行以下常规步骤。

1. 验证控制您的系统映像使用的许可证条款是否允许在虚拟化云环境中使用系统映像。有关 Microsoft 许可的更多信息，请参阅亚马逊云科技和 Microsoft 许可。
2. 在确认可在 Amazon EC2 中使用系统映像后，使用 VM Import/Export 导入该映像。有关如何导入系统映像的信息，请参阅 VM Import/Export 用户指南。
3. 在导入系统映像后，可以在您的账户中的活动专属主机上从该映像中启动实例。
4. 在运行这些实例时，您可能需要针对自己的 KMS 服务器（例如，Windows Server 或 Windows SQL Server）激活这些实例，具体取决于操作系统。您无法针对 Amazon Windows KMS 服务器激活导入的 Windows AMI。

> **Note**
>
> 要跟踪您的映像在 Amazon 中的使用方式，请在 Amazon Config 中启用主机记录。您可以使用 Amazon Config 来记录专属主机的配置更改并将输出用作许可证报告的数据来源。有关更多信息，请参阅 使用 Amazon Config 跟踪 Amazon EC2 专属主机配置更改。

Amazon EC2 专属主机自动置放和主机关联

专属主机的置放控制是在实例级别和主机级别进行的。

自动置放

自动置放是在主机级别配置的。通过使用该功能，您可以管理启动的实例是在特定主机上启动，还是在具有匹配配置的任何可用主机上启动。

如果禁用专属主机的自动置放功能，则只会接受指定其唯一主机 ID 的主机租赁实例启动。这是新专属主机的默认设置。

如果启用专属主机的自动置放功能，则会接受任何与其实例类型配置相匹配的非定向主机租赁实例启动。

在启动实例时，您需要配置其租赁。如果在专属主机上启动实例而不提供特定 `HostId`，则将允许实例在任何已启用自动置放且匹配其实例类型的专属主机上启动。

主机关联

在实例级别配置主机关联。它在实例和专属主机之间建立启动关系。

当关联设置为 Host 时，启动到特定主机的实例在停止时始终在同一主机上重新启动。这适用于定向启动和非定向启动。

如果关联设置为 Default，并且您停止并重新启动实例，则实例可在任何可用主机上重新启动。但是，它将尝试在上次运行它的专属主机上启动 (尽最大努力)。

分配一台 Amazon EC2 专属主机供您的账户使用

要开始使用专属主机，必须先在您的账户中分配主机。在分配专属主机后，将在您的账户中立即提供专属主机容量，您可以开始在专属主机上启动实例。

当您在账户中分配专属主机时，您可以选择支持同一个实例系列中单个实例类型或多种实例类型的配置。您可以在主机上运行的实例数量取决于您选择的配置。有关更多信息，请参阅 Amazon EC2 专属主机实例容量配置。

Console

分配专属主机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 专属主机，然后选择 Allocate 专属主机 (分配专属主机)。
3. 对于 Instance family (实例系列)，为专属主机选择实例系列。
4. 指定 专属主机 是支持选定实例系列中的多种实例大小，还是仅支持特定的实例类型。请执行以下任一操作。

   - 要将专属主机配置为支持选定实例系列中的多种实例类型，请为 Support multiple instance types (支持多种实例类型) 选择 Enable (启用)。通过启用该选项，您可以在 专属主机 上启动同一实例系列中的不同实例大小。例如，如果您选择 m5 实例系列并选择该选项，则可以在专属主机上启动 m5.xlarge 和 m5.4xlarge 实例。
   - 要将专属主机配置为支持选定实例系列中的单个实例类型，请清除 Support multiple instance types (支持多种实例类型)，然后为 Instance type (实例类型) 选择要支持的实例类型。这样，您就可以在专属主机上启动单个实例类型。例如，如果选择该选项并将 m5.4xlarge 指定为支持的实例类型，则只能在专属主机上启动 m5.4xlarge 实例。
5. 对于 Availability Zone (可用区)，选择要在其中分配专属主机的可用区。
6. 要允许专属主机接受与其实例类型匹配的非定向实例启动，请为实例自动置放选择启用。有关自动置放的更多信息，请参阅Amazon EC2 专属主机自动置放和主机关联。
7. 要为专属主机启用主机恢复，请为 Host recovery (主机恢复) 选择 Enable (启用)。有关更多信息，请参阅Amazon EC2 专属主机恢复。

8. 对于 Quantity (数量)，输入要分配的专属主机数量。
9. （可选）选择 Add new tag (添加新标签)，然后输入标签键和标签值。
10. 选择 Allocate。

Amazon CLI

分配专属主机

使用 allocate-hosts 命令。以下示例会在 us-east-1a 可用区中，分配一个支持 m5 实例系列中多种实例类型的专属主机。此外还会启用主机恢复功能并禁用自动置放功能。

```
aws ec2 allocate-hosts \
    --instance-family "m5" \
    --availability-zone "us-east-1a" \
    --auto-placement "off" \
    --host-recovery "on" \
    --quantity 1
```

以下示例会在指定可用区中，分配一个支持非定向实例启动的专属主机，并启用主机恢复和自动置放功能。

```
aws ec2 allocate-hosts \
    --instance-type "m5.large" \
    --availability-zone "eu-west-1a" \
    --auto-placement "on" \
    --host-recovery "on" \
    --quantity 1
```

PowerShell

分配专属主机

使用 New-EC2Host cmdlet。以下示例会在 us-east-1a 可用区中，分配一个支持 m5 实例系列中多种实例类型的专属主机。该主机还会启用主机恢复功能，并禁用自动置放功能。

```
New-EC2Host `
    -InstanceFamily m5 `
    -AvailabilityZone us-east-1a `
    -AutoPlacement Off `
    -HostRecovery On `
```

```
    -Quantity 1
```

以下示例会在指定可用区中，分配一个支持非定向实例启动的专属主机，并启用主机恢复。

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement On `
    -HostRecovery On `
    -Quantity 1
```

在 Amazon EC2 专属主机上启动 Amazon EC2 实例

在分配一个专属主机后，您可以在其中启动实例。对于您启动的实例类型，如果没有具有足够可用容量的活动专属主机，则无法启动具有 host 租赁的实例。

注意事项

- SQL Server、SUSE 和 Amazon EC2 提供的 RHEL AMI 无法用于专属主机。
- 对于支持多种实例尺寸的专属主机，我们建议您首先启动较大的实例大小，然后根据需要用较小的实例大小填充剩余的实例容量。
- 在启动实例之前，请注意限制。有关更多信息，请参阅 专属主机限制。

Console

从专属主机页面中在特定专属主机上启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 在 Dedicated Hosts（专属主机）页面上，选择一个主机，然后依次选择 Actions（操作）、Launch Instance(s) onto host（在主机上启动实例）。
4. 在 Application and OS Images（应用程序和操作系统镜像）部分中，请从列表中选择一个 AMI。
5. 在 Instance type（实例类型）部分中，选择要启动的实例类型。

> **Note**
>
> 如果专属主机仅支持一种实例类型，则默认选择支持的实例类型，而无法进行更改。

如果专属主机支持多种实例类型，您必须根据专属主机的可用实例容量在支持的实例系列中选择一种实例类型。建议您首先启动较大的实例大小，然后根据需要用较小的实例大小填充剩余的实例容量。

6. 在 Key pair（密钥对）部分中，选择要与实例关联的密钥对。
7. 在高级详细信息部分中，对于租赁关联，执行下面的任意一项操作：
   - 关闭：禁用主机关联。实例在指定的主机上启动，但不保证停止后仍在同一专属主机上重新启动。
   - 专属主机 ID：启用主机关联。停止后，实例将始终在此指定主机上重新启动（如果主机有容量）。如果主机没有容量，则无法重新启动实例；您必须与其他主机建立关联。

   有关关联的更多信息，请参阅Amazon EC2 专属主机自动置放和主机关联。

   > **Note**
   > 租赁和主机选项是根据您选择的主机预配置的。

8. 根据需要配置其他实例选项。有关更多信息，请参阅 Amazon EC2 实例配置参数参考。
9. 选择启动实例。

使用启动实例向导在专属主机上启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，依次选择 Instances（实例）、Launch instance（启动实例）。
3. 在 Application and OS Images（应用程序和操作系统镜像）部分中，请从列表中选择一个 AMI。
4. 在 Instance type（实例类型）部分中，选择要启动的实例类型。
5. 在 Key pair（密钥对）部分中，选择要与实例关联的密钥对。
6. 在 Advanced details（高级详细信息）部分中，执行以下操作：
   a. 对于 Tenancy（租赁），选择 Dedicated Host（专属主机）。
   b. 对于 Target host by [目标主机（按）]，选择 Host ID（主机 ID）。
   c. 对于 Target host ID（目标主机 ID），选择要在其上启动实例的主机。
   d. 对于租赁关联，执行下面的任意一项操作：

- 关闭：禁用主机关联。实例在指定的主机上启动，但不保证停止后仍在同一专属主机上重新启动。
- 专属主机 ID：启用主机关联。停止后，实例将始终在此指定主机上重新启动（如果主机有容量）。如果主机没有容量，则无法重新启动实例；您必须与其他主机建立关联。

有关关联的更多信息，请参阅Amazon EC2 专属主机自动置放和主机关联。

7. 根据需要配置其他实例选项。有关更多信息，请参阅 Amazon EC2 实例配置参数参考。
8. 选择启动实例。

Amazon CLI

在专属主机上启动实例

使用 run-instances 命令并在 `--placement` 选项中指定实例亲和性、租赁和主机。

在具有主机关联的特定专属主机上启动（如果实例停止，它将始终在同一主机上重启）：

```
--placement Affinity=host,Tenancy=host,HostId=h-07879acf49EXAMPLE
```

在没有主机关联的特定专属主机上启动（实例可以在任何可用主机上重启）：

```
--placement Tenancy=host,HostId=h-07879acf49EXAMPLE
```

在启用了自动置放和具有匹配实例类型的任何可用专属主机上启动：

```
--placement Tenancy=host
```

PowerShell

在专属主机上启动实例

使用 New-EC2Host cmdlet 并在 `-Placement` 参数中指定实例亲和性、租赁和主机。

在具有主机关联的特定专属主机上启动（如果实例停止，它将始终在同一主机上重启）：

```
-Placement_Affinity host `
-Placement_Tenancy host `
-Placement_HostId h-07879acf49EXAMPLE
```

在没有主机关联的特定专属主机上启动（实例可以在任何可用主机上重启）：

```
-Placement_Tenancy host `
-Placement_HostId h-07879acf49EXAMPLE
```

在启用了自动置放和具有匹配实例类型的任何可用专属主机上启动：

```
-Placement_Tenancy host
```

在主机资源组中启动 Amazon EC2 实例

专属主机也与 Amazon License Manager 相集成。使用 License Manager，您可以创建主机资源组，该组是作为单个实体进行管理的专属主机的集合。创建主机资源组时，可以为专属主机指定主机管理首选项，如自动分配和自动释放。这允许您在专属主机上启动实例，而无需手动分配和管理这些主机。有关更多信息，请参阅 Amazon License Manager 用户指南中的主机 Resource Groups。

在包含具有可用实例容量的专属主机的主机资源组中启动实例时，Amazon EC2 将在该主机上启动实例。如果主机资源组没有包含具有可用实例容量的主机，Amazon EC2 将自动分配主机资源组中的新主机，然后在该主机上启动实例。有关更多信息，请参阅 Amazon License Manager 用户指南中的主机资源组。

要求和限制

- 您必须将基于核心或套接字的许可证配置与 AMI 关联。
- 您不能将 Amazon EC2 提供的 SQL Server、SUSE 或 RHEL AMI 用于专属主机。
- 您无法通过选择主机 ID 来定位特定主机，并且无法在主机资源组中启动实例时启用实例关联。

Console

在主机资源组中启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，依次选择 Instances（实例）、Launch instance（启动实例）。
3. 在 Application and OS Images（应用程序和操作系统镜像）部分中，请从列表中选择一个 AMI。
4. 在 Instance type（实例类型）部分中，选择要启动的实例类型。
5. 在 Key pair（密钥对）部分中，选择要与实例关联的密钥对。

6. 在 Advanced details（高级详细信息）部分中，执行以下操作：

   a. 对于 Tenancy（租赁），选择 Dedicated Host（专属主机）。

   b. 对于 Target host by [目标主机（按）]，选择 Host resource group（主机资源组）。

   c. 对于 Tenancy host resource group（租赁主机资源组），选择要在其中启动实例的主机资源组。

   d. 对于 Tenancy affinity（租赁关联），执行下面的任意一项操作：

      - 选择 Off（关闭）– 实例将在指定的主机上启动，但不保证停止后仍会在同一专属主机上重新启动。
      - 选择专属主机 ID – 如果停止，实例将始终在此特定主机上重新启动。

      有关关联的更多信息，请参阅Amazon EC2 专属主机自动置放和主机关联。

7. 根据需要配置其他实例选项。有关更多信息，请参阅 Amazon EC2 实例配置参数参考。

8. 选择启动实例。

Amazon CLI

在主机资源组中启动实例

可以使用 run-instances 命令。在 `--placement` 选项中，省略租赁并指定主机资源组的 ARN。

```
--placement HostResourceGroupArn=arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

PowerShell

在主机资源组中启动实例

使用 New-EC2Instance cmdlet。在 `-Placement` 参数中，省略租赁并指定主机资源组的 ARN。

```
-Placement_HostResourceGroupArn arn:aws:resource-groups:us-
east-2:123456789012:group/my-resource-group
```

修改现有 Amazon EC2 专属主机的自动置放设置

将专属主机分配到您的 Amazon 账户后，您可以修改其自动置放设置。

Console

修改专属主机的自动置放

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 选择一个主机，然后依次选择 Actions (操作)、Modify host (修改主机)。
4. 对于 Instance auto-placement (实例自动置放)，选择 Enable (启用) 来启用自动置放，或者取消选择 Enable (启用) 来禁用自动置放。有关更多信息，请参阅Amazon EC2 专属主机自动置放和主机关联。
5. 选择保存。

Amazon CLI

修改专属主机的自动置放

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --auto-placement on \
    --host-ids h-012a3456b7890cdef
```

PowerShell

修改专属主机的自动置放

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -AutoPlacement 1 `
    -HostId h-012a3456b7890cdef
```

修改现有 Amazon EC2 专属主机支持的实例类型

您可以修改专属主机以更改它支持的实例类型。如果它当前支持一种实例类型，您可以对其进行修改以支持该实例系列中的多种实例类型。类似地，如果它当前支持多种实例类型，您可以对其进行修改以仅支持特定的实例类型。

要修改专属主机以支持多种实例类型，您必须先停止主机上正在运行的所有实例。完成修改大约需要 10 分钟。在进行修改时，专属主机将转变为 pending 状态。在处于 pending 状态时，您无法在专属主机上启动停止的实例或启动新实例。

要将支持多种实例类型的专属主机修改为仅支持单个实例类型，主机不能具有运行中的实例，或者运行中的实例必须是您希望主机支持的实例类型。例如，要将支持 m5 实例系列中的多种实例类型的主机修改为仅支持 m5.large 实例，则专属主机不能具有正在运行的实例，或者只能在主机上运行 m5.large 实例。

如果为虚拟化实例类型分配主机，在主机分配完成后，您无法将该实例类型修改为 .metal 实例类型。例如，如果您为 m5.large 实例类型分配主机，则无法将实例类型修改为 m5.metal。如果为 .metal 实例类型分配主机，在主机分配完成后，您无法将该实例类型修改为虚拟化实例类型。例如，如果您为 m5.metal 实例类型分配主机，则无法将实例类型修改为 m5.large。

Console

修改专属主机的支持实例类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 专属主机。
3. 选择要修改的专属主机，然后依次选择 Actions (操作)、Modify host (修改主机)。
4. 根据专属主机的当前配置，执行以下某项操作：

   - 如果专属主机当前支持特定实例类型，则未启用 Support multiple instance types (支持多种实例类型)，并且 Instance type (实例类型) 会列出所支持的实例类型。要修改主机以支持当前实例系列中的多种类型，请为 Support multiple instance types (支持多种实例类型) 选择 Enable (启用)。

     您必须先停止主机上正在运行的所有实例，然后再修改主机以支持多种实例类型。

   - 如果专属主机当前支持实例系列中的多种实例类型，则为 Support multiple instance types (支持多种实例类型) 选择了 Enabled (已启用)。要修改主机以支持特定的实例类型，请为 Support multiple instance types (支持多种实例类型) 取消选择 Enable (启用)，然后为 Instance type (实例类型) 选择要支持的特定实例类型。

     您无法更改专属主机支持的实例系列。

5. 选择保存。

Amazon CLI

修改专属主机的支持实例类型

使用 modify-hosts 命令。

以下示例会修改专属主机以支持 m5 实例系列中的多种实例类型。

```
aws ec2 modify-hosts \
    --instance-family m5 \
    --host-ids h-012a3456b7890cdef
```

以下示例会修改专属主机以仅支持 m5.xlarge 实例。

```
aws ec2 modify-hosts \
    --instance-type m5.xlarge \
    --instance-family --host-ids h-012a3456b7890cdef
```

PowerShell

修改专属主机的支持实例类型

使用 Edit-EC2Host cmdlet。

以下示例会修改专属主机以支持 m5 实例系列中的多种实例类型。

```
Edit-EC2Host `
    -InstanceFamily m5 `
    -HostId h-012a3456b7890cdef
```

以下示例会修改专属主机以仅支持 m5.xlarge 实例。

```
Edit-EC2Host `
    -InstanceType m5.xlarge `
    -HostId h-012a3456b7890cdef
```

修改 Amazon EC2 实例的 Amazon EC2 专属主机租赁和关联

启动实例后，可以更改其租赁属性。您还可以修改实例的亲和性以定位特定主机，或者允许其在您账户中具有匹配属性的任何可用专属主机上启动。要修改实例租赁或关联，实例必须处于 stopped 状态。

您实例的操作系统详细信息以及是否安装了 SQL Server 会影响支持的转换。有关实例可用的租赁转换路径的更多信息，请参阅《License Manager User Guide》中的 Tenancy conversion。

> **Note**
>
> 对于 T3 实例，您必须在专属主机上启动实例才能使用 `host` 租赁。对于 T3 实例，您不能将租赁从 `host` 改为 `dedicated` 或 `default`。尝试进行这些不受支持的租赁更改会导致 `InvalidRequest` 错误代码。

Console

修改实例租赁或关联

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择实例并选择要修改的实例。
3. 依次选择实例状态、停止。
4. 在实例处于选中状态时，依次选择操作、实例设置、修改实例置放。
5. 在修改实例置放页面上，配置以下项：
   - 租赁 — 选择下列项之一：
     - 运行专用硬件实例 — 将实例作为专用实例启动。有关更多信息，请参阅Amazon EC2 专用实例。
     - 在专属主机上启动实例 — 在具有可配置关联的专属主机上启动实例。
   - 关联 — 选择下列项之一：
     - 此实例可以在任一主机上运行 — 实例在您的账户中支持该实例类型的任何可用专属主机上启动。
     - 此实例只能在选定的主机上运行 — 实例只能在为目标主机选择的专属主机上运行。
   - 目标主机 — 选择实例必须在其中运行的专属主机。如果未列出目标主机，则账户中可能没有可用的兼容专属主机。

   有关更多信息，请参阅Amazon EC2 专属主机自动置放和主机关联。
6. 选择保存。

Amazon CLI

修改实例租赁或关联

使用 modify-instance-placement 命令。以下示例将指定实例的关联从 `default` 更改为 `host`，并指定实例已关联到的专属主机。

```
aws ec2 modify-instance-placement \
    --instance-id i-1234567890abcdef0 \
    --affinity host \
    --tenancy host \
    --host-id h-012a3456b7890cdef
```

PowerShell

修改实例租赁或关联

使用 Edit-EC2InstancePlacement cmdlet。以下示例将指定实例的关联从 `default` 更改为 `host`，并指定实例已关联到的专属主机。

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
    -Affinity host `
    -Tenancy host `
    -HostId h-012a3456b7890cdef
```

释放 Amazon EC2 专属主机

如果您不再需要某个专属主机，则可以停止在该主机上运行的实例，指示实例在另一主机上启动，然后释放该主机。

必须先停止专属主机上运行的所有实例，然后才能释放主机。这些实例可以迁移至您账户的其他专属主机，这样您就可以继续使用它们。这些步骤只适用于按需专属主机。

Console

释放专属主机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。

3. 在专属主机页面上，选择要释放的专属主机。
4. 依次选择 Actions (操作)、Release host (释放主机)。
5. 选择 Release (释放) 确认。

Amazon CLI

释放专属主机

使用 release-hosts 命令。

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

释放专属主机

使用 Remove-EC2Host cmdlet。

```
Remove-EC2Host -HostId h-012a3456b7890cdef
```

在释放专属主机后，您无法再次重新使用同一主机或主机 ID，并将不再根据按需账单费率向您收费。专属主机的状态将变为 `released`，您将无法在该主机上启动任何实例。

> **Note**
>
> 如果您是刚刚释放的专属主机，则系统可能需要过一些时间才会停止将其计入限制。如果您在这段时间内尝试分配新的专属主机，可能会遇到 `LimitExceeded` 错误。如果出现这种情况，请在几分钟后再次尝试分配新的主机。

已停止的实例仍可以使用和列在 Instances 页面上。这些实例将保留其 `host` 租赁设置。

迁移到基于 Nitro 的 Amazon EC2 专属主机

Nitro 系统是由 Amazon 打造的硬件和软件组件集合，可实现高性能、高可用性和高安全性。与基于 Xen 的专属主机相比，基于 Nitro 的专属主机具有更高的性价比。如果您的账户中有任何基于 Xen 的专属主机，我们建议您将工作负载迁移到基于 Nitro 的专属主机。有关更多信息，请参阅 Amazon Nitro 系统。

要从基于 Xen 的专属主机迁移到基于 Nitro 的专属主机，需要将专属主机上基于 Xen 的实例迁移到基于 Nitro 的实例类型，分配新的基于 Nitro 的专属主机，然后将迁移的基于 Nitro 的实例移动到基于 Nitro 的新专属主机。

本主题提供了从基于 Xen 的专属主机迁移到基于 Nitro 的专属主机的详细步骤。

迁移步骤

- 步骤 1：确定基于 Xen 的专属主机
- 步骤 2：将基于 Xen 的实例迁移到基于 Nitro 的实例类型
- 步骤 3：分配基于 Nitro 的专属主机
- 步骤 4：将迁移的实例移动至新的基于 Nitro 的专属主机
- 步骤 5：释放未使用的基于 Xen 的专属主机

步骤 1：确定基于 Xen 的专属主机

以下专属主机基于 Xen，有资格迁移到基于 Nitro 的专属主机。

- 通用型：M3 | M4
- 计算优化型：C3 | C4
- 内存优化型：R3 | R4 | X1 | X1e
- 存储优化型：D2 | H1 | I2 | I3
- 加速计算型：G3 | P3

检查您的账户中是否有基于 Xen 的专属主机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 在搜索字段中，使用实例系列筛选条件搜索上述基于 Xen 的专属主机。例如，实例系列 = m3。

步骤 2：将基于 Xen 的实例迁移到基于 Nitro 的实例类型

在基于 Xen 的专属主机上运行的实例也基于 Xen。必须先将这些实例迁移到基于 Nitro 的实例类型，然后才能将其移动至基于 Nitro 的专属主机。

> **Important**
>
> 在开始迁移实例之前，我们建议您对数据进行备份。有关更多信息，请参阅 Create multi-volume Amazon EBS snapshots from an Amazon EC2 instance。

查找在基于 Xen 的专属主机上运行的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 选择要迁移的基于 Xen 的主机，然后选择正在运行的实例选项卡。该选项卡列出了在选定主机上运行的所有实例。

要迁移 Linux 实例，请参阅Amazon EC2 实例类型更改。

要迁移 Windows 实例，请参阅将 EC2 Windows 实例迁移到基于 Nitro 的实例类型。

> **Note**
>
> 确保将您的实例迁移到与您打算迁移到的基于 Nitro 的专属主机相匹配的实例类型。例如，如果您打算迁移到 M7i 专属主机，请确保将实例迁移到 M7i 实例类型。

步骤 3：分配基于 Nitro 的专属主机

查找支持的基于 Nitro 的专属主机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例类型。
3. 应用以下筛选条件：
   - 虚拟机监控程序 = nitro
   - 专属主机支持 = true

找到合适的基于 Nitro 的实例类型后，请分配新的专属主机。

步骤 4：将迁移的实例移动至新的基于 Nitro 的专属主机

分配基于 Nitro 的专属主机且主机状态变为 `available` 后，您可以将之前迁移到基于 Nitro 的实例类型的实例移动至新专属主机。

将您的实例移动至基于 Nitro 的新专属主机

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航面板中，选择实例。
3. 选择您迁移的实例，然后依次选择操作、实例设置、修改实例置放。
4. 对于目标专属主机，选择基于 Nitro 的新专属主机，然后选择保存。
5. 重新启动实例。选择实例，然后依次选择实例状态、开始实例。

步骤 5：释放未使用的基于 Xen 的专属主机

将工作负载从基于 Xen 的专属主机迁移到基于 Nitro 的新专属主机后，如果您不再需要它，则可以释放基于 Xen 的专属主机。

购买专属主机预留，享受专属主机账单折扣

与按需专属主机定价相比，专属主机预留可提供高达 70% 的折扣。您的账户中必须分配活动的专属主机，然后才能购买专属主机预留。有关更多信息，请参阅 Dedicated Host Reservations。

Console

购买预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 专属主机、专属主机预留 和 Purchase 专属主机预留 (购买专属主机预留)。
3. 在查找产品屏幕上，执行以下操作：
   a. 对于实例系列，选择要为其购买专属主机预留的专属主机实例系列。
   b. 对于付款选项，选择并配置您的首选付款选项。
4. 选择下一步。
5. 选择要与专属主机预留关联的专属主机，然后选择下一步。
6. （可选）为专属主机预留分配标签。
7. 审核您的订单，然后选择购买。

Amazon CLI

购买预留

1. 使用 describe-host-reservation-offerings 命令列出符合您需求的可用服务产品。以下示例列出了支持 m4 实例系列中的实例并具有一年期限的服务产品。

   期限以秒为单位指定。一年期限包括 31536000 秒，三年期限包括 94608000 秒。

   ```
   aws ec2 describe-host-reservation-offerings \
       --filter Name=instance-family,Values=m4 \
       --max-duration 31536000
   ```

   这条命令会返回符合您条件的服务产品的列表。记下要购买的服务或产品的 ID。

2. 使用 purchase-host-reservation 命令购买服务产品并提供上一步中提到的 offeringId。以下示例购买指定的预留，并将其与Amazon账户中已分配的特定专属主机关联，同时还应用具有键 purpose 和值 production 的标签。

   ```
   aws ec2 purchase-host-reservation \
       --offering-id hro-03f707bf363b6b324 \
       --host-id-set h-013abcd2a00cbd123 \
       --tag-specifications 'ResourceType=host-
   reservation,Tags={Key=purpose,Value=production}'
   ```

PowerShell

购买预留

1. 使用 Get-EC2HostReservationOffering cmdlet 列出符合需求的可用服务和产品。以下示例列出了支持 m5 实例系列中的实例并具有一年期限的产品。

   期限以秒为单位指定。一年期限包括 31536000 秒，三年期限包括 94608000 秒。

   ```
   $filter = @{Name="instance-family"; Values="m5"}
   Get-EC2HostReservationOffering `
       -Filter $filter `
       -MaxDuration 31536000
   ```

   这条命令会返回符合您条件的服务产品的列表。记下要购买的服务或产品的 ID。

2. 使用 New-EC2HostReservation cmdlet 购买服务和产品并提供上一步中记下的服务和产品 ID。以下示例购买指定的预留，并将其与Amazon账户中已分配的特定专属主机关联。

```
New-EC2HostReservation `
    -OfferingId hro-03f707bf363b6b324 `
    -HostIdSet h-013abcd2a00cbd123
```

Amazon EC2 专属主机恢复

如果在专属主机上检测到具有特定问题的条件，专属主机恢复自动在新的替换主机上重新启动实例。主机恢复减少了人工干预的需求，并降低了发生关于系统电源或网络连接的意外专属主机故障时的运营负担。其它专属主机问题需要进行手动干预才能从中恢复。

内容

- Amazon EC2 专属主机恢复功能的工作原理
- 支持的实例类型
- 定价
- 管理 Amazon EC2 专属主机恢复
- 查看 Amazon EC2 专属主机的主机恢复设置
- 手动恢复 Amazon EC2 专属主机恢复不支持的实例

Amazon EC2 专属主机恢复功能的工作原理

专属主机和主机资源组恢复程序使用主机级运行状况检查以评估专属主机可用性，以及检测基本系统故障。专属主机故障的类型决定了专属主机能否自动恢复。可能导致主机级运行状况检查失败的问题示例包括：

- 网络连接丢失
- 系统电源损耗
- 物理主机上的硬件或软件问题

> **Important**
>
> 主机计划停用后，专属主机将无法自动恢复。

专属主机自动恢复

在专属主机上检测到系统电源或网络连接故障时，将启动专属主机自动恢复，并且 Amazon EC2 自动在与原始专属主机相同的可用区中分配一个替换专属主机。替换专属主机收到新的主机 ID，但保留与原始专属主机相同的属性，包括：

- 可用区
- 实例类型
- 标签
- 自动置放设置
- 预留

在分配替换专属主机时，实例将恢复到替换专属主机上。恢复的实例保留与原始实例相同的属性，包括：

- 实例 ID
- 私有 IP 地址
- 弹性 IP 地址
- EBS 卷附加
- 所有实例元数据

此外，与 Amazon License Manager 的内置集成自动跟踪和管理您的许可证。

> **Note**
>
> 仅在提供了 Amazon License Manager 的区域中支持 Amazon License Manager 集成。

如果实例与受损专属主机之间具有主机关联关系，恢复的实例将与替换专属主机建立主机关联。

在所有实例已恢复到替换专属主机时，将释放受损专属主机并且替换专属主机变为可用。

在启动了主机恢复时，将通过电子邮件和 Amazon 事件通知 Amazon Health Dashboard 账户所有者。在成功完成主机恢复后，将发送第二个通知。

如果使用 Amazon License Manager 跟踪您的许可证，Amazon License Manager 根据许可证配置限制为替换专属主机分配新的许可证。如果由于主机恢复而违反许可证配置的硬限制，则不允许执行恢复

过程，并通过 Amazon SNS 通知向您通知主机恢复失败（如果已经为 Amazon License Manager 配置了通知设置）。如果由于主机恢复而违反许可证配置的软限制，则允许继续执行恢复，并通过 Amazon SNS 通知向您通知违反了限制。有关更多信息，请参阅 Amazon License Manager 用户指南中的使用许可证配置和 License Manager 中的设置。

主机恢复状态

当检测到专属主机故障时，受损专属主机将进入 `under-assessment` 状态，并且所有实例会进入 `impaired` 状态。在处于 `under-assessment` 状态时，您无法在受损的专属主机上启动实例。

在分配替换专属主机后，它进入 `pending` 状态。它保持该状态，直到主机恢复过程完成。在处于 `pending` 状态时，您无法在替换专属主机上启动实例。在恢复过程中，在替换专属主机上恢复的实例保持 `impaired` 状态。

在主机恢复完成后，替换专属主机进入 `available` 状态，并且恢复的实例恢复为 `running` 状态。在进入 `available` 状态后，您可以在替换专属主机上启动实例。将永久释放原始受损专属主机，并且它进入 `released-permanent-failure` 状态。

如果受损专属主机具有不支持主机恢复的实例（例如，具有实例存储根卷的实例），则不会释放该专属主机。相反，它标记为停用并进入 `permanent-failure` 状态。

专属主机无法自动恢复的场景

主机计划停用后，专属主机将无法自动恢复。您将在 Amazon Health Dashboard 中收到停用通知、Amazon CloudWatch 活动以及 Amazon 账户所有者的电子邮件地址会收到有关专属主机故障的消息。在指定的时间段内，按照停用通知中所述的纠正步骤手动恢复停用主机上的实例。

停止的实例不会恢复到替换专属主机上。如果您尝试启动将受损专属主机作为目标的停止实例，实例启动将失败。我们建议您修改停止的实例以将不同的专属主机作为目标，或者在任何可用的专属主机上启动并启用匹配的配置和自动置放。

具有实例存储的实例不会恢复到替换专属主机上。作为一项纠正措施，将受损专属主机标记为停用，并且您在主机恢复完成后收到停用通知。在指定的时间段内，按照停用通知中所述的纠正步骤手动恢复受损专属主机上的其余实例。

支持的实例类型

以下实例系列支持主机恢复：

- 通用型：A1 | M3 | M4 | M5 | M5n | M5zn | M6a | M6g | M6i | T3 | Mac1 | Mac2 | Mac2-m1ultra | Mac2-m2 | Mac2-m2pro
- 计算优化型：C3 | C4 | C5 | C5n | C6a | C6g | C6i

- 内存优化型：R3 | R4 | R5 | R5b | R5n | R6g | R6i | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | U-24tb1 | X1 | X1e | X2iezn
- 加速计算型：Inf1 | G3 | G5g | P3

要恢复不支持的实例，请参阅手动恢复 Amazon EC2 专属主机恢复不支持的实例。

> **Note**
>
> 与非裸机实例类型相比，受支持的裸机实例类型的专属主机的自动恢复将需要更长时间来进行检测和恢复。

定价

使用主机恢复不会收取额外的费用，但会收取正常的专属主机费用。有关更多信息，请参阅 Amazon EC2 专属主机定价。

一旦启动了主机恢复，将不再对受损专属主机计费。仅在进入 `available` 状态后，才会开始对替换专属主机计费。

如果使用按需费率对受损专属主机进行计费，还会使用按需费率对替换专属主机进行计费。如果受损专属主机具有有效专属主机预留，则会将其转移到替换专属主机。

管理 Amazon EC2 专属主机恢复

如果在专属主机上检测到具有特定问题的条件，专属主机恢复自动在新的替换主机上重新启动实例。您可以在分配专属主机时或分配后启用主机恢复。

分配主机时，请使用以下过程启用主机恢复。

Console

在分配时启用主机恢复

使用 Amazon EC2 控制台分配专属主机时，对于主机恢复，请选择启用。有关更多信息，请参阅分配一台 Amazon EC2 专属主机供您的账户使用。

Amazon CLI

在分配时启用主机恢复

使用 allocate-hosts 命令。

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

PowerShell

在分配时启用主机恢复

使用 New-EC2Host cmdlet。

```
New-EC2Host `
    -InstanceType m5.large `
    -AvailabilityZone eu-west-1a `
    -AutoPlacement on `
    -HostRecovery on `
    -Quantity 1
```

使用以下过程管理专属主机的主机恢复。

Console

在分配后管理主机恢复

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 选择专属主机。
4. 依次选择操作和修改主机。
5. 对于主机恢复，请选择或清除启用。
6. 选择保存。

Amazon CLI

在分配后启用主机恢复

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --host-recovery on \
    --host-ids h-012a3456b7890cdef
```

在分配后禁用主机恢复

使用 modify-hosts 命令并指定 host-recovery 参数的值为 off。

```
aws ec2 modify-hosts \
    --host-recovery off \
    --host-ids h-012a3456b7890cdef
```

PowerShell

在分配后启用主机恢复

使用 Edit-Host cmdlet。

```
Edit-EC2Host `
    -HostRecovery on `
    -HostId h-012a3456b7890cdef
```

在分配后禁用主机恢复

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -HostRecovery off `
    -HostId h-012a3456b7890cdef
```

查看 Amazon EC2 专属主机的主机恢复设置

您可以随时查看专属主机的主机恢复配置。

Console

查看专属主机的主机恢复配置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。

3. 选择专属主机，然后在描述选项卡中查看主机恢复字段。

Amazon CLI

查看专属主机的主机恢复配置

可以使用 describe-hosts 命令。

```
aws ec2 describe-hosts \
    --host-ids h-012a3456b7890cdef \
    --query Hosts[].HostRecovery
```

下面是示例输出。

```
on
```

PowerShell

查看专属主机的主机恢复配置

使用 Get-EC2Host cmdlet。

```
(Get-EC2Host -HostId h-012a3456b7890cdef).Hosts | Select HostRecovery
```

下面是示例输出。

```
HostRecovery
------------
on
```

手动恢复 Amazon EC2 专属主机恢复不支持的实例

主机恢复不支持恢复使用实例存储卷的实例。请按照以下说明手动恢复无法自动恢复的任何实例。

> **Warning**
>
> 在您停止、休眠或终止实例后，实例存储卷上的数据将会丢失。这包括附加到具有 EBS 根卷的实例的实例存储卷。要保护实例存储卷中的数据，请在停止或终止实例之前将其备份到持久性存储中。

手动恢复 EBS 支持的实例

对于无法自动恢复的 EBS 支持的实例，我们建议您手动停止并启动实例以将其恢复到新的专属主机上。有关停止实例以及在停止实例后对实例配置进行的更改的更多信息，请参阅启动和停止 Amazon EC2 实例。

手动恢复具有实例存储根卷的实例

有关无法自动恢复的具有实例存储根卷的实例，我们建议您执行以下操作：

1. 在新的专属主机上从最新的 AMI 中启动替换实例。
2. 将所需的所有数据迁移到替换实例中。
3. 终止受损专属主机上的原始实例。

Amazon EC2 专属主机的主机维护

使用主机维护功能时，如果出现专属主机降级的极少见情况，我们会自动将该主机上运行的受支持实例迁移到运行正常的替代专属主机上。这有助于尽可能减少工作负载的停机时间，并简化专属主机的管理。此外，还会针对计划内和例行的 Amazon EC2 维护，执行主机维护。

Amazon EC2 支持两种类型的主机维护：

- 实时迁移主机维护 – 实例将在 24 小时内自动迁移到替换主机，而无需停止和重新启动实例。
- 基于重启的主机维护 – 将实例列入实例重启计划事件，实例将在该计划事件期间自动停止并在替换主机上重新启动。

内容

- 主机维护与主机恢复
- 注意事项
- 相关服务
- 定价
- Amazon EC2 专属主机的主机维护工作原理
- 配置 Amazon EC2 专属主机的主机维护设置

主机维护与主机恢复

下表显示了主机恢复与主机维护之间的主要区别。

	主机恢复	主机维护
实例的可连接性	无法访问	可以访问
专属主机状态	`under-assessment`	`permanent-failure`
主机资源组	支持	不支持

有关主机恢复的更多信息，请参阅主机恢复。

注意事项

- 主机维护功能已在除中国区域和 Amazon GovCloud (US) Regions 的所有 Amazon Web Services 区域 开放。
- Amazon Outposts、Amazon Local Zone 和 Amazon Wavelength Zone 不支持主机维护。
- 无法为已经位于主机资源组中的主机开启或关闭主机维护。添加到主机资源组的主机保留其主机维护设置。有关更多信息，请参阅主机资源组。
- 以下实例类型具有实例存储根卷，因此不支持主机维护功能：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

相关服务

专属主机与Amazon License Manager 集成 – 跨 Amazon EC2 专属主机跟踪许可证（仅在提供 Amazon License Manager 的区域支持跟踪）。有关更多信息，请参阅 Amazon License Manager 用户指南。

Amazon Web Services 账户 必须具备足够许可证才能使用新专属主机。在计划事件完成后释放主机时，与性能降级主机关联的许可证也将被释放。

定价

使用主机维护不会产生额外费用，但会收取正常的专属主机费用。有关更多信息，请参阅 Amazon EC2 专属主机定价。

启动主机维护后，不再对降级专属主机计费。仅在进入 `available` 状态后，才会开始对替换专属主机计费。

如果降级的专属主机使用按需费率计费，则替换的专属主机也将用按需费率计费。如果降级专属主机具有有效专属主机预留，则会将其转移到新专属主机。

Amazon EC2 专属主机的主机维护工作原理

检测到启用了主机维护功能的专属主机性能降级后，我们会自动在您的账户中分配一个替换专属主机。替换专属主机收到新的主机 ID，但保留与原始专属主机相同的属性，包括：

- 自动置放设置
- 可用区
- 专属主机预留关联
- 主机关联
- 主机维护设置
- 主机恢复设置
- 实例类型
- 标签

分配替换主机后，我们会使用实时迁移主机维护或基于重启的主机维护来迁移实例，具体取决于所涉及的实例。

性能降级的主机不再有正在运行的实例后，将会从您的账户中永久释放。

实时迁移主机维护

需要实时迁移主机维护的实例将在 24 小时内自动迁移到替换主机，而无需停止和重新启动实例。迁移后的实例会保留其现有属性，包括：

- 实例 ID
- 实例元数据
- Amazon EBS 卷附加
- 弹性 IP 地址和私有 IP 地址
- 内存、CPU 和联网状态

在迁移过程中，一些较大的实例型号可能会出现性能轻微下降的情况。

在实例自动迁移到替换主机后，我们会向您发送电子邮件通知和 Amazon Health 控制面板通知，其中包括性能降级主机和替换主机的 ID、有关使用实时迁移主机维护自动迁移的实例的信息以及有关其余实例的信息。

基于重启的主机维护

对于需要重启以进行主机维护的实例，我们会将其列入自通知之日起 14 天内的实例重启计划事件。在计划事件开始之前，您可以继续访问性能降级的专属主机上的实例。

您可以将重启事件调整为距离原始事件日期和时间 7 天以内的任何日期。有关更多信息，请参阅 重新计划 EC2 实例的计划事件。

Amazon EC2 会自动在替换主机上为这些实例预留容量。您无法利用此预留容量运行实例。

Amazon EC2 控制台会将预留容量显示为已用容量。可能会出现实例同时在降级主机和替换主机上运行的情况。但是，这些实例在降级主机上继续运行的时间仅截止到它们停止运行，或迁移到替换主机的预留容量中。

当计划事件的日期和时间到来时，实例将会自动停止并利用替换主机上的预留容量重新启动。迁移后的实例会保留其现有属性，包括：

- 实例 ID
- 实例元数据
- Amazon EBS 卷附加
- 弹性 IP 地址和私有 IP 地址

但是，由于迁移期间会停止并重新启动实例，因此不会保留实例的内存、CPU 和联网状态。

您也可以在计划事件开始之前随时手动停止并重启这些实例，以将其迁移到替换主机或其他主机。您可能需要修改实例的主机关联，才能在其他主机上重启实例。如果您在计划事件开始前停止了实例，则替换主机上的预留容量将被释放并可供使用。

主机维护状态

当主机性能降级时，将会进入 `permanent-failure` 状态。您无法在处于 `permanent-failure` 状态的专属主机上启动实例。

分配替换主机后，该主机将一直保持在 `pending` 状态，直到将支持实时迁移主机维护的实例自动从性能降级的主机迁移，并为其余实例计划了计划事件时为止。完成这些操作后，替换主机将进入 `available` 状态。

替换主机进入 `available` 状态后，您可以按照与账户中任何主机相同的方式来使用该主机。但是，替换主机上的某些实例容量是为需要重启以进行主机迁移的实例预留的。您无法利用这些预留容量启动新实例。

性能降级的主机不再有正在运行的实例后，将会进入 `released, permanent-failure` 状态，并且会从您的账户中永久释放。请注意，该主机及其资源在短时间内仍会在控制台上可见。

自动迁移

某些实例无法自动迁移到替换主机。

采用 EBS 支持的根卷的实例

对于此类实例，我们会将其列入自通知之日起 28 天内的实例停止事件。当计划事件的日期和时间到来时，实例将被停止。我们建议您在替换主机或其他主机上手动停止并重启实例。您可能需要修改实例的主机关联，才能在其他主机上重启实例。

具有实例存储根卷的实例

对于此类实例，我们会将其列入自通知之日起 28 天内的实例停用事件。当计划事件的日期和时间到来时，实例将被永久终止。我们建议您在替换主机上手动启动替换实例，然后在计划事件开始之前将所需的数据迁移到替换实例。

以下实例具有实例存储根卷：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

在计划事件开始之前，您可以继续访问性能降级的专属主机上的实例。

配置 Amazon EC2 专属主机的主机维护设置

启用主机维护，确保在计划维护事件期间，在专属主机上运行的实例自动恢复到新的专属主机上。

如果禁用主机维护，您会收到一封电子邮件通知，要求在 28 天内驱逐降级主机并将实例手动迁移到另一台主机。如果您有专属主机预留，则会分配替换主机。28 天后，降级主机上运行的实例将终止，主机会自动释放。

Console

为专属主机启用主机维护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 依次选择专属主机 > 操作 > 修改主机。
4. 在主机维护字段中选择启用。

为专属主机禁用主机维护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 依次选择专属主机 > 操作 > 修改主机。
4. 在主机维护字段中选择关闭。

Amazon CLI

为专属主机启用主机维护

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --host-maintenance on \
    --host-ids h-0d123456bbf78910d
```

为专属主机禁用主机维护

使用 modify-hosts 命令。

```
aws ec2 modify-hosts \
    --host-maintenance off \
    --host-ids h-0d123456bbf78910d
```

PowerShell

为专属主机启用主机维护

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
    -HostMaintenance on `
    -HostId h-0d123456bbf78910d
```

为专属主机禁用主机维护

使用 Edit-EC2Host cmdlet。

```
Edit-EC2Host `
```

```
    -HostMaintenance off `
    -HostId h-0d123456bbf78910d
```

监控 Amazon EC2 专属主机的状态

Amazon EC2 持续监控专属主机的状态。将在 Amazon EC2 控制台上显示更新的状态。您可以使用以下方法查看有关专属主机的信息。

Console

查看专属主机的状态

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择专属主机。
3. 在列表中找到专属主机，并查看状态列中的值。

Amazon CLI

查看专属主机的状态

可以使用 describe-hosts 命令。

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

查看专属主机的状态

使用 Get-EC2Host cmdlet。

```
Get-EC2Host -HostId h-012a3456b7890cdef
```

下表说明了可能的专属主机状态。

State	描述
available	Amazon未检测到专属主机的问题。不会安排维护或修复。实例可在此专属主机上启动。

State	描述
released	已释放专属主机。主机 ID 不再使用。无法重新使用已释放的主机。
under-assessment	Amazon正在寻找专属主机可能存在的问题。如果必须采取措施，系统将通过 Amazon Web Services 管理控制台或电子邮件通知您。无法在处于该状态的专属主机上启动实例。
pending	专属主机不能用于启动新的实例。正在对其进行修改以支持多种实例类型，或者正在进行主机恢复。
permanent-failure	检测到了一个不可恢复的故障。您将通过您的实例和通过电子邮件接收到一个移出通知。实例可能会继续运行。如果在处于此状态的专属主机上停止或终止所有实例，Amazon将停用该主机。Amazon不会在此状态下重新启动实例。无法在处于该状态的专属主机上启动实例。
released-permanent-failure	Amazon永久释放已发生故障的专属主机，不再在这些主机上运行实例。专属主机 ID 不再可供使用。

使用 Amazon Config 跟踪 Amazon EC2 专属主机配置更改

您可以使用 Amazon Config 记录专属主机的配置更改，以及记录在主机上启动、停止或终止的实例的配置更改。然后，您可以将由 Amazon Config 捕获的信息用作许可证报告的数据来源。

Amazon Config 分别记录专属主机和实例的配置信息，并通过关系将这些信息配对。具有三种报告条件：

- Amazon Config 记录状态 – 当其状态为开启时，Amazon Config 将记录一个或多个 Amazon 资源类型，其中可包含专属主机和专用实例。要捕获许可证报告所需的信息，请使用以下字段验证是否记录了主机和实例。
- 主机记录状态 — 当其状态为启用时，将记录专属主机的配置信息。
- 实例记录状态 — 当其状态为启用时，将记录专用实例的配置信息。

如果禁用了这三个条件中的任一个，则 Edit Config Recording 按钮中的图标为红色。要发挥此工具的所有优点，请确保这三种记录方法都已启用。当这三种方法全部启用时，图标为绿色。要编辑设置，请选择 Edit Config Recording。您将被定向到 Amazon Config 控制台中的 Set up Amazon Config 页面，

在该页面中，您可以设置 Amazon Config 并启动对您的主机、实例和其他支持的资源类型的记录。有关更多信息，请参阅 Amazon Config 开发人员指南 中的使用控制台设置 Amazon Config。

> **Note**
>
> Amazon Config 将在发现您的资源后记录它们，此过程可能需要几分钟。

在 Amazon Config 开始记录对您的主机和实例的配置更改后，您可以获取已分配或已释放的任何主机以及已启动、已停止或已终止的任何实例的配置历史记录。例如，在专属主机的配置历史记录中的任何时间点上，您均可以查看在该主机上启动的实例的数量以及该主机上的套接字和内核的数量。对于其中的任何实例，您还可以查找其亚马逊机器映像（AMI）的 ID。您可以使用此信息来报告您拥有的服务器端绑定软件 (按插槽或按内核授予许可) 的许可。

您可以使用以下任一方式查看配置历史记录：

- 通过使用 Amazon Config 控制台。对于每个已记录的资源，您可以查看一个时间线页面，该页面提供了配置详细信息的历史记录。要查看此页面，请选择专属主机页面的配置时间线列中的灰色图标。有关更多信息，请参阅 Amazon Config 开发人员指南 中的在 Amazon Config 控制台中查看配置详细信息。
- 通过运行 Amazon CLI 命令。首先，您可以使用 list-discovered-resources 命令获取一个包含所有主机和实例的列表。然后，您可以使用 get-resource-config-history 命令获取特定时间间隔内某个主机或实例的配置详细信息。
- 通过在您的应用程序中使用 Amazon Config API。首先，您可以使用 ListDiscoveredResources 操作获取一个包含所有主机和实例的列表。然后，您可以使用 GetResourceConfigHistory 操作获取特定时间间隔内某个主机或实例的配置详细信息。

例如，要从 Amazon Config 中获取所有专属主机的列表，请运行 CLI 命令，如下所示。

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Host
```

要从 Amazon Config 中获取专属主机的配置历史记录，请运行 CLI 命令，如下所示。

```
aws configservice get-resource-config-history \
    --resource-type AWS::EC2::Instance \
    --resource-id i-1234567890abcdef0
```

使用控制台管理 Amazon Config 设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在专属主机页面上，选择编辑配置记录。
3. 在 Amazon Config 控制台中，按照提供的步骤来启用记录。有关更多信息，请参阅使用控制台设置 Amazon Config。

有关更多信息，请参阅在 Amazon Config 控制台中查看配置详细信息。

使用命令行或 API 激活 Amazon Config

- Amazon CLI：Amazon CLI 开发人员指南 中的查看配置详细信息 (Amazon Config)。
- Amazon EC2 API：GetResourceConfigHistory。

Amazon EC2 专用实例

默认情况下，EC2 实例将在共享租赁硬件上运行。这表示多个 Amazon 账户可能会共享相同物理硬件。

专用实例是在专用于单个 Amazon 账户的硬件上运行的 EC2 实例。这就表示，专用实例在主机硬件层面与属于其他 Amazon Web Services 账户 的实例存在物理隔离(即使这些账户都与单个付款人账户相关联也是如此)。不过，专用实例可能与来自同一 Amazon Web Services 账户但非专用实例的其他实例共享硬件。

专用实例不提供对实例置放的可见性和控制力，也不支持主机关联性。停止专用实例后再启动，它可能不会在同一台主机上运行。同样，您也不能指定特定主机启动或运行该实例。此外，专用实例对自带许可(BYOL)的支持有限。

如果您需要实例置放方面的可见性和控制力，以及更全面的 BYOL 支持，请考虑改用专属主机。专用实例和专属主机均可用于在专用物理服务器上启动 Amazon EC2 实例。专用实例与专属主机上的实例在性能、安全性或物理特性方面没有区别。但是，它们之间也存在一些关键差异。下表重点介绍专用实例和专用主机之间的一些重要区别：

	Dedicated Host	Dedicated Instance
专用的物理服务器	实例容量完全供您专用的物理服务器。	专供单个客户账户使用的物理服务器。

	Dedicated Host	Dedicated Instance
实例容量共享	可以与其他账户共享实例容量。	不支持
计费	按主机计费	按实例计费
套接字、内核和主机 ID 的可见性	提供套接字数和物理内核数的可见性	无可见性
主机和实例关联	允许您在一段时间内将您的实例一致地部署到同一物理服务器	不支持
定向实例置放	提供额外可见性以及对在物理服务器上放置实例的方式的控制	不支持
自动实例恢复	支持。有关更多信息，请参阅Amazon EC2 专属主机恢复。	支持
自带许可 (BYOL)	支持	部分支持*
容量预留	不支持	支持

* 通过软件保障计划提供许可证移动性的 Microsoft SQL Server 和 Windows 虚拟桌面访问 (VDA) 许可证可用于专用实例。

有关更多信息，请参阅 Amazon EC2 专属主机。

内容

- 专用实例 基础知识
- 支持的特征
- 专用实例 限制
- 专用实例定价
- 在具有默认租赁的 VPC 中启动专用实例
- 更改 EC2 实例的租赁
- 更改 VPC 的实例租赁

专用实例 基础知识

VPC 的租赁可以为 default 或 dedicated。默认情况下，VPC 的租赁将为 default，并且在 default 租赁 VPC 中启动的实例租赁也为 default。要启动专用实例，请执行以下操作：

- 创建一个租赁为 dedicated 的 VPC，从而确保该 VPC 中的所有实例都将作为专用实例运行。有关更多信息，请参阅 在具有默认租赁的 VPC 中启动专用实例。
- 创建一个租赁为 default 的 VPC，然后为要作为专用实例运行的实例手动指定租赁 dedicated。有关更多信息，请参阅 在具有默认租赁的 VPC 中启动专用实例。

支持的特征

专用实例支持以下功能和 Amazon 服务集成：

特征

- 预留实例
- 自动扩缩
- 自动恢复
- 专用竞价型实例
- 具爆发能力的实例

预留实例

要为您的专用实例预留容量，可以购买专用预留实例或容量预留。有关更多信息，请参阅Amazon EC2 的预留实例概览和使用 EC2 按需容量预留来预留计算容量。

如果购买专用预留实例，则可以按极其优惠的使用费率购买容量来启动专用实例；而这种使用费率价格优惠仅在您使用专用租赁启动实例时才有效。当您购买具有原定设置租赁的预留实例时，它仅适用于具有 default 租赁的运行实例；它不适用于具有 dedicated 租赁的运行实例。

您在购买Reserved Instance之后将无法使用修改过程来更改其租赁。但是，您可以将可转换预留实例换成具有不同租赁的新可转换预留实例。

自动扩缩

您可以使用 Amazon EC2 Auto Scaling 启动专用实例。有关更多信息，请参阅Amazon EC2 Auto Scaling 用户指南中的使用高级设置创建启动模板 。

自动恢复

如果专用实例因需要 Amazon 参与才能修复的基础硬件故障或问题而受损，您可以为它配置自动恢复。有关更多信息，请参阅 实例自动恢复。

专用竞价型实例

创建 Spot 实例请求时，您可以通过指定租赁 `dedicated` 来运行专用 Spot 实例。有关更多信息，请参阅 在单租户硬件上启动。

具爆发能力的实例

您可以通过the section called “可突增性能实例”，利用在专用租赁硬件上运行的优势。默认情况下，T3 专用实例以无限模式启动，提供了基准水平的 CPU 性能，并且在工作负载需要时能够突增到更高的 CPU 性能水平。T3 基准性能和突增能力由 CPU 积分控制。由于 T3 实例类型的可突增性质，我们建议您监控 T3 实例如何使用专用硬件的 CPU 资源，以获得最佳性能。T3 专用实例面向具有多种工作负载的客户，这些工作负载表现出随机 CPU 行为，但理想情况下，平均 CPU 使用率等于或低于基准使用率。有关更多信息，请参阅the section called “重要概念”。

Amazon EC2 具有识别和纠正性能可变性的系统。但是，如果您启动多个具有关联 CPU 使用模式的 T3 专用实例，仍可能遇到短暂的变化。对于这些要求更高或关联的工作负载，我们建议使用 M5 或 M5a 专用实例，而不是 T3 专用实例。

专用实例 限制

使用专用实例时请记住以下事项：

- 某些 Amazon 服务或其功能不受实例租期设置为 `dedicated` 的 VPC 支持。请参阅各自的服务文档以确认是否存在任何限制。
- 某些实例类型无法启动至实例租期设置为 `dedicated` 的 VPC 中。有关支持的实例类型的更多信息，请参阅 Amazon EC2 专用实例。
- 当启动 Amazon EBS 支持的专用实例时，EBS 卷不会在单一租户硬件上运行。

专用实例定价

专用实例的定价不同于按需实例的定价。有关更多信息，请参阅 Amazon EC2 专用实例。

在具有默认租赁的 VPC 中启动专用实例

当您创建 VPC 时，您可以选择指定它的实例租期。如果在实例租赁为 `dedicated` 的 VPC 中启动实例，则该实例将作为专用实例在专用硬件上运行。

有关启动租期为 host 的实例的更多信息，请参阅在 Amazon EC2 专属主机上启动 Amazon EC2 实例。

有关 VPC 租赁选项的更多信息，请参阅《Amazon VPC 用户指南》中的创建 VPC。

要求

- 选择一个受支持的实例类型。有关更多信息，请参阅 Amazon EC2 专用实例。

Console

在默认租赁 VPC 中启动专用实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，依次选择 Instances（实例）、Launch instance（启动实例）。
3. 在 Application and OS Images（应用程序和操作系统镜像）部分中，请从列表中选择一个 AMI。
4. 在 Instance type（实例类型）部分中，选择要启动的实例类型。
5. 在 Key pair（密钥对）部分中，选择要与实例关联的密钥对。
6. 在 Advanced details（高级详细信息）部分中，对于 Tenancy（租赁），选择 Dedicated（专属）。
7. 根据需要配置其他实例选项。有关更多信息，请参阅 Amazon EC2 实例配置参数参考。
8. 选择启动实例。

Amazon CLI

在启动过程中设置实例的租赁选项

使用 run-instances 命令并包含 Tenancy 及 --placement 选项。

```
--placement Tenancy=dedicated
```

PowerShell

在启动过程中设置实例的租赁选项

将 New-EC2Instance cmdlet 与 -Placement_Tenancy 参数结合使用。

```
-Placement_Tenancy dedicated
```

更改 EC2 实例的租赁

启动实例后，可以更改已停止实例的租赁属性。您所做的更改将在下次实例开启时生效。

或者，您可以更改虚拟私有云（VPC）的租赁。有关更多信息，请参阅 the section called “更改 VPC 的租期”。

限制

- 不能使用 Amazon Web Services 管理控制台更改实例的租赁。
- 该实例必须处于 `stopped` 状态。
- 您实例的操作系统详细信息以及是否安装了 SQL Server 会影响支持的转换。有关实例可用的租赁转换路径的更多信息，请参阅《License Manager User Guide》中的 Tenancy conversion。
- 对于 T3 实例，您必须在专属主机上启动实例才能使用 `host` 租赁。您不能将租赁从 `host` 更改 `dedicated` 或 `default`。尝试进行这些不受支持的租赁更改会导致 `InvalidRequest` 错误代码。

Amazon CLI

修改实例的租赁值

使用 modify-instance-placement 命令。

```
aws ec2 modify-instance-placement \
    --instance-id i-1234567890abcdef0 \
    --tenancy dedicated
```

PowerShell

修改实例的租赁值

使用 Edit-EC2InstancePlacement cmdlet。

```
Edit-EC2InstancePlacement `
    -InstanceId i-1234567890abcdef0 `
```

```
    -Tenancy Dedicated
```

更改 VPC 的实例租赁

在创建虚拟私有云（VPC）之后，可以将其实例租赁从 dedicated 改为 default。修改 VPC 的实例租赁不会影响 VPC 中任何现有实例的租赁。下次在 VPC 中启动一个实例时，该实例将具有 default 租赁，除非您在实例启动过程中另有指定。

或者，您可以更改特定实例的租赁。有关更多信息，请参阅 the section called “更改实例的租期”。

限制

- 在创建 VPC 之后，您无法将其实例租赁从 default 更改为 dedicated。
- 不能使用 Amazon Web Services 管理控制台更改 VPC 的实例租赁。

Amazon CLI

修改 VPC 的实例租赁属性

使用 modify-vpc-tenancy 命令。唯一受支持的租赁值为 default。

```
aws ec2 modify-vpc-tenancy \
    --vpc-id vpc-1234567890abcdef0 \
    --instance-tenancy default
```

PowerShell

修改 VPC 的实例租赁属性

使用 Edit-EC2VpcTenancy cmdlet。唯一受支持的租赁值为 Default。

```
Edit-EC2VpcTenancy `
    -VpcId vpc-1234567890abcdef0 `
    -InstanceTenancy Default
```

按需容量预留和适用于机器学习的容量块

通过使用容量预留，您可以为特定可用区中的 Amazon EC2 实例预留计算容量。对于不同的使用案例，有两种类型的容量预留。

容量预留类型

- 按需容量预留
- 适用于 ML 的容量块

以下是按需容量预留的一些常见使用案例：

- 扩展事件 - 您可以在业务关键型事件之前创建按需容量预留，以确保在需要时进行扩展。
- 监管要求和灾难恢复 - 使用按需容量预留来满足高可用性的监管要求，并在不同的可用区或区域中预留容量以进行灾难恢复。
- 共享未使用的容量：使用可中断容量预留将未使用的容量用于您的账户中的其他工作负载，同时保持控制权，以便在需要时回收这些容量。

以下是适用于 ML 的容量块的一些常见使用案例：

- 机器学习（ML）模型训练和微调 - 不间断地访问您为完成 ML 模型训练和微调而预留的 GPU 实例。
- ML 实验和原型 - 运行实验并构建需要短期 GPU 实例的原型。

何时使用按需容量预留

如果您有严格的容量要求，并且当前或未来的关键业务工作负载需要容量保证，请使用按需容量预留。通过按需容量预留，您可以确保只要您需要，就随时可以访问预留的 Amazon EC2 容量。

何时使用适用于 ML 的容量块

如果您需要确保从未来某个日期开始的一段规定的时间内可以不间断地访问 GPU 实例，请使用适用于 ML 的容量块。容量块非常适合于训练和微调 ML 模型、短期实验运行以及未来处理推理需求的临时激增。使用容量块，您可以确保在特定日期访问 GPU 资源，以运行您的 ML 工作负载。

使用 EC2 按需容量预留来预留计算容量

通过使用 Amazon EC2 容量预留，您可以在特定可用区中为 Amazon EC2 实例预留计算容量达任意持续时间。如果您对当前或未来的关键业务工作负载有严格的容量要求，需要一定程度的长期或短期容量保证，我们建议您创建容量预留，以帮助确保在需要时始终可以获得所需时长的 Amazon EC2 容量。

您能够随时创建容量预留，并且可以选择何时启动。您可以请求立即使用的容量预留，也可以请求未来某个日期的容量预留。

- 如果您请求立即使用的容量预留，则容量预留将立即可用，并且没有期限承诺。您可以随时修改容量预留，也可以随时取消，进而释放预留容量并停止产生费用。
- 如果您请求未来日期的容量预留，则需要指定需要容量预留可供使用的未来日期。您还必须指定一个承诺期限，承诺在指定日期之后在您的账户中保留所请求的容量。在请求的日期和时间，容量预留可供使用，承诺期限开始。在承诺期限内，您不能将实例数量或承诺期限减少到初始承诺以下，也不能取消容量预留。承诺期限过后，您可以以任何方式修改容量预留，或者如果不再需要容量预留，也可以取消它。

容量预留只能由匹配其属性的实例使用。默认情况下，容量预留自动将新实例与具有匹配属性（实例类型、平台、可用区和租赁）的运行中实例进行匹配。这意味着，任何具有匹配属性的实例都将自动在容量预留中运行。不过，您还可以将容量预留定位到特定工作负载。这可以明确控制允许哪些实例在预留容量中运行。您还可指定实例仅在容量预留或容量预留资源组中运行。

> **Important**
>
> 未来日期的容量预留用于帮助您启动和涵盖增量实例，而不是涵盖现有正在运行的实例。如果需要涵盖现有正在运行的实例，请改用立即启动的容量预留。

具有匹配属性（即实例类型、平台、可用区和租赁）的所有受支持的 Amazon EC2 实例都有资格在该容量预留中运行。Amazon EC2 实例可以由您启动（非托管式实例），也可以由 Amazon 服务代表您启动（托管式实例）。开放的容量预留尤其如此，因为系统会自动与任何具有匹配属性的正在运行的实例匹配。例如，以下服务代表您启动的托管式实例有资格在您创建和管理的容量预留中运行。

- Amazon EC2 Auto Scaling
- Amazon ECS
- Amazon EKS
- Amazon EMR
- Amazon SageMaker AI
- Amazon Batch
- Amazon Elastic Beanstalk
- Amazon ParallelCluster
- Amazon 并行计算服务 (Amazon PCS)

内容

- Amazon EC2 容量预留的概念
- 支持的平台
- 配额
- 限制
- 容量预留定价和计费
- 创建 容量预留
- 查看容量预留的状态
- 在现有 容量预留 中启动实例
- 修改活动容量预留
- 修改您的实例的容量预留设置
- 在容量预留之间移动容量
- 从现有容量预留中拆分容量
- 取消 容量预留
- 使用集群置放群组的容量预留
- 容量预留组
- Local Zones 中的容量预留
- Wavelength 区中的 容量预留
- 共享容量预留
- 容量预留实例集
- 使用 CloudWatch 指标监控容量预留使用情况
- 监控容量预留利用率不足
- 监控未来日期容量预留的状态变化
- 可中断容量预留

Amazon EC2 容量预留的概念

以下关键概念适用于容量预留。

主题

- 开始日期和时间

- 结束日期和时间
- 承诺期限
- 未来日期的容量预留评估
- 容量预留属性
- 实例匹配条件

开始日期和时间

开始日期和时间定义了容量预留何时可供使用。容量预留可以立即启动，也可以在未来的某个日期启动。

- 如果选择立即启动容量预留，则预留容量会在您创建后立即可用，并且在容量预留进入活动状态后立即开始计费。您无需作出任何期限承诺。您可以根据需要随时修改容量预留来满足自己的需求，并且可以随时取消容量预留来释放容量并停止产生费用。
- 如果您选择在未来的某个日期启动容量预留，请指定将来需要预留容量的日期和时间以及承诺时长，即在预置请求的容量预留后您承诺在您的账户中保留请求的容量预留的最短期限。在指定的未来日期，一旦容量预留进入活动状态，容量预留将可供使用，并且计费也将从那时开始。一旦您的账户中预置了容量预留，承诺持续时间就会开始。在此期间，您不能将实例计数减少到承诺的实例计数以下、选择早于承诺期限的结束日期，也不能取消容量预留。但是，承诺期限过后，您可以随意以任何方式修改容量预留，也可以取消它以释放预留容量并停止产生费用。

结束日期和时间

结束日期和时间定义容量预留的结束时间以及从您的账户释放预留容量的时间。您可以将容量预留配置为在特定日期和时间自动结束，也可以将其配置为无限期地保持活动状态，直到手动取消它。

如果将容量预留配置为自动结束，则容量预留将在指定时间后一小时内过期。例如，如果指定 `5/31/2019, 13:30:55`，则容量预留保证在 `5/31/2019` 的 `13:30:55` 到 `14:30:55` 之间到期。

预留结束之后，将从您的账户释放预留容量，并且您无法再将实例定位到容量预留。在预留容量中运行的实例继续运行，不会中断。如果定位到容量预留的实例已停止，则在您删除其容量预留定位首选项或者将其配置为定位到其他容量预留之前，无法重启这些实例。有关更多信息，请参阅 修改您的实例的容量预留设置。

承诺期限

承诺期限仅适用于未来日期的容量预留。

承诺期限是您承诺未来日期的容量预留预置后在您的账户中处于活动状态的最短期限。您可以将未来日期的容量预留保留比承诺期限更长的时间，但不能更短。在承诺期限内，以下事项适用：

- 在承诺期限内不能取消容量预留。
- 不能将实例计数减少到承诺的实例计数以下，但可以将其增加。
- 不能将容量预留配置为在承诺期限内的某个日期或时间自动结束。可以在承诺期内延长结束日期和时间。

Amazon EC2 使用您指定的承诺期限来评测是否可以支持该请求。最低承诺期限为 14 天。在评估请求时，Amazon EC2 可能会确定它是否可以支持较短的承诺期限。在这种情况下，Amazon EC2 将安排具有较短承诺期限的未来日期的容量预留。这意味着您承诺在您的账户中保留容量预留的时间比您最初请求的期限要短。

未来日期的容量预留评估

当您请求未来日期的容量预留时，Amazon EC2 会根据容量可用性和您指定的承诺期限评估该请求，以确定是否可以支持它。评估通常会在 5 天内完成。Amazon EC2 在评估请求时会考虑多种因素，包括：

- 预测的容量供应
- 承诺期限
- 相对于开始日期，您提前多久请求容量预留
- 您的请求的大小

您可以提前 5 至 120 天请求未来日期的容量预留。但我们建议您至少提前 56 天（8 周）提出请求，以提高我们为您的请求提供支持的能力。最低承诺期限为 14 天，最小实例数为 32 个 vCPU。

在评估请求时，容量预留保持为 `assessing` 状态。

如果可以支持该请求，则容量预留将进入 `scheduled` 状态，并安排在请求的日期和时间进行传送。在容量预留处于 `scheduled` 状态期间，实例总数保持为 0。计划的容量预留将在请求的日期变为 `active` 状态并可供使用。

如果无法支持该请求，则容量预留将进入 `unsupported` 状态。不传送不支持的容量预留。

当未来日期的容量预留处于 `assessing` 状态时，您可以取消它。

有关更多信息，请参阅 创建未来日期的容量预留。

容量预留属性

创建容量预留时，必须指定下列属性：

- 可用区
- 实例类型
- 平台（操作系统类型）
- 租赁（default 或 dedicated）

只有匹配这些属性的实例才能在容量预留中启动或运行。

实例匹配条件

实例匹配条件或实例资格决定了哪些实例可以在容量预留中启动和运行。容量预留可以具有以下匹配条件之一：

- 开放：容量预留自动匹配所有具有匹配属性（实例类型、平台和可用区）的实例。具有匹配属性的新实例和现有实例会自动在容量预留中运行，无需任何额外配置。
- 针对性：容量预留仅接受具有匹配属性（实例类型、平台和可用区）并明确定位到容量预留的实例。实例必须专门定位到容量预留，才能在其预留容量中启动或运行。这让您能够明确控制允许哪些实例在预留容量中运行，并帮助您避免无意中使用预留容量。

当您请求未来日期的容量预留时，只能指定有针对性的匹配条件。这可确保容量预留交付的容量是增量的，或者是您在交付时拥有的任何正在运行的实例或预留容量的额外容量。当容量预留在您的账户中变为活动状态后，您可以根据需要将实例匹配条件更改为开放。然而，请记住，任何匹配的实例都将自动在容量预留中运行，这可能会导致无意中使用容量，并让您无法为完整的请求实例计数启动新实例。

支持的平台

您必须使用正确的平台创建容量预留，以确保它与您的实例正确匹配。容量预留支持 platform 的以下值：

- Linux/UNIX
- 含有 SQL Server Standard 的 Linux
- 含有 SQL Server Web 的 Linux
- 含有 SQL Server Enterprise 的 Linux
- SUSE Linux

- Red Hat Enterprise Linux
- 含有 SQL Server Standard 的 RHEL
- 含有 SQL Server Enterprise 的 RHEL
- 含有 SQL Server Web 的 RHEL
- 含有 HA 的 RHEL
- 含有 HA 和 SQL Server Standard 的 RHEL
- 含有 HA 和 SQL Server Enterprise 的 RHEL
- Windows
- 含有 SQL Server 的 Windows
- 含有 SQL Server Web 的 Windows
- 含有 SQL Server Standard 的 Windows
- 含有 SQL Server Enterprise 的 Windows

为确保实例在特定的容量预留中运行，容量预留的平台必须与用于启动该实例的 AMI 平台相匹配。对于 Linux AMI，请务必检查 AMI 平台是使用常规值 Linux/UNIX 还是具体值（如 SUSE Linux）。

Console

检查 AMI 平台

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 AMI。
3. 选择 AMI。
4. 在详细信息选项卡上，记下平台详细信息的值。

Amazon CLI

检查 AMI 平台

使用 describe-images 命令并检查 `PlatformDetails` 的值。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Images[*].PlatformDetails
```

下面是示例输出。

```
[
    "Linux/UNIX"
]
```

PowerShell

检查 AMI 平台

使用 Get-EC2Image cmdlet 并检查 PlatformDetails 的值。

```
Get-EC2Image `
    -ImageId ami-0abcdef1234567890 | `
    Select PlatformDetails
```

下面是示例输出。

```
PlatformDetails
---------------
Linux/UNIX
```

配额

允许您预留容量的实例数基于您账户的个按需型实例配额。您可以在配额允许的数量减去已经运行的实例数量范围内，为任意数量的实例预留容量。

处于 assessing、scheduled、pending、active 和 delayed 状态的容量预留计入您的按需型实例配额。

限制

在创建容量预留之前，请注意以下限制。

- 活动和未使用的容量预留会计入您的个按需型实例限制中。
- 容量预留无法从一个Amazon账户转移到另一个账户。但是，您可以与其他 Amazon 账户共享容量预留。有关更多信息，请参阅 共享容量预留。
- 可用区预留实例账单折扣不适用于容量预留。
- 您可以在集群置放群组中创建容量预留。不支持分布和分区置放群组。

- 容量预留不能与专属主机一起使用。容量预留可以与专用实例一起使用。
- [Windows 实例] 容量预留不能用于自带许可（BYOL）。
- [Red Hat 实例] 容量预留可用于自带许可（BYOL）。
- 容量预留不能确保休眠的实例在尝试启动后可以恢复。
- 您可以为至少具有 32 个 vCPU 的实例计数请求未来日期的容量预留。例如，如果您为 `m5.xlarge` 实例请求未来日期的容量预留，则必须请求至少 8 个实例（8 * m5.xlarge = 32 个 vCPU）。
- 您只能为 C、G、I、M、R 和 T 系列中的实例类型请求未来日期的容量预留。

容量预留定价和计费

本节中的主题旨在概述容量预留的定价和计费。

主题

- 定价
- 计费
- 账单折扣
- 查看您的账单

定价

无论您是否在预留容量中运行实例，都按等同的按需费率为您计算容量预留费用，包括专用实例任何使用的区域额外费用。如果您没有使用预留，这将在您的 Amazon EC2 账单中显示为未使用的预留。如果您运行的实例属性与预留匹配，则您只需要为该实例付费，不需要为预留付费。没有任何预付费用或额外收费。

例如，如果您为 20 个 `m4.large` Linux 实例创建容量预留并在同一个可用区中运行 15 个 `m4.large` Linux 实例，则会向您收取 15 个活动的实例和预留中 5 个未使用的实例的费用。

区域性预留实例的账单折扣适用于容量预留。有关更多信息，请参阅账单折扣。

有关更多信息，请参阅 Amazon EC2 定价。

计费

计费在您的账户中预置容量预留后立即开始，并在容量预留保持在账户中预置状态期间继续计费。对于未来日期的容量预留，这意味着只有在请求的未来日期在您的账户中预置了容量预留后才开始计费。

容量预留以秒为单位计费。这意味着会向您收取不足一小时的费用。例如，如果容量预留在您的账户中保持预置状态 24 小时 15 分钟，则会向您收取 24.25 个预留小时的费用。

下面的示例说明如何对容量预留计费。为一个 m4.large Linux 实例创建了容量预留，其按需费率为每使用一小时 0.10 美元。在此示例中，账户内的容量预留预置了五个小时。第一个小时未使用容量预留，因此按照 m4.large 实例类型的标准按需费率计入一小时未使用费用。从第二个小时到第五个小时，m4.large 实例占用了容量预留。在这段时间内，容量预留不会产生任何费用，改为向账户收取占用这部分容量的 m4.large 实例的费用。在第六个小时取消了容量预留，并在预留容量之外正常运行 m4.large 实例。对于这一个小时，将以 m4.large 实例类型的按需费率进行收费。

Hour	1	2	3	4	5	6	Total cost
Unused Capacity Reservation	$0.10	$0.00	$0.00	$0.00	$0.00	$0.00	**$0.10**
On-demand Instance Usage	$0.00	$0.10	$0.10	$0.10	$0.10	$0.10	**$0.50**
Hourly cost	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.10**	**$0.60**

账单折扣

区域性预留实例的账单折扣适用于容量预留。Amazon 自动将这些折扣应用于具有匹配属性的容量预留。当容量预留由某个实例使用时，折扣将适用于该实例。折扣将优先适用于已产生的实例使用量，然后再用于未使用的容量预留。

区域预留实例的账单折扣不适用于容量预留。

有关更多信息，请参阅下列内容：

- Amazon EC2 的预留实例概览
- 计费和购买选项

查看您的账单

您可以在 Amazon 账单与成本管理 控制台上查看您账户的费用情况。

- 控制面板显示了您的账户的花费汇总。
- 在 Bills (账单) 页面上的 Details (详细信息) 下，展开 Elastic Compute Cloud 部分及区域，以获取有关您的容量预留的账单信息。

您可以在线查看费用，也可以下载 CSV 文件。有关更多信息，请参阅 Capacity Reservation line items。

创建 容量预留

您可以随时创建容量预留，以确保在特定可用区拥有可用的计算容量。容量预留可以立即启动，也可以在未来的某个日期启动。只有在容量预留进入 `active` 状态后，容量才可供使用。

> **Note**
>
> 如果您创建了具有 `open` 实例匹配条件的容量预留，并且在容量预留变为活动状态时，您正在运行具有匹配属性的实例，则这些实例将自动在预留容量中运行。为避免这种情况，请使用 `targeted` 实例匹配条件。有关更多信息，请参阅 实例匹配条件。

如果出现以下情况之一，创建容量预留的请求会失败：

- Amazon EC2没有足够的容量来满足请求。请稍后重试、尝试不同的可用区或者尝试较小的请求。如果您的应用程序灵活地跨实例类型和大小，请尝试不同的实例属性。
- 请求的数量超过选定实例系列的个按需型实例限制。增加该实例系列的个按需型实例限制，然后重试。有关更多信息，请参阅 按需型实例限额。

主题

- 创建可立即使用的容量预留
- 创建未来日期的容量预留

创建可立即使用的容量预留

创建可立即使用的容量预留。

Console

创建容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 容量预留 (容量预留)，然后选择 Create 容量预留 (创建容量预留)。
3. 在实例详细信息部分中配置以下设置。
   a. 实例类型：为其预留容量的实例类型。
   b. 平台：您的实例的操作系统。有关更多信息，请参阅 支持的平台。

c. 可用区：在其中预留容量的可用区。

d. 租赁：用于预留容量的租赁类型。选择“默认”可在共享硬件上预留容量，或选择“专用”则在专用于您的账户的硬件上预留容量。

e. （可选）置放群组 ARN：要在其中创建容量预留的集群置放群组的 ARN。有关更多信息，请参阅 使用集群置放群组的容量预留。

f. 总实例数：为其预留容量的实例的数量。如果指定的数量超过了选定实例类型的剩余按需型实例配额，则请求将失败。

4. 在预留详细信息部分中配置以下设置：

a. 容量预留开始：选择立即。

b. 容量预留结束：选择以下选项之一：

- 手动：预留容量，直到您明确取消它。
- 特定时间：在指定的日期和时间自动取消容量预留。

c. 实例资格：选择以下选项之一：

- 开放：（默认值）容量预留匹配具有匹配属性（实例类型、平台、可用区和租赁）的任何实例。如果您启动具有匹配属性的实例，则会自动将其放置到预留容量中。
- 针对性：容量预留仅接受具有匹配属性（实例类型、平台、可用区和租赁）并明确针对预留的实例。

5. 选择创建。

Amazon CLI

创建容量预留

使用 create-capacity-reservation 命令。

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria open|targeted
```

PowerShell

创建容量预留

使用 Add-EC2CapacityReservation cmdlet。

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion open|targeted
```

创建未来日期的容量预留

如果您需要预留容量在未来某个日期和时间可用，请请求未来日期的容量预留。

在您请求未来日期的容量预留后，该请求将接受评估以确定是否可以支持。有关更多信息，请参阅 未来日期的容量预留评估。

注意事项

- 您可以为 C、G、I、M、R 和 T 系列中的实例类型请求未来日期的容量预留。
- 您可以为至少具有 32 个 vCPU 的实例计数请求未来日期的容量预留。例如，如果您为 m5.xlarge 实例请求未来日期的容量预留，则必须请求至少 8 个实例的容量（8 * m5.xlarge = 32 个 vCPU）。
- 您可以提前 5 至 120 天请求未来日期的容量预留。但是，我们建议您至少提前 56 天（8 周）提出请求以提高可支持性。
- 最低承诺期限为 14 天。

Console

创建容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 容量预留 (容量预留)，然后选择 Create 容量预留 (创建容量预留)。
3. 在实例详细信息部分中配置以下设置。
   a. 实例类型：为其预留容量的实例类型。
   b. 平台：您的实例的操作系统。有关更多信息，请参阅 支持的平台。
   c. 可用区：在其中预留容量的可用区。
   d. 租赁：用于预留容量的租赁类型。选择“默认”可在共享硬件上预留容量，或选择“专用”则在专用于您的账户的硬件上预留容量。

e. 总实例数：为其预留容量的实例的数量。如果指定的数量超过了选定实例类型的剩余按需型实例配额，则请求将失败。

4. 在预留详细信息部分中配置以下设置：

   a. 容量预留开始：选择在特定时间。

   b. 开始日期：指定容量预留必须可供使用的日期和时间。有关更多信息，请参阅 开始日期和时间。

   c. 承诺期限：指定您承诺在容量预留交付后保留它的最短期限。有关更多信息，请参阅 承诺期限。

   d. 容量预留结束：选择以下选项之一：

      - 当我取消它时：预留容量，直至您明确取消它。
      - 特定时间：在指定的日期和时间自动取消容量预留。

5. 选择创建。

Amazon CLI

创建容量预留

使用 create-capacity-reservation 命令。

```
aws ec2 create-capacity-reservation \
    --availability-zone az_name \
    --instance-type instance_type \
    --instance-count number_of_instances \
    --instance-platform operating_system \
    --instance-match-criteria targeted \
    --delivery-preference incremental \
    --commitment-duration commitment_in_seconds \
    --start-date YYYY-MMDDThh:mm:ss.sssZ
```

PowerShell

创建容量预留

使用 Add-EC2CapacityReservation cmdlet。

```
Add-EC2CapacityReservation `
    -AvailabilityZone az_name `
```

```
    -InstanceType instance_type `
    -InstanceCount number_of_instances `
    -InstancePlatform operating_system `
    -InstanceMatchCriterion targeted `
    -DeliveryPreference incremental `
    -CommitmentDuration commitment_in_seconds `
    -StartDate  YYYY-MMDDThh:mm:ss.sssZ
```

查看容量预留的状态

Amazon EC2 会持续监控容量预留状态。

由于 Amazon EC2 API 遵循最终一致性模型，创建容量预留后，最多可能需要 5 分钟来表明容量预留处于 `active` 状态。在此期间，容量预留可能会保持 `pending` 状态。但是，容量预留可能已经可供使用，在这种情况下，如果您尝试在其中启动实例，将会成功。

Console

查看容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择容量预留。
3. 选择容量预留。

Amazon CLI

描述容量预留

使用 describe-capacity-reservations 命令。

例如，以下命令会描述所有容量预留。

```
aws ec2 describe-capacity-reservations
```

下面是示例输出。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE",
```

```
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-16T09:03:18.000Z",
            "AvailableInstanceCount": 1,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 1,
            "State": "active",
            "Tenancy": "default",
            "EbsOptimized": true,
            "InstanceType": "a1.medium",
            "PlacementGroupArn": "arn:aws:ec2:us-east-1:123456789012:placement-
group/MyPG"
        },
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef",
            "EndDateType": "unlimited",
            "AvailabilityZone": "eu-west-1a",
            "InstanceMatchCriteria": "open",
            "Tags": [],
            "EphemeralStorage": false,
            "CreateDate": "2019-08-07T11:34:19.000Z",
            "AvailableInstanceCount": 3,
            "InstancePlatform": "Linux/UNIX",
            "TotalInstanceCount": 3,
            "State": "cancelled",
            "Tenancy": "default",
            "EbsOptimized": true,
            "InstanceType": "m5.large"
        }
    ]
}
```

PowerShell

删除容量预留

使用 Get-EC2CapacityReservation cmdlet。

```
Get-EC2CapacityReservation `
    -CapacityReservationId cr-1234abcd56EXAMPLE
```

容量预留状态

容量预留可能具有以下几种状态。

州	说明
active	容量可供使用。
expired	容量预留已在您预留请求中指定的日期和时间自动到期。预留容量不再可供您使用。
cancelled	容量预留已取消。预留容量不再可供您使用。
pending	容量预留请求已成功，但容量预置仍待处理。
failed	容量预留请求失败。请求可能由于无效的请求参数、容量限制或实例限制等约束条件失败。您可以查看 60 分钟内的失败请求。
scheduled	（仅限未来日期的容量预留）未来日期的容量预留请求已获批准，并且容量预留计划于请求的开始日期交付。
assessing	（仅限未来日期的容量预留）Amazon EC2 正在评估您对未来日期的容量预留的请求。有关更多信息，请参阅 未来日期的容量预留评估。
delayed	（仅限未来日期的容量预留）Amazon EC2 在预置请求的未来日期的容量预留时遇到延迟。Amazon EC2 无法在请求的开始日期和时间之前提供请求的容量。
unsupported	（仅限未来日期的容量预留）由于容量限制，Amazon EC2 无法支持未来日期的容量预留请求。您可以查看 30 天内不受支持的请求。将不会交付容量预留。

在现有 容量预留 中启动实例

您只能将实例启动到具有以下条件的容量预留中：

- 具有匹配属性（实例类型、平台、可用区和租赁）
- 具有足够的可用容量
- 处于 active 状态

启动实例时，可以指定在任意 open 容量预留中、特定容量预留中还是容量预留组中启动实例。

或者，您也可以将实例配置为避免在容量预留中运行，即使您拥有具有匹配属性和可用容量的 open 容量预留也是如此。

在容量预留中启动实例会将其可用容量减去所启动实例的数量。例如，如果您启动 3 个实例，容量预留的可用容量将减去 3。

Console

在现有容量预留中启动实例

1. 按照步骤启动实例，但请在完成以下步骤以指定置放群组和容量预留的设置之后再启动实例。
2. 展开高级详细信息，并执行以下操作：
   a. 对于置放群组，选择要在其中启动实例的集群置放群组。
   b. 对于 Capacity Reservation（容量预留），请选择以下选项之一，具体取决于容量预留的配置：
      - 无 – 阻止实例在容量预留中启动。实例使用按需容量运行。
      - 开放 – 在具有匹配属性以及对于所选实例数具有足够容量的任意容量预留中启动实例。如果没有匹配的容量预留具有足够容量，实例使用按需容量。
      - 指定容量预留：将实例启动到选定的容量预留中。如果选定的容量预留没有足够的容量来运行所选数量的实例，实例启动将失败。
      - 指定容量预留资源组：将实例启动到所选容量预留组中任何具有匹配属性和可用容量的容量预留中。如果所选组中没有具有匹配属性和可用容量的容量预留，将会在按需容量中启动实例。
      - 仅指定容量预留：将实例启动到容量预留中。如果未指定容量预留 ID，则实例将启动到开放的容量预留中。如果容量不可用，则实例无法启动。
      - 仅指定容量预留资源组 – 将实例启动到容量预留资源组中的容量预留中。如果未指定容量预留资源组 ARN，则实例将启动到开放的容量预留中。如果容量不可用，则实例无法启动。
3. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。

Amazon CLI

在现有容量预留中启动实例

使用 run-instances 命令并指定 --capacity-reservation-specification 选项。

以下示例在具有匹配属性和可用容量的任意开放容量预留中启动实例：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification CapacityReservationPreference=open
```

以下示例在 targeted 容量预留中启动实例：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

以下示例在指定容量预留组中启动实例：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

以下示例仅在容量预留中启动实例。由于未指定容量预留 ID，该实例将在具有匹配属性和可用容量的任何开放容量预留中启动：

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
```

```
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only
```

以下示例仅在指定容量预留中启动实例。如果指定容量预留中没有可用容量，则实例将启动失败。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification \
        CapacityReservationPreference=capacity-reservations-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234abcd56EXAMPLE}
```

PowerShell

在现有容量预留中启动实例

使用 New-EC2Instance cmdlet。

以下示例在具有匹配属性和可用容量的任意开放容量预留中启动实例：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "open"
```

以下示例在 targeted 容量预留中启动实例：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

以下示例在指定容量预留组中启动实例：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

以下示例仅在容量预留中启动实例。由于未指定容量预留 ID，该实例将在具有匹配属性和可用容量的任何开放容量预留中启动：

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservations-only"
```

以下示例仅在指定容量预留中启动实例。如果指定容量预留中没有可用容量，则实例将启动失败。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType t2.micro `
    -KeyName "my-key-pair" `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservations-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234abcd56EXAMPLE
```

修改活动容量预留

如果您现有的容量预留不适合需要容量的工作负载，则可以修改实例数量、实例资格（open 或 targeted）和结束时间（At specific time 或 Manually）。如果指定的新实例数量超过了选定实例类型的剩余按需型实例限制，则更新将失败。

允许的修改取决于容量预留的状态：

- assessing 或 scheduled 状态：只能修改标签。
- pending 状态：无法以任何方式修改容量预留。

- `active` 状态但仍在承诺期限内：不能将实例计数减少到承诺的实例计数以下，也不能将结束日期设置为早于承诺期限。允许进行所有其他修改。
- `active` 状态但没有承诺期限或承诺期限已过：允许进行所有修改。
- `expired`、`cancelled`、`unsupported` 或 `failed` 状态：不能以任何方式修改容量预留。

注意事项

- 创建后，您无法更改实例类型、平台、可用区或租赁。如果您需要修改任意这些属性，我们建议您取消预留，然后使用所需属性创建新的预留。
- 如果通过将实例资格从 `targeted` 更改为 `open` 来修改现有容量预留，那么与容量预留属性匹配、已将 `CapacityReservationPreference` 参数设置为 `open` 且尚未在容量预留中运行的任何运行中实例，都将自动使用修改后的容量预留。
- 要更改实例资格，容量预留必须完全处于空闲状态（零使用率），因为当实例在预留内运行时，Amazon EC2 无法修改实例资格。

Console

修改容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择容量预留，选择要修改的容量预留，然后选择编辑。
3. 根据需要修改总容量、容量预留结束或实例资格选项，然后选择保存。

Amazon CLI

修改容量预留

使用 modify-capacity-reservation 命令。以下示例将修改指定容量预留，使其为 8 个实例预留容量。

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --instance-count 8
```

PowerShell

修改容量预留

使用 Edit-EC2CapacityReservation cmdlet。以下示例将修改指定容量预留，使其为 8 个实例预留容量。

```
Edit-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0 `
    -InstanceCount 8
```

修改您的实例的容量预留设置

您随时可以为已停止实例修改以下容量预留设置：

- 以具有匹配属性（实例类型、平台、可用区和租赁）以及可用容量的任意容量预留启动。
- 在特定容量预留中启动实例。
- 在容量预留组中任何具有匹配属性和可用容量的容量预留中启动
- 阻止实例在容量预留中启动。

Console

修改实例容量预留设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择实例并选择要修改的实例。停止实例（如果尚未停止）。
3. 依次选择操作、实例设置和修改容量预留设置。
4. 对于容量预留，请选择下列选项之一：

   - 开放 – 在具有匹配属性以及对于所选实例数具有足够容量的任意容量预留中启动实例。如果没有匹配的容量预留具有足够容量，实例使用按需容量。
   - 无 – 阻止实例在容量预留中启动。实例使用按需容量运行。
   - 指定容量预留：将实例启动到选定的容量预留中。如果选定的容量预留没有足够的容量来运行所选数量的实例，实例启动将失败。
   - 指定容量预留组：将实例启动到所选容量预留组中任何具有匹配属性和可用容量的容量预留。如果所选组中没有具有匹配属性和可用容量的容量预留，将会在按需容量中启动实例。
   - 仅指定容量预留：将实例启动到容量预留中。如果未指定容量预留 ID，则实例将启动到开放的容量预留中。如果容量不可用，则实例无法启动。
   - 仅指定容量预留资源组 – 将实例启动到容量预留资源组中的容量预留中。如果未指定容量预留资源组 ARN，则实例将启动到开放的容量预留中。如果容量不可用，则实例无法启动。

Amazon CLI

修改实例容量预留设置

使用 modify-instance-capacity-reservation-attributes 命令。

以下示例将容量预留首选项更改为 none。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=none
```

以下示例的目标是特定容量预留。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
    CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

以下示例将目标更改为特定容量预留组。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \

 CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-
groups:us-west-2:123456789012:group/my-cr-group}
```

以下示例将容量预留首选项更改为 capacity-reservation-only。由于未指定容量预留，实例将在具有匹配属性和可用容量的任何开放容量预留中启动。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification CapacityReservationPreference=capacity-
reservation-only
```

以下示例将容量预留首选项更改为 capacity-reservation-only，并将目标更改为特定容量预留。如果指定容量预留中没有可用容量，则实例将启动失败。

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-1234567890abcdef0 \
    --capacity-reservation-specification \
```

```
        CapacityReservationPreference=capacity-reservation-only \
        CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

PowerShell

修改实例容量预留设置

使用 Edit-EC2InstanceCapacityReservationAttribute cmdlet。

以下示例将容量预留首选项更改为 none。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "none"
```

以下示例的目标是特定容量预留。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

以下示例将目标更改为特定容量预留组。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationTarget_CapacityReservationResourceGroupArn `
        "arn:aws:resource-groups:us-west-2:123456789012:group/my-cr-group"
```

以下示例将容量预留首选项更改为 capacity-reservation-only。由于未指定容量预留，实例将在具有匹配属性和可用容量的任何开放容量预留中启动。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only"
```

以下示例将容量预留首选项更改为 capacity-reservation-only，并将目标更改为特定容量预留。如果指定容量预留中没有可用容量，则实例将启动失败。

```
Edit-EC2InstanceCapacityReservationAttribute `
    -InstanceId i-1234567890abcdef0 `
```

```
    -CapacityReservationSpecification_CapacityReservationPreference "capacity-
reservation-only" `
    -CapacityReservationTarget_CapacityReservationId cr-1234567890abcdef0
```

在容量预留之间移动容量

您可以将容量从一个容量预留移动到另一个容量预留，以根据需要重新分配预留的计算资源。例如，如果您需要在容量预留中随着使用量增长而增加容量，并且另一个预留中有可用容量，则可以在两个预留之间重新分配该容量。

移动容量的先决条件

作为先决条件，两个容量预留必须满足下面的要求：

- 两个预留必须处于活动状态。
- 两个预留都必须归您的 Amazon Web Services 账户所有。您不能在不同的 Amazon Web Services 账户拥有的预留之间移动容量。
- 两个预留必须具有相同的：
  - 实例类型
  - 平台
  - 可用区
  - 租赁
  - 置放群组
  - 结束时间

目标容量预留实例资格（open 或 targeted）和标签不必与源预留匹配。两个预留的配置保持不变，只是源预留的容量减少了，而目标预留的容量增加了。

当您指定要移动的实例数量时，默认情况下，将首先移动任何可用容量，然后移动任何符合条件的正在运行的实例（预留中的已用容量）。例如，如果您从具有 5 个已使用实例和 3 个可用实例的预留中移动 4 个实例，那么 3 个可用实例和 1 个已使用实例也将被移动。

> **Note**
>
> 当您通过指定大于可用容量的移动数量来从预留中移动已用的容量时，只有在容量预留规范为 open 的情况下启动的实例才会被移动。

注意事项

将容量从一个预留移动到另一个预留时，需要考虑以下事项：

- 已用容量只能在与同一组账户共享的具有 open 实例资格的容量预留之间移动。
- 当您移动已用的容量时，系统将随机选择符合条件的实例。您无法指定移动哪些正在运行的实例。如果找不到足够数量的合格实例来满足移动数量，则移动操作将失败。
- 如果您从源预留中移动所有容量，则容量预留将被自动取消。
- 未来日期的容量预留：在承诺期内，您不能移动未来日期的容量预留的容量。

> **Note**
>
> 不支持从容量块移动容量。

移动容量

可以将容量从源容量预留移动到目标容量预留。

Console

移动容量

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择容量预留。
3. 选择具有可移动容量的按需容量预留 ID。
4. 在操作、管理容量下，选择移动。
5. 在移动容量页面的目标容量预留下，从列表中选择预留。
6. 在要移动的数量下，使用滑块或键入要从源容量预留移动到目标容量预留的实例数量。
7. 查看摘要，准备就绪后，选择移动。

Amazon CLI

移动容量

使用 move-capacity-reservation-instances 命令。以下示例将 10 个实例从指定的源容量预留移动到指定的目标容量预留。

```
aws ec2 move-capacity-reservation-instances \
    --source-capacity-reservation-id cr-1234567890abcdef0 \
    --destination-capacity-reservation-id cr-021345abcdef56789 \
    --instance-count 10
```

PowerShell

移动容量

使用 Move-EC2CapacityReservationInstance cmdlet。以下示例将 10 个实例从指定的源容量预留移动到指定的目标容量预留。

```
Move-EC2CapacityReservationInstance `
    -SourceCapacityReservationId cr-1234567890abcdef0 `
    -DestinationCapacityReservationId cr-021345abcdef56789 `
    -InstanceCount 10
```

从现有容量预留中拆分容量

您可以从现有容量预留中拆分容量来创建新的预留。通过拆分容量，您可以将原始预留的一部分分配给特定工作负载或将其与其他 Amazon Web Services 账户共享。例如，要与其他账户部分共享容量预留，您可以拆分部分容量以创建较小大小的容量预留。然后，可以使用 Amazon Resource Access Manager 与其他账户共享较小大小的容量预留。

当您从现有容量预留中拆分容量时，系统会自动创建新容量预留。现有预留将保持不变，但由于拆分的实例数量而减少的总容量除外。在现有容量预留中运行的实例不受影响。您只能将现有预留拆分为一个新容量预留。

除标签外，新容量预留将与现有容量预留具有相同的配置。默认情况下，新容量预留没有任何标签。您可以在拆分操作期间指定新标签。如有必要，也可以在创建新容量预留之后对其进行修改。

当您指定要拆分的实例数量时，默认情况下，将首先拆分任何可用容量，然后移动任何符合条件的正在运行的实例（预留中的已用容量）。例如：如果您从具有 5 个已使用实例和 3 个可用实例的容量预留中拆分 4 个实例，则 3 个可用实例和 1 个已使用实例将拆分到新的预留中。

拆分容量的先决条件

作为先决条件，您的容量预留必须满足下面的要求：

- 源预留必须处于活动状态。
- 源预留必须归您的 Amazon Web Services 账户所有。

> **Note**
>
> 当您通过指定大于可用容量的拆分数量来从预留中拆分已用容量时，只有在容量预留规范为 open 的情况下启动的实例才会被拆分。

注意事项

将容量从一个预留拆分到新的预留时，需要考虑以下事项：

- 已用容量只能分配给具有“开放”实例资格且未与任何账户共享的容量预留。
- 当您拆分已用容量时，系统将随机选择符合条件的实例。您无法指定拆分哪些正在运行的实例。如果找不到足够数量的合格实例来满足拆分数量，则拆分操作将失败。
- 从现有预留中拆分的最大实例数量等于预留的大小减一。例如，如果预留总容量为 5 个实例，则最多可以将 4 个实例拆分为一个新的预留。
- 未来日期的容量预留 – 在承诺期内，您不能拆分未来日期的容量预留的容量。
- 资源组：如果现有容量预留属于资源组，则新容量预留不会自动添加到该资源组。如有必要，可以在创建新容量预留后将其添加到资源组。
- 共享：如果现有容量预留与消费者账户共享，则新容量预留不会自动与消费者账户共享。如有必要，您可以在创建新容量预留后将其共享。
- 集群置放群组：如果现有容量预留是集群置放群组的一部分，则新容量预留将在同一个集群置放群组中创建。

> **Note**
>
> 不支持从容量块中拆分容量。

使用标签控制拆分容量预留的访问权限

您可以使用标签来控制对 Amazon EC2 资源的访问，包括从现有容量预留中拆分容量以创建新容量预留。有关更多信息，请参阅《IAM 用户指南》中的使用标签控制对 Amazon 资源的访问。

要使用标签控制拆分容量预留的访问权限，请确保在策略语句中同时指定资源和请求标签，因为 IAM 策略是根据源容量预留和新创建的容量预留进行评估的。以下示例策略包括带有源容量预留标签 `Owner=ExampleDepartment1` 的 `ec2:ResourceTag` 条件键和带有新创建容量预留标签 `stack=production` 的条件键 `ec2:RequestTag`。

```
{
  "Statement": [
    {
      "Sid": "AllowSourceCapacityReservation",
      "Effect": "Allow",
      "Action": "ec2:CreateCapacityReservationBySplitting",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:capacity-reservation/
cr-1234567890abcdef0",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Owner": "ExampleDepartment1"
        }
      }
    },
    {
      "Sid": "AllowNewlyCreatedCapacityReservation",
      "Effect": "Allow",
      "Action": ["ec2:CreateCapacityReservationBySplitting", "ec2:CreateTags"],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:capacity-reservation/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stack": "production"
        }
      }
    }
  ]
}
```

拆分容量

可以从现有容量预留中拆分容量来创建新的容量预留。

Console

拆分容量

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择容量预留。
3. 选择具有可拆分容量的按需容量预留 ID。
4. 在操作、管理容量下，选择拆分。

5. 在拆分容量预留页面的要拆分的数量下，使用滑块或键入要从当前预留中拆分的实例数量。
6. （可选）为新容量预留添加标签。
7. 查看摘要，准备就绪后，选择拆分。

Amazon CLI

拆分容量

使用 `create-capacity-reservation-by-splitting` 命令。以下示例通过从指定容量预留中拆分 10 个实例来创建新的容量预留。

```
aws ec2 create-capacity-reservation-by-splitting \
    --source-capacity-reservation-id cr-1234567890abdef0 \
    --instance-count 10
```

PowerShell

拆分容量

使用 New-EC2CapacityReservationBySplitting cmdlet。以下示例通过从指定容量预留中拆分 10 个实例来创建新的容量预留。

```
New-EC2CapacityReservationBySplitting `
    -SourceCapacityReservationId cr-1234567890abdef0 `
    -InstanceCount 10
```

取消 容量预留

可以取消处于以下状态之一的容量预留：

- `assessing`
- `active` 但没有承诺期限或承诺期限已过。在承诺期限内不能取消未来日期的容量预留。

> **Note**
>
> 您无法修改或取消容量块。有关更多信息，请参阅 ML 容量块。

如果未来日期的容量预留进入 delayed 状态，则承诺期限将被免除，并且您可以在其进入 active 状态后立即将其取消。

取消容量预留之后，该容量将立即释放，不再保留供您使用。

您可以取消空容量预留以及具有正在运行的实例的容量预留。如果您取消具有正在运行的实例的容量预留，这些实例将继续在容量预留之外正常运行并应用标准按需型实例费率；或者，如果您有匹配的区域预留实例，则应用折扣费率。

取消容量预留之后，定位到其中的实例无法再启动。修改这些实例，使其定位到不同容量预留、启动到任意处于“开放”状态且具有匹配属性和充足容量的容量预留，或者避免将其启动到容量预留中。有关更多信息，请参阅 修改您的实例的容量预留设置。

Console

取消容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择容量预留，然后选择要取消的容量预留。
3. 依次选择取消预留、取消预留。

Amazon CLI

取消容量预留

使用 cancel-capacity-reservation 命令。

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234567890abcdef0
```

PowerShell

取消容量预留

使用 Remove-EC2CapacityReservation cmdlet。

```
Remove-EC2CapacityReservation `
    -CapacityReservationId cr-1234567890abcdef0
```

使用集群置放群组的容量预留

您可以在集群置放群组中创建容量预留，以便为您的工作负载预留 Amazon EC2 计算容量。集群置放群组具有低网络延迟和高网络吞吐量的好处。

在集群置放群组中创建容量预留可以确保您在需要时访问集群置放群组中的计算容量。这非常适合为需要计算扩缩的高性能 (HPC) 工作负载预留容量。它允许您缩减集群，同时确保容量仍可供您使用，以便在需要时可以扩展备份。

在集群置放群组中创建容量预留后，可以与其他 Amazon 账户共享。有关更多信息，请参阅 共享集群置放群组中的容量预留。

主题

- 限制
- 使用集群置放群组中的容量预留
- 共享集群置放群组中的容量预留

限制

在集群置放群组中创建容量预留时，请牢记以下几点：

- 如果现有容量预留不在置放群组中，则您无法修改容量预留以在置放群组中预留容量。要在置放群组中预留容量，您必须在置放群组中创建容量预留。
- 在置放群组中创建容量预留后，您无法对其进行修改以在置放群组之外预留容量。
- 您可以通过修改置放群组中的现有容量预留或在置放群组中创建其他容量预留来增加置放群组中的预留容量。但是，您增加了出现容量不足错误的机会。
- 您只能从所拥有的集群置放群组共享容量预留。不能共享非所属的集群置放群组中的容量预留。
- 您无法删除具有 `active` 容量预留的集群置放群组。删除前，您必须先取消集群置放群组中的所有容量预留。

使用集群置放群组中的容量预留

要开始对集群置放群组使用容量预留，请执行以下步骤。

> **Note**
>
> 如果要在现有集群置放群组中创建容量预留，请跳过步骤 1。然后，对于步骤 2 和 3，请指定现有集群置放群组的 ARN。

任务

- 步骤 1：（条件）创建用于容量预留的集群置放群组
- 步骤 2：在集群置放群组中创建容量预留
- 步骤 3：将实例启动到在集群置放群组的容量预留中

步骤 1：（条件）创建用于容量预留的集群置放群组

仅在需要创建新的集群置放群组时，执行此步骤。要使用现有的集群置放群组，请跳过此步骤，然后对于步骤 2 和 3，使用该集群置放群组的 ARN。

Console

创建集群置放群组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Placement Groups（置放群组），然后选择 Create placement group（创建置放群组）。
3. 对于 Name（名称），为置放群组指定一个描述性名称。
4. 对于 Placement strategy（置放策略），选择 Cluster（集群）。
5. 选择创建群组。
6. 在置放群组表的群组 ARN 列中，记下您创建的集群置放群组的 ARN。下一步中您将需要使用该值。

Amazon CLI

创建集群置放群组

使用 create-placement-group 命令。

```
aws ec2 create-placement-group \
```

```
    --group-name MyPG \
    --strategy cluster
```

请记下输出中返回的置放群组 ARN，因为下一个步骤需要用到。

PowerShell

创建集群置放群组

使用 New-EC2PlacementGroup cmdlet。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy "cluster"
```

请记下输出中返回的置放群组 ARN，因为下一个步骤需要用到。

步骤 2：在集群置放群组中创建容量预留

在集群置放群组中创建容量预留的方式与创建任何容量预留的方式相同。但是，您还必须指定要在其中创建容量预留的集群置放群组的 ARN。

注意事项

- 指定的集群置放群组必须处于 `available` 状态。如果集群置放群组处于 `pending`、`deleting` 或 `deleted` 状态，则请求将失败。
- 容量预留和集群置放群组必须位于同一可用区。如果创建容量预留的请求指定的可用区与集群置放群组的可用区不同，则请求将失败。
- 您只能为集群置放群组支持的实例类型创建容量预留。如果您指定了不受支持的实例类型，则请求将失败。
- 如果您在集群置放群组中创建 open 容量预留，并且现有正在运行的实例具有匹配属性（置放群组 ARN、实例类型、可用区、平台和租期），这些实例将在容量预留中自动运行。
- 如果出现以下情况之一，创建容量预留的请求会失败：
  - Amazon EC2没有足够的容量来满足请求。请稍后重试、尝试不同的可用区或者尝试较小的容量。如果您的工作负载灵活地跨实例类型和大小，请尝试不同的实例属性。
  - 请求的数量超过选定实例系列的个按需型实例限制。增加该实例系列的个按需型实例限制，然后重试。有关更多信息，请参阅 按需型实例限额。

Console

创建容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 容量预留 (容量预留)，然后选择 Create 容量预留 (创建容量预留)。
3. 在创建容量预留页面上，根据需要指定实例类型、平台、可用区、租赁、数量和结束日期。
4. 对于置放群组，选择要在其中创建容量预留的集群置放群组的 ARN。
5. 选择创建。

有关更多信息，请参阅 创建 容量预留。

Amazon CLI

创建容量预留

使用 create-capacity-reservation 命令。对于 `--placement-group-arn`，请指定要在其中创建容量预留的集群置放群组的 ARN。

```
aws ec2 create-capacity-reservation \
    --instance-type instance_type \
    --instance-platform platform \
    --availability-zone-id az_id \
    --instance-count quantity \
    --placement-group-arn "placement_group_arn"
```

PowerShell

创建容量预留

使用 Add-EC2CapacityReservation cmdlet。对于 `-PlacementGroupArn`，请指定要在其中创建容量预留的集群置放群组的 ARN。

```
Add-EC2CapacityReservation `
    -InstanceType instance_type `
    -InstancePlatform platform `
    -AvailabilityZoneId az_id `
    -InstanceCount quantity `
    -PlacementGroupArn "placement_group_arn"
```

步骤 3：将实例启动到在集群置放群组的容量预留中

您可以使用以下一种选项将实例启动到集群置放群组中的容量预留。

- 指定要启动实例的目标集群置放群组的 ARN — 当提供了集群置放群组的 ARN 时，Amazon EC2 会将实例启动到该集群置放群组中。您可以使用以下方法之一：
  - 指定 *open*：无需在实例启动请求中指定容量预留。如果实例具有匹配指定置放群组中的容量预留的属性（置放群组 ARN、实例类型、可用区、平台和租期），这些实例将在容量预留中自动运行。
  - 指定容量预留 - 如果容量预留仅接受目标实例启动，则除了请求中的集群置放群组之外，还必须指定目标容量预留。
  - 指定容量预留组：有关更多信息，请参阅在容量预留组中使用集群置放群组的容量预留。
- 仅指定容量预留组：有关更多信息，请参阅在容量预留组中使用集群置放群组的容量预留。
- 仅指定容量预留：可以将实例启动到集群置放群组的容量预留中。

> **Note**
>
> 通过仅指定容量预留或仅指定容量预留组来启动实例时，实例将启动到集群置放群组中创建的容量预留中，但这些实例不会直接附加到集群置放群组。

Console

在现有容量预留中启动实例

1. 按照步骤启动实例，但请在完成以下步骤以指定置放群组和容量预留的设置之后再启动实例。
2. 展开高级详细信息，并执行以下操作：
   a. 对于置放群组，选择要在其中启动实例的集群置放群组。
   b. 对于 Capacity Reservation（容量预留），请选择以下选项之一，具体取决于容量预留的配置：
      - 打开 – 将实例启动到集群置放群组中任何具有匹配属性和足够容量的 open 容量预留。
      - 按 ID 定位 – 将实例启动到只接受目标实例启动的容量预留中。
      - 按组定位 –在所选容量预留组中任何具有匹配属性和可用容量的容量预留中启动实例。
3. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

在现有容量预留中启动实例

可以使用 run-instances 命令。如果您需要定位特定容量预留或容量预留组，请指定 --capacity-reservation-specification 参数。对于 --placement，请指定 GroupName 参数，然后指定您在上述步骤中创建的置放群组的名称。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count quantity \
    --instance-type instance_type \
    --key-name key_pair_name \
    --subnet-id subnet-0abcdef1234567890 \
    --capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_reservation_id} \
    --placement "GroupName=cluster_placement_group_name"
```

PowerShell

在现有容量预留中启动实例

使用 New-EC2Instance cmdlet。对于 -Placement，请指定 GroupName 参数，然后指定您在上述步骤中创建的置放群组的名称。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType instance_type `
    -KeyName key_pair_name `
    -SubnetId subnet-0abcdef1234567890 `
    -CapacityReservationTarget_CapacityReservationId capacity_reservation_id `
    -Placement_GroupName cluster_placement_group_name
```

共享集群置放群组中的容量预留

可通过仅共享容量预留或同时共享容量预留和创建容量预留的集群置放群组，来共享集群置放群组的容量预留。

通过仅共享容量预留，您仅向使用者账户提供该容量预留的访问权限。使用者账户无法看到或访问创建容量预留的集群置放群组。这样就对使用者账户的访问提供了细粒度的控制。使用者账户无法查看有关集群置放群组的任何信息，包括其 ARN。

当共享集群置放群组和容量预留时，使用者账户可以看到并访问集群置放群组。使用者账户可启动实例并在实例中创建自己的容量预留。

有关详细信息，请参阅以下资源：

- 将实例启动到在集群置放群组的容量预留中
- 共享容量预留
- 共享置放群组

容量预留组

您可以使用 Amazon Resource Groups 创建称为 Resource Groups 的容量预留逻辑集合。资源组是全部位于同一 Amazon 区域的 Amazon 资源的逻辑分组。有关 Resource Groups 的更多信息，请参阅 Amazon Resource Groups 用户指南中的什么是 Resource Groups？

您可以将您拥有的容量预留包含在您的账户中，并将其他 Amazon 账户与您共享的容量预留包含在单个资源组中。您也可以将具有不同属性（实例类型、平台、可用区和租赁）的容量预留包含在单个资源组中。

为容量预留创建资源组时，可以将实例定位到一组容量预留，而不是单个容量预留。定位到一组容量预留的实例与该组中具有匹配属性（实例类型、平台、可用区和租赁）和可用容量的任何容量预留匹配。如果该组中没有具有匹配属性和可用容量的容量预留，相关实例将会使用按需容量运行。如果稍后将匹配的容量预留添加到目标组中，则实例将自动与其预留容量匹配并移动到该容量中。

为防止意外使用组中的容量预留，请将组中的容量预留配置为仅接受明确定位到该容量预留的实例。为此，请在使用 Amazon EC2 控制台创建容量预留时将实例资格设置为仅指定此预留的实例。如果使用 Amazon CLI，请在创建容量预留时指定 `--instance-match-criteria targeted`。这样做可确保只有明确定位到该组或该组中的某个容量预留的实例才能在该组中运行。

如果组中的容量预留在其中的实例正在运行时被取消或过期，则这些实例会自动移动到该组中具有匹配属性和可用容量的另一个容量预留中。如果该组中没有其他具有匹配属性和可用容量的容量预留，则实例将使用按需容量运行。如果稍后将匹配的容量预留添加到目标组中，则实例将自动移动到其预留容量中。

任务

- 创建容量预留组
- 向组中添加容量预留
- 从组中删除容量预留

- 删除容量预留组
- 在容量预留组中使用集群置放群组的容量预留

创建容量预留组

您可以使用以下示例，通过以下请求参数为容量预留创建资源组。

- `AWS::EC2::CapacityReservationPool`：可确保资源组可以作为实例启动的目标。
- `AWS::ResourceGroups::Generic`，并将 `allowed-resource-types` 设置为 `AWS::EC2::CapacityReservation`：可确保资源组仅接受容量预留。

创建组后，您可以向该组添加容量预留。

Amazon CLI

为容量预留创建组

使用 create-group Amazon CLI 命令。

```
aws resource-groups create-group \
    --name MyCRGroup \
    --configuration \
        '{"Type": "AWS::EC2::CapacityReservationPool"}' \
        '{"Type": "AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

PowerShell

为容量预留创建组

使用 New-RGGroup cmdlet。

```
New-RGGroup `
    -Name MyCRGroup `
    -Configuration `
        @{"Type"="AWS::EC2::CapacityReserationPool"} `
        @{"Type"="AWS::ResourceGroups::Generic"; "Parameters"=@{"allowed-resource-
types"=@{"Values"="AWS::EC2::CapacityReservations"}}}
```

向组中添加容量预留

如果您将与您共享的容量预留添加到组中，此时该容量预留取消共享，则它会自动从组中删除。

Amazon CLI

向组中添加容量预留

使用 group-resources 命令。

以下示例向指定组添加两个容量预留。

```
aws resource-groups group-resources \
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1
 \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

向组中添加容量预留

使用 Add-RGResource cmdlet。

以下示例向指定组添加两个容量预留。

```
Add-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-1234567890abcdef1", `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

从组中删除容量预留

您可以使用以下示例从组中删除容量预留。

Amazon CLI

从组中删除容量预留

使用 ungroup-resources 命令。

以下示例从指定组中删除两个容量预留。

```
aws resource-groups ungroup-resources \
    --group MyCRGroup \
    --resource-arns \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd
 \
        arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

PowerShell

从组中删除容量预留

使用 Remove-RGResource cmdlet。

以下示例从指定组中删除两个容量预留。

```
Remove-RGResource `
    -Group MyCRGroup `
    -ResourceArn `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-0e154d26a16094dd", `
        "arn:aws:ec2:sa-east-1:123456789012:capacity-
reservation/cr-54321abcdef567890"
```

删除容量预留组

您可以使用以下示例删除容量预留组。

Amazon CLI

删除组

使用 delete-group 命令。

```
aws resource-groups delete-group --group MyCRGroup
```

PowerShell

删除组

使用 Remove-RGGroup cmdlet。

```
Remove-RGGroup -GroupName MyCRGroup
```

在容量预留组中使用集群置放群组的容量预留

Amazon EC2 提供不同的启动方法供您在容量预留中使用集群置放群组的容量预留。您可以根据工作负载要求，选择以下一种方法来定位容量预留组：

- 指定集群置放群组和容量预留组的 ARN：这将在所选容量预留组中使用任何具有匹配属性和可用容量的可用容量预留。如果所选组中没有具有匹配属性和可用容量的容量预留，将会在按需容量中启动实例。

  > **Note**
  >
  > 使用此方法启动实例时，实例将被置于指定的集群置放群组中。

- 仅指定容量预留组：通过仅指定容量预留组，使用容量预留组中的所有可用容量。启动实例时，将按以下顺序使用容量：
  - 未与任何集群置放群组关联的容量预留。
  - 容量预留组内任何集群置放群组中的容量预留。
  - 如果该组中没有具有匹配属性和可用容量的容量预留，相关实例将会使用按需容量运行，而不会被置于集群置放群组中。

  > **Note**
  >
  > 仅通过指定容量预留组来启动实例时，实例将启动到集群置放群组中创建的容量预留中，但这些实例不会不直接附加到集群置放群组。

Local Zones 中的容量预留

本地区域是在地理位置上靠近用户的Amazon区域的扩展。在本地区域中创建的资源可以通过非常低延迟的通信为本地用户提供服务。有关更多信息，请参阅 Amazon Local Zones。

您可以通过在本地区域中创建新的子网，将 VPC 从其父Amazon区域扩展到该本地区域。当您在本地扩展区中创建子网时，VPC 也会扩展到该本地扩展区。本地区域中的子网与 VPC 中其他子网的运行相同。

通过使用 Local Zones，您可以将容量预留放在更靠近用户的多个位置。您可以按照在常规可用区中创建和使用容量预留的方法，在 Local Zones 中创建和使用容量预留。相同的功能和实例匹配行为适用。有关在 Local Zones 中支持的定价模式的更多信息，请参阅 Amazon Local Zones 常见问题解答。

注意事项

您不能在本地区域中使用容量预留组。

在本地区域中使用容量预留

1. 启用本地区域以在您的Amazon账户中使用。有关更多信息，请参阅《Amazon Local Zones User Guide》中的 Getting started with Amazon Local Zones。
2. 在本地区域中创建容量预留。对于 Availability Zone (可用区)，选择本地区域。本地区域由Amazon区域代码后跟一个指示位置的标识符表示，例如 `us-west-2-lax-1a`。有关更多信息，请参阅创建 容量预留。
3. 在本地扩展区中创建子网。对于 Availability Zone (可用区)，选择本地区域。有关更多信息，请参阅《Amazon VPC 用户指南》中的在 VPC 中创建子网。
4. 启动一个实例。对于 Subnet (子网)，在本地区域（例如 `subnet-123abc | us-west-2-lax-1a`）中选择子网，对于 Capacity Reservation (容量预留)，选择您在本地区域中创建的容量预留所需的规格（可以为 `open` 或按 ID 定位）。有关更多信息，请参阅在现有 容量预留 中启动实例。

Wavelength 区中的 容量预留

利用 Amazon Wavelength，开发人员可以为移动设备和最终用户打造具有超低延迟的应用程序。Wavelength 可以将标准 Amazon 计算和存储服务部署到电信运营商的 5G 网络边缘。您可以将 Amazon Virtual Private Cloud (VPC) 扩展到一个或多个 Wavelength 区域。然后，您可以使用 Amazon EC2 实例之类的Amazon资源来运行需要超低延迟和连接到该区域中的Amazon服务的应用程序。有关更多信息，请参阅 Amazon Wavelength 域。

创建按需 容量预留 时，您可以选择 Wavelength 区域，然后通过指定与 Wavelength 区域关联的子网，将实例启动到 Wavelength 区域内的 容量预留 中。Wavelength 区域由Amazon区域代码后跟一个指示位置的标识符表示，例如 `us-east-1-wl1-bos-wlz-1`。

Wavelength 区域并非在每个区域中都可用。有关支持 Wavelength 区域的区域的信息，请参阅 Amazon Wavelength 开发人员指南中的可用 Wavelength 区域。

注意事项

您不能在 Wavelength 区中使用 容量预留 组。

在 Wavelength 区中使用 容量预留

1. 启用 Wavelength 区域以在您的Amazon账户中使用。有关更多信息，请参阅 Amazon Wavelength 开发人员指南中的 Amazon Wavelength 入门。
2. 在 Wavelength 区中创建 容量预留。对于可用区，选择 Wavelength。Wavelength 由Amazon区域代码后跟一个指示位置的标识符表示，例如 `us-east-1-wl1-bos-wlz-1`。有关更多信息，请参阅创建 容量预留。
3. 在 Wavelength 区中创建子网。对于可用区，选择 Wavelength 区。有关更多信息，请参阅《Amazon VPC 用户指南》中的在 VPC 中创建子网。
4. 启动一个实例。对于子网，在 Wavelength 区（例如 `subnet-123abc | us-east-1-wl1-bos-wlz-1`）中选择子网，对于容量预留，选择您在 Wavelength 中创建的 容量预留 所需的规格（可以为 `open` 或按 ID 定位）。有关更多信息，请参阅在现有 容量预留 中启动实例。

共享容量预留

容量预留共享使容量预留拥有者能够与其他Amazon账户或在Amazon组织内共享其预留容量。这使您能够集中创建和管理容量预留，并跨多个Amazon账户或在Amazon组织内共享预留容量。

在此模型中，拥有容量预留的Amazon账户（拥有者）将与其他Amazon账户（使用者）共享它。使用者可以在与其共享的容量预留中启动实例，所用方式与他们在自己的账户中拥有的容量预留中启动实例的方式相同。容量预留拥有者负责管理容量预留以及它们启动到其中的实例。拥有者无法修改使用者启动到已共享的容量预留中的实例。使用者负责管理启动到与其共享的容量预留中的实例。使用者无法查看或修改由其他使用者或容量预留拥有者拥有的实例。

容量预留拥有者可与以下对象共享容量预留：

- 其 Amazon 组织内部或外部的特定 Amazon 账户
- 其 Amazon 组织内的组织部门
- 其整个 Amazon 组织

共享容量预留的先决条件

- 要共享容量预留，您必须在您的Amazon账户拥有它。无法共享已与您共享的容量预留。
- 您只能为共享租赁实例共享 容量预留。您无法为专用租赁实例共享 容量预留。
- 容量预留共享不适用于新的Amazon账户或具有有限账单历史记录的Amazon账户。

- 要与您的Amazon组织或Amazon组织内的组织部门共享容量预留，您必须允许与 Amazon Organizations 共享。有关更多信息，请参阅 Amazon RAM 用户指南中的允许与 Amazon Organizations 共享。
- 您可以共享处于 `active` 或 `scheduled` 状态的容量预留。您不能共享处于其他状态（如 `assessing` 或 `unsupported`）的容量预留。

相关服务

容量预留共享与 Amazon Resource Access Manager (Amazon RAM) 集成。Amazon RAM 是一项服务，允许您与任何Amazon账户或通过 Amazon Organizations 共享Amazon资源。利用 Amazon RAM，您可通过创建资源共享来共享您拥有的资源。资源共享指定要共享的资源以及与之共享资源的使用者。使用者可以是单个 Amazon 账户或 Amazon Organizations 中的组织部门或整个组织。

Amazon RAM有关 的更多信息，请参阅 Amazon RAM 用户指南。

跨可用区共享

为确保资源分配到区域的各可用区，我们将可用区独立映射到每个账户的名称。这可能会导致账户之间的可用区命名差异。例如，您的 `us-east-1a` 账户的可用区 Amazon 可能与另一 `us-east-1a` 账户的 Amazon 不在同一位置。

要确定容量预留相对于账户的位置，您必须使用可用区 ID (AZ ID)。AZ ID 是跨所有 Amazon 账户的可用区的唯一且一致的标识符。例如，`use1-az1` 是 `us-east-1` 区域的 AZ ID，它在每个 Amazon 账户中的位置均相同。

当您与其他账户共享容量预留时，该容量预留会与由其可用区 ID 标识的特定位置相关联。使用者账户只能在映射到其账户中同一可用区 ID 的可用区内使用共享容量预留。例如，如果您在 `us-east-1a`（可用区 ID `use1-az1`）中创建了容量预留，则使用者必须在其账户中映射到 `use1-az1` 的可用区内启动实例。该可用区可能具有不同的名称，例如 `us-east-1b`。

查看账户中的可用区的 AZ ID

1. 打开 Amazon RAM 控制台 (https://console.aws.amazon.com/ram/home)。
2. 当前区域的 AZ ID 显示在屏幕右侧的 Your AZ ID (您的 AZ ID) 面板中。

共享 容量预留

在与其他 Amazon Web Services 账户共享您拥有的容量预留时，您必须启用它们以便将实例启动到预留容量中。如果您共享开放容量预留，请记住以下内容，因为它可能导致意外的容量预留使用：

- 如果使用者拥有与容量预留的属性匹配的运行中实例、已将 `CapacityReservationPreference` 参数设置为 `open` 且尚未在预留容量中运行，他们将自动使用共享容量预留。
- 如果使用者启动具有匹配属性（实例类型、平台、可用区和租赁）的实例，并且已将 `CapacityReservationPreference` 参数设置为 `open`，则它们将自动启动到共享容量预留中。

要共享容量预留，您必须将它添加到资源共享。资源共享是一项 Amazon RAM 资源，可让您跨 Amazon 账户共享资源。资源共享指定要共享的资源以及与之共享资源的使用者。在使用 Amazon EC2 控制台共享容量预留时，必须将它添加到现有资源共享。要将容量预留添加到新的资源共享，您必须使用 Amazon RAM 控制台创建资源共享。

如果您属于 Amazon Organizations 中的某个企业并且已在您的组织中启用共享，当达到共享先决条件时，组织中的使用者将自动获得对所共享容量预留的访问权限。如果与外部账户共享了容量预留，他们会收到加入资源共享的邀请，并在接受邀请后获得对所共享容量预留的访问权限。

> **Important**
>
> 在与您共享的容量预留中启动实例之前，请通过以下任一方式验证您是否拥有访问共享的容量预留的权限：在控制台中进行查看，或使用 describe-capacity-reservations Amazon CLI 命令对其进行描述。如果您可以在控制台中查看共享的容量预留或者使用 Amazon CLI 描述它，即表示您可以使用它，并且可以在其中启动实例。如果您尝试在容量预留中启动实例，但由于共享失败而无法访问容量预留，则实例将在按需型容量中启动。

Console

使用 Amazon EC2 控制台共享您拥有的容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择容量预留。
3. 选择要共享的容量预留，然后选择操作、共享预留。
4. 选择要将容量预留添加到的资源共享，然后选择共享容量预留。

   使用者可能需要几分钟的时间才能访问共享容量预留。

使用 Amazon RAM 控制台共享您拥有的容量预留

请参阅《Amazon RAM 用户指南》中的创建资源共享。

Amazon CLI

共享您拥有的容量预留

使用 create-resource-share 命令。

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

共享您拥有的容量预留

使用 New-RAMResourceShare cmdlet。

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

停止共享容量预留

容量预留拥有者可以随时停止共享容量预留。以下规则适用：

- 在停止共享时在共享容量中运行的使用者所拥有的实例继续在预留容量之外正常运行，并且根据 Amazon EC2 容量可用性将容量还原到容量预留。
- 与之共享容量预留的使用者不再能够在预留容量中启动新实例。

要停止共享您拥有的共享容量预留，必须从资源共享中将其删除。

Console

使用 Amazon EC2 控制台停止共享您拥有的容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择容量预留。
3. 选择容量预留并选择共享选项卡。

4. 共享选项卡列出了已将容量预留添加到的资源共享。选择要从中删除容量预留的资源共享，然后选择从资源共享中删除。

使用 Amazon RAM 控制台停止共享您拥有的容量预留

请参阅 Amazon RAM 用户指南中的更新资源共享。

Amazon CLI

停止共享您拥有的容量预留

使用 disassociate-resource-share 命令。

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE \
 --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE
```

PowerShell

停止共享您拥有的容量预留

使用 Disconnect-RAMResourceShare cmdlet。

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE"
```

共享 Amazon EC2 容量预留的账单分配

默认情况下，共享容量预留时，所有者需要为其在容量预留中运行的实例以及容量预留中的任何可用容量（也称为未使用容量）付费；而使用者仅为其在共享容量预留中运行的实例付费。

需要时，容量预留所有者可以将容量预留中任何可用容量的账单分配给与之共享容量预留的任何账户。将账单分配给其他账户后，该账户将成为容量预留中任何可用容量的账单所有者。从那时起，容量预留中可用容量的任何费用都将计入分配的账户，而不是所有者的账户。容量预留所有者和与之共享容量预留的账户将继续为其在容量预留中运行的实例收费。

> **Important**
>
> 容量预留所有者始终是资源所有者，并且继续负责管理其容量预留。向其分配了账单的账户并未获得任何其他权限；无法以任何方式取消、修改或共享容量预留。

主题

- 工作原理
- 注意事项
- 将共享 EC2 容量预留的账单分配给其他账户
- 查看共享 EC2 容量预留的账单分配请求
- 接受或拒绝共享 EC2 容量预留的账单
- 取消或撤销共享 EC2 容量预留的账单分配请求
- 监控共享容量预留的账单分配请求

工作原理

只有容量预留所有者才能将共享容量预留的账单分配给其他账户。账单只能分配给与之共享容量预留，且与容量预留所有者合并到同一个 Amazon Organizations 付款人账户下的账户。

要将容量预留的可用容量账单分配给其他账户，容量预留所有者必须向所需账户发起请求。指定账户收到请求后，必须在 12 小时内接受或拒绝请求。

- 如果该账户接受，则将成为容量预留中任何可用容量（也称为未使用容量）的账单所有者。从那时起，容量预留中任何可用容量的费用都将计入其账户，而不是所有者的账户。接受后，只有容量预留所有者才能撤销已分配账户的账单。
- 如果该账户拒绝，则容量预留所有者仍是容量预留中可用容量的账单所有者。容量预留中任何可用容量的费用都将继续计入所有者的账户。
- 如果指定账户在 12 小时内没有接受或拒绝该请求，则该请求将会过期，并且所有者的账户会继续产生容量预留中任何可用容量的费用。

将账单分配给其他账户后，`Reservation` 和 `UnusedBox` 行项目将在分配的账户的成本和使用情况报告（CUR）中显示，而不是在所有者的 CUR 中显示。

下表显示了在将账单分配给其他账户之前，容量预留所有者和使用者账户的 CUR 中显示的行项目。

Account	在分配账单之前的 CUR 行项目
容量预留所有者	• Reservation • BoxUsage * • UnusedBox
与之共享了容量预留的使用者账户	• BoxUsage *

下表显示了在将账单分配给其他账户之后，容量预留所有者和使用者账户的 CUR 中显示的行项目。

Account	在分配账单之后的 CUR 行项目
容量预留所有者	• BoxUsage *
向其分配了账单的使用者账户	• Reservation • BoxUsage * • UnusedBox
与之共享了容量预留的其他使用者账户	• BoxUsage *

> **Note**
>
> - * 仅当有使用容量预留的实例正在运行时，BoxUsage 行项目才会在账户的 CUR 中显示。有关 CUR 行项目的更多信息，请参阅监控容量预留。
> - 使用 CUR 中的容量预留 ARN 可确定谁是容量预留的所有者。如果 ARN 包含您的 Amazon 账户 ID，则您就是容量预留的所有者。否则，容量预留由其他账户所有，但账单分配给了您。
> - 所有者分配给容量预留的成本分配标签不会出现在使用者账户的 CUR 中。成本分配标签仅在容量预留所有者的 CUR 中显示。

注意事项

在分配共享容量预留的账单时，请注意以下几点：

- 不能进行部分账单或拆分账单分配。一次只能将一个容量预留的所有可用容量的账单分配给一个账户。
- 容量预留的可用容量可能会持续变化。这将影响分配的账户的账单。例如，假设容量预留所有者增加了容量预留的大小，或者其他使用者账户停止或终止了在容量预留中运行的实例，则可用容量可能会增加。
- 账单只能分配给合并到同一 Amazon Organizations 付款人账户下的使用者账户。如果该使用者账户离开组织或者不再与其共享容量预留，则会自动撤销其账单。
- 只有容量预留所有者可以取消待处理的账单分配请求，以及在请求被接受后撤销已分配账户的账单。

将共享 EC2 容量预留的账单分配给其他账户

要将共享容量预留的可用容量账单分配给其他账户，容量预留所有者必须向所需账户发起请求。在 Amazon EC2 控制台中，此请求称为转移请求。

只有在满足下列条件时，容量预留所有者才能将容量预留的可用容量的账单分配给某个账户：

- 该容量预留已与该账户共享。
- 该账户与容量预留所有者合并到同一 Amazon Organizations 付款人账户下。

只有在指定账户接受请求后，才会将账单分配给该账户。

容量预留所有者发起请求时，系统会将一个 Amazon EventBridge 事件发送给被请求的账户。有关更多信息，请参阅 监控共享容量预留的账单分配请求。

Console

分配共享容量预留的账单

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航面板中，选择容量预留，然后选择该共享容量预留。
3. 在可用容量的账单部分中，选择分配账单。
4. 在分配账单屏幕中，选择要向其分配账单的使用者账户，然后选择请求。

Amazon CLI

分配共享容量预留的账单

使用 associate-capacity-reservation-billing-owner 命令。对于 `--capacity-reservation-id`，指定共享容量预留的 ID。对于 `--unused-reservation-billing-owner-id`，请指定要向其分配账单的 Amazon 账户的 ID。

```
aws ec2 associate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

分配共享容量预留的账单

使用 Register-EC2CapacityReservationBillingOwner cmdlet。对于 `-CapacityReservationId`，指定共享容量预留的 ID。对于 `-UnusedReservationBillingOwnerId`，请指定要向其分配账单的 Amazon 账户的 ID。

```
Register-EC2CapacityReservationBillingOwner `
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

查看共享 EC2 容量预留的账单分配请求

容量预留所有者只能查看其最新发起的账单分配请求。使用者账户只能查看向其发送的最新账单分配请求。

在请求进入 `cancelled`、`expired` 或 `revoked` 状态后，可以在 24 小时内查看请求。24 小时后，其将无法再查看。

Console

（容量预留所有者）查看您发起的请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航面板中，选择容量预留，然后选择要查看请求的共享容量预留。
3. 可用容量的账单部分会显示最新的请求及其当前状态。

（使用者账户）发送给您的请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航面板中，选择容量预留。
3. 如果您有待处理的请求，则屏幕顶部会显示待处理的账单分配请求横幅。如果未显示该横幅，则表示您没有待处理的请求。

   要查看请求，请选择横幅中的查看请求。

Amazon CLI

（容量预留所有者）查看您发起的请求

使用 describe-capacity-reservation-billing-requests 命令。

```
aws ec2 describe-capacity-reservation-billing-requests \
    --role odcr-owner
```

（使用者账户）查看发送给您的请求

使用 describe-capacity-reservation-billing-requests 命令。

```
aws ec2 describe-capacity-reservation-billing-requests \
```

```
    --role unused-reservation-billing-owner
```

PowerShell

（容量预留所有者）查看您发起的请求

使用 Get-EC2CapacityReservationBillingRequest cmdlet。

```
Get-EC2CapacityReservationBillingRequest `
    -Role odcr-owner
```

（使用者账户）查看发送给您的请求

使用 Get-EC2CapacityReservationBillingRequest cmdlet。

```
Get-EC2CapacityReservationBillingRequest `
    -Role unused-reservation-billing-owner
```

请求可以处于以下某种状态。

州	说明			
pending	该请求尚未被接受或拒绝，也未过期。			
accepted	该请求已被指定账户接受。容量预留的可用容量账单已分配给该使用者账户。			
rejected	该请求也被使用者账户拒绝。			
cancelled	该请求在处于 pending 状态时被容量预留所有者取消。			
revoked	由于以下原因之一，已从该使用者账户撤销了账单： • 容量预留所有者显式撤销了账单。			

州	说明			
	• 容量预留不再与该使用者账户共享。 • 该使用者账户不再是该 Amazon 组织的成员。			
`expired`	由于使用者账户未在 12 小时内接受或拒绝请求，因此请求已过期。			

接受或拒绝共享 EC2 容量预留的账单

收到与您共享的容量预留的账单分配请求后，您可以接受也可以拒绝该请求。请求在被接受或拒绝之前，会保持 `pending` 状态。

如果您接受该请求，则将进入 `accepted` 状态，并且从那时起，该容量预留中任何可用或未使用容量的账单将分配给您的账户。在您接受该请求后，只有容量预留所有者才可以撤销您账户的账单。

如果您拒绝该请求，请求将进入 `rejected` 状态，容量预留的可用容量账单仍分配给容量预留所有者。

如果未在 12 小时内接受或拒绝请求，则请求将会过期。如果请求过期，则容量预留中任何未使用容量的账单仍分配给容量预留所有者。

请求被接受或拒绝时，系统会将一个 Amazon EventBridge 事件发送给容量预留所有者的账户。请求过期时，系统会将一个 Amazon EventBridge 事件发送给容量预留所有者账户和该使用者账户。有关更多信息，请参阅 监控共享容量预留的账单分配请求。

Console

接受或拒绝请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航面板中，选择容量预留。

3. 如果您有待处理的请求，则屏幕顶部会显示待处理的账单分配请求横幅。如果未显示该横幅，则表示您没有待处理的请求。

   要查看请求，请选择横幅中的查看请求。

4. 选择要接受或拒绝的请求，然后选择接受或拒绝。

Amazon CLI

接受请求

使用 accept-capacity-reservation-billing-ownership 命令。对于 `--capacity-reservation-id`，请指定要接受请求的容量预留的 ID。

```
aws ec2 accept-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

拒绝请求

使用 reject-capacity-reservation-billing-ownership 命令。对于 `--capacity-reservation-id`，请指定要拒绝请求的容量预留的 ID。

```
aws ec2 reject-capacity-reservation-billing-ownership \
    --capacity-reservation-id cr-01234567890abcdef
```

PowerShell

接受请求

使用 Approve-EC2CapacityReservationBillingOwnership cmdlet。

```
Approve-EC2CapacityReservationBillingOwnership `
    -CapacityReservationId cr-01234567890abcdef
```

拒绝请求

使用 Deny-EC2CapacityReservationBillingOwnership cmdlet。

```
Deny-EC2CapacityReservationBillingOwnership `
```

```
    -CapacityReservationId cr-01234567890abcdef
```

取消或撤销共享 EC2 容量预留的账单分配请求

只有容量预留所有者才能取消 pending 账单分配请求。如果待处理的请求被取消，请求将进入 cancelled 状态，容量预留中任何可用（即未使用）容量的账单仍分配给容量预留所有者。

请求处于 accepted 状态后，只有容量预留所有者才能撤销已分配账户的账单。如果账单被撤销，请求将进入 revoked 状态，容量预留中任何可用容量的账单将重新分配给容量预留所有者。

请求被取消或撤销时，系统会将 Amazon EventBridge 事件发送给容量预留所有者账户和指定的使用者账户。有关更多信息，请参阅 监控共享容量预留的账单分配请求。

Console

取消或撤销请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航面板中，选择容量预留，然后选择要取消或撤销请求的容量预留。
3. 在可用容量的账单部分中，根据请求的当前状态选择取消转移或撤销转移。

Amazon CLI

取消或撤销请求

使用 disassociate-capacity-reservation-billing-owner 命令。对于 --unused-reservation-billing-owner-id，请指定向其发送了请求的 Amazon 账户的 ID。

```
aws ec2 disassociate-capacity-reservation-billing-owner \
    --capacity-reservation-id cr-01234567890abcdef \
    --unused-reservation-billing-owner-id 123456789012
```

PowerShell

取消或撤销请求

使用 Unregister-EC2CapacityReservationBillingOwner cmdlet。对于 -UnusedReservationBillingOwnerId，请指定向其发送了请求的 Amazon 账户的 ID。

```
Unregister-EC2CapacityReservationBillingOwner `
```

```
    -CapacityReservationId cr-01234567890abcdef `
    -UnusedReservationBillingOwnerId 123456789012
```

监控共享容量预留的账单分配请求

当账单分配请求的状态发生变化时，Amazon EC2 会发送 Amazon EventBridge 事件。

- 当请求进入以下状态时，将向容量预留所有者发送事件：accepted | rejected | expired | revoked。
- 当请求进入以下状态时，将向请求的使用者账户发送事件：pending | expired | cancelled | revoked。

有关 Amazon EventBridge 的信息，请参阅 Amazon EventBridge 用户指南。

以下是 Amazon EventBridge 事件的模式。

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request pending|
accepted|rejected|cancelled|revoked|expired",
   "source":"aws.ec2",
   "account":"account_id",
   "time":"state_change_timestamp",
   "region":"region",
   "resources":[
      "arn:aws:ec2:region:cr_owner_account_id:capacity-reservation/cr_id"
   ],
   "detail":{
      "capacity-reservation-id":"cr_id",
      "updateTime":timestamp,
      "ownerAccountId":"cr_owner_account_id",
      "unusedReservationChargesOwnerID":"consumer_account_id",
      "status":"pending|accepted|rejected|cancelled|revoked|expired",
      "statusMessage":"message
   }
}
```

以下是当使用者账户 (111111111111) 接受共享容量预留 (cr-01234567890abcdef) 的账单分配请求时，将向容量预留所有者 (222222222222) 发送的事件示例。

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"On-Demand Capacity Reservation Billing Ownership Request accepted",
   "source":"aws.ec2",
   "account":"222222222222",
   "time":"2024-09-01Thh:59:59Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:222222222222:capacity-reservation/cr-01234567890abcdef"
   ],
   "detail":{
      "capacity-reservation-id":"cr-01234567890abcdef",
      "updateTime":"2024-08-01Thh:59:59Z",
      "ownerAccountId":"222222222222",
      "unusedReservationChargesOwnerID":"111111111111",
      "status":"accepted",
      "statusMessage":"billing transfer status message"
   }
}
```

共享的容量预留权限

拥有者的权限

拥有者负责管理和取消其共享的容量预留。拥有者无法修改由其他账户拥有的共享容量预留中运行的实例。拥有者仍然负责管理其启动到共享的容量预留中的实例。

使用者的权限

使用者负责管理其正在运行共享的容量预留的实例。使用者不能以任何方式修改共享的容量预留，也不能查看或修改由其他使用者或容量预留拥有者拥有的实例。使用者只能查看共享预留中的总容量和可用容量。

计费和计量

共享容量预留不会产生额外的费用。

默认情况下，容量预留所有者需要为其在容量预留中运行的实例以及未使用的预留容量付费，而使用者需要为其在共享容量预留中运行的实例付费。但是，您可以将共享容量预留的可用容量账单分配给特定的使用者账户。有关更多信息，请参阅 共享 Amazon EC2 容量预留的账单分配。

如果容量预留所有者属于不同的付款人账户，并且容量预留由区域预留实例或节省计划承保，则容量预留所有者将继续为区域预留实例或节省计划付费。在这些情况下，容量预留所有者将为区域预留实例或节省计划付费，使用者需要为在共享容量预留中运行的实例付费。

实例限制

所有容量预留使用量都计入容量预留拥有者的个按需型实例限制。这包括：

- 未使用的预留容量
- 容量预留拥有者拥有的实例的使用情况
- 使用者拥有的实例的使用情况

使用者在共享容量中启动的实例将计入容量预留拥有者的个按需型实例限制。使用者的实例限制是他们自己的个按需型实例限制和他们可以访问的共享容量预留中的可用容量的总和。

容量预留实例集

按需容量预留实例集是一组容量预留。

容量预留实例集请求包含启动容量预留实例集所需的所有配置信息。使用单个请求，您可以跨多个实例类型为您的工作负载预留大量 Amazon EC2 容量，最多可达到您指定的目标容量。

创建容量预留实例集后，您可以通过修改或取消容量预留实例集来集体管理实例集中的容量预留。

主题

- 容量预留实例集的工作原理
- 注意事项
- 定价
- 容量预留实例集概念和规划
- 创建容量预留实例集
- 修改容量预留实例集
- 取消容量预留实例集
- 容量预留实例集配置示例
- 将服务相关角色用于容量预留实例集

容量预留实例集的工作原理

创建容量预留实例集时，实例集会尝试创建单个容量预留以满足您在实例集请求中指定的总目标容量。

实例集为其预留容量的实例数取决于您指定的总目标容量和实例类型权重。实例集为其预留容量的实例类型取决于您使用的分配策略和实例类型优先级。

如果实例集创建时容量不足，且无法立即满足其总目标容量，则实例集会异步尝试创建容量预留，直到其预留了请求的容量为止。

当实例集达到其总目标容量时，它会尝试维持该容量。如果实例集中的容量预留被取消，实例集将根据您的实例集配置自动创建一个或多个容量预留，以替换丢失的容量并维持其总目标容量。

实例集中的容量预留无法单独管理。它们必须通过修改实例集进行集体管理。修改实例集时，实例集中的容量预留将自动更新以反映更改。

目前，容量预留实例集支持 open 实例匹配条件，且实例集启动的所有容量预留都会自动使用此实例匹配条件。在此条件下，具有匹配属性（实例类型、平台、可用区和租赁）的新实例和现有实例在实例集创建的容量预留中自动运行。容量预留实例集不支持目标实例匹配条件。

注意事项

使用容量预留实例集时，请记住以下事项：

- 容量预留实例集可以使用 Amazon CLI 和 Amazon API 创建、修改、查看和取消。
- 实例集中的容量预留无法单独管理。它们必须通过修改或取消实例集进行集体管理。
- 容量预留实例集不能跨区域。
- 容量预留实例集不能跨可用区。
- 容量预留实例集创建的容量预留会自动使用 Amazon 生成的以下标签进行标记：
  - 密钥 — `aws:ec2-capacity-reservation-fleet`
  - 值 — *`fleet_id`*

  您可以使用此标签来标识容量预留实例集创建的容量预留。

定价

使用容量预留实例集不会产生额外的费用。您需要为容量预留实例集创建的单个容量预留付费。有关预留容量计费方式的更多信息，请参阅 容量预留定价和计费。

容量预留实例集概念和规划

以下主题旨在介绍容量预留实例集的规划和概念，包括总目标容量、分配策略、实例类型权重和实例类型优先级。

主题

- 规划容量预留实例集
- 总目标容量
- 分配策略
- 实例类型权重
- 实例类型优先级

规划容量预留实例集

规划容量预留实例集时，建议执行以下操作：

1. 确定您的工作负载所需的计算容量。
2. 决定要使用的实例类型和可用区。
3. 根据您的需求和首选项为每种实例类型分配优先级。有关更多信息，请参阅 实例类型优先级。
4. 创建对您的工作负载有意义的容量加权系统。为每种实例类型分配权重并确定您的总目标容量。有关更多信息，请参阅 实例类型权重 和 总目标容量。
5. 确定您是无限期需要容量预留还是仅在指定时间段内需要容量预留。

总目标容量

总目标容量定义容量预留实例集预留的总计算容量。在创建容量预留实例集时，您需要指定总目标容量。实例集创建后，Amazon EC2 会自动创建容量预留，以预留最高可达到总目标容量的容量。

容量预留实例集为其预留容量的实例的数量取决于总目标容量和您在容量预留实例集中为每个实例类型指定的实例类型权重 (`total target capacity/instance type weight=number of instances`)。

您可以根据对您的工作负载有意义的单位分配总目标容量。例如，如果您的工作负载需要一定数量的 vCPU，则可以根据所需的 vCPU 数量分配总目标容量。如果您的工作负载需要 2048 个 vCPU，则指定总目标容量 2048，然后根据实例集中的实例类型提供的 vCPU 数量分配实例类型权重。有关示例，请参阅实例类型权重。

分配策略

容量预留实例集的分配策略决定了如何根据容量预留实例集配置中的实例类型规范来满足针对预留容量的请求。

目前，仅支持 prioritized 分配策略。通过此策略，容量预留实例集使用您在容量预留实例集配置中为每个实例类型规范分配的优先级来创建容量预留。优先级值越低表示使用优先级越高。例如，假设您创建了使用以下实例类型和优先级的容量预留实例集：

- m4.16xlarge — 优先级 = 1
- m5.16xlarge — 优先级 = 3
- m5.24xlarge — 优先级 = 2

实例集首先尝试为 m4.16xlarge 创建容量预留。如果 Amazon EC2 的 m4.16xlarge 容量不足，实例集将尝试为 m5.24xlarge 创建容量预留。如果 Amazon EC2 的 m5.24xlarge 容量不足，实例集将为 m5.16xlarge 创建容量预留。

实例类型权重

实例类型权重是您分配给容量预留实例集中每种实例类型的权重。权重决定了该指定实例类型的每个实例计入实例集的总目标容量的容量单位数。

您可以根据对您的工作负载有意义的单位分配权重。例如，如果您的工作负载需要一定数量的 vCPU，则可以根据容量预留实例集中每种实例类型提供的 vCPU 数量来分配权重。在这种情况下，如果您使用 m4.16xlarge 和 m5.24xlarge 实例创建容量预留实例集，您将为每个实例分配与 vCPU 数量对应的权重，如下所示：

- m4.16xlarge — 64 个 vCPU，权重 = 64 个单位
- m5.24xlarge — 96 个 vCPU，权重 = 96 个单位

实例类型权重决定了容量预留实例集为其预留容量的实例的数量。例如，如果总目标容量为 384 个单位的容量预留实例集使用前面示例中的实例类型和权重，实例集可以为 6 个 m4.16xlarge 实例（384 总目标容量/64 实例类型权重=6 个实例）或 4 个 m5.24xlarge 实例 (384 / 96 = 4) 预留容量。

如果您没有分配实例类型权重，或者您将实例类型权重分配为 1，则总目标容量完全基于实例数量。例如，如果总目标容量为 384 个单位的容量预留实例集使用前面示例中的实例类型，但忽略权重或为这两种实例类型都指定了 1 的权重，则实例集都可以为 384 个 m4.16xlarge 实例或 384 个 m5.24xlarge 实例预留容量。

实例类型优先级

实例类型优先级是您分配给实例集中的实例类型的值。优先级用于确定为实例集指定的哪些实例类型应优先使用。

优先级值越低表示使用的优先级越高。

创建容量预留实例集

当您创建容量预留实例集时，它会自动为实例集请求中指定的实例类型创建容量预留，最多可达到指定的总目标容量。容量预留实例集为其预留容量的实例数取决于总目标容量和您在请求中指定的实例类型权重。有关更多信息，请参阅 实例类型权重 和 总目标容量。

创建实例集时，您必须指定要使用的实例类型以及每种实例类型的优先级。有关更多信息，请参阅 分配策略 和 实例类型优先级。

> **Note**
>
> Amazon Web Services ServiceRoleForEC2CapacityReservationFleet 服务相关角色在您首次创建容量预留实例集时将在您的账户中自动创建。有关更多信息，请参阅 将服务相关角色用于容量预留实例集。

目前，容量预留实例集仅支持 open 实例匹配条件。

Amazon CLI

要创建容量预留实例集

使用 create-capacity-reservation-fleet 命令。

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 24 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-12-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

以下是 instanceTypeSpecification.json 的内容。

```
[
  {
    "InstanceType": "m5.xlarge",
    "InstancePlatform": "Linux/UNIX",
```

```
    "Weight": 3.0,
    "AvailabilityZone":"us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

下面是示例输出。

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

PowerShell

要创建容量预留实例集

使用 New-EC2CapacityReservationFleet cmdlet。

```
New-EC2CapacityReservationFleet `
    -TotalTargetCapacity 24 `
    -AllocationStrategy "prioritized" `
    -InstanceMatchCriterion "open" `
    -Tenancy "default" `
    -EndDate 2021-12-31T23:59:59.000Z `
    -InstanceTypeSpecification $specification
```

规范定义如下。

```
$specification = New-Object Amazon.EC2.Model.ReservationFleetInstanceSpecification
$specification.InstanceType = "m5.xlarge"
$specification.InstancePlatform = "Linux/UNIX"
$specification.Weight = 3.0
$specification.AvailabilityZone = "us-east-1a"
$specification.EbsOptimized = $true
$specification.Priority = 1
```

修改容量预留实例集

您可以随时修改容量预留实例集的总目标容量和日期。当您修改容量预留实例集的总目标容量时，实例集会自动创建新的容量预留，或者修改或取消实例集中的现有容量预留以满足新的总目标容量。当您修改实例集的结束日期时，所有单个容量预留的结束日期都会相应更新。

注意事项

- 修改实例集后，其状态将转换为 modifying。当实例集处于 modifying 状态时，您无法尝试对其进行其他修改。
- 您无法修改容量预留实例集使用的租赁、可用区、实例类型、实例平台、优先级或权重。如果您需要更改这些参数中的任何一个，您可能需要取消现有实例集并创建具有所需参数的新实例集。
- 您不能在同一个命令中指定 --end-date 和 --remove-end-date。

Amazon CLI

修改容量预留实例集

使用 modify-capacity-reservation-fleet 命令。

示例 1：修改总目标容量

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

示例 2：修改结束日期

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2021-07-04T23:59:59.000Z
```

示例 3：删除结束日期

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --remove-end-date
```

PowerShell

修改容量预留实例集

使用 Edit-EC2CapacityReservationFleet cmdlet。

示例 1：修改总目标容量

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -TotalTargetCapacity 160
```

示例 2：修改结束日期

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -EndDate 2021-07-04T23:59:59.000Z
```

示例 3：删除结束日期

```
Edit-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-01234567890abcedf `
    -RemoveEndDate
```

取消容量预留实例集

当您不再需要容量预留实例集及其预留的容量时，可以取消它。当您取消实例集时，实例集状态将更改为 cancelled，且其不能再创建新的容量预留。此外，实例集中的所有单个容量预留都将取消。之前在预留容量中运行的实例将在共享容量中继续正常运行。

Amazon CLI

取消容量预留实例集

使用 cancel-capacity-reservation-fleets 命令。

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

下面是示例输出。

```
{
    "SuccessfulFleetCancellations": [
        {
```

```
            "CurrentFleetState": "cancelling",
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
}
```

PowerShell

取消容量预留实例集

使用 Stop-EC2CapacityReservationFleet cmdlet。

```
Stop-EC2CapacityReservationFleet `
    -CapacityReservationFleetId crf-abcdef01234567890
```

容量预留实例集配置示例

以下示例创建了使用两种实例类型的容量预留实例集：m5.4xlarge 和 m5.12xlarge。

它基于指定实例类型提供的 vCPU 数量使用权重系统。总目标容量是 480 个 vCPU。m5.4xlarge 提供 16 个 vCPU 并获取 16 的权重，m5.12xlarge 提供 48 个 vCPU 并获取 48 的权重。此权重系统对容量预留实例集进行了配置，为 30 个 m5.4xlarge 实例 (480/16=30) 或 10 个 m5.12xlarge 实例 (480/48=10) 预留容量。

实例集被配置为确定 m5.12xlarge 容量的优先级并获得优先级 1，而 m5.4xlarge 则会获得较低的优先级 2。这意味着实例集首先将尝试预留 m5.12xlarge 容量，并在 Amazon EC2 的 m5.12xlarge 容量不足时仅尝试预留 m5.4xlarge 容量。

实例集为 Windows 实例预留容量，且预留在 October 31, 2021 的 23:59:59 UTC 时会自动过期。

```
aws ec2 create-capacity-reservation-fleet \
    --total-target-capacity 480 \
    --allocation-strategy prioritized \
    --instance-match-criteria open \
    --tenancy default \
    --end-date 2021-10-31T23:59:59.000Z \
    --instance-type-specifications file://instanceTypeSpecification.json
```

下面是 instanceTypeSpecification.json 的内容。

```
[
    {
        "InstanceType": "m5.4xlarge",
        "InstancePlatform":"Windows",
        "Weight": 16,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 2
    },
    {
        "InstanceType": "m5.12xlarge",
        "InstancePlatform":"Windows",
        "Weight": 48,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 1
    }
]
```

将服务相关角色用于容量预留实例集

按需容量预留实例集将使用 Amazon Identity and Access Management（IAM）服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与容量预留实例集直接相关。服务相关角色由容量预留实例集预定义，并包含该服务代表您调用其他 Amazon 服务所需的一切权限。

服务相关角色可让您更轻松地设置容量预留实例集，因为您不必手动添加必要的权限。容量预留实例集定义其服务相关角色的权限，除非另外定义，否则只有容量预留实例集可以代入其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其他 IAM 实体的权限策略。

只有在首先删除相关资源后，您才能删除服务关联角色。这将保护您的容量预留实例集资源，因为您不会无意中删除对资源的访问权限。

容量预留实例集的服务相关角色权限

容量预留实例集使用名为 AWSServiceRoleForEC2CapacityReservationFleet 的服务相关角色，来创建、描述、修改和取消以前代表您在容量预留实例集中创建的容量预留。

AWSServiceRoleForEC2CapacityReservationFleet 服务相关角色信任以下实体来担任角色：

- capacity-reservation-fleet.amazonaws.com

该角色使用 AWSEC2CapacityReservationFleetRolePolicy Amazon 托管式策略。有关更多信息，请参阅 Amazon 托管式策略：AWSEC2CapacityReservationFleetRolePolicy。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅《IAM 用户指南》中的服务关联角色权限。

创建容量预留实例集的服务相关角色

您无需手动创建服务关联角色。使用 create-capacity-reservation-fleet Amazon CLI 命令或 CreateCapacityReservationFleet API 创建容量预留实例集时，系统将自动为您创建服务相关角色。

如果您删除该服务关联角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。创建容量预留实例集时，容量预留实例集会再次为您创建服务相关角色。

编辑容量预留实例集的服务相关角色

容量预留实例集不允许您编辑 Amazon Web Services ServiceRoleForEC2CapacityReservationFleet 服务相关角色。创建服务关联角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色描述。

删除容量预留实例集的服务相关角色

如果不再需要使用某个需要服务关联角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。但是，您必须先删除服务相关角色的资源，然后才能手动删除它。

> **Note**
>
> 如果在您试图删除资源时容量预留实例集服务正在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

要删除 Amazon Web Services ServiceRoleForEC2CapacityReservationFleet 服务相关角色

1. 使用 delete-capacity-reservation-fleet Amazon CLI 命令或 DeleteCapacityReservationFleet API 删除您的账户中的容量预留实例集。
2. 使用 IAM 控制台、Amazon CLI 或 Amazon API 删除 Amazon Web Services ServiceRoleForEC2CapacityReservationFleet 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

容量预留实例集服务相关角色支持的区域

容量预留实例集支持在服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅 Amazon 区域和端点。

使用 CloudWatch 指标监控容量预留使用情况

使用 CloudWatch 指标，您可以通过设置 CloudWatch 警报以在达到使用情况阈值时通知您，高效地监控您的容量预留并识别未使用的容量。这可以帮助您保持恒定的容量预留卷并实现更高级别的利用率。

容量预留每五分钟向 CloudWatch 发送一次指标数据。对于处于活动状态少于五分钟的容量预留，不支持指标。

有关在 CloudWatch 控制台中查看指标的更多信息，请参阅使用 Amazon CloudWatch 指标。有关创建警报的更多信息，请参阅创建 Amazon CloudWatch 警报。

目录

- 容量预留使用情况指标
- 容量预留指标维度
- 查看用于 容量预留 的 CloudWatch 指标

容量预留使用情况指标

AWS/EC2CapacityReservations 命名空间包括以下使用情况指标，可用于监控和维护在为预留指定的阈值内的按需容量。

指标	描述
UsedInstanceCount	当前正在使用的实例数。 单位：计数
AvailableInstanceCount	可用实例的数量。 单位：计数
TotalInstanceCount	您预留的实例总数。

指标	描述
	单位：计数
InstanceUtilization	当前正在使用的预留容量实例的百分比。 单位：百分比

容量预留指标维度

您可以使用以下维度来优化所选区域和账户内上表中列出的指标。

维度	说明
（无维度）	此维度筛选所有容量预留的指定指标。
CapacityReservationId	此维度筛选已确定容量预留的指定指标。
InstanceType	此维度筛选已确定实例类型的指定指标。
AvailabilityZone	此维度筛选已确定可用区的指定指标。
InstanceMatchCriteria	此维度筛选已确定实例匹配条件的指定指标（open 或 targeted）。
InstancePlatform	此维度筛选已确定平台的指定指标数据。
Tenancy	此维度筛选已确定租赁的指定指标。

查看用于 容量预留 的 CloudWatch 指标

指标首先按服务命名空间进行分组，然后按支持的维度进行分组。您可以按照以下过程查看容量预留的各项指标。

使用 CloudWatch 控制台查看容量预留指标

1. 访问 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 如果需要，可以更改区域。从导航栏中，选择您的容量预留所在的区域。有关更多信息，请参阅 Regions and Endpoints。
3. 在导航窗格中，选择指标。
4. 对于所有指标，选择 EC2 容量预留。
5. 从上述指标维度中选择跨所有容量预留、按容量预留、按实例类型、按可用区、按平台、按实例匹配条件、或、按租赁，指标将分别按“无维度”、CapacityReservationId、InstanceType、AvailabilityZone、Platform、InstanceM和 Tenancy 分组。
6. 要对指标进行排序，请使用列标题。要为指标绘制图表，请选中该指标旁边的复选框。

使用 Amazon CLI 查看容量预留指标

使用以下 list-metrics 命令：

```
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

监控容量预留利用率不足

您可以使用以下方法监控容量预留利用率不足：

主题

- Amazon EventBridge 事件
- 电子邮件和 Amazon Health 控制面板通知

Amazon EventBridge 事件

当您账户中的容量预留在特定时间段内的使用率低于 20% 时，Amazon Health 会向 Amazon EventBridge 发送事件。通过 EventBridge，您可以建立触发编程操作的规则，以响应此类事件。例如，您可以创建一个规则：当容量预留的利用率在 7 天内降至 20% 以下时，自动取消容量预留。

EventBridge 中的事件表示为 JSON 对象。该事件独有的字段包含在 JSON 对象的“详细信息”部分。“事件”字段包含事件名称。“结果”字段包含触发事件的操作的已完成状态。有关更多信息，请参阅 Amazon EventBridge 用户指南中的 Amazon EventBridge 事件模式。

有关更多信息，请参阅 Amazon EventBridge 用户指南。

Amazon GovCloud (US) 不支持此功能。

Events

当容量预留的容量使用率低于 20% 时，Amazon Health 会发送以下事件。

- AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION

  以下是当新创建的容量预留在 24 小时内容量使用率低于 20% 时生成的事件示例。

```
{
    "version": "0",
    "id": "b3e00086-f271-12a1-a36c-55e8ddaa130a",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-03-10T12:03:38Z",
    "region": "ap-south-1",
    "resources": [
        "cr-01234567890abcdef"
    ],
    "detail": {
        "eventArn": "arn:aws:health:ap-south-1::event/EC2/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_cr-01234567890abcdef-6211-4d50-9286-0c9fbc243f04",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION",
        "eventTypeCategory": "accountNotification",
        "startTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "endTime": "Fri, 10 Mar 2023 12:03:38 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef"
            }
        ]
```

```
    }
    }
```

- AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY

  以下是当一个或多个容量预留在 7 天内容量使用率低于 20% 时生成的事件示例。

```
{
    "version": "0", "id":"7439d42b-3c7f-ad50-6a88-25e2a70977e2",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-03-07T06:06:01Z",
    "region": "us-east-1",
    "resources": [
        "cr-01234567890abcdef | us-east-1b | t3.medium | Linux/UNIX | 0.0%",
        "cr-09876543210fedcba | us-east-1a | t3.medium | Linux/UNIX | 0.0%"
    ],
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/
EC2/AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY/
AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY_726c1732-d6f6-4037-b9b8-
bec3c2d3ba65",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION_SUMMARY",
        "eventTypeCategory": "accountNotification",
        "startTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "endTime": "Tue, 7 Mar 2023 06:06:01 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "A description of the event will be provided
 here"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cr-01234567890abcdef | us-east-1b | t3.medium |
 Linux/UNIX | 0.0%"
            },
            {
                "entityValue": "cr-09876543210fedcba | us-east-1a | t3.medium |
 Linux/UNIX | 0.0%"
            }
```

```
        ]
    }
}
```

创建 EventBridge 规则

如要在容量预留利用率降至 20% 以下时接收电子邮件通知，请创建一个 Amazon SNS 主题，然后为 AWS_EC2_ODCR_UNDERUTILIZATION_NOTIFICATION 事件创建 EventBridge 规则。

创建 Amazon SNS 主题

1. 通过 https://console.aws.amazon.com/sns/v3/home 打开 Amazon SNS 控制台。
2. 在导航窗格中，选择主题，然后选择创建主题。
3. 对于类型，选择标准。
4. 对于名称，输入新主题的名称。
5. 选择创建主题。
6. 选择创建订阅。
7. 对于协议，选择电子邮件，然后对于端点，输入接收通知的电子邮件地址。
8. 选择创建订阅。
9. 上面输入的电子邮件地址将收到具有以下主题行的电子邮件：AWS Notification - Subscription Confirmation。请按照说明确认订阅。

创建 EventBridge 规则

1. 访问 https://console.aws.amazon.com/events/，打开 Amazon EventBridge 控制台。
2. 在导航窗格中，选择规则，然后选择创建规则。
3. 对于名称，输入新规则的名称。
4. 对于规则类型，选择具有事件模式的规则。
5. 选择下一步。
6. 在事件模式中，执行以下操作：
    a. 对于事件源，选择Amazon 服务。
    b. 对于 Amazon Service，选择 Amazon Health。
    c. 对于事件类型，选择 EC2 ODCR 利用不足通知。

7. 选择下一步。
8. 对于目标 1，执行以下操作：
   a. 对于目标类型，选择Amazon 服务。
   b. 对于 Select a target（选择一个目标），选择 SNS topic（SNS 主题）。
   c. 对于主题，选择您之前创建的主题。
9. 选择下一步，然后再次选择下一步。
10. 选择 Create rule（创建规则）。

电子邮件和 Amazon Health 控制面板通知

当您的账户中容量预留的容量利用率降至 20% 以下时，Amazon Health 会发送以下电子邮件和 Amazon Health Dashboard 通知。

- 单独通知：针对在过去 24 小时内利用率低于 20% 的每个新创建的容量预留。
- 摘要通知：针对在过去 7 天内利用率低于 20% 的所有容量预留。

电子邮件通知和 Amazon Health Dashboard 通知将发送到与拥有容量预留的 Amazon 账户相关联的电子邮件地址。这些通知包含以下信息：

- 容量预留的 ID。
- 容量预留的可用区。
- 容量预留的平均利用率。
- 容量预留的实例类型和平台（操作系统）。

此外，当您的账户中容量预留的容量利用率在 24 小时和 7 天内下降到 20% 以下时，Amazon Health 会向 EventBridge 发送事件。通过 EventBridge，您可以创建规则来激活自动操作，例如发送电子邮件通知或触发 Amazon Lambda 功能，以响应此类事件。有关更多信息，请参阅 监控容量预留利用率不足。

监控未来日期容量预留的状态变化

当未来日期容量预留的状态发生变化时，Amazon EC2 会向 Amazon EventBridge 发送事件。

以下是此事件的示例。在此示例中，未来日期的容量预留进入 `scheduled` 状态。请注意 `detail-type` 字段中突出显示的状态。

```
{
   "version":"0",
   "id":"12345678-1234-1234-1234-123456789012",
   "detail-type":"EC2 Capacity Reservation Scheduled",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"yyyy-mm-ddThh:mm:ssZ",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdefg"
   ],
   "detail":{
      "capacity-reservation-id":"cr-1234567890abcdefg",
      "state":"scheduled"
   }
}
```

`detail-type` 字段可能的值为：

- `Scheduled`
- `Active`
- `Delayed`
- `Unsupported`
- `Failed`
- `Expired`

有关这些状态的更多信息，请参阅 查看容量预留的状态。

您可以创建 Amazon EventBridge 事件来监控这些事件，然后在事件发生时触发特定的操作。有关更多信息，请参阅 Creating rules that react to events in Amazon EventBridge。

要创建监控所有状态改变事件的规则，您可以使用以下事件模式。

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation"
  }]
}
```

要创建仅监控特定状态变化的规则，您可以使用以下事件模式。

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation state"
  }]
}
```

例如，以下事件模式监控未来日期的容量预留进入 active 状态时发送的事件。

```
{
  "source": ["aws.ec2"],
  "detail-type": [{
    "prefix": "EC2 Capacity Reservation Active"
  }]
}
```

可中断容量预留

可中断容量预留可帮助您将未使用的容量临时用于您的账户中的其他工作负载。这使您可以获得控制权，以在需要时回收容量。当您回收容量时，在可中断预留内运行的所有实例都将被终止。创建可中断预留后，您可以使用 Amazon Resource Access Manager（RAM）与其他 Amazon 账户或您的 Amazon 组织共享该预留。

如果您在非高峰时段、部署间隙或工作负载缩减时拥有未使用的预留容量，请使用可中断容量预留。如果您知道另一个团队可以使用此容量，则可以通过创建可中断容量预留来使其可用。当您的关键工作负载需要恢复容量时，您可以回收该容量。

您可以将可中断容量预留用作以下对象之一：

- 容量所有者：您拥有源容量预留并创建可中断容量预留，以与其他团队共享未使用的容量，同时保留控制权，以便在需要时回收该容量。
- 容量使用者：您将实例启动到共享的可中断预留中，但须知，当所有者回收容量时，您的实例可能会被终止。

主题

- 工作原理
- 计费

- 注意事项
- 容量所有者的可中断容量预留
- 容量使用者的可中断容量预留
- 使用 EventBridge 和 CloudTrail 监控可中断容量预留

工作原理

要将未使用的容量提供给其他团队，请通过指定要从源预留共享的未使用实例数量来创建可中断预留。当您创建可中断预留时，我们会将这些实例从您的源预留转移到您账户中新的可中断预留。

我们会保留源预留和可中断容量预留之间的关联。因此，当您回收容量时，所有正在运行的使用者实例都将被终止，并且容量将恢复到原始源预留。

主要功能：

- 让未使用的容量暂时可用，同时保持控制权以回收容量
- 随时回收容量。有关更多信息，请参阅 回收过程和跟踪。
- 使用 Amazon Resource Access Manager（RAM）与其他账户或您的 Amazon 组织共享

计费

创建可中断预留时，系统会将其作为独立的新预留进行计费。这将拆分您的计费：

- 源预留：按总容量减去分配的容量得出的差额进行计费
- 可中断预留：按分配的容量进行计费

有关按需容量预留计费的更多信息，请参阅容量预留定价和计费。

注意事项

在使用可中断容量预留之前，请查看以下适用于容量所有者和使用者的限制和要求。

容量所有者

- 您不能直接修改或取消可中断容量预留。要编辑它，请更新从源容量预留中分配的容量。
- 您只能查看、启动、标记、共享和分配可中断预留的计费。
- 每个源容量预留只能创建一个可中断分配。

容量使用者

- 可中断容量预留默认情况下是有针对性的容量预留，因此您需要在实例启动时将其作为目标。
- 您无法将可中断容量预留添加到容量预留组。
- 我们建议您仅对可能中断的应用程序使用可中断容量预留。
- 当所有者回收容量时，您的实例将被终止，不会回退为按需型实例或竞价型实例。有关更多信息，请参阅 中断体验。

容量所有者的可中断容量预留

容量所有者是拥有源容量预留的账户，它创建可中断容量预留以与其他团队共享未使用的容量，同时保留控制权，以便在需要时回收该容量。

本节介绍您（容量所有者）如何创建、修改、回收和跟踪可中断容量预留。

主题

- 创建可中断容量预留
- 查看可中断容量预留
- 修改可中断容量预留
- 回收过程和跟踪
- 共享可中断预留

创建可中断容量预留

创建可中断容量预留，使源预留中未使用的容量可用于其他工作负载，同时保持控制权，以便在需要时回收该容量。

先决条件

在创建可中断分配之前，请确保源按需容量预留满足下面的要求：

- 容量预留必须处于活跃状态，且未设置结束日期。您无法从待处理、已过期、已取消或具有计划的结束日期的预留创建分配。
- 容量预留必须具有可供分配的容量。您只能分配可用的实例（也称为未使用的容量）。
- 每个源容量预留只能创建一个可中断分配。如果分配已存在，则必须在创建新分配之前先修改或取消该分配。

- 一次最多可以为可中断容量预留分配 1000 个实例。

用户可以使用控制台或 Amazon CLI 创建可中断容量预留。

Console

创建可中断容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择容量预留。
3. 选择容量预留。
4. 选择操作、创建可中断分配。
5. 对于要分配的实例，输入要分配的实例数量。
6. （可选）添加标签。
7. 选择创建可中断容量分配。

Amazon CLI

创建可中断容量预留

使用 create-interruptible-capacity-reservation-allocation 命令：

```
aws ec2 create-interruptible-capacity-reservation-allocation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --instance-count 10
```

查看可中断容量预留

在创建可中断容量预留后，您可以在您的账户中或从特定资源查看可中断预留。

查看您账户中的所有可中断容量预留

请按照以下过程查看您的账户中的可中断容量预留。

Console

查看您账户中的可中断容量预留

1. 前往控制台中的“容量预留”页面。

2. 查找类型列中为可中断的预留。
3. 选择可中断预留以查看详细信息。

Amazon CLI

查看您账户中的可中断容量预留

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-id cr-interruptible-id \
    --filters Name=interruptible,Values=true
```

查看来自特定源的可中断容量预留

请按照以下过程查看从特定源容量预留创建的可中断容量预留。

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-id cr-source-id
```

在响应中，您将找到一个包含可中断容量预留 ID 和分配详细信息的 `interruptibleCapacityAllocations` 对象。有关响应结构的信息，请参阅《Amazon EC2 API 参考》中的 InterruptibleCapacityAllocation。

修改可中断容量预留

请按照以下过程编辑或取消可中断容量预留。

> **Note**
>
> - 当您减少分配时，我们会先回收可用的实例，然后再回收正在运行的实例，直到达到请求的计数。如果我们能够完全利用可用实例满足计数，则不会发生终止。对已分配实例计数所做的所有修改都是通过源容量预留完成的，而不是直接在可中断容量预留上进行。
> - 一次最多只能修改 1000 个实例的可中断容量预留（增加或减少）。

编辑可中断容量预留

请按照以下过程编辑可中断容量预留。

Console

1. 在源容量预留详细信息页面中，选择操作。然后，编辑可中断容量预留。
2. 对于要分配的实例，输入新数字：
   - 添加更多共享容量
   - 将容量回收到源容量预留
3. 选择更新。

Amazon CLI

```
aws ec2 update-interruptible-capacity-reservation-allocation \
    --capacity-reservation-id cr-1234567890abcdef0 \
    --target-instance-count 80
```

取消可中断容量预留

请按照以下过程永久移除分配并返回所有容量。

Console

1. 在源容量预留详细信息页面上，导航至可中断容量分配详细信息。
2. 选择编辑可中断分配。
3. 对于实例计数，输入 0。
4. 选择更新。

Amazon CLI

```
aws ec2 update-interruptible-capacity-reservation-allocation \
--capacity-reservation-id cr-1234567890abcdef0 \
--target-instance-count 0
```

回收过程和跟踪

当您回收容量时：

- 正在运行的实例会通过 EventBridge 事件收到 2 分钟中断警告。

- 在通知期之后，以回收容量运行的实例将进入关闭状态并被终止。
- 终止后，回收的实例将在源容量预留中可供立即使用。
- 完成后，分配状态将从正在更新变为活跃。

完全回收可能需要几分钟，具体取决于实例类型和关闭时间。有关过程完成后您收到的 EventBridge 通知的更多信息，请参阅回收完成。

跟踪回收状态

通过描述您的源预留来监控回收进度：

```
aws ec2 describe-capacity-reservations \
--capacity-reservation-id cr-1234567890abcdef0
```

响应会显示 `interruptibleCapacityAllocation` 对象中的以下字段：

- `instance-count`：当前分配的实例
- `target-instance-count`：回收后的请求数量
- `status`：回收期间为正在更新，完成后为活跃

共享可中断预留

您只能使用 Amazon Resource Access Manager（RAM）在您的 Amazon 组织内共享可中断预留。

注意事项：

- 如果使用者账户离开您的组织，则该账户将自动取消共享可中断预留。
- 在取消共享的预留中运行的任何实例最终将被终止。
- 所有其他共享功能的工作原理与标准容量预留相同。

有关完整的共享过程，请参阅共享容量预留。

容量使用者的可中断容量预留

容量使用者是将实例启动到共享的可中断容量预留的账户，但须知，当所有者回收容量时，其实例可能会被终止。

本节介绍您（容量使用者）如何将实例启动到可中断容量预留，并了解当所有者回收容量时会发生什么情况。

主题

- 查看可中断容量预留
- 将实例启动到可中断预留中
- 中断体验

查看可中断容量预留

请按照以下过程查看可中断容量预留。

Console

查看您账户中的可中断容量预留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择容量预留。
3. 在类型列中，查找标记为可中断的预留。
4. 请记下预留 ID，以便在您的实例启动时使用。

Amazon CLI

查找您账户中的所有可中断容量预留

使用 describe-capacity-reservations 命令：

```
aws ec2 describe-capacity-reservations \
--filters Name=state,Values=active
```

查找响应中 `Interruptible` 设置为 `true` 的预留。

专门筛选可中断预留

使用以下命令：

```
aws ec2 describe-capacity-reservations \
--capacity-reservation-ids cr-example123 \
--query 'CapacityReservations[?Interruptible==`true`]'
```

> **Note**
>
> 可中断容量预留默认情况下是有针对性的容量预留，因此您需要在实例启动时专门将其作为目标。与开放式预留不同，可中断预留不会自动涵盖匹配的实例。启动时必须明确指定预留 ID。

将实例启动到可中断预留中

请按照以下过程将 Amazon EC2 实例启动到您的账户中的可中断容量预留中。

> **Note**
>
> 我们建议您仅对可能中断的应用程序使用可中断容量预留。

Console

将实例启动到可中断容量预留中

1. 打开 Amazon EC2 控制台，网址为 https://console.aws.amazon.com/ec2/。
2. 在 Amazon EC2 控制面板中，选择 Launch Instance (启动实例)。
3. 配置您的实例设置。
4. 在容量预留的高级详细信息中，选择在您的有效预留中启动可中断实例。
5. 选择可中断预留 ID 和新的实例购买选项。
6. 选择启动实例。

Amazon CLI

```
aws ec2 run-instances \
--instance-type m5.large \
--count 2 \
--image-id ami-12345678 \
--instance-market-options '{
    "MarketType": "interruptible-capacity-reservation"
}' \
--capacity-reservation-specification '{
    "CapacityReservationTarget": {
        "CapacityReservationId": "cr-abcdef1234567890"
    }
```

```
}'
```

使用自动扩缩组启动实例

您还可以将自动扩缩组与启动模板结合使用来将实例启动到可中断的预留中。为启动模板配置可中断市场类型和预留 ID，然后使用该模板创建自动扩缩组。有关更多信息，请参阅 Interruptible Capacity Reservations with EC2 Auto Scaling。

中断体验

当所有者回收容量时，您将在实例终止前 2 分钟收到中断通知。此警告来自 EventBridge 事件，让您有时间执行以下操作：

- 保存所做的工作或检查应用程序
- 关闭过程
- 准备终止实例

EventBridge 事件包括有关哪些实例将被终止以及确切终止时间的详细信息。有关更多信息，请参阅 实例中断警告。

使用 EventBridge 和 CloudTrail 监控可中断容量预留

可中断容量预留会发送 EventBridge 通知和 CloudTrail 事件，以帮助您监控和响应容量更改。

主题

- EventBridge 通知
- CloudTrail 事件

EventBridge 通知

您会收到两种类型的 EventBridge 通知。有关如何设置 EventBridge 通知的信息，请参阅 Creating Amazon EventBridge rules。

实例中断警告

如果您在可中断预留中运行实例，则会在实例终止前 2 分钟收到此通知：

```
{
```

```
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Capacity Reservation Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "[instance owner Account ID]",
    "time": "[Current time in yyyy-mm-ddThh:mm:ssZ]",
    "resources": "[instance arn]",
    "region": "[region]",
    "detail": {
        "instance-id": "[instance-id]",
        "instance-action": "terminate",
        "instance-termination-time": "yyyy-mm-ddThh:mm:ssZ",
        "azId": "[availability-zone-id]"
    }
}
```

回收完成

如果您拥有源预留，则在容量回收完成后会收到此通知：

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Interruptible Capacity Reservation Allocation Reclamation
 Completed",
    "source": "aws.ec2",
    "account": "[source Capacity Reservation Owner Account ID]",
    "time": "[Current time in yyyy-mm-ddThh:mm:ssZ]",
    "region": "us-east-1",
    "resources": ["source_cr_arn"],
    "detail": {
        "sourceCapacityReservationId": "string",
        "instanceType": "string",
        "availabilityZoneId": "string",
        "TotalInstanceCount": "current total count in the source",
        "ReclaimedInstanceCount": "count of instances added to the source",
        "targetInstanceCount": "number"
    }
}
```

CloudTrail 事件

CloudTrail 会对可中断容量预留记录以下事件：

- `InterruptibleCapacityReservationCreated`：当您创建可中断分配时
- `InterruptibleCapacityReservationAllocationUpdated`：当您修改分配时
- `InterruptibleCapacityReservationCancelled`：当您取消分配时
- `CapacityReservationModified`：当我们修改源预留以进行分配时
- `InterruptibleCapacityReservationInstancesTerminated`：当我们在回收期间终止实例时

ML 容量块

机器学习容量块允许您在未来预留基于 GPU 的加速型计算实例，以支持您的短期机器学习工作负载。在容量块内运行的实例会自动紧密放置在 Amazon EC2 UltraClusters 中，以实现低延迟、PB 级非阻塞联网。

您还可以使用容量块为 Amazon EC2 UltraServers 预留容量。UltraServer 在低延迟、高带宽的加速器互连中连接多个 Amazon EC2 实例。您可以使用 UltraServer 来处理训练、微调和推理中最需要计算和内存的 AI/ML 工作负载。有关更多信息，请参阅 Amazon EC2 UltraServers。

使用容量块，您可以了解 GPU 实例容量在未来何时可用，并安排容量块在最适合您的时间启动。当您预留容量块时，您可以获得 GPU 实例的可预测容量保证，同时只需为所需的时间付费。如果您需要 GPU 一次支持几天或几周的 ML 工作负载，并且不想在未使用 GPU 实例时支付预留费用，我们建议您使用容量块。

以下是容量块的一些常见使用案例。

- ML 模型训练和微调 - 无中断地访问您为完成 ML 模型训练和微调而预留的 GPU 实例。
- ML 实验和原型 - 运行实验并构建需要短期 GPU 实例的原型。

某些 Amazon 区域的部分实例类型可使用容量块。有关更多信息，请参阅 支持的实例类型和区域。

您可以预留容量块，预留开始时间最长为未来 8 周。每个容量块最多可以有 64 个实例，您最多可以跨容量块拥有 256 个实例。

主题

- 支持的实例类型和区域
- 支持的平台
- 注意事项

- 相关资源
- Amazon EC2 容量块工作原理
- 容量块定价和计费
- 查找和购买容量块
- 使用容量块启动实例
- 查看容量块
- 延长容量块期限
- 共享容量块
- 为 UltraServer 容量块创建资源组
- 使用 EventBridge 监控容量块
- 使用 Amazon CloudTrail 记录容量块 API 调用

支持的实例类型和区域

实例和 UltraServer 容量块可用于以下实例类型和 Amazon 区域。

> **Note**
>
> 并非所有 Amazon Web Services 区域 中的所有实例类型都支持 64 个实例的容量块大小。

实例容量块

实例类型	美国东部（弗吉尼亚州北部）eas st-1	美国东部（俄亥俄州）us eas 2	美国西部（北加利福尼亚）wes t-1	美国西部（俄勒冈州）us wes 2	欧洲地区（斯德哥尔摩）nort h-1	欧洲地区（伦敦）eu-wes	欧洲（西班牙）- sou	亚太地区（东京）p- nort st-1	亚太地区（首尔）p- nort st-2	亚太地区（孟买）p- sou	亚太地区（悉尼）p- sou st-2	亚太地区（雅加达）ap- sou ast-	亚太地区（墨尔本）ap- sou ast-	南美洲（圣保罗）a- eas	Ama Gov 美国东部）g ov- eas	Ama Gov 美国西部）g ov- wes	美国东部（亚特兰大）us eas 1- atl-2	美国西部（凤凰城）us- west-2 - phx-2a
p6-b300.8xlar	✓			✓											✓			
p6-b200.8xlar	✓	✓		✓											✓	✓		
p5.4xe	✓	✓		✓		✓		✓		✓	✓			✓				
p5.48ge	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		✓			✓	
p5e.4rge	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		✓				✓
p5en.arge	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓						
p4d.2rge	✓	✓		✓														

实例类型	美国东部（弗吉尼亚州北部）east-1	美国东部（俄亥俄州）us-east-2	美国西部（北加利福尼亚）west-1	美国西部（俄勒冈州）us-west-2	欧洲地区（斯德哥尔摩）north-1	欧洲地区（伦敦）eu-wes	欧洲（西班牙）-sou	亚太地区（东京）p-nort st-1	亚太地区（首尔）p-nort st-2	亚太地区（孟买）p-sou	亚太地区（悉尼）p-sou st-2	亚太地区（雅加达）ap-sou ast-	亚太地区（墨尔本）ap-sou ast-	南美洲（圣保罗）a-eas	Ama Gov 美国东部）g ov-eas	Ama Gov 美国西部）g ov-wes	美国东部（亚特兰大）us east 1-atl-2	美国西部（凤凰城）us-west-2 -phx-2a
p4de.arge	✓			✓														
trn1.arge	✓	✓		✓	✓					✓	✓		✓					
trn2.rge													✓	✓				
trn2.arge		✓																

UltraServer 容量块

实例类型	美国东部（俄亥俄州）us-east-2	美国东部（达拉斯）us-east-1-dfw-2a
Trn2	✓	
P6e-GB200		✓

支持的平台

适用于 ML 的容量块目前仅支持具有默认租赁的实例和 UltraServer。当您使用 Amazon Web Services 管理控制台 购买容量块时，默认平台选项为 Linux/UNIX。当您使用 Amazon Command Line Interface（Amazon CLI）或 Amazon SDK 购买容量块时，可以使用以下平台选项：

- Linux/Unix
- Red Hat Enterprise Linux
- 含有 HA 的 RHEL
- SUSE Linux
- Ubuntu Pro

注意事项

在使用容量块之前，请考虑以下细节和限制。

- 如果我们检测到影响了 UltraServer 容量块的缺陷，我们会通知您，但通常不会采取措施终止您在容量块上的实例。这是为了最大限度地减少对工作负载的意外中断。收到此通知后，您可以继续按原样使用 UltraServer 容量块，也可以通过终止容量块上的所有实例并提交 Amazon 支持案例来请求补救。收到您的支持案例后，我们将在完成补救后通知您，之后您便可以将实例重新启动到 UltraServer 容量块上。
- 对于 P6e-GB200 UltraServer 容量块，您必须在容量块结束时间前至少 60 分钟终止实例。
- 要购买和使用 Local Zones 中的容量块，您必须主动启用该本地区域。
- 每个容量块最多可以有 64 个实例，您最多可以跨容量块拥有 256 个实例。
- 您可以描述最快 30 分钟内即可启动的容量块产品。
- 容量块于协调世界时 (UTC) 上午 11:30 结束。
- 在容量块中运行的实例的终止过程从预留最后一天协调世界时（UTC）上午 11:00 开始。
- 容量块的预留起始时间最长为未来 8 周。
- 不允许取消容量块。
- UltraServer 容量块不能跨 Amazon 账户或在 Amazon 组织内部共享。
- 无法移动或拆分容量块。
- 只有 UltraServer 容量块可与资源组一起使用。实例容量块不能与资源组一起使用。有关更多信息，请参阅 为 UltraServer 容量块创建资源组。
- 在特定日期，Amazon 组织中所有账户的容量块中可预留的实例总数不能超过 256 个。

- 要使用容量块，实例必须具体定位预留 ID。
- 容量块中的实例不计入按需型实例限制。
- 对于使用自定义 AMI 的 P5 实例，请确保拥有 EFA 所需的软件和配置。
- 对于 Amazon EKS 托管式节点组，请参阅创建带适用于 ML 的 Amazon EC2 容量块的托管式节点组。对于 Amazon EKS 自行管理的节点组，请参阅将适用于机器学习的容量块与自行管理的节点配合使用。

相关资源

创建容量块后，您可以使用容量块执行以下操作：

- 在容量块中启动实例。有关更多信息，请参阅 使用容量块启动实例。
- 创建 Amazon EC2 自动扩缩组。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的将容量块用于机器学习工作负载。

> **Note**
>
> 如果使用 Amazon EC2 Auto Scaling 或 Amazon EKS，则您可以将扩展计划为在容量块预留开始时运行。计划扩展时，Amazon 会自动为您处理重试，让您无需担心实现重试逻辑来处理暂时性故障。

- 使用 Amazon 并行计算服务增强 ML 工作流。有关更多信息，请参阅 Capacity Blocks support for Amazon Parallel Computing Service。
- 使用 Amazon ParallelCluster 增强 ML 工作流。有关更多信息，请参阅 Enhancing ML workflows with Amazon ParallelCluster and Amazon EC2 Capacity Blocks for ML。

有关 Amazon 并行计算服务的更多信息，请参阅 What is Amazon Parallel Computing Service。

有关 Amazon ParallelCluster 的更多信息，请参阅什么是 Amazon ParallelCluster。

Amazon EC2 容量块工作原理

您可以预留具有以下规格的容量块：

- 最早提前 8 周预留开始时间
- 设置预留时长，时长为 1 至 14 天或 7 天的倍数，最多 182 天（例如：21 天、28 天）
- 每个容量块最多可配置 64 个实例

- 多个容量块最多可配置 256 个实例

对于 Amazon EC2 UltraServers，每个 UltraServer 对应一个容量块。您可以通过一次请求来请求多个 UltraServer。

您可以使用容量块来预留 p6-b200、p5、p5e、p5en、p4d、p4de、trn1 和 trn2 实例。您可以通过容量块购买以下 UltraServer 类型：P6e-GB200 和 Trn2（预览版）。

要预留容量块，首先要指定容量需求，包括所需的实例类型或 UltraServer 类型、实例或 UltraServer 数量、时间、最早开始日期和最晚结束日期。然后，您可以看到符合您要求的可用容量块产品。容量块产品包括开始时间、可用区和预留价格等详细信息。容量块产品的价格取决于产品交付时的供需情况。预留容量块后，价格不会改变。有关更多信息，请参阅 容量块定价和计费。

当您购买容量块产品时，系统会根据您选择的日期和实例数创建预留。当容量块预留开始时，您可以通过在启动请求中指定预留 ID 来定位实例启动。

您可以在容量块结束时间前 30 分钟（适用于实例类型）或 60 分钟（适用于 UltraServer 类型）之前使用您预留的所有实例。在您的容量块预留还剩 30 分钟（适用于实例类型）或 60 分钟（适用于 UltraServer 类型）时，我们将开始终止在容量块中运行的所有实例。我们会利用这段时间清理您的实例，然后将容量块交付给下一个客户。系统将在终止过程开始前 10 分钟通过 EventBridge 发布一个事件。有关更多信息，请参阅 使用 EventBridge 监控容量块。

容量块定价和计费

使用适用于 ML 的 Amazon EC2 容量块，您只需为预留的容量付费。容量块的价格取决于购买时容量块的供需情况。在预留之前，您可以查看容量块产品的价格。容量块是在预留时预先收费的。当您搜索某个日期范围内的容量块时，系统会返回价格最低的可用容量块产品。预留容量块后，价格不会改变。

使用容量块时，您需要为实例运行时使用的操作系统付费。有关操作系统价格的更多信息，请参阅适用于 ML 的 Amazon EC2 容量块定价。

计费

容量块产品是预先收费的。在您购买容量块后 5 分钟到 12 小时内，将向您的 Amazon 账户发送付款账单。在处理您的付款时，您的容量块预留资源仍处于 payment-pending 状态。如果无法在您的块开始时间前至少 5 分钟或 12 小时内（以先到者为准）处理您的付款，则您的容量块将被释放，并且预留状态将更改为 payment-failed。

成功处理您的付款后，容量块资源状态将从 payment-pending 变为 scheduled。您会收到一张反映一次性预付款的发票。在发票中，您可以将支付的金额与容量块预留 ID 关联。

当您的容量块预留开始时，您只需按照实例在预留中运行时使用的操作系统付费。您可以在年度账单中查看 Amazon 成本和使用情况报告 中的当月使用情况和相关费用。

> **Note**
>
> 节省计划和预留实例折扣不适用于容量块。

查看您的账单

您可以在 Amazon 账单与成本管理 控制台中查看账单。容量块的预付款显示在您购买预留的月份当中。

预留开始后，您的账单会单独显示一行，显示块预留的已使用和未使用时间。您可以使用这些行项目来查看您的预留使用了多少时间。如果您使用的是高级操作系统，则只会在已使用时间行中看到使用费用。有关更多信息，请参阅 容量块定价和计费。未使用时间无需额外付费。

有关更多信息，请参阅 Amazon 账单与成本管理 用户指南中的查看您的账单。

如果您的容量块的起始月份与您购买预留的月份不同，则预付价格和预留使用情况会显示在单独的账单月份下。在您的 Amazon 成本和使用情况报告 中，容量块预留 ID 列在预付费用的 reservation/ReservationARN 行项目以及年度账单的 lineitem/ResourceID 中，以便您可以将使用情况与相应的预付费用相关联。

查找和购买容量块

要预留容量块，您首先需要找到符合您需求的可用容量的时间段。要查找可供预留的容量块，请指定以下内容。

- 您需要的实例数
- 您需要的实例持续时间
- 您需要预留的日期范围

要搜索可用的容量块产品，您可以指定预留持续时间和实例数量。您指定预留持续时间必须以 1 天为增量，最多 14 天；以 7 天为增量，最多 182 天。每个容量块最多可以有 64 个实例，您最多可以跨容量块拥有 256 个实例。

当您请求符合您规格的容量块时，我们会提供最多 6 个可用块的详细信息。所有容量块均于 UTC 时间上午 11:30 结束，因此同一天开始的块的持续时间将与您所需的持续时间最接近。一个块的持续时间将略小于您所需的持续时间，而另一个块的持续时间将略大于您所需的持续时间。

产品详细信息包括预留开始时间、预留可用区和预留价格。有关更多信息，请参阅 容量块定价和计费。

您可以购买所显示的容量块产品，也可以修改搜索条件以查看其他可用选项。该产品没有预定义的到期时间，但产品仅按先到先得的原则进行提供。

当您购买容量块产品时，您会立即收到回复，确认已预留您的容量块。确认后，您将在账户中看到一个新的容量预留，其预留类型为 `capacity-block`，并且 `start-date` 设置为所购买产品的开始时间。您的容量块预留创建时的状态为 `payment-pending`。成功处理预付款后，预留状态变为 `scheduled`。有关更多信息，请参阅 计费。

> **Note**
>
> 要购买和使用 Local Zones 中的容量块，您必须主动启用该本地区域。

Console

查找和购买容量块

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，选择一个 Amazon Web Services 区域。此选项很重要，因为并非所有区域中的所有实例类型都支持 64 个实例的容量块大小。
3. 在导航窗格中，依次选择容量预留、创建容量块。
4. 在容量块类型下，选择实例或 UltraServer。
5. 在容量属性下，您可以定义容量块搜索参数。默认情况下，平台为 Linux。如果要选择其他操作系统，请使用 Amazon CLI。有关更多信息，请参阅 支持的平台。
6. 在总容量（适用于实例）或 UltraServer 计数（适用于 UltraServer）下，指定要预留的实例或 UltraServer 的数量。
7. 在持续时间下，输入您需要预留的天数或周数。
8. 在容量块的搜索日期范围下，输入预留的最早开始日期。
9. 选择查找容量块。
10. 如果有符合您要求的容量块，您将在推荐的容量块下方看到产品。如果有多个产品符合要求，则会显示开始日期最早的可用容量块产品。要查看其他容量块产品，请调整搜索输入，然后再次选择查找容量块。
11. 当您找到要购买的容量块产品时，选择下一步。

12. （可选）在添加标签页面上，选择添加新标签。
13. 查看和购买页面列出了开始和结束日期、持续时间、实例总数和价格。

> **Note**
>
> 预留后，无法取消容量块。

14. 在购买容量块弹出窗口中，键入确认，然后选择购买。

Amazon CLI

查找实例容量块

使用 describe-capacity-block-offerings 命令。

以下示例查找实例容量块。

```
aws ec2 describe-capacity-block-offerings \
--instance-type p5.48xlarge \
--instance-count 16 \
--start-date-range 2023-08-14T00:00:00Z \
--end-date-range 2023-10-22-T00:00:00Z \
--capacity-duration-hours 48 \
--all-availability-zones
```

以下示例查找 UltraServer 容量块。

```
aws ec2 describe-capacity-block-offerings \
--ultraserver-type u-p6e-gb200x72 \
--ultraserver-count 1 \
--start-date-range 2023-08-14T00:00:00Z \
--end-date-range 2023-10-22-T00:00:00Z \
--capacity-duration-hours 48
```

购买容量块

将 purchase-capacity-block 命令与上一个示例输出中容量块的产品 ID 结合使用。

```
aws ec2 purchase-capacity-block \
--capacity-block-offering-id cb-0123456789abcdefg \
--instance-platform Linux/UNIX
```

PowerShell

查找容量块

使用 Get-EC2CapacityBlockOffering cmdlet。

以下示例查找实例容量块。

```
Get-EC2CapacityBlockOffering `
-InstanceType p5.48xlarge `
-InstanceCount 16 `
-CapacityDurationHour 48 `
-StartDateRange 2023-08-14T00:00:00Z `
-EndDateRange 2023-10-22-T00:00:00Z `
-AllAvailabilityZones $true
```

以下示例查找 UltraServer 容量块。

```
Get-EC2CapacityBlockOffering `
-UltraserverType u-p6e-gb200x72 `
-UltraserverCount  1 `
-CapacityDurationHour 48 `
-StartDateRange 2023-08-14T00:00:00Z `
-EndDateRange 2023-10-22-T00:00:00Z
```

购买容量块

将 New-EC2EC2CapacityBlock cmdlet 与上一个示例输出中容量块的产品 ID 结合使用。

```
New-EC2EC2CapacityBlock `
-CapacityBlockOfferingId cb-0123456789abcdefg `
-InstancePlatform Linux/UNIX
```

使用容量块启动实例

要使用容量块，您必须在启动实例时指定容量块预留 ID。在容量块中启动实例后，可用容量会减去已启动实例的数量。例如，如果您购买的实例容量为 8 个实例，而您启动了 4 个实例，则可用容量将减去 4。

如果您在预留结束之前终止在容量块中运行的实例，则可以在其位置启动新实例。当您停止或终止容量块中的实例时，系统需要几分钟的时间来清理实例，然后才能启动另一个实例来替换它。在此期

间，您的实例将处于停止或 shutting-down 状态。此过程完成后，实例状态将变为 stopped 或 terminated。然后，容量块中的可用容量将会更新，以显示另一个可用实例。

要求

- 您的实例无法在与您的容量块所在可用区不同的可用区的子网中启动。
- 实例无法使用其平台与容量块上的平台不同的 AMI 启动。
- 要使用 Local Zones 中的容量块，您必须主动启用该本地区域。

Console

要将实例启动到容量块中

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，选择容量块预留的区域。
3. 从 Amazon EC2 控制台控制面板中，选择启动实例。
4. 按照程序启动实例。
5. 展开高级详细信息，然后在购买选项中选择容量块。然后，执行以下操作之一：
   - 要将实例启动到特定的容量块，请在容量预留中选择指定容量预留，然后选择容量块。
   - （仅限 UltraServer ）要将实例启动到 UltraServer 容量块资源组中，请在容量预留中选择指定容量预留资源组，然后选择相应的资源组。
6. 选择启动实例。

Amazon CLI

要将实例启动到容量块中

使用 run-instances 命令和 instance-market-options MarketType 选项。

以下示例将实例启动到特定的容量块中。

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--count 1 \
--instance-type p5.48xlarge \
--key-name my-key-pair \
--subnet-id subnet-0abcdef1234567890 \
```

```
--instance-market-options MarketType='capacity-block' \
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationId=capacity_block_id}
```

以下示例将实例启动到 UltraServer 容量块资源组中。

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--count 1 \
--instance-type p6e-gb200.36xlarge \
--key-name my-key-pair \
--subnet-id subnet-0abcdef1234567890 \
--instance-market-options MarketType='capacity-block' \
--capacity-reservation-specification
 CapacityReservationTarget={CapacityReservationResourceGroupArn=resource_group_arn}
```

PowerShell

要将实例启动到容量块中

使用 New-EC2Instance cmdlet，其 `-InstanceMarketOption` 选项定义如下。

```
$marketoption = New-Object Amazon.EC2.Model.InstanceMarketOptionsRequest
$marketoption.MarketType = "capacity-block"
```

以下示例将实例启动到特定的容量块中。

```
New-EC2Instance `
-ImageId ami-0abcdef1234567890 `
-InstanceType p5.48xlarge `
-KeyName "my-key-pair" `
-SubnetId subnet-0abcdef1234567890 `
-InstanceMarketOptions $marketoption `
-CapacityReservationTarget_CapacityReservationId capacity_block_id
```

以下示例将实例启动到 UltraServer 容量块资源组中。

```
New-EC2Instance `
-ImageId ami-0abcdef1234567890 `
-InstanceType p6e-gb200.36xlarge `
-KeyName "my-key-pair" `
```

```
-SubnetId subnet-0abcdef1234567890 `
-InstanceMarketOptions $marketoption `
-CapacityReservationTarget_CapacityReservationResourceGroupArn "resource_group_arn"
```

相关资源

- 要创建将容量块作为目标的启动模板，请参阅the section called “启动模板”。
- 要使用 EC2 实例集在容量块中启动实例，请参阅教程：将 EC2 实例集配置为将实例启动到容量块中。
- 要设置带容量块的 EKS 托管式节点组，请参阅《Amazon EKS 用户指南》中的创建带适用于机器学习的容量块的托管式节点组。
- 要使用容量块设置 Amazon ParallelCluster，请参阅 ML on Amazon ParallelCluster。

查看容量块

预留容量块后，您可以在 Amazon 账户中查看容量块预留。您可以查看 `start-date` 和 `end-date`，了解您的预留何时开始和结束。在容量块预留开始之前，可用容量显示为零。您可以通过标签键 `aws:ec2capacityreservation:incrementalRequestedQuantity` 的标签值查看容量块中有多少可用实例。

当容量块预留开始时，预留状态从 `scheduled` 变为 `active`。系统通过 Amazon EventBridge 发出一个事件，通知您容量块可以使用。有关更多信息，请参阅 使用 EventBridge 监控容量块。

容量块具有以下状态：

- `payment-pending` - 预付款尚未处理。
- `payment-failed` - 无法在 12 小时内处理付款。您的容量块已释放。
- `scheduled` - 付款已处理，但容量块预留尚未开始。
- `active` - 预留容量可供使用。
- `expired` - 容量块预留将在您的预留请求中指定的日期和时间自动到期。预留容量不再可供您使用。

Console

查看容量块

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择容量预留。
3. 在容量预留概述页面上，您会看到一个资源表，其中包含有关所有容量预留资源的详细信息。要查找您的容量块预留，请从容量预留 ID 上方的下拉列表中选择容量块。在表中，您可以看到有关容量块的信息，比如开始和结束日期、持续时间和状态。
4. 有关容量块的更多详细信息，请选择要查看的容量块的预留 ID。容量预留详细信息页面将显示预留的所有属性以及容量块中正在使用和可用的实例数。

> **Note**
>
> 在容量块预留开始之前，可用容量显示为零。您可以通过使用标签键的以下标签值查看容量块预留开始时有多少可用实例：`aws:ec2capacityreservation:incrementalRequestedQuantity`。

Amazon CLI

查看容量块

默认情况下，当您使用 describe-capacity-reservations 命令时，会列出按需容量预留和容量块预留。要仅查看容量块预留，可通过 `capacity-block` 预留类型进行筛选。

```
aws ec2 describe-capacity-reservations \
    --filters Name=reservation-type,Values=capacity-block
```

PowerShell

查看容量块

使用 Get-EC2CapacityReservation cmdlet。默认情况下，会列出按需容量预留和容量块预留。要仅查看容量块预留，可通过 `capacity-block` 预留类型进行筛选。

```
Get-EC2CapacityReservation `
    -Filter @{Name="reservation-type"; Values="capacity-block"}
```

延长容量块期限

使用容量块，您可以为工作负载预留计算容量，以确保可用性和一致性。为了满足您不断变化的需求，您可以根据需要延长现有容量块的持续时间。

要延长容量块，其状态必须为 `active` 或 `scheduled`，并且没有处于 `payment-pending` 状态的延期。您可以在容量块过期前，申请将其期限延长至最少 1 小时或最多 57 天。您可以将容量块以 1 天为增量延长至总计最长 14 天，或者以 7 天为增量延长至总计最长 182 天（26 周）。当您延长容量块期限时，其结束日期将会更新，以便您的实例可以继续运行而不中断。

- 可以应用于容量块的延期数量没有限制
- 延长块期限后，您的容量预留 ID 将保持不变
- 只有当具有足够的可用容量来支持容量块时，才能对其进行延期，但这并不能保证。

计费

容量块产品是预先收费的。延期将保持 `payment-pending` 状态，直到账单付清为止。如果您的付款无法在 12 小时内或容量块预计结束前的 35 分钟内处理（以先到者为准），则您的延期将失败，并且状态将更改为 `payment-failed`。您的容量块预留将保持 `active` 状态，并将于原始结束日期终止。

成功处理您的付款后，容量块延期状态将更改为 `payment-succeeded`，并且容量块预留的结束日期将更新为新的结束日期。延期的详细信息可以在控制台的容量块延期详细信息部分中查看，也可以使用 describe-capacity-block-extension-history 命令查看。

延长您的容量块期限

Console

延长容量块期限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择容量预留。
3. 在容量预留概述页面上，您会看到一个资源表，其中包含有关所有容量预留资源的详细信息。选择要延期的容量块的预留 ID。
4. 从操作下拉菜单中，选择延长容量块期限。
5. 在持续时间下，输入您需要延长预留的天数或周数。
6. 选择查找容量块。
7. 如果有符合您规格的容量块，则会在推荐的容量块下显示一个产品。要查看其他容量块产品，请调整搜索输入，然后再次选择查找容量块。
8. 当您找到要购买的容量块产品时，请选择延期。
9. 在延长容量块期限弹出窗口中，输入确认，然后选择延期。

Amazon CLI

查找容量块延期

使用 describe-capacity-block-extension-offerings 命令。以下示例搜索指定预留的 48 小时容量块延期。

```
aws ec2 describe-capacity-block-extension-offerings \
    --capacity-reservation-id cr-1234567890abcdefg \
    --capacity-block-extension-duration-hours 48
```

延长容量块期限

使用 purchase-capacity-block-extension 命令。从上一个示例的输出中指定延期产品 ID。

```
aws ec2 purchase-capacity-block-extension \
    --capacity-block-extension-offering-id cbe-0123456789abcdefg \
    --capacity-reservation-id cr-1234567890abcdefg
```

PowerShell

查找容量块延期

使用 Get-EC2CapacityBlockExtensionOffering cmdlet。以下示例搜索指定预留的 48 小时容量块延期。

```
Get-EC2CapacityBlockExtensionOffering `
    -CapacityReservationId cr-1234567890abcdefg `
    -CapacityBlockExtensionDurationHour 48
```

延长容量块期限

使用 Invoke-EC2CapacityBlockExtension cmdlet。从上一个示例的输出中指定延期产品 ID。

```
Invoke-EC2CapacityBlockExtension `
    -CapacityBlockExtensionOfferingId cbe-0123456789abcdefg `
    -CapacityReservationId cr-1234567890abcdefg
```

共享容量块

容量块共享使容量块拥有者能够与 Amazon 组织内的其他 Amazon 账户共享 Amazon EC2 容量块。这样，您便可以最大限度地利用不同团队和项目的预留 GPU 容量，从而有效地使用容量块。

拥有容量块的 Amazon 账户（拥有者）可以与其他 Amazon 账户（使用者）共享它。拥有者可以与其 Amazon 组织内的特定 Amazon 账户、其 Amazon 组织内的组织单位或整个 Amazon 组织共享容量块。使用者可以像在自己拥有的容量块中启动实例一样，在与他们共享的容量块中启动实例。

共享容量块的先决条件

在共享容量块之前，必须满足下面的条件：

- 您必须拥有容量块：您无法共享已与您共享的容量块。
- 容量块状态必须为活跃或已计划：处于其他状态（例如 `expired` 或`payment-pending`）的容量块无法共享。
- 仅在您的 Amazon 组织内共享：拥有者可以与其 Amazon 组织内的特定 Amazon 账户、其 Amazon 组织内的组织单位或整个 Amazon 组织共享容量块。
- 不支持 UltraServer 容量块：您无法共享 Amazon EC2 UltraServers 的容量块。
- 账户资格：容量块共享不适用于新的 Amazon 账户或具有有限账单历史记录的 Amazon 账户。

相关服务

容量块共享与 Amazon Resource Access Manager (Amazon RAM) 集成。Amazon RAM 是一项服务，允许您与任何 Amazon 账户或通过 Amazon Organizations 共享 Amazon 资源。利用 Amazon RAM，您可通过创建资源共享来共享您拥有的资源。资源共享指定要共享的资源以及与之共享资源的使用者。使用者可以是单个 Amazon 账户或 Amazon Organizations 中的组织部门或整个组织。

Amazon RAM有关 的更多信息，请参阅 Amazon RAM 用户指南。

共享容量块权限

拥有者的权限

容量块拥有者仍然负责管理容量块（如扩展、共享）以及在其中启动的实例。拥有者无法修改使用者启动到已共享的容量块中的实例。

使用者的权限

使用者可以在共享容量中启动实例，并负责管理这些实例。使用者无法查看或修改由其他使用者或容量块拥有者拥有的实例。使用者还只能查看共享容量块中的总容量和可用容量。

共享容量块

要共享容量块，您必须将它添加到资源共享。资源共享是一项 Amazon RAM 资源，可让您跨 Amazon Web Services 账户共享资源。

如果您已将容量块添加到与整个 Amazon 组织共享的资源共享，则组织中的使用者将能够访问共享的容量块。

Console

要使用 Amazon EC2 控制台共享您拥有的容量块

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择容量预留。
3. 选择要共享的容量块，然后选择操作、共享预留。
4. 选择要将容量块添加到的资源共享，然后选择共享容量预留。

使用者可能需要几分钟的时间才能访问共享容量块。

要将容量块添加到新的资源共享中

必须使用 Amazon RAM 控制台先创建资源共享。有关更多信息，请参阅《Amazon RAM User Guide》中的 Creating a resource share。

Amazon CLI

共享您拥有的容量块

使用 create-resource-share 和 associate-resource-share 命令。

```
aws ram create-resource-share \
    --name my-resource-share \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

```
aws ram associate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE
```

PowerShell

共享您拥有的容量块

使用 New-RAMResourceShare 和 Connect-RAMResourceShare cmdlet。

```
New-RAMResourceShare `
    -Name my-resource-share `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

```
Connect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-
reservation/cr-1234abcd56EXAMPLE"
```

对于所有账户，无论所有权状态如何，容量块都按先到先得的原则运行。当您共享容量块时，如果使用者在拥有者之前启动实例，则这些实例会占用容量，直到使用者终止实例或直到容量块到期前 30 分钟。

停止共享容量块

您可以在容量块到期前 30 分钟随时停止共享容量块。

停止共享时会发生什么情况：

- 使用者无法再在未共享的容量块中启动新实例。
- 除非使用者终止，否则任何正在运行的实例都会持续运行到容量块到期日前 30 分钟。

Console

要使用 Amazon EC2 控制台停止共享您拥有的容量块

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择容量预留。
3. 选择容量块并选择共享选项卡。
4. 共享选项卡列出了已将容量块添加到的资源共享。选择要从中移除容量块的资源共享。
5. 选择从资源共享中移除。

Amazon CLI

停止共享您拥有的容量块

使用 disassociate-resource-share 命令。

```
aws ram disassociate-resource-share \
    --resource-share-arn arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE
```

PowerShell

停止共享您拥有的容量块

使用 Disconnect-RAMResourceShare cmdlet。

```
Disconnect-RAMResourceShare `
    -ResourceShareArn "arn:aws:ram:us-east-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE" `
    -ResourceArn "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-1234abcd56EXAMPLE"
```

监控共享容量块使用情况

容量块拥有者可以监控哪些账户正在使用其共享容量块，并跟踪每个账户的实例使用情况。

Amazon CLI

监控容量块的使用情况

使用 get-capacity-reservation-usage 命令。

```
aws ec2 get-capacity-reservation-usage \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

此 API 使拥有者能够：

- 查看哪些账户当前正在使用容量块。
- 查看每个账户正在运行的实例数。

实例终止通知

在容量块中运行实例的拥有者和使用者账户将在容量块预留结束前 40 分钟收到一个 EventBridge 事件，指明预留中运行的所有实例将在 10 分钟后开始终止。有关更多信息，请参阅 使用 EventBridge 监控容量块。

容量块扩展

容量块可以在共享时进行扩展。只有拥有者账户才能扩展共享容量区块。

扩展容量块后，拥有者或使用者启动的正在运行的实例会自动继承新的到期日期，使用者可以继续使用共享容量，直到新的到期日，而不会中断任何实例。

定价和计费

拥有者需要为他们共享的容量块付费，在购买容量块时需预先支付容量块的费用。拥有者还需为他们在容量块上运行的实例支付操作系统费用。

使用者只需为他们在共享容量块中运行的实例支付操作系统费用。使用者无需为容量块预留本身付费。

为 UltraServer 容量块创建资源组

您可以使用 Amazon Resource Groups 创建 UltraServer 容量块的逻辑集合。创建资源组后，您可以添加您账户中拥有的 UltraServer 容量块。添加 UltraServer 容量块后，您可以将实例启动定位到资源组，而不是单个容量块。针对资源组的实例与组中具有匹配属性和可用容量的任何 UltraServer 容量块匹配。如果资源组没有具有匹配属性和可用容量的 UltraServer 容量块，则实例启动失败。

如果在 UltraServer 容量块有正在运行的实例时将其从资源组中删除，则这些实例将继续在容量块中运行。如果组中的 UltraServer 容量块在具有正在运行的实例时结束，则实例将终止。

您无法将实例容量块添加到资源组。

要为 UltraServer 容量块创建资源组，请使用以下方法之一。

Amazon CLI

要为 UltraServer 容量块创建资源组

使用 create-group Amazon CLI 命令，并为 `--configuration` 指定以下内容：

```
{
  "Configuration": [
```

```
    {
      "Type": "AWS::EC2::CapacityReservationPool",
      "Parameters": [
        {
          "Name": "instance-type",
          "Values": [
            "instance_type"
          ]
        },
        {
          "Name": "reservation-type",
          "Values": [
            "capacity-block"
          ]
        }
      ]
    },
    {
      "Type": "AWS::ResourceGroups::Generic",
      "Parameters": [
        {
          "Name": "allowed-resource-types",
          "Values": [
            "AWS::EC2::CapacityReservation"
          ]
        }
      ]
    }
  ]
}
```

PowerShell

要为 UltraServer 容量块创建资源组

使用 New-RGGroup cmdlet。对于 `-Configuration`，请指定以下内容：

```
{
  "Configuration": [
    {
      "Type": "AWS::EC2::CapacityReservationPool",
      "Parameters": [
        {
          "Name": "instance-type",
```

```
          "Values": [
            "instance_type"
          ]
        },
        {
          "Name": "reservation-type",
          "Values": [
            "capacity-block"
          ]
        }
      ]
    },
    {
      "Type": "AWS::ResourceGroups::Generic",
      "Parameters": [
        {
          "Name": "allowed-resource-types",
          "Values": [
            "AWS::EC2::CapacityReservation"
          ]
        }
      ]
    }
  ]
}
```

为 UltraServer 容量块创建资源组后，使用以下方法之一将现有的 UltraServer 容量块添加到其中。

Amazon CLI

要将 UltraServer 容量块添加到资源组

使用 group-resources 命令。对于 `--group`，请指定您创建的资源组的名称。对于 `--resource-arns`，请指定要添加的 UltraServer 容量块的 ARN。

```
aws resource-groups group-resources \
--group MyCRGroup \
--resource-arns CapacityReservationArn
```

PowerShell

要将 UltraServer 容量块添加到资源组

使用 Add-RGResource cmdlet。对于 `-Group`，请指定您创建的资源组的名称。对于 `-ResourceArn`，请指定要添加的 UltraServer 容量块的 ARN。

以下示例向指定组添加两个容量预留。

```
Add-RGResource `
-Group MyCRGroup `
-ResourceArn CapacityReservationArn
```

使用 EventBridge 监控容量块

当您的容量块预留开始时，Amazon EC2 将通过 EventBridge 发出一个事件，表明您的容量已准备就绪，可以使用。在容量块预留结束前 40 分钟，您会收到另一个 EventBridge 事件，通知您预留中运行的所有实例将在 10 分钟后开始进行终止。有关 EventBridge 事件的更多信息，请参阅 Amazon EventBridge 事件。

容量块发出的事件结构如下：

已交付的容量块

以下示例显示了已交付的容量块的事件。

```
{
  "customer_event_id": "[Capacity Reservation Id]-delivered",
  "detail_type": "Capacity Block Reservation Delivered",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

容量块过期警告

以下示例显示了容量块过期警告事件。

```
{
  "customer_event_id": "[Capacity Reservation Id]-approaching-expiry",
  "detail_type": "Capacity Block Reservation Expiration Warning",
  "source": "aws.ec2",
  "account": "[Customer Account ID]",
  "time": "[Current time]",
  "resources": [
    "[ODCR ARN]"
  ],
  "detail": {
    "capacity-reservation-id": "[ODCR ID]",
    "end-date": "[ODCR End Date]"
  }
}
```

容量预留实例中断警告

下面的示例显示了 EC2 容量预留实例中断警告事件。

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail_type": "EC2 Capacity Reservation Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "[Customer Account ID]",
    "time": "[Current time]",
    "region": "[Region]",
    "resources": [
        "[Instance ARN]"
    ],
    "detail": {
        "instance-id": "[Instance ID]",
        "instance-action": "terminate",
        "instance-termination-time": "[Current time]",
        "availability-zone-id": "[Availability Zone ID]",
        "instance-lifecycle": "capacity-block"
    }
}
```

使用 Amazon CloudTrail 记录容量块 API 调用

容量块与 Amazon CloudTrail 集成，后者是一种服务，提供用户、角色或 Amazon 服务在容量块中采取的操作的记录。CloudTrail 将容量块的 API 调用作为事件捕获。捕获的调用包含来自容量块控制

台的调用以及对容量块 API 操作的代码调用。如果您创建跟踪，则可以将 CloudTrail 事件持续交付到 Amazon S3 存储桶，包括容量块事件。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的事件历史记录中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向容量块发出的请求、发出请求的 IP 地址、请求方、请求时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参阅《Amazon CloudTrail 用户指南》。

CloudTrail 中的容量块信息

在您创建 Amazon Web Services 账户 时，将在该账户上启用 CloudTrail。当容量块中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 Amazon 服务事件一同记录在事件历史记录中。您可以在 Amazon Web Services 账户 中查看、搜索和下载最新事件。有关更多信息，请参阅使用 CloudTrail 事件历史记录查看事件。

要持续记录 Amazon Web Services 账户 中的事件（包括容量块事件），请创建跟踪。通过跟踪，CloudTrail 可将日志文件传送至 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 Amazon Web Services 区域。此跟踪记录在 Amazon 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

CloudTrail 会记录所有容量块操作，并记录在 Amazon EC2 API 参考中。例如，对 `CapacityBlockScheduled` 和 `CapacityBlockActive` 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 Amazon Identity and Access Management（IAM）用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 Amazon 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解容量块日志文件条目

跟踪记录是一种配置，可用于将事件作为日志文件传送到您指定的 Simple Storage Service（Amazon S3）存储桶。CloudTrail 日志文件包含一个或多个记录条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公有 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

以下示例显示了 CloudTrail 日志条目：

- TerminateCapacityBlocksInstances
- CapacityBlockPaymentFailed
- CapacityBlockScheduled
- CapacityBlockActive
- CapacityBlockFailed
- CapacityBlockExpired

> **Note**
>
> 为了保护数据隐私，已从示例中删除了一些字段。

TerminateCapacityBlocksInstances


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "Amazon Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "TerminateCapacityBlockInstances",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
```

```
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
    }
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Instance",
      "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      }
}
```

CapacityBlockPaymentFailed

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "Amazon Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockPaymentFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
```

```
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "payment-failed"
      }
}
```

CapacityBlockScheduled

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "Amazon Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockScheduled",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "scheduled"
      }
}
```

CapacityBlockActive


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "Amazon Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockActive",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "active"
      }
 }
```


CapacityBlockFailed


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "Amazon Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
```

```
  "eventName": "CapacityBlockFailed",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "failed"
      }
 }
```


CapacityBlockExpired


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "Amazon Internal;"
  },
  "eventTime": "2023-10-02T00:06:08Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "CapacityBlockExpired",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.25",
  "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a1b2c3d4-EXAMPLE",
  "readOnly": false,
  "resources": [
    {
```

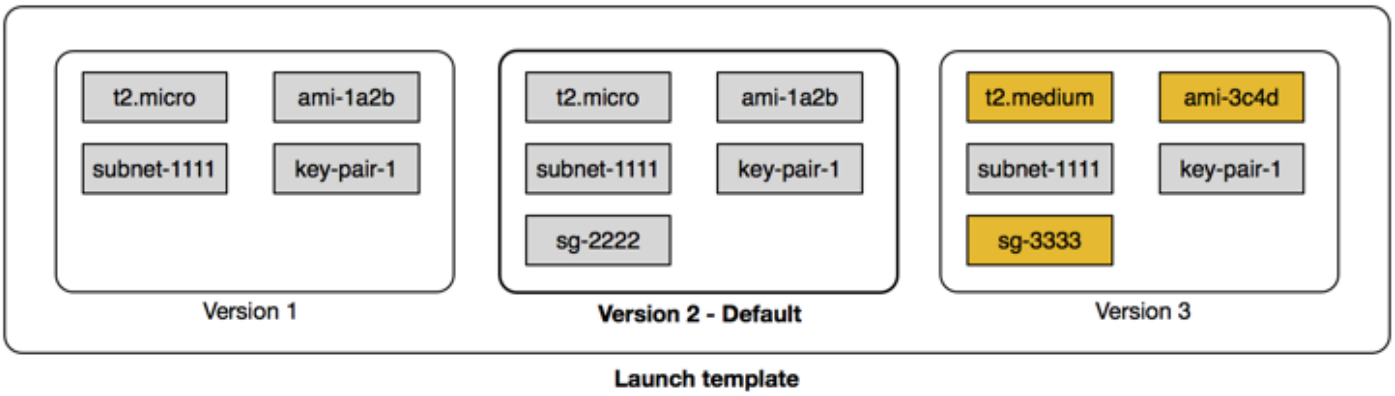
```
      "ARN": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-12345678",
      "accountId": "123456789012",
      "type": "AWS::EC2::CapacityReservation"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
      "capacityReservationId": "cr-12345678",
      "capacityReservationState": "expired"
      }
}
```

在 Amazon EC2 启动模板中存储实例启动参数

您可以使用 Amazon EC2 启动模板来存储实例启动参数，以便无需在每次启动 Amazon EC2 实例时都指定这些参数。例如，您可以创建一个启动模板，其中存储您通常用于启动实例的 AMI ID、实例类型和网络设置。在使用 Amazon EC2 控制台、Amazon SDK 或命令行工具启动实例时，您可以指定要使用的启动模板，而无需再次输入参数。

对于每个启动模板，您可以创建一个或多个编号的启动模板版本。每个版本可能具有不同的启动参数。在通过启动模板启动实例时，您可以使用任何版本的启动模板。如果未指定版本，则使用默认版本。您可以将任何启动模板版本设置为默认版本 — 默认情况下，这是启动模板的第一个版本。

下图显示了具有三个版本的启动模板。第一个版本指定用于启动实例的实例类型、AMI ID、子网和密钥对。第二个版本基于第一个版本，并且还为实例指定了一个安全组。第三个版本在某些参数中使用不同的值。版本 2 设置为默认版本。如果通过该启动模板启动实例，并且未指定任何其他版本，则使用版本 2 中的启动参数。



内容

- Amazon EC2 启动模板的限制
- Amazon EC2 启动模板所需的 IAM 权限
- 使用 Amazon EC2 启动模板控制启动 Amazon EC2 实例
- 创建 Amazon EC2 启动模板
- 修改启动模板（管理启动模板版本）
- 删除启动模板或启动模板版本

Amazon EC2 启动模板的限制

以下限制适用于启动模板和启动模板版本：

- 限额 – 每个区域最多可以创建 5,000 个启动模板，每个启动模板最多可以有 10,000 个版本。
- 参数可选 – 启动模板参数是可选的。不过，您必须确保实例启动请求包含所需的所有参数。例如，如果启动模板不包含 AMI ID，您必须在使用此启动模板启动实例时指定 AMI ID。
- 参数未验证 – 创建启动模板时，不会完全验证启动模板参数。如果您指定了错误的值或使用不支持的参数组合，则实例将无法使用此启动模板启动。为避免出现问题，请确保指定正确的值，并使用支持的参数组合。例如，要在置放群组中启动实例，您必须指定一种支持的实例类型。
- 标签：您可以标记启动模板，但无法标记启动模板版本。
- 不可变 – 启动模板不可变。要修改启动模板，必须创建新版本启动模板。
- 版本号 – 启动模板版本按创建顺序编号。在创建启动模板版本时，您无法自行指定版本号。

Amazon EC2 启动模板所需的 IAM 权限

您可以使用 IAM 权限来控制用户是否可以列出、查看、创建或删除启动模板或启动模板版本。

> **Important**
>
> 您不能使用资源级权限来限制用户在创建启动模板或启动模板版本时可以在启动模板中指定的资源。因此，请确保仅授予受信任的管理员创建启动模板和启动模板版本的权限。

您必须向将使用启动模板的任何人授予创建和访问启动模板中所指定资源所需的权限。例如：

- 要从共享的私有亚马逊机器映像（AMI）启动实例，用户必须拥有 AMI 的启动权限。

- 要使用现有快照的标签创建 EBS 卷，用户必须具有快照的读取权限以及创建和标记卷的权限。

内容

- ec2:CreateLaunchTemplate
- ec2:DescribeLaunchTemplates
- ec2:DescribeLaunchTemplateVersions
- ec2:DeleteLaunchTemplate
- 控制版本控制权限
- 控制对启动模板上标签的访问权限

ec2:CreateLaunchTemplate

要在控制台中或使用 API 创建启动模板，主体必须具有 IAM policy 中的 `ec2:CreateLaunchTemplate` 权限。尽可能使用标签来帮助您控制对账户中启动模板的访问权限。

例如，以下 IAM policy 声明仅在模板使用指定标签（*purpose*=*testing*）时才向主体授予创建启动模板的权限。

```
{
    "Sid": "IAMPolicyForCreatingTaggedLaunchTemplates",
    "Action": "ec2:CreateLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
        }
    }
}
```

创建启动模板的主体可能需要一些相关权限，例如：

- ec2:CreateTags – 要在 `CreateLaunchTemplate` 操作期间将标签添加到启动模板，`CreateLaunchTemplate` 调用者必须具有 IAM policy 中的 `ec2:CreateTags` 权限。
- ec2:RunInstances – 要从其创建的启动模板启动 EC2 实例，主体还必须具有 IAM policy 中的 `ec2:RunInstances` 权限。

对于应用标签的资源创建操作，用户必须具有 ec2:CreateTags 权限。以下 IAM policy 语句使用 ec2:CreateAction 条件键，只允许用户在 CreateLaunchTemplate 上下文中创建标签。用户无法标记现有启动模板或任何其他资源。有关更多信息，请参阅 在创建过程中授予标记 Amazon EC2 资源的权限。

```
{
    "Sid": "IAMPolicyForTaggingLaunchTemplatesOnCreation",
    "Action": "ec2:CreateTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateLaunchTemplate"
        }
    }
}
```

创建启动模板的 IAM 用户不会自动获得使用其创建的启动模板的权限。与任何其他主体一样，启动模板创建者需要通过 IAM policy 获得权限。如果 IAM 用户想要从启动模板启动 EC2 实例，则他们必须拥有 ec2:RunInstances 权限。在授予这些权限时，您可以指定用户只能使用带有特定标签或特定 ID 的启动模板。您还可以通过指定 RunInstances 调用的资源级权限，控制 AMI 和使用启动模板的任何人都可以在启动实例时引用和使用的其他资源。有关示例策略，请参阅 启动模板。

ec2:DescribeLaunchTemplates

要列出和查看账户中的启动模板，主体必须在 IAM 策略中具有 ec2:DescribeLaunchTemplates 权限。由于 Describe 操作不支持资源级权限，因此必须无条件地指定这些权限，并且策略中资源元素的值必须为 "*"。

例如，以下 IAM 策略语句授予主体列出和查看账户中所有启动模板的权限。

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplates",
    "Action": "ec2:DescribeLaunchTemplates",
    "Effect": "Allow",
    "Resource": "*"
}
```

ec2:DescribeLaunchTemplateVersions

列出和查看启动模板的主体还应具有 ec2:DescribeLaunchTemplateVersions 权限以检索构成启动模板的整组属性。

要列出和查看账户中的启动模板版本，主体必须在 IAM 策略中具有 ec2:DescribeLaunchTemplateVersions 权限。由于 Describe 操作不支持资源级权限，因此必须无条件地指定这些权限，并且策略中资源元素的值必须为 "*"。

例如，以下 IAM 策略语句授予主体列出和查看账户中所有启动模板版本的权限。

```
{
    "Sid": "IAMPolicyForDescribingLaunchTemplateVersions",
    "Effect": "Allow",
    "Action": "ec2:DescribeLaunchTemplateVersions",
    "Resource": "*"
}
```

ec2:DeleteLaunchTemplate

> **Important**
>
> 在授予主体删除资源的权限时请谨慎操作。删除启动模板可能会导致依赖该启动模板的 Amazon 资源出现故障。

要删除启动模板，主体必须在 IAM policy 中具有 ec2:DeleteLaunchTemplate 权限。在可能的情况下，使用基于标签的条件键来限制权限。

例如，以下 IAM 策略语句仅在模板具有指定标签（*purpose*=*testing*）时才向主体授予删除启动模板的权限。

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplates",
    "Action": "ec2:DeleteLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/purpose": "testing"
```

```
        }
    }
}
```

或者，您可以使用 ARN 来识别 IAM policy 适用的启动模板。

启动模板具有以下 ARN。

```
"Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/lt-09477bcd97b0d310e"
```

您可以通过将多个 ARN 括在列表中来指定它们，也可以指定 Resource 值 "*"（不带 Condition 元素），以允许主体删除账户中的任何启动模板。

控制版本控制权限

对于受信任的管理员，您可以使用类似于以下示例的 IAM policy，授予创建和删除启动模板版本以及更改启动模板默认版本的访问权限。

> **Important**
>
> 授予主体创建启动模板版本或修改启动模板的权限时，请谨慎行事。
>
> - 创建启动模板版本时，会影响允许 Amazon EC2 代表您使用 Latest 版本启动实例的任何 Amazon 资源。
> - 修改启动模板版本时，可以更改哪个版本为 Default，并会因此影响允许 Amazon EC2 代表您使用此修改后的版本启动实例的任何 Amazon 资源。
>
> 您还需要谨慎处理与 Latest 或 Default 启动模板版本交互的 Amazon 资源，例如 EC2 实例集和竞价型实例集。将不同的启动模板版本用于 Latest 或 Default 时，Amazon EC2 不会在启动新实例以满足实例集的目标容量时重新检查要完成的操作的权限，因为没有用户与 Amazon 资源交互。向用户授予调用 CreateLaunchTemplateVersion 和 ModifyLaunchTemplate API 的权限，即是同时授予用户将实例集指向包含实例配置文件（IAM 角色的容器）的不同启动模板版本的 iam:PassRole 权限。这意味着即使用户没有 iam:PassRole 权限，也可以更新启动模板以将 IAM 角色传递给实例。您可以通过谨慎授予创建和管理启动模板版本的权限来管理这种风险。

ec2:CreateLaunchTemplateVersion

要创建启动模板的新版本，主体必须具有 IAM policy 中启动模板的 `ec2:CreateLaunchTemplateVersion` 权限。

例如，以下 IAM policy 声明仅在版本使用指定标签（*environment*=*production*）时才向主体授予创建启动模板版本的权限。或者，您可以指定一个或多个启动模板 ARN，也可以指定 `Resource` 的值为 `"*"`（不带 `Condition` 元素），以允许主体创建账户中任何启动模板的版本。

```
{
    "Sid": "IAMPolicyForCreatingLaunchTemplateVersions",
    "Action": "ec2:CreateLaunchTemplateVersion",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:DeleteLaunchTemplateVersion

> **Important**
>
> 与往常一样，在授予委托人删除资源的权限时，您应该谨慎操作。删除启动模板版本可能会导致依赖该启动模板版本的 Amazon 资源出现故障。

要删除启动模板版本，主体必须具有 IAM policy 中启动模板的 `ec2:DeleteLaunchTemplateVersion` 权限。

例如，以下 IAM policy 声明仅在版本使用指定标签（*environment*=*production*）时才向主体授予删除启动模板版本的权限。或者，您可以指定一个或多个启动模板 ARN，也可以指定 `Resource` 的值为 `"*"`（不带 `Condition` 元素），以允许主体删除账户中任何启动模板的版本。

```
{
    "Sid": "IAMPolicyForDeletingLaunchTemplateVersions",
    "Action": "ec2:DeleteLaunchTemplateVersion",
    "Effect": "Allow",
```

```
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

ec2:ModifyLaunchTemplate

要更改与启动模板关联的 `Default` 版本，主体必须具有 IAM policy 中启动模板的 `ec2:ModifyLaunchTemplate` 权限。

例如，以下 IAM policy 声明仅在启动模板使用指定标签（*environment*=*production*）时才向主体授予修改启动模板的权限。或者，您可以指定一个或多个启动模板 ARN，也可以指定 `Resource` 的值为 `"*"`（不带 `Condition` 元素），以允许主体修改账户中的任何启动模板。

```
{
    "Sid": "IAMPolicyForModifyingLaunchTemplates",
    "Action": "ec2:ModifyLaunchTemplate",
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/environment": "production"
        }
    }
}
```

控制对启动模板上标签的访问权限

当资源为启动模板时，您可以使用条件键限制标记权限。例如，以下 IAM policy 允许仅从指定账户和区域的启动模板中移除带有 *temporary* 键的标签。

```
{
    "Sid": "IAMPolicyForDeletingTagsOnLaunchTemplates",
    "Action": "ec2:DeleteTags",
    "Effect": "Allow",
    "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
    "Condition": {
        "ForAllValues:StringEquals": {
            "aws:TagKeys": ["temporary"]
```

```
        }
    }
}
```

有关可用于控制可应用于 Amazon EC2 资源的标签键和值的条件键的更多信息，请参阅 控制对特定标签的访问。

使用 Amazon EC2 启动模板控制启动 Amazon EC2 实例

通过指定用户仅在使用启动模板时才能启动实例，且只能使用特定的启动模板，您可以控制 Amazon EC2 实例的配置。您还可以控制哪些用户能创建、修改、描述和删除启动模板和启动模板版本。

使用启动模板控制启动参数

启动模板可以包含在启动时配置实例的全部或部分参数。不过，在使用启动模板启动实例时，您可以覆盖启动模板中指定的参数。或者，也可以指定在启动模板中不包含的额外参数。

> **Note**
>
> 您无法在启动期间删除启动模板参数（例如，无法为参数指定空值）。要删除某个参数，请创建不包含该参数的新启动模板版本，并使用该版本启动实例。

要启动实例，用户必须有权使用 `ec2:RunInstances` 操作。用户还必须有权创建或使用已创建或与该实例关联的资源。您可以使用 `ec2:RunInstances` 操作的资源级权限控制用户可以指定的启动参数。或者，您可以为用户授予使用启动模板启动实例的权限。这样，您就可以在启动模板中管理启动参数，而不是在 IAM policy 中管理，并将启动模板作为授权方法以启动实例。例如，您可以指定用户只能使用启动模板启动实例，并且他们只能使用特定的启动模板。您还可以控制用户可以在启动模板中覆盖的启动参数。有关示例策略，请参阅 启动模板。

控制启动模板的使用

默认情况下， 用户无权使用启动模板。您可以创建一个策略，以便为用户授予创建、修改、描述和删除启动模板和启动模板版本的权限。您还可以将资源级权限应用于某些启动模板操作，以控制用户能否在这些操作中使用特定的资源。有关更多信息，请参阅以下示例策略：示例：使用启动模板。

在为用户授予使用 `ec2:CreateLaunchTemplate` 和 `ec2:CreateLaunchTemplateVersion` 操作的权限时，应格外小心。您不能使用资源级权限来控制用户可以在启动模板中指定的资源。要限制用于启动实例的资源，请确保仅为相应的管理员授予创建启动模板和启动模板版本的权限。

对 EC2 实例集或竞价型实例集使用启动模板时的重要安全问题

> **Note**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

要使用启动模板，您必须授予创建、修改、描述和删除启动模板和启动模板版本的权限。您可以通过控制对 `ec2:CreateLaunchTemplate` 和 `ec2:CreateLaunchTemplateVersion` 操作的访问来控制谁可以创建启动模板和启动模板版本。您还可以通过控制 `ec2:ModifyLaunchTemplate` 操作的访问权限来控制哪些用户能修改启动模板。

> **Important**
>
> 如果 EC2 实例集或竞价型实例集配置为使用“最新”或“默认”启动模板版本，则实例集不了解稍后是否将“最新”或“默认”更改为指向不同的启动模板版本。将不同的启动模板版本用于“最新”或“默认”时，Amazon EC2 不会在启动新实例以满足实例集的目标容量时重新检查要完成的操作的权限。在向可以创建和管理启动模板版本（尤其是执行允许用户更改“默认”启动模板版本的 `ec2:ModifyLaunchTemplate` 操作）的人员授予权限时，这是一个重要的考虑因素。

向用户授予将 EC2 操作用于启动模板 API 的权限，即是授予该用户（在创建或更新 EC2 实例集或竞价型实例集的情况下）指向包含实例配置文件（IAM 角色的容器）的不同启动模板版本的 `iam:PassRole` 权限。这意味着即使用户没有 `iam:PassRole` 权限，也可以更新启动模板以将 IAM 角色传递给实例。有关更多信息和示例 IAM policy，请参阅《IAM 用户指南》中的使用 IAM 角色向在 Amazon EC2 实例上运行的应用程序授予权限。

有关更多信息，请参阅控制启动模板的使用和示例：使用启动模板。

创建 Amazon EC2 启动模板

您可以通过为实例配置参数指定自己的值，或者从现有启动模板或 Amazon EC2 实例中获取值来创建 Amazon EC2 启动模板。

您无需为启动模板中的每个参数指定值；只需指定一个实例配置参数即可创建启动模板。要指明您选择不指定的参数，请在使用控制台时选择不包括在启动模板中。使用命令行工具时，不包含参数表示您选择不在启动模板中指定这些参数。

如果想在启动模板中指定 AMI，您可以选择一个 AMI，也可以指定一个在实例启动时将指向 AMI 的 Systems Manager 参数。

使用启动模板启动实例时，将使用启动模板中指定的值来配置相应的实例参数。如果启动模板中未指定值，则使用相应实例参数的默认值。

任务

- 通过指定参数创建启动模板
- 从现有启动模板创建启动模板
- 从实例创建启动模板
- 使用 Systems Manager 参数而非 AMI ID

通过指定参数创建启动模板

要创建启动模板，您必须指定启动模板名称和至少一个实例配置参数。

有关每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

Console

创建启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择启动模板，然后选择创建启动模板。
3. 在启动模板名称和说明下，执行以下操作：
   a. 对于设备模板名称，请为您的启动模板输入描述性名称。
   b. 对于模板版本说明，提供此版本的启动模板的简短说明。
   c. 要在创建时标记启动模板，请展开模板标签，选择添加新标签，然后输入标签键值对。为每个要添加的其他标签选择添加新标签。

> **Note**
>
> 要标记启动实例时创建的资源，必须在 Resource tags（资源标签）下面指定标签。有关更多信息，请参阅此过程中的步骤 9。

4. 在应用程序和操作系统映像（亚马逊机器映像）下，您可以保持选中不包括在启动模板中，也可以选择实例的操作系统（OS），然后选择 AMI。或者，您可以指定 Systems Manager 参数而非指定 AMI。有关更多信息，请参阅 使用 Systems Manager 参数而非 AMI ID。

   AMI 是一个模板，其中包含启动实例所需的操作系统和软件。

5. 在实例类型下，您可以保持选中不包括在启动模板中，也可以选择一个实例类型，或者指定实例属性并让 Amazon EC2 使用这些属性识别实例类型。

   > **Note**
   >
   > 仅当自动扩缩组、EC2 实例集和竞价型实例集使用启动模板来启动实例时，才支持指定实例属性。有关更多信息，请参阅 使用基于属性的实例类型选择创建一个混合实例组 和 指定 EC2 实例集或竞价型实例集的实例类型选择属性。
   > 如果您计划使用启动实例向导中的启动模板，或搭配使用 RunInstances API，则无法指定实例类型属性。

   实例类型决定了用于实例的硬件配置（CPU、内存、存储和网络容量）和主机大小。

   如果不确定要选择哪种实例类型，则可以执行以下操作：

   - 选择获取建议，从 EC2 实例类型查找器中获取实例类型的指导和建议。有关更多信息，请参阅 从 EC2 实例类型查找器获取建议。

6. 在密钥对（登录）下，对于密钥对名称，请保持选中不包括在启动模板中、选择现有密钥对或创建一个新密钥对。

7. 在网络设置下，您可以保持选中不包括在启动模板中，也可以为各种网络设置指定值。

8. 在配置存储下，如果您在启动模板中指定了 AMI，则 AMI 会包含一个或多个存储卷，包括根卷 [卷 1（AMI 根）]。您可以选择指定要附加到实例的其他卷。要添加新卷，请选择 Add new volume（添加新卷）。

9. 要标记启动实例时创建的资源，在资源标签下，选择添加标签，然后输入标签键值对。对于 Resource types（资源类型），指定创建时要标记的资源。您可以为所有资源指定相同的标签，也可以为不同的资源指定不同的标签。为每个要添加的其它标签选择添加标签。

   您可以为使用启动模板时创建的以下资源指定标签：

   - 实例
   - 卷

- Elastic Graphics
- 竞价型实例请求
- 网络接口

> **Note**
>
> 要为启动模板本身添加标签，您必须在 Template tags（模板标签）下指定标签。有关更多信息，请参阅此过程中的步骤 3。

10. 对于高级详细信息，请展开该部分以查看字段并（可选）为实例指定任何其他参数。
11. 使用摘要面板查看启动模板配置。您可以通过选择其链接导航到任何部分，然后进行任何必要的更改。
12. 当您准备好创建启动模板时，请选择创建启动模板。

Amazon CLI

创建启动模板

使用 create-launch-template 命令。

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --version-description WebVersion1 \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

以下是一个示例 JSON，其指定实例配置的启动模板数据。将 JSON 保存到文件中并将其包含在 `--launch-template-data` 参数中，如示例命令中所示。

```
{
    "NetworkInterfaces": [{
        "AssociatePublicIpAddress": true,
        "DeviceIndex": 0,
        "Ipv6AddressCount": 1,
        "SubnetId": "subnet-0abcdef1234567890"
    }],
    "ImageId": "ami-0abcdef1234567890",
    "InstanceType": "r5.4xlarge",
```

```
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }],
    "CpuOptions": {
        "CoreCount":4,
        "ThreadsPerCore":2
    }
}
```

下面是示例输出。

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

PowerShell

创建启动模板

使用 New-EC2LaunchTemplate cmdlet。

```
$launchTemplateData = [Amazon.EC2.Model.RequestLaunchTemplateData]@{
    ImageId = 'ami-0abcdef1234567890'
    InstanceType = 'r5.4xlarge'
    NetworkInterfaces = @(

 [Amazon.EC2.Model.LaunchTemplateInstanceNetworkInterfaceSpecificationRequest]@{
            AssociatePublicIpAddress = $true
            DeviceIndex = 0
            Ipv6AddressCount = 1
            SubnetId = 'subnet-0abcdef1234567890'
        }
```

```
    )
    TagSpecifications = @(
        [Amazon.EC2.Model.LaunchTemplateTagSpecificationRequest]@{
            ResourceType = 'instance'
            Tags = [Amazon.EC2.Model.Tag]@{
                Key = 'Name'
                Value = 'webserver'
            }
        }
    )
    CpuOptions = [Amazon.EC2.Model.LaunchTemplateCpuOptionsRequest]@{
        CoreCount = 4
        ThreadsPerCore = 2
    }
}
$tagSpecificationData = [Amazon.EC2.Model.TagSpecification]@{
    ResourceType = 'launch-template'
    Tags = [Amazon.EC2.Model.Tag]@{
        Key = 'purpose'
        Value = 'production'
    }
}
New-EC2LaunchTemplate -LaunchTemplateName 'TemplateForWebServer' `
    -VersionDescription 'WebVersion1' `
    -LaunchTemplateData $launchTemplateData `
    -TagSpecification $tagSpecificationData
```

下面是示例输出。

```
CreatedBy            : arn:aws:iam::123456789012:root
CreateTime           : 9/19/2023 16:57:55
DefaultVersionNumber : 1
LatestVersionNumber  : 1
LaunchTemplateId     : lt-01238c059eEXAMPLE
LaunchTemplateName   : TemplateForWebServer
Tags                 : {purpose}
```

从现有启动模板创建启动模板

您可以克隆现有的启动模板，然后调整参数以创建新的启动模板。但是，您只能在使用 Amazon EC2 控制台时执行此操作。Amazon CLI 不支持克隆模板。有关每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

Console

从现有启动模板创建启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择启动模板，然后选择创建启动模板。
3. 对于设备模板名称，请为您的启动模板输入描述性名称。
4. 对于模板版本说明，提供此版本的启动模板的简短说明。
5. 要在创建时标记启动模板，请展开模板标签，选择添加新标签，然后输入标签键值对。
6. 展开源模板，对于启动模板名称，选择要作为新启动模板基础的启动模板。
7. 对于源模板版本，请选择新启动模板版本所基于的启动模板版本。
8. 根据需要，调整任何启动参数，然后选择Create launch template (创建启动模板)。

从实例创建启动模板

您可以克隆现有 Amazon EC2 实例的参数，然后调整参数以创建启动模板。有关每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

Console

从实例创建启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择操作、图像和模板、从实例创建模板。
4. 提供名称、描述和标签，然后根据需要调整启动参数。

> **Note**
>
> 通过实例创建启动模板时，该实例的网络接口 ID 和 IP 地址将不包含在模板中。

5. 选择创建启动模板。

Amazon CLI

您可以使用 Amazon CLI 从现有实例创建启动模板，方法是先从实例获取启动模板数据，然后使用启动模板数据创建启动模板。

从实例获取启动模板数据

- 使用 get-launch-template-data 命令，并指定实例 ID。您可以将输出作为基础以创建新的启动模板或启动模板版本。默认情况下，输出包含一个顶级 LaunchTemplateData 对象，无法在启动模板数据中指定该对象。请使用 --query 选项排除该对象。

```
aws ec2 get-launch-template-data \
    --instance-id i-0123d646e8048babc \
    --query "LaunchTemplateData"
```

下面是示例输出。

```
    {
        "Monitoring": {},
        "ImageId": "ami-8c1be5f6",
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "DeleteOnTermination": true
                }
            }
        ],
        "EbsOptimized": false,
        "Placement": {
            "Tenancy": "default",
            "GroupName": "",
            "AvailabilityZone": "us-east-1a"
        },
        "InstanceType": "t2.micro",
        "NetworkInterfaces": [
            {
                "Description": "",
                "NetworkInterfaceId": "eni-35306abc",
                "PrivateIpAddresses": [
                    {
                        "Primary": true,
                        "PrivateIpAddress": "10.0.0.72"
                    }
                ],
                "SubnetId": "subnet-7b16de0c",
                "Groups": [
```

```
                    "sg-7c227019"
                ],
                "Ipv6Addresses": [
                    {
                        "Ipv6Address": "2001:db8:1234:1a00::123"
                    }
                ],
                "PrivateIpAddress": "10.0.0.72"
            }
        ]
    }
```

您可以将输出直接写入到一个文件中，例如：

```
aws ec2 get-launch-template-data \
    --instance-id i-0123d646e8048babc \
    --query "LaunchTemplateData" >> instance-data.json
```

使用启动模板数据创建启动模板

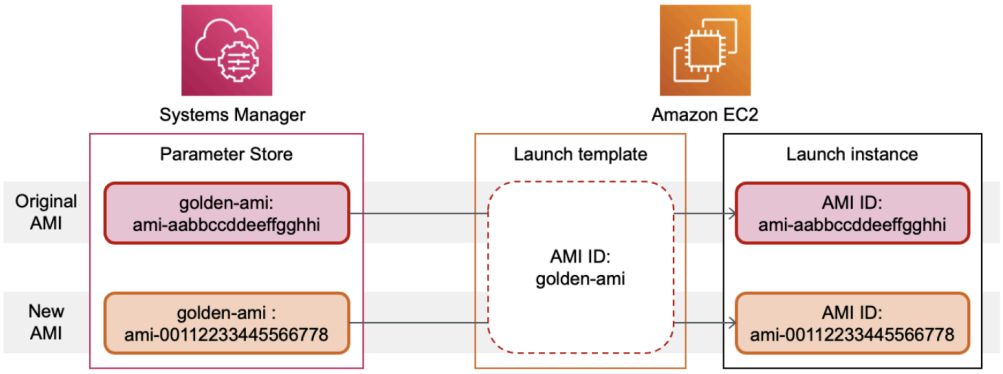
- 使用 create-launch-template 命令可以使用上一步骤的输出创建启动模板。有关使用 Amazon CLI 创建启动模板的更多信息，请参阅通过指定参数创建启动模板。

使用 Systems Manager 参数而非 AMI ID

您可以指定 Amazon Systems Manager 参数，而不是在启动模板中指定 AMI ID。如果 AMI ID 发生变更，您可以通过更新 Systems Manager Parameter Store 中的 Systems Manager 参数在一个位置更新 AMI ID。参数也可以与其他 Amazon Web Services 账户 共享。您可以在一个账户中集中存储和管理 AMI 参数，并与需要引用这些参数的所有其他账户共享。使用 Systems Manager 参数，只需一次操作即可更新所有启动模板。

Systems Manager 参数是用户定义的键值对，您可以在 Amazon Systems Manager Parameter Store 中创建该键值对。Parameter Store 提供了一个集中位置来存储应用程序配置值。

在下图中，golden-ami 参数首先映射到 Parameter Store 中的原始 AMI ami-aabbccddeeffgghhi。在启动模板中，AMI ID 的值为 golden-ami。使用此启动模板启动实例时，AMI ID 解析为 ami-aabbccddeeffgghhi。之后，AMI 会更新，从而生成新的 AMI ID。在 Parameter Store 中，golden-ami 参数将映射到新的 ami-00112233445566778。启动模板保持不变。使用此启动模板启动实例时，AMI ID 解析为新的 ami-00112233445566778。



AMI ID 的 Systems Manager 参数格式

根据启动模板要求，用户定义的 Systems Manager 参数在替换 AMI ID 时应遵循以下格式：

- 参数类型：`String`
- 参数数据类型：`aws:ec2:image` – 确保 Parameter Store 验证您输入的值是否采用 AMI ID 的正确格式。

有关为 AMI ID 创建有效参数的更多信息，请参阅《Amazon Systems Manager 用户指南》中的创建 Systems Manager 参数。

启动模板中的 Systems Manager 参数格式

要在启动模板中使用 Systems Manager 参数替换 AMI ID，必须在启动模板中指定参数时采用以下格式之一：

要引用公有参数，请执行以下操作：

- `resolve:ssm:public-parameter`

要引用存储在同一账户中的参数，请执行以下操作：

- `resolve:ssm:parameter-name`
- `resolve:ssm:parameter-name:version-number` – 版本号本身是默认标签
- `resolve:ssm:parameter-name:label`

要引用其他 Amazon Web Services 账户 共享的参数，请执行以下操作：

- `resolve:ssm:parameter-ARN`

- `resolve:ssm:parameter-ARN:version-number`
- `resolve:ssm:parameter-ARN:label`

参数版本

Systems Manager 参数是版本控制资源。更新参数时，会创建该参数的连续新版本。Systems Manager 支持参数标签，您可以将其映射到参数的特定版本。

例如，golden-ami 参数可能有三个版本：1、2 和 3。您可以创建映射到版本 2 的参数标签 beta，以及映射到版本 3 的参数标签 prod。

在启动模板中，您可以使用以下任一格式指定 golden-ami 参数的版本 3：

- `resolve:ssm:golden-ami:3`
- `resolve:ssm:golden-ami:prod`

指定版本或标签是可选的。如果未指定版本或标签，将使用参数的最新版本。

在启动模板中指定 Systems Manager 参数

创建启动模板或其新版本时，可以在启动模板中指定 Systems Manager 参数，而不是 AMI ID。

Console

在启动模板中指定 Systems Manager 参数

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择启动模板，然后选择创建启动模板。
3. 对于设备模板名称，请为您的启动模板输入描述性名称。
4. 在应用程序和操作系统镜像（Amazon 机器映像）下，选择浏览其他 AMI。
5. 选择搜索栏右侧的箭头按钮，然后选择指定自定义值/Systems Manager 参数。
6. 在指定自定义值或 Systems Manager 参数对话框中，执行以下操作：

   a. 对于 AMI ID 或 Systems Manager 参数字符串，使用以下格式之一输入 Systems Manager 参数名称：

      要引用公有参数，请执行以下操作：

      - **`resolve:ssm:public-parameter`**

要引用存储在同一账户中的参数，请执行以下操作：

- **resolve:ssm:*parameter-name***
- **resolve:ssm:*parameter-name*:*version-number***
- **resolve:ssm:*parameter-name*:*label***

要引用其他 Amazon Web Services 账户 共享的参数，请执行以下操作：

- **resolve:ssm:*parameter-ARN***
- **resolve:ssm:*parameter-ARN*:*version-number***
- **resolve:ssm:*parameter-ARN*:*label***

b. 选择保存。

7. 根据需要指定任何其他启动模板参数，然后选择创建启动模板。

有关更多信息，请参阅 通过指定参数创建启动模板。

Amazon CLI

在启动模板中指定 Systems Manager 参数

- 使用 create-launch-template 命令创建启动模板。要指定要使用的 AMI，请使用以下格式之一输入 Systems Manager 参数名称：

  要引用公有参数，请执行以下操作：

  - **resolve:ssm:*public-parameter***

  要引用存储在同一账户中的参数，请执行以下操作：

  - **resolve:ssm:*parameter-name***
  - **resolve:ssm:*parameter-name*:*version-number***
  - **resolve:ssm:*parameter-name*:*label***

  要引用其他 Amazon Web Services 账户 共享的参数，请执行以下操作：

  - **resolve:ssm:*parameter-ARN***

- **resolve:ssm:*parameter-ARN*:*version-number***
- **resolve:ssm:*parameter-ARN*:*label***

下面的示例创建一个指定以下内容的启动模板：

- 启动模板的名称（*TemplateForWebServer*）
- 启动模板的标签（*purpose*=*production*）
- 在 JSON 文件中指定的实例配置数据：
  - 要使用的 AMI（resolve:ssm:*golden-ami*）
  - 要启动的实例类型（*m5.4xlarge*）
  - 实例的标签（*Name*=*webserver*）

```
aws ec2 create-launch-template \
    --launch-template-name TemplateForWebServer \
    --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
    --launch-template-data file://template-data.json
```

以下是一个示例 JSON 文件，其中包含实例配置的启动模板数据。ImageId 的值是以所需格式 resolve:ssm:*golden-ami* 输入的 Systems Manager 参数名称。

```
{"LaunchTemplateData": {
    "ImageId": "resolve:ssm:golden-ami",
    "InstanceType": "m5.4xlarge",
    "TagSpecifications": [{
        "ResourceType": "instance",
        "Tags": [{
            "Key":"Name",
            "Value":"webserver"
        }]
    }]
  }
}
```

验证启动模板是否获得正确的 AMI ID

将 Systems Manager 参数解析为实际的 AMI ID

使用 describe-launch-template-versions 命令并包含 `--resolve-alias` 参数。

```
aws ec2 describe-launch-template-versions \
    --launch-template-name my-launch-template \
    --versions $Default \
    --resolve-alias
```

响应包含 `ImageId` 的 AMI ID。在本示例中，使用此启动模板启动实例时，AMI ID 解析为 `ami-0ac394d6a3example`。

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-089c023a30example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2022-12-28T19:52:27.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0ac394d6a3example",
                "InstanceType": "t3.micro",
            }
        }
    ]
}
```

相关资源

有关使用 Systems Manager 参数的更多信息，请参阅 Systems Manager 文档中的以下参考资料。

- 有关如何查找 Amazon EC2 支持的 AMI 公有参数的信息，请参阅 Calling AMI public parameters。
- 有关与其他 Amazon 账户共享参数或通过 Amazon Organizations 共享参数的信息，请参阅 Working with shared parameters。
- 有关监控参数是否成功创建的信息，请参阅 Native parameter support for Amazon Machine Image IDs。

限制

- 只有 `instant` 类型的 EC2 实例集支持使用指定了 Systems Manager 参数而不是 AMI ID 的启动模板。
- `maintain` 和 `request` 类型的 EC2 实例集以及竞价型实例集不支持使用指定了 Systems Manager 参数而不是 AMI ID 的启动模板。对于 `maintain` 和 `request` 类型的 EC2 实例集以及竞价型实例集，如果您在启动模板中指定 AMI，则必须指定 AMI ID。
- 如果在 EC2 实例集中使用基于属性的实例选择，则无法指定 Systems Manager 参数来代替 AMI ID。如果基于属性的实例选择，则必须指定 AMI ID。
- Amazon EC2 Auto Scaling 提供其他限制。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的在启动模板中使用 Amazon Systems Manager 参数代替 AMI ID。

修改启动模板（管理启动模板版本）

启动模板是不可变的；在创建启动模板之后，您无法修改它。相反，您可以创建包含所需更改的新版本启动模板。

您可以创建不同版本的启动模板，设置默认版本，描述启动模板版本以及删除不再需要的版本。

任务

- 创建启动模板版本
- 设置默认启动模板版本
- 描述启动模板版本

创建启动模板版本

在创建启动模板版本时，您可以指定新的启动参数，或者将现有版本作为基础以创建新的版本。有关每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

Console

创建启动模板版本

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Launch Templates。
3. 选择启动模板，然后依次选择操作、修改模板（创建新版本）。

4. 对于 Template version description（模板版本说明），输入此版本启动模板的说明。
5. （可选）展开 Source template (源模板)，然后选择要用作新启动模板版本基础的启动模板版本。新启动模板版本从此启动模板版本继承启动参数。
6. 根据需要修改启动参数。
7. 选择Create launch template（创建启动模板）。

Amazon CLI

创建启动模板版本

使用 create-launch-template-version 命令。您可以指定新版本所基于的源版本。新版本从此版本继承启动参数，您可以使用 `--launch-template-data` 覆盖参数。以下示例根据启动模板的版本 1 创建新的版本并指定不同的 AMI ID。

```
aws ec2 create-launch-template-version \
    --launch-template-id lt-0abcd290751193123 \
    --version-description WebVersion2 \
    --source-version 1 \
    --launch-template-data "ImageId=ami-0abcdef1234567890"
```

PowerShell

创建启动模板版本

使用 New-EC2LaunchTemplateVersion Cmdlet。您可以指定新版本所基于的源版本。新版本从此版本继承启动参数，您可以使用 `LaunchTemplateData` 覆盖参数。以下示例根据启动模板的版本 1 创建新的版本并指定不同的 AMI ID。

```
New-EC2LaunchTemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -VersionDescription WebVersion2 `
    -SourceVersion 1 `
    -LaunchTemplateData (
        New-Object `
            -TypeName Amazon.EC2.Model.RequestLaunchTemplateData `
            -Property @{ImageId = 'ami-0abcdef1234567890'}
    )
```

设置默认启动模板版本

您可以设置启动模板的默认版本。如果通过启动模板启动实例并且未指定版本，将使用默认版本的参数启动实例。

Console

设置默认启动模板版本

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Launch Templates。
3. 选择启动模板，然后依次选择操作及设置默认版本。
4. 对于 Template version (模板版本)，选择要设置为默认版本的版本号，然后选择 Set as default version (设置为默认版本)。

Amazon CLI

设置默认启动模板版本

使用 modify-launch-template 命令。

```
aws ec2 modify-launch-template \
    --launch-template-id lt-0abcd290751193123 \
    --default-version 2
```

PowerShell

设置默认启动模板版本

使用 Edit-EC2LaunchTemplate cmdlet。

```
Edit-EC2LaunchTemplate `
    -LaunchTemplateId lt-0abcd290751193123 `
    -DefaultVersion 2
```

描述启动模板版本

使用控制台，您可以查看所选启动模板的所有版本，或获取其最新版本或默认版本与特定版本号匹配的启动模板列表。使用 Amazon CLI，您可以描述指定启动模板的所有版本、单个版本或一系列版本。您还可以描述账户中所有启动模板的所有最新版本或所有默认版本。

Console

描述启动模板版本

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Launch Templates。
3. 您可以查看特定启动模板的某个版本，或获取其最新版本或默认版本与特定版本号匹配的启动模板列表。
   - 要查看启动模板的版本，请执行以下操作：选择启动模板。在版本选项卡上，从版本中选择一个版本，以查看其详细信息。
   - 要获取最新版本与特定版本号匹配的所有启动模板的列表，请执行以下操作：从搜索栏中选择最新版本，然后选择版本号。
   - 要获取其默认版本与特定版本号匹配的所有启动模板的列表，请执行以下操作：从搜索栏中选择默认版本，然后选择版本号。

Amazon CLI

描述启动模板版本

使用 describe-launch-template-versions 命令，并指定版本号。在以下示例中，指定了版本 *1* 和 *3*。

```
aws ec2 describe-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1 3
```

描述您账户中最新的和默认的启动模板版本

使用 describe-launch-template-versions 命令并指定 `$Latest` 和/或 `$Default`。您必须在调用中省略启动模板 ID 和名称。您不能指定版本号。

```
aws ec2 describe-launch-template-versions \
    --versions "$Latest,$Default"
```

PowerShell

描述启动模板版本

使用 Get-EC2TemplateVersion Cmdlet 并指定版本号。在以下示例中，指定了版本 *1* 和 *3*。

```
Get-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1,3
```

描述您账户中最新的和默认的启动模板版本

使用 Get-EC2TemplateVersion Cmdlet 并指定 $Latest、$Default 或两者。您必须在调用中省略启动模板 ID 和名称。您不能指定版本号。

```
Get-EC2TemplateVersion `
    -Version '$Latest','$Default'
```

删除启动模板或启动模板版本

如果不再需要某个启动模板，您可以将其删除。如果删除启动模板，则会删除该模板的所有版本。如果只想删除启动模板的某个特定版本，可以在保留启动模板其他版本的同时删除该版本。

删除启动模板或启动模板版本不会影响您从启动模板启动的任何实例。

删除启动模板及其所有版本

如果不再需要某个启动模板（包括其所有版本），您可以删除该启动模板。如果删除启动模板，则会删除该模板的所有版本。

Console

删除启动模板及其所有版本

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Launch Templates。
3. 选择启动模板，然后依次选择操作、删除模板。
4. 输入 **Delete** 以确认删除，然后选择删除。

Amazon CLI

删除启动模板及其所有版本

使用 delete-launch-template 命令，并指定启动模板。

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

PowerShell

删除启动模板及其所有版本

使用 Remove-EC2LaunchTemplate（Amazon Tools for PowerShell）命令，并指定启动模板。如果省略 `-Force`，PowerShell 会提示您进行确认。

```
Remove-EC2LaunchTemplate -LaunchTemplateId lt-0123456789example -Force
```

删除启动模板版本

如果不再需要某个启动模板版本，您可以将其删除。

注意事项

- 在删除该版本后，您将无法替换版本号。
- 您无法删除启动模板的默认版本；您必须先分配一个不同的版本以作为默认版本。如果默认版本是启动模板的唯一版本，您必须删除整个启动模板。
- 使用控制台时，一次可以删除一个启动模板版本。使用 Amazon CLI 时，您可以在单个请求中删除多达 200 个启动模板版本。要在单个请求中删除 200 多个版本，您可以删除启动模板，这样还会删除其所有版本。

Console

删除启动模板版本

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Launch Templates。
3. 选择启动模板，然后依次选择操作、删除模板版本。
4. 选择要删除的版本，然后选择删除。

Amazon CLI

删除启动模板版本

使用 delete-launch-template-versions 命令，并指定要删除的版本号。您可以在单个请求中删除多达 200 个启动模板版本。

```
aws ec2 delete-launch-template-versions \
    --launch-template-id lt-0abcd290751193123 \
    --versions 1
```

PowerShell

删除启动模板版本

使用 Remove-EC2TemplateVersion cmdlet 并指定要删除的版本号。您可以在单个请求中删除多达 200 个启动模板版本。

```
Remove-EC2TemplateVersion `
    -LaunchTemplateId lt-0abcd290751193123 `
    -Version 1
```

启动 Amazon EC2 实例

实例是 Amazon 云中的虚拟服务器。您可以从亚马逊机器映像（AMI）中启动实例。AMI 为实例提供操作系统、应用程序服务器和应用程序。

在实例运行期间，即使该实例处于闲置状态，您也需为其付费，。有关更多信息，请参阅 Amazon EC2 定价。

当您启动实例时，可以在与以下一项资源关联的子网中启动实例：

- 可用区：此选项为默认选项。
- 本地区：要在本地区中启动实例，您必须选择加入本地区，然后在该区域中创建子网。有关更多信息，请参阅 Get started with Local Zones。
- Wavelength 区：要在 Wavelength 区中启动实例，您必须选择加入 Wavelength 区，然后在该区域中创建子网。有关如何在 Wavelength 区域中启动实例的信息，请参阅 Get started with Amazon Wavelength。
- Outpost：要在 Outpost 中启动实例，您必须创建 Outpost。有关如何创建 Outpost 的信息，请参阅 Get started with Amazon Outposts。

启动实例之后，您可以连接并使用该实例。开始时，实例的状态为 pending。当实例状态为 running 时，实例已经开始启动。可能要过一小段时间才能连接到实例。请注意，裸机实例类型可能需要更长时间才能启动。

您可能需要在启动实例时进行某些配置，具体取决于您计划用于连接实例的方式。这些配置可能包括为某些流量指定入站安全组规则或关联实例配置文件角色。要详细了解可用于连接的连接方法及其要求，请参阅连接到您的 EC2 实例。

实例将获得一个公有 DNS 名称，您可使用此名称通过 Internet 与实例通信。实例还会获得一个私有 DNS 名称，相同 VPC 网络内的其他实例可以用其与该实例通信。

当您完成实例时，为避免产生不必要的成本，请确保终止该实例。有关更多信息，请参阅 终止 Amazon EC2 实例。

如果您需要启动大量实例、使用多种实例类型或使用多种购买选项（例如按需型实例、预留实例和竞价型实例），请考虑使用 EC2 实例集。有关更多信息，请参阅 EC2 实例集和竞价型实例集。

如果您希望实现实例生命周期自动化，包括自动扩展、运行状况检查和替换运行状况不佳的实例，请考虑使用 Amazon EC2 Auto Scaling。

以下方法是可以启动实例的一些方式。

方法	Tool	文档
使用启动实例向导指定启动参数。	Amazon EC2 控制台	使用控制台中的启动实例向导来启动 EC2 实例
创建启动模板并通过启动模板启动实例。	Amazon EC2 控制台	使用启动模板来启动 EC2 实例
将现有实例作为基础。	Amazon EC2 控制台	使用现有实例的详细信息来启动 EC2 实例
使用从 Amazon Web Services Marketplace 购买的 AMI。	Amazon EC2 控制台	从 Amazon Web Services Marketplace AMI 中启动 Amazon EC2 实例
使用您指定的 AMI。	Amazon CLI	在 Amazon CLI 中启动、列出和删除 Amazon EC2 实例

方法	Tool	文档
使用您指定的 AMI。	Amazon Tools for Windows PowerShell	使用 Windows PowerShell 启动 Amazon EC2 实例
使用 EC2 实例集跨不同的 EC2 实例类型和可用区以及跨按需型实例、预留实例和竞价型实例购买选项预置容量。	Amazon CLI	EC2 实例集和竞价型实例集
使用 Amazon CloudFormation 模板指定实例。	Amazon CloudFormation	Amazon CloudFormation 用户指南中的 AWS::EC2::Instance
使用特定于语言的 Amazon SDK 启动实例。	Amazon SDK	Amazon SDK for .NET 适用于 C++ 的 Amazon SDK 适用于 Go 的 Amazon SDK Amazon SDK for Java 适用于 JavaScript 的 Amazon SDK 适用于 PHP V3 的 Amazon SDK Amazon SDK for Python 适用于 Ruby V3 的 Amazon SDK

EC2 实例的启动教程

启动和配置 Amazon EC2 实例的方法有很多，具体使用的方法和配置取决于具体的使用案例。

以下教程有助您了解如何启动 EC2 实例。如果您刚开始接触 Amazon EC2，我们建议您从第一个教程开始学习。这些教程从基础知识开始，然后逐渐增加更多配置选项，从而帮助您进一步提升。

目标	指向教程的链接
启动我的第一个 EC2 实例 了解如何使用 Amazon EC2 启动实例向导中的默认设置快速启动 Amazon EC2 实例。此外还将学习如何检查实例配置字段和终止实例。 时长：10 分钟	教程 1：启动我的第一个 Amazon EC2 实例
启动并连接到一个 EC2 测试实例 了解如何启动可用于测试的 Amazon EC2 实例。这种实例没有高级配置，也不会存储敏感信息。您还将学习基本的实例配置设置、如何连接实例以及如何停止实例。 时长：30 分钟	教程 2：启动并连接到一个 EC2 测试实例

需要学习其他教程？

- 教程：在 AL2023 上安装 LAMP Web 服务器
- 教程：在 AL2023 上配置 SSL/TLS
- 教程：在 AL2023 上托管 WordPress 博客
- 教程：完成使用 EC2 Instance Connect 连接到实例所需的配置
- Tutorial: Connect an Amazon EC2 instance to an Amazon RDS database（教程：将 Amazon EC2 实例连接到 Amazon RDS 数据库）

教程 1：启动我的第一个 Amazon EC2 实例

教程学习目标	了解如何使用 Amazon EC2 启动实例向导中的默认设置快速启动 Amazon EC2 实例。此外还将学习如何检查实例配置字段和终止实例。
EC2 经验水平	初学者

Duration	10 分钟
成本	符合免费套餐资格 创建 Amazon Web Services 账户后，您可以通过 Amazon Web Services 中国区免费套餐开始免费使用 Amazon EC2。 如果您在 2025 年 7 月 15 日之前创建过 Amazon Web Services 账户，其使用时间未满 12 个月，并且未超出 Amazon EC2 的免费套餐权益范围，则学完本教程将不需要任何费用，因为我们会帮助您选择免费套餐权益范围内的选项。否则，从启动实例之时起至您终止实例之时止，您需要支付标准的 Amazon EC2 使用费（即使实例处于空闲状态也不例外）。 如果您是在 2025 年 7 月 15 日当天或之后创建的 Amazon Web Services 账户，其使用时间未满 6 个月，且尚未用完所有服务抵扣金，则学完本教程将不需要任何费用，因为我们会帮助您选择免费套餐权益范围内的选项。
先决条件	• 您必须拥有一个 Amazon 账户，配置了一个具有管理员访问权限的用户，并使用该管理员用户登录 Amazon Web Services 管理控制台。不确定如何执行此操作？ 试试下面的教程：设置 Amazon 环境 • 您必须拥有 Amazon 控制台的基本知识。不确定从哪里开始？ 试试下面的入门指南：Amazon Web Services 管理控制台 入门

教程概述

本教程专为没有 Amazon EC2 使用经验的初学者设计。我们将指导您完成使用 EC2 控制台创建（我们称之为启动）第一个 EC2 实例的步骤。实例本质上是一种位于 Amazon 云中的 Web 服务器。启动实

例后，我们将演示如何在控制台中找到该实例。最后，为帮助管理成本，我们将演示如何删除（我们称之为终止）实例。

本教程分为以下几个简短的任务。您必须完成每个任务后才能进入下一个任务。

- 任务 1：启动实例
- 任务 2：查找实例
- 任务 3：查看实例配置
- 任务 4：终止实例

任务 1：启动实例

在此任务中，您将仅完成基本的操作，以最快的速度启动实例。我们将使用 EC2 启动实例向导，这是一种基于 Web 的表单，其中提供了用于配置和启动实例的所有字段。向导通过提供实例配置字段的默认值来简化启动过程。

开始之前

确保您已完成上表所列的先决条件，包括使用管理员用户身份登录 Amazon Web Services 管理控制台。

按照以下步骤快速启动实例

1. 打开 Amazon EC2 控制台：

   转到 https://console.aws.amazon.com/ec2/。
2. 打开 EC2 启动实例向导：

   从 EC2 控制面板中，选择启动实例。

   这时将打开基于 Web 的启动实例表单。这是 EC2 启动实例向导。
3. 为实例命名：

   在名称和标签下，对于名称，输入一个描述性的名称，例如 **`My first EC2 instance`**。

   尽管为实例命名并非必需的操作，但有助于以后识别实例。
4. 在不提供密钥对的情况下继续操作：

   在密钥对（登录）下的密钥对名称，请选择继续操作但不提供密钥对（不推荐）。

密钥对可用于安全登录。不过，由于我们在本教程中不会登录到实例，因此暂时不需要密钥对。

5. 启动实例：

   在右侧的摘要面板中，选择启动实例。

   Amazon EC2 会使用默认设置快速启动实例。成功横幅确认启动成功。

恭喜！您已经成功启动了第一个 EC2 实例！

任务 2：查找实例

在此任务中，您将在 EC2 控制台中查找您刚刚启动的实例。

按照这些步骤在 EC2 控制台中查找实例

1. 打开实例页面：

   如果您仍在“成功”页面上，请选择屏幕顶部面包屑菜单中的实例。您可能需要首先选择三个省略号才能访问此菜单。

   如果您已离开此页面，请从导航窗格中选择实例。

2. 查找实例

   在名称列中，按照您提供的实例名称查找您的实例。

任务 3：查看实例配置

在此任务中，您将了解如何查看实例的配置详细信息。

按照以下步骤查看实例的配置

1. 找到实例 ID：

   在实例 ID 列中，记下实例的唯一 ID。实例 ID 以 i- 开头，然后接 17 个字母数字字符，例如 i-01aeed690c9fb5322。

   实例 ID 是在实例启动时自动分配给实例的。

2. 打开实例详细信息页面：

在实例 ID 列中，选择该 ID 的链接以打开实例详细信息页面，然后您可以在其中查看实例的配置。

3. 浏览实例配置详细信息：

   花几分钟时间浏览实例的配置详细信息。在下一个教程中，我们将更深入地了解配置。现在，请利用这段时间来熟悉实例详细信息页面。

   提示：要快速找到某个字段，可按键盘上的 Ctrl+F 或 command+F。

   a. 实例类型：您能找到实例类型吗？ 例如，其可能是 t3.micro。
   b. 公有 IPv4 地址：您能找到分配给实例的公有 IPv4 地址吗？ 其格式与以下示例类似：34.242.148.128。
   c. 实例所有者：您能确定此实例的所有者吗？ 就是您！您的 Amazon Web Services 账户编号列在所有者字段下。
   d. 实例标签：您为实例提供的名称实际上是一个标签。您能找到实例的标签吗？ 选择标签选项卡。键是 Name，值是您提供的名称。
   e. 启动时间：您能找到您启动实例的时间吗？ 选择详细信息选项卡，然后找到启动时间字段。
   f. 实例状态：您能验证实例的状态吗？ 这应该是正在运行。

再花几分钟时间浏览其他实例配置字段。如果已准备就绪，请继续完成下一任务。

任务 4：终止实例

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

在此任务中，您将删除实例以保护您的免费套餐权益。在 EC2 中，删除实例称为终止。

按照以下步骤终止实例

1. 发起终止：

如果您仍在实例详细信息页面上，请选择实例状态菜单（右上角），然后选择终止（删除）实例。

如果您已离开此页面，请从导航窗格中选择实例。然后在实例页面上，选中实例名称旁边的复选框，然后选择实例状态菜单（右上角），然后选择终止（删除）实例。

2. 确认终止：

   在打开的终止（删除）实例窗口中，选择终止（删除）按钮以确认您确实要终止实例。

3. 监控实例状态：

   在实例页面上，检查实例状态列。实例状态将变为正在关闭。如果您没有看到完整的显示，请尝试展开该列。

   实例关闭后，Amazon EC2 会将其删除，并该实例将从实例页面中消失。

关键要点

在本教程中，您接触了以下几个关键概念：

- 实例是一种位于 Amazon 云中 Amazon EC2 Web 服务器。
- 启动是指创建 EC2 实例。
- 终止是指删除 EC2 实例。
- EC2 启动实例向导包含实例配置的默认值，让您能够轻松快速启动实例。
- 实例 ID 是自动分配给实例的唯一标识符，实例名称是您为了便于识别而分配的可选标签。

后续步骤

要提高启动和终止实例的信心，可以考虑重复练习本教程中的步骤。务必要终止您启动的所有实例，以保护您的免费套餐权益。

熟悉了这些基础知识后，可以继续学习下一个教程。该教程将更深入地介绍关键实例配置字段。

教程 2：启动并连接到一个 EC2 测试实例

教程学习目标	了解如何启动可用于测试的 Amazon EC2 实例。这种实例没有高级配置，也不会存储敏感信息。您还将学习基本的实例配置设置、如何连接实例以及如何停止实例。

EC2 经验水平	初学者
Duration	30 分钟
成本	符合免费套餐资格 创建 Amazon Web Services 账户后，您可以通过 Amazon Web Services 中国区免费套餐开始免费使用 Amazon EC2。 如果您在 2025 年 7 月 15 日之前创建过 Amazon Web Services 账户，其使用时间未满 12 个月，并且未超出 Amazon EC2 的免费套餐权益范围，则学完本教程将不需要任何费用，因为我们会帮助您选择免费套餐权益范围内的选项。否则，从启动实例之时起至您终止实例之时止，您需要支付标准的 Amazon EC2 使用费（即使实例处于空闲状态也不例外）。 如果您是在 2025 年 7 月 15 日当天或之后创建的 Amazon Web Services 账户，其使用时间未满 6 个月，且尚未用完所有服务抵扣金，则学完本教程将不需要任何费用，因为我们会帮助您选择免费套餐权益范围内的选项。
先决条件	完成 教程 1：启动我的第一个 Amazon EC2 实例。

教程概述

本教程专为想要启动 EC2 实例以满足测试需要的初学者而设计。

我们将介绍关键的实例配置字段，然后指导您完成在 EC2 控制台中使用默认值启动测试实例的步骤。启动实例后，我们将演示如何登录（我们称之为连接到）您的实例。在此教程中，我们还将演示如何创建密钥对。连接到实例时必需要有此密钥对。最后，为帮助管理成本，我们将演示如何停止实例以避免产生使用费。

您在此教程中将启动一个 Linux 实例。虽然本教程中的步骤也可用于启动使用其他操作系统的实例，但连接到实例部分的说明仅适用于 Linux 实例。

本教程分为以下几个简短的任务。您必须完成每个任务后才能进入下一个任务。

- 任务 1：自行了解用于启动实例的关键组件
- 任务 2：检查技术示意图
- 任务 3：创建密钥对
- 任务 4：启动测试实例
- 任务 5：查找实例
- 任务 6：查看实例配置
- 任务 7：熟悉用于连接实例的关键组件
- 任务 8：连接到实例
- 任务 9：停止实例

任务 1：自行了解用于启动实例的关键组件

在此任务中，您将探索启动 EC2 实例所需的关键组件。这包括 AMI、实例类型、密钥对、安全组、网络（VPC 和子网）和 Amazon EBS 卷。此外还将探索一个可选组件，即 Name 标签。

为更形象地了解这些组件，可以将实例比作一个出租屋。正如租房为您提供了一个居所，同时又无需您拥有和维护房子，EC2 实例可为您提供计算能力，同时又无需您拥有和维护相关基础设施。

在决定要启动的实例种类时，您需要考虑实例的配置标准，就好比在选择房子时要考虑的标准。虽然这一类比有些简单化，不过也确实非常实用，有助您在进一步熟悉相关组件前形象地了解。

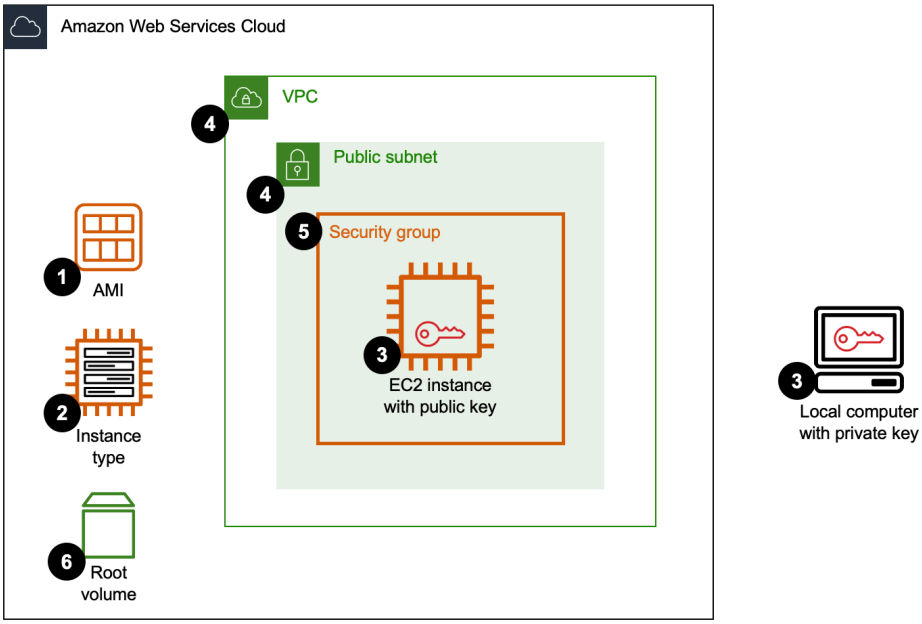
- AMI – 房屋建筑材料和便利设施：亚马逊机器映像（AMI）决定了实例启动时所具有的操作系统和应用程序。这就好比房屋建筑材料（如砖块、钢材或木材）和便利设施（如电器和家具）。基础 AMI 就好比只有基本电器的简装房子，而带有预装软件的自定义 AMI 就好比精装修的房子。
- 实例类型 – 房屋大小和功率：实例类型决定了 EC2 实例的大小和功能，就好比选择房屋的大小、房间数量和能源容量。每种实例类型都决定了实例的 CPU 数量、内存、存储和网络容量。所选的 AMI 可能会限制您可以选择的实例类型。
- 密钥对 – 入户门钥匙：密钥对就好比房子的入户门锁和钥匙。公有密钥充当实例的锁，而私有密钥则是您必须在本地计算机上安全保管的钥匙。如果其他人掌握了您的私有密钥，就可以访问您的实例，就好比有人拿着您的入户门钥匙进入您的房子。
- 网络（VPC 和子网）– 物业边界、分隔区域和门牌号：您的虚拟私有云（VPC）就好比房屋所在的整个物业，子网是房屋的围合分隔区域。如果您所在物业上有多个房屋（实例），则可能需要根据用途将其分成不同的区域（不同的子网）。有些房屋允许游客在花园中自由漫游（可以访问互联网的公

有子网），而另一些房屋则将花园围起来以限制进入（无法访问互联网的私有子网）。每个子网都包含一个 IP 地址范围，就好比门牌号，可以将其分配给子网内的实例。

- 安全组 – 看门人：安全组就好比看门人，控制谁可以进入您的房子。安全组会强制执行一组规则，控制允许哪些流量到达实例。例如，一条允许来自特定 IP 地址的 SSH 流量的规则就好比看门人只允许特定的人上门配送日用品。同样，允许来自任何地方的 HTTPS 流量就好比允许公众来参观您房子的外观。
- Amazon EBS 卷 – 储物单元：EBS 卷就好比您用来存放物品的储物单元。每个实例都有一个根卷（存储 AMI 的地方），您可以根据需要随时添加其他卷（存储）。
- Name 标签 – 房屋名称：Name 标签就好比房屋的标志，有助您轻松标识住在房子中的人。虽然 Name 标签能让您更轻松地区分实例，但对于实例的启动并非必需。

任务 2：检查技术示意图

在此任务中，您将了解我们在 Amazon 文档中使用的典型技术示意图。下图代表了将在本教程中启动的测试实例的配置。在上一个任务中，我们将这些组件比作出租房屋。现在，我们将重点介绍实际的 EC2 组件本身。带编号的标签对应于相应的说明。



1. AMI – AMI 是您在启动实例时选择的映像。这是一个模板，其中包含在实例上运行的操作系统和软件。例如，假设您要启动一个 Linux 实例，则可以选择 Amazon Linux 2023 AMI。或者，如果要启动一个 Windows 实例，则可以选择 Microsoft Windows Server 2022 基本 AMI。Amazon EC2 控制台中的 AMI 目录包含数千个映像供您选择。

2. 实例类型 – 实例类型是实例所用主机的硬件，决定了主机的 CPU、内存、存储和联网容量。Amazon EC2 提供 600 多种实例类型选择，每种实例类型具有不同的硬件配置和大小，从而让您可以根据应用程序的需求选择合适的实例类型。
3. 密钥对 – 密钥对是您在连接到实例时用于证明个人身份的一组安全凭证。公有密钥位于您的实例上，而私有密钥位于您的本地计算机中。

   在 EC2 中，连接您的实例是指从本地计算机登录您的实例。虽然通过其他方法也可安全地连接到您的实例，但在本教程中，我们将使用密钥对。
4. 网络 – 网络由 VPC 和一个或多个子网组成。VPC 是 Amazon Web Services 云 中的虚拟网络。每个 Amazon 客户都有专属于自己 Amazon Web Services 账户 的 VPC。您将在 VPC 的一个子网中启动实例。子网 是 VPC 内的 IP 地址范围。您的默认子网是一个公有子网，因此将分配公有 IP 地址，并提供从 Amazon 网络外部通过互联网访问实例的权限。
5. 安全组 – 安全组是用来控制实例流量的防火墙。安全组包含允许某些类型的流量进入实例的规则。要通过 SSH 从本地计算机连接到实例（使用密钥对），您需要一个允许来自本地计算机的 SSH 流量的规则。
6. EBS 卷 – Amazon EBS 卷是一种功能类似于物理硬盘的存储设备。您的实例有一个根卷，这是一个特殊的 EBS 卷，用于存储 AMI 以及启动实例所需的操作系统和软件。您可以选择性地添加数据卷。不过，由于测试实例不会存储任何敏感数据，因此不需要额外的加密数据卷。

恭喜您！您已完成本教程中的概念任务。在以下任务中，您将使用 Amazon EC2 控制台来创建您已了解的组件。

任务 3：创建密钥对

在此任务中，您将创建一个密钥对。密钥对由两部分组成：一个公有密钥（将添加到您的实例中）和对应的私有密钥（用来安全地连接到您的实例）。在下一个任务中，您将在启动实例时选择此密钥对。将私有密钥安全地存储在本地计算机上至关重要，因为任何有权访问私有密钥的人都可以连接到您的实例。

如果您希望在启动测试实例时使用现有的密钥对，可以跳过此任务。否则，请继续创建新的密钥对。

开始之前

确保您已完成上表所列的先决条件，包括使用管理员用户身份登录 Amazon Web Services 管理控制台。

执行以下步骤创建用户。

1. 打开 Amazon EC2 控制台：

转到 https://console.aws.amazon.com/ec2/。

2. 导航至密钥对控制台页面：

   在导航窗格中的 Network & Security 下，选择 Key Pairs。

   - 如果您之前创建了密钥对，则会在该表中显示。
   - 如果不存在任何密钥对，则该表为空。

3. 创建新的密钥对：

   选择创建密钥对按钮（右上角），打开创建密钥对基于 Web 的表单，然后输入您的密钥对详细信息，如下所示：

   a. 为密钥对命名：对于名称，输入一个有助您标识密钥对的名称，例如 **`test-instance-key-pair`**。

      该名称最多为 255 个 ASCII 字符。它不能包含前导空格或尾随空格。

   b. 选择密钥对类型：对于密钥对类型，请选择 ED25519。

      Linux 实例同时支持 RSA 和 ED25519 密钥类型，而 Windows 实例仅支持 RSA 密钥类型。由于本教程中将会启动一个 Linux 实例，因此可以使用 ED25519 密钥。

   c. 选择私有密钥文件格式：对于私有密钥文件格式，选择 .pem。

      这是用于保存私有密钥文件的格式。

4. 将公有密钥保存到 Amazon EC2 并下载私有密钥：

   选择创建密钥对按钮（右下角）。

   Amazon EC2 将会保存公有密钥，而您的浏览器会自动将私有密钥文件下载到本地计算机。根据您为密钥对指定的名称命名文件，而扩展名是您选择的文件格式。将私有密钥文件移动到本地计算机中的某个安全位置。

   > **Important**
   >
   > 这是您保存私有密钥文件的唯一机会。

5. 设置密钥的权限（适用于 macOS 和 Linux 用户）：

如果您计划在 macOS 或 Linux 计算机上通过 SSH 连接到实例，则必须为私有密钥文件设置正确的权限。打开一个终端窗口并运行以下命令，并将 *test-instance-key-pair* 替换为密钥对的名称：

```
chmod 400 test-instance-key-pair.pem
```

此命令可确保只有您才能读取该私有密钥文件，这是与您的实例建立安全连接所必需的。如果没有这些权限，您将无法使用此密钥对进行连接。

恭喜您！您已成功创建密钥对！

任务 4：启动测试实例

在此任务中，您将使用 EC2 启动实例向导快速启动一个测试实例。您将配置 Linux 实例的主要实例配置设置，其他字段则使用默认值。

为帮助管理成本，我们建议您选择符合免费套餐资格的组件。

按照以下步骤启动测试实例。

1. 打开 Amazon EC2 控制台：

   转到 https://console.aws.amazon.com/ec2/。

2. 打开 EC2 启动实例向导：

   从 EC2 控制面板中，选择启动实例。

   这时将打开基于 Web 的启动实例表单。这是 EC2 启动实例向导。

3. 为实例命名：

   在名称与标签下，针对名称输入一个描述性名称，例如 **Test instance**。

   实例名称是一个标签，其中密钥为名称，而值为您指定的名称。

   提示：对于测试实例，提供一个 name 标签即可。但对于生产实例，则建议制定标记策略，以标准化所有资源的标记。

4. 选择操作系统和软件 – 亚马逊机器映像（AMI）：

在应用程序和操作系统映像（亚马逊机器映像）下；对于亚马逊机器映像（AMI），默认选择为 Amazon Linux 2023 AMI。此 AMI 标记为符合免费套餐资格。在本教程中，您将启动一个 Linux 实例，因此请保留默认设置以确保不超出免费套餐的限制。

5. 选择硬件 – 实例类型：

   在实例类型下，对于实例类型，保留本教程的默认选择。默认实例类型可在免费套餐下使用，其硬件也适合您的测试实例。

6. 准备使用密钥对进行安全登录：

   在（密钥对（登录）下，对于密钥对名称，选择在上一个任务重创建的密钥对。如果您未在列表中看到您的密钥对，请选择刷新图标（位于列表右侧）。

   当实例启动时，系统会将公有密钥放置到实例上。要在实例启动后连接到该实例，您需要使用您在上一个任务中下载的对应私有密钥。

7. 配置网络设置以启用互联网访问权限：

   在网络设置下，网络（您的 VPC）和子网字段已默认配置好。保留本教程的默认设置，这有助您快速开始。如果您未修改默认子网，则实例将可以访问互联网。

   提示：您的默认子网是一个公有子网，因此将分配公有 IP 地址，并提供从 Amazon 网络外部通过互联网访问实例的权限。对于测试实例，可以使用提供互联网访问权限的默认子网设置。但对于生产实例，建议仅在绝对必要时才分配公有 IP 地址并使用可访问互联网的子网。

8. 设置实例防火墙（安全组）：

   在网络设置下的防火墙（安全组）下，确保选中允许来自任何位置 (0.0.0.0) 的 SSH 流量复选框。这将为测试实例创建一个新的安全组，允许来自任何 IP 地址的 SSH 流量。

   安全组充当防火墙以控制实例的流量。要通过 SSH 从本地计算机连接到您的实例，您需要允许来自本地计算机的 SSH 流量的规则。

   提示：如果您的互联网服务提供商使用动态 IP 分配，则本地计算机的 IP 地址可能会持续变化。我们假定您在使用实例进行测试时，不会使用该实例来存储敏感信息，因此安全措施可以不那么严格。对于测试实例，通常可以允许来自任何 IP 地址 (`0.0.0.0/0`) 的流量，因此即使您的 IP 地址发生变化，也可以随时建立连接。但对于生产实例，尤其是那些存有敏感数据的实例，建议只允许来自已知 IP 地址的流量。

9. 配置实例存储：

   在配置存储下，系统会默认配置根卷（加密）字段。保留设置不变，以确保不超出免费套餐限制。

由于测试实例不会存储任何敏感数据，因此不需要额外的加密数据卷。

10. 检查实例配置：

    在启动实例之前，您可以在右侧的摘要面板中检查高级别设置。

11. 启动实例：

    准备好启动实例后，请在摘要面板中选择启动实例。

    Amazon EC2 会使用您指定的设置快速启动实例。如果您未指定设置，则将使用默认设置。成功横幅确认启动成功。

恭喜您！您已成功启动测试实例！

任务 5：查找实例

在此任务中，您将在 EC2 控制台中查找您刚刚启动的实例。

按照这些步骤在 EC2 控制台中查找实例

1. 打开实例页面：

   如果您仍在“成功”页面上，请在成功横幅中选中实例 ID。

   如果您已离开此页面，请从导航窗格中选择实例。

2. 查找实例

   在名称列中，按照您提供的实例名称查找您的实例。

任务 6：查看实例配置

在此任务中，您将了解如何查看实例的配置详细信息。

按照这些步骤查看您的实例配置

1. 查找实例

   在名称列中，按照您提供的实例名称查找您的实例。

2. 打开实例详细信息页面：

选中实例名称旁边的复选框，然后选择操作菜单（右上角），然后选择查看详细信息以打开实例详细信息页面，您可以在其中检查实例配置。

在上一教程中，您选择了实例的 ID 链接来打开实例详细信息页面。您会发现，在 EC2 控制台中完成任务的方法不止一种。

3. 浏览实例配置详细信息：

   利用几分钟时间浏览您的实例配置详细信息。

   提示：要快速找到某个字段，可按键盘上的 Ctrl+F 或 command+F。

   a. AMI：您能找到用于启动实例的 AMI 吗？ 您可以在详细信息选项卡的 AMI ID 和 AMI 名称中找到相关信息。
   b. 实例类型：您能找到实例类型吗？ 例如，其可能是 t3.micro。
   c. 密钥对：您能找到启动实例时选择的密钥对吗？ 这是为启动时分配的密钥对指定的。请注意，如果您将来更改了密钥对，此处的值也不会更改。
   d. VPC：您能找到您的 VPC ID 吗？ 您可以在联网选项卡上找到所有联网相关的配置设置。VPC ID 的格式与以下示例类似：vpc-1a2b3c4d
   e. 子网：您能找到启动实例的子网的 ID 吗？ 它的格式类似于以下示例：subnet-1a2b3c4d
   f. 公有 IPv4 地址：您能找到分配给实例的公有 IPv4 地址吗？ 其格式与以下示例类似：34.242.148.128。
   g. 安全组：能否找到为了允许来自任何位置 (0.0.0.0./0) 的 SSH 流量而创建的入站规则？ 您可以在安全选项卡上找到所有与安全相关的配置设置。
   h. 存储：您能找到为此实例创建的卷吗？ 您可以在存储选项卡上找到所有与存储相关的配置设置。
   i. 实例标签：您为实例提供的名称实际上是一个标签。您能找到实例的标签吗？ 选择标签选项卡。键是 Name，值是您提供的名称。
   j. 实例状态：您能验证实例的状态吗？ 这应该是正在运行。

再花几分钟时间浏览其他实例配置字段。如果准备就绪，请继续下一任务。

任务 7：熟悉用于连接实例的关键组件

在此任务中，您将探索连接 EC2 实例所需的关键组件。这包括连接协议、公有 DNS、安全组、密钥对和实例用户名。

为形象地了解这些组件，可以将连接到实例比作去您家：

- 连接协议 – 您的交通方式：就好比选择如何回家一样，您可以选择用于连接到实例的连接协议。在本教程中，我们将使用 SSH（安全 Shell），这会创建一个安全隧道，用于通过互联网将您的计算机连接到您的实例。
- 公有 DNS – 房屋地址：与房屋具有一个唯一的地址一样，您的 EC2 实例也有自己的公有 DNS 名称（例如 `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com`）。此公有 DNS 名称让 SSH 能够直接连接到您的实例。
- 安全组 – 看门人：好比您的房子里有一个看门人来控制谁可以出入。同样，EC2 实例的安全组类似于看门人，用于控制允许哪些类型的网络流量可以进出您的实例。仅允许您显式允许的流量（例如，来自您计算机 IP 地址的 SSH 流量）进入。
- 私有密钥 – 您的前门密钥：启动实例时，您指定了一个密钥对。公有密钥放在您的实例上，而私有密钥在您的计算机上保管。私有密钥好比您的入户门钥匙，没有私有密钥就无法进入您的实例。
- 实例用户名 – 居民：到达您家时，您需要表明自己的身份以证明自己是居民。同样，连接到实例时您需要提供一个用户名。不同的实例具有不同的默认用户名，具体取决于操作系统。例如，Amazon Linux 实例使用 `ec2-user` 作为默认用户名。

连接命令

要连接到您的 EC2 实例，请在终端窗口中使用以下命令：

```
ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
west-1.compute.amazonaws.com
```

该命令的具体作用如下：

- `ssh` – 此命令指定连接协议，发起与实例的 SSH（安全 Shell）连接。
- `-i "test-instance-key-pair.pem"` – `-i` 标志指示验证连接所需的私有密钥文件。此私有密钥文件必须与您在启动实例时指定的密钥对匹配。如果您的私有密钥文件保存在特定文件夹中，请指定该文件的完整路径。
- `ec2-user` – 这是登录实例的用户名。对于 Amazon Linux 实例，默认用户名为 `ec2-user`。其他 AMI 可能会使用其他默认用户名，例如 Ubuntu 实例会使用 `ubuntu`。
- `@` – 此符号用于将用户名与实例地址分隔。
- `ec2-18-201-118-201.eu-west-1.compute.amazonaws.com` – 这是实例的公有地址（公有 DNS），其中包括公有 IPv4 地址和 Amazon Web Services 区域，用于唯一标识实例。

当您运行此命令时会发生什么

运行此命令后，SSH 会建立一个安全隧道并使用您的私有密钥进行身份验证。如果实例的安全组允许该流量，则可以访问您的 EC2 实例。现在，您可以通过本地计算机控制实例，就像坐在实例前一样。您可以像在本地计算机上一样运行命令、安装软件和管理文件。

任务 8：连接到实例

在此任务中，您将在本地计算机上使用 SSH 客户端连接到实例。在上一个任务中，我们用到您家去的比喻，来介绍了用于连接到实例的组件。现在，我们将重点介绍如何连接到实际的 EC2 实例。

可以通过多种方法连接到实例。可用于连接的方法取决于实例的操作系统。由于您启动的是 Linux 实例，因此将在本地计算机上使用 SSH 客户端。

首先，检查您的计算机是否安装了 SSH 客户端

大多数计算机都预装了 SSH 客户端。要进行检查，请打开计算机上的终端窗口并运行以下命令：

```
ssh
```

如果命令能被识别，则可以进行连接。

如果命令无法被识别，则必须安装 SSH 客户端。本教程不包含安装 SSH 客户端的操作说明。如果需要帮助，请参阅本用户指南中的 SSH 连接先决条件或在线搜索有关如何在操作系统上安装 SSH 客户端的说明。

按照这些步骤连接到您的实例

1. 发起连接：

   如果您位于 Amazon EC2 控制台的实例详细信息页面上，请选择连接按钮（右上角）。

   如果您已离开此页面，请从导航窗格中选择实例。然后在实例页面上，选中实例名称旁边的复选框并选择连接按钮（右上角）。

   这时将打开连接到实例页面。
2. 选择连接方法：

   在连接到实例页面上，选择 SSH 客户端选项卡。

   请仔细检查此页面上的内容，因为这些是您接下来要执行的步骤。

3. 检查 SSH 命令：

   在示例下，您将看到一个根据实例详细信息自动生成和自定义的命令。私有密钥名称是从启动时指定的公有密钥名称推导出来的。

   命令类似下面这样：

   ```
   ssh -i "test-instance-key-pair.pem" ec2-user@ec2-18-201-118-201.eu-
   west-1.compute.amazonaws.com
   ```

4. 复制 SSH 命令：

   选择示例 SSH 命令旁边的复制图标。

5. 打开一个终端窗口：

   在本地计算机上打开终端窗口。

6. 粘贴并运行该 SSH 命令：

   将该 SSH 命令粘贴到终端窗口中。如果您将私有密钥文件保存在特定文件夹中，请编辑命令以包含完整文件路径。

   按键盘上的回车键。

   您看到的响应与以下内容类似：

   ```
   The authenticity of host 'ec2-18-201-118-201.eu-west-1.compute.amazonaws.com
    (18-201-118-201)' can't be established.
   ED25519 key fingerprint is SHA256:examplehxj9aOr1MogvKOoMNskVVIRBQBoq0example.This
    key is not known by any other names.
   Are you sure you want to continue connecting (yes/no/[fingerprint])?
   ```

7. 完成连接：

   输入 **yes** 并按键盘上的回车键。

   验证指纹不属于本教程的范围。要了解更多信息，请参阅（可选）获取实例指纹。

   成功连接后，终端提示将变为显示实例的公有 DNS。

恭喜您！您已成功连接到您的实例！

任务 9：停止实例

在此任务中，您将停止实例以保护您的免费套餐权益。实例停止后，您就不再产生该实例的费用。如果您是在 2025 年 7 月 15 日之前创建的 Amazon Web Services 账户，并且符合免费套餐的资格，您将继续承担 EBS 存储的费用。

按照以下步骤停止实例

1. 发起停止：

   如果您仍在连接到实例页面上，请从面包屑菜单中选择实例。如果您已离开此页面，请从导航窗格中选择实例。

   然后在实例页面上，选中实例名称旁边的复选框，选择实例状态菜单（右上角），然后选择停止实例。当出现提示时，请选择停止。
2. 监控实例状态：

   在实例页面上，检查实例状态列。实例的状态将变为正在停止，然后变为已停止。如果您没有看到完整的显示，请尝试展开该列。

   如果您认为实例状态已从正在停止变为已停止，但尚未看到该状态，请选择刷新图标（该表上方）刷新实例表。

关键要点

在本教程中，您接触了以下几个关键概念：

- AMI 是指亚马逊机器映像，这是包含启动实例所需操作系统和软件的模板。
- 实例类型是指实例所用主机的硬件。它决定您实例的 CPU、内存、存储和网络容量。
- 密钥对是指可用于安全连接到实例的一组公有密钥和私有密钥。
- 网络是指 VPC（您在 Amazon 云中的账户专属的虚拟私有云）和子网（VPC 中的一个 IP 地址 范围）。
- 安全组是指一组用于控制哪些流量可以到达实例的规则。
- EBS 卷是指实例的数据存储。每个实例都有一个用于存储 AMI 的根卷和一个或多个可选的数据卷。
- 标签是您可以选择分配给实例的元数据。实例名称是一个标签，其键为 Name，值由您选择。
- 连接是指通过互联网访问您的实例。
- SSH 是指可用于连接到实例的安全 Shell 连接协议。

- 公有 DNS 是实例的唯一公有地址。
- 实例用户名由实例的操作系统决定，对于连接是必需的。
- 停止实例会停止产生实例费用，但仍会继续产生 EBS 存储费用。

后续步骤

要提高启动、连接到和停止实例的信心，可以考虑重复练习本教程中的步骤。务必要终止您启动的所有实例，以保护您的免费套餐权益。

熟悉这些基础知识后，就可以浏览更高级的教程。对于其他教程，请参阅 需要学习其他教程？

如果您是在 2025 年 7 月 15 日之前创建的Amazon Web Services 账户，可观看以下 6 分钟的视频：How can I avoid charges on my account when using Amazon Web Services 中国区免费套餐 services

如果您是在 2025 年 7 月 15 日当天或之后创建的 Amazon Web Services 账户，请考虑以下信息：《Amazon Billing User Guide》中的 Explore Amazon services with Amazon Free Tier

Amazon EC2 实例配置参数参考

Amazon EC2 控制台中的启动实例向导和启动模板提供了用于配置 Amazon EC2 实例的所有参数。

除了密钥对外，启动实例向导还会为每个参数提供默认值。您可以接受任何或全部默认值，也可以通过自己的值来配置实例。创建启动模板时，参数是可选的。如果您使用启动模板启动实例，则启动模板中指定的参数将覆盖启动实例向导中的默认值。启动模板中未指定的任何参数都将默认为启动实例向导提供的值。

这些参数在启动实例向导和启动模板中进行了分组。以下描述根据控制台中的参数分组呈现。

实例配置的参数

- 名称和标签
- 应用程序和操作系统映像（亚马逊机器映像）
- 实例类型
- 密钥对（登录）
- 网络设置
- 配置存储
- 高级详细信息

- 摘要

名称和标签

实例名称是一个标签，其中密钥为 Name（名称），而值为您指定的名称。您可以标记实例、卷和网络接口。对于竞价型实例，您只能标记竞价型实例请求。有关标签的信息，请参阅 标记 Amazon EC2 资源。

指定实例名称和其它标签为可选项。

- 对于 Name（名称），为实例输入一个描述性名称。如果您没有指定名称，则可以通过其 ID 标识实例，该 ID 将在您启动实例时自动生成。
- 要添加其它标签，请选择 Add additional tags（添加其它标签）。选择 Add tag（添加标签），然后输入密钥和值，然后选择要标记的资源类型。为每个要添加的其它标签选择 Add tag（添加标签）。

您只能在启动实例时指定实例名称。创建启动模板时您无法命名实例，但可以为实例启动时创建的资源添加标签。

应用程序和操作系统映像（亚马逊机器映像）

亚马逊机器映像（AMI）中包含了创建实例所需的信息。例如，AMI 可能包含充当 Web 服务器所需的软件，例如 Linux、Apache 和您的网站。

您可以找到适合的 AMI，如下所示。通过查找 AMI 的每个选项，您可以选择右上角的 Cancel（取消）以返回启动实例向导，而不选择 AMI。

搜索栏

要搜索所有可用的 AMI，请在 AMI 搜索栏中输入关键字，然后按 Enter 键。要选择 AMI，请选择 Select（选择）。

最近使用的项目

您最近使用的 AMI。

选择最近启动或当前使用，然后在亚马逊机器映像（AMI）中，选择一个 AMI。

我的 AMI

您拥有的私有 AMI，或与您共享的私有 AMI。

选择我拥有的或与我共享，然后在亚马逊机器映像（AMI）中选择一个 AMI。

快速入门

AMI 按操作系统 (OS) 分组以助您快速入门。

首先选择所需的操作系统，然后从亚马逊机器映像（AMI）中选择一个 AMI。要选择可以在 Amazon 免费套餐下使用的 AMI，请确保该 AMI 已标记符合免费套餐资格。

浏览更多 AMI

选择 Browse more AMIs（浏览更多 AMI）以浏览完整的 AMI 目录。

- 要搜索所有可用的 AMI，请在搜索栏中输入关键字，然后按 Enter 键。
- 要使用 Systems Manager 参数查找 AMI，请选择搜索栏右侧的箭头按钮，然后选择 Search by Systems Manager parameter（按 Systems Manager 参数搜索）。有关更多信息，请参阅 使用 Systems Manager 参数引用 AMI。
- 要按类别搜索，请依次选择 Quickstart AMI、My AMIs（我的 AMI）、Amazon Web Services Marketplace AMI 或者 Community AMIs（社区 AMI）。

  Amazon Web Services Marketplace 是一个在线商店，您可以从中购买在 Amazon 上运行的软件（包括 AMI）。有关从 Amazon Web Services Marketplace 启动实例的更多信息，请参阅从 Amazon Web Services Marketplace AMI 中启动 Amazon EC2 实例。在 Community AMIs（社区 AMI）中，您可以找到 Amazon 社区成员提供给其它人使用的 AMI。将来自 Amazon 或经过验证的合作伙伴的 AMI 标记为经过验证的提供商。
- 要筛选 AMI 列表，请在屏幕左侧的 Refine results（优化结果）下方选中一个或多个复选框。筛选条件选项会因所选搜索类别而有所不同。
- 检查对每个 AMI 列出的 Root device type (根设备类型)。请注意哪些 AMI 是您需要的类型：即 ebs（由 Amazon EBS 支持）或 instance-store（由实例存储支持）。有关更多信息，请参阅 根卷类型。
- 检查对每个 AMI 列出的 Virtualization type (虚拟化类型)。注意哪些 AMI 类型是您需要的类型：即 hvm（全虚拟化）或 paravirtual（半虚拟化）。例如，一些实例类型需要 HVM。有关 Linux 虚拟化类型更多信息，请参阅 虚拟化类型。
- 检查对每个 AMI 列出的启动模式。请注意哪些 AMI 使用您需要的启动模式：legacy-bios、uefi 或 uefi-preferred。有关更多信息，请参阅 使用 Amazon EC2 启动模式的实例启动行为。
- 选择满足您的需求的 AMI，然后选择 Select。

更改 AMI 时发出警告

启动实例时，如果您修改了与所选 AMI 关联的任何卷或安全组的配置，然后选择了其他 AMI，则会打开一个窗口，警告您当前的某些设置将被更改或删除。您可以查看对安全组和卷的更改。此外，您可以查看将添加和删除哪些卷，也可以仅查看要添加的卷。创建启动模板时不会出现此警告。

实例类型

实例类型定义了实例的硬件配置和大小。更大的实例类型拥有更多的 CPU 和内存。有关更多信息，请参阅 Amazon EC2 实例类型。

- Instance type (实例类型)：确保实例类型与指定的 AMI 兼容。有关更多信息，请参阅 Amazon EC2 实例类型。
- 获取建议：您可以从 EC2 实例类型查找器中获取实例类型指导和建议。有关更多信息，请参阅 从 EC2 实例类型查找器获取建议。
- （仅限启动模板）高级：要指定实例属性并让 Amazon EC2 使用这些属性识别实例类型，请选择高级，然后选择指定实例类型属性。
  - Number of vCPUs（vCPU 量）：输入您的计算要求的最小 vCPU 数和最大 vCPU 数。若要表示为无限制，请输入最小值为 **0**，然后将最大值留空。
  - Amount of memory (MiB) (内存量 (MiB))：输入您的计算要求的最小内存量和最大内存量（以 MiB 为单位）。若要表示为无限制，请输入最小值为 **0**，然后将最大值留空。
  - 展开 Optional instance type attributes（可选的实例类型属性），然后选择 Add attribute（添加属性）以更详细地表达您的计算需求。有关每个属性的信息，请参阅 Amazon EC2 API 参考中的 InstanceRequirementsRequest。
  - Resulting instance types（生成的实例类型）：您可以预览与指定属性匹配的实例类型。若要排除实例类型，请选择 Add attribute（添加属性），并从 Attribute（属性）列表中选择 Excluded instance types（排除的实例类型）。从 Attribute value（属性值）列表中，选择要排除的实例类型。

密钥对（登录）

为 Key pair name（密钥对名称）选择一个现有密钥对，或选择 Create new key pair（创建新密钥对）来新建一个密钥对。有关更多信息，请参阅 Amazon EC2 密钥对和 Amazon EC2 实例。

> **Important**
>
> 如果您选择 Proceed without key pair (Not recommended)（在没有密钥对的情况下继续（不推荐））选项，则将无法连接到此实例，除非您选择配置为允许用户以其它方式登录的 AMI。

网络设置

网络设置定义实例的 IP 地址、安全组和网络接口。您可以使用默认网络设置或根据需要进行配置。

- （仅限启动实例向导）VPC：为您的实例选择现有 VPC。默认选中该区域的默认 VPC。或者，您可以选择自己创建的或与您共享的 VPC。有关更多信息，请参阅 适用于您的 EC2 实例的虚拟私有云。
- 子网：为实例选择子网，或者选择创建新子网，使用 Amazon VPC 控制台创建新子网。
  - 您可以在任何可用区、本地区域、Wavelength 区域或 Outpost 区域内为所选 VPC 创建子网。
  - 要在仅使用 IPv6 的子网中启动实例，该实例必须是基于 Nitro 的实例。
- （仅限启动实例向导）自动分配公有 IP：启用或禁用自动分配公有 IPv4 地址。在默认子网中启动实例时，默认值为启用。在非默认子网中启动实例时，默认值为禁用。有关更多信息，请参阅 公有 IPv4 地址。

  如果添加辅助网络接口，则无法为非默认子网启用此选项。有关更多信息，请参阅 the section called “在启动时分配公有 IPv4 地址”。
- （仅限启动实例向导）自动分配 IPv6 IP：启用或禁用自动分配 IPv6 地址。有关更多信息，请参阅 IPv6 地址。
- 防火墙（安全组）：选择现有安全组或创建一个新安全组。确保您的安全组有允许流量进出实例的规则。所有其他的流量将被忽略。

  如果您创建一个新安全组，我们会自动创建一条入站规则，允许您通过 SSH（Linux 实例）或 RDP（Windows 实例）从所有 IP 地址连接到您的实例。您可以根据需要删除或者修改此规则。您可以根据需要添加规则。有关更多信息，请参阅 配置安全组规则。

  > **Warning**
  >
  > 如果您只是短暂地启动测试实例并将很快停止或终止该实例，那么允许所有 IP 地址通过 SSH 或 RDP 访问实例的规则是可以接受的，但这些规则对生产环境来说是不安全的。您应该仅授权特定 IP 地址范围访问您的实例。

  此安全组将添加到主网络接口和所有辅助网络接口。您可以为网络接口选择其他安全组，但不能删除在此处选择的安全组。
- 高级网络配置：您可以根据需要配置主网络接口。要添加辅助网络接口，请选择添加网络接口。您可以添加的网络接口数量取决于所选实例的类型。请注意，此部分仅在选择子网时可用。
  - 设备索引：设备的索引。主网络接口必须分配给索引 0。

- 网络接口：网络的接口。选择新接口让 Amazon EC2 创建新的接口，或选择现有的可用网络接口。如果您选择现有网络接口作为主网络接口，则无法为非默认子网启用自动分配公有 IP。
- 描述：新网络接口的描述。
- Subnet（子网）：要在其中创建新网络接口的子网。实例在与主网络接口相同的子网中启动。

  您必须从与主网络接口子网相同的可用区选择辅助网络接口的子网。如果选择其他 VPC 的子网，该网络接口旁将会显示多 VPC 标签。这使您能够在具有不同网络和安全配置的 VPC 之间创建多宿主实例。

  要在仅使用 IPv6 的子网中启动 EC2 实例，您必须使用基于 Nitro 的实例。启动仅 IPv6 实例时，DHCPv6 可能不会立即为该实例提供 IPv6 DNS 名称服务器。在此初始延迟期间，实例可能无法解析公有域。您可以更改配置文件并重新映像 AMI，以便该文件在启动时立即具有 IPv6 DNS 名称服务器地址。
- 安全组：要与该网络接口关联的安全组。您必须从与网络接口子网相同的 VPC 中选择安全组。
- （仅限启动模板）自动分配公有 IP：指定您的实例是否接收公有 IPv4 地址。默认情况下，默认子网中的实例会收到公有 IPv4 地址，而非默认子网中的实例不会收到。可以选择 Enable (启用) 或 Disable (禁用) 以覆盖子网的默认设置。有关更多信息，请参阅 公有 IPv4 地址。
- Primary IP (主要 IP)：您的子网范围内的一个私有 IPv4 地址。保留空白会让 Amazon EC2 为您选择一个私有 IPv4 地址。
- 辅助 IP：子网范围内的其他私有 IPv4 地址。选择手动分配，然后输入 IPv4 地址。选择添加 IP 添加另一个 IPv4 地址。或者，选择自动分配并输入一个值，来指示 Amazon EC2 为您选择的 IPv4 地址数量。
- （仅限 IPv6）IPv6 IP：子网范围内的 IPv6 地址。选择手动分配，然后输入 IPv6 地址。选择添加 IP 添加另一个 IPv6 地址。或者，选择自动分配并输入一个值，来指示 Amazon EC2 为您选择的 IPv6 地址数量。
- IPv4 前缀：网络接口的 IPv4 前缀。选择手动分配，然后输入 IPv4 前缀。或者，选择自动分配并输入一个值，来指示 Amazon EC2 为您选择的 IPv4 前缀数量。
- IPv6 前缀：网络接口的 IPv6 前缀。选择手动分配，然后输入 IPv6 前缀。或者，选择自动分配并输入一个值，来指示 Amazon EC2 为您选择的 IPv6 前缀数量。
- （双堆栈和仅 IPv6）分配主 IPv6 IP：如果您选择双堆栈或仅 IPv6 子网，则分配主 IPv6 地址。这有助于防止到实例或网络接口的流量中断。如果您希望 IPv6 地址不变，则启用此选项。以后将无法移除主 IPv6 地址。当您将 IPv6 GUA 地址启用为主 IPv6 时，第一个 IPv6 GUA 将成为主 IPv6 地址，直到实例终止或网络接口断开为止。如果您有多个 IPv6 地址与网络接口关联，并且可让 Amazon EC2 分配主 IPv6 地址，则与该网络接口关联的第一个 IPv6 GUA 地址将成为主 IPv6 地址。

- 终止时删除：指示在删除实例时是否删除网络接口。
- 接口类型：网络接口类型：
  - ENA：一种高性能网络接口，旨在满足高吞吐量和每秒数据包速率的 TCP/IP 协议处理需求，同时尽可能降低 CPU 使用率。这是默认值。有关更多信息，请参阅弹性网络适配器。
  - 带 ENA 的 EFA：一种同时支持 ENA 和 EFA 设备的网络接口，用于传统基于 TCP/IP 的传输以及基于 SRD 的传输。有关 EFA 的更多信息，请参阅 Elastic Fabric Adapter。
  - 仅 EFA：一种高性能网络接口，旨在满足基于 SRD 的传输对高吞吐量、低延迟节点间通信的处理需求，同时避开操作系统堆栈。仅 EFA 网络接口不支持 IP 地址。有关 EFA 的更多信息，请参阅 Elastic Fabric Adapter。
- Elastic Fabric Adapter：指示网络接口是否为 Elastic Fabric Adapter。有关更多信息，请参阅 适用于 Amazon EC2 上 AI/ML 和 HPC 工作负载的 Elastic Fabric Adapter。
- 网卡索引 ：网卡的索引。必须将主网络接口分配给网卡索引 0。有些实例类型支持多个网卡。
- ENA Express：ENA Express 由 Amazon 可扩展的可靠数据报（SRD）技术提供支持。SRD 技术使用数据包散射机制来分配负载并避免网络拥塞。启用 ENA Express 允许支持的实例在可能的情况下在常规 TCP 流量之上使用 SRD 进行通信。除非您从列表中选择启用或禁用，否则启动实例向导或启动模板不包括实例的 ENA Express 配置。
- ENA Express UDP：如果您已启用 ENA Express，则可以选择将其用于 UDP 流量。除非您选择启用或禁用，否则启动实例向导或启动模板不包括实例的 ENA Express 配置。

配置存储

您选择的 AMI 包含一个或多个存储卷，包括根卷。您可以指定要附加到实例的其它卷。

（仅限启动实例向导）您可以使用简单或高级视图。借助 Simple（简单）视图，您可以指定卷的大小和类型。若要指定所有卷参数，请选择位于卡片的右上角的 Advanced（高级）视图。

在 Advanced（高级）视图中，您可以按如下方式配置每个卷：

- Storage type（存储类型）：选择要与实例关联的 Amazon EBS 或实例存储卷。列表中可用的卷类型取决于您选择的实例类型。有关更多信息，请参阅 适用于 EC2 实例的实例存储临时块存储 和 Amazon EBS 卷。
- Device name（设备名称）：从卷的可用设备名称列表中进行选择。
- Snapshot（快照）：选择要从其中还原卷的快照。您可以通过在 Snapshot（快照）字段中输入文本来搜索可用的共享快照和公有快照。
- Size (GiB)（大小 (GiB)）：对于 EBS 卷，您可以指定存储大小。

- 卷类型：对于 EBS 卷，请选择卷类型。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 卷类型。
- IOPS：如果选择了 io1、io2 或 gp3 卷类型，则可以输入卷支持的每秒 I/O 操作数 (IOPS)。io1、io2 和 gp3 卷必需。gp2、st1、sc1 或标准卷不支持。如果您省略了启动模板中的此参数，必须在从启动模板启动实例时指定其值。
- Delete on termination（终止时删除）：对于 Amazon EBS 卷，选择 Yes（是）以在终止实例时删除此卷或选择 No（否）以保留此卷。有关更多信息，请参阅 实例终止时保留数据。
- Encrypted（加密）：如果实例类型支持 EBS 加密，则可以选择 Yes（是）以为此卷启用加密。如果默认情况下在此区域中启用了加密，则会为您启用加密。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 加密。
- KMS key（KMS 密钥）：如果您将 Encrypted（加密）选择为 Yes（是），则您必须选择一个客户托管式密钥来加密卷。如果默认情况下在此区域中启用了加密，则将为您选择默认的客户托管密钥。您可以选择不同的密钥或指定由您创建的任何客户托管密钥的 ARN。
- 吞吐量：如果选择 gp3 卷类型，则可以输入该卷支持的吞吐量（以 MiB/s 为单位）。
- 卷初始化速率：选择某个快照后，您可以选择指定卷初始化速率（以 MiB/s 为单位），亦即将快照块从 Amazon S3 下载到卷的速率。有关更多信息，请参阅 Use an Amazon EBS Provisioned Rate for Volume Initialization。要使用默认初始化速率或快速快照还原功能（如果已为所选快照启用此功能），请不要指定速率。
- File systems（文件系统）：将 Amazon EFS 或 Amazon FSx 文件系统挂载到实例。有关挂载 Amazon EFS 文件系统的更多信息，请参阅 将 Amazon EFS 与 Amazon EC2 Linux 实例结合使用。有关挂载 Amazon FSx 文件系统的更多信息，请参阅 将 Amazon FSx 与 Amazon EC2 实例结合使用

高级详细信息

对于高级详细信息，请展开该部分以查看字段并为实例指定任何其他参数。

- （仅限启动实例向导）域加入目录：选择您的实例在启动后加入到的 Amazon Directory Service 目录（域）。如果选择一个域，则必须选择一个具有所需权限的 IAM 角色。有关域加入域更多信息，请参阅将 Amazon EC2 Linux 实例无缝加入 Amazon 托管 Microsoft AD 目录（Linux 实例）和将 Amazon EC2 Windows 实例无缝加入 Amazon 托管 Microsoft AD 目录（Windows 实例）。
- IAM 实例配置文件：选择要与实例关联的 IAM 实例配置文件。这是 IAM 角色的容器。有关更多信息，请参阅 适用于 Amazon EC2 的 IAM 角色。
- Hostname type（主机名类型）：选择实例的来宾操作系统主机名将包括资源名称还是 IP 名称。有关更多信息，请参阅 EC2 实例主机名和域。

- DNS Hostname（DNS 主机名）：确定对资源名称或 IP 名称的 DNS 查询（根据您对 Hostname type（主机名类型）的选择）是否将以 IPv4 地址（A 记录）、IPv6 地址（AAAA 记录）响应，还是同时以两者响应。有关更多信息，请参阅 EC2 实例主机名和域。
- 实例自动恢复：启用后，如果系统状态检查失败，则会恢复您的实例。默认情况下，对于支持的实例类型，此设置在启动时启用。有关更多信息，请参阅 在 Amazon EC2 实例上配置简化的自动恢复。
- Shutdown behavior (关闭行为)：选择关闭时实例应该停止还是终止。有关更多信息，请参阅 更改实例启动的关闭行为。
- Stop - Hibernate behavior（停止 – 休眠行为）：要启用休眠，请选择 Enable（启用）。只有当实例满足休眠先决条件时，此字段才可用。有关更多信息，请参阅 将您的 Amazon EC2 实例休眠。
- Termination protection（终止保护）：要防止意外终止，请选择 Enable（启用）。有关更多信息，请参阅 更改实例终止保护。
- Stop protection（停止保护）：为防止意外停止，请选择 Enable（启用）。有关更多信息，请参阅 启用停止保护。
- Detailed CloudWatch monitoring（详细的 CloudWatch 监控）：选择 Enable（启用）将使用 Amazon CloudWatch 开启对您的实例的详细监控。将收取额外费用。有关更多信息，请参阅 使用 CloudWatch 监控您的实例。
- Credit specification（积分规范）：选择 Unlimited（无限）以允许应用程序只要有需要即突增到基准以上。此字段仅适用于 T 实例。可能收取额外费用。有关更多信息，请参阅 可突增性能实例。
- 置放群组：指定要在其中启动实例的置放群组。您可以选择现有置放群组或创建新组。并非所有实例类型都支持在放置群组中启动实例。有关更多信息，请参阅 Amazon EC2 实例的置放群组。
- EBS-optimized instance（EBS 优化的实例）：Amazon EBS 优化的实例使用优化的配置堆栈，并为 Amazon EBS I/O 提供额外的专用容量。如果实例类型支持此功能，请选择 Enable（启用）来启用该功能。将收取额外费用。有关更多信息，请参阅 the section called “EBS 优化”。
- 实例带宽配置：您可以增加网络带宽或 EBS 带宽。仅适用于支持的实例类型。有关更多信息，请参阅 EC2 实例带宽权重配置。
- 购买选项：选择竞价型实例即可按照竞价型价格请求竞价型实例，上限为按需价格，选择自定义竞价型实例选项即可更改默认的竞价型实例设置。您可以设置最高价（不建议），并更改请求类型、请求时长和中断行为。如果您未请求竞价型实例，则默认情况下 Amazon EC2 会启动按需型实例。有关更多信息，请参阅 管理您的竞价型实例。
- Capacity Reservation（容量预留）：指定是在任何开放容量预留 [Open（开放）]、特定容量预留 [Target by ID（按 ID 定位）]，还是容量预留组 [Target by group（按组定位）] 中启动实例。要指定不使用容量预留，请选择 None（无）。有关更多信息，请参阅 在现有 容量预留 中启动实例。
- Tenancy (租期)：选择是在共享硬件（Shared (共享)）、隔离的专用硬件（Dedicated (专用)），还是在 专用主机（Dedicated host (专用主机)）上运行您的实例。如果您选择在专用主机上启动实例，

则可以指定是否在主机资源组中启动实例，也可以定位特定专用主机。可能收取额外费用。有关更多信息，请参阅 Amazon EC2 专用实例 和 Amazon EC2 专属主机。

- RAM disk ID（RAM 磁盘 ID）：[仅对半虚拟 (PV) AMI 有效] 为实例选择一个 RAM 磁盘。如果您选择了一个内核，则您可能需要选择带有可支持该内核的驱动程序的某个特定 RAM 磁盘。
- Kernel ID（内核 ID）：[仅对半虚拟 (PV) AMI 有效] 为实例选择一个内核。
- Nitro Enclave：允许您从 Amazon EC2 实例创建隔离的执行环境，称为 Enclave。选择 Enable（启用）以启用 Amazon Nitro Enclaves 的实例。有关更多信息，请参阅 Amazon Nitro Enclaves 用户指南中的什么是 Amazon Nitro Enclaves？
- 许可证配置：您可以根据指定的许可证配置启动实例，以跟踪您的许可证使用情况。有关更多信息，请参阅《Amazon License Manager 用户指南》中的创建许可证配置。
- 指定 CPU 选项：在启动实例向导中，仅当所选实例类型支持指定 CPU 选项时，此字段才可见。选择指定 CPU 选项可在启动期间指定自定义 vCPU 数。设置 CPU 内核数和每内核线程数。有关更多信息，请参阅 Amazon EC2 实例适用的 CPU 选项。
- 可访问的元数据：您可以启用或禁用对实例元数据服务（IMDS）的访问权限。有关更多信息，请参阅 为新实例配置实例元数据选项。
- 元数据 IPv6 端点：让实例能够使用 IMDS IPv6 地址 [fd00:ec2::254] 来检索实例元数据。此选项仅当您在支持 IPv6 的子网（双栈或仅 IPv6）中启动基于 Nitro 的实例时可用。有关检索实例元数据的更多信息，请参阅 访问 EC2 实例的实例元数据。
- 元数据版本：如果您启用对 IMDS 的访问权限，可以选择在请求实例元数据时要求使用实例元数据服务版本 2。有关更多信息，请参阅 为新实例配置实例元数据选项。
- 元数据响应跃点限制：如果您启用 IMDS，可以为元数据标记设置允许的网络跃点数。有关更多信息，请参阅 为新实例配置实例元数据选项。
- Allow tags in metadata（允许元数据中的标签）：如果选择Enable（启用），该实例将允许从其元数据访问其所有标签。如果未指定值，则默认情况下，将不允许对实例元数据中的标签的访问。有关更多信息，请参阅 启用对实例元数据中标签的访问权限。
- User data：您可以指定用户数据在启动时配置实例或运行配置脚本。有关 Linux 实例用户数据的更多信息，请参阅 在启动包含用户数据输入的 EC2 实例时运行命令。有关 Windows 实例用户数据的更多信息，请参阅 Amazon EC2 如何处理 Windows 实例的用户数据。

摘要

使用 Summary（摘要）面板来指定要启动的实例数、查看实例配置和启动实例。

- Number of instances (实例的数量)：输入要启动的实例的数量。所有实例都将以相同的配置启动。

> **Tip**
>
> 为确保更快地启动实例，请将大量请求分成较小的批次。例如，创建五个独立的请求批次，每个批次包含 100 个实例启动请求，而不要创建一个包含 500 个实例的启动请求。

- （可选）如果您指定的实例不止一个，为了帮助确保保持正确数量的实例来处理应用程序的需求，您可以选择 consider EC2 Auto Scaling（考虑 EC2 Auto Scaling）以创建启动模板和 Auto Scaling 组。Auto Scaling 将根据您的规格来扩展组中的实例数。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南。

> **Note**
>
> 如果 Amazon EC2 Auto Scaling 将某个 Auto Scaling 组中的实例标记为运行状况不佳，则会自动安排该实例进行替换，此时将终止此实例而启动另一个实例，并且您将丢失原始实例上的数据。如果您停止或重新引导实例，或者其他事件将实例标记为运行状况不佳，则此实例将被标记为运行状况不佳。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的自动扩缩组中实例的运行状况检查。

- 查看实例的详细信息并进行必要的更改。您可以在 Summary（摘要）面板选择某部分的链接以直接导航到该部分。
- 当您准备好启动您的实例时，请选择 Launch instance（启动实例）。

使用控制台中的启动实例向导来启动 EC2 实例

您可以使用 Amazon EC2 控制台中的启动实例向导启动 Amazon EC2 实例。该向导提供了启动参数的默认值，您可以接受这些值，也可以修改这些值以满足您的要求。唯一未指定的参数是密钥对。如果您选择接受默认值，则可以通过仅选择密钥对来快速启动实例。

> **Important**
>
> 实例处于 `running` 状态时，即使其一直处于空闲状态，您也需为其付费。

有关启动实例向导中每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

主题

- 快速启动实例

- 使用定义的参数启动实例

快速启动实例

如需快速设置实例以进行测试，请按照以下步骤。您需要选择操作系统和密钥对，然后接受默认值。除了密钥对外，启动实例向导还会为所有参数提供默认值。您可以接受任何或全部默认值，也可以通过为每个参数指定自己的值来配置实例。

有关启动实例向导中每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

使用启动实例向导快速启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，会显示当前 Amazon 区域，例如，美国东部美国东部（俄亥俄州）。如果需要，选择要在其中启动实例的不同区域。
3. 从 Amazon EC2 控制台控制面板中，选择启动实例。
4. （可选）在 Name and tags（名称与标签）下，为 Name（名称）输入实例的描述性名称。
5. 在 Application and OS Images（Amazon Machine Image）（应用程序和操作系统镜像（亚马逊机器映像））下，选择 Quick Start（快速启动），然后为您的实例选择操作系统（OS）。
6. （可选）在 Key pair (login)（密钥对（登录））下，为 Key pair name（密钥对名称）选择一个现有密钥对或新建一个密钥对。
7. 在 Summary（摘要）面板中，选择 Launch instance（启动实例）。

使用定义的参数启动实例

如果您要启动将在生产中使用的实例，则需要配置该实例以满足您的要求。有关启动实例向导中每个参数的说明，请参阅Amazon EC2 实例配置参数参考。

使用启动实例向导通过定义所有启动参数来启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在屏幕顶部的导航栏中，会显示当前 Amazon 区域，例如，美国东部美国东部（俄亥俄州）。如果需要，选择要在其中启动实例的不同区域。
3. 从 Amazon EC2 控制台控制面板中，选择启动实例。
4. （可选）在名称和标签下的名称中，输入实例的描述性名称，以便您可以轻松地对其进行跟踪。

实例名称是一个标签，其中密钥为名称，而值为您指定的名称。

5. 在应用程序和操作系统镜像（亚马逊机器映像）下，为您的实例选择操作系统（OS），然后选择 AMI。

   AMI 是一个模板，其中包含启动实例所需的操作系统和软件。

6. 在 Instance type（实例类型）下，选择一个实例类型。

   实例类型决定了用于实例的硬件配置（CPU、内存、存储和网络容量）和主机大小。

   如果不确定要选择哪种实例类型，则可以执行以下操作：

   - 选择获取建议，从 EC2 实例类型查找器中获取实例类型的指导和建议。有关更多信息，请参阅从 EC2 实例类型查找器获取建议。

7. （可选）在密钥对（登录）下，为密钥对名称选择一个现有密钥对或新建一个密钥对。如果您不需要密钥对即可连接到实例，则可以选择在没有密钥对的情况下继续（不推荐）。

8. 在网络设置下，如果您要启动测试实例，则可以保留默认设置。如果您要启动生产实例，则最佳实践是使用您定义的网络设置和安全组来控制进出实例的流量。

9. 在配置存储下，您可以保留默认值或指定其他存储。您选择的 AMI 包含一个或多个存储卷，包括根卷。您可以指定要附加到实例的其它卷。

   您可以使用 Simple（简单）或 Advanced（高级）视图。借助 Simple（简单）视图，您可以指定卷的大小和类型。若要指定所有卷参数，请选择位于卡片的右上角的 Advanced（高级）视图。

10. 对于高级详细信息，请展开该部分以查看字段并为实例指定任何其他参数。

11. 在摘要面板中，您可以执行以下操作：

    a. 指定要启动的实例数量。

    b. 查看实例配置，选择某部分的链接以直接导航到该部分。

    c. 当您准备好启动您的实例时，请选择 Launch instance（启动实例）。

    如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

12. （可选）您可以为实例创建账单提醒。在确认屏幕上的 Next Steps（下一步）下选择 Create billing alerts（创建账单提醒）并按照指示操作。还可以在启动实例后创建账单提醒。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的创建账单告警以监控预估 Amazon 费用。

使用启动模板来启动 EC2 实例

Amazon EC2 启动模板可存储实例启动参数，以便无需在每次启动实例时都指定这些参数。

一些实例启动服务在启动实例时可以选择使用启动模板，而对于其他服务，如 EC2 实例集，除非使用启动模板，否则无法启动实例。本主题介绍如何在使用 EC2 启动实例向导、Amazon EC2 Auto Scaling、EC2 实例集和竞价型实例集启动实例时，使用启动模板。

有关启动模板的更多信息，包括如何创建启动模板，请参阅在 Amazon EC2 启动模板中存储实例启动参数。

主题

- 使用启动模板启动 Amazon EC2 实例
- 使用启动模板在 Amazon EC2 Auto Scaling 组中启动实例
- 使用启动模板启动 EC2 实例集
- 使用启动模板启动竞价型实例集

使用启动模板启动 Amazon EC2 实例

您可以使用启动模板中包含的参数启动 Amazon EC2 实例。选择启动模板后，但在启动实例之前，您可以修改启动参数。

将自动为使用启动模板启动的实例分配两个具有 `aws:ec2launchtemplate:id` 和 `aws:ec2launchtemplate:version` 键的标签。您无法删除或编辑这些标签。

Console

使用启动模板启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 使用以下选项之一选择启动模板：
   - 从 Amazon EC2 控制台的控制面板中，选择启动实例旁边的向下箭头，选择从模板启动实例，然后对源模板选择启动模板。
   - 在导航窗格中，选择启动模板，选择该启动模板，然后选择操作、从模板启动实例。
3. 对于 Source template version (源模板版本)，选择要使用的启动模板版本。
4. （可选）您可以修改任何启动参数的值。如果您未修改值，则使用启动模板定义的值。如果启动模板中未指定任何值，则使用该参数的默认值。

5. 在摘要面板中，对于实例数，输入要启动的实例数量。
6. 选择启动实例。

   如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

Amazon CLI

通过启动模板启动实例

- 使用 run-instances 命令并指定 `--launch-template` 参数。可以选择指定要使用的启动模板版本。如果未指定版本，则使用默认版本。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- 要覆盖启动模板参数，请在 run-instances 命令中指定该参数。以下示例覆盖在启动模板 (如果有) 中指定的实例类型。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --instance-type t2.small
```

- 如果指定复杂结构包含的嵌套参数，则使用启动模板中指定的复杂结构以及您指定的任何其他嵌套参数启动实例。

  在以下示例中，将使用标签 *Owner*=*TeamA* 以及在启动模板中指定的任何其他标签启动实例。如果启动模板包含具有 *Owner* 键的现有标签，该值将替换为 *TeamA*。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
    --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]"
```

  在以下示例中，将使用具有设备名称 */dev/xvdb* 的卷以及在启动模板中指定的任何其他块设备映射启动实例。如果启动模板为 */dev/xvdb* 定义了一个现有卷，则其值将替换为指定的值。

```
aws ec2 run-instances \
    --launch-template LaunchTemplateId=lt-0abcd290751193123 \
```

```
    --block-device-mappings "DeviceName=/dev/
xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

PowerShell

使用 Amazon Tools for PowerShell 通过启动模板启动实例

- 使用 New-EC2Instance 命令并指定 `-LaunchTemplate` 参数。可以选择指定要使用的启动模板版本。如果未指定版本，则使用默认版本。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
)
```

- 要覆盖启动模板参数，请在 New-EC2Instance 命令中指定该参数。以下示例覆盖在启动模板 (如果有) 中指定的实例类型。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
)
```

- 如果指定复杂结构包含的嵌套参数，则使用启动模板中指定的复杂结构以及您指定的任何其他嵌套参数启动实例。

  在以下示例中，将使用标签 *Owner*=*TeamA* 以及在启动模板中指定的任何其他标签启动实例。如果启动模板包含具有 *Owner* 键的现有标签，该值将替换为 *TeamA*。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
        }
) `
    -TagSpecification (
        New-Object -TypeName Amazon.EC2.Model.TagSpecification -Property @{
            ResourceType = 'instance';
            Tags         = @(
                @{key = "Owner"; value = "TeamA" },
                @{key = "Department"; value = "Operations" }
            )
        }
)
```

在以下示例中，将使用具有设备名称 */dev/xvdb* 的卷以及在启动模板中指定的任何其他块设备映射启动实例。如果启动模板为 */dev/xvdb* 定义了一个现有卷，则其值将替换为指定的值。

```
Import-Module AWS.Tools.EC2
New-EC2Instance `
    -InstanceType t4g.small  `
    -LaunchTemplate (
        New-Object -TypeName Amazon.EC2.Model.LaunchTemplateSpecification -
Property @{
            LaunchTemplateId = 'lt-0abcd290751193123';
            Version          = '4'
    }
) `
    -BlockDeviceMapping  (
        New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping -Property @{
            DeviceName = '/dev/xvdb';
            EBS        = (
                New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property @{
                    VolumeSize = 25;
                    VolumeType = 'gp3'
                }
            )
```

```
        }
)
```

如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

使用启动模板在 Amazon EC2 Auto Scaling 组中启动实例

您可以创建一个 Auto Scaling 组，并指定一个用于该组的启动模板。在 Amazon EC2 Auto Scaling 启动 Auto Scaling 组中的实例时，它使用关联的启动模板中定义的启动参数。

在可以使用启动模板创建自动扩缩组之前，您必须先创建启动模板，其中包括启动自动扩缩组中的实例所需的参数。有些参数是必需的，例如 AMI 的 ID，有些参数不可用于自动扩缩组。控制台提供指导，以帮助您创建可与 Amazon EC2 Auto Scaling 结合使用的模板。

使用控制台通过启动模板创建自动扩缩组

- 有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Create an Auto Scaling group using a launch template。

使用 Amazon CLI 通过启动模板创建或更新自动扩缩组

- 使用 create-auto-scaling-group 或 update-auto-scaling-group 命令，并指定 `--launch-template` 参数。

有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的以下主题：

- 为自动扩缩组创建启动模板
- 使用高级设置创建启动模板
- Examples for creating and managing launch templates with the Amazon Command Line Interface（Amazon CLI）：提供展示如何使用各种参数组合创建启动模板的示例。
- 使用启动模板创建自动扩缩组
- 更新自动扩缩组

使用启动模板启动 EC2 实例集

创建 EC2 实例集请求时，启动模板是必需的。在 Amazon EC2 完成 EC2 队列 请求时，它使用关联的启动模板中定义的启动参数。您可以覆盖启动模板中指定的某些参数。有关更多信息，请参阅创建 EC2 实例集。

使用 Amazon CLI 通过启动模板创建 EC2 实例集

- 使用 create-fleet 命令。请使用 `--launch-template-configs` 参数指定启动模板，并为启动模板指定任何覆盖值。

使用启动模板启动竞价型实例集

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

创建竞价型实例集请求时，启动模板是可选。如果您未使用启动模板，则可以手动指定启动参数。如果使用启动模板，则在 Amazon EC2 完成竞价型实例集请求时，它会使用关联的启动模板中定义的启动参数。您可以覆盖启动模板中指定的某些参数。有关更多信息，请参阅 创建竞价型实例集。

使用启动模板创建竞价型实例集请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择请求 Spot 实例。
4. 在 Launch parameters（启动参数）下，选择 Use a launch template（使用启动模板）。
5. 对于 Launch template（启动模板），选择一个启动模板，然后从右侧字段中选择启动模板版本。
6. 在此屏幕上选择不同的选项来配置竞价型实例集。有关选项的更多信息，请参阅使用已定义的参数创建竞价型实例集请求。
7. 准备好创建竞价型实例集后，请选择 Launch（启动）。

使用启动模板创建竞价型实例集请求

- 使用 request-spot-fleet 命令。请使用 `LaunchTemplateConfigs` 参数指定启动模板，并为启动模板指定任何覆盖值。

使用现有实例的详细信息来启动 EC2 实例

Amazon EC2 控制台提供了 Launch more like this（启动更多类似实例）选项，允许将当前实例作为基础来启动其他实例。该选项自动使用所选实例中的特定配置详细信息来填充 Amazon EC2 启动实例向导。

注意事项

- 我们不会克隆您的实例；我们只复制一些配置详细信息。要创建实例的副本，请先从它创建 AMI，然后从 AMI 启动更多实例。创建启动模板，以确保使用相同的启动详细信息启动实例。
- 当前实例必须处于 `running` 状态。

已复制详细信息

以下配置详细信息会从所选实例复制到启动实例向导中：

- AMI ID
- 实例类型
- 可用区，或所选实例所在的 VPC 和子网
- 公有 IPv4 地址。如果所选实例当前具有公有 IPv4 地址，则无论所选实例的默认公有 IPv4 地址设置如何，新实例都会收到公有 IPv4 地址。有关公有 IPv4 地址的更多信息，请参阅公有 IPv4 地址。
- 置放群组，如果适用
- 与实例关联的 IAM 角色 (如果适用)
- 关闭操作设置 (停止或中止)
- 终止保护设置 (true 或 false)
- CloudWatch 监控 (启用或禁用)
- Amazon EBS 优化的设置 (true 或 false)
- 租期设置 (如果在 VPC (共享或专用) 中启动)
- 内核 ID 和 RAM 磁盘 ID (如果适用)
- 用户数据，如果指定

- 与实例关联的标签 (如果适用)
- 与实例关联的安全组
- [Windows 实例] 关联信息。如果所选实例已与某个配置文件关联，那么该文件会自动与新实例关联。如果该配置文件包含加入的域配置，则新实例将加入到同一个域中。有关加入域的更多信息，请参阅《Amazon Directory Service Administration Guide》中的 Seamlessly join a Windows EC2 instance to your Amazon Managed Microsoft AD Active Directory。

未复制详细信息

向导不会从所选实例中复制以下配置详细信息，而会应用其默认设置或行为：

- 网络接口数量：默认为一个网络接口，即主网络接口（eth0）。
- 存储：默认存储配置由 AMI 和实例类型确定。

启动更多实例（如现有实例）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择操作、映像和模板、启动更多类似实例。
4. 此时会打开启动实例向导。通过在此屏幕上选择不同的选项，您可以对实例配置进行必要的更改。

   当您准备好启动实例时，请选择 Launch instance（启动实例）。
5. 如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

从 Amazon Web Services Marketplace AMI 中启动 Amazon EC2 实例

您可以订阅 Amazon Web Services Marketplace AMI 并使用 Amazon EC2 控制台或命令行工具从中启动实例。有关 Amazon Web Services Marketplace AMI 的更多信息，请参阅Amazon Web Services Marketplace 中适用于 Amazon EC2 实例的付费 AMI。

要在启动后取消对 AMI 的订阅，必须先终止从该 AMI 启动的所有实例。有关更多信息，请参阅 管理您的 Amazon Web Services Marketplace 订阅。

使用 Amazon EC2 控制台从 Amazon Web Services Marketplace AMI 启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 从 Amazon EC2 控制台控制面板中，选择启动实例。
3. （可选）在 Name and tags（名称与标签）下，为 Name（名称）输入实例的描述性名称。
4. 在应用程序和操作系统映像（Amazon 机器映像）下，选择浏览更多 AMI，然后选择 Amazon Web Services Marketplace AMI 选项。通过浏览类别或使用搜索功能找到合适的 AMI。选择 Select（选择）以选择产品。
5. 系统将打开一个窗口，显示所选产品的概览。您可以查看定价信息，以及供应商提供的任何其他信息。准备就绪后，选择订阅并启动。将立即开始您的订阅。在订阅过程中，您可以继续执行此过程中的步骤来配置实例。如果存在与信用卡详细信息有关的任何问题，会提示您更新账户详细信息。

   > **Note**
   >
   > 在使用 AMI 启动实例之前，您无需为使用产品付费。选择实例类型时，记下每种受支持实例类型的定价。还可能对产品征收其他税款。

6. 对于 Instance type（实例类型），请为您的实例选择实例类型。实例类型定义了要启动的实例的硬件配置和大小。
7. （可选）在密钥对（登录）下，为密钥对名称选择一个现有密钥对或新建一个密钥对。
8. 在网络设置下，对于防火墙（安全组），记下按照产品供应商规格创建的新安全组。安全组中的规则可能允许通过 Linux 上的 SSH（端口 22）或 Windows 上的 RDP（端口 3389）进行所有 IPv4 地址 (`0.0.0.0/0`) 访问。我们建议您调整这些规则，以仅允许特定地址或地址范围通过这些端口访问您的实例。
9. 您可以使用屏幕上的其他字段来配置实例、添加存储以及添加标签。有关可以配置的不同选项的信息，请参阅 Amazon EC2 实例配置参数参考。
10. 在 Summary（摘要）面板的 Software Image (AMI) [软件映像（AMI）] 下，检查要从中启动实例的 AMI 的详细信息。此外，还要检查您指定的其他配置的详细信息。当您准备好启动您的实例时，请选择 Launch instance（启动实例）。
11. 根据订阅的产品，实例可能需要几分钟或更多时间来启动。如果存在与信用卡详细信息有关的任何问题，会提示您更新账户详细信息。启动确认页面显示时，选择 View all instances（查看所有实例）以转到 Instances（实例）页面。

    > **Note**
    >
    > 只要实例处于 `running` 状态（即使处于空闲状态），就会收取订阅费用。如果实例停止，仍可能会收取存储费。

12. 当实例处于 `running` 状态时，可以连接到实例。为此，请在列表中选择实例，然后选择 Connect（连接）并选择一个连接选项。有关连接到实例的更多信息，请参阅连接到您的 EC2 实例。

> **Important**
>
> 仔细查看供应商的使用说明，因为您可能需要使用特定用户名来连接实例。有关访问订阅详细信息的信息，请参阅 管理您的 Amazon Web Services Marketplace 订阅。

13. 如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

使用命令行工具从 Amazon Web Services Marketplace AMI 启动实例

要使用命令行工具从 Amazon Web Services Marketplace 产品启动实例，请首先确保订阅了该产品。然后您可使用以下方法通过该产品的 AMI ID 启动一个实例：

方法	文档
Amazon CLI	使用 run-instances 命令，或参阅《Amazon Command Line Interface 用户指南》中的启动实例主题了解更多信息。
Amazon Tools for Windows PowerShell	使用 New-EC2Instance 命令，或参阅以下主题了解更多信息：使用 Windows PowerShell 启动 Amazon EC2 实例
查询 API	使用 RunInstances 请求。

连接到您的 EC2 实例

Amazon EC2 实例是 Amazon Web Services 云中的虚拟服务器。要登录实例，必须先与实例建立连接。连接实例的方式取决于实例的操作系统，以及用于连接实例的计算机上的操作系统。下表详细说明了每种连接方法的要求。

连接选项	实例操作系统	入站流量规则	IAM 权限	实例配置文件角色	实例上的软件	连接系统上的软件	密钥对
SSH 客户端	Linux	是	否	否	否	是	是
EC2 Instance Connect	Linux	支持	是	否	是[1]	否	否
PuTTY	Linux	是	否	否	否	是	是
RDP 客户端	Windows	是	否	否	否	是	是[2]
Fleet Manager	Windows	否	是	是	是[1]	否	是
会话管理器	Linux、Windows	否	是	是	是[1]	否	否
EC2 Instance Connect Endpoint	Linux、Windows	支持	是	否	否	否	否[3]

[1] 仅某些 AMI 会预装必需的软件。您可以根据需要在支持的操作系统上手动安装所需的软件。

[2] 只有在使用随机生成的本地管理员用户账户密码时，才需要此密钥对。

[3] 如果使用 SSH 连接方法，则需要密钥对。

有关详细信息，请参阅要使用的连接选项的文档。

连接选项

- 使用 SSH 客户端连接到 Linux 实例
- 使用 PuTTY 连接到 Linux 实例
- 使用 RDP 客户端连接到 Windows 实例

- 使用 Fleet Manager 连接到 Windows 实例
- 使用会话管理器进行连接
- 使用公有 IP 和 EC2 Instance Connect 进行连接
- 使用私有 IP 和 EC2 Instance Connect Endpoint 进行连接

通用连接先决条件

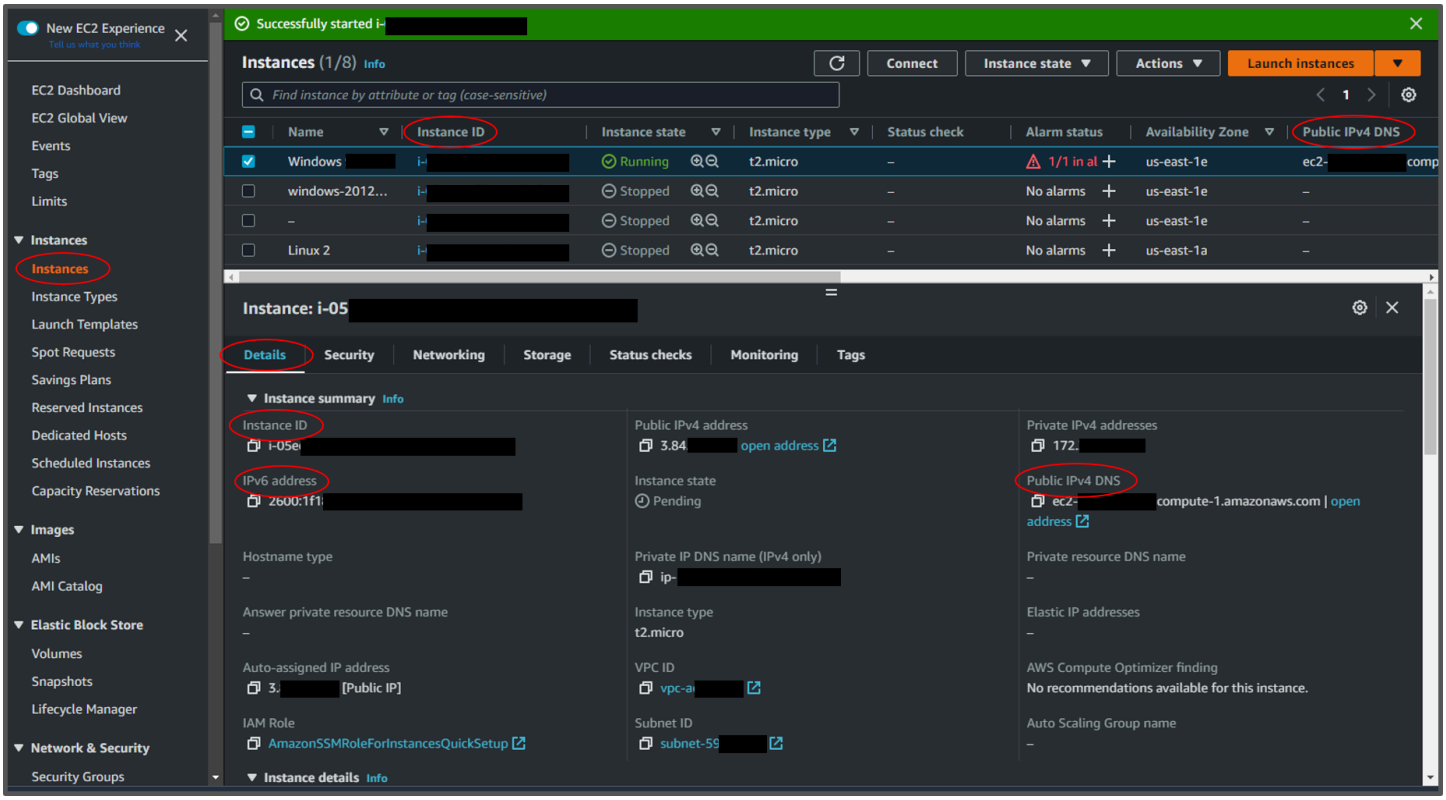
连接到实例时需要满足的通用先决条件如下。请注意，可能存在与所选连接选项相关的特定先决条件。

一般先决条

- 检查您的实例是否通过了状态检查。可能需要花几分钟时间，实例才能准备好接受连接请求。有关更多信息，请参阅 查看状态检查。
- 获取所需的实例详细信息.
- 查找私有密钥并设置权限.
- （可选）获取实例指纹.

获取所需的实例详细信息

要做好连接到实例的准备工作，请通过 Amazon EC2 控制台或使用命令行来获取以下信息。



- 获取实例的公有 DNS 名称。

  您可以从 Amazon EC2 控制台获取实例的公有 DNS。查看实例窗格的公有 IPv4 DNS 列。如果此列已隐藏，请选择屏幕右上角的设置图标（ ），然后选择公有 IPv4 DNS。您也可以在实例窗格的实例信息部分找到公有 DNS。在 Amazon EC2 控制台的实例窗格中选择实例时，有关该实例的信息将显示在页面的下半部分。在详细信息选项卡下，查找公有 IPv4 DNS。

  如果您愿意，也可以使用 describe-instances（Amazon CLI）或 Get-EC2Instance（Amazon Tools for Windows PowerShell）命令。

  如果未显示公有 IPv4 DNS，请验证实例状态是否正在运行，并确保您没有在私有子网中启动实例。如果您使用启动实例向导来启动实例，则可能已编辑网络设置下的自动分配公有 IP 字段，并且已将值更改为禁用。如果您禁用自动分配公有 IP 选项，则启动实例后不会为其分配公有 IP 地址。

- （仅适用于 IPv6）获取实例的 IPv6 地址。

  如果您已为实例分配 IPv6 地址，则可以选择使用 IPv6 地址代替公有 IPv4 地址或公有 IPv4 DNS 主机名，以连接到实例。您的本地计算机必须拥有 IPv6 地址，且必须配置为使用 IPv6。您可以从 Amazon EC2 控制台获取实例的 IPv6 地址。请查看实例窗格的 IPv6 IP 列。或者，您可以在实例信

息部分找到 IPv6 地址。在 Amazon EC2 控制台的实例窗格中选择实例时，有关该实例的信息将显示在页面的下半部分。在详细信息选项卡下，查找 IPv6 地址。

如果您愿意，也可以使用 describe-instances（Amazon CLI）或 Get-EC2Instance（Amazon Tools for Windows PowerShell）命令。有关 IPv6 的更多信息，请参阅 IPv6 地址。

- （Linux 实例）获取实例的用户名。

  您可以使用用户账户的用户名或用于启动实例的 AMI 的默认用户名连接到实例。

  - 获取用户账户的用户名。

    有关如何创建用户账户的更多信息，请参阅管理 Amazon EC2 Linux 实例上的系统用户。

  - 获取用于启动实例的 AMI 的默认用户名。
    - Amazon Linux – `ec2-user`
    - CentOS – `centos` 或 `ec2-user`
    - Debian – `admin`
    - Fedora – `fedora` 或 `ec2-user`
    - FreeBSD – `ec2-user`
    - RHEL – `ec2-user` 或 `root`
    - SUSE – `ec2-user` 或 `root`
    - Ubuntu – `ubuntu`
    - Oracle – `ec2-user`
    - Bitnami – `bitnami`
    - Rocky Linux – `rocky`
    - Other – 检查 AMI 提供程序

查找私有密钥并设置权限

必须清楚私有密钥文件的位置，才能使用 SSH 建立与 Linux 实例的初始连接，或者使用 RDP 建立与 Windows 实例的初始连接。对于 SSH 连接，必须设置文件权限，确保只有您可以读取私有密钥。

有关使用 Amazon EC2 时密钥对工作原理的信息，请参阅 Amazon EC2 密钥对和 Amazon EC2 实例。

- 查找私有密钥。

获取适用于启用实例时所指定密钥对的 .pem 文件在电脑上的全限定路径。有关更多信息，请参阅 the section called “确定启动时指定的公有密钥”。

如果无法找到私有密钥文件，则请参阅 我丢失了私有密钥。如何连接到我的实例？

（Linux 实例）如果您使用 PuTTY 连接到实例并且需要将 .pem 文件转换为 .ppk，请参阅使用 PuTTYgen 转换私有密钥。

- （Linux 实例）设置私有密钥的权限，确保只有您可以读取该密钥。
  - 从 macOS 或 Linux 进行连接

    如果您计划在 macOS 或 Linux 计算机上使用 SSH 客户端连接到 Linux 实例，请使用以下命令设置私有密钥文件的权限，以确保只有您可以读取该文件。

    ```
    chmod 400 key-pair-name.pem
    ```

    如果不设置这些权限，则无法使用此密钥对连接到实例。有关更多信息，请参阅 错误：未保护的私有密钥文件。
  - 从 Windows 进行连接

    打开“文件资源管理器”，然后右键单击 .pem 文件。依次选择属性 > 安全选项卡，然后选择高级。选择禁用继承。删除对所有用户（当前用户除外）的访问权限。

（可选）获取实例指纹

若要保护自己免遭中间人攻击，您可以通过验证显示的指纹，确认将要连接的实例的真实性。如果您从第三方提供的公有 AMI 启动实例，验证指纹将很有用。

任务概述

首先，从实例获取实例指纹。然后，当您连接到实例并收到验证指纹的提示时，请将在此过程中获取的指纹与显示的指纹进行比较。如果指纹不匹配，则表示有人可能在试图实施中间人攻击。如果二者匹配，则您可以放心地连接到您的实例。

获取实例指纹的先决条件

- 实例不能处于 pending 状态。只有在第一次引导实例完成后，才能使用指纹。
- 您必须为实例拥有者，才能获取控制台输出。

- 可以采用多种方法获取实例指纹。如果要使用 Amazon CLI，则必须将其安装在本地计算机上。有关安装 Amazon CLI 的信息，请参阅《Amazon Command Line Interface 用户指南》中的开始使用 Amazon CLI。

获取实例指纹

在步骤 1 中，您将获得控制台输出，其中包括实例指纹。在步骤 2 中，您可以在控制台输出中找到实例指纹。

1. 使用以下方法之一获取控制台输出。

   Console

   1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   2. 从左侧导航窗格中选择实例。
   3. 选择实例，然后依次选择操作、监控和故障排除、获取系统日志、

   Amazon CLI

   在本地计算机上（而不是正在连接的实例上），使用 get-console-output 命令。如果输出过大，可将输出通过管道传输到文本文件，文本形式可能更易于阅读。

   ```
   aws ec2 get-console-output \
       --instance-id i-1234567890abcdef0 \
       --query Output \
       --output text > temp.txt
   ```

   PowerShell

   在本地计算机上使用以下 Get-EC2ConsoleOutput cmdlet。

   ```
   $encodedOutput = (Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0).Output
   [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($encodedOutput
   ```

2. 在控制台输出中，找到位于 BEGIN SSH HOST KEY FINGERPRINTS 下方的实例（主机）指纹。可能有多个实例指纹。当您连接到实例时，它将仅显示其中一个指纹。

   确切的输出可能因操作系统、AMI 版本以及是否是由 Amazon 创建密钥对而异。下面是示例输出。

```
ec2:#############################################################
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
ec2: 256 SHA256:l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY no comment (ECDSA)
ec2: 256 SHA256:kpEa+rw/Uq3zxaYZN8KT501iBtJOIdHG52dFi66EEfQ no comment (ED25519)
ec2: 2048 SHA256:L8l6pepcA7iqW/jBecQjVZClUrKY+o2cHLI0iHerbVc no comment (RSA)
ec2: -----END SSH HOST KEY FINGERPRINTS-----
ec2: #############################################################
```

> **Note**
>
> 连接到实例时，您将引用此指纹。

使用 SSH 连接到 Linux 实例

有许多方法可以使用 SSH 连接到 Linux 实例。有些方法会因发起连接的本地计算机的操作系统而有所不同。其他方法基于浏览器，例如 EC2 Instance Connect 或 Amazon Systems Manager Session Manager，可以在任何计算机上使用。您可以使用 SSH 连接到 Linux 实例并运行命令，也可以使用 SSH 在本地计算机和实例之间传输文件。

在使用 SSH 连接到 Linux 实例之前，请先完成以下先决条件：

- 检查您的实例是否通过了状态检查。可能需要花几分钟时间，实例才能准备好接受连接请求。有关更多信息，请参阅 查看状态检查。
- 确保与您的实例关联的安全组允许来自您的 IP 地址的入站 SSH 流量。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。
- 获取所需的实例详细信息.
- 查找私有密钥并设置权限.
- （可选）获取实例指纹.

然后，选择以下某个选项来使用 SSH 连接到 Linux 实例。

- 使用 SSH 客户端进行连接
- 使用 PuTTY 进行连接
- 使用 SCP 传输文件

如果无法连接到实例，并且需要进行问题排查，请参阅排查 Amazon EC2 Linux 实例的连接问题。

使用 SSH 客户端连接到 Linux 实例

使用 Secure Shell（SSH）可以从本地计算机连接到 Linux 实例。有关其他选项的更多信息，请参阅连接到您的 EC2 实例。

> **Note**
>
> 如果您在尝试连接到您的实例时收到错误，请确保您的实例满足所有 SSH 连接先决条件 要求。如果它满足所有先决条件，但您仍然无法连接到 Linux 实例，请参阅排查 Amazon EC2 Linux 实例的连接问题。

内容

- SSH 连接先决条件
- 使用 SSH 客户端连接到 Linux 实例

SSH 连接先决条件

使用 SSH 连接到 Linux 实例之前，请先完成以下任务。

完成一般先决条件。

- 检查您的实例是否通过了状态检查。可能需要花几分钟时间，实例才能准备好接受连接请求。有关更多信息，请参阅 查看状态检查。
- 获取所需的实例详细信息.
- 查找私有密钥并设置权限.
- （可选）获取实例指纹.

允许来自您 IP 地址的入站 SSH 流量。

确保与您的实例关联的安全组允许来自您的 IP 地址的入站 SSH 流量。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。

在本地计算机上安装 SSH 客户端（如果需要）。

您的本地计算机可能已默认安装了 SSH 客户端。在终端窗口输入以下命令可以验证该情况。如果计算机无法识别该命令，则必须安装 SSH 客户端。

```
ssh
```

以下是适用于 Windows 的一些可能选项。如果计算机运行别的操作系统，请参阅相应操作系统的文档来了解 SSH 客户端选项。

在 Windows 系统上安装 OpenSSH

在 Windows 上安装 OpenSSH 后，即可使用 SSH 从 Windows 计算机连接到 Linux 实例。开始操作之前，务必确认已满足以下要求。

Windows 版本

计算机上的 Windows 版本必须是 Windows Server 2019 或更高版本。

若为较早版本的 Windows，请改为下载并安装 Win32-OpenSSH。

PowerShell 要求

若要在 Windows 操作系统上使用 PowerShell 安装 OpenSSH，您必须运行 PowerShell 版本 5.1 或更高版本，并且您的账户必须是内置管理员组的成员。在 PowerShell 中运行 `$PSVersionTable.PSVersion` 以查看 PowerShell 版本。

若要查看您是否是内置管理员组的成员，请运行以下 PowerShell 命令：

```
(New-Object
 Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurrent())).Is
```

如果您是内置管理员组的成员，则输出为 `True`。

若要使用 PowerShell 安装适用于 Windows 的 OpenSSH，请运行以下 PowerShell 命令。

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

下面是示例输出。

```
Path          :
Online        : True
RestartNeeded : False
```

若要使用 PowerShell 将 OpenSSH 从 Windows 卸载，请运行以下 PowerShell 命令。

```
Remove-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

下面是示例输出。

```
Path          :
Online        : True
RestartNeeded : True
```

安装适用于 Linux 的 Windows 子系统（WSL）

在 Windows 上安装 WSL 后，即可使用 Linux 命令行工具（例如 SSH 客户端）从 Windows 计算机连接到 Linux 实例。

按照在 EC2 Windows 实例上安装适用于 Linux 的 Windows 子系统中的说明进行操作。如果您按照 Microsoft 安装指南中的说明进行操作，他们会安装 Linux 的 Ubuntu 发行版。如果愿意，您可以安装不同的 Linux 发行版。

在 WSL 终端窗口中，将 .pem 文件（适用于您在启动时为实例指定的密钥对）从 Windows 复制到 WSL。记下在连接到实例时要使用的 WSL 上的 .pem 文件的完全限定路径。有关如何指定 Windows 硬盘的路径的信息，请参阅如何访问我的 C 驱动器？。

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

有关卸载 Windows Subsystem for Linux 的信息，请参阅如何卸载 WSL 发行版？

使用 SSH 客户端连接到 Linux 实例

通过以下过程使用 SSH 客户端连接到您的 Linux 实例。

使用 SSH 客户端连接到实例

1. 在计算机上打开终端窗口。
2. 使用 ssh 命令连接到实例。此时需要用到在先决条件中收集的有关实例的详细信息。例如，您需要私有密钥（.pem 文件）的位置、用户名以及公有 DNS 名称或 IPv6 地址。以下是命令示例。
   - （公有 DNS）要使用公有 DNS 名称，请输入以下命令。

     ```
     ssh -i /path/key-pair-name.pem instance-user-name@instance-public-dns-name
     ```

- （IPv6）或者，如果实例具有 IPv6 地址，请输入以下命令来使用 IPv6 地址。

```
ssh -i /path/key-pair-name.pem instance-user-name@2001:db8::1234:5678:1.2.3.4
```

以下为响应示例。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'
 can't be established.
ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
Are you sure you want to continue connecting (yes/no)?
```

3. （可选）验证安全警报中的指纹是否与该指纹相匹配。如果这些指纹不匹配，则表示有人可能在试图实施中间人攻击。如果匹配，请继续到下一步。有关更多信息，请参阅获取实例指纹。
4. 输入 **yes**。

   您会看到如下响应：

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to
 the list of known hosts.
```

使用 PuTTY 连接到 Linux 实例

适用于 Windows 的免费 SSH 客户端 PuTTY 可用于连接 Linux 实例。

如果运行的是 Windows Server 2019 或更高版本，建议使用 OpenSSH（这是一款使用 SSH 协议进行远程登录的开源连接工具）。

> **Note**
>
> 如果您在尝试连接到您的实例时收到错误，请确保您的实例满足所有 SSH 连接先决条件 要求。如果它满足所有先决条件，但您仍然无法连接到 Linux 实例，请参阅排查 Amazon EC2 Linux 实例的连接问题。

内容

- 前提条件
- （可选）使用 PuTTYgen 转换私有密钥

- 连接到您的 Linux 实例

前提条件

使用 PuTTY 连接到 Linux 实例之前，请先完成以下任务。

完成一般先决条件。

- 检查您的实例是否通过了状态检查。可能需要花几分钟时间，实例才能准备好接受连接请求。有关更多信息，请参阅 查看状态检查。
- 获取所需的实例详细信息.
- 查找私有密钥并设置权限.
- （可选）获取实例指纹.

允许来自您 IP 地址的入站 SSH 流量。

确保与您的实例关联的安全组允许来自您的 IP 地址的入站 SSH 流量。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。

在本地计算机上安装 PuTTY（如果需要）。

从 PuTTY 下载页面下载 PuTTY 并安装。如果您安装的是早期版本的 PuTTY，建议您下载最新版本。确保安装整个套件。

使用 PuTTYgen 将私有密钥转换为 PPK 格式。

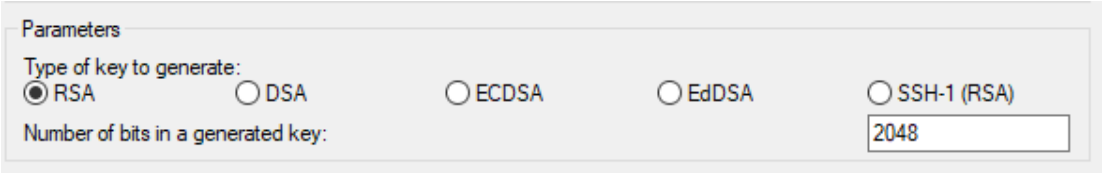
必须指定启动实例时指定的密钥对的私有密钥。如果创建了.pem 格式的私有密钥，则必须将其转换为 PPK 文件，才能与 PuTTY 一起使用。找到私有密钥（.pem 文件），然后按照使用 PuTTYgen 转换私有密钥中的步骤操作。

（可选）使用 PuTTYgen 转换私有密钥

PuTTY 自身并不支持适用于 SSH 密钥的 PEM 格式。PuTTY 提供一种名为 PuTTYgen 的工具，该工具可以将 PEM 密钥转换为 PuTTY 所需的 PPK 格式。如果使用 PEM 格式而非 PPK 格式创建了密钥，则必须将私有密钥（.pem 文件）转换为此格式（.ppk 文件），才能与 PuTTY 一起使用。

将私有密钥 从 PEM 格式转换为 PPK 格式

1. 从 Start (开始) 菜单中，依次选择 All Programs (所有程序)、PuTTY、PuTTYgen。
2. 在 Type of key to generate 下，选择 RSA。如果您的 PuTTYgen 版本不包含此选项，请选择 SSH-2 RSA。



3. 选择 Load。默认情况下，PuTTYgen 仅显示扩展名为 .ppk 的文件。要找到您的 .pem 文件，请选择显示所有类型的文件的选项。

4. 选择在启动实例时指定的密钥对的 .pem 文件，然后选择打开。PuTTYgen 显示一个通知，即 .pem 文件已导入成功。选择 OK (确定)。
5. 要以 PuTTY 可以使用的格式保存密钥，请选择 Save private key (保存私有密钥)。PuTTYgen 显示一条关于在没有密码的情况下保存密钥的警告。选择是。

> **Note**
>
> 私有密钥上的密码提供额外一层保护。即使发现了您的私有密钥，也不能在没有密码的情况下使用该密钥。使用密码的缺点是自动化更难实现，因为需要人工干预以登录到实例或将文件复制到实例中。

6. 为密钥指定您用于密钥对的相同名称（例如 key-pair-name）并选择 Save (保存)。PuTTY 会自动添加 .ppk 文件扩展名。

您的私有密钥格式现在是正确的 PuTTY 使用格式了。您现在可以使用 PuTTY 的 SSH 客户端连接到实例。

连接到您的 Linux 实例

通过以下过程使用 PuTTY 连接到您的 Linux 实例。您需要使用为私有密钥创建的 .ppk 文件。有关更多信息，请参阅上一个部分中的（可选）使用 PuTTYgen 转换私有密钥。如果您在尝试连接到实例时收到错误，请参阅 排查 Amazon EC2 Linux 实例的连接问题。

最后测试版本 – PuTTY .78

使用 PuTTY 连接到您的实例

1. 启动 PuTTY（从开始菜单中，搜索 PuTTY，然后选择打开）。
2. 在 Category 窗格中，选择 Session 并填写以下字段：

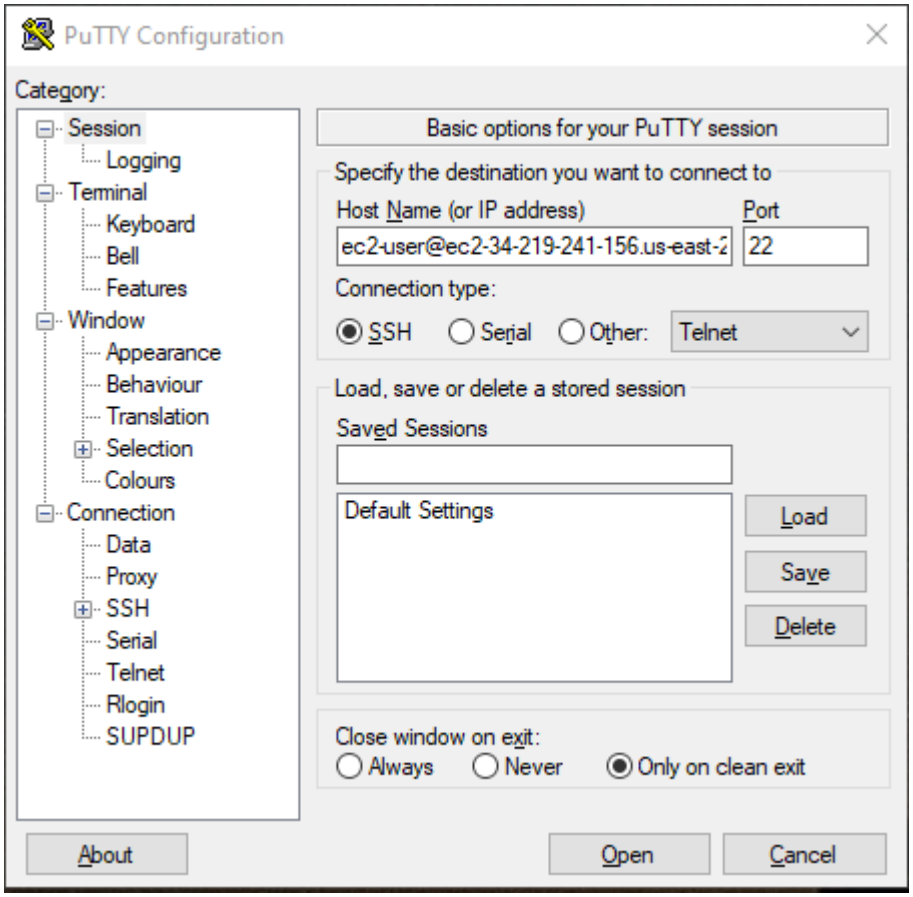
a. 在主机名框中，执行以下操作之一：

- （公有 DNS）要使用实例的公有 DNS 名称进行连接，请输入 *instance-user-name*@*instance-public-dns-name*。
- （IPv6）或者，如果实例具有 IPv6 地址，要使用实例的 IPv6 地址进行连接，请输入 *instance-user-name*@*2001:db8::1234:5678:1.2.3.4*。

有关如何获取实例的用户名以及实例的公有 DNS 名称或 IPv6 地址的信息，请参阅获取所需的实例详细信息。

b. 确保端口值为 22。

c. 在连接类型下，选择 SSH。



3. (可选) 您可以配置 PuTTY 以定期自动发送“保持连接”数据以将会话保持活动状态。要避免由于会话处于不活动状态而与实例断开连接，这是非常有用的。在类别 窗格中，选择连接，然后在 keepalives 之间的秒数中输入所需的间隔。例如，如果您的会话在处于不活动状态 10 分钟后断开连接，请输入 180 以将 PuTTY 配置为每隔 3 分钟发送一次保持活动数据。
4. 在类别窗格中，依次展开连接、SSH 和 Auth。选择凭证。

5. 在用于身份验证的私有密钥文件旁，选择浏览。在选择私有密钥文件对话框中，选择您为密钥对生成的 .ppk 文件。您可以双击该文件，也可以在选择私有密钥文件对话框中选择打开。
6. （可选）如果打算在此会话后再次连接到此实例，则可以保存此会话信息以便日后使用。在类别窗格中，选择会话。在已保存的会话中输入会话名称，然后选择保存。
7. 要连接到实例，请选择打开。
8. 如果这是第一次连接到该实例，PuTTY 将显示安全警报对话框，以询问您是否信任要连接到的主机。

   a. (可选) 验证安全警报对话框中的指纹是否与您之前在 （可选）获取实例指纹 中获得的指纹相匹配。如果这些指纹不匹配，则表示有人可能在试图实施“中间人”攻击。如果匹配，请继续到下一步。

   b. 选择 Accept (接受)。将打开一个窗口，并且您连接到实例。

   > **Note**
   >
   > 如果您在将私有密钥转换成 PuTTY 格式时指定了密语，当您登录到实例时，您必须提供该密语。

如果您在尝试连接到实例时收到错误，请参阅 排查 Amazon EC2 Linux 实例的连接问题。

使用 SCP 将文件传输到 Linux 实例

在您的本地计算机与 Linux 实例之间传输文件的一种方法是使用安全复制协议 (SCP)。SCP 非常适合简单的操作，例如一次性文件副本。SCP 用来保护文件传输的 .pem 文件与您用来通过 SSH 连接到实例的文件相同。如果需要保持文件同步，或者文件很大，则 rsync 比 SCP 更快、更高效。为安全起见，请通过 SSH 使用 rsync，因为 rsync 默认情况下会使用明文传输数据。

在使用 SCP 连接到 Linux 实例之前，请先完成以下任务：

- 完成一般先决条件。
  - 检查您的实例是否通过了状态检查。可能需要花几分钟时间，实例才能准备好接受连接请求。有关更多信息，请参阅 查看状态检查。
  - 获取所需的实例详细信息.
  - 查找私有密钥并设置权限.
  - （可选）获取实例指纹.
- 允许来自您 IP 地址的入站 SSH 流量。

确保与您的实例关联的安全组允许来自您的 IP 地址的入站 SSH 流量。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。

- 安装 SCP 客户端。

  默认情况下，大多数 Linux、Unix 和 Apple 计算机都包含 SCP 客户端。如果您的计算机不含 SSH 客户端，OpenSSH 项目提供了整套 SSH 工具免费使用的功能，包括 SCP 客户端。有关更多信息，请参阅 https://www.openssh.com。

以下过程将指导您使用实例的公有 DNS 名称或 IPv6 地址（如果实例具有该地址），通过 SCP 传输文件。

使用 SCP 在您的计算机和实例之间传输文件

1. 确定源文件在计算机上的位置以及在实例上的目标路径。在以下示例中，私有密钥文件的名称是 `key-pair-name.pem`，要传输的文件是 `my-file.txt`，实例的用户名是 ec2-user，实例的公有 DNS 名称是 `instance-public-dns-name`，实例的 IPv6 地址是 `2001:db8::1234:5678:1.2.3.4`。
   - （公有 DNS）要将文件传输到实例上的目标位置，请在计算机中输入以下命令。

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@instance-public-dns-name:path/
     ```

   - (IPv6) 如果实例具有 IPv6 地址，要将文件传输到实例上的目标位置，请在计算机中输入以下命令。IPv6 地址必须用方括号 ([ ]) 括起来，并且必须对方括号进行转义 (\)。

     ```
     scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@
     \[2001:db8::1234:5678:1.2.3.4\]:path/
     ```

2. 如果您尚未使用 SSH 连接到实例，则会看到如下响应：

   ```
   The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'
   can't be established.
   RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
   Are you sure you want to continue connecting (yes/no)?
   ```

   （可选）您可以选择验证安全警报中的指纹是否与实例指纹匹配。有关更多信息，请参阅（可选）获取实例指纹。

输入 **yes**。

3. 如果传输成功，则响应的形式与下方类似：

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
my-file.txt                                   100%   480     24.4KB/s   00:00
```

4. 要在另一个方向上传输文件（从 Amazon EC2 实例中传输到计算机），请颠倒主机参数的顺序。例如，您可以将 `my-file.txt` 从 EC2 实例传输到本地计算机上的目标位置，保存为 `my-file2.txt`，如以下示例所示。

   - （公有 DNS）要将文件传输到计算机中的目标位置，请在计算机中输入以下命令。

     ```
     scp -i /path/key-pair-name.pem ec2-user@instance-public-dns-name:path/my-file.txt path/my-file2.txt
     ```

   - (IPv6) 如果实例具有 IPv6 地址，要将文件传输到计算机中的目标位置，请在计算机中输入以下命令。IPv6 地址必须用方括号 ([ ]) 括起来，并且必须对方括号进行转义 (\)。

     ```
     scp -i /path/key-pair-name.pem ec2-user@\[2001:db8::1234:5678:1.2.3.4\]:path/my-file.txt path/my-file2.txt
     ```

管理 Amazon EC2 Linux 实例上的系统用户

每个 Linux 实例均以默认的 Linux 系统用户启动。您可以向实例添加用户，也可以删除用户。

对于默认用户，默认用户名由您在启动实例时指定的 AMI 决定。

> **Note**
>
> 默认情况下，密码身份验证和根用户登录将被禁用，而 sudo 将会启用。要登录您的实例，必须使用密钥对。有关日志记录的更多信息，请参阅 使用 SSH 连接到 Linux 实例。
> 您可以允许对您的实例进行密码身份验证和根用户登录。有关更多信息，请参阅您的操作系统文档。

> **Note**
>
> Linux 系统用户不应与 IAM 用户混淆。有关更多信息，请参阅 IAM 用户指南中的 IAM 用户。

内容

- 默认用户名
- 注意事项
- 创建用户
- 删除用户

默认用户名

EC2 实例的默认用户名由您在启动实例时指定的 AMI 决定。

默认用户名为：

- 对于 Amazon Linux AMI，用户名是 `ec2-user`。
- 对于 CentOS AMI，用户名是 `centos` 或 `ec2-user`。
- 对于 Debian AMI，用户名是 `admin`。
- 对于 Fedora AMI，用户名是 `fedora` 或 `ec2-user`。
- 对于 FreeBSD AMI，用户名是 `ec2-user`。
- 对于 RHEL AMI，用户名是 `ec2-user` 或 `root`。
- 对于 SUSE AMI，用户名是 `ec2-user` 或 `root`。
- 对于 Ubuntu AMI，用户名为 `ubuntu`。
- 对于 Oracle AMI，用户名是 `ec2-user`。
- 对于 Bitnami AMI，用户名是 `bitnami`。

> **Note**
>
> 要查找其他 Linux 发行版的默认用户名，请咨询 AMI 提供商。

注意事项

对于许多应用程序来说，使用默认用户是适当的。但是，您可以选择添加用户，以便个人能够拥有自己的文件和工作区。此外，为新用户创建用户比向多个（可能缺乏经验的）用户授予对默认用户的访问权限更安全，因为如果使用不当，该默认用户可能对系统造成严重破坏。

要使用 Linux 系统用户启用对 EC2 实例的用户 SSH 访问，您必须与用户分享 SSH 密钥。此外，您可以使用 EC2 Instance Connect 来向用户提供访问权限而无需共享和管理 SSH 密钥。有关更多信息，请参阅 使用公有 IP 地址和 EC2 Instance Connect 连接到 Linux 实例。

创建用户

先创建用户，然后添加允许用户连接并登录实例的 SSH 公有密钥。

> **Important**
>
> 在此过程的步骤 1 中，创建新的密钥对。由于密钥对的功能类似于密码，因此安全处理至关重要。如果您为用户创建密钥对，则必须确保将私有密钥安全地发送给这些用户。或者，用户可以通过创建自己的密钥对来完成步骤 1 和步骤 2，确保其计算机上的私有密钥安全，然后将公有密钥发送给您以完成步骤 3 中的过程。

创建用户

1. 创建新的密钥对。您必须将 .pem 文件提供给要为其创建用户的用户。他们必须使用此文件来连接到实例。
2. 从您在上一步中创建的密钥对中检索公有密钥。

   ```
   $ ssh-keygen -y -f /path_to_key_pair/key-pair-name.pem
   ```

   该命令返回公有密钥，如以下示例所示。

   ```
   ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6Vhz2ItxCih
   +PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
   d6RJhJOI0iBXrlsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/
   i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco
   +CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
   +z7wB3RbBQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
   ```

3. 连接到实例。

4. 使用 adduser 命令创建用户并将其添加到系统（/etc/passwd 文件中会有一个条目）。该命令还可以为用户创建一个组和一个主目录。在此示例中，用户名为 *newuser*。

   - AL2023 和 Amazon Linux 2

     对于 AL2023 和 Amazon Linux 2，创建用户时将默认禁用密码身份验证。

     ```
     [ec2-user ~]$ sudo adduser newuser
     ```

   - Ubuntu

     添加 --disabled-password 参数以创建用户并禁用密码身份验证。

     ```
     [ubuntu ~]$ sudo adduser newuser --disabled-password
     ```

5. 切换到新用户，以便所创建的目录和文件具有适当的所有权。

   ```
   [ec2-user ~]$ sudo su - newuser
   ```

   如果提示从 ec2-user 变为 *newuser*，则表示您已将 Shell 会话切换到新用户。

6. 将 SSH 公有密钥添加到用户。首先在 SSH 密钥文件对应的用户主目录中创建一个目录，然后创建密钥文件，最后将公有密钥粘贴到该密钥文件中，如以下分步说明中所述。

   a. 在 *newuser* 主目录中创建一个 .ssh 目录，并将其权限更改为 700 (只有文件所有者能够读取、写入或打开该目录。)

      ```
      [newuser ~]$ mkdir .ssh
      ```

      ```
      [newuser ~]$ chmod 700 .ssh
      ```

      > **Important**
      >
      > 如果没有这些确切的文件权限，用户将无法登录。

   b. 在 .ssh 目录中，创建一个名为 authorized_keys 的文件，并将该文件权限更改为 600（只有文件所有者才能够读取或写入此文件)。

      ```
      [newuser ~]$ touch .ssh/authorized_keys
      ```

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

> **Important**
>
> 如果没有这些确切的文件权限，用户将无法登录。

c. 使用您常用的文本编辑器 (如 vim 或 nano) 打开 `authorized_keys` 文件。

```
[newuser ~]$ nano .ssh/authorized_keys
```

将您在步骤 2 中检索到的公有密钥粘贴到文件中并保存更改。

> **Important**
>
> 确保将公有密钥粘贴到一个连续行中。不得将公有密钥拆分为多行。

用户现在能够使用添加到 *newuser* 文件的公有密钥所对应的私有密钥登录实例上的 `authorized_keys` 用户。有关连接到 Linux 实例的不同方法的更多信息，请参阅使用 SSH 连接到 Linux 实例。

删除用户

如果不再需要某个用户，可以将其删除，使它不再可用。

使用 userdel 命令从系统中删除用户。当您指定 `-r` 参数时，用户的主目录和邮件后台打印将被删除。要保留用户的主目录和邮件后台打印，请省略 `-r` 参数。

```
[ec2-user ~]$ sudo userdel -r olduser
```

使用 RDP 连接到 Windows 实例

您可以通过远程桌面，连接到从大多数 Windows 亚马逊机器映像（AMI）创建的 Amazon EC2 实例。远程桌面使用远程桌面协议 (RDP)，并且可让您像使用您面前的电脑一样（本地电脑），来连接并使用您的实例。它在大多数的 Windows 版本上可用，并且也适用于 Mac OS。

借助适用于 Windows Server 操作系统的许可证，可以同时进行两个远程连接以进行管理。适用于 Windows Server 的许可证包含在您的 Windows 实例的价格中。如果您需要同时进行两个以上的远程连接，则必须购买远程桌面服务 (RDS) 许可证。如果尝试第三个连接，将产生错误。

> **Tip**
>
> 如果需要连接到您的实例，以排查基于 Amazon Nitro System 构建的实例的启动、网络配置和其他问题，您可以使用 适用于实例的 EC2 Serial Console。

内容

- 使用 RDP 客户端连接到 Windows 实例
- 使用 Fleet Manager 连接到 Windows 实例
- 使用 RDP 将文件传输到 Windows 实例

使用 RDP 客户端连接到 Windows 实例

您可以使用 RDP 客户端连接到 Windows 实例，如下所示。

> **Tip**
>
> 或者，您可以使用 Systems Manager Fleet Manager 或 EC2 Instance Connect Endpoint 连接到 Windows 实例。

先决条件

必须满足以下先决条件，才能使用 RDP 客户端连接到 Windows 实例。

- 完成一般先决条件。
  - 检查您的实例是否通过了状态检查。可能需要花几分钟时间，实例才能准备好接受连接请求。有关更多信息，请参阅 查看状态检查。
  - 获取所需的实例详细信息.
  - 查找私有密钥并设置权限.
  - （可选）获取实例指纹.
- 安装 RDP 客户端。

  - （Windows）默认情况下，Windows 包含一个 RDP 客户端。要进行验证，请在命令提示符窗口键入 mstsc。如果您的计算机不能识别此命令，请从 Microsoft Store 下载 Microsoft Remote Desktop 应用程序。
  - （macOS X）从 Mac App Store 下载 Windows App for Mac（此前名为 Microsoft Remote Desktop）。
  - （Linux）使用 Remmina。
- 允许来自您 IP 地址的入站 RDP 流量。

  确保与实例关联的安全组允许来自您 IP 地址的传入 RDP 流量。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。

检索管理员密码

如果您已将实例加入域，则可以使用 Amazon Directory Service 中的域凭证来连接自己的实例。在远程桌面登录屏幕上，将完全限定用户名用于管理员（例如 **`corp.example.com\Admin`**），将密码用于此账户，而不是使用本地计算机名称和生成的密码。

要使用 RDP 连接到 Windows 实例，您必须检索初始管理员密码，然后在连接到实例时输入此密码。此密码在实例启动几分钟之后才可用。您的账户必须拥有调用 GetPasswordData 操作的权限。有关更多信息，请参阅 用于控制访问 Amazon EC2 API 的示例策略。

管理员账户的默认用户名取决于 AMI 中包含的操作系统（OS）的语言。要确定正确的用户名，请识别操作系统的语言，然后选择相应的用户名。例如，对于英语操作系统，用户名是 `Administrator`；对于法语操作系统，用户名是 `Administrateur`；对于葡萄牙语操作系统，则是 `Administrador`。如果某个语言版本的操作系统没有相同语言的用户名，请选择用户名 `Administrator (Other)`。有关更多信息，请参阅 Microsoft 网站中的 Localized Names for Administrator Account in Windows。

检索初始管理员密码

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择该实例，然后选择 Connect (连接)。
4. 在连接到实例页面上，选择 RDP 客户端选项卡。
5. 在用户名中，选择管理员账户的默认用户名。您选择的用户名必须与用于启动实例的 AMI 中包含的操作系统（OS）语言相匹配。如果没有与操作系统相同语言的用户名，请选择管理员（其他）。
6. 选择获取密码。

7. 在获取 Windows 密码页面上，执行以下操作：

   a. 选择上传私有密钥文件，然后导航至您启动实例时指定的私有密钥（`.pem`）文件。选择文件并选择 Open（打开），以便将文件的全部内容复制到此窗口。

   b. 选择解密密码。获取 Windows 密码页面将关闭，实例的默认管理员密码将显示在密码下，替换先前显示的获取密码链接。

   c. 复制密码并将其保存在安全的位置。需要使用此密码连接实例。

连接到 Windows 实例

以下过程使用适用于 Windows 的远程桌面连接客户端（MSTSC）。如果使用的是其他 RDP 客户端，请下载 RDP 文件，然后查看 RDP 客户端的文档，了解建立 RDP 连接的步骤。

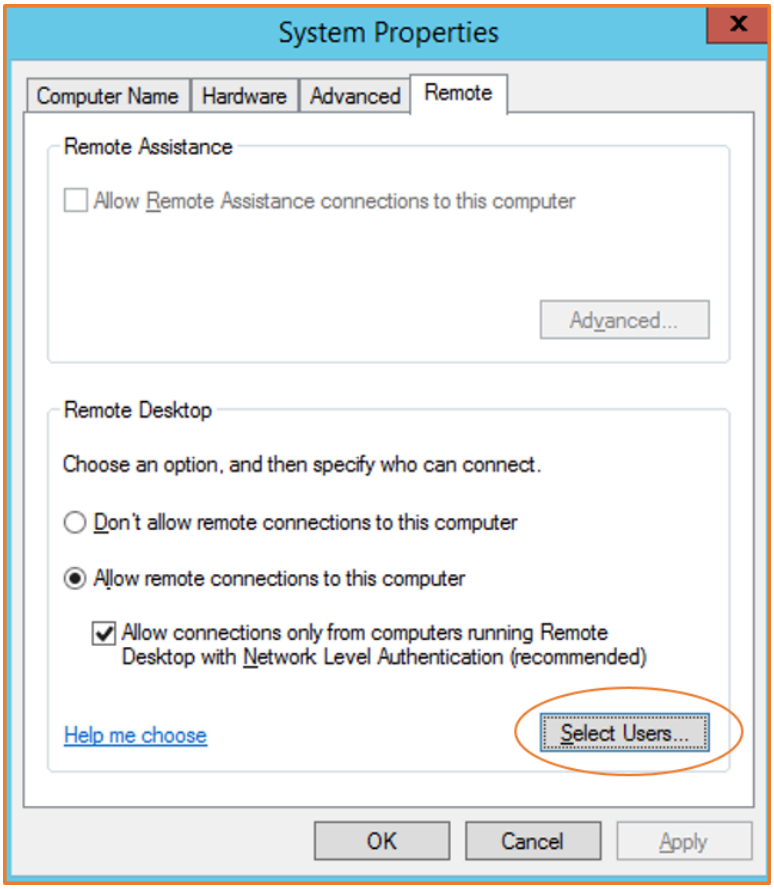
使用 RDP 客户端连接到 Windows 实例

1. 在连接到实例页面上，选择下载远程桌面文件。完成文件下载后，选择取消，然后返回实例页面。RDP 文件已下载到 `Downloads` 文件夹。
2. 运行 `mstsc.exe` 来打开 RDP 客户端。
3. 展开显示选项，选择打开，然后从 `Downloads` 文件夹中选择.rdp 文件。
4. 默认情况下，计算机是实例的公有 IPv4 DNS 名称，用户名是管理员账户。要改用 IPv6 连接到实例，请将实例的公有 IPv4 DNS 名称替换为其 IPv6 地址。查看默认设置并根据需要进行更改。
5. 选择连接。如果收到一条警告，指出远程连接发布者未知，请选择连接继续进行操作。
6. 输入之前保存的密码，然后选择确定。
7. 由于自签名证书的固有特性，您可能会看到一条警告，指出无法验证该安全证书。请执行以下操作之一：
   - 如果您信任该证书，则请选择是连接到您的实例。
   - [Windows] 在继续操作之前，请将证书的指纹与系统日志中的值进行比较，以确认远程计算机的身份。选择查看证书，然后从详细信息选项卡选择指纹。将此值与操作、监控和故障排除、获取系统日志中的 `RDPCERTIFICATE-THUMBPRINT` 值进行比较。
   - [Mac OS X] 在继续操作之前，请将证书的指纹与系统日志中的值进行比较，以确认远程计算机的身份。选择显示证书，展开详细信息，然后选择 SHA1 指纹。将此值与操作、监控和故障排除、获取系统日志中的 `RDPCERTIFICATE-THUMBPRINT` 值进行比较。
8. 如果 RDP 连接成功，RDP 客户端会先显示 Windows 登录屏幕，再显示 Windows 桌面。如果收到错误消息，请参阅the section called “远程桌面无法连接到远程计算机”。完成 RDP 连接后，可以关闭 RDP 客户端。

配置用户账户

通过 RDP 连接到实例之后，建议执行以下任务：

- 更改默认的管理员密码。您可以在登录实例本身后更改密码，其做法与任何电脑运行的 Windows Server 上的做法类似。
- 在实例上创建另一个具有管理员权限的用户。这是一种保护措施，以防止您万一忘记管理员密码或管理员账户出现问题。新用户必须拥有远程访问实例的权限。通过右键单击 Windows 桌面上或文件资源管理器中的此电脑图标并选择属性，打开系统属性。选择远程设置，然后选择选择用户以将用户添加到远程桌面用户组。



使用 Fleet Manager 连接到 Windows 实例

您可以使用 Fleet Manager（Amazon Systems Manager 的一项功能）通过远程桌面协议（RDP）连接到 Windows 实例，并在 Amazon Web Services 管理控制台 中的同一页面上显示最多四个 Windows 实例。您可以直接从 Amazon EC2 控制台中的实例页面连接到 Fleet Manager 远程桌面中的第一个实例。有关 Fleet Manager 的更多信息，请参阅《Amazon Systems Manager 用户指南》中的使用 Remote Desktop 连接到托管式实例。

如果您使用 Fleet Manager 进行连接，则无需特别允许来自您 IP 地址的传入 RDP 流量。Fleet Manager 会进行处理。

先决条件

在尝试使用 Fleet Manager 连接到实例之前，必须先设置环境。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的设置环境。

使用 Fleet Manager 连接到 Windows 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航窗格中选择 Instances（实例）。
3. 选择该实例，然后选择 Connect (连接)。
4. 在 RDP 客户端选项卡的连接类型中，选择使用 Fleet Manager 进行连接。
5. 选择 Fleet Manager 远程桌面。此操作将打开 Amazon Systems Manager 控制台中的 Fleet Manager Remote Desktop（Fleet Manager 远程桌面）页面。
6. 输入凭证，然后选择连接。
7. 如果 RDP 连接成功，Fleet Manager 会显示 Windows 桌面。完成会话后，依次选择操作、结束会话。

有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的使用 Remote Desktop 连接到 Windows Server 托管式实例。

使用 RDP 将文件传输到 Windows 实例

您可以像操作任何 Windows 服务器那样操作您的 Windows 实例。例如，您可以使用 Microsoft 远程桌面连接（RDP）软件的本地文件共享功能，在 Windows 实例与您的本地计算机之间传输文件。您可以访问硬盘驱动器、DVD 驱动器、便携式媒体驱动器和映射网络驱动器上的本地文件。

要从 Windows 实例访问本地文件，必须通过将远程会话驱动器映射到本地驱动器来启用本地文件共享功能。这些步骤略有不同，具体取决于您的本地电脑操作系统是 Windows 还是 macOS X。

要详细了解使用 RDP 进行连接的先决条件，请参阅先决条件。

Windows

在本地 Windows 电脑上将远程会话驱动器映射到本地驱动器

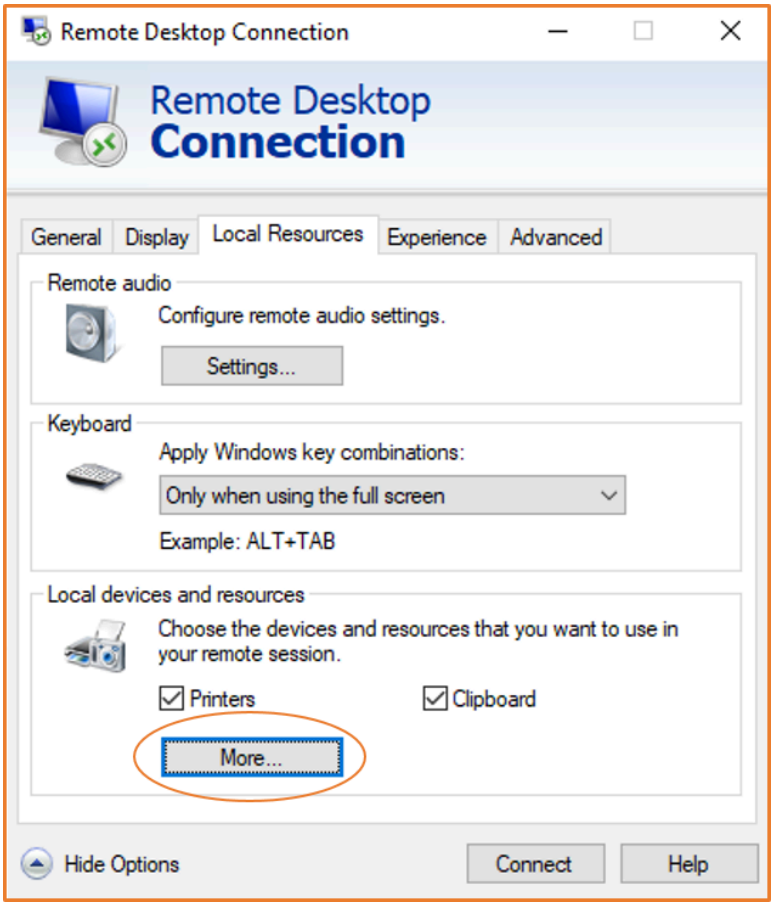
1. 打开远程桌面连接客户端。

2. 选择 Show Options。
3. 将实例主机名添加到计算机字段，将用户名添加到用户名字段，如下所示：

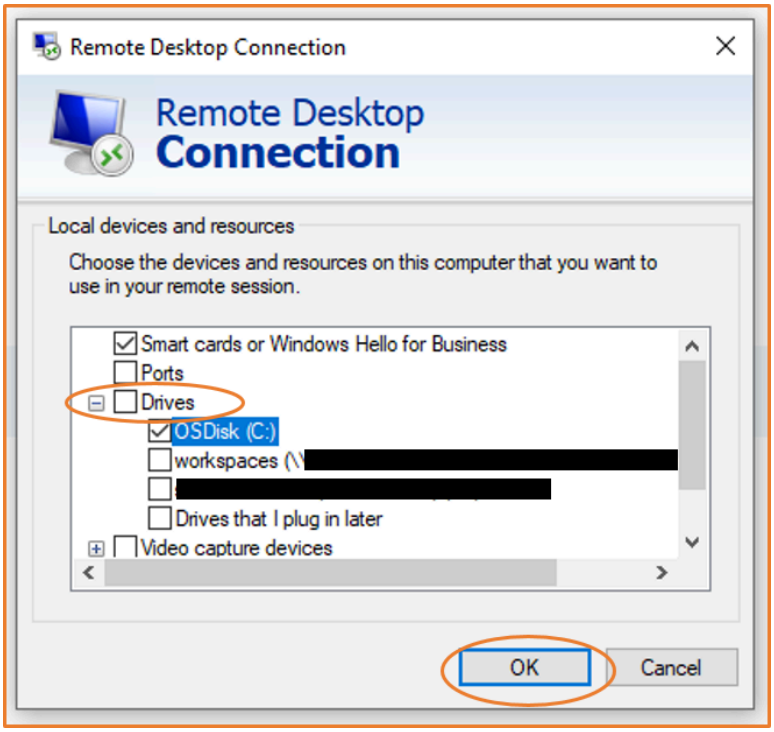
   a. 在 Connection settings（连接设置）项下，选择 Open...（打开…），然后浏览到您从 Amazon EC2 控制台下载的 RDP 快捷方式文件。该文件包含公有 IPv4 DNS 主机名（用于标识实例）和管理员用户名。

   b. 选择文件，然后选择 Open（打开）。使用 RDP 快捷方式文件中的值填充 Computer（电脑）和 User name（用户名）字段。

   c. 选择保存。
4. 选择 Local Resources (本地资源) 选项卡。
5. 在 Local Devices and resources (本地设备和资源) 项下，选择 More...（更多…）。



6. 打开 Drives (驱动器) 并选择要映射到 Windows 实例的本地驱动器。
7. 选择 OK (确定)。

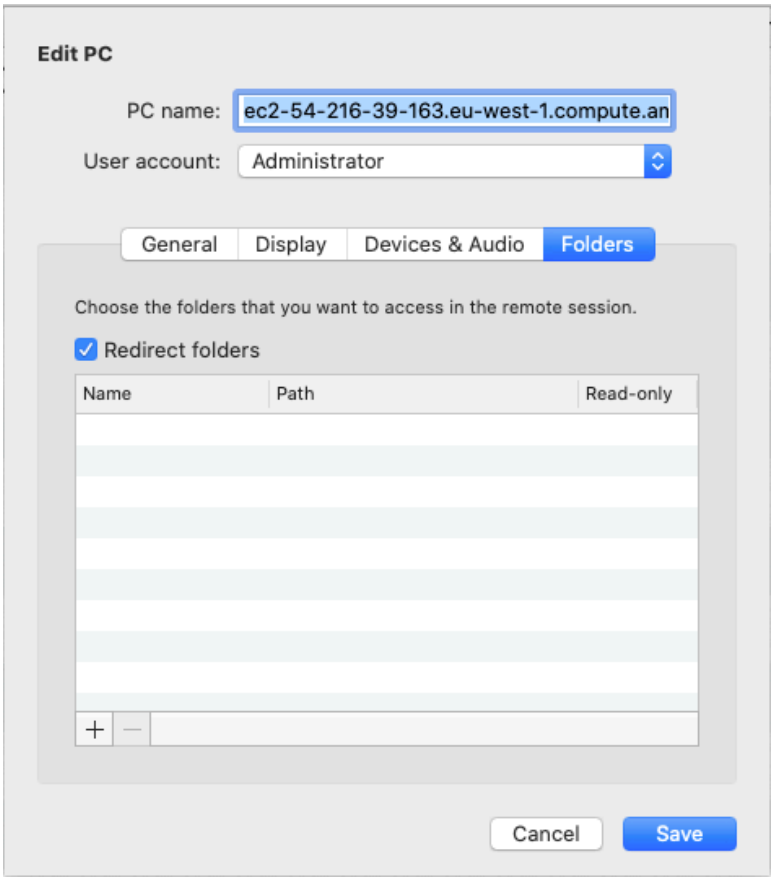


8. 选择 Connect (连接) 以连接到您的 Windows 实例。

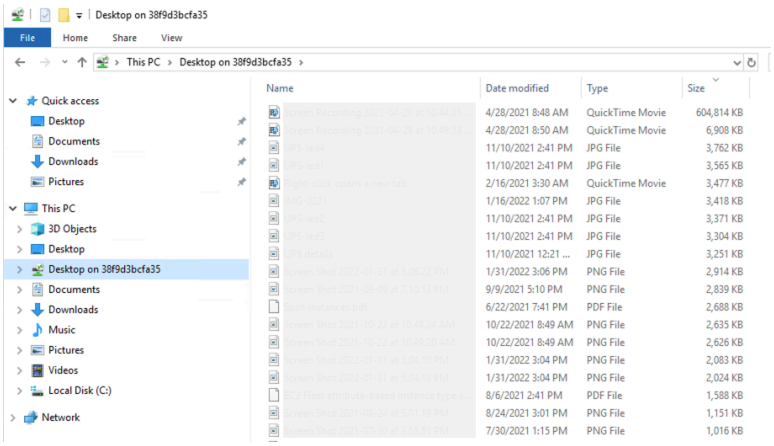
macOS X

在本地 macOS X 电脑上将远程会话驱动器映射到本地文件夹

1. 打开远程桌面连接客户端。
2. 浏览到从 Amazon EC2 控制台下载的 RDP 文件（初次连接到实例时），然后将其拖到远程桌面连接客户端上。
3. 右键单击该 RDP 文件并选择 Edit（编辑）。
4. 选择文件夹选项卡，然后选择重定向文件夹复选框。



5. 选择左下角的 + 图标，浏览到要映射的文件夹，然后选择 Open（打开）。对要映射的每个文件夹重复此步骤。
6. 选择保存。
7. 选择 Connect (连接) 以连接到您的 Windows 实例。系统将提示您输入密码。
8. 在实例的文件资源管理器中，展开 This PC（此电脑），然后找到您可以从中访问本地文件的共享文件夹。在以下屏幕截图中，本地电脑上的 Desktop（桌面）文件夹已映射到实例上的远程会话驱动器。



有关使本地设备可用于 Mac 电脑上远程会话的更多信息，请参阅开始使用 macOS 客户端。

使用 Session Manager 连接到 Amazon EC2 实例

会话管理器是一项完全托管的 Amazon Systems Manager 功能，可让您通过基于浏览器的交互一键式 Shell 或通过 Amazon CLI 管理 Amazon EC2 实例。您可以使用会话管理器通过您账户中的实例来开启会话。开启会话后，您可以像对任何其他连接类型一样在实例上运行交互式命令。有关会话管理器的更多信息，请参阅 Amazon Systems Manager 用户指南 中的 Amazon Systems Manager 会话管理器。

先决条件

在尝试使用 Session Manager 连接到实例之前，必须先完成必要的设置步骤。例如，实例必须由 SSM 托管且具有附加了 AmazonSSMManagedInstanceCore 策略的 IAM 角色。有关更多信息，请参阅设置 Session Manager。

在 Amazon EC2 控制台上通过会话管理器连接到 Amazon EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例，然后选择连接。
4. 对于连接方法，请选择 Session Manager。
5. 选择连接来启动会话。

故障排除

如果您收到一个错误，该错误指示您无权执行一个或多个 Systems Manager 操作 (`ssm:command-name`)，则必须更新策略，以便允许您从 Amazon EC2 控制台开始会话。有关更多信息和说明，请参阅《Amazon Systems Manager 用户指南》中的 Session Manager 快速入门默认 IAM policy。

使用公有 IP 地址和 EC2 Instance Connect 连接到 Linux 实例

Amazon EC2 Instance Connect 提供了一种安全的方法，可使用 Secure Shell（SSH）连接到 Linux 实例。借助 EC2 Instance Connect，您可以使用 Amazon Identity and Access Management (IAM) 策略和主体来控制对实例的 SSH 访问，无需共享和管理 SSH 密钥。使用 EC2 Instance Connect 的所有连接请求将记录到 Amazon CloudTrail 中，以便您可以审核连接请求。

您可以通过 Amazon EC2 控制台或所选的 SSH 客户端使用 EC2 Instance Connect 连接到实例。

在使用 EC2 Instance Connect 连接到实例时，EC2 Instance Connect API 会将一个 SSH 公有密钥推送到实例元数据并在其中保留 60 秒。附加到用户的 IAM policy 会授权用户将公有密钥推送到实例元数据。SSH 进程守护程序使用在安装 EC2 Instance Connect 时配置的 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser`，从实例元数据中查找公有密钥以进行身份验证，并将您连接到实例。

> **Tip**
>
> EC2 Instance Connect 是连接到 Linux 实例的选项之一。有关其他选项，请参阅 使用 SSH 连接到 Linux 实例。要连接到 Windows 实例，请参阅 使用 RDP 连接到 Windows 实例。

定价

EC2 Instance Connect 无需额外付费。

区域可用性

EC2 Instance Connect 适用于所有 Amazon Web Services 区域，但在本地区中不受支持。

内容

- 教程：完成使用 EC2 Instance Connect 连接到实例所需的配置
- EC2 Instance Connect 的先决条件
- 为 EC2 Instance Connect 授予 IAM 权限
- 在您的 EC2 实例上安装 EC2 Instance Connect
- 使用 EC2 Instance Connect 连接到 Linux 实例
- 卸载 EC2 Instance Connect

有关讨论如何使用 EC2 Instance Connect 提高堡垒主机安全性的博客文章，请参阅使用 Amazon EC2 Instance Connect 保护您的堡垒主机。

教程：完成使用 EC2 Instance Connect 连接到实例所需的配置

要在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到实例，首先需要完成让您能够成功连接到实例的先决条件配置。本教程旨在引导您完成各项任务以完成先决条件配置。

教程概述

在本教程中，您将完成以下四项任务：

- 任务 1：授予使用 EC2 Instance Connect 所需的权限

  首先，您将创建一个 IAM 策略，其中包含允许您将公有密钥推送到实例元数据的 IAM 权限。您将此策略附加到您的 IAM 身份（用户、用户组或角色），以便您的 IAM 身份能够获得这些权限。

- 任务 2：允许从 EC2 Instance Connect 服务到实例的入站流量

  然后，您将创建一个安全组，以允许从 EC2 Instance Connect 服务到实例的流量。当您在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到实例时，将需要此安全组。

- 任务 3：启动实例

  然后，您将使用预装了 EC2 Instance Connect 的 AMI 启动一个 EC2 实例，然后添加上一步中创建的安全组。

- 任务 4：连接到实例

  最后，您将在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到您的实例。如果可以连接，则可以确定在任务 1、2 和 3 中完成的先决条件配置已经成功。

任务 1：授予使用 EC2 Instance Connect 所需的权限

在使用 EC2 Instance Connect 连接到实例时，EC2 Instance Connect API 会将一个 SSH 公有密钥推送到实例元数据并在其中保留 60 秒。您需要将一个 IAM 策略附加到您的 IAM 身份（用户、用户组或角色），以授予您将公有密钥推送到实例元数据的必要权限。

任务目标

您要创建 IAM 策略来授予将公有密钥推送到实例的权限。要允许的具体操作是 `ec2-instance-connect:SendSSHPublicKey`。您还必须允许 `ec2:DescribeInstances` 操作，以确保您能够在 Amazon EC2 控制台中查看和选择您的实例。

创建好策略后，将此策略附加到 IAM 身份（用户、用户组或角色），以便 IAM 身份能够获得相关权限。

您将创建一个配置如下的策略：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
```

```
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

> **Important**
>
> 本教程中创建的 IAM 策略是一个权限十分宽松的策略；它允许您使用任何 AMI 用户名连接到任何实例。我们使用这种高度宽松的策略来确保教程的简单性，并侧重于本教程所讲授的具体配置。但在生产环境中，我们建议您将 IAM 策略配置为提供最低权限。有关示例 IAM 策略，请参阅 为 EC2 Instance Connect 授予 IAM 权限。

创建并附加一个允许您使用 EC2 Instance Connect 连接到实例的 IAM 策略

1. 首先创建 IAM 策略
   a. 打开 IAM 控制台：https://console.aws.amazon.com/iam/。
   b. 在导航窗格中，选择策略。
   c. 选择创建策略。
   d. 在指定权限页面中，请执行以下操作：
      i. 对于服务，选择 EC2 Instance Connect。
      ii. 在允许的操作下，在搜索字段中开始键入 **send** 以显示相关操作，然后选择 SendSSHPublicKey。
      iii. 在资源下，选择全部。对于生产环境，我们建议用 ARN 来指定实例，但在本教程中，您会允许所有实例。
      iv. 选择添加更多权限。
      v. 对于服务，选择 EC2。
      vi. 在允许的操作下，在搜索字段中开始键入 **describein** 以显示相关操作，然后选择 DescribeInstances。

vii. 选择下一步。

e. 在查看和创建页面中，请执行以下操作：

i. 对于 Policy name（策略名称），输入此策略的名称。

ii. 选择创建策略。

2. 然后将该策略附加到您的身份

a. 在 IAM 控制台的导航窗格中，选择 Policies（策略）。

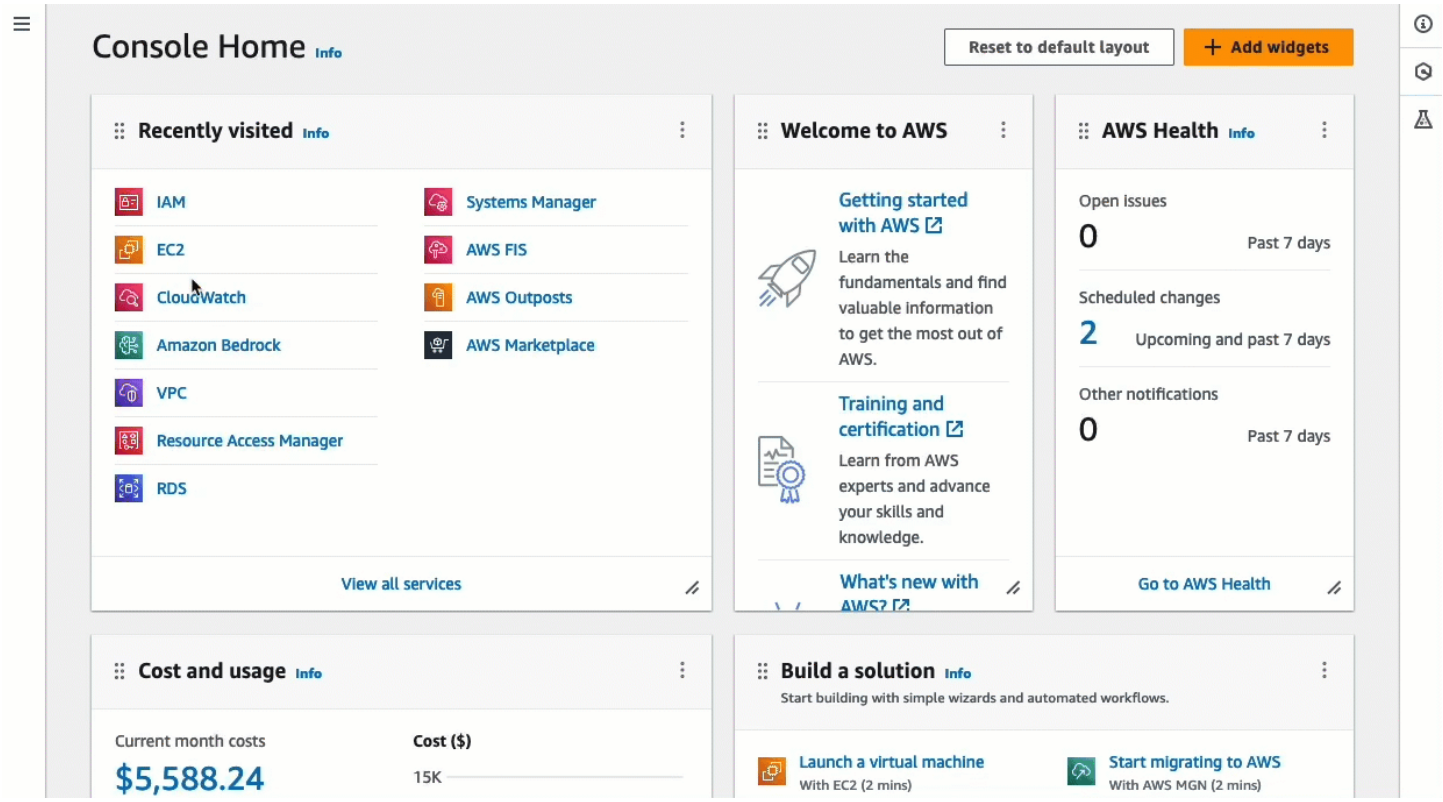
b. 在策略列表中，选中要附加的策略名称旁边的选项按钮。您可以使用搜索框筛选策略列表。

c. 依次选择操作、附加。

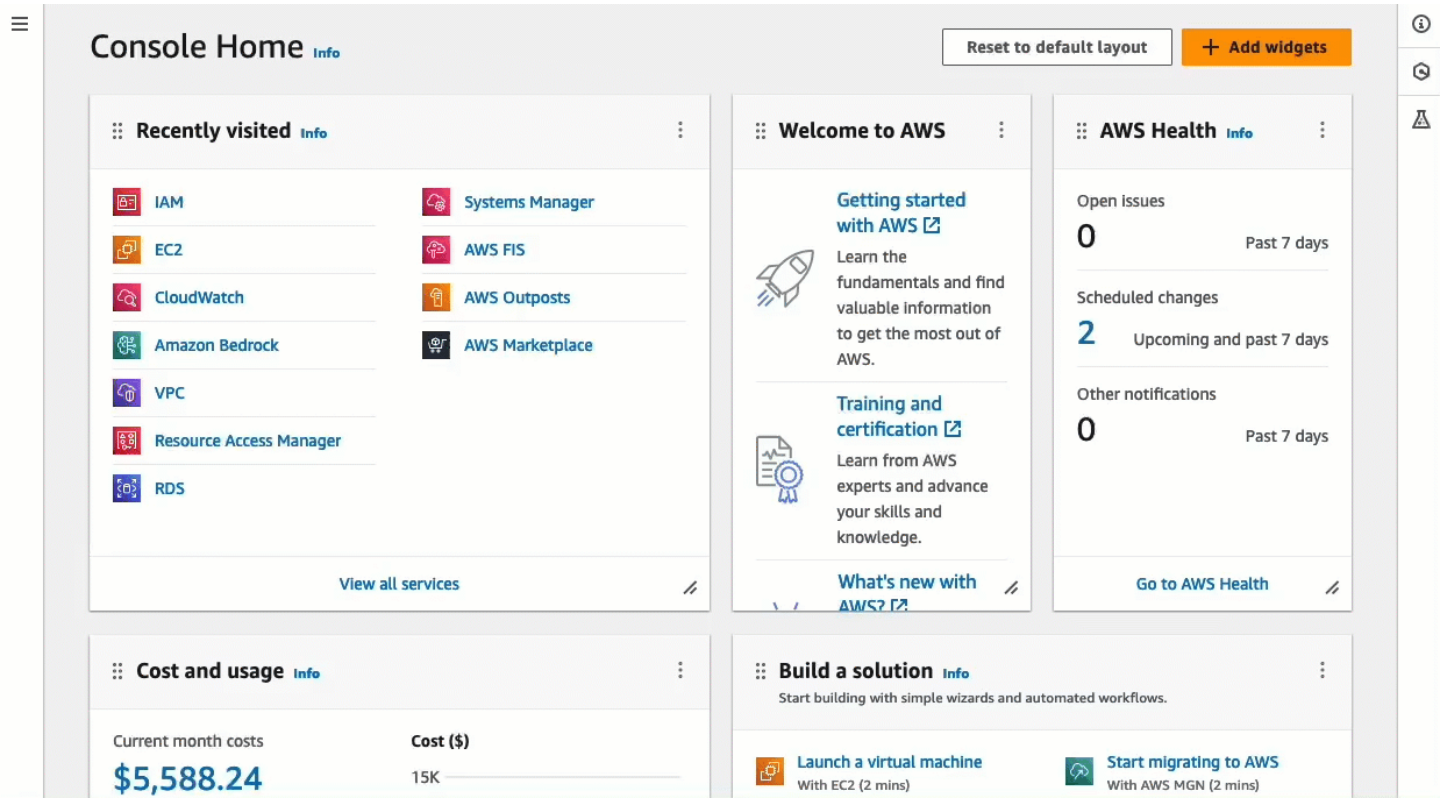
d. 在 IAM 实体下，选择您的身份（用户、用户组或角色）旁的复选框。您可以使用搜索框筛选实体列表。

e. 选择附加策略。

观看动画：创建 IAM 策略



观看动画：附加 IAM 策略



任务 2：允许从 EC2 Instance Connect 服务到实例的入站流量

在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到实例时，必须允许来自 EC2 Instance Connect 服务的流量到达实例。这与从本地计算机连接到实例不同；对于后者，您必须允许从本地计算机到实例的流量。要允许来自 EC2 Instance Connect 服务的流量，您必须创建一个安全组，以允许来自 EC2 Instance Connect 服务的 IP 地址范围的入站 SSH 流量。

Amazon 使用前缀列表来管理 IP 地址范围。EC2 Instance Connect 前缀列表的名称如下，请将 *region* 替换为区域代码：

- IPv4 前缀列表名称：`com.amazonaws.region.ec2-instance-connect`
- IPv6 前缀列表名称：`com.amazonaws.region.ipv6.ec2-instance-connect`

任务目标

您将创建一个安全组，以允许来自实例所在区域的 IPv4 前缀列表的入站 SSH 流量通过端口 22。

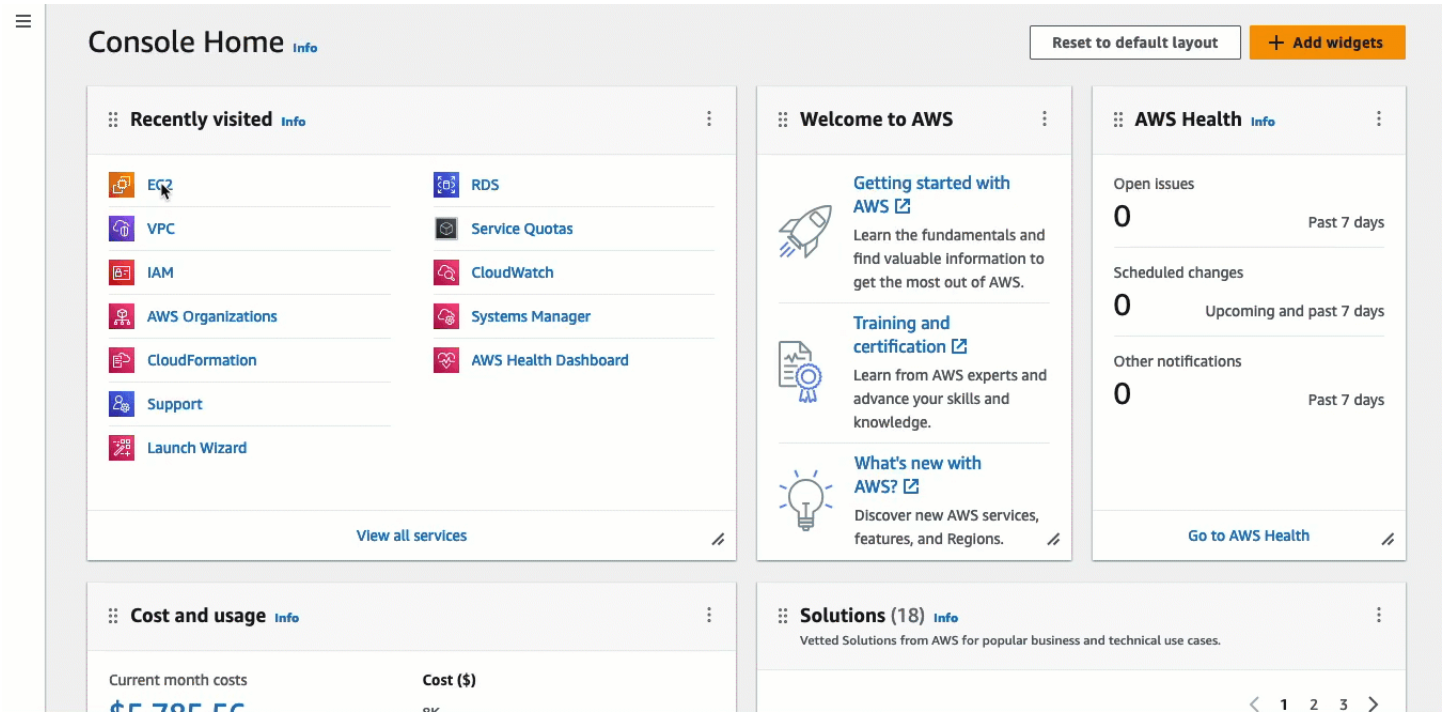
创建一个安全组，以允许从 EC2 Instance Connect 服务到实例的入站流量

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 Security Groups（安全组）。
3. 选择创建安全组。
4. 在基本详细信息下面，执行以下操作：
   a. 对于安全组名称，为您的安全组输入一个有意义的名称。
   b. 对于描述，为您的安全组输入一个有意义的描述。
5. 在入站规则下，执行以下操作：
   a. 选择添加规则。
   b. 对于 Type，选择 SSH。
   c. 对于源，请保留自定义。
   d. 在来源旁的字段中，选择 EC2 Instance Connect 的前缀列表。

   例如，假设实例位于美国东部（弗吉尼亚州北部）(us-east-1) 区域，并且用户将连接到其公有 IPv4 地址，则选择以下前缀列表：com.amazonaws.us-east-1.ec2-instance-connect
6. 选择创建安全组。

观看动画：创建安全组



任务 3：启动实例

启动实例时，必须指定包含启动实例所需信息的 AMI。您可以选择启动已预装或未预装 EC2 Instance Connect 的实例。在本任务中，我们将指定一个预装了 EC2 Instance Connect 的 AMI。

如果您启动未预装 EC2 Instance Connect 的实例，同时又需要使用 EC2 Instance Connect 连接到实例，则需要执行额外的配置步骤。这些步骤不在本教程的介绍范围之内。

任务目标

您要使用预装了 EC2 Instance Connect 的 Amazon Linux 2023 AMI 来启动实例。您还需要指定之前创建的安全组，以便能够在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到实例。由于您将使用 EC2 Instance Connect 连接到实例，这会将一个公有密钥推送到实例元数据中，因此在启动实例时无需指定 SSH 密钥。

启动可以在 Amazon EC2 控制台中使用 EC2 Instance Connect 进行连接的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 屏幕顶部的导航栏中会显示当前所在 Amazon 区域（例如，爱尔兰）。选择要在其中启动实例的区域。这一选择十分重要，因为您创建了一个允许特定区域流量的安全组，因此必须选择在同一区域启动实例。
3. 从 Amazon EC2 控制台控制面板中，选择启动实例。
4. （可选）在 Name and tags（名称与标签）下，为 Name（名称）输入实例的描述性名称。
5. 在应用程序和操作系统映像（亚马逊机器映像）下，选择快速启动。默认会选择 Amazon Linux。在亚马逊机器映像（AMI）下，已默认选择了 Amazon Linux 2023 AMI。对于此任务，请保留默认选择。
6. 对于实例类型下的实例类型，请保留默认选择，也可选择其他实例类型。
7. 在密钥对（登录）下的密钥对名称，请选择继续操作但不提供密钥对（不推荐）。使用 EC2 Instance Connect 连接到实例时，EC2 Instance Connect 会将一个密钥对推送到该实例的元数据，该密钥正是要用于连接的密钥对。
8. 在 Network settings（网络设置）下，执行以下操作：

   a. 对于自动分配公有 IP，请保留启用。

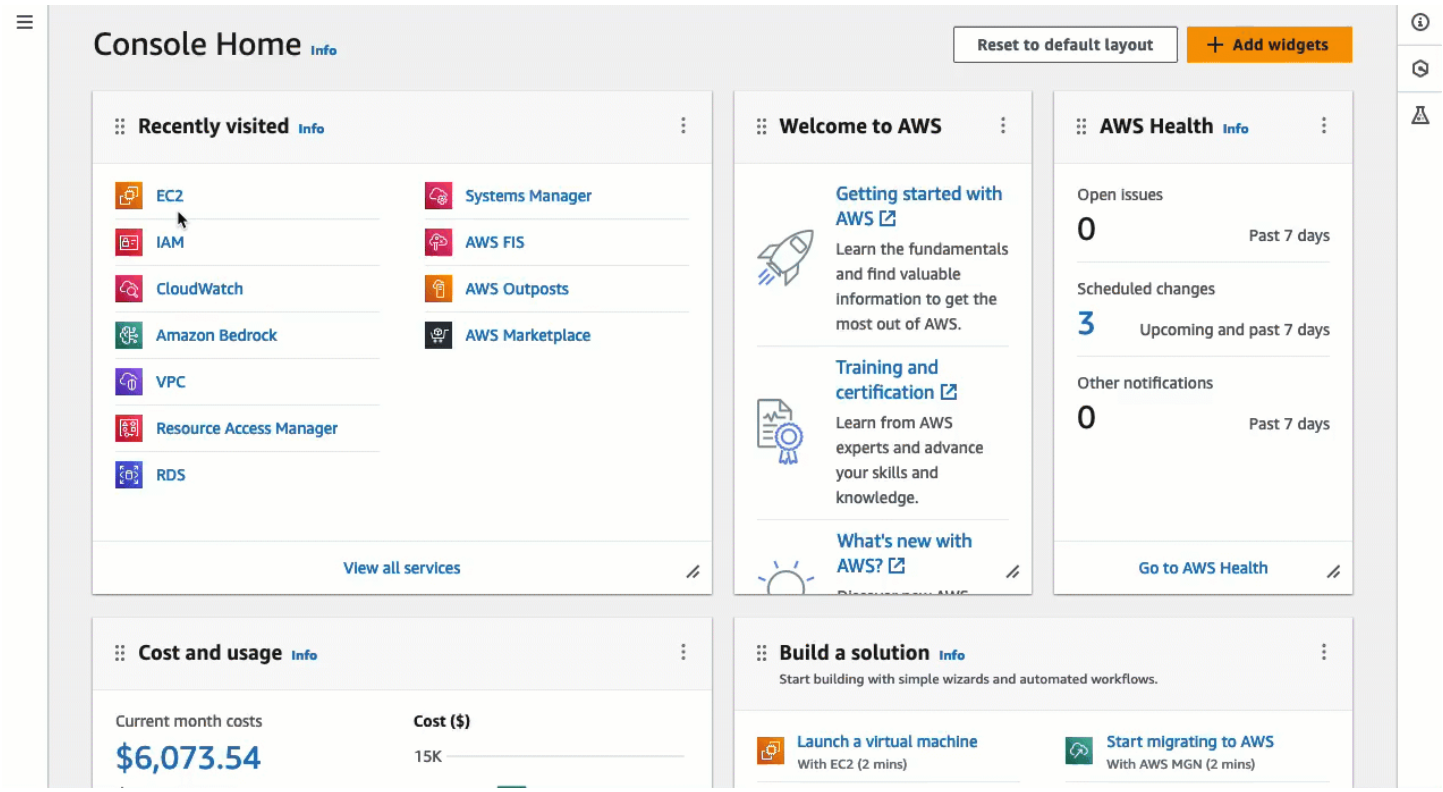
> **Note**
>
> 要在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到实例，实例必须具有一个公有 IPv4 或 IPv6 地址。

b. 对于防火墙（安全组），请选择选择现有安全组。

c. 在常用安全组下，选择您之前创建的安全组。

9. 在 Summary（摘要）面板中，选择 Launch instance（启动实例）。

观看动画：启动实例



任务 4：连接到实例

在使用 EC2 Instance Connect 连接到实例时，EC2 Instance Connect API 会将一个 SSH 公有密钥推送到实例元数据并在其中保留 60 秒。SSH 进程守护程序使用 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser` 来查找实例元数据的公有密钥以用于身份验证，然后将您连接到实例。

任务目标

在此任务中，您将在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接到实例。如果您完成了前提任务 1、2 和 3，则连接应该会成功。

连接到实例的步骤

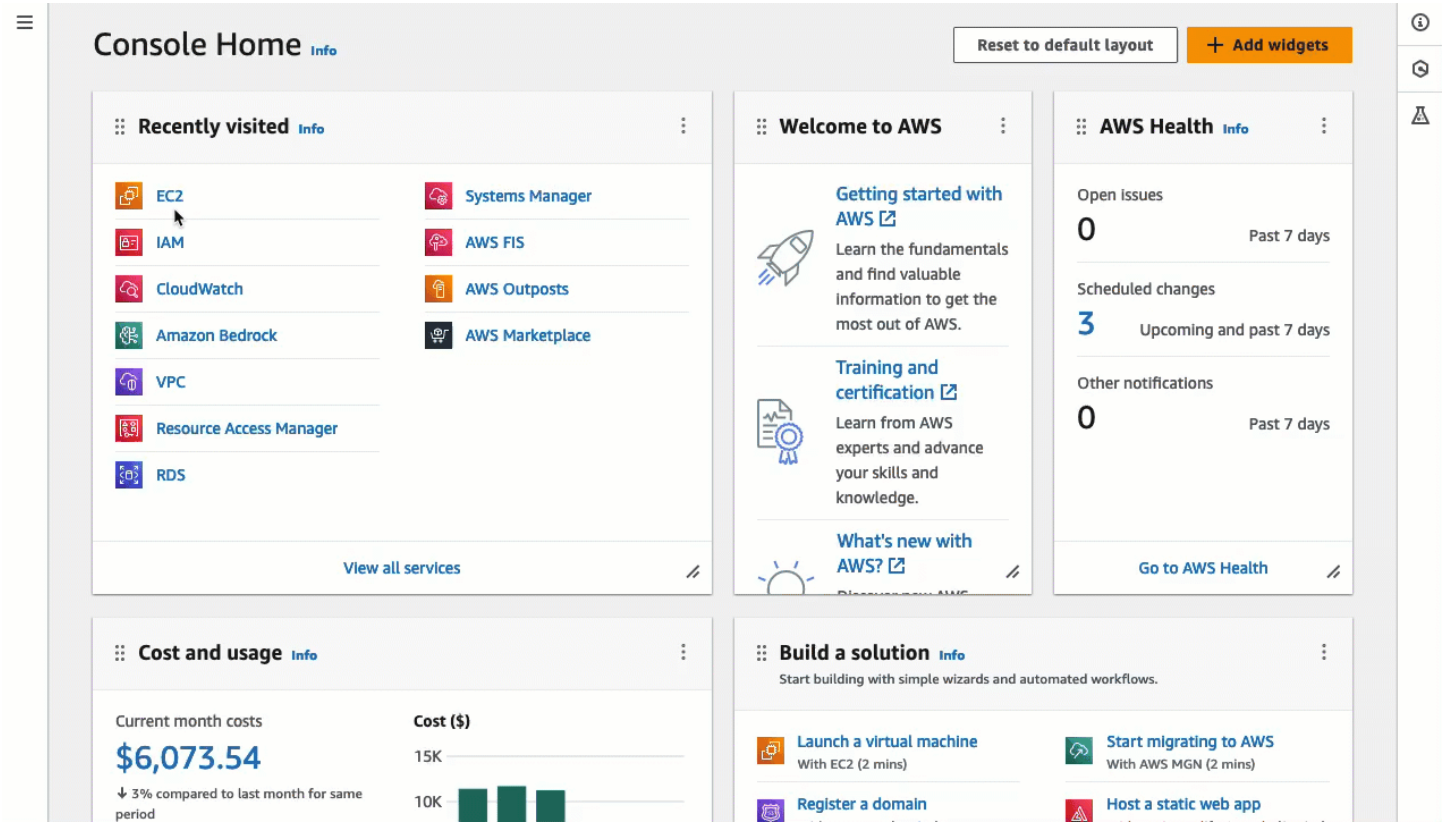
使用以下步骤连接到实例。要观看这些步骤的动画，请参阅 观看动画：连接到实例。

在 Amazon EC2 控制台中使用 EC2 Instance Connect 连接实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 屏幕顶部的导航栏中会显示当前所在 Amazon 区域（例如，爱尔兰）。选择实例所在的区域。
3. 在导航窗格中，选择 Instances (实例)。
4. 选择您的实例，然后选择连接。
5. 选择 EC2 Instance Connect 选项卡。
6. 选择使用公有 IP 连接。
7. 选择连接。

   这时将在浏览器中打开一个终端窗口，并且您已连接到实例。

观看动画：连接到实例



EC2 Instance Connect 的先决条件

以下是安装和使用 EC2 Instance Connect 的先决条件：

- 安装 EC2 Instance Connect

- 保障网络连接质量
- 允许入站 SSH 流量
- 授予权限
- 在本地计算机上安装 SSH 客户端
- 满足用户名要求

安装 EC2 Instance Connect

要使用 EC2 Instance Connect 连接到实例，该实例必须安装 EC2 Instance Connect。您可以使用预装 EC2 Instance Connect 的 AMI 启动实例，也可以在使用支持的 AMI 启动的实例上安装 EC2 Instance Connect。有关更多信息，请参阅 在您的 EC2 实例上安装 EC2 Instance Connect。

保障网络连接质量

可以将实例配置为允许用户通过互联网或通过实例的私有 IP 地址连接到您的实例。根据您的用户使用 EC2 Instance Connect 连接到实例的方式，您必须配置以下网络访问：

- 如果用户通过互联网连接到您的实例，则实例必须具有一个公有 IPv4 或 IPv6 地址，并且位于具有互联网路由的公有子网中。如果您尚未修改默认的公有子网，则其包含的互联网路由仅适用于 IPv4，而不适用于 IPv6。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用互联网网关启用 VPC 互联网访问。
- 如果用户要通过实例的私有 IPv4 地址连接到您的实例，则必须建立与您的 VPC 的私有网络连接，例如使用 Amazon Direct Connect、Amazon Site-to-Site VPN 或 VPC 对等连接，以便用户能够访问实例的私有 IP 地址。

如果您的实例没有公有 IPv4 或 IPv6 地址，并且您不希望按照上述方式配置网络访问权限，则可以考虑将 EC2 Instance Connect 端点作为 EC2 Instance Connect 的替代方案。借助 EC2 Instance Connect 端点，您可以通过 SSH 或 RDP 连接到实例，即使实例没有公有 IPv4 或 IPv6 地址。有关更多信息，请参阅 使用 Amazon EC2 控制台连接到您的 Linux 实例。

允许入站 SSH 流量

使用 Amazon EC2 控制台连接到实例时

当用户使用 Amazon EC2 控制台连接到实例时，必须允许到达实例的流量是来自 EC2 Instance Connect 服务的流量。该服务由特定的 IP 地址范围标识，Amazon 通过前缀列表来管理这些地址范围。您必须创建一个安全组，以允许来自 EC2 Instance Connect 服务的入站 SSH 流量。要进行此配置，对于入站规则，请在来源旁的字段中，选择 EC2 Instance Connect 前缀列表。

Amazon 为每个区域的 IPv4 和 IPv6 地址提供了不同的托管前缀列表。EC2 Instance Connect 前缀列表的名称如下，请将 *region* 替换为区域代码：

- IPv4 前缀列表名称：com.amazonaws.*region*.ec2-instance-connect
- IPv6 前缀列表名称：com.amazonaws.*region*.ipv6.ec2-instance-connect

有关创建安全组的说明，请参阅 任务 2：允许从 EC2 Instance Connect 服务到实例的入站流量。有关更多信息，请参阅《Amazon VPC 用户指南》中的可用的 Amazon 托管前缀列表。

使用 CLI 或 SSH 连接到实例时

确保与您实例关联的安全组允许来自您 IP 地址端口 22 或您网络上的传入 SSH 流量。默认情况下，VPC 的默认安全组不允许传入 SSH 流量。默认情况下，启动实例向导创建的安全组允许传入的 SSH 流量。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。

授予权限

您必须向使用 EC2 Instance Connect 连接到实例的每个 IAM 用户授予所需的权限。有关更多信息，请参阅 为 EC2 Instance Connect 授予 IAM 权限。

在本地计算机上安装 SSH 客户端

如果您的用户使用 SSH 进行连接，他们必须确保其本地计算机具有 SSH 客户端。

用户的本地计算机可能已默认安装 SSH 客户端。他们可以通过在命令行键入 ssh 来检查 SSH 客户端。如果他们的本地计算机无法识别该命令，可安装 SSH 客户端。有关在 Linux 或 macOS X 上安装 SSH 客户端的信息，请参阅 http://www.openssh.com。有关在 Windows 10 上安装 SSH 客户端的信息，请参阅 Windows 中的 OpenSSH。

如果用户仅使用 Amazon EC2 控制台连接到实例，则无需在本地计算机上安装 SSH 客户端。

满足用户名要求

使用 EC2 Instance Connect 连接到实例时，用户名必须满足以下要求：

- 第一个字符：必须是字母（A-Z、a-z）、数字（0-9）或下划线（_）
- 后续字符：可以是字母（A-Z、a-z）、数字 (0-9) 或以下字符：@ . _ -
- 最小长度是 1 个字符
- 最大长度为 31 个字符

为 EC2 Instance Connect 授予 IAM 权限

要使用 EC2 Instance Connect 连接到实例，您必须创建 IAM policy，从而向您的用户授予以下操作和条件的权限：

- `ec2-instance-connect:SendSSHPublicKey` 操作 - 授予将公有密钥推送到实例的权限。
- `ec2:osuser` 条件 - 指定可以将公有密钥推送到实例的操作系统用户的名称。使用用于启动实例的 AMI 的默认用户名。对于 AL2023 和 Amazon Linux 2，默认用户名为 `ec2-user`；对于 Ubuntu 则为 `ubuntu`。
- `ec2:DescribeInstances` 操作 - 使用 EC2 控制台时是必需的，因为包装程序会调用此操作。用户可能已经拥有从另一个策略调用此操作的权限。
- `ec2:DescribeVpcs` 操作 – 连接到 IPv6 地址时为必填项。

考虑限制对特定 EC2 实例的访问。否则，具有适用于 `ec2-instance-connect:SendSSHPublicKey` 操作的权限的所有 IAM 主体可以连接到所有 EC2 实例。您可以通过指定资源 ARN 或使用资源标签作为条件键来限制访问。

有关更多信息，请参阅 Amazon EC2 Instance Connect 的操作、资源和条件键。

有关如何创建 IAM policy 的信息，请参阅 IAM 用户指南中的创建 IAM policy。

允许用户连接到特定实例

以下 IAM policy 授予连接到特定实例的权限，这些实例由其资源 ARN 标识。

在以下示例 IAM policy 中，指定了以下操作和条件：

- `ec2-instance-connect:SendSSHPublicKey` 操作向用户授予连接到两个实例的权限，这两个实例由资源 ARN 指定。要向用户授予连接到所有 EC2 实例的权限，请将资源 ARN 替换为 * 通配符。
- 只有在连接时指定 *`ami-username`* 时，`ec2:osuser` 条件才会授予连接到实例的权限。
- 指定 `ec2:DescribeInstances` 操作向使用控制台连接到您的实例的用户授予权限。如果您的用户仅使用 SSH 客户端连接到您的实例，可以省略 `ec2:DescribeInstances`。注意，`ec2:Describe*` API 操作不支持资源级权限。因此，`Resource` 元素中需要 * 通配符。
- 指定 `ec2:DescribeVpcs` 操作向使用 IPv6 地址从控制台连接到实例的用户授予权限。如果用户仅使用公有 IPv4 地址，则可以忽略 `ec2:DescribeVpcs`。注意，`ec2:Describe*` API 操作不支持资源级权限。因此，`Resource` 元素中需要 * 通配符。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:111122223333:instance/i-0598c7d356eba48d7"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

允许用户连接到具有特定标签的实例

基于属性的访问权限控制（ABAC）是一种授权策略，该策略根据可附加到用户和 Amazon 资源的标签来定义权限。您可以使用资源标签来控制对实例的访问。有关使用标签控制对Amazon资源的访问的更多信息，请参阅 IAM 用户指南中的控制对Amazon资源的访问。

在以下示例 IAM policy 中，`ec2-instance-connect:SendSSHPublicKey` 操作向用户授予连接到任何实例的权限（由资源 ARN 中的 * 通配符指示），条件是该实例的资源标签带有 key=`tag-key` 和 value=`tag-value`。

指定 ec2:DescribeInstances 操作向使用控制台连接到您的实例的用户授予权限。如果用户仅使用 SSH 客户端连接到实例，可以省略 ec2:DescribeInstances。注意，ec2:Describe* API 操作不支持资源级权限。因此，Resource 元素中需要 * 通配符。

指定 ec2:DescribeVpcs 操作向使用 IPv6 地址从控制台连接到实例的用户授予权限。如果用户仅使用公有 IPv4 地址，则可以忽略 ec2:DescribeVpcs。注意，ec2:Describe* API 操作不支持资源级权限。因此，Resource 元素中需要 * 通配符。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/tag-key": "tag-value"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

在您的 EC2 实例上安装 EC2 Instance Connect

要使用 EC2 Instance Connect 连接到 Linux 实例，该实例必须安装 EC2 Instance Connect。安装 EC2 Instance Connect 将在实例上配置 SSH 进程守护程序。

有关 EC2 Instance Connect 程序包的更多信息，请参阅 GitHub 网站上的 aws/aws-ec2-instance-connect-config。

> **Note**
>
> 如果为 SSH 身份验证配置了 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser` 设置，则 EC2 Instance Connect 安装不会更新它们。因此，无法使用 EC2 Instance Connect。

安装必备组件

安装 EC2 Instance Connect 之前，务必确认已满足以下先决条件。

- 验证实例使用了以下某一项：
  - 版本 2.0.20190618 之前的 Amazon Linux 2 *
  - AL2023 AMI 最低版本或经 Amazon ECS 优化的 AMI
  - CentOS Stream 8 和 9
  - Red Hat Enterprise Linux（RHEL）8 和 9
  - Ubuntu 16.04 和 18.04 *

  > **Tip**
  >
  > * 对于 Amazon Linux 2、macOS 和 Ubuntu：如果使用比上述版本更高的版本启动实例，则会预装 EC2 Instance Connect，无需手动安装。

- 验证 EC2 Instance Connect 的一般先决条件。

  有关更多信息，请参阅 EC2 Instance Connect 的先决条件。
- 确认满足使用本地计算机上的 SSH 客户端连接到实例的先决条件。

  有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
- 获取实例 ID。

  您可以使用 Amazon EC2 控制台获取实例 ID（从 Instance ID (实例 ID) 列）。如果您愿意，可以使用 describe-instances (Amazon CLI) 或 Get-EC2Instance (Amazon Tools for Windows PowerShell) 命令。

手动安装 EC2 Instance Connect

> **Note**
>
> 如果使用以下某一项 AMI 启动了实例，则该实例已预安装 EC2 Instance Connect，因此可以跳过此过程：
>
> - AL2023 标准 AMI
> - Amazon Linux 2 2.0.20190618 或更高版本
> - macOS Sonoma 14.2.1 或更高版本
> - macOS Ventura 13.6.3 或更高版本
> - macOS Monterey 12.7.2 或更高版本
> - Ubuntu 20.04 或更高版本

根据实例的操作系统，使用以下过程之一安装 EC2 Instance Connect。

Amazon Linux 2

在使用 Amazon Linux 2 启动的实例上安装 EC2 Instance Connect

1. 使用 SSH 连接到您的实例。

   在以下命令中，将示例值替换为自己的值。可以使用在启动实例时为其分配的 SSH 密钥对以及用于启动实例的 AMI 的默认用户名。对于 Amazon Linux 2，默认用户名为 `ec2-user`。

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   有关连接到实例的更多信息，请参阅使用 SSH 客户端连接到 Linux 实例。

2. 在实例上安装 EC2 Instance Connect 程序包。

   ```
   [ec2-user ~]$ sudo yum install ec2-instance-connect
   ```

   您将会在 `/opt/aws/bin/` 文件夹中看到三个新脚本：

   ```
   eic_curl_authorized_keys
   eic_parse_authorized_keys
   ```

```
eic_run_authorized_keys
```

3. （可选）验证是否在实例上成功安装了 EC2 Instance Connect。

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config
```

如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，则成功安装 EC2 Instance Connect：

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand 设置 eic_run_authorized_keys 文件以从实例元数据中查找密钥
- AuthorizedKeysCommandUser 将系统用户设置为 ec2-instance-connect

> **Note**
>
> 如果以前配置了 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser，则 EC2 Instance Connect 安装不会更改这些值，并且无法使用 EC2 Instance Connect。

CentOS

在通过 CentOS 启动的实例上安装 EC2 Instance Connect

1. 使用 SSH 连接到实例。

   在以下命令中，将示例值替换为自己的值。可以使用在启动实例时为其分配的 SSH 密钥对以及用于启动实例的 AMI 的默认用户名。对于 CentOS，默认用户名为 centos 或 ec2-user。

```
$ ssh -i my_ec2_private_key.pem centos@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

   有关连接到实例的更多信息，请参阅使用 SSH 客户端连接到 Linux 实例。

2. 如果使用 HTTP 或 HTTPS 代理，则必须在当前 Shell 会话中设置 http_proxy 或 https_proxy 环境变量。

如果不使用代理，则可跳过此步骤。

- 对于 HTTP 代理服务器，请运行以下命令：

```
$ export http_proxy=http://hostname:port
$ export https_proxy=http://hostname:port
```

- 对于 HTTPS 代理服务器，请运行以下命令：

```
$ export http_proxy=https://hostname:port
$ export https_proxy=https://hostname:port
```

3. 运行以下命令，在实例上安装 EC2 Instance Connect 软件包。

适用于 CentOS 的 EC2 Instance Connect 配置文件在 Red Hat Package Manager（RPM）包中提供，适用于 CentOS 8 和 CentOS 9，以及在 Intel/AMD（x86_64）或 ARM（AArch64）上运行的实例类型有不同的 RPM 包。

使用适用于操作系统和 CPU 架构的命令块。

- CentOS 8

  Intel/AMD（x86_64）

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-5.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

  ARM（AArch64）

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
```

```
connect-2.0.0-5.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- CentOS 9

  Intel/AMD（x86_64）

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
  west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
  connect-2.0.0-5.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
  west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
  selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
  selinux.rpm
  [ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  ```

  ARM（AArch64）

  ```
  [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
  west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
  connect-2.0.0-5.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm
  [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
  west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
  selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
  selinux.rpm
  [ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
  connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
  ```

在 /opt/aws/bin/ 文件夹中会看到以下新脚本：

```
eic_run_authorized_keys
```

4. （可选）验证是否在实例上成功安装了 EC2 Instance Connect。

   - 对于 CentOS 8：

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-connect.conf
     ```

   - 对于 CentOS 9：

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

   如果 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser` 行包含以下值，则成功安装 EC2 Instance Connect：

   ```
   AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - `AuthorizedKeysCommand` 设置 `eic_run_authorized_keys` 文件以从实例元数据中查找密钥
   - `AuthorizedKeysCommandUser` 将系统用户设置为 `ec2-instance-connect`

   > **Note**
   >
   > 如果以前配置了 `AuthorizedKeysCommand` 和 `AuthorizedKeysCommandUser`，则 EC2 Instance Connect 安装不会更改这些值，并且无法使用 EC2 Instance Connect。

RHEL

在使用 Red Hat Enterprise Linux（RHEL）启动的实例上安装 EC2 Instance Connect

1. 使用 SSH 连接到实例。

在以下命令中，将示例值替换为自己的值。可以使用在启动实例时为其分配的 SSH 密钥对以及用于启动实例的 AMI 的默认用户名。对于 RHEL，默认用户名为 `ec2-user` 或 `root`。

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

有关连接到实例的更多信息，请参阅使用 SSH 客户端连接到 Linux 实例。

2. 如果使用 HTTP 或 HTTPS 代理，则必须在当前 Shell 会话中设置 `http_proxy` 或 `https_proxy` 环境变量。

   如果不使用代理，则可跳过此步骤。

   - 对于 HTTP 代理服务器，请运行以下命令：

     ```
     $ export http_proxy=http://hostname:port
     $ export https_proxy=http://hostname:port
     ```

   - 对于 HTTPS 代理服务器，请运行以下命令：

     ```
     $ export http_proxy=https://hostname:port
     $ export https_proxy=https://hostname:port
     ```

3. 运行以下命令，在实例上安装 EC2 Instance Connect 软件包。

   在 Red Hat Package Manager（RPM）包中提供适用于 RHEL 的 EC2 Instance Connect 配置文件，适用于 RHEL 8 和 RHEL 9，以及在 Intel/AMD（x86_64）或 ARM（AArch64）上运行的实例类型有不同的 RPM 包。

   使用适用于操作系统和 CPU 架构的命令块。

   - RHEL 8

     Intel/AMD（x86_64）

     ```
     [ec2-user ~]$ mkdir /tmp/ec2-instance-connect
     [ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
     west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
     connect-2.0.0-5.rhel8.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
     connect.rpm
     ```

```
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

ARM（AArch64）

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
connect-2.0.0-5.rhel8.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

- RHEL 9

  Intel/AMD（x86_64）

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_amd64/ec2-instance-
connect-2.0.0-5.rhel9.x86_64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_amd64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

  ARM（AArch64）

```
[ec2-user ~]$ mkdir /tmp/ec2-instance-connect
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-
west-2.s3.us-west-2.amazonaws.com/latest/linux_arm64/ec2-instance-
```

```
connect-2.0.0-5.rhel9.aarch64.rpm -o /tmp/ec2-instance-connect/ec2-instance-
connect.rpm
[ec2-user ~]$ curl https://amazon-ec2-instance-connect-us-west-2.s3.us-
west-2.amazonaws.com/latest/linux_arm64/ec2-instance-connect-
selinux-2.0.0-5.noarch.rpm -o /tmp/ec2-instance-connect/ec2-instance-connect-
selinux.rpm
[ec2-user ~]$ sudo yum install -y /tmp/ec2-instance-connect/ec2-instance-
connect.rpm /tmp/ec2-instance-connect/ec2-instance-connect-selinux.rpm
```

在 /opt/aws/bin/ 文件夹中会看到以下新脚本：

```
eic_run_authorized_keys
```

4. （可选）验证是否在实例上成功安装了 EC2 Instance Connect。

   - 对于 RHEL 8：

     ```
     [ec2-user ~]$ sudo less /lib/systemd/system/sshd.service.d/ec2-instance-
     connect.conf
     ```

   - 对于 RHEL 9：

     ```
     [ec2-user ~]$ sudo less /etc/ssh/sshd_config.d/60-ec2-instance-connect.conf
     ```

   如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，则成功安装 EC2 Instance Connect：

   ```
   AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
   AuthorizedKeysCommandUser ec2-instance-connect
   ```

   - AuthorizedKeysCommand 设置 eic_run_authorized_keys 文件以从实例元数据中查找密钥
   - AuthorizedKeysCommandUser 将系统用户设置为 ec2-instance-connect

> **Note**
>
> 如果以前配置了 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser，则 EC2 Instance Connect 安装不会更改这些值，并且无法使用 EC2 Instance Connect。

Ubuntu

在使用 Ubuntu 16.04 或更高版本启动的实例上安装 EC2 Instance Connect

1. 使用 SSH 连接到实例。

   在以下命令中，将示例值替换为自己的值。可以使用在启动实例时为其分配的 SSH 密钥对以及用于启动实例的 AMI 的默认用户名。对于 Ubuntu AMI，用户名为 ubuntu。

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

   有关连接到实例的更多信息，请参阅使用 SSH 客户端连接到 Linux 实例。

2. （可选）确保您的实例具有最新 Ubuntu AMI。

   使用以下命令更新实例上的所有程序包。

   ```
   ubuntu:~$ sudo apt-get update
   ```

   ```
   ubuntu:~$ sudo apt-get upgrade
   ```

3. 在实例上安装 EC2 Instance Connect 程序包。

   ```
   ubuntu:~$ sudo apt-get install ec2-instance-connect
   ```

   您将会在 /usr/share/ec2-instance-connect/ 文件夹中看到三个新脚本：

   ```
   eic_curl_authorized_keys
   eic_parse_authorized_keys
   eic_run_authorized_keys
   ```

4. （可选）验证是否在实例上成功安装了 EC2 Instance Connect。

```
ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

如果 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser 行包含以下值，则成功安装 EC2 Instance Connect：

```
AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %
%u %%f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand 设置 eic_run_authorized_keys 文件以从实例元数据中查找密钥
- AuthorizedKeysCommandUser 将系统用户设置为 ec2-instance-connect

> **Note**
>
> 如果以前配置了 AuthorizedKeysCommand 和 AuthorizedKeysCommandUser，则 EC2 Instance Connect 安装不会更改这些值，并且无法使用 EC2 Instance Connect。

使用 EC2 Instance Connect 连接到 Linux 实例

以下说明旨在介绍如何通过 Amazon EC2 控制台、Amazon CLI 或 SSH 客户端使用 EC2 Instance Connect 连接到 Linux 实例。

在使用 EC2 Instance Connect 通过控制台或 Amazon CLI 连接到实例时，EC2 Instance Connect API 会将一个 SSH 公有密钥推送到实例元数据并在其中保留 60 秒。附加到用户的 IAM 策略负责授权此操作。如果您更喜欢使用自己的 SSH 密钥，则可以使用 SSH 客户端，并使用 EC2 Instance Connect 将您的 SSH 密钥显式推送到实例。

注意事项

使用 EC2 Instance Connect 连接到实例后，连接会一直持续到 SSH 会话终止。连接的持续时间并不是由 IAM 凭证的持续时间决定的。如果您的 IAM 凭证过期，连接将继续保持。使用 EC2 Instance Connect 控制台体验时，如果 IAM 凭证过期，请关闭浏览器页面终止连接。使用自己的 SSH 客户端和 EC2 Instance Connect 推送密钥时，可以设置 SSH 超时值来自动终止 SSH 会话。

要求

开始操作之前，务必查看先决条件。

连接选项

- 使用 Amazon EC2 控制台连接
- 使用 Amazon CLI进行连接
- 使用您自己的密钥和 SSH 客户端进行连接
- 故障排除

使用 Amazon EC2 控制台连接

您可以通过 Amazon EC2 控制台使用 EC2 Instance Connect 连接到实例。

要求

要使用 Amazon EC2 控制台进行连接，实例必须具有一个公有 IPv4 或 IPv6 地址。如果实例只有私有 IPv4 地址，则可使用 ec2-instance-connect Amazon CLI 命令进行连接。

使用 Amazon EC2 控制台连接到实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例，然后选择连接。
4. 选择 EC2 Instance Connect 选项卡。
5. 选择使用公有 IP 连接。
6. 如果可以选择，请选择要连接到的 IP 地址。否则，系统会自动选择 IP 地址。
7. 对于用户名，请验证用户名。
8. 选择连接以建立连接。浏览器内终端窗口随即打开。

使用 Amazon CLI进行连接

可以使用 ec2-instance-connect Amazon CLI 命令，通过 SSH 客户端连接到实例。EC2 Instance Connect 尝试根据指定的连接类型，按预定义顺序使用可用的 IP 地址建立连接。如果 IP 地址不可用，它会自动尝试顺序中的下一个 IP 地址。

连接类型

`auto`（默认）

EC2 Instance Connect 尝试按以下顺序使用实例的 IP 地址并使用相应的连接类型进行连接：

1. 公有 IPv4：`direct`
2. 私有 IPv4：`eice`
3. IPv6：`direct`

`direct`

EC2 Instance Connect 尝试按以下顺序使用实例的 IP 地址进行连接：

1. 公有 IPv4
2. IPv6
3. 私有 IPv4（不通过 EC2 Instance Connect 端点进行连接）

`eice`

EC2 Instance Connect 尝试使用实例的私有 IPv4 地址和 EC2 Instance Connect 端点进行连接。

> **Note**
>
> 将来，我们可能会更改 `auto` 连接类型的行为。为确保使用所需的连接类型，我们建议您将 `--connection-type` 明确设置为 `direct` 或 `eice`。

要求

您必须使用 Amazon CLI 版本 2。有关更多信息，请参阅 Install or update to the latest version of the Amazon CLI。

要使用实例 ID 连接到实例

如果只知道实例 ID，并想让 EC2 Instance Connect 确定连接到实例时要使用的连接类型，请使用 ec2-instance-connect ssh CLI 命令及实例 ID。

```
aws ec2-instance-connect ssh --instance-id i-1234567890example
```

要使用实例 ID 和 EC2 Instance Connect Endpoint 连接到实例

如果您想通过 EC2 Instance Connect Endpoint 连接到您的实例，请使用前面的命令并通过 `eice` 值指定 `--connection-type` 参数。

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --connection-type eice
```

要使用实例 ID 和您自己的私有密钥文件连接到实例

如果您想使用自己的私有密钥通过 EC2 Instance Connect Endpoint 连接到您的实例，请指定实例 ID 和私有密钥文件的路径。不要在路径中包含 *file://*；以下示例将会失败：*file:///path/to/key*。

```
aws ec2-instance-connect ssh --instance-id i-1234567890example --private-key-file /path/to/key.pem
```

> **Tip**
>
> 如果使用这些命令时出现错误，务必确保使用的是 Amazon CLI 版本 2，因为 `ssh` 命令仅在此主要版本中可用。我们还建议定期更新到 Amazon CLI 版本 2 的最新次要版本，以便使用最新功能。有关更多信息，请参阅《Amazon Command Line Interface 开发人员指南》中的关于 Amazon CLI 版本 2。

使用您自己的密钥和 SSH 客户端进行连接

您可以使用自己的 SSH 密钥，并在使用 EC2 Instance Connect API 时从您选择的 SSH 客户端连接到您的实例。这使您能够从将公有密钥推送到实例的 EC2 Instance Connect 功能中受益。此连接方法适用于具有公有和私有 IP 地址的实例。

要求

- 密钥对的要求
  - 支持的类型：RSA（OpenSSH 和 SSH2）和 ED25519
  - 支持的长度：2048 和 4096
  - 有关更多信息，请参阅 使用第三方工具创建密钥对，并将公有密钥导入 Amazon EC2。
- 当连接到仅具有私有 IP 地址的实例时，启动 SSH 会话的本地计算机必须连接到 EC2 Instance Connect 服务端点（以便将 SSH 公有密钥推送到实例）且与实例的私有 IP 地址有网络连接，才能建立 SSH 会话。EC2 Instance Connect 服务端点可以通过互联网或 Amazon Direct Connect 公共虚

拟接口访问。要连接到实例的私有 IP 地址，您可以利用 Amazon Direct Connect、Amazon Site-to-Site VPN 或 VPC 对等连接等服务。

使用您自己的密钥和任何 SSH 客户端连接到实例

1. （可选）生成新的 SSH 私有密钥和公有密钥

   您可以使用以下命令生成新的 SSH 私有密钥和公有密钥（my_key 和 my_key.pub）：

   ```
   ssh-keygen -t rsa -f my_key
   ```

2. 将 SSH 公有密钥推送到实例

   使用 send-ssh-public-key 命令可将 SSH 公有密钥推送到实例。如果使用 AL2023 或 Amazon Linux 2 启动实例，则 AMI 的默认用户名为 ec2-user。如果使用 Ubuntu 启动实例，则 AMI 的默认用户名为 ubuntu。

   以下示例将公有密钥推送到指定的可用区中的指定实例，以对 ec2-user 进行身份验证。

   ```
   aws ec2-instance-connect send-ssh-public-key \
       --region us-west-2 \
       --availability-zone us-west-2b \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --instance-os-user ec2-user \
       --ssh-public-key file://my_key.pub
   ```

3. 使用私有密钥连接到实例

   从实例元数据中删除公有密钥之前，可以使用 ssh 命令通过私有密钥连接到实例（在 60 秒后删除）。指定与公有密钥对应的私有密钥、用于启动实例的 AMI 的默认用户名以及实例的公有 DNS 名称（如果通过私有网络连接，请指定私有 DNS 名称或 IP 地址）。添加 IdentitiesOnly=yes 选项以确保仅使用 ssh config 中的文件和指定的密钥进行连接。

   ```
   ssh -o "IdentitiesOnly=yes" -i my_key ec2-
   user@ec2-198-51-100-1.compute-1.amazonaws.com
   ```

   以下示例使用 timeout 3600 将 SSH 会话设置为 1 小时后终止。会话终止后，会话期间启动的进程可能会继续在您的实例上运行。

```
timeout 3600 ssh -o “IdentitiesOnly=yes” -i my_key ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

故障排除

如果您在尝试连接到实例时遇到错误，请参阅下文：

- 排查 Amazon EC2 Linux 实例的连接问题
- 如何排查使用 EC2 Instance Connect 连接 EC2 实例时遇到的问题？

卸载 EC2 Instance Connect

要禁用 EC2 Instance Connect，请连接到 Linux 实例并卸载操作系统上安装的 `ec2-instance-connect` 程序包。如果 `sshd` 配置与您安装 EC2 Instance Connect 时设置的配置匹配，卸载 `ec2-instance-connect` 还会删除 `sshd` 配置。如果在安装 EC2 Instance Connect 后修改了 `sshd` 配置，您必须手动更新该配置。

Amazon Linux

您可以在 AL2023 和 Amazon Linux 2 2.0.20190618 或更高版本上卸载 EC2 Instance Connect，其中预配置了 EC2 Instance Connect。

在使用 Amazon Linux 启动的实例上卸载 EC2 Instance Connect

1. 使用 SSH 连接到实例。指定启动实例时用于实例的 SSH 密钥对，以及 AL2023 和 Amazon Linux 2 AMI 的默认用户名 (`ec2-user`)。

   例如，以下 ssh 命令使用密钥对 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com` 连接到具有公有 DNS 名称 `my_ec2_private_key.pem` 的实例。

   ```
   $ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

2. 使用 `ec2-instance-connect` 命令卸载 yum 软件包。

   ```
   [ec2-user ~]$ sudo yum remove ec2-instance-connect
   ```

Ubuntu

在使用 Ubuntu AMI 启动的实例上卸载 EC2 Instance Connect

1. 使用 SSH 连接到实例。指定在启动实例时用于实例的 SSH 密钥对以及 Ubuntu AMI 的默认用户名，即 ubuntu。

   例如，以下 ssh 命令使用密钥对 ec2-a-b-c-d.us-west-2.compute.amazonaws.com 连接到具有公有 DNS 名称 my_ec2_private_key.pem 的实例。

   ```
   $ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
   west-2.compute.amazonaws.com
   ```

2. 使用 ec2-instance-connect 命令卸载 apt-get 软件包。

   ```
   ubuntu:~$ sudo apt-get remove ec2-instance-connect
   ```

使用私有 IP 地址和 EC2 Instance Connect Endpoint 连接到您的实例

借助 EC2 Instance Connect 端点，您可以通过互联网安全地连接到实例，无需使用堡垒主机，也不需要您的虚拟私有云（VPC）直接连接到互联网。

优势

- 您可以连接到实例，而实例无需具有公有 IPv4 或 IPv6 地址。Amazon 对所有公有 IPv4 地址收费，包括与正在运行的实例关联的公有 IPv4 地址，以及弹性 IP 地址。有关更多信息，请参阅 Amazon VPC 定价页面中的公有 IPv4 地址定价选项卡。
- 您可以通过互联网连接到实例，并且您的 VPC 也无需通过互联网网关直接建立互联网连接。
- 您可以使用 IAM 策略和权限来控制创建 EC2 Instance Connect 端点以及使用此端点连接到实例的权限。
- 所有连接到实例的尝试，无论成功还是失败，都将记录到 CloudTrail。

定价

使用 EC2 Instance Connect 端点不会产生额外的成本。如果您使用 EC2 Instance Connect 端点连接到位于其他可用区的实例，则将产生跨可用区传输数据的额外费用。

内容

- 工作原理
- 注意事项
- 授予使用 EC2 Instance Connect 端点的权限
- EC2 Instance Connect Endpoint 安全组
- 创建 EC2 Instance Connect Endpoint
- 修改 EC2 Instance Connect Endpoint
- 删除 EC2 Instance Connect 端点
- 使用 EC2 Instance Connect 端点连接到 Amazon EC2 实例
- 通过 EC2 Instance Conect Endpoint 建立的日志连接
- EC2 Instance Connect Endpoint 的服务相关角色
- EC2 Instance Connect 端点的限额

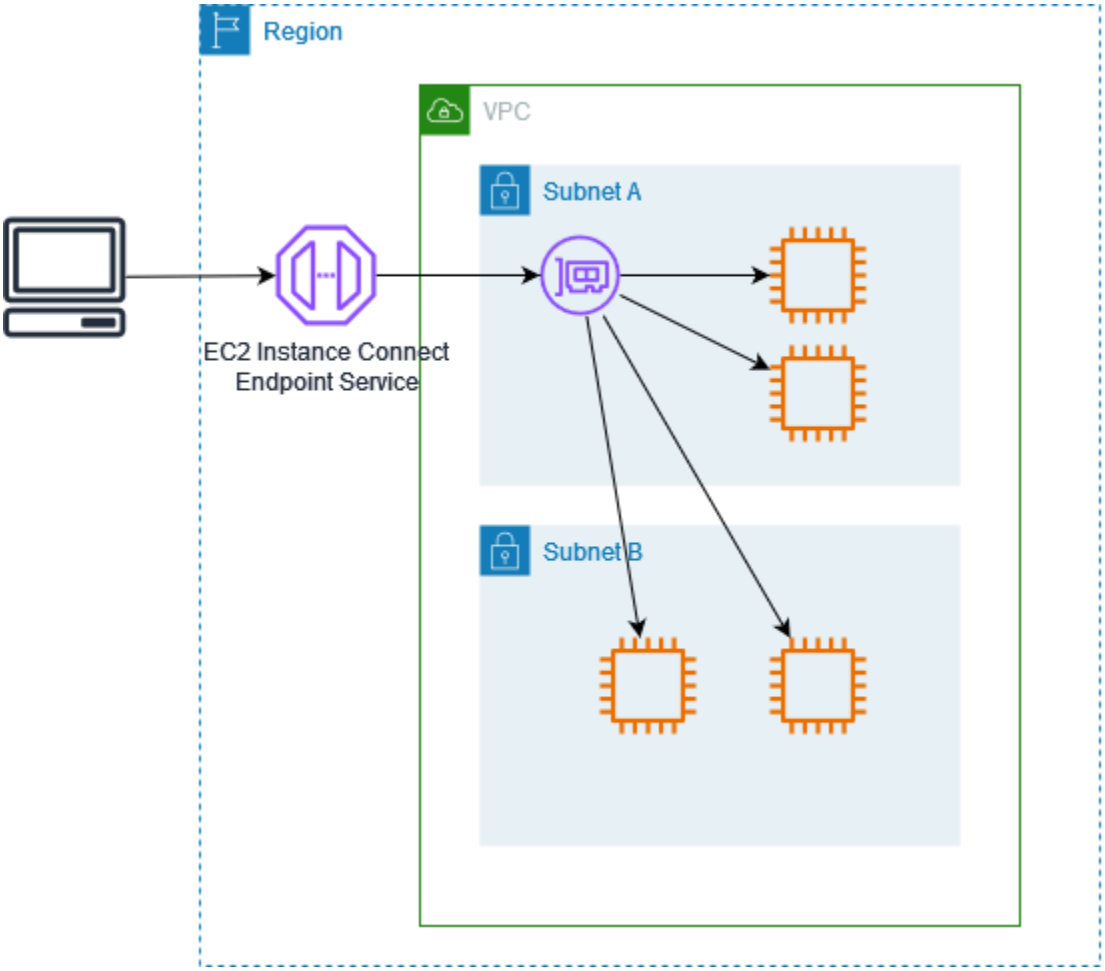
工作原理

EC2 Instance Connect 端点是一种身份感知型 TCP 代理。EC2 Instance Connect 端点服务将使用您的 IAM 实体的凭证，建立从您的计算机到该端点的私有隧道。流量在到达您的 VPC 之前已通过身份验证和授权。

您可以配置其他安全组规则来限制实例的入站流量。例如，您可以使用入站规则仅允许来自 EC2 Instance Connect 端点的流量通过管理端口。

您可以配置路由表规则来允许该端点连接到 VPC 的任何子网中的任何实例。

下图显示了用户如何使用 EC2 Instance Connect 端点从互联网连接到实例。首先，在子网 A 中创建一个 EC2 Instance Connect 端点。我们为子网中为该端点创建一个网络接口，以作为发往 VPC 中实例的流量的入口点。如果子网 B 的路由表允许来自子网 A 的流量，则可以使用该端点连接到子网 B 中的实例。



注意事项

开始之前，请注意以下因素：

- EC2 Instance Connect 端点专用于管理流量使用案例，不适用于大容量数据传输。大容量数据传输将被节流。
- 您可以创建一个 EC2 Instance Connect 端点，以支持访问具有私有 IPv4 地址或 IPv6 地址的实例的流量。端点的 IP 地址类型必须与实例的 IP 地址匹配。您可以创建支持所有 IP 地址类型的端点。
- （Linux 实例）如果您使用自己的密钥对，则可以使用任何 Linux AMI。否则，实例必须已经安装了 EC2 Instance Connect。有关哪些 AMI 包含 EC2 Instance Connect 以及如何将其安装在其他支持的 AMI 上的信息，请参阅 安装 EC2 Instance Connect。
- 您可以为 EC2 Instance Connect 端点分配一个安全组。如果未分配安全组，则将使用 VPC 的默认安全组。EC2 Instance Connect 端点的安全组必须允许指向目标实例的出站流量。有关更多信息，请参阅 EC2 Instance Connect Endpoint 安全组。

- 您可以将 EC2 Instance Connect 端点配置为在将请求路由到实例时保留客户端的源 IP 地址。否则，网络接口的 IP 地址将成为所有传入流量的客户端 IP 地址。
  - 如果您开启了客户端 IP 保留功能，则实例的安全组必须允许来自客户端的流量。此外，这些实例必须与 EC2 Instance Connect 端点位于同一 VPC 中。
  - 如果您关闭了客户端 IP 保留功能，则实例的安全组必须允许来自 VPC 的流量。这是默认值。
  - 只有 IPv4 EC2 Instance Connect 端点支持客户端 IP 保留。要使用客户端 IP 保留，EC2 Instance Connect 端点的 IP 地址类型必须为 IPv4。当 IP 地址类型为双堆栈或 IPv6 时，不支持客户端 IP 保留。
  - 以下实例类型不支持客户端 IP 保留：C1、CC1、CG2、G1、HI1、M1、M2、M3 和 T1。如果您开启了客户端 IP 保留功能并尝试使用 EC2 Instance Connect 端点连接到属于上述任何实例类型的实例，则连接将会失败。
  - 当流量通过中转网关路由时，不支持客户端 IP 保留功能。
- 当您创建 EC2 Instance Connect Endpoint 时，系统将为 Amazon Identity and Access Management（IAM）中的 Amazon EC2 服务自动创建服务相关的角色。Amazon EC2 使用服务相关的角色在您的账户中预置配网络接口，这是创建 EC2 Instance Connect Endpoint 时所必需的。有关更多信息，请参阅 EC2 Instance Connect Endpoint 的服务相关角色。
- 每个 VPC 和每个子网只能创建一个 EC2 Instance Connect Endpoint。有关更多信息，请参阅 EC2 Instance Connect 端点的限额。如需在同一 VPC 内的不同可用区中创建另一个 EC2 Instance Connect Endpoint，则必须先删除现有的 EC2 Instance Connect Endpoint。否则，您会收到限额错误。
- 每个 EC2 Instance Connect 端点最多可支持 20 个并发连接。
- 已建立 TCP 连接的最长持续时间为 1 小时（3,600 秒）。您可以在 IAM 策略中指定允许的最大持续时间，最长可以为 3,600 秒。有关更多信息，请参阅 使用 EC2 Instance Connect 端点连接到实例的权限。

  连接的持续时间并不是由 IAM 凭证的持续时间决定的。如果您的 IAM 凭证过期，连接将继续保持，直到达到指定的最长持续时间。当您使用 EC2 Instance Connect Endpoint 控制台体验连接到实例时，将最大隧道持续时间（秒）设置为小于 IAM 凭证持续时间的值。如果您的 IAM 凭证提前过期，请关闭浏览器页面以终止与实例的连接。

授予使用 EC2 Instance Connect 端点的权限

默认情况下，IAM 实体没有创建、描述或修改 EC2 Instance Connect 端点的权限。IAM 管理员必须创建授予所需权限的 IAM 策略，才能对所需资源执行特定的操作。

有关如何创建 IAM 策略 的信息，请参阅 IAM 用户指南中的创建 IAM 策略。

以下示例策略演示了如何控制用户对 EC2 Instance Connect 端点的权限。

示例

- 创建、描述、修改和删除 EC2 Instance Connect 端点的权限
- 使用 EC2 Instance Connect 端点连接到实例的权限
- 仅从特定 IP 地址范围进行连接的权限

创建、描述、修改和删除 EC2 Instance Connect 端点的权限

要创建和修改 EC2 Instance Connect Endpoint，用户需要具有执行以下操作的权限：

- `ec2:CreateInstanceConnectEndpoint`
- `ec2:CreateNetworkInterface`
- `ec2:CreateTags`
- `ec2:ModifyInstanceConnectEndpoint`
- `iam:CreateServiceLinkedRole`

要描述和删除 EC2 Instance Connect Endpoint，用户需要具有执行以下操作的权限：

- `ec2:DescribeInstanceConnectEndpoints`
- `ec2:DeleteInstanceConnectEndpoint`

您可以创建一个策略，授予在所有子网中创建、描述、修改和删除 EC2 Instance Connect Endpoint 的权限。或者，您只能通过将子网 ARN 指定为允许的 `Resource` 或使用 `ec2:SubnetID` 条件键来限制对指定子网的操作。您还可以使用 `aws:ResourceTag` 条件键明确允许或拒绝使用特定标签创建端点。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。

示例 IAM policy

在以下示例 IAM 策略中，`Resource` 部分授予在所有子网中创建、修改和删除端点的权限，由星号 (*) 指定。`ec2:Describe*` API 操作不支持资源级权限。因此，`Resource` 元素中需要 `*` 通配符。

使用 EC2 Instance Connect 端点连接到实例的权限

`ec2-instance-connect:OpenTunnel` 操作授予建立 TCP 连接到实例的权限，以通过 EC2 Instance Connect Endpoint 进行连接。您可以指定要使用的 EC2 Instance Connect Endpoint。或者，

带有星号 (*) 的 Resource 允许用户使用任何可用的 EC2 Instance Connect Endpoint。您还可以根据是否存在作为条件键的资源标签来限制对实例的访问。

Conditions

- ec2-instance-connect:remotePort - 实例上可用于建立 TCP 连接的端口。使用此条件键时，尝试连接到策略中指定的端口以外的任何其他端口上的实例都会导致失败。
- ec2-instance-connect:privateIpAddress - 与要建立 TCP 连接的实例关联的目标私有 IP 地址。您可以指定单个 IP 地址（例如 10.0.0.1/32），也可以通过 CIDR 指定一系列 IP（例如 10.0.1.0/28）。使用此条件键时，尝试连接到具有不同私有 IP 地址或 CIDR 范围之外的实例会导致失败。
- ec2-instance-connect:maxTunnelDuration - 已建立的 TCP 连接的最长持续时间。单位是秒，持续时间范围从最小 1 秒到最大 3600 秒（1 小时）。如果未指定条件，则将默认持续时间设置为 3600 秒（1 小时）。尝试连接到实例的时间超过 IAM policy 中指定的持续时间或超过默认最大值会导致失败。在指定的持续时间过后，连接将断开。

  如果在 IAM policy 中指定 maxTunnelDuration 且指定的值小于 3600 秒（默认值），则在连接到实例时必须在命令中指定 --max-tunnel-duration。有关如何连接到实例的信息，请参阅 使用 EC2 Instance Connect 端点连接到 Amazon EC2 实例。

还可以根据 EC2 Instance Connect 端点上是否存在资源标签来授予用户建立实例连接的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。

对于 Linux 实例，ec2-instance-connect:SendSSHPublicKey 操作授予将公有密钥推送到实例的权限。ec2:osuser 条件指定可以将公有密钥推送到实例的 OS（操作系统）用户的名称。使用用于启动实例的 AMI 的默认用户名。有关更多信息，请参阅 为 EC2 Instance Connect 授予 IAM 权限。

示例 IAM 策略

以下示例 IAM 策略允许 IAM 主体仅使用指定的 EC2 Instance Connect Endpoint（由指定的端点 ID eice-123456789abcdef 标识）连接到实例。只有满足所有条件，才能成功建立连接。

> **Note**
>
> ec2:Describe* API 操作不支持资源级权限。因此，Resource 元素中需要 * 通配符。

Linux

此示例评估与实例的连接是否在端口 22（SSH）上建立，实例的私有 IP 地址是否在 10.0.1.0/31 范围内（10.0.1.0 与 10.0.1.1 之间），并且 maxTunnelDuration 小于或等于 3600 秒。3600 秒（1 小时）后连接断开。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
```

```
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Windows

此示例评估与实例的连接是否在端口 3389（RDP）上建立，实例的私有 IP 地址是否在 10.0.1.0/31 范围内（10.0.1.0 与 10.0.1.1 之间），并且 maxTunnelDuration 小于或等于 3600 秒。3600 秒（1 小时）后连接断开。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "EC2InstanceConnect",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "3389"
                },
                "IpAddress": {
                    "ec2-instance-connect:privateIpAddress": "10.0.1.0/31"
                },
                "NumericLessThanEquals": {
                    "ec2-instance-connect:maxTunnelDuration": "3600"
                }
            }
        },
        {
            "Sid": "Describe",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
```

```
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

仅从特定 IP 地址范围进行连接的权限

以下示例 IAM policy 允许 IAM 主体连接到实例，条件是它们从策略中指定的 IP 地址范围内的 IP 地址进行连接。如果 IAM 主体从不在 192.0.2.0/24（本策略中的示例 IP 地址范围）内的 IP 地址调用 OpenTunnel，则响应为 Access Denied。有关更多信息，请参阅《IAM 用户指南》中的 aws:SourceIp。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "ec2-instance-connect:OpenTunnel",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance-connect-
endpoint/eice-123456789abcdef",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NumericEquals": {
                    "ec2-instance-connect:remotePort": "22"
                }
            }
        },
        {
            "Sid": "SSHPublicKey",
            "Effect": "Allow",
            "Action": "ec2-instance-connect:SendSSHPublicKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:osuser": "ami-username"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceConnectEndpoints"
            ],
            "Resource": "*"
        }
    ]
}
```

EC2 Instance Connect Endpoint 安全组

安全组控制允许到达和离开与其关联资源的流量。例如，除非与 Amazon EC2 实例关联的安全组特别允许，否则我们会拒绝进出该实例的流量。

以下示例演示了如何为 EC2 Instance Connect 端点和目标实例配置安全组规则。

示例

- EC2 Instance Connect 端点安全组规则
- 目标实例安全组规则

EC2 Instance Connect 端点安全组规则

EC2 Instance Connect 端点的安全组规则必须允许离开该端点并指向目标实例的出站流量。您可以将实例安全组或 VPC 的 IPv4 或 IPv6 地址范围指定为目标。

由于指向端点的流量来自 EC2 Instance Connect 端点服务，因此无论端点安全组的入站规则如何，该流量都会被允许。要控制谁可以使用 EC2 Instance Connect 端点连接到实例，请使用 IAM 策略。有关更多信息，请参阅 使用 EC2 Instance Connect 端点连接到实例的权限。

示例出站规则：安全组引用

以下示例使用了安全组引用，这意味着目标是一个与目标实例关联的安全组。此规则允许从该端点到使用此安全组的所有实例的出站流量。

协议	目标	端口范围	Comment
TCP	*###### ID*	22	允许指向与实例安全组关联的所有实例的出站 SSH 流量

示例出站规则：IPv4 地址范围

以下示例允许指向指定 IPv4 地址范围的出站流量。实例的 IPv4 地址是从其子网分配的，因此可以使用 VPC 的 IPv4 地址范围。

协议	目标	端口范围	Comment
TCP	*VPC IPv4 CIDR*	22	允许指向 VPC 的出站 SSH 流量

示例出站规则：IPv6 地址范围

以下示例允许指向指定 IPv6 地址范围的出站流量。实例的 IPv6 地址是从其子网分配的，因此可以使用 VPC 的 IPv6 地址范围。

协议	目标	端口范围	Comment
TCP	*VPC IPv6 CIDR*	22	允许指向 VPC 的出站 SSH 流量

目标实例安全组规则

目标实例安全组规则必须允许来自该 EC2 Instance Connect 端点的入站流量。您可以将端点安全组或 IPv4 或 IPv6 地址范围指定为源。如果指定 IPv4 地址范围，则源取决于关闭还是开启了客户端 IP 保留功能。有关更多信息，请参阅 注意事项。

由于安全组是有状态的，因此无论实例安全组的出站规则如何，都将允许响应流量离开 VPC。

示例入站规则：安全组引用

以下示例使用了安全组引用，这意味着源是与该端点关联的安全组。此规则允许从端点到使用此安全组的所有实例的入站 SSH 流量，无论客户端 IP 保留功能处于开启还是关闭状态。如果 SSH 没有其他入站安全组规则，则实例仅接受来自该端点的 SSH 流量。

协议	来源	端口范围	Comment
TCP	*###### ID*	22	允许来自与端点安全组关联的资源的入站 SSH 流量

示例入站规则：客户端 IP 保留功能已关闭

以下示例允许来自指定 IPv4 地址范围的入站 SSH 流量。因为客户端 IP 保留功能已关闭，因此源 IPv4 地址是端点网络接口的地址。端点网络接口的地址由其子网分配，因此您可以使用 VPC 的 IPv4 地址范围来允许连接到 VPC 中的所有实例。

协议	来源	端口范围	Comment
TCP	*VPC IPv4 CIDR*	22	允许来自 VPC 的入站 SSH 流量

示例入站规则：客户端 IP 保留功能已开启

以下示例允许来自指定 IPv4 地址范围的入站 SSH 流量。因为客户端 IP 保留功能已开启，因此源 IPv4 地址是客户端的地址。

协议	来源	端口范围	Comment
TCP	*## IPv4 ####*	22	允许来自指定客户端 IPv4 地址范围的入站流量

创建 EC2 Instance Connect Endpoint

您可以创建 EC2 Instance Connect 端点来确保安全连接到实例。

注意事项

- 共享子网：您可以在与您共享的子网中创建 EC2 Instance Connect 端点。但您无法使用 VPC 所有者在与您共享的子网中创建的 EC2 Instance Connect 端点。
- IP 地址类型：EC2 Instance Connect 端点支持以下地址类型，这些地址类型必须与您的子网兼容：
  - `ipv4`：仅连接到具有私有 IPv4 地址的 EC2 实例。
  - `dualstack`：连接到具有私有 IPv4 地址或 IPv6 地址的 EC2 实例。
  - `ipv6`：仅连接到具有 IPv6 地址的 EC2 实例。

先决条件

您必须拥有创建 EC2 Instance Connect Endpoint 所需的 IAM 权限。有关更多信息，请参阅 创建、描述、修改和删除 EC2 Instance Connect 端点的权限。

Console

创建 EC2 实例连接端点

1. 通过 https://console.aws.amazon.com/vpc/ 打开 Amazon VPC 控制台。
2. 在左侧导航窗格中，选择端点。
3. 选择创建端点，然后按如下所示指定端点设置：

   a. （可选）对于名称标签，输入端点的名称。

   b. 对于类型，选择 EC2 Instance Connect Endpoint。

   c. 在网络设置下，对于 VPC，选择具有目标实例的 VPC。

   d. （可选）要保留客户端 IP 地址，请展开其他设置，然后选中保留客户端 IP 复选框。否则会默认将端点网络接口用作客户端 IP 地址。

   > **Note**
   >
   > 仅当端点的 IP 地址类型配置为 IPv4 时，此选项才可用。

   e. （可选）对于安全组，请选择要与端点关联的安全组。否则会默认使用 VPC 的默认安全组。有关更多信息，请参阅 EC2 Instance Connect Endpoint 安全组。

   f. 对于子网，选择要在其中创建端点的子网。

   g. 对于 IP 地址类型，选择端点的 IP 地址类型。如果需要同时支持实例的 IPv4 和 IPv6 连接，则请选择双堆栈。如果需要支持客户端 IP 保留，则请选择 IPv4。

h. （可选）若要添加标签，请选择添加新标签，然后输入该标签的键和值。

4. 检查您的设置，然后选择创建端点。

   端点的初始状态为待处理。您必须等待端点状态变为可用后，才能使用此端点连接到实例。这可能需要几分钟的时间。

5. 要使用端点连接到实例，请参阅 连接到实例。

Amazon CLI

创建 EC2 实例连接端点

使用 create-instance-connect-endpoint 命令。

```
aws ec2 create-instance-connect-endpoint \
    --subnet-id subnet-0123456789example
```

要指定端点支持的流量类型，请包含 `--ip-address-type` 参数。有效值为 `ipv4`、`dualstack` 或 `ipv6`。子网必须支持您指定的 IP 地址类型。省略 `--ip-address-type` 参数时，默认值由子网支持的 IP 地址类型决定。

```
aws ec2 create-instance-connect-endpoint \
    --subnet-id subnet-0123456789example \
    --ip-address-type ipv4
```

下面是示例输出。

```
{
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "create-complete",
        "StateMessage": "",
        "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
        "NetworkInterfaceIds": [
            "eni-0123abcd"
        ],
```

```
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1a",
        "AvailabilityZoneId": "use1-az4",
        "CreatedAt": "2023-04-07T15:43:53.000Z",
        "SubnetId": "subnet-0123abcd",
        "PreserveClientIp": false,
        "SecurityGroupIds": [
            "sg-0123abcd"
        ],
        "Tags": [],
        "IpAddressType": "ipv4"
}
```

监控创建状态

State 字段的初始值为 create-in-progress。在使用此端点连接到实例之前，请等待状态变为 create-complete。使用 describe-instance-connect-endpoints 命令监控 EC2 Instance Connect 端点的状态。--query 参数会将结果筛选到 State 字段。

```
aws ec2 describe-instance-connect-endpoints --instance-connect-endpoint-
ids eice-0123456789example --query InstanceConnectEndpoints[*].State --output text
```

下面是示例输出。

```
create-complete
```

PowerShell

创建 EC2 Instance Connect Endpoint

使用 New-EC2InstanceConnectEndpoint cmdlet。

```
New-EC2InstanceConnectEndpoint -SubnetId subnet-0123456789example
```

要指定端点支持的流量类型，请包含 -IpAddressType 参数。有效值为 ipv4、dualstack 或 ipv6。子网必须支持您指定的 IP 地址类型。省略 -IpAddressType 参数时，默认值由子网支持的 IP 地址类型决定。

```
New-EC2InstanceConnectEndpoint -SubnetId subnet-0123456789example -
IpAddressType ipv4
```

下面是示例输出。

```
OwnerId                     : 111111111111
InstanceConnectEndpointId   : eice-0123456789example
InstanceConnectEndpointArn  : arn:aws:ec2:us-east-1:111111111111:instance-connect-
endpoint/eice-0123456789example
State                       : create-complete
StateMessage                :
DnsName                     : eice-0123456789example.0123abcd.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com
FipsDnsName                 : eice-0123456789example.0123abcd.fips.ec2-instance-
connect-endpoint.us-east-1.amazonaws.com
NetworkInterfaceIds         : {eni-0123abcd}
VpcId                       : vpc-0123abcd
AvailabilityZone            : us-east-1a
AvailabilityZoneId          : use1-az4
CreatedAt                   : 4/7/2023 3:43:53 PM
SubnetId                    : subnet-0123abcd
PreserveClientIp            : False
SecurityGroupIds            : {sg-0123abcd}
Tags                        : {}
IpAddressType               : ipv4
```

监控创建状态

State 字段的初始值为 create-in-progress。在使用此端点连接到实例之前，请等待状态变为 create-complete。使用 Get-EC2InstanceConnectEndpoint cmdlet 监控 EC2 Instance Connect Endpoint 的状态。.State.Value 将结果筛选到 State 字段。

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

下面是示例输出。

```
create-complete
```

修改 EC2 Instance Connect Endpoint

可使用 Amazon CLI 或 SDK 修改现有 EC2 Instance Connect Endpoint。Amazon EC2 控制台不支持修改端点。

在开始之前，您必须具备所需的 IAM 权限。有关更多信息，请参阅 创建、描述、修改和删除 EC2 Instance Connect 端点的权限。

可修改的参数

可修改以下 EC2 Instance Connect Endpoint 参数：

安全组

可以为 EC2 Instance Connect Endpoint 指定新的安全组。新的安全组将取代当前的安全组。

修改安全组时，必须指定：

- 至少一个安全组，即使只是 VPC 中的默认安全组。
- 安全组的 ID，而不是名称。

IP 地址类型

可以为 EC2 Instance Connect Endpoint 指定新的 IP 地址。

有效值：`ipv4` | `dualstack` | `ipv6`

保留客户端 IP 设置

可指定是否保留客户端 IP 地址作为源。

> **Note**
>
> 只有 IPv4 EC2 Instance Connect Endpoint 支持保留客户端 IP。启用 `PreserveClientIp` 后，端点的现有 IP 地址类型必须为 `ipv4`，或者如果修改相同请求中的 IP 地址类型，则新值必须为 `ipv4`。

Amazon CLI

修改 EC2 Instance Connect Endpoint

使用 modify-instance-connect-endpoint 命令并指定要修改的 EC2 Instance Connect Endpoint 和参数。修改单个请求中的所有参数见以下示例。

```
aws ec2 modify-instance-connect-endpoint \
    --instance-connect-endpoint-id eice-0123456789example \
    --security-group-ids sg-0123456789example \
```

```
    --ip-address-type dualstack \
    --no-preserve-client-ip
```

下面是示例输出。

```
{
    "Return": true
}
```

监视更新状态

在修改期间，EC2 Instance Connect Endpoint 状态更改为 update-in-progress。更新过程异步运行，完成时状态为 update-complete 或 update-failed。在状态更改为 update-complete 之前，端点将使用其旧配置。

使用 describe-instance-connect-endpoints 命令监视更新状态。--query 参数会将结果筛选到 State 字段。

```
aws ec2 describe-instance-connect-endpoints \
    --instance-connect-endpoint-ids eice-0123456789example \
    --query InstanceConnectEndpoints[*].State --output text
```

下面是示例输出。

```
update-complete
```

PowerShell

修改 EC2 Instance Connect Endpoint

使用 Edit-EC2InstanceConnectEndpoint cmdlet 并指定要修改的 EC2 Instance Connect Endpoint 和参数。修改单个请求中的所有参数见以下示例。

```
Edit-EC2InstanceConnectEndpoint `
    -InstanceConnectEndpointId eice-0123456789example `
    -SecurityGroupIds sg-0123456789example `
    -IpAddressType dualstack `
    -PreserveClientIp $false
```

下面是示例输出。

```
True
```

监视更新状态

在修改期间，EC2 Instance Connect Endpoint 状态更改为 update-in-progress。更新过程异步运行，完成时状态为 update-complete 或 update-failed。在状态更改为 update-complete 之前，端点将使用其旧配置。

使用 Get-EC2InstanceConnectEndpoint 命令监视更新状态。.State.Value 将结果筛选到 State 字段。

```
(Get-EC2InstanceConnectEndpoint -InstanceConnectEndpointId
 "eice-0123456789example").State.Value
```

下面是示例输出。

```
update-complete
```

删除 EC2 Instance Connect 端点

用完 EC2 Instance Connect 端点后，您可以将其删除。

您必须拥有创建 EC2 Instance Connect Endpoint 所需的 IAM 权限。有关更多信息，请参阅 创建、描述、修改和删除 EC2 Instance Connect 端点的权限。

使用控制台删除 EC2 Instance Connect 端点时，该端点首先会进入正在删除状态。如果删除成功，已删除的端点不会再出现。如果删除失败，则状态为 delete-failed，同时状态消息会提供失败原因。

使用 Amazon CLI 删除 EC2 Instance Connect 端点时，该端点会进入 delete-in-progress 状态。如果删除成功，则会进入 delete-complete 状态。如果删除失败，则状态为 delete-failed，同时 StateMessage 会提供失败原因。

Console

要删除 EC2 Instance Connect Endpoint

1. 通过 https://console.aws.amazon.com/vpc/ 打开 Amazon VPC 控制台。
2. 在左侧导航窗格中，选择端点。

3. 选择端点。
4. 选择操作、删除 VPC 端点。
5. 当系统提示进行确认时，输入 **delete**。
6. 选择删除。

Amazon CLI

要删除 EC2 Instance Connect Endpoint

使用 delete-instance-connect-endpoint 命令并指定要删除的 EC2 Instance Connect Endpoint 的 ID。

```
aws ec2 delete-instance-connect-endpoint --instance-connect-endpoint-
id eice-03f5e49b83924bbc7
```

下面是示例输出。

```
{
    "InstanceConnectEndpoint": {
        "OwnerId": "111111111111",
        "InstanceConnectEndpointId": "eice-0123456789example",
        "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example",
        "State": "delete-in-progress",
        "StateMessage": "",
        "NetworkInterfaceIds": [],
        "VpcId": "vpc-0123abcd",
        "AvailabilityZone": "us-east-1d",
        "AvailabilityZoneId": "use1-az2",
        "CreatedAt": "2023-02-07T12:05:37+00:00",
        "SubnetId": "subnet-0123abcd"
    }
}
```

PowerShell

删除 EC2 实例连接端点

使用 Remove-EC2InstanceConnectEndpoint cmdlet 并指定要删除的 EC2 Instance Connect Endpoint 的 ID。

```
Remove-EC2InstanceConnectEndpoint -InstanceConnectEndpointId eice-03f5e49b83924bbc7
```

下面是示例输出。

```
@{
    InstanceConnectEndpoint = @{
        OwnerId = "111111111111"
        InstanceConnectEndpointId = "eice-0123456789example"
        InstanceConnectEndpointArn = "arn:aws:ec2:us-east-1:111111111111:instance-
connect-endpoint/eice-0123456789example"
        State = "delete-in-progress"
        StateMessage = ""
        NetworkInterfaceIds = @()
        VpcId = "vpc-0123abcd"
        AvailabilityZone = "us-east-1d"
        AvailabilityZoneId = "use1-az2"
        CreatedAt = "2023-02-07T12:05:37+00:00"
        SubnetId = "subnet-0123abcd"
    }
}
```

使用 EC2 Instance Connect 端点连接到 Amazon EC2 实例

您可以使用 EC2 Instance Connect 端点连接到支持 SSH 或 RDP 的 Amazon EC2 实例。

先决条件

- 您必须拥有连接到 EC2 Instance Connect Endpoint 所需的 IAM 权限。有关更多信息，请参阅 使用 EC2 Instance Connect 端点连接到实例的权限。
- EC2 Instance Connect 端点必须处于以下状态之一：
  - 对于新端点为 create-complete
  - 对于进行修改的现有端点为 update-in-progress、update-complete 或 update-failed。修改端点时，该端点继续使用其原来的配置，直到状态更改为 update-complete。

    如果 VPC 没有 EC2 Instance Connect 端点，您可以创建一个。有关更多信息，请参阅 创建 EC2 Instance Connect Endpoint。
- EC2 Instance Connect 端点 IP 地址类型必须与实例的 IP 地址类型兼容。如果您的端点 IP 地址类型为双堆栈，则其可以同时适用于 IPv4 和 IPv6 地址。

- （Linux 实例）要使用 Amazon EC2 控制台连接到您的实例，或者要使用 CLI 进行连接并让 EC2 Instance Connect 处理临时密钥，您的实例必须安装 EC2 Instance Connect。有关更多信息，请参阅 安装 EC2 Instance Connect。
- 确保实例安全组允许来自 EC2 Instance Connect 端点的入站 SSH 流量。有关更多信息，请参阅 目标实例安全组规则。

连接选项

- 使用 Amazon EC2 控制台连接到您的 Linux 实例
- 使用 SSH 连接到 Linux 实例
- 使用 Amazon CLI 通过其实例 ID 连接到 Linux 实例
- 使用 RDP 连接到 Windows 实例
- 故障排除

使用 Amazon EC2 控制台连接到您的 Linux 实例

可以使用 Amazon EC2 控制台（基于浏览器的客户端）连接到实例，如下所示。

使用 Amazon EC2 控制台连接到实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择该实例，然后选择连接。
4. 选择 EC2 Instance Connect 选项卡。
5. 对于连接类型，选择使用私有 IP 进行连接。
6. 选择私有 IPv4 地址或 IPv6 地址。这些选项的可用性取决于分配给实例的 IP 地址。如果某个选项显示为灰色，则表示您的实例没有为其分配该类型的 IP 地址。
7. 对于 EC2 Instance Connect 端点，选择该 EC2 Instance Connect 端点的 ID。

   > **Note**
   >
   > EC2 Instance Connect 端点必须与您在上一步中选择的 IP 地址兼容。如果您的端点 IP 地址类型为双堆栈，则其可以同时适用于 IPv4 和 IPv6 地址。有关更多信息，请参阅 创建 EC2 Instance Connect Endpoint。

8. 对于用户名，如果您用于启动实例的 AMI 使用的用户名不是 `ec2-user`，请输入正确的用户名。

9. 对于最大隧道持续时间（秒），输入 SSH 连接允许的最大持续时间。

   该持续时间必须符合 IAM 策略中指定的任何 maxTunnelDuration 条件。如果您无权访问该 IAM 策略，请联系您的管理员。

10. 选择连接。这将为您的实例打开一个终端窗口。

使用 SSH 连接到 Linux 实例

您可以使用 SSH 连接到您的 Linux 实例，然后使用 open-tunnel 命令建立私有隧道。可在单连接或多连接模式下使用 open-tunnel。您可以指定实例 ID、私有 IPv4 地址或 IPv6 地址。

有关如何使用 Amazon CLI 通过 SSH 连接到您的 Linux 实例，请参阅 使用 Amazon CLI进行连接。

下面的示例使用了 OpenSSH。您可以使用任何其他支持代理模式的 SSH 客户端。

单一 连接

仅允许使用 SSH 和 **open-tunnel** 命令建立到实例的单一连接

按照如下所示使用 ssh 和 open-tunnel Amazon CLI 命令。-o 代理命令包含创建通往实例的私有隧道的 open-tunnel 命令。

```
ssh -i my-key-pair.pem ec2-user@i-1234567890abcdef0 \
    -o ProxyCommand='aws ec2-instance-connect open-tunnel --instance-
id i-1234567890abcdef0'
```

对于：

- -i - 指定用于启动实例的密钥对。
- *ec2-user*@*i-1234567890abcdef0* - 指定用于启动实例的 AMI 的用户名和实例 ID。对于具有 IPv6 地址的实例，您必须指定 IPv6 地址而不是实例 ID。
- --instance-id - 指定要连接到的实例的 ID。或者，指定 %h，它从用户那里提取实例 ID。对于具有 IPv6 地址的实例，请将 --instance-id *i-1234567890abcdef0* 替换为 --private-ip-address *2001:db8::1234:5678:1.2.3.4*。

多连接

要允许与一个实例建立多连接，请先运行 open-tunnel Amazon CLI 命令开始侦听新的 TCP 连接，然后使用 ssh 创建到实例的新 TCP 连接和私有隧道。

要使用 SSH 和 **open-tunnel** 命令允许建立到实例的多连接

1. 运行以下命令开始侦听本地机器上指定端口上的新 TCP 连接。

```
aws ec2-instance-connect open-tunnel \
    --instance-id i-1234567890abcdef0 \
    --local-port 8888
```

预期输出：

```
Listening for connections on port 8888.
```

2. 在新的终端窗口中，运行以下 ssh 命令以创建到实例的新 TCP 连接和私有隧道。

```
ssh -i my-key-pair.pem ec2-user@localhost -p 8888
```

预期输出 - 在第一个终端窗口中，您将看到以下内容：

```
[1] Accepted new tcp connection, opening websocket tunnel.
```

您可能还会看到以下内容：

```
[1] Closing tcp connection.
```

使用 Amazon CLI 通过其实例 ID 连接到 Linux 实例

如果仅知道实例 ID，则可以使用 ec2-instance-connect ssh Amazon CLI 命令通过 SSH 客户端连接到实例。有关更多信息，请参阅 使用 Amazon CLI进行连接。

先决条件

- 安装 Amazon CLI 版本 2 并使用您的凭证对其进行配置。有关更多信息，请参阅《Amazon Command Line Interface 用户指南》中的 Install or update to the latest version of the Amazon CLI 和 Configure the Amazon CLI。
- 或者，在其预先经过身份验证的 shell 中打开 Amazon CloudShell 并运行 Amazon CLI 命令。

要使用实例 ID 和 EC2 Instance Connect Endpoint 连接到实例

如果仅知道实例 ID，请使用 ec2-instance-connect ssh CLI 命令，并指定 ssh 命令、实例 ID 和 --connection-type 参数及 eice 值，以使用 EC2 Instance Connect 端点。如果实例只有 IPv6 地址，您还必须包含 --instance-ip 参数和 IPv6 地址。

- 如果实例有私有 IPv4 地址（也可以有 IPv6 地址），则请使用以下命令和参数：

```
aws ec2-instance-connect ssh \
    --instance-id i-1234567890example \
    --os-user ec2-user \
    --connection-type eice
```

- 如果实例只有 IPv6 地址，则请包含 --instance-ip 参数和 IPv6 地址：

```
aws ec2-instance-connect ssh \
    --instance-id i-1234567890example \
    --instance-ip 2001:db8::1234:5678:1.2.3.4 \
    --os-user ec2-user \
    --connection-type eice
```

> **Tip**
>
> 如果出现错误，则请确保使用的是 Amazon CLI 版本 2。ssh 参数仅适用于 Amazon CLI 版本 2。有关更多信息，请参阅《Amazon Command Line Interface 开发人员指南》中的关于 Amazon CLI 版本 2。

使用 RDP 连接到 Windows 实例

您可以在 EC2 Instance Connect Endpoint 上使用远程桌面协议（RDP）连接到没有公有 IPv4 地址或公有 DNS 名称的 Windows 实例。

使用 RDP 客户端连接到 Windows 实例

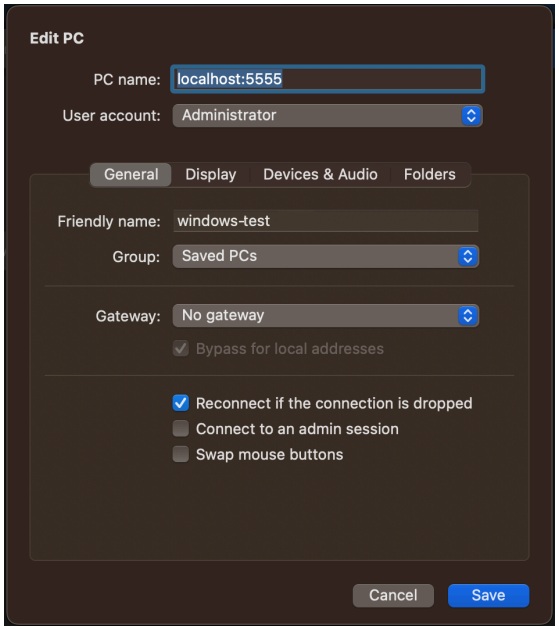
1. 完成使用 RDP 连接到 Windows 实例中的步骤 1 至 8。在步骤 8 中下载 RDP 桌面文件后，您将收到无法连接的消息，这是意料之中的，因为您的实例没有公有 IP 地址。
2. 运行以下命令，建立到实例所在 VPC 的私有隧道。--remote-port 必须是 3389，因为 RDP 默认使用端口 3389。

```
aws ec2-instance-connect open-tunnel \
```

```
    --instance-id i-1234567890abcdef0 \
    --remote-port 3389 \
    --local-port any-port
```

3. 在下载文件夹中，找到您下载的 RDP 桌面文件，然后将其拖到 RDP 客户端窗口中。
4. 右键单击该 RDP 桌面文件并选择编辑。
5. 在编辑 PC 窗口中，对于 PC 名称（要连接到的实例），输入 `localhost:local-port`，其中 *local-port* 使用的值与在步骤 2 中指定的值相同，然后选择保存。

   请注意，以下编辑 PC 窗口的屏幕截图来自 Mac 上的 Microsoft 远程桌面。如果您使用的是 Windows 客户端，窗口可能会有所不同。



6. 在 RDP 客户端中，右键单击您刚才配置的 PC，然后选择连接来连接到您的实例。
7. 根据提示，输入管理员账户的解密密码。

故障排除

使用以下信息帮助您诊断和修复您在使用 EC2 Instance Connect Endpoint 连接实例时可能遇到的问题。

无法连接到您的实例

以下是您可能无法连接到实例的常见原因：

- 安全组 – 检查分配给 EC2 Instance Connect Endpoint 和您的实例的安全组。有关所需安全组规则的更多信息，请参阅 EC2 Instance Connect Endpoint 安全组。

- 实例状态 – 验证您的实例是否处于 `running` 状态。
- 密钥对 – 如果您用于连接的命令需要私钥，请确认您的实例是否有公钥，以及您是否拥有相应的私钥。
- IAM 权限 – 验证您是否拥有所需的 IAM 权限。有关更多信息，请参阅 授予使用 EC2 Instance Connect 端点的权限。

有关 Linux 实例的更多故障排除技巧，请参阅 排查 Amazon EC2 Linux 实例的连接问题。有关 Windows 实例的故障排除技巧，请参阅 the section called “Windows 实例 RDP 问题”。

ErrorCode: AccessDeniedException

如果您收到 `AccessDeniedException` 错误，并且 IAM policy 中已指定 `maxTunnelDuration` 条件，确保当您连接到实例时指定 `--max-tunnel-duration` 参数。有关该参数的更多信息，请参阅 Amazon CLI 命令参考中的 open-tunnel。

通过 EC2 Instance Conect Endpoint 建立的日志连接

您可以使用日志记录资源操作并使用 Amazon CloudTrail 日志审核通过 EC2 Instance Connect Endpoint 建立的连接。

有关将 Amazon CloudTrail 与 Amazon EC2 结合使用的更多信息，请参阅 使用 Amazon CloudTrail 记录 Amazon EC2 API 调用。

使用 Amazon CloudTrail 记录 EC2 Instance Connect Endpoint API 调用日志

EC2 Instance Connect Endpoint 资源操作将作为管理事件记录到 CloudTrail。当进行以下 API 调用时，该活动将在事件历史记录中作为 CloudTrail 事件记录：

- `CreateInstanceConnectEndpoint`
- `DescribeInstanceConnectEndpoints`
- `DeleteInstanceConnectEndpoint`

您可以在 Amazon Web Services 账户中查看、搜索和下载最新事件。有关更多信息，请参阅 Amazon CloudTrail 用户指南中的使用 CloudTrail 事件历史记录查看事件。

使用 Amazon CloudTrail 审核使用 EC2 Instance Connect Endpoint 连接到实例的用户

通过 EC2 Instance Connect Endpoint 对实例的连接尝试将记录在 CloudTrail 的事件历史记录中。当通过 EC2 Instance Connect Endpoint 启动对实例的连接时，该连接将被记录为带有 `OpenTunnel` 的 `eventName` 的 CloudTrail 管理事件。

您可以创建将 CloudTrail 事件路由到目标的 Amazon EventBridge 规则。有关更多信息，请参阅 Amazon EventBridge 用户指南。

以下是在 CloudTrail 中记录的 `OpenTunnel` 管理事件示例。

```
{
     "eventVersion": "1.08",
     "userIdentity": {
         "type": "IAMUser",
         "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
         "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
         "accountId": "123456789012",
         "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
         "userName": "IAM-friendly-name"
     },
     "eventTime": "2023-04-11T23:50:40Z",
     "eventSource": "ec2-instance-connect.amazonaws.com",
     "eventName": "OpenTunnel",
     "awsRegion": "us-east-1",
     "sourceIPAddress": "1.2.3.4",
     "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
     "requestParameters": {
         "instanceConnectEndpointId": "eici-0123456789EXAMPLE",
         "maxTunnelDuration": "3600",
         "remotePort": "22",
         "privateIpAddress": "10.0.1.1"
     },
     "responseElements": null,
     "requestID": "98deb2c6-3b3a-437c-a680-03c4207b6650",
     "eventID": "bbba272c-8777-43ad-91f6-c4ab1c7f96fd",
     "readOnly": false,
     "resources": [{
         "accountId": "123456789012",
         "type": "AWS::EC2::InstanceConnectEndpoint",
         "ARN": "arn:aws:ec2:us-east-1:123456789012:instance-connect-endpoint/
eici-0123456789EXAMPLE"
     }],
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

EC2 Instance Connect Endpoint 的服务相关角色

Amazon EC2 使用 Amazon Identity and Access Management（IAM）服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon EC2 直接相关。服务相关角色由 Amazon EC2 预定义，并包含 Amazon EC2 代表您调用其他 Amazon Web Services 服务 所需的一切权限。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色。

EC2 Instance Connect 端点的服务相关角色

Amazon EC2 使用 AWSServiceRoleForEC2InstanceConnect 在您的账户中创建和管理 EC2 Instance Connect 端点需要的网络接口。

AWSServiceRoleForEC2InstanceConnect 服务相关角色信任以下服务来担任该角色：

- `ec2-instance-connect.amazonaws.com`

AWSServiceRoleForEC2InstanceConnect 服务相关角色使用以下托管式策略：

- Ec2InstanceConnectEndpoint

要查看托管式策略的权限，请参阅《Amazon 托管式策略参考》中的 Ec2InstanceConnectEndpoint。

创建 EC2 Instance Connect 端点的服务相关角色

您无需手动创建该服务相关角色。创建 EC2 Instance Connect 端点时，Amazon EC2 将为您创建该服务相关角色。

编辑 EC2 Instance Connect 端点的服务相关角色

EC2 Instance Connect Endpoint 不允许您编辑 AWSServiceRoleForEC2InstanceConnect 服务相关角色。

删除 EC2 Instance Connect 端点的服务相关角色

如果您不再需要使用 EC2 Instance Connect Endpoint，我们建议您删除 AWSServiceRoleForEC2InstanceConnect 服务相关角色。

必须首先删除所有 EC2 Instance Connect 端点资源后，然后才能删除该服务相关角色。

要删除服务相关角色，请参阅《IAM 用户指南》中的删除服务相关角色。

您必须配置权限，允许 IAM 实体（用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务关联角色权限。

EC2 Instance Connect 端点的限额

对于每项 Amazon 服务，您的 Amazon Web Services 账户都具有默认配额（以前称为限制）。除非另有说明，否则，每个配额是区域特定的。

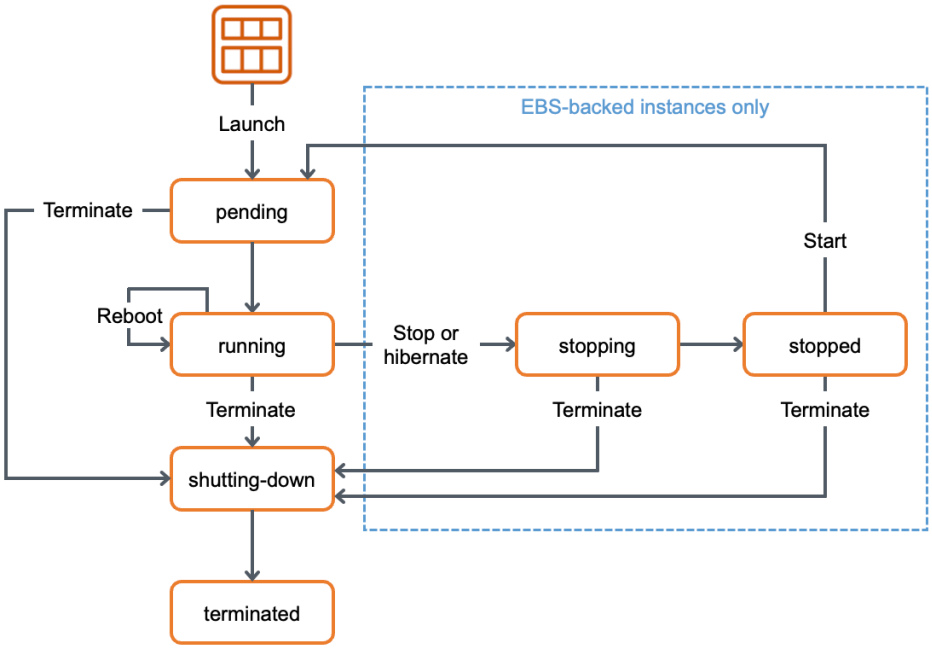
您的 Amazon Web Services 账户具有以下与 EC2 Instance Connect 端点相关的限额。

名称	默认值	可调整
在每个 Amazon Web Services 区域 中，每个 Amazon Web Services 账户 的最大 EC2 Instance Connect 端点数量	5	否
每个 VPC 的最大 EC2 Instance Connect 端点数量	1	否
每个子网的最大 EC2 Instance Connect 端点数量	1	否
每个 EC2 Instance Connect Endpoint 的最大并发连接数	20	否

Amazon EC2 实例状态更改

一个 Amazon EC2 实例从启动一直到其终止，将经过不同的状态转换。

下图显示实例状态之间的转换。



您可在实例状态更改时收到通知。有关更多信息，请参阅 the section called “状态更改事件”。

按实例状态计费

下表简要说明了每个实例状态，并说明实例使用是否已计费。一些Amazon资源（如 Amazon EBS 卷和弹性 IP 地址）无论实例的状态如何，都将产生费用。有关更多信息，请参阅 Amazon Billing 用户指南中的避免意外费用。

实例状态	描述	实例使用率计费
pending	实例正准备进入 running 状态。实例在首次启动或者在处于 stopped 状态后开机时将进入 pending 状态。	不计费
running	实例正在运行，并且做好了使用准备。	计费
stopping	实例正准备进入停止状态。	不计费

实例状态	描述	实例使用率计费
		Note 如果您使实例休眠，则在实例处于 stopping 状态时会向您收费。
stopped	实例已关闭，不能使用。可随时启动实例。	不计费
shutting-down	实例正准备终止。	不计费
terminated	实例已永久删除，无法启动。	不计费 **Note** 应用于已终止实例的预留实例将按照其付款选项进行计费，直到其期限结束。有关更多信息，请参阅 Amazon EC2 的预留实例概览

待处理实例

当您启动实例时，实例进入 pending 状态。启动时指定的实例类型将决定您的实例的主机硬件。我们使用您在启动时指定的亚马逊机器映像（AMI）来启动实例。当实例准备就绪后，其进入 running 状态。您可以连接到正在运行的实例，然后像使用您面前的计算机一样来使用它。

只要您的实例转换为 running 状态，您就需要为实例运行的每一秒，至少一分钟付费，即使实例处于闲置状态并且您并未连接到实例也是如此。

已停止的实例

如果您的实例无法通过状态检查或未按预期运行应用程序，并且实例的根卷为 Amazon EBS 卷，则您可以先停止该实例再启动，以尝试解决该问题。

当您停止实例时，它会进入 stopping 状态，然后进入 stopped 状态。当实例 stopped 时，您无需支付实例的使用费或数据传输费。任何 Amazon EBS 卷的存储均会产生费用。当实例处于 stopped 状态时，您可以修改实例的某些属性，包括实例类型。

当您启动实例时，它会进入 pending 状态，然后实例会被移动到新的主机上（尽管在某些情况下，它仍将保留在当前主机上）。如果您停止并启动实例，将丢失附加在先前主机上的的实例存储卷上的所有数据。

您的实例会保留其私有 IPv4 地址，这意味着与该私有 IPv4 地址或网络接口关联的弹性 IP 地址仍然与您的实例关联。如果您的实例具有 IPv6 地址，则它将保留 IPv6 地址。

您每次将实例从 stopped 状态转换到 running 状态时，我们都按实例运行的每秒收费，您每次启动实例时至少一分钟。

有关停止和启动实例的更多详细信息，请参阅 启动和停止 Amazon EC2 实例。

休眠实例

当您使实例休眠时，我们向操作系统发出信号来执行休眠 (suspend-to-disk)，这会将实例内存 (RAM) 中的内容保存到您的 Amazon EBS 根卷。我们保留实例的 Amazon EBS 根卷以及任何附加的 Amazon EBS 数据卷。当您启动实例时，Amazon EBS 根卷将还原到其之前的状态，并且 RAM 内容将重新加载。之前附加的数据卷会重新附加，实例也会保留其实例 ID。

当您将实例休眠时，它会进入 stopping 状态，然后进入 stopped 状态。当已休眠的实例处于 stopped 状态时，我们不会对其收取使用费，但是这些实例处于 stopping 状态时，这与您停止实例而未将其休眠时不同，我们会对其进行收费。我们不收取数据传输费，但我们会对所有 Amazon EBS 卷的存储（包括 RAM 数据的存储）收费。

当您启动休眠实例时，它会进入 pending 状态，然后我们会将其移动到新的主机上（尽管在某些情况下，它仍将保留在当前主机上）。

您的实例会保留其私有 IPv4 地址，这意味着与该私有 IPv4 地址或网络接口关联的弹性 IP 地址仍然与您的实例关联。如果您的实例具有 IPv6 地址，则它将保留其 IPv6 地址。

有关更多信息，请参阅 将您的 Amazon EC2 实例休眠。

重启实例

您可以使用 Amazon EC2 控制台、命令行工具和 Amazon EC2 API 来重启实例。我们建议您使用 Amazon EC2 来重启实例，而非在实例中运行操作系统重启命令。

重启实例相当于重启操作系统。实例位于同一主机上并保留其公有 DNS 名称、私有 IP 地址以及其实例存储卷上的所有数据。完成重启通常需要花费几分钟的时间，该时间具体取决于实例配置。

重启实例不会启动新的实例计费周期；将继续按秒计费，不再收取最低一分钟的费用。

有关更多信息，请参阅 重启 Amazon EC2 实例。

终止实例

当您决定不再需要实例时，可以终止该实例。实例的状态一旦变为 `shutting-down` 或 `terminated`，就不再产生与该实例相关的费用。

如果您启用终止保护，则无法使用控制台、CLI 或 API 终止实例。

在您终止实例之后，短时间内仍可在控制台中看见该实例，然后该条目将自动被删除。您还可以使用 CLI 和 API 来描述已终止的实例。资源 (例如标签) 会逐步与终止的实例取消关联，因此过一小段时间后，它们可能在终止的实例上不再可见。您无法连接到或恢复已终止的实例。

Amazon EBS 支持的每个实例都支持 `InstanceInitiatedShutdownBehavior` 属性，该属性决定当从实例内部启动关闭命令时（例如，在 Linux 上使用 shutdown 命令），实例是停止还是终止。默认行为是停止实例。您可以在实例运行或停止时修改此属性的设置。

每个 Amazon EBS 卷都支持 `DeleteOnTermination` 属性，该属性控制当您终止卷所连接的实例时是删除还是保留该卷。默认为删除根卷并保留所有其他 EBS 卷。

有关更多信息，请参阅 终止 Amazon EC2 实例。

实例状态之间的区别

下表总结了重启、停止、休眠与终止实例之间的主要区别。

特征	重启	停止/启动 (仅限 Amazon EBS 支持的实例)	休眠（仅限 Amazon EBS 支持的实例）	终止
主机	实例保持在同一主机上运行。	我们会将实例移动到新的主机上（尽管在某些情况下，它仍将保留在当前主机上）。	我们会将实例移动到新的主机上（尽管在某些情况下，它仍将保留在当前主机上）。	无

特征	重启	停止/启动 (仅限 Amazon EBS 支持的实例)	休眠（仅限 Amazon EBS 支持的实例）	终止
私有 IPv4 地址	实例保留其私有 IPv4 地址。	实例保留其私有 IPv4 地址。	实例保留其私有 IPv4 地址。	无
公有 IPv4 地址	实例保留其公有 IPv4 地址。	该实例将获取新的公有 IPv4 地址，除非其具有辅助网络接口或与弹性 IP 地址关联的辅助私有 IPv4 地址。	该实例将获取新的公有 IPv4 地址，除非其具有辅助网络接口或与弹性 IP 地址关联的辅助私有 IPv4 地址。	无
弹性 IP 地址 (IPv4)	弹性 IP 地址仍旧与实例相关联	弹性 IP 地址仍旧与实例相关联	弹性 IP 地址仍旧与实例相关联	弹性 IP 地址不再与实例相关联
IPv6 地址	实例保留其 IPv6 地址	实例保留其 IPv6 地址	实例保留其 IPv6 地址	无
实例存储卷	数据保留	数据将擦除	数据将擦除	数据将擦除
根卷	卷将保留	卷将保留	卷将保留	默认情况下将删除卷
RAM（内存中的内容）	RAM 将擦除	RAM 将擦除	RAM 将保存到根卷上的某一文件	RAM 将擦除

特征	重启	停止/启动 (仅限 Amazon EBS 支持的实例)	休眠（仅限 Amazon EBS 支持的实例）	终止
计费	实例计费小时不更改	实例的状态一旦变为 stopping，就不再产生与该实例相关的费用。实例每次从stopped转换为running时，我们都会启动新的实例计费周期，您每次启动实例时，最低收取一分钟费用。	当实例处于 stopping 状态时，将会产生费用；但实例处于 stopped 状态时，将会停止产生费用。实例每次从stopped转换为running时，我们都会启动新的实例计费周期，您每次启动实例时，最低收取一分钟费用。	实例的状态一旦变为 shutting-down ，就不再产生与该实例相关的费用

操作系统的关闭命令始终会终止具有实例存储根卷的实例。您可以控制操作系统关闭命令是停止还是终止具有 EBS 根卷的实例。有关更多信息，请参阅 更改实例启动的关闭行为。

启动和停止 Amazon EC2 实例

您可以停止和启动将 Amazon EBS 卷作为其根卷的实例。当您停止一个实例时，其会关闭。启动实例时，其通常会迁移到新的底层主机，并分配新的公有 IPv4 地址。

实例停止可以由用户启动（手动停止实例）或由 Amazon 启动（当 Amazon 检测到实例的底层主机发生不可修复的故障时，响应计划停止事件）。

对于由用户启动的停止，我们建议使用 Amazon EC2 控制台、CLI 或 API 而非在实例中运行操作系统停止命令。使用 Amazon EC2 时，如果实例未在几分钟内完全关闭，Amazon EC2 会执行强制关机。此外，Amazon CloudTrail 还会创建一条关于实例停止时间的 API 记录。

本主题介绍如何执行由用户启动的停止 。有关 Amazon 执行的停止的信息，请参阅 管理计划停止或停用的 Amazon EC2 实例。

当您停止某个实例时，它不会被删除。当您决定不再需要实例时，可以终止该实例。有关更多信息，请参阅 终止 Amazon EC2 实例。如果您想休眠某个实例以保存实例内存（RAM）中的内容，请参阅 将您的 Amazon EC2 实例休眠。有关实例生命周期操作之间的区别，请参阅 实例状态之间的区别。

内容

- 停止和启动 EC2 实例的工作原理
- 停止实例的方法
- 手动停止和启动实例
- 自动停止和启动您的实例
- 查找所有正在运行和已停止的实例
- 查找初始和最近的启动时间
- 为 EC2 实例启用停止保护

停止和启动 EC2 实例的工作原理

您停止 Amazon EC2 实例时，将在实例的操作系统（OS）级别注册更改，一些资源会丢失，而另一些资源会持续存在。启动实例时，将在实例级别注册更改。

主题

- 在停止实例时发生的情况
- 在启动实例时发生的情况
- 测试应用程序对停止和启动的响应
- 与实例停止和启动相关的成本

在停止实例时发生的情况

以下内容描述了使用默认停止方法停止实例时通常会发生的情况。请注意，某些方面可能会因您使用的停止方法而异。

在操作系统级别注册的更改

- API 请求会向访客发送按钮按下事件。
- 该按钮按下事件致使各种系统服务停止。来自管理程序的 ACPI 关闭按钮按下事件触发操作系统正常关闭。
- 启动 ACPI 关闭。
- 当操作系统正常关闭进程退出时，实例即会关闭。没有可配置的操作系统关闭时间。
- 如果实例操作系统未在几分钟内完全关闭，则会执行硬关闭。
- 实例将停止运行。

- 实例的状态将先更改为 `stopping`，然后更改为 `stopped`。
- [自动扩缩] 如果您的实例在自动扩缩组中，当实例状态处于除 `running` 以外的其他 Amazon EC2 状态，或者状态检查的状态变为 `impaired`，则 Amazon EC2 Auto Scaling 会认为实例运行不正常并予以替换。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的自动扩缩组中实例的运行状况检查。
- [Windows 实例] 当您停止和启动 Windows 实例时，启动代理将对该实例执行任务，例如更改所有附加的 Amazon EBS 卷的驱动器号。有关这些默认值以及如何更改它们的更多信息，请参阅 the section called “EC2Launch v2”。

资源丢失

- 存储在 RAM 中的数据。
- 实例存储卷中存储的数据。
- Amazon EC2 在启动或开始时自动分配给实例的公有 IPv4 地址。要保留永不更改的公有 IPv4 地址，您可以将弹性 IP 地址与您的实例关联。

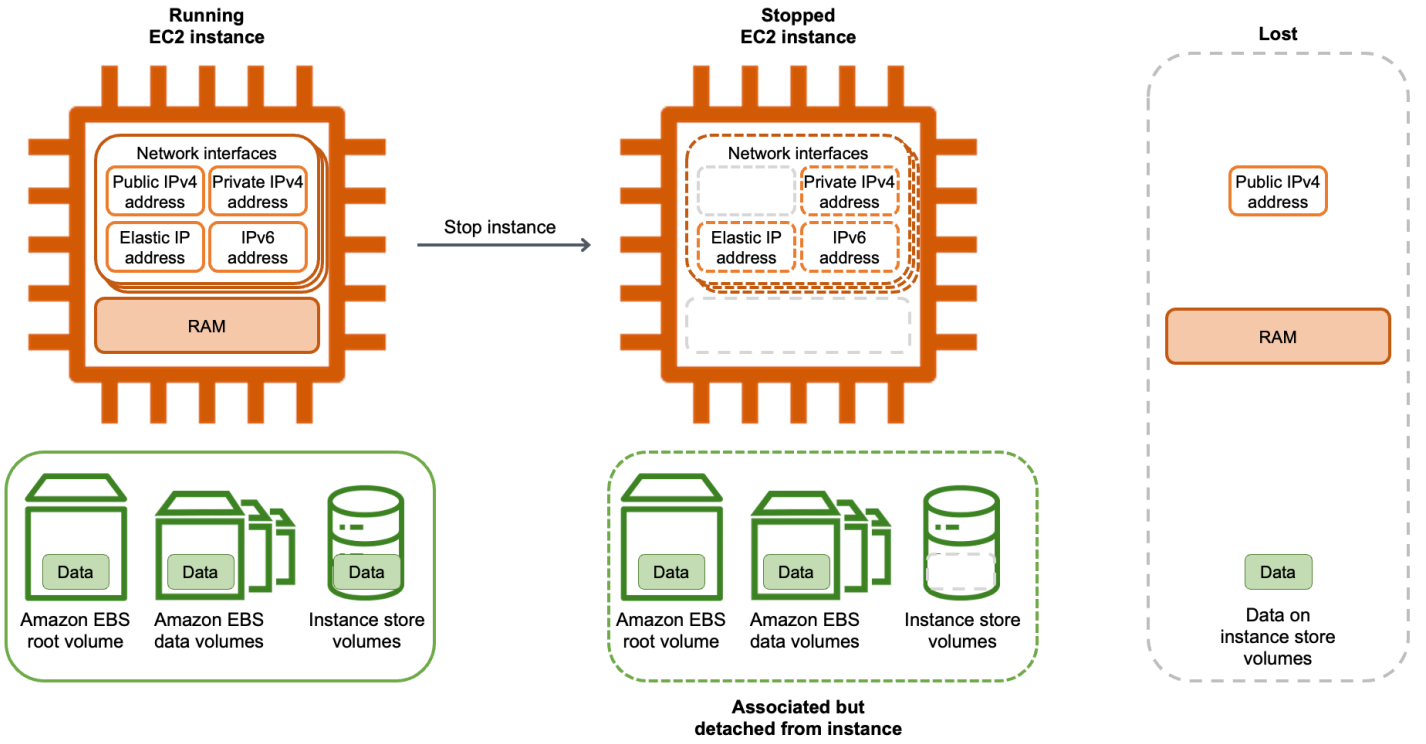
持续存在的资源

- 任何挂载的 Amazon EBS 根卷和数据卷。
- 存储在 Amazon EBS 卷中的数据。
- 任何挂载的网络接口。

  网络接口包括以下也会持久保留的资源：

  - 私有 IPv4 地址。
  - IPv6 地址。
  - 与实例关联的弹性 IP 地址。请注意，当实例停止时，我们向您收取相关弹性 IP 地址的费用。

下图说明了在 EC2 实例停止后，哪些资源会持久保留，哪些资源会丢失。该图分为三个部分：第一部分标记为正在运行的 EC2 实例，显示处于 `running` 状态的实例及其资源。第二部分标记为已停止的 EC2 实例，显示处于 `stopped` 状态且相关资源会持久保留的实例。第三部分标记为丢失，显示实例停止后会丢失的资源。



在启动实例时发生的情况

- 大多数情况下，实例会迁移到新的底层主机（尽管在某些情况下，例如在专属主机配置中将实例分配到主机时，它仍会保留在当前主机上）。
- 关联的 EBS 卷和网络接口将重新连接到实例。
- 如果该实例配置为接收公有 IPv4 地址，则 Amazon EC2 会为该实例分配一个新的公有 IPv4 地址，除非其具有辅助网络接口或与弹性 IP 地址关联的辅助私有 IPv4 地址。
- 如果您停止置放群组中的某个实例，然后重启该实例，则其仍将在该置放群组中运行。但是，如果没有足够容量可用于该实例，则启动将会失败。如果您在已有正在运行的实例的置放群组中启动实例时接收到容量错误，请在该置放群组中停止所有实例，然后再次启动它们。启动实例可能会将实例迁移至具有针对所有请求实例的容量的硬件。

测试应用程序对停止和启动的响应

您可以使用 Amazon Fault Injection Service 测试您的实例停止和启动时您的应用程序是如何响应的。有关更多信息，请参阅《Amazon Fault Injection Service 用户指南》。

与实例停止和启动相关的成本

以下费用与停止和启动实例有关。

正在停止：某个实例的状态一旦变为 shutting-down 或 terminated，该实例就不再产生费用。您无需支付已停止实例的使用费或数据传输费。存储 Amazon EBS 存储卷会产生费用。

正在启动：每次您启动已停止的实例时，我们便会收取最低一分钟的使用费用。一分钟之后，我们仅按您使用实例的秒数收费。例如，如果您运行一个实例 20 秒后停止实例，我们将按一分钟收取费用。如果您运行一个实例 3 分 40 秒，我们将收取 3 分 40 秒的使用费用。

停止实例的方法

有四种方法可以执行用户启动的停止：默认停止、通过跳过操作系统关闭而停止、强制停止以及通过跳过操作系统关闭强制停止。下表比较了各种停止方法之间的关键差异：

停止方法	关键用途	使用案例	CLI 命令
默认停止	正常关闭实例，尝试正常关闭操作系统。	典型实例停止。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0`
停止并跳过操作系统关闭	在停止实例时，可绕过操作系统正常关闭。	需要绕过正常关闭操作系统时。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --skip-os-shutdown`
强制停止	处理卡住的实例。先尝试默认停止；如果实例无法停止，则强制停止实例。	当实例卡住在stopping状态时。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --force`

停止方法	关键用途	使用案例	CLI 命令
强制停止并跳过操作系统关闭	停止实例时强制停止并绕过操作系统正常关闭。	需要强制停止并绕过正常关闭操作系统时。	`aws ec2 stop-instances \ --instance-id i-1234567890abcdef0 \ --force \ --skip-os-shutdown`

有关如何使用每种方法的说明，请参阅以下内容：

- 通过正常关闭操作系统来停止实例
- 停止实例并绕过操作系统正常关闭
- 强制停止实例

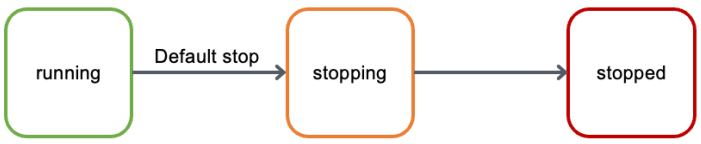
停止方法：

- 默认停止
- 停止并跳过操作系统关闭
- 强制停止
- 强制停止并跳过操作系统关闭

以下各节提供了有关四种不同的用户启动的停止方法的更多详细信息。

默认停止

默认的停止方法是停止实例的标准方法。当您发出 StopInstances 命令时，实例会从`running`状态转换到`stopping`状态，最后转换到`stopped`状态，如下图所示：



用途：正常关闭实例，尝试正常关闭操作系统。

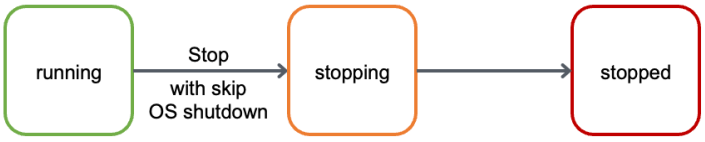
数据影响：保留 EBS 根卷和数据卷上的数据。实例存储卷上的数据丢失。

何时使用：典型停止的第一次尝试停止。

> **Note**
>
> 如果您已经尝试通过跳过操作系统关闭来停止，则在同一状态转换会话期间的后续默认停止尝试将不执行正常关闭操作系统。对于实例的当前会话，绕过操作系统的正常关闭是不可逆的。

停止并跳过操作系统关闭

当需要绕过操作系统正常关闭时，可以使用停止并跳过操作系统关闭方法来停止实例并绕过操作系统的正常关闭，如下图所示：



> **Warning**
>
> 绕过操作系统正常关闭可能会导致数据丢失或损坏（例如，内存内容未刷新到磁盘或正在运行的 IO 丢失）或跳过关闭脚本。

用途：停止实例时绕过操作系统正常关闭。

数据影响：可能导致数据丢失或损坏。内存中的内容可能无法刷新到磁盘，正在运行的 IO 可能会丢失。可能会跳过关闭脚本。

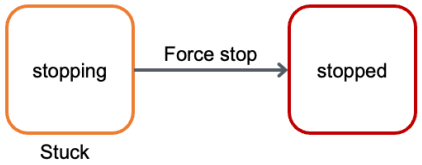
何时使用：需要绕过操作系统正常关闭时。如果在执行操作系统正常关闭的默认停止时使用，则将绕过操作系统正常关闭。

> **Note**
>
> 对于实例的当前状态转换会话，绕过操作系统的正常关闭是不可逆的。在此会话期间的后续默认停止尝试不会尝试正常关闭操作系统。

强制停止

强制停止方法用于处理在stopping状态卡住的实例。实例通常会由于底层硬件问题（由系统状态检查失败表示）而被卡住。

强制停止方法首先尝试默认停止。如果实例仍在stopping状态卡住，则force参数会强制关闭实例并将实例转换为stopped状态，如下图所示：



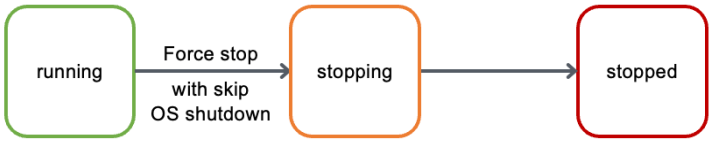
用途：处理在stopping状态卡住的实例。先尝试默认停止。如果实例无法停止，则强制关闭该实例。

数据影响：先尝试默认停止，但如果继续强制停止，则可能会导致数据丢失或损坏。在极少数情况下，会导致停止后写入 EBS 卷或其他共享资源。

何时使用：当实例在默认停止后仍卡住时进行第二次停止尝试。有关更多信息，请参阅 排查 Amazon EC2 实例的停止问题。

强制停止并跳过操作系统关闭

当需要强制停止并绕过操作系统正常关闭时，可以使用强制停止并跳过操作系统关闭方法使实例进入stopped状态，如下图所示：



用途：停止实例时强制停止并绕过操作系统正常关闭。

数据影响：跳过操作系统关闭可能导致数据丢失或损坏。内存中的内容可能无法刷新到磁盘，正在运行的 IO 可能会丢失。可能会跳过关闭脚本。如果继续强制停止，则可能会导致更多数据丢失或损坏。在极少数情况下，会导致停止后写入 EBS 卷或其他共享资源。

何时使用：当您想确保您的实例将停止并且想要绕过操作系统正常关闭时。如果在执行操作系统正常关闭的默认停止时使用，则将绕过操作系统正常关闭。

手动停止和启动实例

您可以停止并启动 Amazon EBS 支持的实例（具有 EBS 根卷的实例）。不能停止并启动具有实例存储根卷的实例。

使用默认方法停止实例时，会尝试正常关闭操作系统 (OS)。您可以绕过操作系统的正常关闭；但是，这可能会危及数据的完整性。

> **Warning**
>
> 当您停止某个实例时，任何实例存储卷上的数据都将被擦除。在停止实例之前，请确认您是否已将所需数据从实例存储卷复制到持久性存储，例如 Amazon EBS 或 Amazon S3。

[Linux 实例] 从实例中使用操作系统 halt 命令不会启动关闭。如果您使用 halt 命令，则实例不会终止；相反，它会将 CPU 放入 HLT，从而暂停 CPU 操作。实例仍在运行中。

您可以使用操作系统 shutdown 或 poweroff 命令启动关机。当您使用操作系统命令时，实例会默认停止。您可以更改此行为。有关更多信息，请参阅 更改实例启动的关闭行为。

> **Note**
>
> 如果您停止了由 Amazon EBS 支持的实例，而该实例“卡在”了 `stopping` 状态，则可以强制停止它。有关更多信息，请参阅 排查 Amazon EC2 实例的停止问题。

停止和启动方法

- 通过正常关闭操作系统来停止实例
- 停止实例并绕过操作系统正常关闭
- 启动实例

通过正常关闭操作系统来停止实例

您可以使用默认的停止方法停止实例，其中包括尝试正常关闭操作系统。有关更多信息，请参阅 默认停止。

Console

使用默认的停止方法停止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择实例，然后选择实例。

3. 依次选择实例状态、停止实例。如果此选项处于禁用状态，则表示实例已停止，或者其根卷是实例存储卷。
4. 当系统提示您确认时，选择 Stop。停止实例可能需要几分钟时间。

Amazon CLI

使用默认的停止方法停止实例

使用 stop-instances 命令。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

PowerShell

使用默认的停止方法停止实例

使用 Stop-EC2Instance cmdlet

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0
```

停止实例并绕过操作系统正常关闭

在停止实例时，您可以绕过操作系统正常关闭。有关更多信息，请参阅 停止并跳过操作系统关闭。

> **Warning**
> 绕过操作系统正常关闭可能会导致数据丢失或损坏（例如，内存内容未刷新到磁盘或正在运行的 IO 丢失）或跳过关闭脚本。

Console

停止实例并绕过操作系统正常关闭

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择实例。
3. 依次选择实例状态、停止实例。

4. 在跳过操作系统关闭下，选中跳过操作系统关闭复选框。如果您在控制台中看不到此选项，则当前区域的控制台尚未提供此选项。但是，您可以使用 Amazon CLI 或 SDK 访问此功能，也可以在控制台中尝试其他区域。
5. 选择停止。

Amazon CLI

停止实例并绕过操作系统正常关闭

结合使用 stop-instances 命令和 `--skip-os-shutdown`。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --skip-os-shutdown
```

PowerShell

停止实例并绕过操作系统正常关闭

结合使用 Stop-EC2Instance cmdlet 和 `-SkipOsShutdown $true`。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -SkipOsShutdown $true
```

启动实例

您可以启动已停止的实例。

Console

启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择实例状态、启动实例。

   实例进入 `running` 状态可能需要几分钟时间。

Amazon CLI

启动实例

使用 start-instances 命令。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

启动实例

使用 Start-EC2Instance cmdlet。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

自动停止和启动您的实例

您可以使用以下服务自动停止和启动实例：

Amazon 上的实例调度器

您可以在 Amazon 上使用实例调度器以自动开启和停止 EC2 实例。有关更多信息，请参阅如何将实例调度器与 CloudFormation 一起使用来制定 EC2 实例的计划？ 请注意，需要支付额外费用。

Amazon Lambda 和 Amazon EventBridge 规则

您可以使用 Lambda 和 EventBridge 规则按计划停止和开启您的实例。有关更多信息，请参阅如何使用 Lambda，按固定间隔停止和启动 Amazon EC2 实例？

Amazon EC2 Auto Scaling

为确保您有正确数量的 Amazon EC2 实例来处理应用程序的负载，请创建自动扩缩组。Amazon EC2 Auto Scaling 确保您的应用程序始终具有处理流量需求的适当容量，并通过仅在需要时启动实例来节省成本。请注意，Amazon EC2 Auto Scaling 会终止而不是停止不需要的实例。要设置自动扩缩组，请参阅 Amazon EC2 Auto Scaling 入门。

查找所有正在运行和已停止的实例

您可以在 Amazon EC2 全局视图中的单个页面上找到所有 Amazon Web Services 区域 中正在运行和已停止的所有实例。此功能对于清点资源和查找忘记的实例特别有用。有关如何使用全局视图的信息，请参阅 使用 Amazon 全局视图查看跨区域的资源。

您也可以在您有实例的每个区域中运行命令或 cmdlet。

Amazon CLI

获取某个区域中的 EC2 实例数量

使用以下 describe-instances 命令统计当前区域中的实例数。您必须在您有实例的每个区域运行此命令。

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "length(Reservations[].Instances[])"
```

下面是示例输出。

```
27
```

获取有关您在某个区域中的 EC2 实例的汇总信息

使用以下 describe-instances 命令。您必须在您有实例的每个区域运行此命令。

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "Reservations[].Instances[].[InstanceId,InstanceType,PrivateIpAddress]"
 \
    --output table
```

下面是示例输出。

```
-------------------------------------------------------------
|                   DescribeInstances                       |
+---------------------+---------------+-----------------+
|  i-0e3e777f4362f1bf7|  t2.micro     |  10.0.12.9      |
|  i-09453945dcf1529e9|  t2.micro     |  10.0.143.213   |
|  i-08fd74f3f1595fdbd|  m7i.4xlarge  |  10.0.1.103     |
+---------------------+---------------+-----------------+
```

PowerShell

获取某个区域中的 EC2 实例数量

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -Region us-east-2).Instances.Length
```

下面是示例输出。

```
27
```

获取有关您在某个区域中的 EC2 实例的汇总信息

使用 Get-EC2Instance cmdlet。您必须在您有实例的每个区域运行此命令。

```
(Get-EC2Instance).Instances | Select InstanceId, InstanceType, PrivateIpAddress
```

下面是示例输出。

```
InstanceId          InstanceType PrivateIpAddress
----------          ------------ ----------------
i-0e3e777f4362f1bf7 t2.micro     10.0.12.9
i-09453945dcf1529e9 t2.micro     10.0.143.213
i-08fd74f3f1595fdbd m7i.4xlarge  10.0.1.103
```

查找初始和最近的启动时间

当您描述一个实例时，该实例的启动时间是其最近的启动时间。在您停止并启动一个实例后，启动时间将反映新实例的启动时间。要查找一个实例的初始启动时间（即使在停止和启动该实例之后），请查看主网络接口连接到该实例的时间。

Console

查找最近启动时间

选择该实例，然后在详细信息选项卡上的实例详细信息下查找启动时间。

查找初始启动时间

选择该实例，然后在联网选项卡的网络接口下查找主网络接口（设备索引为 0）。

Amazon CLI

查找初始和最近启动时间

使用以下 describe-instances 命令，显示指定实例的初始启动时间和最近启动时间。

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query 'Reservations[].Instances[].
{InstanceID:InstanceId,InitialLaunch:NetworkInterfaces[0].Attachment.AttachTime,LastLaunch:L
```

下面是示例输出。

```
[
    {
        "InstanceID": "i-1234567890abcdef0",
        "InitialLaunch": "2024-04-19T00:47:08+00:00",
        "LastLaunch": "2024-05-27T06:24:06+00:00"
    }
]
```

PowerShell

查找最近启动时间

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.LaunchTime
```

下面是示例输出。

```
Monday, May 27, 2024 6:24:06 AM
```

查找初始启动时间

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -
InstanceId i-1234567890abcdef0).Instances.NetworkInterfaces.Attachment.AttachTime
```

下面是示例输出。

```
Friday, April 19, 2024 12:47:08 AM
```

为 EC2 实例启用停止保护

要防止实例意外停止，可以为实例启用停止保护。停止保护还可以保护您的实例免遭意外终止。

Amazon EC2 `ModifyInstanceAttribute` API 的 `DisableApiStop` 属性可控制是否可以使用 Amazon EC2 控制台、Amazon CLI，或 Amazon EC2 API 停止实例。您可以在实例启动、运行或已停止时设置该属性值。

注意事项

- 启用停止保护并不能防止通过使用操作系统命令（如 shutdown、poweroff）从实例启动关闭，来意外停止实例。
- 在计划事件停止实例时，启用停止保护不会阻止 Amazon 停止实例。
- 启用停止保护不会阻止 Amazon EC2 Auto Scaling 在实例运行状况不佳或在横向缩减事件期间终止实例。您可以通过使用实例横向缩减保护控制自动扩缩组在横向缩减时是否可以终止特定实例。
- 停止保护不仅可防止实例意外停止，还可以防止使用控制台、Amazon CLI 或 API 时意外终止实例。但是，它不会自动设置 `DisableApiTermination` 属性。请注意，当 `DisableApiStop` 属性设置为 `false` 时，`DisableApiTermination` 属性设置确定是否可以使用控制台、Amazon CLI 或 API 终止实例。有关更多信息，请参阅 终止 Amazon EC2 实例。
- 无法针对具有实例存储根卷的实例启用停止保护。
- 无法针对竞价型实例启用停止保护。
- 启用或禁用停止保护时，Amazon EC2 API 遵循最终一致性模型。这意味着运行设置停止保护属性的命令的结果可能不会立即对您运行的所有后续命令可见。有关更多信息，请参阅《Amazon EC2 开发人员指南》中的最终一致性。

停止保护任务

- 在实例启动时启用停止保护
- 为正在运行或已停止的实例启用停止保护
- 为正在运行或已停止的实例禁用停止保护

在实例启动时启用停止保护

您可以在启动实例时为实例启用停止保护。

Console

在实例启动时为实例启用停止保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在控制面板上，选择启动实例。
3. 在新启动实例向导中配置您的实例。
4. 在向导中，通过在高级详细信息下为停止保护选择启用来启用停止保护。

Amazon CLI

在实例启动时为实例启用停止保护

使用 run-instances 命令启动实例。添加以下参数。

```
--disable-api-stop
```

PowerShell

在实例启动时为实例启用停止保护

使用 New-EC2Instance cmdlet 启动实例。添加以下参数。

```
-DisableApiStop $true
```

为正在运行或已停止的实例启用停止保护

您可以在实例正在运行或已停止时启用停止保护。

Console

为实例启用停止保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择操作>实例设置>更改停止保护。
4. 选择启用复选框，然后选择保存。

Amazon CLI

为实例启用停止保护

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-stop
```

PowerShell

为实例启用停止保护

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $true
```

为正在运行或已停止的实例禁用停止保护

您可以使用以下方法之一为正在运行或已停止的实例禁用停止保护。

Console

为正在运行或已停止的实例禁用停止保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择 Actions（操作）、Instance Settings（实例设置）、Change Stop Protection（更改停止保护）。
4. 清除启用复选框，然后选择保存。

Amazon CLI

为正在运行或已停止的实例禁用停止保护

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
```

```
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-stop
```

PowerShell

为实例禁用停止保护

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiStop $false
```

将您的 Amazon EC2 实例休眠

当您使实例休眠时，Amazon EC2 会向操作系统发出信号来执行休眠 (suspend-to-disk)。休眠会将实例内存 (RAM) 中的内容保存到您的 Amazon Elastic Block Store (Amazon EBS) 根卷。Amazon EC2 保存实例的 EBS 根卷以及任何附加的 EBS 数据卷。当您的实例启动时：

- EBS 根卷会恢复为之前的状态
- 会重新加载 RAM 内容
- 并恢复实例上之前运行的进程
- 之前附加的数据卷会重新附加，实例也会保留其实例 ID

只有当实例已启用休眠并且满足休眠先决条件，您才可以使该实例休眠。

如果实例或应用程序在引导和进行内存占用以开始发挥全部生产功能时所需的时间较长，您可以使用休眠来预热实例。要预热实例，您需要执行以下操作：

1. 启动实例时启用休眠。
2. 将其设置为所需的状态。
3. 使实例休眠，并在需要时随时恢复到所需状态。

您无需为处于 stopped 状态的休眠实例支付实例使用费用，也无需为将 RAM 内容转移到 EBS 根卷的数据传输付费。您需要支付所有 EBS 卷的存储费用，包括 RAM 内容的存储费用。

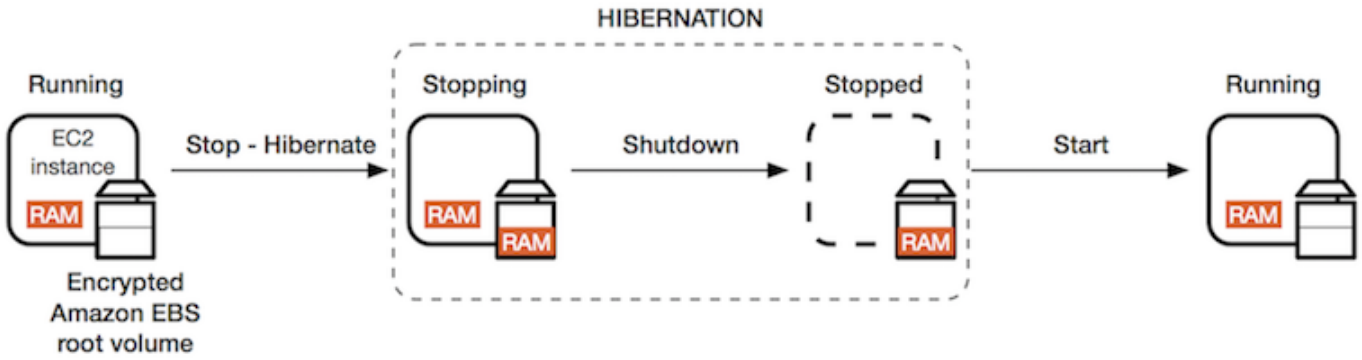
如果您不再需要某个实例，可以随时终止它，包括当实例处于 stopped（已休眠）状态时。有关更多信息，请参阅 终止 Amazon EC2 实例。

内容

- Amazon EC2 实例休眠的工作原理
- EC2 实例休眠的先决条件
- 配置 Linux AMI 以支持休眠
- 为 Amazon EC2 实例启用休眠
- 在实例上禁用 KASLR（仅限 Ubuntu）
- 将 Amazon EC2 实例休眠
- 启动已休眠的 Amazon EC2 实例
- 排查 Amazon EC2 实例休眠问题

Amazon EC2 实例休眠的工作原理

下图显示了 EC2 实例休眠过程的基本概述。



将实例休眠时发生的情况

当您使实例休眠时，将出现以下情况：

- 该实例移至 `stopping` 状态。Amazon EC2 会向操作系统发出信号来执行休眠 (suspend-to-disk)。休眠会冻结所有进程、将 RAM 中的内容保存到 EBS 根卷，然后执行常规关闭。
- 关闭完成后，实例将进入 `stopped` 状态。
- 所有 EBS 卷保持附加到实例，而且其数据将保留下来，包括已保存的 RAM 内容。
- 所有 Amazon EC2 实例存储卷都会继续保持连接到实例，但实例存储卷上的数据将会丢失。
- 大多数情况下，实例会在启动时迁移到新的底层主机。当您停止并启动实例时，也会发生此类情况。
- 当实例启动时，实例将启动，操作系统从 EBS 根卷读取 RAM 内容，然后解冻进程以恢复其状态。

- 实例会保留其私有 IPv4 地址和任何 IPv6 地址。当实例启动时，实例将继续保留其私有 IPv4 地址和所有 IPv6 地址。
- Amazon EC2 释放公有 IPv4 地址。当实例启动时，Amazon EC2 会为实例分配一个新的公有 IPv4 地址。
- 实例会保留其关联的弹性 IP 地址。您需要为与已休眠实例关联的所有弹性 IP 地址付费。

有关休眠与重启、停止和终止之间的区别，请参阅实例状态之间的区别。

限制

- 当您休眠某个实例时，任何实例存储卷上的数据都会丢失。
- （Linux 实例）您不能将具有超过 150 GiB RAM 的 Linux 实例休眠。
- （Windows 实例）您不能将具有超过 16 GiB RAM 的 Windows 实例休眠。
- 当实例处于休眠状态时，将无法对其进行修改。这与已停止的实例不同，已停止的实例并未休眠，并且您可以修改某些属性，例如实例类型或大小。
- 如果您从已休眠或已启用休眠的实例创建快照或 AMI，则可能无法连接到从该 AMI 启动的新实例，或者通过从快照创建的 AMI 启动的新实例。
- （仅限竞价型实例）如果 Amazon EC2 对您的竞价型实例进行休眠，则只有 Amazon EC2 可以恢复您的实例。如果您对竞价型实例进行休眠（用户启动的休眠），则您可以恢复您的实例。仅当容量可用且竞价价格低于或等于您指定的最高价格时，才能恢复休眠的竞价型实例。
- 您不能使位于自动扩缩组中或者由 Amazon ECS 使用的实例休眠。如果实例位于自动扩缩组中并且您尝试使该实例休眠，则 Amazon EC2 Auto Scaling 服务会将已停止的实例标记为运行状况不佳，可能会终止它并启动替换实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的自动扩缩组中实例的运行状况检查。
- 如果您休眠启动到 容量预留 的实例，容量预留 不能确保休眠的实例在尝试启动后可以恢复。
- 如已启用美国联邦信息处理标准（FIPS）模式，则无法将内核低于 5.10 的实例休眠。
- 我们支持的实例持续休眠时间不超过 60 天。要保留实例超过 60 天，您必须启动已休眠的实例，停止该实例，然后启动它。
- 我们不断通过升级和安全补丁更新平台，这可能会与现有已休眠的实例冲突。我们会通知您有关需要启动已休眠实例的关键更新，这样我们才会执行关闭或重启操作以应用必需的升级和安全补丁。

对竞价型实例进行休眠的注意事项

- 如果您对竞价型实例进行休眠，则只要容量可用且竞价价格低于或等于您指定的最高价格，就可以重新进行启动。

- 如果 Amazon EC2 对您的竞价型实例进行休眠：
  - 只有 Amazon EC2 可以恢复您的实例。
  - 当容量可用且竞价价格低于或等于您指定的最高价格时，Amazon EC2 将恢复休眠的竞价型实例。
  - 在 Amazon EC2 对您的竞价型实例进行休眠之前，您将在休眠开始前两分钟收到中断通知。

  有关更多信息，请参阅 竞价型实例中断。

EC2 实例休眠的先决条件

您可以在启动按需型实例或竞价型实例时为其启用休眠支持。您不能在现有实例（无论其正在运行或已停止）上启用休眠。有关更多信息，请参阅 启用实例休眠。

对实例进行休眠的要求

- Amazon Web Services 区域
- AMI
- 实例系列
- 实例 RAM 大小
- 根卷类型
- 根卷大小
- 根卷加密
- EBS 卷类型
- 竞价型实例请求

Amazon Web Services 区域

您可以对所有 Amazon Web Services 区域 中的实例使用休眠。

AMI

您必须使用支持休眠的 HVM AMI。下列 AMI 支持休眠：

Linux AMI

Intel 和 AMD 实例类型的 AMI

- AL2023 AMI 发布了 2023.09.20 版或更高版本[1]

- Amazon Linux 2 AMI 发布了 2019.08.29 版或更高版本。
- Amazon Linux AMI 2018.03 发布了 2018.11.16 版或更高版本。
- CentOS 版本 8 AMI[2]（需要其他配置）
- Fedora 版本 34 或更高版本 AMI[2]（需要其他配置）
- Red Hat Enterprise Linux (RHEL) 9 AMI[2]（需要其他配置）
- Red Hat Enterprise Linux (RHEL) 8 AMI[2]（需要其他配置）
- 发布序列号为 20230303 或更晚的 Ubuntu 22.04.2 LTS (Jammy Jellyfish) AMI[3]
- 发布序列号为 20210820 或更晚的 Ubuntu 20.04 LTS (Focal Fossa) AMI[3]
- 发布序列号为 20190722.1 或更晚的 Ubuntu 18.04 LTS (Bionic Beaver) AMI[3] [5]
- Ubuntu 16.04 LTS（Xenial Xerus）AMI[3] [4] [5]（需要其他配置）

Graviton 实例类型的 AMI

- AL2023 AMI（64 位 Arm）发布了 2024.07.01 版或更高版本[1]
- Amazon Linux 2 AMI（64 位 Arm）发布了 2024.06.20 版或更高版本
- 发布序列号为 20240701 或更晚的 Ubuntu 22.04.2 LTS（64 位 Arm）(Jammy Jellyfish) AMI[3]
- 发布序列号为 20240701 或更晚的 Ubuntu 20.04 LTS（64 位 Arm）(Focal Fossa) AMI[3]

[1] 对于 AL2023 最低 AMI，需要其他配置。

[2] 对于 CentOS、Fedora 和 Red Hat Enterprise Linux，仅在基于 Nitro 的实例上支持休眠。

[3] 我们建议在采用 Ubuntu 22.04.2 LTS (Jammy Jellyfish)、Ubuntu 20.04 LTS (Focal Fossa)、Ubuntu 18.04 LTS (Bionic Beaver) 和 Ubuntu 16.04 LTS (Xenial Xerus) 的实例上禁用 KASLR。有关更多信息，请参阅 在实例上禁用 KASLR（仅限 Ubuntu）。

[4] 对于 Ubuntu 16.04 LTS (Xenial Xerus) AMI，在 `t3.nano` 实例类型上不支持休眠。补丁将不可用，因为 Ubuntu（Xenial Xerus）于 2021 年 4 月终止了支持。要使用 `t3.nano` 实例类型，我们建议您升级到 Ubuntu 22.04.2 LTS (Jammy Jellyfish)、Ubuntu 20.04 LTS（Focal Fossa）AMI 或 Ubuntu 18.04 LTS（Bionic Beaver）AMI。

[5] 对 Ubuntu 18.04 LTS (Bionic Beaver) 和 Ubuntu 16.04 LTS (Xenial Xerus) 的支持已结束。

要配置您自己的 AMI 以支持休眠，请参阅 配置 Linux AMI 以支持休眠。

即将支持 Ubuntu 的其他版本和其他操作系统。

Windows AMI

- Windows Server 2022 AMI 发布了 2023.09.13 版或更高版本
- Windows Server 2019 AMI 发布了 2019.09.11 版或更高版本。
- Windows Server 2016 AMI 发布了 2019.09.11 版或更高版本。
- Windows Server 2012 R2 AMI 发布了 2019.09.11 版或更高版本。
- Windows Server 2012 AMI 发布了 2019.09.11 版或更高版本。

实例系列

您必须使用支持休眠的实例系列。但是，不支持裸机实例。

- 通用型：
  M3、M4、M5、M5a、M5ad、M5d、M6a、M6g、M6gd、M6i、M6id、M6idn、M6in、M7a、M7g、M7gd
  flex、M8a、M8azn、M8g、M8gb、M8gd、M8gn、M8i、M8i-
  flex、M8in、M8idn、M8ib、M8idb、M9g、M9gd、T2、T3、T3a、T4g
- 计算优化型：
  C3、C4、C5、C5d、C6a、C6g、C6gd、C6gn、C6i、C6id、C6in、C7a、C7g、C7gd、C7gn、C7i、C7
  flex、C8a、C8g、C8gb、C8gd、C8gn、C8i、C8i-flex、C8in、C8ib
- 内存优化型：
  R3、R4、R5、R5a、R5ad、R5d、R6a、R6g、R6gd、R6idn、R6in、R7a、R7g、R7gd、R7i、R7iz、R
  flex、R8in、R8idn、R8ib、R8idb、X2gd、X8aedz、X8i
- 存储优化型：I3、I3en、I4g、I7i、I7ie、I8g、I8ge、Im4gn、Is4gen

Console

获取支持休眠的实例类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instance Types (实例类型)。
3. 添加筛选条件按需休眠支持 = true。
4. （可选）添加筛选条件以进一步筛选相关的特定实例类型。

Amazon CLI

获取支持休眠的实例类型

请使用 describe-instance-types 命令。请注意，可用的实例类型因区域而异。

```
aws ec2 describe-instance-types \
    --filters Name=hibernation-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

获取支持休眠的实例类型

使用 Get-EC2InstanceType cmdlet。请注意，可用的实例类型因区域而异。

```
(Get-EC2InstanceType `
    -Filter @{Name="hibernation-supported"; Values="true"}).InstanceType | Sort-
Object
```

实例 RAM 大小

Linux 实例 – 必须小于 150 GiB。

Windows 实例 – 必须小于或等于 16 GiB。如需将 T3 或 T3a Windows 实例休眠，建议内存至少为 1 GiB RAM。

根卷类型

根卷必须是 EBS 卷，而不是实例存储卷。

根卷大小

根卷必须足够大，以存储 RAM 内容并满足您的预期使用量，例如，操作系统或应用程序。如果您启用休眠，则启动时在根卷上分配空间以存储 RAM。

根卷加密

必须加密根卷以确保在休眠时保护内存中的敏感内容。将 RAM 数据移动到 EBS 根卷时，它始终加密。根卷的加密在实例启动时实施。

可以使用以下三个选项之一，以确保根卷是加密的 EBS 卷：

- EBS encryption by default（EBS 原定设置加密）- 您可以启用 EBS 原定设置加密，以确保对在您的 Amazon 账户中创建的所有新 EBS 卷进行加密。这样，您就可以为实例启用休眠，而无需在实例启动时指定加密意图。有关更多信息，请参阅 Enable encryption by default。
- EBS "single-step" encryption（EBS“单步”加密）- 您可以从未加密的 AMI 中启动 EBS 支持的加密 EC2 实例，并且还可以同时启用休眠。有关更多信息，请参阅 将加密与 EBS 支持的 AMI 结合使用。
- Encrypted AMI（加密的 AMI）- 您可以使用加密的 AMI 启动实例以启用 EBS 加密。如果 AMI 没有加密的根快照，则可以将其复制到新的 AMI 并请求加密。有关更多信息，请参阅在复制过程中将未加密映像加密和复制 AMI。

EBS 卷类型

EBS 卷必须使用以下 EBS 卷类型之一：

- 通用型 SSD（`gp2` 和 `gp3`）
- 预调配 IOPS SSD（`io1` 和 `io2`）

如果选择预调配 IOPS SSD 卷类型，您必须为 EBS 卷预调配合适的 IOPS，以实现休眠状态的最佳性能。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 卷类型。

竞价型实例请求

对于竞价型实例，以下要求适用：

- 竞价型实例请求类型必须为 `persistent`。
- 您不能在竞价型实例请求中指定启动组。

配置 Linux AMI 以支持休眠

以下 Linux AMI 可以支持 Amazon EC2 实例休眠，前提是您完成了本节中所述的其他配置步骤。

需要使用额外的配置：

- AL2023 最低 AMI 发布了 2023.09.20 版或更高版本
- Amazon Linux 2 最小 AMI 发布了 2019.08.29 版或更高版本

- 在 2019.08.29 前发布的 Amazon Linux 2
- 在 2018.11.16 前发布的 Amazon Linux
- CentOS 版本 8 或更高版本
- Fedora 34 版或更高版本
- Red Hat Enterprise Linux 版本 8 或 9
- 发布序列号早于 20210820 的 Ubuntu 20.04 LTS（Focal Fossa）
- 发布序列号早于 20190722.1 的 Ubuntu 18.04（Bionic Beaver）
- Ubuntu 16.04 (Xenial Xerus)

有关支持休眠且无需额外配置的 Linux 和 Windows AMI，请参阅AMI。

有关更多信息，请参阅 Update instance software on your Amazon Linux 2 instance。

AL2023 最低 AMI 发布了 2023.09.20 版或更高版本

配置 2023.09.20 版或更高版本的 AL2023 最低 AMI 以支持休眠

1. 从存储库安装 `ec2-hibinit-agent` 程序包。

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

2. 重新启动服务。

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

Amazon Linux 2 最小 AMI 发布了 2019.08.29 版或更高版本

配置 2019.08.29 版或更高版本的 Amazon Linux 2 最小 AMI 以支持休眠

1. 从存储库安装 `ec2-hibinit-agent` 程序包。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

2. 重新启动服务。

```
[ec2-user ~]$ sudo systemctl start hibinit-agent
```

在 2019.08.29 前发布的 Amazon Linux 2

配置在 2019.08.29 前发布的 Amazon Linux 2 AMI 以支持休眠

1. 将内核更新为 4.14.138-114.102 或更高版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 从存储库安装 ec2-hibinit-agent 程序包。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

4. 确认内核版本已更新为 4.14.138-114.102 或更高版本。

```
[ec2-user ~]$ uname -a
```

5. 停止实例并创建 AMI。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。

在 2018.11.16 前发布的 Amazon Linux

配置在 2018.11.16 前发布的 Amazon Linux AMI 以支持休眠

1. 将内核更新为 4.14.77-70.59 或更高版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 从存储库安装 ec2-hibinit-agent 程序包。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

4. 确认内核版本已更新为 4.14.77-70.59 或更高版本。

```
[ec2-user ~]$ uname -a
```

5. 停止实例并创建 AMI。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。

CentOS 版本 8 或更高版本

配置 CentOS 版本 8 或更高版本 AMI 以支持休眠

1. 将内核更新为 `4.18.0-305.7.1.el8_4.x86_64` 或更高版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 安装 Extra Packages for Enterprise Linux (EPEL) 存储库。

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

3. 从存储库安装 `ec2-hibinit-agent` 程序包。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. 启用休眠代理以在引导时开启。

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

6. 确认内核版本已更新为 `4.18.0-305.7.1.el8_4.x86_64` 或更高版本。

```
[ec2-user ~]$ uname -a
```

Fedora 34 版或更高版本

配置 Fedora 版本 34 或更高版本 AMI 以支持休眠

1. 将内核更新为 `5.12.10-300.fc34.x86_64` 或更高版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 从存储库安装 `ec2-hibinit-agent` 程序包。

```
[ec2-user ~]$ sudo dnf install ec2-hibinit-agent
```

3. 启用休眠代理以在引导时开启。

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

4. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

5. 确认内核版本已更新为 5.12.10-300.fc34.x86_64 或更高版本。

```
[ec2-user ~]$ uname -a
```

Red Hat Enterprise Linux 版本 8 或 9

配置 Red Hat Enterprise Linux 8 或 9 AMI 以支持休眠

1. 将内核更新为 4.18.0-305.7.1.el8_4.x86_64 或更高版本。

```
[ec2-user ~]$ sudo yum update kernel
```

2. 安装 Extra Packages for Enterprise Linux (EPEL) 存储库。

   RHEL 版本 8：

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

   RHEL 版本 9：

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

3. 从存储库安装 ec2-hibinit-agent 程序包。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

4. 启用休眠代理以在引导时开启。

```
[ec2-user ~]$ sudo systemctl enable hibinit-agent.service
```

5. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

6. 确认内核版本已更新为 4.18.0-305.7.1.el8_4.x86_64 或更高版本。

```
[ec2-user ~]$ uname -a
```

发布序列号早于 20210820 的 Ubuntu 20.04 LTS（Focal Fossa）

配置发布序列号早于 20210820 的 Ubuntu 20.04 LTS（Focal Fossa）AMI 以支持休眠

1. 将 linux-aws-kernel 更新为 5.8.0-1038.40 或更高版本，并将 grub2 更新为 2.04-1ubuntu26.13 或更高版本。

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

3. 确认内核版本已更新为 5.8.0-1038.40 或更高版本。

```
[ec2-user ~]$ uname -a
```

4. 确认 grub2 版本已更新为 2.04-1ubuntu26.13 或更高版本。

```
[ec2-user ~]$ dpkg --list | grep grub2-common
```

发布序列号早于 20190722.1 的 Ubuntu 18.04（Bionic Beaver）

配置在序列号 20190722.1 前发布的 Ubuntu 18.04 LTS AMI 以支持休眠

1. 将内核更新为 4.15.0-1044 或更高版本。

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 从存储库安装 ec2-hibinit-agent 程序包。

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

4. 确认内核版本已更新为 4.15.0-1044 或更高版本。

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04 (Xenial Xerus)

要将 Ubuntu 16.04 LTS 配置为支持休眠，您需要安装 Linux-aws-hwe 内核软件包版本 4.15.0-1058-aws 或更高版本以及 ec2-hibinit-agent。

> **Important**
>
> linux-aws-hwe 内核程序包受规范支持。对 Ubuntu 16.04 LTS 的标准支持于 2021 年 4 月结束，该软件包不再定期更新。但是，它将接收其他安全更新，直到扩展安全性维护支持在 2024 年结束。有关更多信息，请参阅规范 Ubuntu 博客上现已提供的适用于 Ubuntu 16.04 LTS 的 Amazon EC2 休眠。
> 我们建议您升级到 Ubuntu 20.04 LTS（Focal Fossa）AMI 或 Ubuntu 18.04 LTS（Bionic Beaver）AMI。

配置 Ubuntu 16.04 LTS AMI 以支持休眠

1. 将内核更新为 4.15.0-1058-aws 或更高版本。

```
[ec2-user ~]$ sudo apt update
[ec2-user ~]$ sudo apt install linux-aws-hwe
```

2. 从存储库安装 ec2-hibinit-agent 程序包。

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

4. 确认内核版本已更新为 4.15.0-1058-aws 或更高版本。

```
[ec2-user ~]$ uname -a
```

为 Amazon EC2 实例启用休眠

要使实例休眠，您必须在启动实例时先为其启用休眠。

> **Important**
> 启动实例后，无法为实例启用或禁用休眠。

任务

- 为按需型实例启用休眠
- 为竞价型实例启用休眠
- 查看实例是否已启用休眠

为按需型实例启用休眠

您可以为按需型实例启用休眠。

Console

要为按需型实例启用休眠

1. 按照步骤启动实例，但请在完成以下步骤以启用休眠之后再启动实例。
2. 若要启用休眠，请在启动实例向导中配置以下字段：
   a. 在应用程序和操作系统映像（Amazon 机器映像）下，选择支持休眠的 AMI。有关更多信息，请参阅 AMI。

b. 在 Instance type（实例类型）下，选择一个受支持的实例类型。有关更多信息，请参阅 实例系列。

c. 在 Configure storage（配置存储）下，选择右侧的 Advanced（高级），并为根卷指定以下信息：

- 在大小 (GiB) 中，输入 EBS 根卷大小。卷必须足够大，以存储 RAM 内容并满足您的预期使用量。
- 对于 Volume Type（卷类型），选择支持的 EBS 卷类型：通用型 SSD（gp2 和 gp3）或预调配 IOPS SSD（io1 和 io2）。
- 对于 Encrypted（已加密），选择 Yes（是）。如果您在此 Amazon 区域中启用了默认加密，则系统会选择 Yes（是）。
- 对于 KMS key（KMS 密钥），请选择卷的加密密钥。如果您在此 Amazon 区域中启用了默认加密，则会选择默认加密密钥。

有关根卷先决条件的更多信息，请参阅EC2 实例休眠的先决条件。

d. 展开 Advanced details（高级详细信息），找到 Stop - Hibernate behavior（停止 – 休眠行为），然后选择 Enable（启用）。

3. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

要为按需型实例启用休眠

使用 run-instances 命令启动实例。使用 `--block-device-mappings file://mapping.json` 参数指定 EBS 根卷参数，并使用 `--hibernation-options Configured=true` 参数启用休眠。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type m5.large \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
```

在 `mapping.json` 中指定以下内容。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

DeviceName 的值必须匹配与 AMI 关联的根设备名称。要查找根设备名称，请使用 describe-images 命令。

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

如果您在此 Amazon 区域中启用了默认加密，则可以省略 "Encrypted": true。

PowerShell

要为按需型实例启用休眠

使用 New-EC2Instance 命令启动实例。通过首先定义块设备映射，然后使用 -BlockDeviceMappings 参数将其添加到命令来指定 EBS 根卷。使用 -HibernationOptions_Configured $true 参数启用休眠。

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair
```

DeviceName 的值必须匹配与 AMI 关联的根设备名称。要查找根设备名称，请使用 Get-EC2Image 命令。

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

如果您在此 Amazon 区域中启用了默认加密，则可以在块设备映射中省略 Encrypted = $true。

为竞价型实例启用休眠

您可以为竞价型实例启用休眠。有关在中断时休眠竞价型实例的更多信息，请参阅 竞价型实例中断。

Console

要为竞价型实例启用休眠

1. 请遵循使用启动实例向导请求竞价型实例的过程操作，但请在完成以下步骤启用休眠之后再启动实例。
2. 若要启用休眠，请在启动实例向导中配置以下字段：
   a. 在应用程序和操作系统映像（Amazon 机器映像）下，选择支持休眠的 AMI。有关更多信息，请参阅 AMI。
   b. 在 Instance type（实例类型）下，选择一个受支持的实例类型。有关更多信息，请参阅 实例系列。
   c. 在 Configure storage（配置存储）下，选择右侧的 Advanced（高级），并为根卷指定以下信息：
      - 在大小 (GiB) 中，输入 EBS 根卷大小。卷必须足够大，以存储 RAM 内容并满足您的预期使用量。
      - 对于 Volume Type（卷类型），选择支持的 EBS 卷类型：通用型 SSD（gp2 和 gp3）或预调配 IOPS SSD（io1 和 io2）。
      - 对于 Encrypted（已加密），选择 Yes（是）。如果您在此 Amazon 区域中启用了默认加密，则系统会选择 Yes（是）。
      - 对于 KMS key（KMS 密钥），请选择卷的加密密钥。如果您在此 Amazon 区域中启用了默认加密，则会选择默认加密密钥。

   有关根卷先决条件的更多信息，请参阅EC2 实例休眠的先决条件。

d. 展开高级详细信息，除了用于配置竞价型实例的字段外，还要执行以下操作：

i. 对于请求类型，选择持续。

ii. 对于中断行为，选择休眠。或者，对于停止 - 休眠行为，选择启用。这两个字段都可以在竞价型实例上启用休眠。您只需要配置其中一个。

3. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

要为竞价型实例启用休眠

使用 run-instances 命令请求竞价型实例。使用 `--block-device-mappings file://mapping.json` 参数指定 EBS 根卷参数，并使用 `--hibernation-options Configured=true` 参数启用休眠。竞价型实例请求类型（`SpotInstanceType`）必须为 `persistent`。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c4.xlarge \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
    --instance-market-options
        {
           "MarketType":"spot",
           "SpotOptions":{
              "MaxPrice":"1",
              "SpotInstanceType":"persistent"
            }
        }
```

按如下方式在 `mapping.json` 中指定 EBS 根卷参数。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 30,
```

```
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

DeviceName 的值必须匹配与 AMI 关联的根设备名称。要查找根设备名称，请使用 describe-images 命令。

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

如果您在此 Amazon 区域中启用了默认加密，则可以省略 "Encrypted": true。

PowerShell

要为竞价型实例启用休眠

使用 New-EC2Instance 命令请求竞价型实例。通过首先定义块设备映射，然后使用 -BlockDeviceMappings 参数将其添加到命令来指定 EBS 根卷。使用 -HibernationOptions_Configured $true 参数启用休眠。

```
$ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
$ebs_encrypt.DeviceName = "/dev/xvda"
$ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs_encrypt.Ebs.VolumeSize = 30
$ebs_encrypt.Ebs.VolumeType = "gp2"
$ebs_encrypt.Ebs.Encrypted = $true

New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType m5.large `
    -BlockDeviceMappings $ebs_encrypt `
    -HibernationOptions_Configured $true `
    -MinCount 1 `
    -MaxCount 1 `
    -KeyName MyKeyPair `
    -InstanceMarketOption @(
        MarketType = spot;
        SpotOptions @{
        MaxPrice = 1;
        SpotInstanceType = persistent}
    )
```

DeviceName 的值必须匹配与 AMI 关联的根设备名称。要查找根设备名称，请使用 Get-EC2Image 命令。

```
Get-EC2Image -ImageId ami-0abcdef1234567890
```

如果您在此 Amazon 区域中启用了默认加密，则可以在块设备映射中省略 Encrypted = $true。

查看实例是否已启用休眠

您可以检查实例是否已启用休眠。

Console

查看实例是否启用休眠

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择所需实例，然后在详细信息选项卡上的实例详细信息部分中检查停止 – 休眠操作。已启用表明已为实例启用休眠。

Amazon CLI

查看实例是否启用休眠

使用 describe-instances 命令并指定 --filters "Name=hibernation-options.configured,Values=true" 参数以筛选启用了休眠的实例。

```
aws ec2 describe-instances \
    --filters "Name=hibernation-options.configured,Values=true"
```

输出中的以下字段指示实例已启用了休眠。

```
"HibernationOptions": {
    "Configured": true
}
```

PowerShell

查看实例是否启用休眠

使用 Get-EC2Instance cmdlet 并筛选启用了休眠的实例。

```
(Get-EC2Instance `
    -Filter @{Name="hibernation-options.configured"; Values="true"}).Instances
```

在实例上禁用 KASLR（仅限 Ubuntu）

要使用 Ubuntu 16.04 LTS（Xenial Xerus）、发布序列号为 20190722.1 或更晚的 Ubuntu 18.04 LTS - Bionic 或者发布序列号为 20210820 或更晚的 Ubuntu 20.04 LTS（Bionic Beaver）在新启动的实例上运行休眠，我们建议禁用 KASLR（内核地址空间布局随机掩码）。在 Ubuntu 16.04 LTS、Ubuntu 18.04 LTS 或 Ubuntu 20.04 LTS 上，默认启用 KASLR。

KASLR 是一项标准 Linux 内核安全功能，它通过随机化内核的基本地址值，来帮助减少尚未发现的内存访问漏洞的风险和后果。启用 KASLR 后，实例在休眠后可能无法恢复。

要详细了解 KASLR，请参阅 Ubuntu 功能。

在使用 Ubuntu 启动的实例上禁用 KASLR

1. 使用 SSH 连接到您的实例。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
2. 在选定编辑器中打开 `/etc/default/grub.d/50-cloudimg-settings.cfg` 文件。编辑 `GRUB_CMDLINE_LINUX_DEFAULT` 行以将 `nokaslr` 选项追加到其末尾，如以下示例所示。

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
    nvme_core.io_timeout=4294967295 nokaslr"
   ```
3. 保存文件并退出您的编辑器。
4. 运行以下命令来重新构建 grub 配置。

   ```
   sudo update-grub
   ```
5. 重启实例。

   ```
   sudo reboot
   ```
6. 运行以下命令确认 `nokaslr` 已添加。

   ```
   cat /proc/cmdline
   ```

命令的输出应包含 nokaslr 选项。

将 Amazon EC2 实例休眠

如果按需型实例或竞价型实例是 EBS 支持的实例、已启用休眠且满足休眠先决条件，则可以在该实例上启动休眠。如果无法成功使实例休眠，则会进行正常关闭。

Console

要对实例进行休眠

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择所需实例，然后依次选择实例状态、休眠实例。如果休眠实例处于已禁用状态，则表示实例已经休眠或停止，或者无法休眠。有关更多信息，请参阅EC2 实例休眠的先决条件。
4. 当系统提示进行确认时，选择休眠。使实例休眠可能需要几分钟时间。实例状态首先会更改为 Stopping（正在停止），然后如果实例已休眠，实例状态会更改为 Stopped（已停止）。

Amazon CLI

要对实例进行休眠

使用 stop-instances 命令并指定 --hibernate 参数。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

PowerShell

要对实例进行休眠

使用 Stop-EC2Instance cmdlet。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Hibernate $true
```

您可以检查是否已在实例上启动休眠。

Console

查看是否已在实例上启动休眠

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后在详细信息选项卡的实例详细信息部分中，检查状态转换消息的值。

   Client.UserInitiatedHibernate：用户启动的休眠表示您在按需型实例或竞价型实例上启动了休眠。

Amazon CLI

查看是否已在实例上启动休眠

使用 describe-instances 命令并指定 `state-reason-code` 筛选条件以查看已启动了休眠的实例。

```
aws ec2 describe-instances \
    --filters "Name=state-reason-code,Values=Client.UserInitiatedHibernate"
```

输出中的以下字段表示已在按需型实例或竞价型实例上启动休眠。

```
"StateReason": {
    "Code": "Client.UserInitiatedHibernate"
}
```

PowerShell

查看是否已在实例上启动休眠

使用 Get-EC2Instance cmdlet 并指定 `state-reason-code` 筛选条件以查看已启动休眠的实例。

```
Get-EC2Instance `
    -Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

启动已休眠的 Amazon EC2 实例

按照启动已停止实例的相同方式，启动已休眠的实例。

对于竞价型实例，如果 Amazon EC2 对实例进行了休眠，则只有 Amazon EC2 可以恢复它。只有在您对竞价型实例进行休眠的情况下，您才能自己恢复该实例。仅当容量可用且竞价价格低于或等于您指定的最高价格时，才能恢复竞价型实例。

Console

启动已休眠实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择已休眠的实例，然后依次选择实例状态、启动实例。实例进入 `running` 状态可能需要几分钟时间。在此期间，实例状态检查显示实例处于失败状态，直至实例已启动。

Amazon CLI

启动已休眠实例

使用 start-instances 命令。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

PowerShell

启动已休眠实例

使用 Start-EC2Instance cmdlet。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

排查 Amazon EC2 实例休眠问题

使用此信息帮助您诊断和修复在使实例休眠时可能遇到的问题。

休眠问题

- 在启动后无法立即休眠
- 从 stopping 转变为 stopped 用时太长，内存状态在启动后无法恢复
- 实例卡在 stopping 状态
- 休眠后无法立即启动竞价型实例
- 恢复竞价型实例失败

在启动后无法立即休眠

如果您在实例启动之后过快地尝试使实例休眠，则会收到错误。

Linux 实例在启动之后，您必须等待大约 2 分钟才会休眠，对于 Windows 实例，在启动之后，您必须等待大约 5 分钟后才会休眠。

从 stopping 转变为 stopped 用时太长，内存状态在启动后无法恢复

如果正在进入休眠的实例从 `stopping` 状态转变为 `stopped` 状态用时过长，并且在启动之后内存状态未恢复，则这可能表明未正确配置休眠。

Linux 实例

检查实例系统日志，查找与休眠相关的消息。要访问系统日志，请连接到实例或者使用 get-console-output 命令。从 `hibinit-agent` 中查找日志行。如果日志行指示出现故障或者缺少日志行，则很有可能在启动时配置休眠失败。

例如，以下消息指明实例根卷不够大：`hibinit-agent: Insufficient disk space. Cannot create setup for hibernation. Please allocate a larger root device.`

如果 `hibinit-agent` 中的最后日志行是 `hibinit-agent: Running: swapoff /swap`，则已成功配置休眠。

如果您未看到来自这些进程的任何日志，您的 AMI 可能不支持休眠。有关支持的 AMI 的信息，请参阅EC2 实例休眠的先决条件。如果您使用自己的 Linux AMI，则请确保按照 配置 Linux AMI 以支持休眠 的说明操作。

Windows Server 2016 及更高版本

检查 EC2 Launch 日志，查找与休眠相关的消息。要访问 EC2 Launch 日志，请连接到实例并在文本编辑器中打开 `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\Ec2Launch.log` 文件。如果使用 EC2Launch v2，请打开 `C:\ProgramData\Amazon\EC2Launch\log\agent.log`。

> **Note**
>
> 默认情况下，Windows 会隐藏 C:\ProgramData 下的文件和文件夹。要查看 EC2 Launch 目录和文件，必须在 Windows 资源管理器中输入路径，或者更改文件夹属性以显示隐藏的文件和文件夹。

查找休眠的日志行。如果日志行指示出现故障或者缺少日志行，则很有可能在启动时配置休眠失败。

例如，以下消息指示休眠配置失败：Message: Failed to enable hibernation. 如果错误消息包含十进制 ASCII 值，则可以将这些 ASCII 值转换为纯文本，以便读取完整的错误消息。

如果日志行包含 HibernationEnabled: true，则已成功配置休眠。

Windows Server 2012 R2 和早期版本

检查 EC2 配置日志，查找与休眠相关的消息。要访问 EC2 配置日志，请连接到实例并在文本编辑器中打开 C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt 文件。从 SetHibernateOnSleep 中查找日志行。如果日志行指示出现故障或者缺少日志行，则很有可能在启动时配置休眠失败。

例如，以下消息指明实例根卷不够大：SetHibernateOnSleep: Failed to enable hibernation: Hibernation failed with the following error: There is not enough space on the disk.

如果日志行为 SetHibernateOnSleep: HibernationEnabled: true，则已成功配置休眠。

Windows 实例大小

如果您使用的是 RAM 小于 1 GiB 的 T3 或 T3a Windows 实例，请尝试将实例的大小增加到至少有 1 GiB RAM 的实例。

实例卡在 stopping 状态

如果您已使实例休眠并且实例卡在 stopping 状态，则可以强制停止它。有关更多信息，请参阅 排查 Amazon EC2 实例的停止问题。

休眠后无法立即启动竞价型实例

如果您尝试在竞价型实例休眠后的两分钟内启动该实例，则可能会出现以下错误：

```
You failed to start the Spot Instance because the associated Spot Instance request is not in an appropriate state to support start.
```

对于 Linux 实例，等待大约 2 分钟，对于 Windows 实例，等待大约 5 分钟，然后重试启动实例。

恢复竞价型实例失败

如果您的竞价型实例成功休眠但无法恢复，而是重新启动（未保留休眠状态的全新重启），则可能是因为用户数据包含以下脚本：

```
/usr/bin/enable-ec2-spot-hibernation
```

从启动模板的用户数据字段中移除此脚本，然后请求新的竞价型实例。

请注意，即使实例无法恢复，在不保留休眠状态的情况下，该实例仍然可以像从 stopped 状态启动一样启动。

重启 Amazon EC2 实例

实例重启相当于操作系统重启。在许多情况下，只需要几分钟时间即可重启您的实例。

重启实例后，将保留以下内容：

- 公有 DNS 名称（IPv4）
- 私有 IPv4 地址
- 公有 IPv4 地址
- IPv6 地址（如适用）
- 其实例存储卷上的所有数据

与停止并启动实例会启动新的计费周期并且最低收取一分钟的费用不同，重启实例不会启动新的实例计费周期。

实例重启可以由用户启动（即手动重启实例），也可以由 Amazon 启动（用于自动恢复实例，或者响应计划重启事件以进行必要的维护，例如应用需要重启的更新）。

对于由用户启动的重启，我们建议使用 Amazon EC2 控制台、CLI 或 API 而非在实例中运行操作系统重启命令。使用 Amazon EC2 时，如果实例未在几分钟内完全关闭，Amazon EC2 会执行强制重启。此外，Amazon CloudTrail 还会创建一条关于实例重启时间的 API 记录。

本主题介绍如何执行由用户启动的重启。有关 Amazon 执行的重启的信息，请参阅实例自动恢复和管理计划重启的 Amazon EC2 实例。

> **Important**
>
> 如果正在您的实例上安装更新，我们建议您在所有更新都安装完毕之后，才使用 Amazon EC2 控制台或命令行重启或关闭您的实例。当您使用 Amazon EC2 控制台或命令行重启或关闭实例时，您的实例会存在硬重启的风险。在安装更新过程中硬重启会将您的实例置于不稳定状态。

Console

重启实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后选择 Instance state（实例状态）、Reboot instance（重新引导实例）。
4. 当系统提示您确认时，选择重启。

   实例仍处于 `running` 状态。

Amazon CLI

重启实例

使用 reboot-instances 命令。

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef0
```

PowerShell

重启实例

使用 Restart-EC2Instance cmdlet。

```
Restart-EC2Instance -InstanceId i-1234567890abcdef0
```

运行受控的故障注入实验

您可以使用 Amazon Fault Injection Service 测试您的实例重新启动时您的应用程序如何响应。有关更多信息，请参阅《Amazon Fault Injection Service 用户指南》。

终止 Amazon EC2 实例

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

当您不再需要实例时，可将其删除。这称为终止 实例。实例的状态一旦变为 `shutting-down` 或 `terminated`，就不再产生与该实例相关的费用。

在您终止之后，您将无法连接到或启动实例。但您可以使用同一 AMI 启动新的实例。

如果您希望停止或休眠实例，请参阅 启动和停止 Amazon EC2 实例 或 将您的 Amazon EC2 实例休眠。有关更多信息，请参阅 实例状态之间的区别。

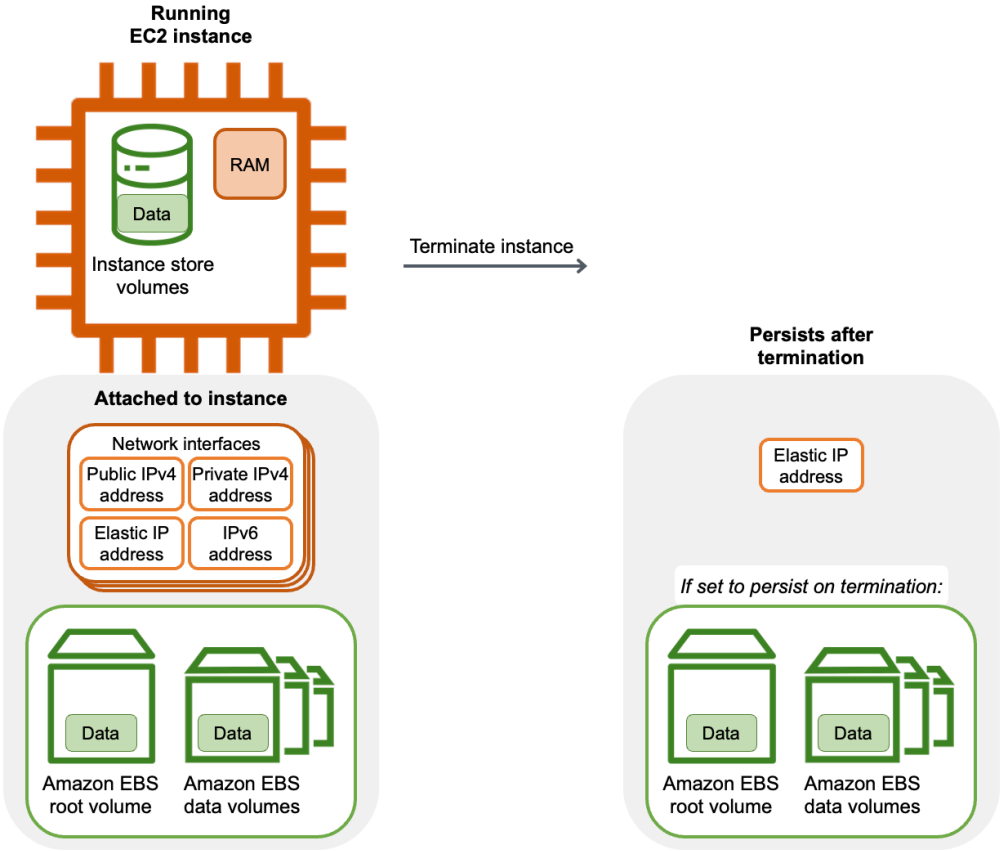
内容

- 实例终止的工作原理
- 终止实例的方法
- 终止实例并正常关闭操作系统
- 终止实例并绕过操作系统正常关闭
- 排查实例终止问题
- 更改实例终止保护
- 更改实例启动的关闭行为
- 实例终止时保留数据

实例终止的工作原理

当终止实例时，则会在实例的操作系统（OS）级别注册更改，一些资源会丢失，而一些资源会持续存在。

下图显示了 Amazon EC2 实例终止后丢失的内容和仍然存在的情况。实例终止后，任何实例存储卷上的数据和存储在实例 RAM 中的数据都将擦除。与实例关联的任何弹性 IP 地址都将分离。对于 Amazon EBS 根卷和数据卷，结果取决于每个卷的终止时删除设置。



注意事项

- 数据持久性
  - 实例存储卷：当实例终止时，所有数据将永久删除。
  - EBS 根卷：
    - 如果启动时附加，则实例终止时默认删除。
    - 如果启动后附加，则实例终止时默认存在。
  - EBS 数据卷：
    - 如果启动时使用控制台附加：实例终止时默认存在。
    - 如果启动时使用 CLI 附加：实例终止时默认删除。
    - 如果启动后使用控制台或 CLI 附加：实例终止时默认存在。

> **Note**
>
> 任何在实例终止时未删除的卷将继续产生费用。您可以更改设置，以便在实例终止时删除或保留卷。有关更多信息，请参阅 实例终止时保留数据。

- 防止意外终止
  - 要防止有人意外终止实例，请启用终止保护。
  - 要控制从实例启动关闭操作时实例停止还是终止，请更改实例启动的关闭行为。
- 关闭脚本：如果您在实例终止时运行脚本，您的实例可能会异常终止，因为我们无法确保关闭脚本运行。Amazon EC2 尝试彻底关闭实例并运行任何系统关闭脚本；但是，某些事件（如硬件故障）可能会阻止这些系统关闭脚本运行。
- 裸机实例：x86 裸机实例不支持协同关闭。

在终止实例时发生的情况

在操作系统级别注册的更改

- API 请求会向访客发送按钮按下事件。
- 该按钮按下事件致使各种系统服务停止。由 systemd（Linux）或系统进程（Windows）提供系统正常关闭。来自管理程序的 ACPI 关闭按钮按下事件触发正常关闭。
- 启动 ACPI 关闭。
- 正常关闭进程退出后，该实例将关闭。没有可配置的操作系统关闭时间。短时间内仍可在控制台中看到该实例，然后该条目将自动被删除。

资源丢失

- 实例存储卷中存储的数据。
- 如果 `DeleteOnTermination` 属性设置为 `true`，则为 EBS 根卷。
- 如果 `DeleteOnTermination` 属性设置为 `true`，则为 EBS 数据卷（在启动时或启动后附加）。

持续存在的资源

- 如果 `DeleteOnTermination` 属性设置为 `false`，则为 EBS 根卷。
- 如果 `DeleteOnTermination` 属性设置为 `false`，则为 EBS 数据卷（在启动时或启动后附加）。

测试应用程序对实例终止的响应

您可以使用 Amazon Fault Injection Service 测试您的实例终止时您的应用程序如何响应。有关更多信息，请参阅《Amazon Fault Injection Service 用户指南》。

终止实例的方法

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

有四种方法可以执行用户启动的实例终止：默认终止、跳过操作系统关闭终止、强制终止和跳过操作系统关闭强制终止。下表比较了各种终止方法之间的关键差异：

> **Note**
>
> 如果启用了终止保护，则无法终止实例。有关更多信息，请参阅更改实例终止保护。

终止方法	关键用途	使用案例	CLI 命令
默认终止	正常关闭实例，尝试正常关闭操作系统。	典型实例终止。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0`
终止并跳过操作系统关闭	终止实例时绕过操作系统正常关闭。	需要绕过正常关闭操作系统时。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --skip-os-shutdown`

终止方法	关键用途	使用案例	CLI 命令
强制终止	处理卡住的实例。首先尝试默认终止；如果实例无法终止，则强制终止实例。	当实例卡住在shutting-down 状态时。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --force`
强制终止并跳过操作系统关闭	强制终止并在终止实例时绕过操作系统正常关闭。	当需要强制终止并绕过操作系统正常关闭时。	`aws ec2 terminate-instances \ --instance-id i-1234567890abcdef0 \ --force \ --skip-os-shutdown`

有关如何使用每种方法的说明，请参阅以下内容：

- 终止实例并正常关闭操作系统
- 终止实例并绕过操作系统正常关闭
- 强制终止实例

终止实例并正常关闭操作系统

您可以使用默认的终止方法终止实例，其中包括尝试正常关闭操作系统。有关更多信息，请参阅 终止实例的方法。

Console

使用默认终止方法终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择相应实例，然后依次选择实例状态、终止（删除）实例。
4. 当系统提示您确认时，选择终止（删除）。
5. 在您终止某个实例之后，它会在短时间内保持可见，状态为 `terminated`。

   如果终止失败，或者已终止实例的可见时间超过几个小时，请参阅 已终止实例仍然显示。

Amazon CLI

使用默认终止方法终止实例

使用 terminate-instances 命令。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

使用默认终止方法终止实例

使用 Remove-EC2Instance cmdlet。

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

终止实例并绕过操作系统正常关闭

在终止实例时，您可以绕过操作系统正常关闭。有关更多信息，请参阅 终止实例的方法。

Console

终止实例并绕过操作系统正常关闭

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择相应实例，然后依次选择实例状态、终止（删除）实例。

4. 在跳过操作系统关闭下，选中跳过操作系统关闭复选框。如果您在控制台中看不到此选项，则当前区域的控制台尚未提供此选项。但是，您可以使用 Amazon CLI 或 SDK 访问此功能，也可以在控制台中尝试其他区域。
5. 选择终止（删除）。
6. 在您终止某个实例之后，它会在短时间内保持可见，状态为 `terminated`。

   如果终止失败，或者已终止实例的可见时间超过几个小时，请参阅 已终止实例仍然显示。

Amazon CLI

终止实例并绕过操作系统正常关闭

使用 terminate-instances 命令和 `--skip-os-shutdown`。

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --skip-os-shutdown
```

PowerShell

终止实例并绕过操作系统正常关闭

使用 Remove-EC2Instance cmdlet 和 `-SkipOsShutdown $true`。

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -SkipOsShutdown $true
```

排查实例终止问题

请求者必须具有调用 `ec2:TerminateInstances` 的权限。有关更多信息，请参阅使用实例的示例策略。

如果您终止您的实例而另一个实例启动，很可能您已通过 EC2 队列 或 Amazon EC2 Auto Scaling 等功能配置了自动扩展。有关更多信息，请参阅 自动启动或终止的实例。

> **Note**
>
> 如果启用了终止保护，则无法终止实例。有关更多信息，请参阅更改实例终止保护。

如果您的实例处于 shutting-down 状态的时间超出正常范围，可尝试强制终止。如果实例仍处于 shutting-down 状态，则应通过 Amazon EC2 服务中的自动进程进行清理（终止）。有关更多信息，请参阅 延迟的实例终止。

更改实例终止保护

要防止在使用 Amazon EC2 API 时意外终止实例（无论是直接还是使用其他接口（例如 Amazon EC2 控制台）调用 TerminateInstances），则可以为该实例启用终止保护。DisableApiTermination 属性用于控制是否可以终止实例。默认情况下，终止保护处于禁用状态。您可以在启动实例时、实例正在运行时或已停止时设置此属性的值。

当 InstanceInitiatedShutdownBehavior 属性设置为 terminate 时，DisableApiTermination 属性不会阻止您通过从实例启动关机来终止实例的操作（例如，使用操作系统的系统关机命令）。有关更多信息，请参阅 更改实例启动的关闭行为。

注意事项

- 在计划事件终止实例时，启用停止保护不会阻止 Amazon 终止实例。
- 启用终止保护不会阻止 Amazon EC2 Auto Scaling 在实例运行状况不佳，或在横向缩减事件期间终止实例。可以通过使用实例横向缩减保护，来控制自动扩缩组在横向缩减时是否可以终止特定实例。可以通过暂停 ReplaceUnhealthy 扩展过程，来控制自动扩缩组是否可以终止运行状况不佳的实例。
- 无法为竞价型实例启用终止保护。

Console

在实例启动时启用终止保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在控制面板上，选择启动实例。
3. 展开高级详细信息。对于终止保护，请选择启用。
4. 指定完实例详细信息后，选择启动实例。

为实例更改终止保护

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例。

4. 依次选择操作、实例设置、更改终止保护。
5. 对于终止保护，选择或清除启用。
6. 选择保存。

Amazon CLI

为实例启用终止保护

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --disable-api-termination
```

为实例禁用终止保护

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --no-disable-api-termination
```

PowerShell

为实例启用终止保护

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -DisableApiTermination $true
```

为实例禁用终止保护

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
```

```
    -DisableApiTermination $false
```

终止具有终止保护的多个实例

如果您在同一请求中终止跨多个可用区的多个实例，并且启用了一个或多个指定实例以进行终止保护，则请求失败，结果如下：

- 与受保护实例位于同一可用区中的指定实例不会终止。
- 位于不同可用区（其他指定实例不受保护）的指定实例将成功终止。

示例

假设您在两个可用区中有以下四个实例。

实例	可用区	终止保护
实例 1	可用区 A	Disabled
实例 2		Disabled
实例 3	可用区 B	Enabled
实例 4		Disabled

如果您尝试终止同一请求中的所有这些实例，请求将报告失败，结果如下：

- 实例 1 和实例 2 已成功终止，因为两个实例均未启用终止保护。
- 实例 3 和实例 4 无法终止，因为实例 3 已启用终止保护。

更改实例启动的关闭行为

Warning

终止实例是永久且不可逆转的。
终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。

在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

当从 Amazon EBS 支持的实例启动关闭（使用 shutdown 或 poweroff 命令）时，该实例默认会停止。您可以通过更改实例的 `InstanceInitiatedShutdownBehavior` 属性更改此行为，从而实例会改为终止。您可以在实例运行或停止时更改此属性。

halt 命令不会启动关闭。如果已经使用，实例并不会终止；而是将 CPU 置于 HLT 状态，实例将继续运行。

> **Note**
>
> 只有当您从实例本身的操作系统中执行关闭操作时，`InstanceInitiatedShutdownBehavior` 属性才适用。当使用 `StopInstances` API 或 Amazon EC2 控制台停止实例时，该属性不适用。

Console

更改实例启动的关闭操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 依次选择操作、实例设置、更改关闭操作。

   关闭行为显示当前行为。
5. 要更改行为，请从关闭行为中选择停止或终止。
6. 选择保存。

Amazon CLI

更改实例启动的关闭操作

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
```

```
    --instance-initiated-shutdown-behavior terminate
```

PowerShell

更改实例启动的关闭操作

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -InstanceInitiatedShutdownBehavior terminate
```

实例终止时保留数据

当 Amazon EC2 实例终止时，您可以在实例存储卷或 Amazon EBS 卷上保留数据。本主题将介绍如何确保数据在实例终止后仍然存在。

实例终止如何影响根卷和数据卷

实例存储卷

实例终止时，实例存储卷会自动删除，数据也会丢失。要在实例生命周期结束后保留这些数据，可在终止实例之前手动将数据复制到持久性存储，如 Amazon EBS 卷、Amazon S3 存储桶或 Amazon EFS 文件系统。有关更多信息，请参阅 适用于 Amazon EC2 实例的存储选项。

Amazon EBS 卷

实例终止时，EBS 卷将被删除或保留，具体取决于每个卷的 `DeleteOnTermination` 属性值：

- 是（控制台）/`true`（CLI）：实例终止时，卷将会删除。
- 否（控制台）/`false`（CLI）：实例终止时，卷将会保留。保留的卷将继续产生费用。

> **Note**
>
> 实例终止后，您可以为保留的卷拍摄快照，或将其附加到其他实例。为避免产生费用，您必须删除该卷。

EBS 卷的默认删除行为

默认的 DeleteOnTermination 值因卷类型、卷是在启动时还是启动后附加以及附加卷所用的方法（控制台或 CLI）而异：

卷类型	附加时间	附加方法	实例终止时的默认行为
根卷	在启动时	控制台或 CLI	删除
根卷	启动后	控制台或 CLI	Preserve
数据量	在启动时	控制台	Preserve
数据量	在启动时	CLI	删除
数据量	启动后	控制台和 CLI	Preserve

检查卷持久性设置

EBS 卷启动时的默认值由 AMI 上设置的 DeleteOnTermination 属性决定。您可以在实例启动时更改该值，以覆盖 AMI 设置。我们建议您在启动实例后验证 DeleteOnTermination 属性的默认设置。

要检查实例终止时是否会删除 Amazon EBS 卷

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 选择存储选项卡。
5. 在块设备下，向右滚动以选中终止时删除列。
   - 如果为是，则在实例终止时将会删除卷。
   - 如果为否，则在实例终止时不会删除卷。任何未删除的卷将继续产生费用。

将根卷更改为在启动时持久保留

您可以在启动实例时更改 EBS 根卷的 `DeleteOnTermination` 属性。您还可以对数据卷使用以下过程。

Console

在启动时更改要持久保留的实例根卷

1. 按照步骤启动实例，但请在完成以下步骤以将根卷更改为持久保留后，再启动实例。
2. 在配置存储窗格上，选择高级。
3. 在 EBS 卷下，展开根卷信息。
4. 对于终止时删除，选择是。
5. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

在启动时更改要持久保留的实例根卷

使用 run-instances 命令更改块设备映射中 `DeleteOnTermination` 的值。

添加 `--block-device-mappings` 选项：

```
--block-device-mappings file://mapping.json
```

在 `mapping.json` 中，指定设备名称，例如 `/dev/sda1` 或 `/dev/xvda`，并为 `DeleteOnTermination` 指定 `false`。

```
[
  {
    "DeviceName": "device_name",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

在启动时更改要持久保留的实例根卷

使用 New-EC2Instance cmdlet 更改块设备映射中 DeleteOnTermination 的值。

添加 -BlockDeviceMapping 选项：

```
-BlockDeviceMapping $bdm
```

在 bdm 中，指定设备名称，例如 /dev/sda1 或 /dev/xvda，并为 DeleteOnTermination 指定 false。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

将运行的实例的根卷更改为持久保留

您可以将正在运行的实例的根卷更改为持久保留。您还可以对数据卷使用以下过程。

Amazon CLI

将根卷更改为持久保留

使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0  \
    --block-device-mappings file://mapping.json
```

在 mapping.json 中，指定设备名称，例如 /dev/sda1 或 /dev/xvda，并为 --DeleteOnTermination 指定 false。

```
[
  {
    "DeviceName": "device_name",
```

```
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

将根卷更改为持久保留

使用 Edit-EC2InstanceAttribute cmdlet。

添加 `-BlockDeviceMapping` 选项：

```
-BlockDeviceMapping $bdm
```

在 bdm 中，指定设备名称，例如 `/dev/sda1` 或 `/dev/xvda`，并为 `DeleteOnTermination` 指定 `false`。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

实例指令引退

计划在 Amazon 检测到托管实例的基础硬件发生无法弥补的故障时停用实例。实例根卷类型会决定实例停用的行为：

- 如果实例的根卷是 Amazon EBS 卷，将停止实例，您可随时重新启动它。启动停止的实例会将其迁移到新的硬件。
- 如果实例根卷是实例存储卷，则实例将终止，且无法再次使用。

有关实例事件类型的更多信息，请参阅Amazon EC2 实例的计划事件。

目录

- 确定计划停用的实例

- 可对计划停用的由 EBS 支持的实例执行的操作
- 可对计划停用的由实例存储支持的实例执行的操作

确定计划停用的实例

如果已计划停用实例，您将在事件发生之前收到包含实例 ID 和停用日期的电子邮件。您还可以检查已计划停用的实例。

> **Important**
>
> 如果某个实例已计划停用，我们建议您尽快采取行动，因为该实例可能已经无法访问。有关更多信息，请参阅 Check if your instance is reachable。

识别已计划停用的实例的选项

- 监控账户联系人的电子邮件
- 检查实例

监控账户联系人的电子邮件

如果某个实例已计划停用，该账户的主要联系人和操作联系人会在该事件生效之前收到电子邮件。该电子邮件会包含实例 ID 和计划的停用日期。有关更多信息，请参阅《Amazon 账户管理 参考指南》中的 Update the primary contact for your Amazon account 和 Update the alternate contacts for your Amazon account。

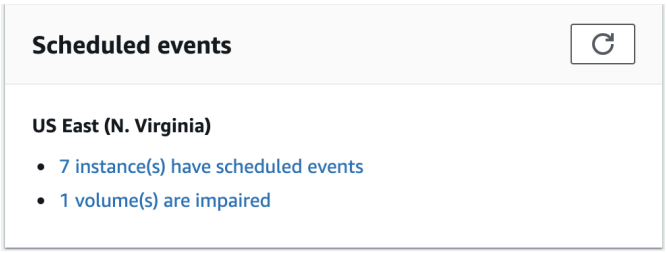
检查实例

如果您不会经常检查所用的电子邮件账户，则可能会错过实例停用通知。您可以随时检查是否有任何实例已计划停用。

Console

识别已计划停用的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 EC2 控制面板。在计划的事件下方，您可以看到与您的 Amazon EC2 实例和卷相关的事件，这些事件按区域列出。



3. 如果您的某个实例列有计划的事件，请选择区域名称下方的链接以转至 Events (事件) 页面。
4. 事件页面会列出与事件相关的所有资源。要查看计划停用的实例，请从第一个筛选列表中选择 Instance resources，然后从第二个筛选列表中选择 Instance stop or retirement。
5. 如果筛选结果显示有实例被计划停用，请选择该实例，并注意详细信息窗格中开始时间字段中的日期和时间。这就是您的实例停用的日期。

Amazon CLI

查找已计划停用的实例

使用以下 describe-instance-status 命令。在您有实例正在运行的每个区域重复此操作。

```
aws ec2 describe-instance-status --filters Name=event.code,Values=instance-
retirement
```

PowerShell

查找已计划停用的实例

使用 Get-EC2InstanceStatus cmdlet。在您有实例正在运行的每个区域重复此操作。

```
Get-EC2InstanceStatus -Filter @{Name="event.code"; Values="instance-retirement"}
```

可对计划停用的由 EBS 支持的实例执行的操作

要保留即将停用的实例上的数据，您可以执行以下某项操作。请务必在实例停用日期之前执行该操作，以防止意外的停机和数据丢失。

对于 Linux 实例，如果您不确定您的实例是由 EBS 还是由实例存储支持，则请参阅 Amazon EC2 实例的根卷。

检查实例是否可访问

当您收到计划停用实例的通知时，建议您尽快采取以下操作：

- 通过连接或 ping 实例来检查实例是否可以访问。
- 如果实例可以访问，则应计划在预定停用日期之前的某个适当的时间停止/启动实例，以影响最小为准。有关停止和启动实例以及在停止实例时可能发生的情况（例如，对与实例关联的公有、私有和弹性 IP 地址的影响）的更多信息，请参阅启动和停止 Amazon EC2 实例。请注意，停止和启动实例后，实例存储卷上的数据将会丢失。
- 如果您的实例无法访问，您应立即采取操作，执行停止/启动来恢复实例。
- 或者，如果您想终止实例，请尽快计划终止，以便停止为实例产生费用。

创建实例备份

从实例创建由 EBS 支持的 AMI，以便您可以有一个备份。为确保数据的完整性，请在创建 AMI 之前停止实例。您可以等到计划的停用日期（实例停止的日期），或者在停用日期之前自行停止实例。您可随时重新启动实例。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。

启动替换实例

从实例创建 AMI 后，您可以使用 AMI 启动替换实例。在 Amazon EC2 控制台中，选择您的新 AMI，然后选择从 AMI 启动实例。为您的实例配置参数，然后选择启动实例。有关每个字段的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

可对计划停用的由实例存储支持的实例执行的操作

要保留即将停用的实例上的数据，您可以执行以下某项操作。请务必在实例停用日期之前执行该操作，以防止意外的停机和数据丢失。

> **Warning**
>
> 如果具有实例存储根卷的实例超过了停用日期，则会终止该实例，并且无法恢复该实例或其中存储的任何数据。无论您的实例根卷是哪种类型，在停用实例后，存储在实例存储卷上的数据都会丢失，即使这些卷连接到具有 EBS 根卷的实例也不例外。

检查实例是否可访问

当您收到计划停用实例的通知时，建议您尽快采取以下操作：

- 通过连接或 ping 实例来检查实例是否可以访问。

- 如果实例无法访问，则很可能无法恢复实例。有关更多信息，请参阅 排查 Amazon EC2 实例无法访问的问题。Amazon 将在计划的停用日期终止您的实例，因此，对于无法访问的实例，您可以立即自行将其终止。

启动替换实例

使用 AMI 工具从实例创建 Amazon S3 支持的 AMI，如创建 Amazon S3 支持的 AMI中所述。在 Amazon EC2 控制台中，选择您的新 AMI，然后选择从 AMI 启动实例。为您的实例配置参数，然后选择启动实例。有关每个字段的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

将实例转换为由 EBS 支持的实例

将数据传输到 EBS 卷，制作卷的快照，然后从快照创建 AMI。您可以从新 AMI 启动替换实例。有关更多信息，请参阅 将 Amazon S3 支持的 AMI 转换为 EBS-backed AMI。

实例自动恢复

> **Important**
>
> 本节旨在介绍如何在 EC2 实例上主动配置恢复机制。这些恢复机制专门用于在 Amazon 检测到导致系统状态检查失败的底层硬件或软件问题时恢复实例可用性。如果目前存在实例访问问题，请参阅 EC2 实例问题排查。

如果 Amazon 检测到某个实例因底层硬件或软件问题而不可用，有两种机制可以用来自动恢复实例的可用性：简化的自动恢复和基于 Amazon CloudWatch 操作的恢复。恢复实例可用性又称实例恢复。

在实例恢复过程中，Amazon 会尝试将实例从存在底层硬件或软件问题的主机移至另一台主机。恢复成功后，实例恢复过程会在实例中显示为计划外重启。您可以验证是否已进行实例恢复。

如果恢复过程不成功，实例可能会继续在存在底层硬件或软件问题的主机上运行。此时需要手动加以干预。如果实例无法访问，或者系统状态检查仍然失败，建议手动停止并启动实例。启动实例时，实例通常会迁移到新的底层主机。不过，与实例自动恢复（即实例保留原有公有 IPv4 地址）不同，除非重启的实例具有弹性 IP 地址，否则其会收到新的公有 IPv4 地址。

要从自动恢复机制中受益，必须在系统状态检查失败之前，在实例上预先对机制进行配置。启动实例期间，默认会启用简化的自动恢复机制。您也可以选择在启动后配置基于 Amazon CloudWatch 操作的恢复机制。配置其中一种机制可以让实例更具弹性。

简化的自动恢复机制和基于 Amazon CloudWatch 操作的恢复机制仅适用于支持的实例。有关更多信息，请参阅启用简化的自动恢复的要求和启用基于 CloudWatch 操作的恢复的要求。

> **Warning**
>
> 在 Amazon 因底层硬件或软件问题而恢复实例后，须注意以下后果：存储在易失性存储器（RAM）中的数据会丢失，操作系统的正常运行时间会从零开始计算。此外，若使用基于 CloudWatch 操作的恢复机制，实例存储卷上的数据也会丢失。为协助防止数据丢失，我们建议您定期创建宝贵数据的备份。有关 EC2 实例备份和恢复最佳实践的更多信息，请参阅 Amazon EC2 最佳实践。
>
> 实例自动恢复机制是为单个实例设计的。有关构建弹性系统的指导，请参阅构建弹性系统。

主题

- 实例自动恢复的关键概念
- 简化的自动恢复机制和基于 CloudWatch 操作的恢复机制之间的区别
- 构建弹性系统
- 验证是否已进行实例自动恢复
- 在 Amazon EC2 实例上配置简化的自动恢复
- 在 EC2 实例上配置基于 CloudWatch 操作的恢复

实例自动恢复的关键概念

实例自动恢复是 Amazon EC2 的一项功能，可在底层硬件或软件发生故障时自动恢复实例可用性，从而增强 EC2 实例的弹性和可靠性。

以下是实例自动恢复的关键概念：

配置选项

可以配置两种机制来支持实例自动恢复：

- 简化的自动恢复：默认在支持的实例上启用。
- 基于 CloudWatch 操作的恢复：必须手动在支持的实例上配置。

系统状态检查

系统状态检查会自动监控运行 EC2 实例的 Amazon 基础设施。

- 如果系统状态检查失败，Amazon 会启动实例自动恢复，尝试将受影响的实例迁移到别的硬件。
- 系统状态检查失败表示主机的硬件或软件存在问题，而不是实例本身有问题。实例自动恢复机制可以恢复未通过系统状态检查的实例。不过，如果只有实例状态检查失败，则无法进行实例自动恢复。
- 有关实例状态检查和系统状态检查这二者间的区别，请参阅状态检查类型。

底层硬件或软件问题示例

可能导致系统状态检查失败的硬件或软件问题包括：网络连接丢失、系统电源损耗、物理主机上的软件问题以及物理主机上影响到网络连接状态的硬件问题。

已恢复实例的特性

除了丢失的元素外，已恢复实例与原始的实例完全相同。

保留的元素：

- 实例 ID
- 公有、私有和弹性 IP 地址
- 实例元数据
- 置放群组
- 附加的 EBS 卷
- 可用区

丢失的元素：

- 存储在易失性存储器（RAM）中的数据
- 存储在实例存储卷中的数据（仅适用于基于 CloudWatch 操作的恢复机制）
- 操作系统的正常运行时间重置为零

使用 CloudWatch 监控系统状态检查情况

CloudWatch 中的 StatusCheckFailed_System 指标表明系统状态检查通过与否。

指标值：

- 0：系统状态检查通过。
- 1：系统状态检查失败。

Amazon Health Dashboard 中的事件

在尝试执行实例自动恢复期间，Amazon 会根据配置的恢复机制以及相应结果向 Amazon Health Dashboard 发送事件：

- 简化的自动恢复
  - 成功事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS
  - 失败事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE
- CloudWatch 基于操作的恢复
  - 成功事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS
  - 失败事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE

简化的自动恢复机制和基于 CloudWatch 操作的恢复机制之间的区别

下表比较了简化的自动恢复和基于 CloudWatch 操作的恢复这两种机制间的主要区别。

比较点	简化的自动恢复	CloudWatch 基于操作的恢复
配置	默认在支持的实例上启用	必须手动配置 CloudWatch 警报和操作
弹性	修复由 Amazon 管理的恢复行为	可自定义的操作和条件
通知	通过 Amazon Health Dashboard 发送基本通知	通过 SNS 发送自定义通知
裸机实例大小	排除	包含
启动实例时附加的实例存储卷	不支持在启动时附加实例存储卷的实例	在选定的实例类型上受支持。请注意，在实例恢复过程中，实例存储卷上的数据会丢失。
恢复时间	标准恢复尝试	恢复尝试速度比简化的自动恢复更快
迁移期间解决了主机问题	迁移可以取消，并且实例仍保留在原主机上	继续迁移到新主机
成本	无额外费用	CloudWatch 可能会产生费用

构建弹性系统

虽然简化的自动恢复机制和基于 CloudWatch 操作的恢复机制都可以有效维护单个实例的可用性，但 Amazon 建议实施高可用性架构，允许流量失效转移到运行状况良好的实例。

为实现该目的，可以考虑使用 Elastic Load Balancing（在多个 EC2 实例之间分配传入流量）和 Amazon EC2 Auto Scaling（根据需求和运行状况自动调整实例数量）等 Amazon 服务。

有关使用 EC2 实例构建具备容错能力的弹性系统的更多信息，请参阅以下资源：

- Amazon YouTube 频道上的 Back to Basics: Designing for Failure with EC2
- Amazon Architecture Blog 网站上的 Disaster Recovery (DR) Architecture on Amazon, Part I: Strategies for Recovery in the Cloud
- 应用程序负载均衡器用户指南
- Amazon EC2 Auto Scaling 用户指南
- 《可靠性支柱 Amazon Well-Architected 框架》中的 REL11-BP02 失效转移到运行状况良好的资源

验证是否已进行实例自动恢复

如果实例出现脱机后又意外重启，则可能是因底层硬件或软件问题而进行了实例自动恢复。您可以在 Amazon Health Dashboard 中检查是否有实例自动恢复事件来加以验证。您还可以查看 StatusCheckFailed_System Amazon CloudWatch 指标，检查是否检测到实例的底层硬件或软件问题。

在 Amazon Health Dashboard 中检查是否有事件

在尝试执行实例自动恢复时，Amazon 会向 Amazon Health Dashboard 发送事件。具体事件取决于配置的恢复机制以及尝试成功与否。

在 Amazon Health Dashboard 中检查是否有实例自动恢复事件

1. 通过以下网址打开 Amazon Health Dashboard：https://phd.aws.amazon.com/phd/home#/。
2. 查找与实例自动恢复相关的事件。这些事件的存在可以确认是否进行了实例自动恢复以及恢复结果。
   - 简化的自动恢复
     - 成功事件：`AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`

    - 失败事件：AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE
  - CloudWatch 基于操作的恢复
    - 成功事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS
    - 失败事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE

使用 CloudWatch 监控系统状态检查情况

在 CloudWatch 中查看 StatusCheckFailed_System 指标，即可验证实例是否检测到了底层硬件或软件问题。该指标值会表明系统状态检查是通过（无硬件或软件问题）还是失败（有硬件或软件问题）。

验证是否检测出底层硬件或软件问题

1. 通过以下网址打开 CloudWatch 控制台的指标页面：https://console.aws.amazon.com/cloudwatch/home?#metricsV2。
2. 确认自己与 EC2 实例位于同一区域。
3. 将以下指标粘贴到指标搜索字段中，然后按 Enter。

   ```
   StatusCheckFailed_System
   ```

4. 选择 EC2 > 每个实例的指标。
5. 在表中，选中要检查的实例旁边的复选框。
6. 将查询时段更改为疑似发生了恢复事件的时间。
7. 选择绘成图表的指标选项卡，然后对 StatusCheckFailed_System 执行以下操作：

   a. 在统计数据字段中选择平均值、最大值或最小值。

   b. 对于周期，选择 1 分钟。
8. 检查 StatusCheckFailed_System 的值。
   - 值为 0：系统状态检查通过，表示没有底层硬件或软件问题。
   - 值为 1：系统状态检查失败，表示存在底层硬件或软件问题。

有关更多信息，请参阅 实例自动恢复。

在 Amazon EC2 实例上配置简化的自动恢复

> **Important**
>
> 本节旨在介绍如何在 EC2 实例上主动配置恢复机制。这些恢复机制专门用于在 Amazon 检测到导致系统状态检查失败的底层硬件或软件问题时恢复实例可用性。如果目前存在实例访问问题，请参阅 EC2 实例问题排查。

如果 Amazon 检测到某个实例因底层硬件或软件问题而无法使用，简化的自动恢复可以将该实例从存在底层问题的主机转移到其他主机，从而自动恢复实例的可用性。

如果进行了简化的自动恢复，Amazon 会根据结果向 Amazon Health Dashboard 发送以下某个事件：

- 成功事件：`AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_SUCCESS`
- 失败事件：`AWS_EC2_SIMPLIFIED_AUTO_RECOVERY_FAILURE`

要接收这些事件的通知，可以配置通知选项。有关更多信息，请参阅《Amazon 用户通知服务 User Guide》中的 Creating your first notification configuration in Amazon 用户通知服务。您还可以使用 Amazon EventBridge 规则来监控简化的自动恢复事件。

启动实例期间，默认在所有支持的实例上启用简化的自动恢复。不过，只有当实例处于 `running` 状态，Amazon Health Dashboard 中未列出任何服务事件，且该实例类型有可用容量时，该机制才能发挥作用。在某些情况下，例如出现严重中断问题，容量限制可能会导致恢复失败。有关更多信息，请参阅 the section called “排查简化的自动恢复失败问题”。

可以在启动期间或启动后禁用简化的自动恢复，再在之后需要时重新启用。

> **Warning**
>
> 在 Amazon 因底层硬件或软件问题而恢复实例后，须注意以下后果：存储在易失性存储器（RAM）中的数据会丢失，操作系统的正常运行时间会从零开始计算。为协助防止数据丢失，我们建议您定期创建宝贵数据的备份。有关 EC2 实例备份和恢复最佳实践的更多信息，请参阅 Amazon EC2 最佳实践。
>
> 实例自动恢复机制是为单个实例设计的。有关构建弹性系统的指导，请参阅构建弹性系统。

内容

- 启用简化的自动恢复的要求
- 配置简化的自动恢复
- 排查简化的自动恢复失败问题

启用简化的自动恢复的要求

只有满足以下条件的实例，才可以启用简化的自动恢复：

实例类型

- 通用
  型：A1、M3、M4、M5、M5a、M5n、M5zn、M6a、M6g、M6i、M6in、M7a、M7g、M7i、M7i-flex、M8a、M8azn、M8g、M8gb、M8gn、M8i、M8i-flex、M8in、M8ine、M8ib、M9g、T1、T2、T3、T3a、T4g
- 计算优化
  型：C3、C4、C5、C5a、C5n、C6a、C6g、C6gn、C6i、C6in、C7a、C7g、C7gn、C7i、C7i-flex、C8a、C8g、C8gb、C8gn、C8i、C8i-flex、C8in、C8ine、C8ib
- 内存优化
  型：R3、R4、R5、R5a、R5b、R5n、R6a、R6g、R6i、R6in、R7a、R7g、R7i、R7iz、R8a、R8g、
  flex、R8in、R8ib、U-3tb1、U-6tb1、U-9tb1、U-12tb1、U-18tb1、U-24tb1、U7i-6tb、U7i-8tb、U7i-12
- 加速计算型：G3、G5g、Inf1、P3、VT1
- 高性能计算：Hpc6a、Hpc7a、Hpc7g、Hpc8a

租赁

- 已共享
- 专用实例

  有关更多信息，请参阅 Amazon EC2 专用实例。

限制

如果实例存在以下特性，则不支持简化的自动恢复机制：

- 实例大小：`metal` 实例
- 租赁：专属主机。对专属主机使用专属主机自动恢复
- 存储：具有实例存储卷的实例

- 网络：使用 Elastic Fabric Adapter 的实例
- 自动扩缩：属于自动扩缩组的实例
- 维护：当前正在进行定期维护事件的实例

配置简化的自动恢复

启动支持的实例时，默认情况下会启用简化的自动恢复。启动实例期间或之后，您可以将自动恢复行为设置为 `disabled`。

`default` 配置不能为不受支持的实例启用简化的自动恢复。

Console

在启动时禁用简化自动恢复

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances（实例），然后选择 Launch instance（启动实例）。
3. 在高级详细信息部分，在实例自动恢复字段中选择禁用。
4. 根据需要配置剩余的实例启动设置，然后启动实例。

在启动后禁用简化自动恢复

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 请选择实例，然后依次选择 Actions（操作）、Instance settings（实例设置）、Change auto-recovery behavior（更改自动恢复行为）。
4. 选择 Off（关闭），然后选择 Save（保存）。

在启动后启用简化自动恢复

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 请选择实例，然后依次选择 Actions（操作）、Instance settings（实例设置）、Change auto-recovery behavior（更改自动恢复行为）。
4. 选择默认（开启），然后选择保存。

Amazon CLI

在启动时禁用简化自动恢复

使用 run-instances 命令和 --maintenance-options 选项。

```
--maintenance-options AutoRecovery=Disabled
```

在启动后禁用简化自动恢复

使用 modify-instance-maintenance-options 命令。

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery disabled
```

在启动后启用简化自动恢复

使用 modify-instance-maintenance-options 命令。

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-1234567890abcdef0 \
    --auto-recovery default
```

PowerShell

在启动时禁用简化自动恢复

使用 New-EC2Instance cmdlet。

```
-MaintenanceOptions_AutoRecovery Disabled
```

在启动后禁用简化自动恢复

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Disabled
```

在启动后启用简化自动恢复

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -AutoRecovery Enabled
```

排查简化的自动恢复失败问题

如果简化的自动恢复机制无法恢复实例，考虑是否存在以下问题：

- Amazon 服务事件正在运行

  简化的自动恢复在 Amazon Health Dashboard 中的服务事件期间不起作用。您可能不会收到此类事件的恢复失败通知。要了解最新的服务可用性信息，请参阅服务运行状况页面。

- 容量不足

  暂无足够的替换硬件来迁移实例。

- 已达到每日恢复尝试次数上限

  该实例已达到每天的恢复尝试操作限制。如果自动恢复失败，并且确定硬件性能下降是初始系统状态检查失败的根本原因，实例随后可能会被停用。

如果尽管多次尝试恢复，但实例的系统状态检查失败仍然存在，请参阅对状态检查失败的实例进行故障排除以获取更多指导。

在 EC2 实例上配置基于 CloudWatch 操作的恢复

> **Important**
>
> 本节旨在介绍如何在 EC2 实例上主动配置恢复机制。这些恢复机制专门用于在 Amazon 检测到导致系统状态检查失败的底层硬件或软件问题时恢复实例可用性。如果目前存在实例访问问题，请参阅 EC2 实例问题排查。

如果 Amazon 检测到某个实例因底层硬件或软件问题而无法使用，基于 CloudWatch 操作的恢复可以将该实例从存在底层问题的主机转移到其他主机，从而自动恢复实例的可用性。

如果进行了基于 CloudWatch 操作的恢复，Amazon 会根据结果向 Amazon Health Dashboard 发送以下某个事件：

- 成功事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_SUCCESS
- 失败事件：AWS_EC2_INSTANCE_AUTO_RECOVERY_FAILURE

您可以配置基于 CloudWatch 操作的恢复机制，将恢复操作添加到 Amazon CloudWatch 警报中。CloudWatch 基于操作的恢复与 StatusCheckFailed_System 指标结合使用。CloudWatch 基于操作的恢复可提供分钟级恢复响应时间粒度，以及有关恢复操作和结果的 Amazon Simple Notification Service（Amazon SNS）通知。与简化的自动恢复相比，这些配置选项可以更精细地控制系统状态检查失败事件响应，从而更快地尝试恢复。有关可用 CloudWatch 选项的更多信息，请参阅实例的状态检查。

不过，只有当实例处于 running 状态，Amazon Health Dashboard 中未列出任何服务事件，且该实例类型有可用容量时，基于 CloudWatch 操作的恢复机制才能发挥作用。在某些情况下，例如出现严重中断问题，容量限制可能会导致恢复失败。有关更多信息，请参阅 the section called “故障排除”。

> **Warning**
>
> 在 Amazon 因底层硬件或软件问题而恢复实例后，须注意以下后果：存储在易失性存储器（RAM）和实例存储卷中的数据会丢失，操作系统的正常运行时间会从零开始计算。为协助防止数据丢失，我们建议您定期创建宝贵数据的备份。有关 EC2 实例备份和恢复最佳实践的更多信息，请参阅 Amazon EC2 最佳实践。
> 实例自动恢复机制是为单个实例设计的。有关构建弹性系统的指导，请参阅构建弹性系统。

内容

- 启用基于 CloudWatch 操作的恢复的要求
- 配置 CloudWatch 基于操作的恢复
- 排查基于 CloudWatch 操作的恢复失败问题

启用基于 CloudWatch 操作的恢复的要求

只有满足以下条件的实例，才可以启用基于 CloudWatch 操作的恢复：

实例类型

- 通用
  型：A1、M3、M4、M5、M5a、M5n、M5zn、M6a、M6g、M6i、M6in、M7a、M7g、M7i、M7i-

flex、M8a、M8azn、M8g、M8gb、M8gn、M8i、M8i-flex、M8in、M8ine、M8ib、M9g、T1、T2、T3、T3a、T4g

- 计算优化
  型：C3、C4、C5、C5a、C5n、C6a、C6g、C6gn、C6i、C6in、C7a、C7g、C7gn、C7i、C7i-flex、C8a、C8g、C8gb、C8gn、C8i、C8i-flex、C8in、C8ine、C8ib
- 内存优化
  型：R3、R4、R5、R5a、R5b、R5n、R6a、R6g、R6i、R6in、R7a、R7g、R7i、R7iz、R8a、R8g、
  flex、R8in、R8ib、U-3tb1、U-6tb1、U-9tb1、U-12tb1、U-18tb1、U-24tb1、U7i-6tb、U7i-8tb、U7i-12
- 加速计算型：G3、G5g、Inf1、P3、VT1
- 高性能计算：Hpc6a、Hpc7a、Hpc7g、Hpc8a
- 裸机实例：具有裸机实例大小的任何上述实例类型。
- 如果在启动时添加了实例存储卷：则仅支持以下实例类型：
  M3、C3、R3、X1、X1e、X2idn、X2iedn

租赁

- 已共享
- 专用实例

有关更多信息，请参阅 Amazon EC2 专用实例。

限制

如果实例存在以下特性，则不支持基于 CloudWatch 操作的恢复机制：

- 租赁：专属主机。对专属主机使用专属主机自动恢复
- 网络：使用 Elastic Fabric Adapter 的实例
- 自动扩缩：属于自动扩缩组的实例
- 维护：当前正在进行定期维护事件的实例

查找受支持的实例类型

您可以查看支持基于 CloudWatch 操作的恢复功能的实例类型。

Console

查看支持基于 CloudWatch 操作的恢复的实例类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instance Types（实例类型）。
3. 在筛选条件栏中，添加筛选条件自动恢复支持 = true。实例类型表显示了支持基于 Amazon CloudWatch 操作的恢复的所有实例类型。
4. （可选）添加筛选条件以进一步筛选相关的特定实例类型。

Amazon CLI

查看支持基于 CloudWatch 操作的恢复的实例类型

使用 describe-instance-types 命令和 `auto-recovery-supported` 筛选条件。

```
aws ec2 describe-instance-types \
    --filters Name=auto-recovery-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

查看支持基于 CloudWatch 操作的恢复的实例类型

使用 Get-EC2InstanceType cmdlet 和 `auto-recovery-supported` 筛选条件。

```
Get-EC2InstanceType `
    -Filter @{Name="auto-recovery-supported";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

配置 CloudWatch 基于操作的恢复

要为 EC2 实例配置基于 CloudWatch 操作的恢复，请创建 CloudWatch 警报来监控指定实例的 `StatusCheckFailed_System` 指标。设置指标值为 1 时触发警报，表示系统状态检查失败。配置警报操作，在警报触发时自动恢复实例。

您可以使用 Amazon EC2 控制台或 CloudWatch 控制台配置警报。如需有关说明，请参阅本用户指南中的在 Amazon CloudWatch 警报中添加恢复操作，或者《Amazon CloudWatch 用户指南》中的在 Amazon CloudWatch 警报中添加恢复操作。

排查基于 CloudWatch 操作的恢复失败问题

如果基于 CloudWatch 操作的恢复机制无法恢复实例，考虑是否存在以下问题：

- Amazon 服务事件正在运行

  CloudWatch 基于操作的恢复在 Amazon Health Dashboard 中的服务事件期间不起作用。您可能不会收到此类事件的恢复失败通知。要了解最新的服务可用性信息，请参阅服务运行状况页面。
- 容量不足

  暂无足够的替换硬件来迁移实例。
- 已达到每日恢复尝试次数上限

  该实例已达到每天的恢复尝试操作限制。如果自动恢复失败，并且确定硬件性能下降是初始系统状态检查失败的根本原因，实例随后可能会被停用。

如果尽管多次尝试恢复，但实例的系统状态检查失败仍然存在，请参阅对状态检查失败的实例进行故障排除以获取更多指导。

使用实例元数据管理 EC2 实例

实例元数据 是有关您的实例的数据，可以用来配置或管理正在运行的实例。实例元数据包括以下内容：

实例元数据属性

实例元数据属性分为几类，例如，主机名、事件和安全组。

动态数据

动态数据是实例启动时生成的元数据，如实例身份文件。有关更多信息，请参阅 动态数据类别。

用户数据

您也可以使用实例元数据访问您启动实例时指定的用户数据。例如，您可以指定参数以配置实例，或者包含简单的脚本。您还可以构建通用 AMI，并使用用户数据修改启动时提供的配置文件。例

如，如果您为各种小型企业运行 Web 服务器，则这些企业可以全部使用相同的通用 AMI，并在启动时从您在用户数据中指定的 Amazon S3 存储桶中检索其内容。要随时添加新客户，请为客户创建一个存储桶，添加其内容，并使用在用户数据中为您的代码提供的唯一存储桶名称启动 AMI。如果使用同一个 `RunInstances` 调用启动多个实例，则用户数据可供该预留中的所有实例使用。属于同一保留的每个实例都具有唯一的 `ami-launch-index` 编号，这允许您编写代码控制实例的行为。例如，第一台主机可能会选择将自己作为集群中的原始节点。有关详细的 AMI 启动示例，请参阅。识别在单个请求中启动的每个实例

Important

虽然您只能从实例本身中访问实例元数据和用户数据，但并未使用身份验证或加密方法对数据进行保护。任何可以直接访问实例的人以及可能在实例上运行的任何软件都可以查看其元数据。因此，您不应将敏感数据（例如密码或长期保存的加密密钥）存储为用户数据。

内容

- 实例元数据类别
- 动态数据类别
- 访问 EC2 实例的实例元数据
- 配置实例元数据服务选项
- 在启动包含用户数据输入的 EC2 实例时运行命令
- 识别在单个请求中启动的每个实例

实例元数据类别

实例元数据属性分为几类。要检索实例元数据属性，请在请求中指定类别，然后响应中会返回元数据。

有新类别发布时，系统会使用新版本号创建新的实例元数据构建。在下表中，Version when category was released（类别发布时的版本）列指定实例元数据类别发布时的构建版本。为避免每次 Amazon EC2 发布新的实例元数据构建时都必须更新您的代码，请在元数据请求中使用 `latest`，而不是版本号。有关更多信息，请参阅 获取实例元数据的可用版本。

当 Amazon EC2 发布新的实例元数据类别时，新类别的实例元数据可能不适用于现有实例。对于基于 Nitro 的实例，只能检索启动时可用类别的实例元数据。对于具有 Xen 虚拟机管理程序的实例，可以停止然后启动实例以更新可用于实例的类别。

下表列举了实例元数据的类别。一些类别名称包括实例独有的数据的占位符。例如，*mac* 表示网络接口的 MAC 地址。在检索实例元数据时，您必须使用实际值替换占位符。

类别	说明	类别发布时的版本
ami-id	用于启动实例的 AMI ID。	1.0
ami-launch-index	如果使用同一个 RunInstances 调用启动多个实例，此值将指明各个实例的启动顺序。第一个启动的实例的值是 0。如果使用自动扩缩或 EC2 实例集启动实例，此值将始终为 0。	1.0
ami-manifest-path	指向 Amazon S3 中的 AMI 清单文件的路径。如果您使用 Amazon EBS-backed AMI 来启动实例，则返回的结果为 unknown。	1.0
ancestor-ami-ids	为创建此 AMI 而重新绑定的任何实例的 AMI ID。仅当 AMI 清单文件包含一个 ancestor-amis 密钥时，此值才存在。	2007-10-10
autoscaling/target-lifecycle-state	显示 Auto Scaling 实例正在转换到的目标 Auto Scaling 生命周期状态的值。当实例在 2022 年 3 月 10 日之后转换到目标生命周期状态之一时显示。可能的值：Detached | InService | Standby | Terminated | Warmed:Hibernated | Warmed:Running | Warmed:Stopped | Warmed:Terminated 。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的通过	2021-07-15

类别	说明	类别发布时的版本
	实例元数据检索目标生命周期状态。	
block-device-mapping/ami	包含根/启动文件系统的虚拟设备。	2007-12-15
block-device-mapping/ebs否	与任何 Amazon EBS 卷关联的虚拟设备。仅当 Amazon EBS 卷在启动时存在或者在上一次启动该实例时存在时，这些卷才在元数据中可用。N 表示 Amazon EBS 卷的索引（例如 ebs1 或 ebs2）。	2007-12-15
block-device-mapping/ephemeral 否	任何非 NVMe 实例存储卷的虚拟设备。N 表示每个卷的索引。块储存设备映射中的实例存储卷数可能与实例的实际实例存储卷数不匹配。实例类型将决定对实例可用的实例存储卷的数量。如果块储存设备映射中的实例存储卷数超过了对实例可用的实例存储卷数，则其他实例存储卷将被忽略。	2007-12-15
block-device-mapping/root	与根设备关联的虚拟设备或分区或虚拟设备上的分区，其中根（/ 或 C:）文件系统与给定实例相关联。	2007-12-15
block-device-mapping/swap	与 swap 关联的虚拟设备。并不总是存在。	2007-12-15
events/maintenance/history	如果实例存在已完成或已取消的维护事件，则包含一个 JSON 字符串，其中包含有关事件的信息。	2018-08-17

类别	说明	类别发布时的版本
events/maintenance/scheduled	如果实例存在活动的维护事件，则包含一个 JSON 字符串，其中包含有关事件的信息。有关更多信息，请参阅查看会影响 Amazon EC2 实例的计划事件。	2018-08-17
events/recommendations/rebalance	为实例发出 EC2 实例再平衡建议通知的大致时间 (UTC)。下面是此类别的元数据示例：{"noticeTime": "2020-11-05T08:22:00Z"} 。此类别仅在发出通知后可用。有关更多信息，请参阅 EC2 实例再平衡建议。	2020-10-27
hostname	如果 EC2 实例使用的是基于 IP 的命名 (IPBN)，则此为实例的私有 IPv4 DNS 主机名。如果 EC2 实例使用的是基于资源的命名 (RBN)，则此为 RBN。在存在多个网络接口的情况下，其指的是 eth0 设备 (设备号为 0 的设备)。有关 IPBN 和 RBN 的更多信息，请参阅 EC2 实例主机名和域。	1.0
iam/info	如果存在与实例关联的 IAM 角色，则包含有关实例配置文件上次更新时间的信息 (包括实例的 LastUpdated 日期、InstanceProfileArn 和 InstanceProfileId)。如果没有，则不显示。	2012-01-12

类别	说明	类别发布时的版本
iam/security-credentials/role-name	如果存在与实例关联的 IAM 角色，则 *role-name* 为角色的名称，并且 *role-name* 包含与角色关联的临时安全凭证 (有关更多信息，请参阅 从实例元数据中检索安全凭证)。如果没有，则不显示。	2012-01-12
identity-credentials/ec2/info	identity-credentials/ec2/security-credentials/ec2-instance 中凭证的相关信息。	2018-05-23
identity-credentials/ec2/security-credentials/ec2-instance	允许实例软件向 Amazon 确认自身身份，以支持 EC2 Instance Connect 及 Amazon Systems Manager 默认主机管理配置等功能的实例身份角色凭证。这些凭证并未附加策略，因此除了将实例标识为 Amazon 功能外，它们没有其他 Amazon API 权限。有关更多信息，请参阅 Amazon EC2 实例的实例身份角色。	2018-05-23
instance-action	通知实例在准备打包时重新启动。有效值：none | shutdown | bundle-pending 。	2008-09-01
instance-id	此实例的 ID。	1.0
instance-life-cycle	此实例的采购选项。有关更多信息，请参阅Amazon EC2 账单和购买选项。	2019-10-01
instance-type	实例的类型。有关更多信息，请参阅Amazon EC2 实例类型。	2007-08-29

类别	说明	类别发布时的版本
ipv6	实例的 IPv6 地址。在存在多个网络接口的情况下，其指的是 eth0 设备（设备号为 0 的设备）的网络接口和第一个分配的 IPv6 地址。若网络接口 [0] 上不存在 IPv6 地址，则不会设置此项目且会导致 HTTP 404 响应。	2021-01-03
kernel-id	此实例启动的内核的 ID，如果适用的话。	2008-02-01
local-hostname	在存在多个网络接口的情况下，其指的是 eth0 设备 (设备号为 0 的设备)。如果 EC2 实例使用的是基于 IP 的命名 (IPBN)，则此为实例的私有 IPv4 DNS 主机名。如果 EC2 实例使用的是基于资源的命名 (RBN)，则此为 RBN。有关 IPBN、RBN 和 EC2 实例命名的更多信息，请参阅 EC2 实例主机名和域。	2007-01-19
local-ipv4	实例的私有 IPv4 地址。在存在多个网络接口的情况下，其指的是 eth0 设备 (设备号为 0 的设备)。若此为仅 IPv6 的实例，则不会设置此项目且会导致 HTTP 404 响应。	1.0
mac	实例的媒体访问控制 (MAC) 地址。在存在多个网络接口的情况下，其指的是 eth0 设备 (设备号为 0 的设备)。	2011 年 1 月 1 日
metrics/vhostmd	不再可用。	2011-05-01

类别	说明	类别发布时的版本
network/interfaces/macs/mac/device-number	与该接口关联的唯一设备号。设备号与设备名称对应；例如，device-number 为 2 对应于 eth2 设备。此类别对应的是 Amazon CLI 的 Amazon EC2 API 和 EC2 命令使用的 DeviceIndex 和 device-index 字段。	2011 年 1 月 1 日
network/interfaces/macs/mac/interface-id	网络接口的 ID。	2011 年 1 月 1 日
network/interfaces/macs/mac/ipv4-associations/public-ip	与每个公用 IP 地址相关联并被分配到该接口的私有 IPv4 地址。	2011 年 1 月 1 日
network/interfaces/macs/mac/ipv6s	分配给接口的 IPv6 地址。	2016-06-30
network/interfaces/macs/mac/ipv6-prefix	分配给网络接口的 IPv6 前缀。	
network/interfaces/macs/mac/local-hostname	实例的私有 IPv4 DNS 主机名。在存在多个网络接口的情况下，其指的是 eth0 设备 (设备号为 0 的设备)。若此为仅 IPv6 的实例，则此为基于资源的名称。有关 IPBN 和 RBN 的更多信息，请参阅 EC2 实例主机名和域。	2007-01-19
network/interfaces/macs/mac/local-ipv4s	与接口关联的私有 IPv4 地址。若此为仅 IPv6 的网络接口，则不会设置此项目且会导致 HTTP 404 响应。	2011 年 1 月 1 日
network/interfaces/macs/mac/mac	该实例的 MAC 地址。	2011 年 1 月 1 日

类别	说明	类别发布时的版本
network/interfaces/macs/*mac*/network-card	网卡的索引。有些实例类型支持多个网卡。	2020-11-01
network/interfaces/macs/mac/owner-id	网络接口拥有者的 ID。在多个接口的环境中，接口可由第三方连接，如 Elastic Load Balancing。接口拥有者需为接口上的流量付费。	2011 年 1 月 1 日
network/interfaces/macs/mac/public-hostname	接口的公有 DNS (IPv4)。仅当 enableDnsHostnames 属性设置为 true 时，才返回此类别。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。如果实例仅具有公有 IPv6 地址而没有公有 IPv4 地址，则不会设置此项目并导致 HTTP 404 响应。	2011 年 1 月 1 日
network/interfaces/macs/mac/public-ipv4s	与接口关联的公有 IP 地址或弹性 IP 地址。一个实例上可能有多个 IPv4 地址。	2011 年 1 月 1 日
network/interfaces/macs/mac/security-groups	网络接口所属的安全组。	2011 年 1 月 1 日
network/interfaces/macs/mac/security-group-ids	网络接口所属的安全组的 ID。	2011 年 1 月 1 日
network/interfaces/macs/mac/subnet-id	接口所驻留的子网的 ID。	2011 年 1 月 1 日
network/interfaces/macs/mac/subnet-ipv4-cidr-block	接口所在子网的 IPv4 CIDR 块。	2011 年 1 月 1 日

类别	说明	类别发布时的版本
network/interfaces/macs/mac/subnet-ipv6-cidr-blocks	接口所在子网的 IPv6 CIDR 块。	2016-06-30
network/interfaces/macs/mac/vpc-id	接口所驻留的 VPC 的 ID。	2011 年 1 月 1 日
network/interfaces/macs/mac/vpc-ipv4-cidr-block	VPC 的主 IPv4 CIDR 块。	2011 年 1 月 1 日
network/interfaces/macs/mac/vpc-ipv4-cidr-blocks	VPC 的 IPv4 CIDR 块。	2016-06-30
network/interfaces/macs/mac/vpc-ipv6-cidr-blocks	接口所在 VPC 的 IPv6 CIDR 块。	2016-06-30
placement/availability-zone	实例启动的可用区。	2008-02-01
placement/availability-zone-id	在其中启动实例的静态可用区 ID。可用区 ID 在账户之间保持一致。但是，它可能与可用区不同，后者可能因账户而异。	2019-10-01
placement/group-name	在其中启动实例的置放群组的名称。	2020-08-24
placement/host-id	启动实例的主机的 ID。仅适用于专用主机。	2020-08-24
placement/partition-number	在其中启动实例的分区编号。	2020-08-24
placement/region	在其中启动实例的 Amazon 区域。	2020-08-24

类别	说明	类别发布时的版本
product-codes	与实例相关联的 Amazon Web Services Marketplace 产品代码（如果有）。	2007-03-01
public-hostname	实例的公有 DNS (IPv4)。仅当 enableDnsHostnames 属性设置为 true 时，才返回此类别。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。如果实例仅具有公有 IPv6 地址而没有公有 IPv4 地址，则不会设置此项目并导致 HTTP 404 响应。	2007-01-19
public-ipv4	公有 IPv4 地址。如果弹性 IP 地址与实例相关联，返回的值是弹性 IP 地址。	2007-01-19
public-keys/0/openssh-key	公有密钥。仅在实例启动时提供了公有密钥的情况下可用。	1.0
ramdisk-id	启动时指定的 RAM 磁盘 的 ID，如果适用的话。	2007-10-10
reservation-id	预留的 ID。	1.0
security-groups	应用到实例的安全组的名称。 在启动后，您可以更改实例的安全组。这些更改将体现在此处和 network/interfaces/macs/*mac*/security-groups 中。	1.0
services/domain	区域的 Amazon 资源所在的域。	2014-02-25

类别	说明	类别发布时的版本
services/partition	资源所处的分区。对于标准 Amazon 区域，分区是 aws。如果资源位于其他分区，则分区是 aws-*partitionname* 。例如，中国（北京）区域中的资源的分区为 aws-cn。	2015-10-20
spot/instance-action	操作（休眠、停止或终止）和操作发生的大致时间（用 UTC 表示）。仅在已将竞价型实例标记为休眠、停止或终止时才提供此项目。有关更多信息，请参阅instance-action。	2016-11-15
spot/termination-time	竞价型实例 操作系统将收到关闭信号的大致时间 (UTC)。仅当竞价型实例已由 Amazon EC2 标记为终止时，此项目才会出现并包含时间值（例如，2015-01-05T18:02:00Z）。如果您自己终止了竞价型实例，那么终止时间项目不会设置为时间。有关更多信息，请参阅termination-time。	2014-11-05
system	实例的底层虚拟化类型（虚拟机监控程序）。	2022-09-24
tags/instance	与实例关联的实例标签。只有明确允许访问实例元数据中的标签时才可用。有关更多信息，请参阅 启用对实例元数据中标签的访问权限。	2021-03-23

动态数据类别

下表列举了动态数据的类别。

类别	说明	类别发布时的版本
fws/instance-monitoring	显示客户是否在 CloudWatch 中启用了详细的一分钟监控的值。有效值：enabled | disabled	2009-04-04
instance-identity/document	包含实例属性 (如实例 ID、私有 IP 地址等) 的 JSON。请参阅 Amazon EC2 实例的实例身份文档。	2009-04-04
instance-identity/pkcs7	用于验证签名的文档的真实性和内容。请参阅 Amazon EC2 实例的实例身份文档。	2009-04-04
instance-identity/signature	可被其他各方用于验证来源和真实性的数据。请参阅 Amazon EC2 实例的实例身份文档。	2009-04-04

访问 EC2 实例的实例元数据

您可以从实例内或者从 EC2 控制台、API、SDK 或 Amazon CLI 访问 EC2 实例元数据。要从控制台或命令行获取实例的当前实例元数据设置，请参阅查询现有实例的实例元数据选项。

您还可以修改具有 EBS 根卷的实例的用户数据。该实例必须处于已停止状态。有关控制台说明，请参阅 更新实例用户数据。有关使用 Amazon CLI 的 Linux 示例，请参阅 modify-instance-attribute。有关使用 Tools for Windows PowerShell 的 Windows 示例，请参阅 the section called “用户数据和 Tools for Windows PowerShell”。

> **Note**
>
> 您无需为用于检索实例元数据和用户数据的 HTTP 请求付费。

实例元数据访问注意事项

为避免实例元数据问题，请注意以下因素。

因 IMDSv2 强制使用 (**`HttpTokensEnforced=enabled`**) 导致实例启动失败

在启用 IMDSv2 强制使用之前，首先需要确保实例上的所有软件都支持 IMDSv2，然后才能将默认设置更改为禁用 IMDSv1 (`httpTokens=required`)，再后才可以启用强制使用。有关更多信息，请参阅 转换为使用 实例元数据服务版本 2。

命令格式

根据使用的是实例元数据服务版本 1 (IMDSv1) 还是实例元数据服务版本 2 (IMDSv2)，命令格式会有所不同。默认情况下，您可以使用两个版本的实例元数据服务。要要求使用 IMDSv2，请参阅使用实例元数据服务访问实例元数据。

如果需要 IMDSv2，则 IMDSv1 不起作用

如果使用 IMDSv1 后未收到任何响应，则很可能需要 IMDSv2。要检查是否需要 IMDSv2，请选择实例以查看其详细信息。IMDSv2 值要么表示必选（即必须使用 IMDSv2），要么表示可选（即可以使用 IMDSv2 或 IMDSv1）。

（IMDSv2）使用 /latest/api/token 来检索令牌

将 PUT 请求发放到任何版本特定的路径（例如 `/2021-03-23/api/token`）将导致元数据服务返回 403 禁止错误。这是预期行为。

元数据版本

为避免每次 Amazon EC2 发布新的实例元数据构建时都必须更新您的代码，我们建议您在路径中使用 `latest`，而不是版本号。

IPv6 支持

要使用 IPv6 地址检索实例元数据，必须确保启用并使用 IPv6 地址 IMDS `[fd00:ec2::254]` 而不是 IPv4 地址 `169.254.169.254`。该实例必须是在支持 IPv6 的子网中启动的基于 Nitro 的实例。

（Windows）使用 Windows Sysprep 创建自定义 AMI

为了确保在您从自定义 Windows AMI 启动实例时 IMDS 能够正常工作，AMI 必须是使用 Windows Sysprep 创建的标准化映像。否则，IMDS 将无法正常运行。有关更多信息，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。

在容器环境中，考虑重新配置或将跃点限制增加到 2

默认情况下，Amazon开发工具包使用 IMDSv2 调用。如果 IMDSv2 调用没有收到任何响应，某些 Amazon SDK 将重试调用，如果仍然不成功，则使用 IMDSv1。这可能会导致延迟，尤其是在容器环境中。对于那些需要 IMDSv2 的 Amazon SDK，如果容器环境中的跃点限制为 1，则调用可能根本收不到响应，因为转到容器被视为额外的网络跃点。

要缓解容器环境中的这些问题，请考虑更改配置以将设置（例如 Amazon Web Services 区域）直接传递到容器，或者考虑将跃点限制增加到 2。有关跃点限制影响的信息，请参阅 Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service。有关更改跃点限制的信息，请参阅更改 PUT 响应跃点限制。

每秒数据包（PPS）限制

对于使用本地链路地址的服务，每秒数据包数（PPS）限制为 1024 个。此限制是 Route 53 Resolver DNS 查询、实例元数据服务（IMDS）请求、Amazon Time Service 网络时间协议（NTP）请求和 Windows 许可服务（适用于基于 Microsoft Windows 的实例）请求的总和。

用户数据访问的其他注意事项

- 用户数据会被视为非透明数据；您指定什么数据就会在检索时得到什么数据。由实例来解释用户数据并对其进行操作。
- 用户数据必须采用 base64 编码。根据您使用的工具或 SDK，可能会为您执行 base64 编码。例如：
  - Amazon EC2 控制台可以为您执行 base64 编码或接受 base64 编码的输入。
  - 默认情况下，Amazon CLI 版本 2 会为您执行二进制参数的 base64 编码。Amazon CLI 版本 1 会为您执行 `--user-data` 参数的 base64 编码。
  - 适用于 Python (Boto3) 的 Amazon SDK 会为您执行 `UserData` 参数的 base64 编码。
- 用户数据在进行 base64 编码之前的原始格式的大小限制为 16 KB。长度为 n 的字符串在进行 base64 编码之后的大小为 ceil(n/3)*4。
- 在检索用户数据时，必须对其进行 base64 解码。如果您使用实例元数据或控制台检索数据，则会自动对数据进行解码。
- 如果您停止实例，修改用户数据，然后启动实例，则在启动实例时，不会自动运行更新后的用户数据。对于 Windows 实例，您可以配置设置，这样更新后的用户数据脚本在您启动实例时运行一次，或者在每次重启或启动实例时运行。
- 用户数据是一种实例属性。如果您从实例创建 AMI，则实例用户数据不包含在该 AMI 中。

从 EC2 实例内访问实例元数据

由于您的正在运行的实例存在实例元数据，因此您无需使用 Amazon EC2 控制台或 Amazon CLI。这在您编写脚本以实现从实例运行时非常有用。例如，您可从实例元数据访问您的实例的本地 IP 地址来以管理与外部应用程序的连接。

以下所有内容均被视为实例元数据，但其访问方式不同。选择代表您要访问的实例元数据类型的选项卡，以查看更多信息。

Metadata

实例元数据属性分为几类。有关每个实例元数据类别的描述，请参阅实例元数据类别。

要访问正在运行的实例内的实例元数据属性，请从以下 IPv4 或 IPv6 URI 获取数据。这些 IP 地址是链路本地地址，仅从该实例访问时有效。有关更多信息，请参阅 链路本地地址。

IPv4

```
http://169.254.169.254/latest/meta-data/
```

IPv6

```
http://[fd00:ec2::254]/latest/meta-data/
```

Dynamic data

要从正在运行的实例中检索动态数据，请使用以下 URI 之一。

IPv4

```
http://169.254.169.254/latest/dynamic/
```

IPv6

```
http://[fd00:ec2::254]/latest/dynamic/
```

示例：使用 cURL 访问

以下示例使用 cURL 检索高级实例身份类别。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/
instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

示例：使用 PowerShell 访问

以下示例使用 PowerShell 检索高级实例身份类别。

IMDSv2

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

IMDSv1

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-
identity/
document
rsa2048
pkcs7
signature
```

有关动态数据的详细信息和如何对其进行检索的示例，请参阅 Amazon EC2 实例的实例身份文档。

User data

要从实例中检索用户数据，请使用以下 URI 之一。要使用 IPv6 地址检索用户数据，必须将启用该功能，并且实例必须是在支持 IPv6 的子网中的基于 Nitro 的实例。

IPv4

```
http://169.254.169.254/latest/user-data
```

IPv6

```
http://[fd00:ec2::254]/latest/user-data
```

请求用户数据时，按原样返回数据 (内容类型 application/octet-stream)。如果该实例没有任何用户数据，则请求将返回 404 - Not Found。

示例：使用 cURL 访问以检索逗号分隔的文本

以下示例使用 cURL 检索以逗号分隔文本形式指定的用户数据。

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

示例：使用 PowerShell 访问以检索逗号分隔的文本

以下示例使用 PowerShell 检索以逗号分隔文本形式指定的用户数据。

IMDSv2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = Invoke-RestMethod -Headers
 @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
-Method PUT -Uri http://169.254.169.254/latest/api/token} -Method GET -uri
 http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

示例：使用 cURL 访问以检索脚本

以下示例使用 cURL 检索指定为脚本的用户数据。

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/user-
data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

示例：使用 PowerShell 访问以检索脚本

以下示例使用 PowerShell 检索指定为脚本的用户数据。

IMDSv2

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri
 http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
```

```
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

IMDSv1

```
Invoke-RestMethod -uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

查询现有实例的实例元数据选项

您可以查询现有实例的实例元数据选项。

Console

查询现有实例的实例元数据选项

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您的实例并选中以下字段：
   - IMDSv2：该值为必填或可选。
   - 允许在实例元数据中使用标签：该值为已启用或已禁用。
4. 选择实例后，依次选择操作、实例设置、修改实例元数据选项。

   该对话框显示所选实例的实例元数据服务是处于已启用状态还是已禁用状态。

Amazon CLI

查询现有实例的实例元数据选项

可以使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-id i-1234567898abcdef0 \
```

```
    --query 'Reservations[].Instances[].MetadataOptions'
```

PowerShell

使用 Tools for PowerShell 查询现有实例的实例元数据选项

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567898abcdef0).Instances.MetadataOptions
```

响应和错误消息

所有实例元数据以文本形式返回（HTTP 内容类型 `text/plain`）。

特定元数据资源的请求返回相应的值；如果资源不可用，则返回 HTTP 错误代码 `404 - Not Found`。

对通用元数据资源的请求 (以 / 结尾的 URI) 会返回一个可用资源列表，如果此类资源不存在，则会返回 HTTP 错误代码 `404 - Not Found`。列表中的各个项目位于被换行符 (ASCII 10) 终止的不同的行上。

如果 IMDSv1 请求没有收到响应，则很可能需要 IMDSv2。

对于使用 IMDSv2 发出的请求，可能会返回以下 HTTP 错误代码：

- `400 - Missing or Invalid Parameters` – PUT 请求无效。
- `401 - Unauthorized` – GET 请求使用无效的令牌。建议的措施是生成新的令牌。
- `403 - Forbidden` – 该请求不被允许，或者 IMDS 已关闭。
- `404 - Not Found` – 资源不可用或没有此类资源。
- `503` – 无法完成请求。重试请求。

如果 IMDS 返回错误，curl 会在输出中打印错误消息并返回成功状态代码。错误消息存储在 TOKEN 变量中，这会导致使用该令牌的 curl 命令失败。如果使用 -f 选项调用 curl，则会在出现 HTTP 服务器错误时返回错误状态代码。如果启用错误处理功能，Shell 可以捕获错误并停止脚本。

查询限制

我们基于每个实例来限制对 IMDS 的查询，并且对从实例到 IMDS 的同时连接数进行限制。

如果您使用 IMDS 检索 Amazon 安全凭证，请避免在每个事务期间查询凭证或从大量线程或进程中并发查询凭证，因为这可能会导致节流。相反，我们建议您缓存凭证，直到凭证开始接近其到期时间。有关 IAM 角色以及与其关联的安全凭证的更多信息，请参阅 从实例元数据中检索安全凭证。

如果在访问 IMDS 时受到限制，请使用指数回退策略重试查询。

使用实例元数据服务访问实例元数据

您可以使用以下其中一种方法，从正在运行的实例中访问实例元数据：

- 实例元数据服务版本 2 (IMDSv2)：一种面向会话的方法

  有关示例，请参阅 IMDSv2 的示例。
- 实例元数据服务版本 1（IMDSv1）– 一种请求/响应方法

  有关示例，请参阅 IMDSv1 的示例。

默认情况下，您可以使用 IMDSv1 和/或 IMDSv2。

您可以在每个实例上配置实例元数据服务（IMDS），使其仅接受 IMDSv2 调用，这将导致 IMDSv1 调用失败。有关如何配置实例以使用 IMDSv2 的信息，请参阅 配置实例元数据服务选项。

PUT 或 GET 标头对于 IMDSv2 是唯一的。如果请求中存在这些标头，则该请求针对 IMDSv2。如果不存在标头，则假定该请求针对 IMDSv1。

有关 IMDSv2 的详细回顾，请参阅借助 EC2 实例元数据服务的增强功能，为开放式防火墙、反向代理和 SSRF 漏洞增加深度防御。

主题

- 实例元数据服务版本 2 的工作原理
- 使用支持的 Amazon 开发工具包
- IMDSv2 的示例
- IMDSv1 的示例

实例元数据服务版本 2 的工作原理

IMDSv2 使用面向会话的请求。对于面向会话的请求，您创建一个会话令牌以定义会话持续时间，该时间最少为 1 秒，最多为 6 小时。在指定的持续时间内，您可以将相同的会话令牌用于后续请求。在指定的持续时间到期后，您必须创建新的会话令牌以用于将来的请求。

> **Note**
>
> 本部分中的示例使用实例元数据服务（IMDS）的 IPv4 地址：169.254.169.254。如果要通过 IPv6 地址检索 EC2 实例的实例元数据，请确保启用并改用 IPv6 地址：[fd00:ec2::254]。IMDS 的 IPv6 地址与 IMDSv2 命令兼容。IPv6 地址仅可在支持 IPv6 的子网（双栈或仅 IPv6）中基于 Nitro 的实例上访问。

以下示例使用 Shell 脚本和 IMDSv2 检索顶级实例元数据项。每个示例：

- 使用 PUT 请求创建持续 6 小时（21600 秒）的会话令牌
- 将会话令牌标头存储在名为 TOKEN（Linux 实例）或 token（Windows 实例）的变量中
- 使用令牌请求顶级元数据项

Linux 示例

您可以运行两个单独的命令，也可以将它们组合使用。

单独的命令

首先，使用以下命令生成令牌。

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
```

然后，通过令牌使用以下命令生成顶级元数据项。

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

组合命令

您可以存储令牌并组合命令。以下示例将上述两个命令组合在一起，并将会话令牌标头存储在名为 TOKEN 的变量中。

> **Note**
>
> 如果创建令牌时发生错误，则不会生成有效令牌，而会在变量中存储一条错误消息，该命令将不起作用。

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

在创建令牌后，您可以重复使用令牌，直到令牌过期。在以下示例命令（获取用于启动实例的 AMI 的 ID）中，将重复使用上一示例中的令牌（存储在 $TOKEN 中）。

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/ami-id
```

Windows 示例

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -
Uri http://169.254.169.254/latest/meta-data/
```

在创建令牌后，您可以重复使用令牌，直到令牌过期。在以下示例命令（获取用于启动实例的 AMI 的 ID）中，将重复使用上一示例中的令牌（存储在 $token 中）。

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
 -Method GET -uri http://169.254.169.254/latest/meta-data/ami-id
```

在使用 IMDSv2 请求实例元数据时，请求必须包含以下内容：

1. 使用 PUT 请求启动到实例元数据服务的会话。PUT 请求返回一个令牌，该令牌必须包含在对实例元数据服务的后续 GET 请求中。需要具有该令牌才能使用 IMDSv2 访问元数据。
2. 将该令牌包含在对 IMDS 的所有 GET 请求中。如果将令牌使用设置为 `required`，没有有效令牌或令牌过期的请求将显示 `401 - Unauthorized` HTTP 错误代码。
   - 令牌是实例特定的密钥。令牌在其他 EC2 实例上无效，如果尝试在生成令牌的实例外部使用，令牌将会被拒绝。
   - PUT 请求必须包含一个标头，它以秒为单位指定令牌的生存时间 (TTL)，最多为 6 小时（21600 秒）。令牌表示一个逻辑会话。TTL 指定令牌的有效时间长度，因而指定会话的持续时间。
   - 在令牌过期后，要继续访问实例元数据，您必须使用另一 PUT 创建新会话。
   - 您可以选择在每个请求中重复使用令牌或创建新的令牌。对于少量请求，每次需要访问 IMDS 时生成令牌并立即使用可能会更容易。但为了提高效率，您可以为令牌指定更长的持续时间并重复

使用令牌，而不必在每次需要请求实例元数据时都编写 PUT 请求。对并发令牌数量没有实际限制，每个令牌表示自己的会话。不过，IMDSv2 仍然受到正常 IMDS 连接和节流限制的制约。有关更多信息，请参阅 查询限制。

允许在 IMDSv2 实例元数据请求中使用 HTTP GET 和 HEAD 方法。如果 PUT 请求包含 X-Forwarded-For 标头，则会被拒绝。

默认情况下，PUT 请求的响应在 IP 协议级别的响应跃点数限制（生存时间）为 1。如果需要更大的跃点数限制，可使用 modify-instance-metadata-options Amazon CLI 命令进行调整。例如，您可能需要使用更大的跃点数限制，以便与实例上运行的容器服务保持向后兼容。有关更多信息，请参阅 为现有实例修改实例元数据选项。

使用支持的 Amazon 开发工具包

要使用 IMDSv2，EC2 实例必须使用支持 IMDSv2 的 Amazon 开发工具包版本。所有 Amazon 开发工具包的最新版本均支持使用 IMDSv2。

> **Important**
>
> 我们建议您及时了解开发工具包版本，以了解其最新功能、安全更新和底层依赖项。不建议继续使用不受支持的开发工具包版本，但是否继续使用由您自行决定。有关更多信息，请参阅《Amazon SDK 和工具参考指南》中的 Amazon SDK 和工具维护策略。

以下是支持使用 IMDSv2 的最低版本：

- Amazon CLI – 1.16.289
- Amazon Tools for Windows PowerShell – 4.0.1.0
- 适用于 .NET 的 Amazon SDK – 3.3.634.1
- 适用于 C++ 的 Amazon SDK – 1.7.229
- 适用于 Go 的 Amazon SDK – 1.25.38
- Amazon SDK for Go v2 – 0.19.0
- 适用于 Java 的 Amazon SDK – 1.11.678
- Amazon SDK for Java 2.x – 2.10.21
- 适用于 Node.js 中 JavaScript 的 Amazon 开发工具包 – 2.722.0
- 适用于 Kotlin 的 Amazon SDK：1.1.4

- 适用于 PHP 的 Amazon SDK – 3.147.7
- 适用于 Python (Botocore) 的 Amazon SDK – 1.13.25
- 适用于 Python (Boto3) 的 Amazon SDK – 1.12.6
- 适用于 Ruby 的 Amazon SDK – 3.79.0

IMDSv2 的示例

在您的 Amazon EC2 实例上运行以下示例，以检索 IMDSv2 的实例元数据。

在 Windows 实例上，您可以使用 Windows PowerShell，也可以安装 cURL 或 wget。如果您在 Windows 实例上安装第三方工具，请确保您仔细阅读随附的文档，因为调用和输出格式可能与此处所述的内容不同。

示例

- 获取实例元数据的可用版本
- 获取顶级元数据项
- 获取元数据项的值
- 获取可用的公有密钥列表
- 显示可以使用公有密钥 0 的格式
- 获取公有密钥 0（采用 OpenSSH 密钥格式）
- 获取实例的子网 ID
- 获取实例的实例标签

获取实例元数据的可用版本

此示例可以获取实例元数据的可用版本。当有新的实例元数据类别发布时，每个版本都引用一个实例元数据构建。实例元数据构建版本与 Amazon EC2 API 版本不相关。如果您有依赖于以前版本中所存在的结构和信息的脚本，则您可使用早期版本。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
1.0
2007-01-19
```

```
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

获取顶级元数据项

此示例获得顶级元数据项目。有关响应中项目的更多信息，请参阅 实例元数据类别。

请注意，只有在允许您访问的情况下，标签才会包含在此输出中。有关更多信息，请参阅 the section called “启用对实例元数据中标签的访问权限”。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

获取元数据项的值

这些示例获取在前面示例中获取的某些顶级元数据项的值。这些请求使用在前面的示例中使用命令创建的存储令牌。令牌不得过期。

cURL

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/reservation-id
r-0efghijk987654321
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/
latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

获取可用的公有密钥列表

此示例获得可用公有密钥的列表。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

显示可以使用公有密钥 0 的格式

此示例显示了可以使用公有密钥 0 的格式。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
openssh-key
```

获取公有密钥 0（采用 OpenSSH 密钥格式）

此示例获得公有密钥 0 (以 OpenSSH 密钥格式)。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-
aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

获取实例的子网 ID

此示例获取实例的子网 ID。

cURL

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

获取实例的实例标签

如果对实例元数据中的实例标签的访问权限已开启，则可以从实例元数据中获得实例标签。有关更多信息，请参阅 从实例元数据中检索标签。

IMDSv1 的示例

在您的 Amazon EC2 实例上运行以下示例，以检索 IMDSv1 的实例元数据。

在 Windows 实例上，您可以使用 Windows PowerShell，也可以安装 cURL 或 wget。如果您在 Windows 实例上安装第三方工具，请确保您仔细阅读随附的文档，因为调用和输出格式可能与此处所述的内容不同。

示例

- 获取实例元数据的可用版本
- 获取顶级元数据项
- 获取元数据项的值
- 获取可用的公有密钥列表
- 显示可以使用公有密钥 0 的格式

- 获取公有密钥 0（采用 OpenSSH 密钥格式）
- 获取实例的子网 ID
- 获取实例的实例标签

获取实例元数据的可用版本

此示例可以获取实例元数据的可用版本。当有新的实例元数据类别发布时，每个版本都引用一个实例元数据构建。实例元数据构建版本与 Amazon EC2 API 版本不相关。如果您有依赖于以前版本中所存在的结构和信息的脚本，则您可使用早期版本。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
```

```
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
...
latest
```

获取顶级元数据项

此示例获得顶级元数据项目。有关响应中项目的更多信息，请参阅 实例元数据类别。

请注意，只有在允许您访问的情况下，标签才会包含在此输出中。有关更多信息，请参阅 the section called “启用对实例元数据中标签的访问权限”。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
```

```
reservation-id
security-groups
services/
tags/
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
tags/
```

获取元数据项的值

这些示例获取在先前示例中获取的某些顶级元数据项的值。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id
```

```
r-0efghijk987654321
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

获取可用的公有密钥列表

此示例获得可用公有密钥的列表。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-keys/ 0=my-public-key
```

显示可以使用公有密钥 0 的格式

此示例显示了可以使用公有密钥 0 的格式。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
openssh-key
```

获取公有密钥 0（采用 OpenSSH 密钥格式）

此示例获得公有密钥 0 (以 OpenSSH 密钥格式)。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key
```

```
ssh-rsa MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

获取实例的子网 ID

此示例获取实例的子网 ID。

cURL

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

PowerShell

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/network/
interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

获取实例的实例标签

如果对实例元数据中的实例标签的访问权限已开启，则可以从实例元数据中获得实例标签。有关更多信息，请参阅 从实例元数据中检索标签。

转换为使用 实例元数据服务版本 2

如果要将实例配置为仅接受实例元数据服务版本 2（IMDSv2）调用，我们建议您使用以下工具和转换途径。

主题

- 转换为 IMDSv2 的工具
- 要求 IMDSv2 的建议途径

转换为 IMDSv2 的工具

以下工具可以帮助您识别、监控和管理软件从 imdsv1 到 imdsv2 的转换。有关如何使用这些工具的说明，请参阅要求 IMDSv2 的建议途径。

Amazon 软件

最新版本的 Amazon CLI 和 Amazon SDK 均支持 IMDSv2。要使用 IMDSv2，请更新 EC2 实例以使用最新版本。有关支持 IMDSv2 的最低 Amazon 开发工具包版本，请参阅 使用支持的 Amazon 开发工具包。

所有的 Amazon Linux 2 和 Amazon Linux 2023 软件包都支持 IMDSv2。Amazon Linux 2023 默认禁用 IMDSv1。

IMDS Packet Analyzer

IMDS Packet Analyzer 是一个开源工具，用于识别和记录您的实例启动阶段和运行时阶段的 IMDSv1 调用。通过分析这些日志，您可以精确地识别对您的实例进行 imdsv1 调用的软件，并确定需要更新哪些内容以仅在您的实例上支持 IMDSv2。您可以从命令行运行 IMDS Packet Analyzer，也可以将其作为服务安装。有关更多信息，请参阅 GitHub 上的 Amazon ImdsPacketAnalyzer。

CloudWatch

CloudWatch 提供以下两个指标来监控实例：

`MetadataNoToken`：IMDSv2 使用由令牌支持的会话，而 IMDSv1 不使用。`MetadataNoToken` 指标跟踪对使用 IMDSv1 的实例元数据服务（IMDS）的调用次数。通过查看该指标是否为零，您可以确定是否以及何时将所有软件升级为使用 IMDSv2。

`MetadataNoTokenRejected`：禁用 IMDSv1 后，您可以使用 `MetadataNoTokenRejected` 指标来跟踪 IMDSv1 调用被尝试和拒绝的次数。通过跟踪该指标，您可以确定您的软件是否需要更新以使用 IMDSv2。

对于每个 EC2 实例，这些指标是互斥的。启用 IMDSV1 (`httpTokens = optional`) 后，将仅会发出 `MetadataNoToken`。禁用 IMDSV1 (`httpTokens = required`) 后，将仅会发出 `MetadataNoTokenRejected`。有关何时该使用这些指标，请参阅要求 IMDSv2 的建议途径。

有关更多信息，请参阅 实例指标。

启动 API

新实例：使用 RunInstances API 启动要求必须使用 IMDSv2 的新实例。有关更多信息，请参阅 为新实例配置实例元数据选项。

现有实例：使用 ModifyInstanceMetadataOptions API 要求必须在现有实例上使用 IMDSv2。有关更多信息，请参阅 为现有实例修改实例元数据选项。

自动扩缩组启动的新实例：要规定必须在自动扩缩组启动的所有新实例上使用 IMDSv2，您的自动扩缩组可以使用启动模板或启动配置。当您创建启动模板或创建启动配置时，必须配置 `MetadataOptions` 参数以要求使用 IMDSv2。自动扩缩组将使用新的启动模板或启动配置来启动新实例，但现有实例不受影响。

自动扩缩组中的现有实例：使用 ModifyInstanceMetadataOptions API 要求必须在现有实例上使用 IMDSv2，也可终止实例，然后由自动扩缩组使用在新启动模板或启动配置中定义的实例元数据选项设置启动新的替换实例。

AMI

将 `ImdsSupport` 参数设置为 `v2.0` 的 AMI 将启动默认需要 imdsv2 的实例。Amazon Linux 2023 配置了 `ImdsSupport = v2.0`。

新 AMI：创建新 AMI 时，使用 register-image CLI 命令将 `ImdsSupport` 参数设置为 `v2.0`。

现有 AMI：修改现有 AMI 时，使用 modify-image-attribute CLI 命令将 `ImdsSupport` 参数设置为 `v2.0`。

有关更多信息，请参阅 配置 AMI。

账户级别控制

可以在账户级别配置所有实例元数据选项的默认值。启动实例时将会自动应用默认值。有关更多信息，请参阅将 IMDSv2 设置为账户默认设置。

您也可以在账户级别强制要求使用 IMDSv2。启用 IMDSv2 强制使用后：

- 新实例：配置为在启动时启用 IMDSv1 的实例将会启动失败
- 已禁用 IMDSv1 的现有实例：将阻止尝试在现有实例上启用 IMDSv1。
- 已启用 IMDSv1 的现有实例：已启用 IMDSv1 的现有实例不受影响。

有关更多信息，请参阅 在账户级别强制使用 IMDSv2。

IAM policy 和 SCP

您可以使用 IAM policy 或 Amazon Organizations 服务控制策略（SCP）来控制用户，具体如下所示：

- 除非将实例配置为使用 IMDSv2，否则无法使用 RunInstances API 启动实例。
- 无法通过使用 ModifyInstanceMetadataOptions API 修改正在运行的实例来重新启用 IMDSv1。

IAM policy 或 SCP 必须包含以下 IAM 条件键：

- `ec2:MetadataHttpEndpoint`
- `ec2:MetadataHttpPutResponseHopLimit`
- `ec2:MetadataHttpTokens`

如果 API 或 CLI 调用中的参数与包含该条件键的策略中指定的状态不一致，则 API 或 CLI 调用将失败并显示 `Unauthorized0peration` 响应。

此外，您还可以选择额外的保护层以强制从 IMDSv1 更改为 IMDSv2。在与通过 EC2 角色凭证调用的 API 相关的访问管理层上，您可以在 IAM 策略或 Amazon Organizations 服务控制策略（SCP）中使用条件键。具体来说，通过在 IAM policy 中使用值为 `ec2:RoleDelivery` 的条件键 `2.0`，使用从 IMDSv1 获取的 EC2 角色凭证进行的 API 调用将会收到 `Unauthorized0peration` 响应。通过使用 SCP 所需的该条件，可以更广泛地实现相同的效果。这会确保通过 IMDSv1 提供的凭证不能实际用于调用 API，因为任何不符合指定条件的 API 调用将会收到 `Unauthorized0peration` 错误。

有关示例 IAM policies，请参阅 使用实例元数据。有关 SCP 的更多信息，请参阅《Amazon Organizations User Guide》中的 Service control policies。

声明性策略

使用声明性策略（Amazon Organizations 的一项功能）为整个组织集中设置 IMDS 账户默认值，包括 IMDSv2 强制使用。有关示例策略，请参阅《Amazon Organizations 用户指南》中 Supported declarative policies 部分的实例元数据选项卡。

要求 IMDSv2 的建议途径

在使用上述工具时，我们建议您按照以下途径转换为 IMDSv2：

- 步骤1：识别 IMDSv2=optional 的实例，并审核 imdsV1 使用情况
- 步骤 2：将软件更新为 IMDSv2
- 步骤 3：在实例上要求 IMDSv2

- 步骤 4：将 IMDSv2=required 设置为默认值
- 步骤 5：强制实例要求 IMDSv2

步骤1：识别 IMDSv2=optional 的实例，并审核 imdsV1 使用情况

要评测 IMDSv2 迁移范围，请识别配置为允许 IMDSv1 或 IMDSv2 的实例，并审核 imdsv1 调用。

1. 识别配置为允许 IMDSv1 或 IMDSv2 的实例：

   Amazon EC2 console

   1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   2. 在导航窗格中，选择 Instances (实例)。
   3. 要仅查看配置为允许 IMDSv1 或 IMDSv2 的实例，请添加筛选条件 IMDSv2 = optional。
   4. 或者，要查看 IMDSv2 对于所有实例是 optional 还是 required，请打开首选项窗口（齿轮图标），打开 IMDSv2，然后选择确认。这会将 IMDSv2 列添加到实例表中。

   Amazon CLI

   使用 describe-instances 命令并按 `metadata-options.http-tokens = optional` 进行筛选，如下所示：

   ```
   aws ec2 describe-instances --filters "Name=metadata-options.http-
   tokens,Values=optional" --query "Reservations[*].Instances[*].[InstanceId]" --
   output text
   ```

2. 审核每个实例上的 IMDSv1 调用：

   使用 CloudWatch 指标 `MetadataNoToken`。此指标显示对实例上 IMDS 的 IMDSv1 调用次数。有关更多信息，请参阅实例指标。

3. 识别您的实例上进行 IMDSv1 调用的软件：

   使用开源 IMDS Packet Analyzer 在实例启动阶段和运行时操作期间识别和记录 IMDSv1调用。使用此信息确定要更新的软件，以使您的实例准备好仅使用 IMDSv2。您可以从命令行运行 IMDS Packet Analyzer，也可以将其作为服务安装。

步骤 2：将软件更新为 IMDSv2

将在您的实例上使用角色凭证的所有 SDK、CLI 和软件更新为与 IMDSv2 兼容的版本。有关更新 CLI 的更多信息，请参阅《Amazon Command Line Interface 用户指南》中的安装或更新最新版本的 Amazon CLI。

步骤 3：在实例上要求 IMDSv2

在通过 `MetadataNoToken` 指标确认零个 IMDSv1 调用后，请将现有实例配置为需要 IMDSv2。此外，将所有新实例配置为需要 IMDSv2。换句话说，在所有现有实例和新实例上禁用 IMDSv1。

1. 将现有实例配置为需要 IMDSv2：

   Amazon EC2 console

   1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   2. 在导航窗格中，选择 Instances (实例)。
   3. 选择实例。
   4. 依次选择操作、实例设置、修改实例元数据选项。
   5. 对于IMDSv2，选择必需。
   6. 选择保存。

   Amazon CLI

   使用 modify-instance-metadata-options CLI 命令指定仅使用 IMDSv2。

   > **Note**
   > 您可以在正在运行的实例上修改此设置。此更改立即生效，无需重启实例。

   有关更多信息，请参阅 要求使用 IMDSv2。
2. 禁用 IMDSv1 后监控是否存在问题：

   1. 使用 CloudWatch 指标 `MetadataNoTokenRejected` 跟踪 IMDSv1 调用尝试次数以及被拒绝的次数。

2. 如果 `MetadataNoTokenRejected` 指标记录了遇到软件问题的实例上的 IMDSv1 调用，则表明该软件需要更新才能使用 IMDSv2。

3. 将新实例配置为需要 IMDSv2：

   Amazon EC2 console

   1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   2. 按照以下步骤启动实例。
   3. 展开高级详细信息，对于元数据版本，选择仅 V2（需要令牌）。
   4. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。

      有关更多信息，请参阅 在启动时配置实例。

   Amazon CLI

   Amazon CLI：使用 run-instances 命令并指定需要使用 IMDSv2。

步骤 4：将 IMDSv2=required 设置为默认值

您可以将 IMDSv2=required 设置为账户或组织级别的默认配置。这确保所有新启动的实例都自动配置为需要 IMDSv2。

1. 设置账户级别默认值：

   Amazon EC2 console

   1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
   2. 在导航窗格中，选择控制面板。
   3. 在账户属性卡的设置下，选择数据保护和安全。
   4. 在 IMDS 默认值下，选择管理。
   5. 对于实例元数据服务，选择启用。
   6. 对于元数据版本，选择仅 V2（需要令牌）。
   7. 选择更新。

Amazon CLI

使用 modify-instance-metadata-defaults CLI 命令并指定 `--http-tokens required` 和 `--http-put-response-hop-limit 2`。

有关更多信息，请参阅 将 IMDSv2 设置为账户默认设置。

2. 或者，使用声明性策略设置组织级别默认值：

使用声明性策略将 IMDSv2 的组织默认值为 required。有关示例策略，请参阅《Amazon Organizations 用户指南》中 Supported declarative policies 部分的实例元数据选项卡。

步骤 5：强制实例要求 IMDSv2

确认没有任何实例依赖 IMDSv1 后，我们建议您在所有新实例上强制使用 IMDSv2。

使用以下选项之一来强制使用 IMDSv2：

1. 使用账户属性强制使用 IMDSv2

可以在账户级别为每个 Amazon Web Services 区域强制要求使用 IMDSv2。强制使用后，实例只有配置为必须使用 IMDSv2 时才能启动。无论实例配置或 AMI 配置如何，都必须遵守此强制使用要求。有关更多信息，请参阅 在账户级别强制使用 IMDSv2。要在组织级别应用此设置，请设置声明性策略。有关示例策略，请参阅《Amazon Organizations 用户指南》中 Supported declarative policies 部分的实例元数据选项卡。

要防止强制使用被撤销，应使用 IAM 策略来阻止对 ModifyInstanceMetadataDefaults API 对访问。有关更多信息，请参阅 使用 IAM policy。

> **Note**
>
> 此设置不会更改现有实例的 IMDS 版本，但会阻止在当前已禁用 IMDSv1 的现有实例上启用 IMDSv1。

> **Warning**
>
> 如果启用了 IMDSv2 强制使用，但未在启动时的实例配置中或在账户设置或 AMI 配置中将 `httpTokens` 设置为 `required`，则实例启动将会失败。有关问题排查信息，请参阅启动启用 IMDSv1 的实例时失败。

2. 也可使用以下 IAM 或 SCP 条件键强制使用 IMDSv2：

   - `ec2:MetadataHttpTokens`
   - `ec2:MetadataHttpPutResponseHopLimit`
   - `ec2:MetadataHttpEndpoint`

   这些条件键控制 RunInstances 和 ModifyInstanceMetadataOptions API 及相应 CLI 的使用。如果创建了策略，并且 API 调用中的参数与使用条件键的策略中指定的状态不匹配，API 或 CLI 调用将失败并显示 `UnauthorizedOperation` 响应。

   有关示例 IAM policies，请参阅 使用实例元数据。

限制访问实例元数据服务

您可以考虑使用本地防火墙规则，以禁止从某些或所有进程中访问实例元数据服务（IMDS）。

对于基于 Nitro 的实例，如果 VPC 内的网络设备（例如虚拟路由器）会将数据包转发到 IMDS 地址，且实例的默认源/目标检查已被禁用，则可以从您自己的网络访问 IMDS。要防止来自 VPC 外的源访问 IMDS，建议您修改网络设备的配置，丢弃目标 IPv4 地址为 IMDS `169.254.169.254` 的数据包，如果您启用了 IPv6 端点，则丢弃目标 IPv6 地址为 IMDS `[fd00:ec2::254]` 的数据包。

限制 Linux 实例的 IMDS 访问

使用 iptables 限制访问

以下示例使用 Linux iptables 及其 `owner` 模块禁止 Apache Web 服务器（基于其默认安装用户 ID `apache`）访问 169.254.169.254。它使用拒绝规则 拒绝来自以该用户身份运行的任何进程的所有实例元数据请求（无论是 IMDSv1 还是 IMDSv2）。

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner
 --uid-owner apache --jump REJECT
```

或者，您可以考虑使用允许规则 以仅允许特定用户或组进行访问。从安全的角度看，允许规则可能更易于管理，因为它们要求您决定哪种软件需要访问实例元数据。如果使用允许规则，在您以后更改实例上的软件或配置时，不太可能会意外允许软件访问元数据服务（您不打算让该软件进行访问）。您还可以将组与允许规则结合使用，以便您可以在允许的组中添加和删除用户，而无需更改防火墙规则。

以下示例禁止所有进程访问 IMDS，但在用户账户 `trustworthy-user` 中运行的进程除外。

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match
 owner ! --uid-owner trustworthy-user --jump REJECT
```

> **Note**
>
> - 要使用本地防火墙规则，您需要修改前面的示例命令以符合您的需求。
> - 默认情况下，不会在系统重新引导之间持久保留 iptables 规则。可以使用此处未介绍的操作系统功能持久保留这些规则。
> - 只有在组是给定本地用户的主要组时，iptables `owner` 模块才会匹配组成员资格。不会匹配其他组。

使用 PF 或 IPFW 限制访问

如果使用 FreeBSD 或 OpenBSD，则还可以考虑使用 PF 或 IPFW。以下示例将 IMDS 访问限制为仅根用户。

PF

```
$ block out inet proto tcp from any to 169.254.169.254
```

```
$ pass out inet proto tcp from any to 169.254.169.254 user root
```

IPFW

```
$ allow tcp from any to 169.254.169.254 uid root
```

```
$ deny tcp from any to 169.254.169.254
```

> **Note**
>
> PF 和 IPFW 命令的顺序至关重要。PF 默认为最后一个匹配规则，而 IPFW 默认为第一个匹配规则。

限制 Windows 实例的 IMDS 访问

使用 Windows 防火墙限制访问

以下 PowerShell 示例使用内置的 Windows 防火墙禁止 Internet Information Server Web 服务器（基于其默认安装用户 ID NT AUTHORITY\IUSR）访问 169.254.169.254。它使用拒绝规则 拒绝来自以该用户身份运行的任何进程的所有实例元数据请求（无论是 IMDSv1 还是 IMDSv2）。

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount ("NT
 AUTHORITY\IUSR")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $BlockPrincipalSDDL = "D:(A;;CC;;;$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service from IIS" -Action
 block -Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $BlockPrincipalSDDL
```

或者，您可以考虑使用允许规则 以仅允许特定用户或组进行访问。从安全的角度看，允许规则可能更易于管理，因为它们要求您决定哪种软件需要访问实例元数据。如果使用允许规则，在您以后更改实例上的软件或配置时，不太可能会意外允许软件访问元数据服务（您不打算让该软件进行访问）。您还可以将组与允许规则结合使用，以便您可以在允许的组中添加和删除用户，而无需更改防火墙规则。

以下示例禁止作为 `blockPrincipal` 变量中指定的操作系统组（在该示例中为 Windows 组 `Everyone`）运行的所有进程访问实例元数据，但在 `exceptionPrincipal` 中指定的进程除外（在该示例中为名为 `trustworthy-users` 的组）。您必须同时指定拒绝委托人和允许委托人，因为与 Linux iptables 中的 `！ --uid-owner trustworthy-user` 规则不同，Windows 防火墙不提供拒绝所有其他委托人以仅允许特定委托人（用户或组）的快捷方法。

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("Everyone")
PS C:\> $BlockPrincipalSID =
 $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $exceptionPrincipal = New-Object -TypeName System.Security.Principal.NTAccount
 ("trustworthy-users")
```

```
PS C:\> $ExceptionPrincipalSID =
 $exceptionPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $PrincipalSDDL = "O:LSD:(D;;CC;;;$ExceptionPrincipalSID)(A;;CC;;;
$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service for
 $($blockPrincipal.Value), exception: $($exceptionPrincipal.Value)" -Action block -
Direction out `
-Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $PrincipalSDDL
```

> **Note**
>
> 要使用本地防火墙规则，您需要修改前面的示例命令以符合您的需求。

使用 netsh 规则限制访问

您可以考虑使用 netsh 规则阻止所有软件，但这些规则不太灵活。

```
C:\> netsh advfirewall firewall add rule name="Block metadata service altogether"
 dir=out protocol=TCP remoteip=169.254.169.254 action=block
```

> **Note**
>
> - 要使用本地防火墙规则，您需要修改前面的示例命令以符合您的需求。
> - 必须从提升权限的命令提示符中设置 netsh 规则，并且不能将其设置为拒绝或允许特定委托人。

配置实例元数据服务选项

实例元数据服务（IMDS）在每个 EC2 实例上本地运行。实例元数据选项是一组配置，可用于控制 EC2 实例上 IMDS 的可访问性和行为。

可在每个实例上配置以下实例元数据选项：

实例元数据服务（IMDS）：enabled | disabled

可在实例上启用或禁用 IMDS。禁用后，您或任何代码都将无法访问实例上的实例元数据。

实例上的 IMDS 有两个端点：IPv4 (169.254.169.254) 和 IPv6 ([fd00:ec2::254])。启用 IMDS 时，会自动启用 IPv4 端点。若要启用 IPv6 端点，需将其显式启用。

IMDS IPv6 端点：enabled | disabled

可在实例上显式启用 IPv6 IMDS 端点。启用 IPv6 端点后，IPv4 端点将保持启用状态。IPv6 端点仅在支持 IPv6 的子网（双栈或仅 IPv6）中基于 Nitro 的实例上受支持。

元数据版本：IMDSv1 or IMDSv2 (token optional) | IMDSv2 only (token required)

请求实例元数据时，IMDSv2 调用需要令牌。IMDSv1 调用不需要令牌。可将实例配置为允许调用 IMDSv1 或 IMDSv2（可以使用令牌），或者仅允许调用 IMDSv2（必须使用令牌）。

元数据响应跃点限制：1 – 64

跃点限制是允许进行 PUT 响应的网络跃点数。可以设置跃点限制，最小值为 1，最大值为 64。在容器环境中，跃点限制 1 可能会导致问题。有关如何缓解这些问题的信息，请参阅实例元数据访问注意事项下方有关容器环境的信息。

访问实例元数据中的标签：enabled | disabled

可以启用或禁用从实例元数据访问实例标签。有关更多信息，请参阅 使用实例元数据来查看 EC2 实例的标签。

要查看实例的当前配置，请参阅 查询现有实例的实例元数据选项。

在何处配置实例元数据选项

可以在不同级别配置实例元数据选项，如下所示：

- 账户：可以在账户级别为每个 Amazon Web Services 区域 设置实例元数据选项的默认值。启动实例后，实例元数据选项会自动设置为账户级别的值。您可以在启动后更改这些值。账户级别的默认值不会影响现有实例。
- AMI：注册或修改 AMI 时，可将 imds-support 参数设置为 v2.0。使用此 AMI 启动实例后，实例元数据版本会自动设置为 IMDSv2，并且跃点限制会设置为 2。
- 实例：可以在启动实例时，更改实例上的所有实例元数据选项，从而覆盖默认设置。也可以在正在运行或已停止的实例上启动后，再更改实例元数据选项。请注意，更改可能受到 IAM 或 SCP 策略的限制。

有关更多信息，请参阅为新实例配置实例元数据选项和为现有实例修改实例元数据选项。

实例元数据选项的优先顺序

每个实例元数据选项的值都会在启动实例时按照优先级分层顺序确定。分层顺序如下，按最高优先级从上至下依次排序：

- 优先级 1：启动时的实例配置：可以在启动模板或实例配置中指定值。此处指定的任何值都会覆盖在账户级别或 AMI 中指定的值。
- 优先级 2：账户设置：如果启动实例时未指定值，则该值将根据账户级别的设置（针对每个 Amazon Web Services 区域 进行设置）确定。账户级别设置通常会包含各个元数据选项的值，或者不会指示任何首选项。
- 优先级 3：AMI 配置：如果启动实例时未指定值或账户级别设置未确定值，则该值将根据 AMI 配置确定。这适用于 `HttpTokens` 和 `HttpPutResponseHopLimit`。

每个元数据选项都将分开评估。可以通过组合使用直接实例配置、账户级别默认设置和 AMI 配置，配置实例。

在正在运行或已停止的实例上启动后，可以更改任何元数据选项的值，除非更改受到 IAM 或 SCP 策略的限制。

> **Note**
>
> 首先需要根据优先顺序确定好实例的 IMDS 设置之后，才会评估账户级别 IMDSv2 强制使用设置。启用 IMDSv2 强制使用后，使用 IMDSv1 启用的实例将会失败。有关强制使用的更多信息，请参阅在账户级别强制使用 IMDSv2。

> **Warning**
>
> 如果启用了 IMDSv2 强制使用，但未在启动时的实例配置中或在账户设置或 AMI 配置中将 `httpTokens` 设置为 `required`，则启动将会失败。

示例 1 – 确定元数据选项的值

在此示例中，EC2 实例会启动到一个区域中，其中 `HttpPutResponseHopLimit` 在账户级别设置为 `1`。指定 AMI 会将 `ImdsSupport` 设置为 `v2.0`。启动时不会直接在实例上指定元数据选项。该实例通过以下元数据选项启动：

```
"MetadataOptions": {
    ...
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    ...
```

这些值如下所示确定：

- 启动时未指定元数据选项：实例启动期间，实例启动参数或启动模板中均未提供元数据选项的特定值。
- 账户设置优先级次之：如果启动时未指定特定值，则会优先应用区域内的账户级别设置。这意味着将应用账户级别配置的默认值。在本例中，HttpPutResponseHopLimit 设置为 1。
- AMI 设置优先级最低：如果启动时或账户级别未指定 HttpTokens（实例元数据版本）的特定值，则应用 AMI 设置。在本例中，由 AMI 设置 ImdsSupport: v2.0 确定将 HttpTokens 设置为 required。请注意，虽然 AMI 设置 ImdsSupport: v2.0 旨在设置 HttpPutResponseHopLimit: 2，但其被账户级别设置 HttpPutResponseHopLimit: 1 覆盖，而后者的优先级更高。

示例 2 – 确定元数据选项的值

在此示例中，使用与前面示例 1 中相同的设置启动 EC2 实例，不过启动时直接在实例上将 HttpTokens 设置为 optional。该实例通过以下元数据选项启动：

```
"MetadataOptions": {
    ...
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    ...
```

HttpPutResponseHopLimit 的值的确定方式与示例 1 中相同。但是，HttpTokens 的值按以下方式确定：启动时在实例上配置的元数据选项优先。即使 AMI 配置了 ImdsSupport: v2.0（也就是 HttpTokens 设置为 required），启动时在实例上指定的值（HttpTokens 设置为 optional）仍优先。

示例 3 – 在启用 HttpTokensEnforced 的情况下确定元数据选项的值

在此示例中，该账户在该区域中的设置为 HttpTokens = required 和 HttpTokensEnforced = enabled。

考虑以下 EC2 实例启动尝试：

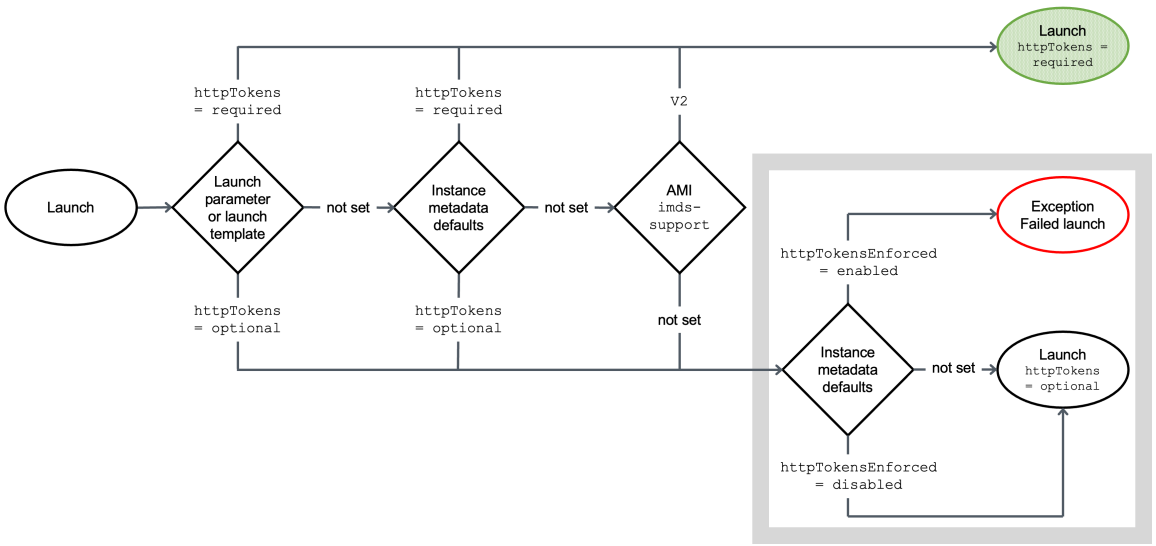
- 在将 HttpTokens 设置为 optional 的情况下尝试启动：启动将会失败，因为启用了账户级别强制使用 (HttpTokensEnforced = enabled)，并且启动参数的优先级高于账户默认设置。
- 在将 HttpTokens 设置为 required 的情况下尝试启动：启动将会成功，因为它符合账户级别强制使用的要求。
- 在未指定 HttpTokens 值的情况下尝试启动：启动将会成功，因为根据账户设置，该值默认为 required。

设置实例元数据版本

实例启动时，实例元数据版本的值为 IMDSv1 或 IMDSv2（令牌可选）(httpTokens=optional) 或仅限 IMDSv2（需要令牌）(**httpTokens=required**)。

实例启动时，您可以手动指定元数据版本的值，也可以使用默认值。如果手动指定该值，则将覆盖所有默认值。如果不手动指定该值，则该值将由默认设置的组合确定。

以下流程图显示了不同配置级别的设置如何确定实例启动时的元数据版本以及强制使用的评估位置。下表列举了每个级别的具体设置。



该表显示了不同配置级别的设置如何确定启动实例时的元数据版本（由第 4 列中生成的实例配置指示）。优先级顺序如下所示，从左到右，第一列优先级最高：

- 第 1 列：启动参数：表示启动时手动指定的实例设置。
- 第 2 列：账户级别默认设置：表示账户设置。
- 第 3 列：AMI 默认设置：表示 AMI 设置。

启动参数	默认账户级别	默认 AMI	生成的实例配置
仅 V2（需要令牌）	无首选项	仅 V2	仅 V2
仅 V2（需要令牌）	仅 V2	仅 V2	仅 V2
仅 V2（需要令牌）	V1 或 V2	仅 V2	仅 V2
V1 或 V2（可以使用令牌）	无首选项	仅 V2	V1 或 V2
V1 或 V2（可以使用令牌）	仅 V2	仅 V2	V1 或 V2
V1 或 V2（可以使用令牌）	V1 或 V2	仅 V2	V1 或 V2
未设置	无首选项	仅 V2	仅 V2
未设置	仅 V2	仅 V2	仅 V2
未设置	V1 或 V2	仅 V2	V1 或 V2
仅 V2（需要令牌）	无首选项	null	仅 V2
仅 V2（需要令牌）	仅 V2	null	仅 V2
仅 V2（需要令牌）	V1 或 V2	null	仅 V2
V1 或 V2（可以使用令牌）	无首选项	null	V1 或 V2
V1 或 V2（可以使用令牌）	仅 V2	null	V1 或 V2
V1 或 V2（可以使用令牌）	V1 或 V2	null	V1 或 V2
未设置	无首选项	null	V1 或 V2
未设置	仅 V2	null	仅 V2

启动参数	默认账户级别	默认 AMI	生成的实例配置
未设置	V1 或 V2	null	V1 或 V2

使用 IAM 条件键限制实例元数据选项

可以在 IAM 策略或 SCP 中使用 IAM 条件键，如下所示：

- 仅在实例配置为要求使用 IMDSv2 时允许实例启动
- 限制允许的跃点数
- 关闭对实例元数据的访问

任务

- 为新实例配置实例元数据选项
- 为现有实例修改实例元数据选项

> **Note**
>
> 在进行任何更改之前，您应谨慎执行操作并进行仔细的测试。记录以下内容：
>
> - 如果您强制使用 IMDSv2，则使用 IMDSv1 访问实例元数据的应用程序或代理将会中断。
> - 如果禁用对实例元数据的所有访问，则依赖于实例元数据访问才能正常工作的应用程序或代理将会中断。
> - 对于 IMDSv2，在检索令牌时必须使用 `/latest/api/token`。
> - （仅限 Windows）如果您的 PowerShell 版本早于 4.0，则必须更新到 Windows Management Framework 4.0 才能要求使用 IMDSv2。

为新实例配置实例元数据选项

可为新实例配置以下实例元数据选项。

选项

- 要求使用 IMDSv2
- 启用 IMDS IPv4 和 IPv6 端点

- 关闭对实例元数据的访问
- 允许访问实例元数据中的标签

> **Note**
>
> 这些选项的设置在账户级别配置，可以直接在账户中配置，也可以使用声明式策略进行配置。必须在要配置实例元数据选项的每个 Amazon Web Services 区域中对选项进行配置。使用声明式策略允许同时将设置应用于多个区域，也可以同时应用于多个账户。当使用声明式策略时，您无法直接在账户中修改设置。本主题介绍如何直接在账户中配置设置。有关使用声明式策略的信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

要求使用 IMDSv2

可以使用以下方法要求在新实例上使用 IMDSv2。

要求 IMDSv2

- 将 IMDSv2 设置为账户默认设置
- 在账户级别强制使用 IMDSv2
- 在启动时配置实例
- 配置 AMI
- 使用 IAM policy

将 IMDSv2 设置为账户默认设置

可以在账户级别为每个 Amazon Web Services 区域 设置实例元数据服务（IMDS）的默认版本。这意味着在启动新实例后，实例元数据版本会自动设置为该账户级别的默认值。但是，您可以在启动实例时或启动实例后手动覆盖该值。有关账户级别设置和手动覆盖会如何影响实例的更多信息，请参阅 实例元数据选项的优先顺序。

> **Note**
>
> 设置账户级别的默认值不会重置现有的实例。例如，假设您将账户级别的默认值设置为 IMDSv2，则任何设置为 IMDSv1 的现有实例都不会受到影响。如果要更改现有实例的值，则必须手动更改相关实例本身的值。

您可以将实例元数据版本的账户默认值设置为 IMDSv2，这样账户中的所有新实例启动时都会需要 IMDSv2，而 IMDSv1 将被禁用。如果应用此账户默认设置，以下是该实例启动时的默认值：

- 控制台：元数据版本设置为仅 V2（需要令牌），并且元数据响应跃点限制设置为 2。
- Amazon CLI：`HttpTokens` 设置为 `required`，并且 `HttpPutResponseHopLimit` 设置为 2。

> **Note**
>
> 在将账户默认值设置为 IMDSv2 之前，请确保您的实例不依赖 IMDSv1。有关更多信息，请参阅 要求 IMDSv2 的建议途径。

Console

将 IMDSv2 设置为指定区域内的账户默认设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择控制面板。
4. 在账户属性卡的设置下，选择数据保护和安全。
5. 选择 IMDS 默认设置旁边的管理。
6. 在管理 IMDS 默认设置页面上，执行以下操作：
   a. 对于实例元数据服务，选择启用。
   b. 对于 Metadata version（元数据版本），选择 V2 only (token required)（仅限 V2（需要令牌））。
   c. 对于元数据响应跃点限制，如果实例将托管容器，请指定 2。否则，请选择无首选项。如果未指定首选项，则在启动时，如果 AMI 具有设置 `ImdsSupport: v2.0`，则该值默认为 2；否则默认为 1。
   d. 选择更新。

Amazon CLI

将 IMDSv2 设置为指定区域内的账户默认设置

使用 modify-instance-metadata-defaults 命令，并指定要在其中修改 IMDS 账户级别设置的区域。如果实例将托管容器，则将 --http-tokens 设置为 required，并将 --http-put-response-hop-limit 设置为 2。否则，请指定 -1，表示无首选项。如果已指定 -1（无首选项），则在启动时，如果 AMI 具有设置 ImdsSupport: v2.0，则该值默认为 2；否则默认为 1。

```
aws ec2 modify-instance-metadata-defaults \
    --region us-east-1 \
    --http-tokens required \
    --http-put-response-hop-limit 2
```

下面是示例输出。

```
{
    "Return": true
}
```

查看指定区域的实例元数据选项的默认账户设置

使用 get-instance-metadata-defaults 命令并指定区域。

```
aws ec2 get-instance-metadata-defaults --region us-east-1
```

下面是示例输出。

```
{
    "AccountLevel": {
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 2
    },
    "ManagedBy": "account"
}
```

ManagedBy 字段表示配置了该设置的实体。在本例中，account 表示是直接在账户中配置的设置。值为 declarative-policy 表示设置是由声明式策略所配置。有关更多信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

将 IMDSv2 设置为所有区域的账户默认设置

使用 modify-instance-metadata-defaults 命令修改所有区域的 IMDS 账户级别设置。如果实例将托管容器，则将 --http-tokens 设置为 required，并将 --http-put-response-hop-limit

设置为 2。否则，请指定 -1，表示无首选项。如果已指定 -1（无首选项），则在启动时，如果 AMI 具有设置 ImdsSupport: v2.0，则该值默认为 2；否则默认为 1。

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
            --http-tokens required \
            --http-put-response-hop-limit 2 \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

下面是示例输出。

```
Region                  Modified
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
...
```

查看所有区域实例元数据选项的默认账户设置

使用 get-instance-metadata-defaults 命令。

```
echo -e "Region   \t Level          Hops    HttpTokens" ; \
echo -e "-------------- \t ------------   ----    ----------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
```

```
    do (output=$(
        aws ec2 get-instance-metadata-defaults \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

下面是示例输出。

```
Region           Level          Hops    HttpTokens
--------------   ------------   ----    ----------
ap-south-1       ACCOUNTLEVEL   2       required
eu-north-1       ACCOUNTLEVEL   2       required
eu-west-3        ACCOUNTLEVEL   2       required
...
```

PowerShell

将 IMDSv2 设置为指定区域内的账户默认设置

使用 Edit-EC2InstanceMetadataDefault cmdlet，并指定要在其中修改 IMDS 账户级别设置的区域。如果实例将托管容器，则将 -HttpToken 设置为 required，并将 -HttpPutResponseHopLimit 设置为 2。否则，请指定 -1，表示无首选项。如果已指定 -1（无首选项），则在启动时，如果 AMI 具有设置 ImdsSupport: v2.0，则该值默认为 2；否则默认为 1。

```
Edit-EC2InstanceMetadataDefault `
    -Region us-east-1 `
    -HttpToken required `
    -HttpPutResponseHopLimit 2
```

下面是示例输出。

```
True
```

查看指定区域的实例元数据选项的默认账户设置

使用 Get-EC2InstanceMetadataDefault cmdlet 并指定区域。

```
Get-EC2InstanceMetadataDefault -Region us-east-1 | Format-List
```

下面是示例输出。

```
HttpEndpoint            :
HttpPutResponseHopLimit : 2
HttpTokens              : required
InstanceMetadataTags    :
```

将 IMDSv2 设置为所有区域的账户默认设置

使用 Edit-EC2InstanceMetadataDefault cmdlet 修改所有区域的 IMDS 账户级别设置。如果实例将托管容器，则将 -HttpToken 设置为 required，并将 -HttpPutResponseHopLimit 设置为 2。否则，请指定 -1，表示无首选项。如果已指定 -1（无首选项），则在启动时，如果 AMI 具有设置 ImdsSupport: v2.0，则该值默认为 2；否则默认为 1。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region   = $_
        Modified = (Edit-EC2InstanceMetadataDefault `
                -Region $_ `
                -HttpToken required `
                -HttpPutResponseHopLimit 2)
    }
} | `
Format-Table Region, Modified -AutoSize
```

预期输出

```
Region         Modified
------         --------
ap-south-1         True
eu-north-1         True
eu-west-3          True
...
```

查看所有区域实例元数据选项的默认账户设置

使用 Get-EC2InstanceMetadataDefault cmdlet。

```
(Get-EC2Region).RegionName | `
```

```
    ForEach-Object {
    [PSCustomObject]@{
        Region = $_
        HttpPutResponseHopLimit = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpPutResponseHopLimit
        HttpTokens              = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpTokens
    }
} | `
Format-Table -AutoSize
```

输出示例

```
Region         HttpPutResponseHopLimit HttpTokens
------         ----------------------- ----------
ap-south-1                           2 required
eu-north-1                           2 required
eu-west-3                            2 required
...
```

在账户级别强制使用 IMDSv2

可以在账户级别为每个 Amazon Web Services 区域强制要求使用 IMDSv2。强制使用后，实例只有配置为必须使用 IMDSv2 时才能启动。无论实例配置或 AMI 配置如何，都必须遵守此强制使用要求。

> **Note**
>
> 在账户级别启用 IMDSv2 强制使用之前，请确保您的应用程序和 AMI 支持 IMDSv2。有关更多信息，请参阅 要求 IMDSv2 的建议途径。如果启用了 IMDSv2 强制使用，但未在启动时的实例配置中或在账户设置或 AMI 配置中将 `httpTokens` 设置为 `required`，则实例启动将会失败。有关问题排查信息，请参阅启动启用 IMDSv1 的实例时失败。

> **Note**
>
> 此设置不会更改现有实例的 IMDS 版本，但会阻止在当前已禁用 IMDSv1 的现有实例上启用 IMDSv1。

Console

在指定区域中为账户强制使用 IMDSv2

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择控制面板。
4. 在账户属性卡的设置下，选择数据保护和安全。
5. 选择 IMDS 默认设置旁边的管理。
6. 在管理 IMDS 默认设置页面上，执行以下操作：
   a. 对于 Metadata version（元数据版本），选择 V2 only (token required)（仅限 V2（需要令牌））。
   b. 对于强制使用 IMDSv2，请选择已启用。
   c. 选择更新。

Amazon CLI

在指定区域中为账户强制使用 IMDSv2

使用 modify-instance-metadata-defaults 命令，并指定要强制使用 IMDSv2 的区域。

```
aws ec2 modify-instance-metadata-defaults \
    --region us-east-1 \
    --http-tokens required \
    --http-tokens-enforced enabled
```

下面是示例输出。

```
{
"Return": true
}
```

查看指定区域中的账户 IMDSv2 强制使用设置

使用 get-instance-metadata-defaults 命令并指定区域。

```
aws ec2 get-instance-metadata-defaults --region us-east-1
```

下面是示例输出。

```
{
    "AccountLevel": {
        "HttpTokens": "required",
        "HttpTokensEnforced": "enabled"
    },
    "ManagedBy": "account"
}
```

ManagedBy 字段表示配置了该设置的实体。在本例中，account 表示是直接在账户中配置的设置。值为 declarative-policy 表示设置是由声明式策略所配置。有关更多信息，请参阅《Amazon 用户指南》中的 Declarative policies。

在所有区域为账户强制使用 IMDSv2

使用 modify-instance-metadata-defaults 命令，以在所有区域强制使用 IMDSv2。

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
            --http-tokens-enforced enabled \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

下面是示例输出。

```
Region                  Modified
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
```

```
...
```

查看所有区域中的账户 IMDSv2 强制使用设置

使用 get-instance-metadata-defaults 命令。

```
echo -e "Region   \t Level           HttpTokensEnforced" ; \
echo -e "-------------- \t ------------   ----------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-instance-metadata-defaults \
            --region $region \
            --query 'AccountLevel.HttpTokensEnforced' \
            --output text)
        echo -e "$region \t ACCOUNTLEVEL $output"
    );
done
```

下面是示例输出。

```
Region           Level          HttpTokensEnforced
--------------   ------------   ------------------
ap-south-1       ACCOUNTLEVEL   enabled
eu-north-1       ACCOUNTLEVEL   enabled
eu-west-3        ACCOUNTLEVEL   enabled
...
```

PowerShell

在指定区域中为账户强制使用 IMDSv2

使用 Edit-EC2InstanceMetadataDefault cmdlet，并指定要强制使用 IMDSv2 的区域。

```
Edit-EC2InstanceMetadataDefault `
    -Region us-east-1 `
    -HttpToken required `
    -HttpPutResponseHopLimit 2
```

下面是示例输出。

```
@{
    Return = $true
}
```

查看指定区域中的账户 IMDSv2 强制使用设置

使用 Get-EC2InstanceMetadataDefault 命令并指定区域。

```
Get-EC2InstanceMetadataDefault -Region us-east-1
```

下面是示例输出。

```
@{
    AccountLevel = @{
        HttpTokens = "required"
        HttpTokensEnforced = "enabled"
    }
    ManagedBy = "account"
}
```

ManagedBy 字段表示配置了该设置的实体。在本例中，account 表示是直接在账户中配置的设置。值为 declarative-policy 表示设置是由声明式策略所配置。有关更多信息，请参阅《Amazon 用户指南》中的 Declarative policies。

在所有区域为账户强制使用 IMDSv2

使用 modify-instance-metadata-defaults 命令，以在所有区域强制使用 IMDSv2。

```
echo -e "Region          \t Modified" ; \
echo -e "--------------  \t ---------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 modify-instance-metadata-defaults \
            --region $region \
```

```
            --http-tokens-enforced enabled \
            --output text)
        echo -e "$region        \t $output"
    );
done
```

下面是示例输出。

```
Region                  Modified
--------------          ---------
ap-south-1              True
eu-north-1              True
eu-west-3               True
...
```

将 IMDSv2 设置为所有区域的账户默认设置

使用 Edit-EC2InstanceMetadataDefault cmdlet 修改所有区域的 IMDS 账户级别设置。如果实例将托管容器，则将 -HttpToken 设置为 required，并将 -HttpPutResponseHopLimit 设置为 2。否则，请指定 -1，表示无首选项。如果已指定 -1（无首选项），则在启动时，如果 AMI 具有设置 ImdsSupport: v2.0，则该值默认为 2；否则默认为 1。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region   = $_
        Modified = (Edit-EC2InstanceMetadataDefault `
                -Region $_ `
                -HttpToken required `
                -HttpPutResponseHopLimit 2)
    }
} | `
Format-Table Region, Modified -AutoSize
```

预期输出

```
Region         Modified
------         --------
ap-south-1         True
eu-north-1         True
eu-west-3          True
```

```
...
```

查看所有区域实例元数据选项的默认账户设置

使用 Get-EC2InstanceMetadataDefault cmdlet。

```
(Get-EC2Region).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region = $_
        HttpPutResponseHopLimit = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpPutResponseHopLimit
        HttpTokens              = (Get-EC2InstanceMetadataDefault -Region
 $_).HttpTokens
    }
} | `
Format-Table -AutoSize
```

输出示例

```
Region         HttpPutResponseHopLimit HttpTokens
------         ----------------------- ----------
ap-south-1                           2 required
eu-north-1                           2 required
eu-west-3                            2 required
...
```

在启动时配置实例

当您 启动实例 时，您可以通过配置以下字段，从而将实例配置为要求使用 IMDSv2：

- Amazon EC2 控制台：将 Metadata version（元数据版本）设置为 V2 only (token required) [仅 V2（必须使用令牌）]。
- Amazon CLI：将 `HttpTokens` 设置为 `required`。

当您指定必须使用 IMDSv2 时，还必须将可访问元数据设置为已启用（控制台），或将 `HttpEndpoint` 设置为 `enabled`（Amazon CLI），以启用实例元数据服务（IMDS）端点。

在容器环境中，需要 IMDSv2 时，我们建议将跃点限制设置为 2。有关更多信息，请参阅 实例元数据访问注意事项。

Console

要求在新实例上使用 IMDSv2

- 在 Amazon EC2 控制台中启动新实例时，展开 Advanced details（高级详细信息），然后执行以下操作：
  - 对于 Metadata accessible（可访问的元数据），选择 Enabled（已启用）。
  - 对于 Metadata version（元数据版本），选择 V2 only (token required)（仅限 V2（需要令牌））。
  - （容器环境）对于元数据响应跃点限制，请选择 2。

  有关更多信息，请参阅 高级详细信息。

Amazon CLI

要求在新实例上使用 IMDSv2

以下 run-instances 示例会启动 `c6i.large` 设置为 `--metadata-options` 的 `HttpTokens=required` 实例。在为 `HttpTokens` 指定值时，还必须将 `HttpEndpoint` 设置为 `enabled`。由于元数据检索请求的安全令牌标头设置为 `required`，这会在请求实例元数据时要求实例使用 IMDSv2。

在容器环境中，需要 IMDSv2 时，我们建议使用 `HttpPutResponseHopLimit=2` 将跃点限制设置为 2。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
 ...
    --metadata-options
 "HttpEndpoint=enabled,HttpTokens=required,HttpPutResponseHopLimit=2"
```

PowerShell

要求在新实例上使用 IMDSv2

以下 New-EC2Instance cmdlet 示例启动了一个 `c6i.large` 实例，并将 `MetadataOptions_HttpEndpoint` 设置为 `enabled`，将 `MetadataOptions_HttpTokens` 参数设置为 `required`。在为 `HttpTokens` 指定值时，还必须将 `HttpEndpoint` 设置为

enabled。由于元数据检索请求的安全令牌标头设置为 required，这会在请求实例元数据时要求实例使用 IMDSv2。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpTokens required
```

Amazon CloudFormation

要使用 Amazon CloudFormation 指定实例的元数据选项，请参阅 Amazon CloudFormation 用户指南中的 亚马逊云科技::EC2::LaunchTemplate MetadataOptions 属性。

配置 AMI

注册新 AMI 或修改现有 AMI 时，可将 imds-support 参数设置为 v2.0。从此 AMI 启动的实例会将元数据版本设置为 仅 V2（必须使用令牌）（控制台），或将 HttpTokens 设置为 required（Amazon CLI）。借助这些设置，实例将要求在请求实例元数据时使用 IMDSv2。

请注意，如果您将 imds-support 设置为 v2.0，从此 AMI 启动的实例还会将 Metadata response hop limit（元数据响应跃点限制）（控制台）或 http-put-response-hop-limit（Amazon CLI）将设置为 2。

> **Important**
>
> 除非 AMI 软件支持 IMDSv2，否则不要使用该参数。将值设置为 v2.0 后，无法撤消。“重置”AMI 的唯一方法是从底层快照创建新的 AMI。

为 IMDSv2 配置新的 AMI

使用以下方法之一将新 AMI 配置为 IMDSv2。

Amazon CLI

下面的 register-image 示例将使用某个 EBS 根卷的指定快照的 AMI 注册为设备 /dev/xvda。为 imds-support 参数指定 v2.0，以便从此 AMI 启动的实例要求在请求实例元数据时使用 IMDSv2。

```
aws ec2 register-image \
```

```
    --name my-image \
    --root-device-name /dev/xvda \
    --block-device-mappings DeviceName=/dev/
xvda,Ebs={SnapshotId=snap-0123456789example} \
    --architecture x86_64 \
    --imds-support v2.0
```

PowerShell

以下 Register-EC2Image cmdlet 示例将 EBS 根卷的指定快照作为设备 /dev/xvda 来注册了一个 AMI。为 ImdsSupport 参数指定 v2.0，以便从此 AMI 启动的实例要求在请求实例元数据时使用 IMDSv2。

```
Register-EC2Image `
    -Name 'my-image' `
    -RootDeviceName /dev/xvda `
    -BlockDeviceMapping  (
    New-Object `
        -TypeName Amazon.EC2.Model.BlockDeviceMapping `
        -Property @{
        DeviceName = '/dev/xvda';
        EBS        = (New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice -Property
 @{
                SnapshotId = 'snap-0123456789example'
                VolumeType = 'gp3'
                } )
        }  ) `
    -Architecture X86_64 `
    -ImdsSupport v2.0
```

为 IMDSv2 配置现有 AMI

使用以下方法之一配置 IMDSv2 的现有 AMI。

Amazon CLI

以下 modify-image-attribute 示例仅为 IMDSv2 修改现有 AMI。为 imds-support 参数指定 v2.0，以便从此 AMI 启动的实例要求在请求实例元数据时使用 IMDSv2。

```
aws ec2 modify-image-attribute \
    --image-id ami-0abcdef1234567890 \
```

```
    --imds-support v2.0
```

PowerShell

以下 Edit-EC2ImageAttribute cmdlet 示例仅修改了 IMDSv2 的一个现有 AMI。为 `imds-support` 参数指定 `v2.0`，以便从此 AMI 启动的实例要求在请求实例元数据时使用 IMDSv2。

```
Edit-EC2ImageAttribute `
    -ImageId ami-0abcdef1234567890 `
    -ImdsSupport 'v2.0'
```

使用 IAM policy

您可以创建一个将执行以下操作之一的 IAM 策略：

- 除非要求必须在新实例上使用 IMDSv2，否则阻止用户启动新实例。
- 阻止用户调用 ModifyInstanceMetadataOptions API 来更改正在运行的实例的元数据选项。限制对 ModifyInstanceMetadataOptions httpTokens 属性的访问，以防止对正在运行的实例进行意外的更新。
- 阻止用户调用 ModifyInstanceMetadataDefaults API 来更改 httpTokens 和 httpTokensEnforced 的账户默认设置。限制对这两个属性的访问，将确保只有获得授权的角色才能修改账户默认值。

使用 IAM policy 强制要求在所有新实例上使用 IMDSv2

要确保用户只能启动要求在请求实例元数据时使用 IMDSv2 的实例，请执行以下操作：

- 限制对 `ModifyInstanceMetadataOptions` 和 `ModifyInstanceMetadataDefaults` API 的访问，更具体而言是限制对 `httpTokens` 和 `httpTokensEnforced` 属性的访问。
- 然后将账户默认值设置为 `httpTokens = required` 和 `httpTokensEnforced = enabled`。

  有关示例 IAM policy，请参阅 使用实例元数据。

启用 IMDS IPv4 和 IPv6 端点

实例上的 IMDS 有两个端点：IPv4 (`169.254.169.254`) 和 IPv6 (`[fd00:ec2::254]`)。启用 IMDS 时，会自动启用 IPv4 端点。即使将实例启动到仅限 IPv6 的子网中，IPv6 端点仍处于禁用状态。要启用 IPv6 端点，需将其显式启用。启用 IPv6 端点后，IPv4 端点将保持启用状态。

您可以在启动实例时或启动实例之后启用 IPv6 端点。

启用 IPv6 端点的要求

- 所选的实例类型是基于 Nitro 的实例。
- 选定的子网支持 IPv6，其中子网要么为双堆栈，要么仅限 IPv6。

使用以下方法之一启动启用了 IMDS IPv6 端点的实例。

Console

在启动实例时启用 IMDS IPv6 端点

- 在 Amazon EC2 控制台中 启动实例，并在 Advanced details（高级详细信息）下指定以下参数：
  - 对于元数据 IPv6 端点，选择启用。

有关更多信息，请参阅 高级详细信息。

Amazon CLI

在启动实例时启用 IMDS IPv6 端点

下方 run-instances 示例会启动一个为 IMDS 启用了 IPv6 端点的 c6i.large 实例。要启用 IPv6 端点，请将 --metadata-options 参数指定为 HttpProtocolIpv6=enabled。在为 HttpProtocolIpv6 指定值时，还必须将 HttpEndpoint 设置为 enabled。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=enabled,HttpProtocolIpv6=enabled"
```

PowerShell

在启动实例时启用 IMDS IPv6 端点

以下 New-EC2Instance cmdlet 示例启动了一个为 IMDS 启用了 IPv6 端点的 c6i.large 实例。要启用 IPv6 端点，请将 MetadataOptions_HttpProtocolIpv6 指定为 enabled。在为 MetadataOptions_HttpProtocolIpv6 指定值时，还必须将 MetadataOptions_HttpEndpoint 设置为 enabled。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint enabled `
    -MetadataOptions_HttpProtocolIpv6 enabled
```

关闭对实例元数据的访问

您可以通过在启动实例时禁用 IMDS 来关闭对实例元数据的访问。您可以稍后通过重新启用 IMDS 来开启访问权限。有关更多信息，请参阅 开启对实例元数据的访问权限。

> **Important**
>
> 您可以选择在启动时或启动后禁用 IMDS。如果您在启动时禁用 IMDS，则可能发生以下情况：
>
> - 您可能不再拥有对实例的 SSH 访问权限。`public-keys/0/openssh-key` 作为您的实例的 SSH 公钥，将无法访问，因为该密钥通常是从 EC2 实例元数据提供和访问的。
> - EC2 用户数据将不可用，也不会在实例启动时运行。EC2 用户数据托管在 IMDS 上。如果您禁用 IMDS，则实际上关闭了对用户数据的访问。
>
> 要访问此功能，您可以在启动后重新启用 IMDS。

Console

要在启动时关闭对实例元数据的访问

- 在 Amazon EC2 控制台中 启动实例，并在 Advanced details（高级详细信息）下指定以下参数：
  - 对于 Metadata accessible（可访问的元数据），选择 Disabled（已禁用）。

有关更多信息，请参阅 高级详细信息。

Amazon CLI

要在启动时关闭对实例元数据的访问

启动实例并将 `--metadata-options` 设置为 `HttpEndpoint=disabled`。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type c6i.large \
    ...
    --metadata-options "HttpEndpoint=disabled"
```

PowerShell

要在启动时关闭对实例元数据的访问

以下 New-EC2Instance cmdlet 示例启动了将 `MetadataOptions_HttpEndpoint` 设置为 `disabled` 的实例。

```
New-EC2Instance `
    -ImageId ami-0abcdef1234567890 `
    -InstanceType c6i.large `
    -MetadataOptions_HttpEndpoint disabled
```

Amazon CloudFormation

要使用 Amazon CloudFormation 指定实例的元数据选项，请参阅 Amazon CloudFormation 用户指南中的 亚马逊云科技::EC2::LaunchTemplate MetadataOptions 属性。

允许访问实例元数据中的标签

默认情况下，无法访问实例元数据中的实例标签。对于每个实例，您必须明确允许访问。如果允许访问，则实例标签键必须符合特定的字符限制，否则实例启动将失败。有关更多信息，请参阅 启用对实例元数据中标签的访问权限。

为现有实例修改实例元数据选项

您可以修改现有实例的实例元数据选项。

您还可以创建 IAM policy，以阻止用户修改现有实例上的实例元数据选项。要控制哪些用户可以修改实例元数据选项，请指定一个策略，阻止除具有指定角色的用户以外的所有用户使用 ModifyInstanceMetadataOptions API。有关示例 IAM policy，请参阅 使用实例元数据。

> **Note**
>
> 如果使用了声明式策略来配置实例元数据选项，则无法在账户中直接对选项进行修改。有关更多信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

要求使用 IMDSv2

使用以下方法之一修改现有实例上的实例元数据选项，以要求在请求实例元数据时使用 IMDSv2。如果 IMDSv2 是必需的，则无法使用 IMDSv1。

> **Note**
>
> 在要求必须使用 IMDSv2 之前，请确保该实例没有进行 IMDSv1 调用。`MetadataNoToken` CloudWatch 指标会跟踪 IMDSv1 调用。当 `MetadataNoToken` 记录某个实例的 IMDSv1 使用量为零时，就可以要求该实例必须使用 IMDSv2。

Console

要求在现有实例上使用 IMDSv2

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例。
4. 依次选择操作、实例设置、修改实例元数据选项。
5. 在修改实例元数据选项对话框中，执行以下操作：
   a. 对于实例元数据服务，选择启用。
   b. 对于IMDSv2，选择必需。
   c. 选择保存。

Amazon CLI

要求在现有实例上使用 IMDSv2

请使用 modify-instance-metadata-options CLI 命令，并将 `http-tokens` 参数设置为 `required`。在为 `http-tokens` 指定值时，还必须将 `http-endpoint` 设置为 `enabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens required \
    --http-endpoint enabled
```

PowerShell

要求在现有实例上使用 IMDSv2

使用 Edit-EC2InstanceMetadataOption cmdlet 并将 HttpTokens 参数设置为 required。在为 HttpTokens 指定值时，还必须将 HttpEndpoint 设置为 enabled。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens required `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

恢复使用 imdsv1

在实例上要求必须使用 IMDSv2 时，使用 IMDSv1 请求将会失败。如果 IMDSv2 是可选的，则 IMDSv2 和 IMDSv1 都将正常工作。因此，要恢复使用 IMDSv1，请使用以下方法之一将 IMDSv2 设为可选 (httpTokens = optional)。

httpTokensEnforced IMDS 属性还可以防止尝试在现有实例上启用 IMDSv1。在某个区域为账户启用此功能后，尝试将 httpTokens 设置为 optional 将导致 UnsupportedOperation 异常。有关更多信息，请参阅 问题排查。

Important

如果实例因强制使用 IMDSv2 而启动失败，则可通过两种方法成功启动：

- 将实例启动为仅限 IMDSv2：如果在实例上运行的软件仅使用 IMDSv2（不依赖 IMDSv1），则可以将实例启动为仅限 IMDSv2。要执行此操作，请通过在启动参数或账户在该区域的元数据默认值中设置 httpTokens = required，从而配置仅限 IMDSv2。
- 禁用强制使用：如果您的软件仍依赖于 IMDSv1，则在该区域将账户的 httpTokensEnforced 设置为 disabled。有关更多信息，请参阅 在账户级别强制使用 IMDSv2。

Console

恢复在实例上使用 IMDSv1

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例。
4. 依次选择操作、实例设置、修改实例元数据选项。
5. 在修改实例元数据选项对话框中，执行以下操作：
   a. 对于实例元数据服务，请确保选择启用。
   b. 对于 IMDSv2，选择可选。
   c. 选择保存。

Amazon CLI

恢复在实例上使用 IMDSv1

您可以使用 modify-instance-metadata-options CLI 命令并将 `http-tokens` 设置为 `optional`，以在请求实例元数据时恢复使用 IMDSv1。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-tokens optional \
    --http-endpoint enabled
```

PowerShell

恢复在实例上使用 IMDSv1

您可以使用 Edit-EC2InstanceMetadataOption cmdlet 并将 `HttpTokens` 设置为 `optional`，以在请求实例元数据时恢复使用 IMDSv1。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpTokens optional `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

更改 PUT 响应跃点限制

对于现有的实例，您可以更改 PUT 响应跃点数限制设置。

目前仅 Amazon CLI 和 Amazon SDK 支持更改 PUT 响应跃点限制。

Amazon CLI

更改 PUT 响应跃点限制

请使用 modify-instance-metadata-options CLI 命令，并将 `http-put-response-hop-limit` 参数设置为所需的跃点数。在以下示例中，跃点数限制设置为 3。请注意，在为 `http-put-response-hop-limit` 指定值时，还必须将 `http-endpoint` 设置为 `enabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-put-response-hop-limit 3 \
    --http-endpoint enabled
```

PowerShell

更改 PUT 响应跃点限制

使用 Edit-EC2InstanceMetadataOption cmdlet 并将 `HttpPutResponseHopLimit` 参数设置为所需的跃点数。在以下示例中，跃点数限制设置为 3。请注意，在为 `HttpPutResponseHopLimit` 指定值时，还必须将 `HttpEndpoint` 设置为 `enabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpPutResponseHopLimit 3 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

启用 IMDS IPv4 和 IPv6 端点

实例上的 IMDS 有两个端点：IPv4 (`169.254.169.254`) 和 IPv6 (`[fd00:ec2::254]`)。启用 IMDS 时，会自动启用 IPv4 端点。即使将实例启动到仅限 IPv6 的子网中，IPv6 端点仍处于禁用状态。要启用 IPv6 端点，需将其显式启用。启用 IPv6 端点后，IPv4 端点将保持启用状态。

您可以在启动实例时或启动实例之后启用 IPv6 端点。

启用 IPv6 端点的要求

- 所选的实例类型是基于 Nitro 的实例。
- 选定的子网支持 IPv6，其中子网要么为双堆栈，要么仅限 IPv6。

目前，仅 Amazon CLI 和 Amazon SDK 支持在启动实例后启用 IMDS IPv6 端点。

Amazon CLI

为实例启用 IMDS IPv6 端点

请使用 modify-instance-metadata-options CLI 命令，并将 `http-protocol-ipv6` 参数设置为 `enabled`。请注意，在为 `http-protocol-ipv6` 指定值时，还必须将 `http-endpoint` 设置为 `enabled`。

```
aws ec2 modify-instance-metadata-options \
 --instance-id i-1234567890abcdef0 \
 --http-protocol-ipv6 enabled \
 --http-endpoint enabled
```

PowerShell

为实例启用 IMDS IPv6 端点

使用 Edit-EC2InstanceMetadataOption cmdlet 并将 `HttpProtocolIpv6` 参数设置为 `enabled`。请注意，在为 `HttpProtocolIpv6` 指定值时，还必须将 `HttpEndpoint` 设置为 `enabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpProtocolIpv6 enabled `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

开启对实例元数据的访问权限

您可以通过启用实例上 IMDS 的 HTTP 端点来开启对实例元数据的访问权限，无论您使用的是哪个版本的 IMDS。您可以随时通过禁用 HTTP 端点来撤消该更改。

使用以下方法之一，即可开启对实例上实例元数据的访问权限。

Console

开启对实例元数据的访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例。
4. 依次选择操作、实例设置、修改实例元数据选项。
5. 在修改实例元数据选项对话框中，执行以下操作：
   a. 对于实例元数据服务，选择启用。
   b. 选择保存。

Amazon CLI

开启对实例元数据的访问权限

请使用 modify-instance-metadata-options CLI 命令，并将 `http-endpoint` 参数设置为 `enabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint enabled
```

PowerShell

开启对实例元数据的访问权限

使用 Edit-EC2InstanceMetadataOption cmdlet 并将 `HttpEndpoint` 参数设置为 `enabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint enabled).InstanceMetadataOptions
```

关闭对实例元数据的访问

您可以通过禁用实例上 IMDS 的 HTTP 端点来关闭对实例元数据的访问权限，无论您使用的是哪个版本的 IMDS。您可以随时启用 HTTP 终端节点以撤消该更改。

使用以下方法之一，即可关闭对实例上实例元数据的访问权限。

Console

关闭对实例元数据的访问

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例。
4. 依次选择操作、实例设置、修改实例元数据选项。
5. 在修改实例元数据选项对话框中，执行以下操作：
   a. 对于实例元数据服务，清除启用。
   b. 选择保存。

Amazon CLI

关闭对实例元数据的访问

请使用 modify-instance-metadata-options CLI 命令，并将 `http-endpoint` 参数设置为 `disabled`。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --http-endpoint disabled
```

PowerShell

关闭对实例元数据的访问

使用 Edit-EC2InstanceMetadataOption cmdlet 并将 `HttpEndpoint` 参数设置为 `disabled`。

```
(Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -HttpEndpoint disabled).InstanceMetadataOptions
```

允许访问实例元数据中的标签

您可允许访问正在运行或已停止实例上的实例元数据中的标签。对于每个实例，您必须明确允许访问。如果允许访问，则实例标签键必须符合特定的字符限制，否则会出错。有关更多信息，请参阅 启用对实例元数据中标签的访问权限。

问题排查

修改启用 IMDSv1 的实例时失败

说明

您收到了以下错误消息：

```
You can't launch instances with IMDSv1 because httpTokensEnforced
is enabled for this account. Either launch the instance with
httpTokens=required or contact your account owner to disable
httpTokensEnforced using the ModifyInstanceMetadataDefaults API or the
account settings in the EC2 console.
```

原因

在 EC2 账户设置或 Amazon 组织声明性策略强制使用 IMDSv2 (`httpTokensEnforced = enabled`) 的账户中，尝试将现有实例修改为启用 IMDSv1 (`httpTokens = optional`) 时，将会引发此错误。

解决方案

如果需要在现有实例上支持 IMDSv1，则需要在该区域为该账户禁用 IMDSv2 强制使用。要禁用 IMDSv2 强制使用，请将 `HttpTokensEnforced` 设置为 `disabled`。有关更多信息，请参阅《Amazon EC2 API 参考》中的 ModifyInstanceMetadataDefaults。如果您更喜欢使用控制台来配置此设置，请参阅在账户级别强制使用 IMDSv2。

建议使用仅限 IMDSv2 (`httpTokens=required`)。有关更多信息，请参阅 转换为使用 实例元数据服务版本 2。

在启动包含用户数据输入的 EC2 实例时运行命令

启动 Amazon EC2 实例后，您可以将用户数据传递给实例，用于执行自动配置任务或在实例启动后运行脚本。

如果您对更复杂的自动化场景感兴趣，可以考虑使用 Amazon CloudFormation。有关更多信息，请参阅《Amazon CloudFormation 用户指南》中的使用 Amazon CloudFormation 在 Amazon EC2 上部署应用程序。

在 Linux 实例上，您可以将两类用户数据传递到 Amazon EC2：Shell 脚本和 cloud-init 指令。您还可以将此数据以纯文本、文件（这非常适合通过命令行工具启动实例）或者 base64 编码文本（用于 API 调用）的形式传递到启动实例向导中。

在 Windows 实例上，启动代理处理您的用户数据脚本。

注意事项

- 用户数据会被视为非透明数据；您提供什么数据您就会得到什么数据。由实例对其进行解释。
- 用户数据必须采用 base64 编码。Amazon EC2 控制台可以为您执行 base64 编码或接受 base64 编码的输入。如果您使用实例元数据或控制台检索用户数据，则会自动为您进行 base64 解码。
- 用户数据在进行 base64 编码之前的原始格式的大小限制为 16 KB。长度为 n 的字符串在进行 base64 编码之后的大小为 ceil(n/3)*4。
- 用户数据是一种实例属性。如果您从实例创建 AMI，则实例用户数据不包含在该 AMI 中。

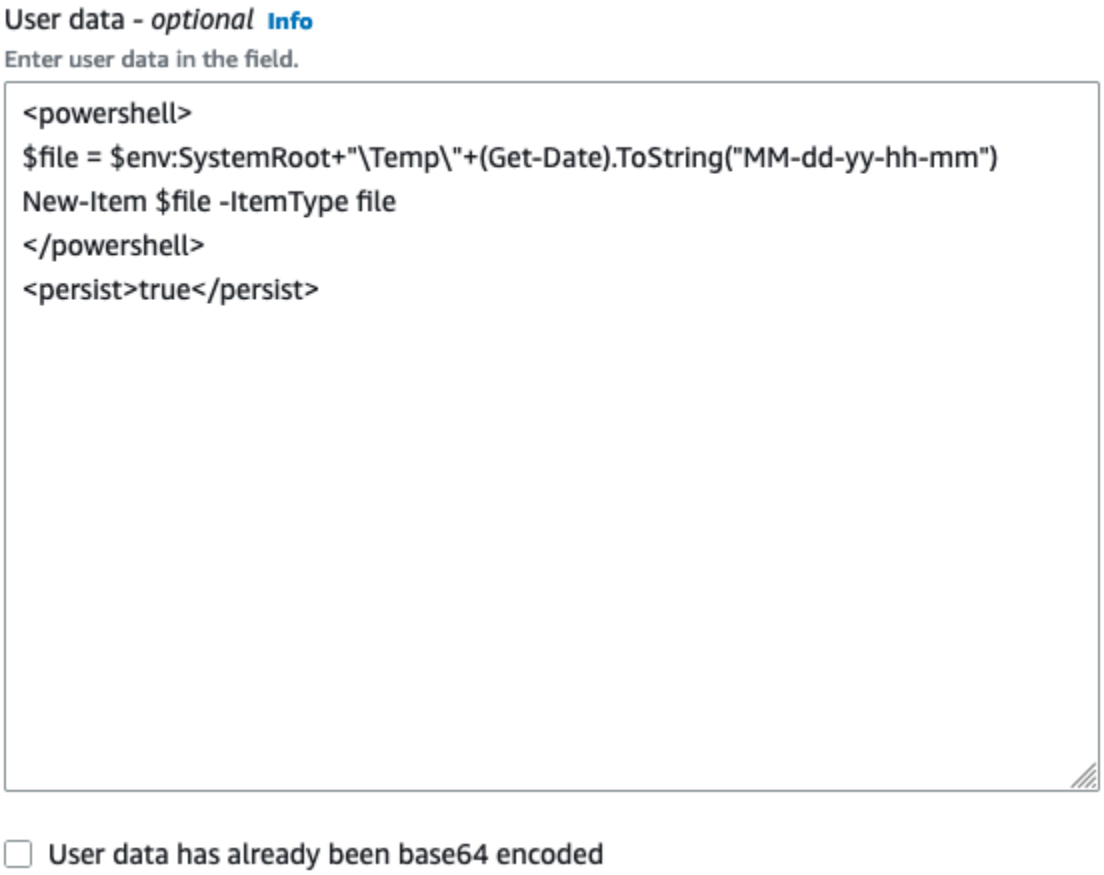
Amazon Web Services 管理控制台 中的用户数据

您可在启动实例时指定实例用户数据。如果实例的根卷是 EBS 卷，您还可以停止实例并更新其用户数据。

使用启动向导在启动时指定实例用户数据

您可以在 EC2 控制台中使用启动向导启动实例时指定用户数据。要在启动时指定用户数据，请按照启动实例的过程进行操作。User data（用户数据）字段位于启动实例向导的 高级详细信息 部分。在用户数据字段中输入您的 PowerShell 脚本，然后完成实例启动程序。

在用户数据字段的屏幕截图中，示例脚本在 Windows 临时文件夹中创建文件，在文件名中使用当前日期和时间。当您包括 `<persist>true</persist>` 时，每次重新引导或启动实例时将运行脚本。如果将用户数据已执行 base64 编码复选框保留为空，则 Amazon EC2 控制台将执行 base64 编码。



有关更多信息，请参阅 使用启动向导在启动时指定实例用户数据。有关使用 Amazon CLI 的 Linux 示例，请参阅 the section called “用户数据和 Amazon CLI”。有关使用 Tools for Windows PowerShell 的 Windows 示例，请参阅 the section called “用户数据和 Tools for Windows PowerShell”。

查看和更新实例用户数据

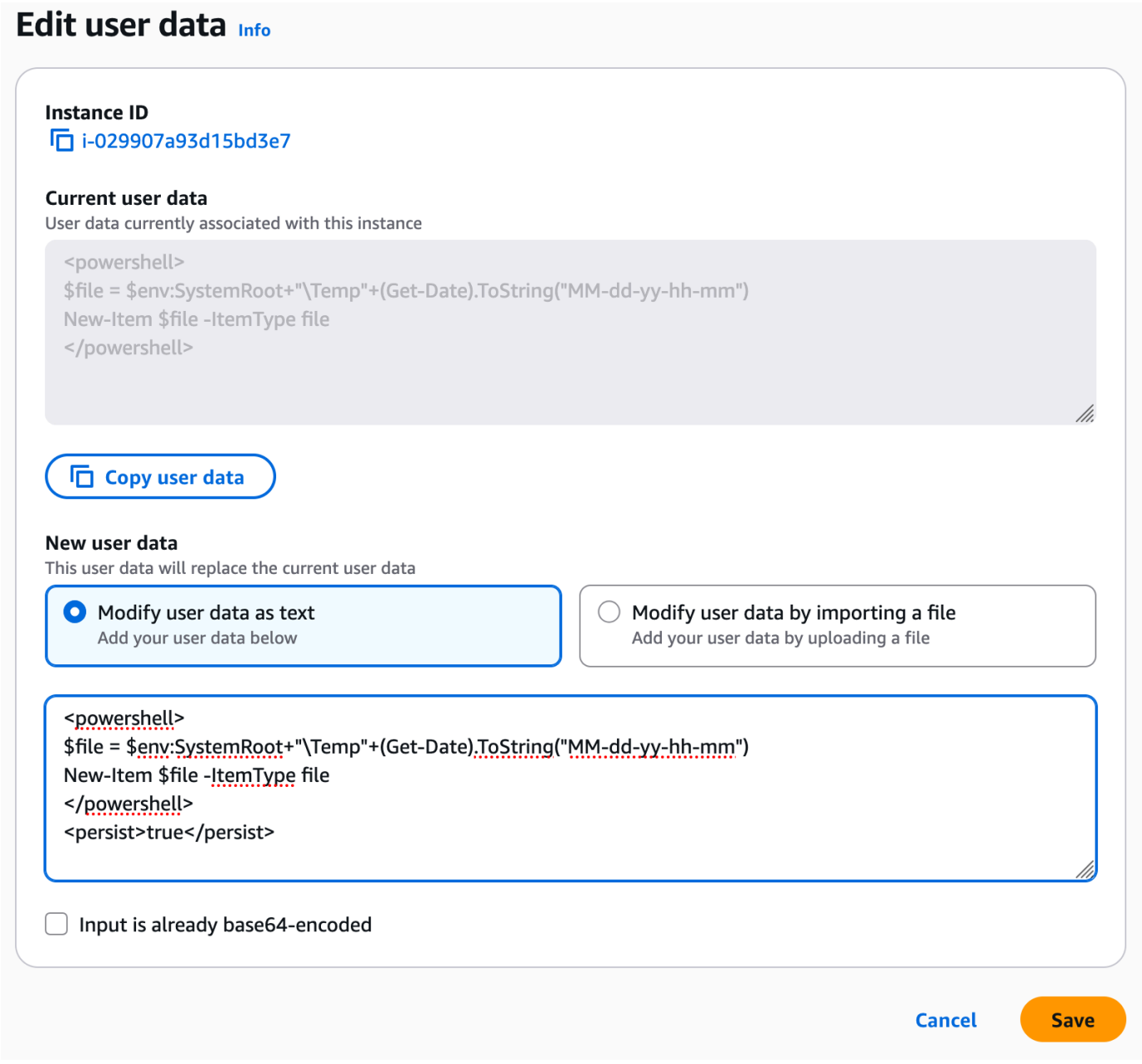
您可以查看任何实例的实例用户数据，也可以更新已停止实例的实例用户数据。

使用控制台更新实例的用户数据

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择所需实例，然后依次选择操作、实例状态、停止实例。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

4. 当系统提示您确认时，选择 Stop。停止实例可能需要几分钟时间。
5. 在实例仍被选中的情况下，依次选择操作、实例设置和编辑用户数据。如果实例正在运行，您不能更改用户数据，但是可以查看。
6. 在 Edit user data (编辑用户数据) 对话框中，更新用户数据，然后选择 Save (保存)。要在每次重新引导或启动实例时运行用户数据脚本，请添加 `<persist>true</persist>`，如下例中所示。



7. 启动实例。如果为后续重新引导或启动启用了用户数据运行，则在实例启动过程中执行更新后的用户数据脚本。

Amazon EC2 如何处理 Linux 实例的用户数据

以下示例使用用户数据在实例启动时运行设置 LAMP 服务器的命令。在每个示例中，执行以下任务：

- 更新发行版软件包。
- 安装 Web 服务器、php 和 `mariadb` 包。
- 启动并启用 `httpd` 服务。
- 用户 `ec2-user` 将添加到 apache 组。
- 为 Web 目录以及其中的文件设置适当的所有权和文件权限。
- 创建简单网页来测试 Web 服务器和 PHP 引擎。

内容

- 前提条件
- 用户数据和 Shell 脚本
- 更新实例用户数据
- 用户数据和 cloud-init 指令
- 用户数据和 Amazon CLI
- 将 Shell 脚本和 cloud-init 指令组合使用

前提条件

本主题中的示例假定以下内容：

- 您的实例具有可从互联网访问的公用 DNS 名称。
- 与您的实例关联的安全组将被配置为允许 SSH（端口 22）流量，以便您可以连接到该实例，以查看输出日志文件。
- 您的实例是使用 Amazon Linux AMI 启动的。这些命令和指令可能不适用于其他 Linux 发行版。有关其他发行版的更多信息（例如它们对 cloud-init 的支持），请参阅特定发行版的文档。

用户数据和 Shell 脚本

如果您熟悉 Shell 脚本编写，要在启动时将指令发送到实例，这是最简单、最完整的方式。在启动时添加这些任务会增加启动实例所需的时间。您应多等待几分钟让这些任务完成，然后测试用户脚本是否已成功完成。

> **Important**
>
> 默认情况下，用户数据脚本和 cloud-init 指令仅在首次启动实例时在引导周期内运行。您可以更新您的配置，以确保您的用户数据脚本和 cloud-init 指令在每次重新启动实例时都会运行。有关更多信息，请参阅Amazon知识中心中的如何利用用户数据在每次重新启动 Amazon EC2 Linux 实例时自动运行脚本？。

用户数据 Shell 脚本必须以 #! 字符以及指向要读取脚本的解释器的路径（通常为 /bin/bash）开头。有关 shell 脚本的介绍，请参阅 GNU 操作系统网站上的 Bash 参考手册。

作为用户数据输入的脚本是作为根用户加以运行的，因此在脚本中不使用 sudo 命令。请注意，您创建的任何文件都将归根用户所有；如果您需要非根用户具有文件访问权限，应在脚本中相应地修改权限。此外，这是因为脚本不交互运行，所以无法包含要求用户反馈的命令（如 yum update，无 -y 标志）。

如果您使用 Amazon API（包括 Amazon CLI），则在启动实例时必须使用实例配置文件。实例配置文件提供用户数据脚本发出 API 调用所需的适当 Amazon 凭证。有关更多信息，请参阅《IAM 用户指南》中的使用实例配置文件。您分配给 IAM 角色的权限取决于您使用 API 调用的服务。有关更多信息，请参阅 适用于 Amazon EC2 的 IAM 角色。

cloud-init 输出日志文件捕获控制台输出，因此，如果实例的行为不符合您的预期，可在启动后方便地调试脚本。要查看日志文件，请连接到实例并打开 `/var/log/cloud-init-output.log`。

在处理用户数据脚本时，该脚本将复制到 `/var/lib/cloud/instances/instance-id/` 目录并从该目录运行。脚本在运行后无法删除。请在从实例创建 AMI 之前务必删除 `/var/lib/cloud/instances/instance-id/` 中的用户数据脚本。否则，该脚本将存在于从 AMI 启动的任何实例上的此目录中。

更新实例用户数据

要更新实例用户数据，您必须先停止实例。如果实例正在运行，那么您可以查看用户数据，但不能进行修改。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

修改实例用户数据

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择所需实例，然后依次选择实例状态、停止实例。如果此选项处于禁用状态，则表示实例已停止，或者其根卷是实例存储卷。
4. 当系统提示您确认时，选择 Stop。停止实例可能需要几分钟时间。
5. 在实例仍被选中的情况下，依次选择操作、实例设置和编辑用户数据。
6. 根据需要修改用户数据，然后选择保存。
7. 开启实例。开启实例后，新的用户数据将在实例上可见；但不会运行用户数据脚本。

用户数据和 cloud-init 指令

cloud-init 程序包在启动时配置新 Amazon Linux 实例的特定方面；最值得注意的是，它为 ec2-user 配置 `.ssh/authorized_keys` 文件，以便您使用自己的私有密钥登录。有关 cloud-init 包为 Amazon Linux 实例执行的配置任务的更多信息，请参阅以下文档：

- Amazon Linux 2023 – 自定义 cloud-init
- Amazon Linux 2 – 在 Amazon Linux 2 上 cloud-init

可在启动时将 cloud-init 用户指令传递给实例，方式与传递脚本相同，只是语法不同。有关 cloud-init 的更多信息，请参阅 https://cloudinit.readthedocs.org/en/latest/index.html。

> **Important**
>
> 默认情况下，用户数据脚本和 cloud-init 指令仅在首次启动实例时在引导周期内运行。您可以更新您的配置，以确保您的用户数据脚本和 cloud-init 指令在每次重新启动实例时都会运行。有关更多信息，请参阅Amazon知识中心中的如何利用用户数据在每次重新启动 Amazon EC2 Linux 实例时自动运行脚本？。

在启动时添加这些任务会增加启动实例所需的时间。您应多等待几分钟让这些任务完成，然后测试用户数据指令是否已完成。

将 cloud-init 指令传递给 Amazon Linux 实例

1. 按照启动实例的程序进行操作。User data（用户数据）字段位于启动实例向导的 高级详细信息 部分。在 User data（用户数据）字段中输入您的 cloud-init 指令文本，然后完成实例启动程序。

   在下面的示例中，指令在 Amazon Linux 上创建并配置了一个 Web 服务器。要将命令标识为 cloud-init 指令，顶部的 `#cloud-config` 行是必需的。

   AL2023

   ```
   #cloud-config
   package_update: true
   package_upgrade: all

   packages:
   - httpd
   - mariadb105-server
   - php8.1
   - php8.1-mysqlnd

   runcmd:
   - systemctl start httpd
   - systemctl enable httpd
   - [ sh, -c, "usermod -a -G apache ec2-user" ]
   - [ sh, -c, "chown -R ec2-user:apache /var/www" ]
   - chmod 2775 /var/www
   - [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
   - [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
   - [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
   ```

   AL2

   ```
   #cloud-config
   package_update: true
   package_upgrade: all

   packages:
   - httpd
   - mariadb-server

   runcmd:
   - [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
   ```

```
- systemctl start httpd
- systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. 让实例有足够的时间启动和运行用户数据中的指令，然后查看指令是否完成了预期的任务。

   对于此示例，在 Web 浏览器中输入指令创建的 PHP 测试文件的 URL。此 URL 是实例的公用 DNS 地址，后接正斜杠和文件名。

   ```
   http://my.public.dns.amazonaws.com/phpinfo.php
   ```

   您应该会看到 PHP 信息页面。如果您无法看到 PHP 信息页，请检查所用的安全组是否包含允许 HTTP (端口 80) 通信的规则。有关更多信息，请参阅 配置安全组规则。

3. （可选）如果您的指令没有完成您期望的任务，或者如果您只是想验证您的指令是否已完成且没有错误，请连接到实例，检查输出日志文件 (`/var/log/cloud-init-output.log`)，然后查找输出中的错误消息。对于其他调试信息，您可以将以下行添加到指令：

   ```
   output : { all : '| tee -a /var/log/cloud-init-output.log' }
   ```

   此指令将 runcmd 输出发送到 `/var/log/cloud-init-output.log`。

用户数据和 Amazon CLI

您可以使用 Amazon CLI 指定、修改和查看实例的用户数据。有关使用实例元数据从实例查看用户数据的信息，请参阅访问 EC2 实例的实例元数据。

在 Windows 上，您可以使用 Amazon Tools for Windows PowerShell 而不是使用 Amazon CLI。有关更多信息，请参阅 用户数据和 Tools for Windows PowerShell。

示例：启动时指定用户数据

要在启动实例时指定用户数据，请结合使用 run-instances 命令与 `--user-data` 参数。使用 run-instances，Amazon CLI 将对您的用户数据执行 base64 编码。

以下示例显示如何在命令行上指定字符串形式的脚本：

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
    --key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234
 \
    --user-data echo user data
```

以下示例显示如何使用文本文件指定脚本。请务必使用 `file://` 前缀指定该文件。

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
    --key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234
 \
    --user-data file://my_script.txt
```

以下是具有 Shell 脚本的示例文本文件。

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

示例：修改停止的实例的用户数据

您可以使用 modify-instance-attribute 命令修改已停止的实例的用户数据。使用 modify-instance-attribute，Amazon CLI 不会对用户数据执行 base64 编码。

- 在 Linux 计算机上，使用 base64 命令对用户数据进行编码。

  ```
  base64 my_script.txt >my_script_base64.txt
  ```

- 在 Windows 计算机上，使用 certutil 命令可对用户数据进行编码。您必须先删除第一行 (BEGIN CERTIFICATE) 和最后一行 (END CERTIFICATE)，然后才能将此文件用于 Amazon CLI。

  ```
  certutil -encode my_script.txt my_script_base64.txt
  notepad my_script_base64.txt
  ```

使用 `--attribute` 和 `--value` 参数可通过编码的文本文件指定用户数据。请务必使用 `file://` 前缀指定该文件。

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData --value file://my_script_base64.txt
```

示例：清除停止的实例的用户数据

要删除现有的用户数据，请按以下方式使用 modify-instance-attribute 命令：

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --user-data Value=
```

示例：查看用户数据

要检索实例的用户数据，请使用 describe-instance-attribute 命令。使用 describe-instance-attribute，Amazon CLI 不会对用户数据执行 base64 解码。

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
 userData
```

以下是具有已进行 base64 编码的用户数据的示例输出。

```
{
    "UserData": {
        "Value":
 "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQpzZXJ2aWNlIGh0dHBkIHN0YXJ0CmNoa2NvbmZpZyBodHRwZCBvbg=="
    },
    "InstanceId": "i-1234567890abcdef0"
}
```

- 在 Linux 计算机上，使用 `--query` 选项获取已编码的用户数据和用于对该数据进行解码的 base64 命令。

  ```
  aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
   userData --output text --query "UserData.Value" | base64 --decode
  ```

- 在 Windows 计算机上，使用 `--query` 选项获取已编码的用户数据和用于对该数据进行解码的 certutil 命令。请注意，编码的输出存储在一个文件中，解码的输出存储在另一个文件中。

  ```
  aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
   userData --output text --query "UserData.Value" >my_output.txt
  certutil -decode my_output.txt my_output_decoded.txt
  type my_output_decoded.txt
  ```

下面是示例输出。

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

将 Shell 脚本和 cloud-init 指令组合使用

默认情况下，一次只能在用户数据中包含一个内容类型。不过，您可以在 mime-multi part 文件中使用 `text/cloud-config` 和 `text/x-shellscript`，以便在用户数据中同时包含 Shell 脚本和 cloud-init 指令。

下面显示了 mime-multi part 的格式。

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
cloud-init directives

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
shell script commands
--//--
```

例如，以下用户数据包含了 cloud-init 指令和 bash Shell 脚本。cloud-init 指令会创建一个文件（`/test-cloudinit/cloud-init.txt`），然后将 `Created by cloud-init` 写入该文件。bash Shell 脚本会创建一个文件（`/test-userscript/userscript.txt`），然后将 `Created by bash shell script` 写入该文件。

```
Content-Type: multipart/mixed; boundary="//"
```

```
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
runcmd:
- [ mkdir, /test-cloudinit ]
write_files:
- path: /test-cloudinit/cloud-init.txt
content: Created by cloud-init

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
mkdir test-userscript
touch /test-userscript/userscript.txt
echo "Created by bash shell script" >> /test-userscript/userscript.txt
--//--
```

Amazon EC2 如何处理 Windows 实例的用户数据

在 Windows 实例上，启动代理会执行与用户数据相关的任务。有关更多信息，请参阅下列内容：

- EC2Launch v2
- EC2Launch
- EC2Config 服务

有关 `UserData` 模板中 Amazon CloudFormation 属性的程序集示例，请参阅 Base64 编码 UserData 属性和带 AccessKey 和 SecretKey 的 Base64 编码 UserData 属性。

有关在使用生命周期挂钩的自动扩缩组中实例上运行命令的示例，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Tutorial: Configure user data to retrieve the target lifecycle state through instance metadata。

内容

- 用户数据脚本
- 压缩后的用户数据
- 用户数据执行
- 用户数据和 Tools for Windows PowerShell

用户数据脚本

要让 EC2Config 或 EC2Launch 运行脚本，在将脚本添加到用户数据中时，您必须将脚本封装在特殊标签中。您使用的标签取决于命令是在命令提示符窗口（批处理命令）中还是使用 Windows PowerShell 运行。

如果同时指定批处理脚本和 Windows PowerShell 脚本，则批处理脚本先运行，然后运行 Windows PowerShell 脚本，不论这些脚本在实例用户数据中的显示顺序如何。

如果您在用户数据脚本中使用 Amazon API（包括 Amazon CLI），则在启动实例时必须使用实例配置文件。实例配置文件提供用户数据脚本执行 API 调用所需的适当 Amazon 凭证。有关更多信息，请参阅实例配置文件。您分配给 IAM 角色的权限取决于您使用 API 调用的服务。有关更多信息，请参阅 适用于 Amazon EC2 的 IAM 角色。

脚本类型

- 批处理脚本的语法
- Windows PowerShell 脚本的语法
- YAML 配置脚本的语法
- Base64 编码

批处理脚本的语法

使用 script 标签指定批处理脚本。使用换行符分隔命令，如以下示例所示。

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
```

默认情况下，用户数据脚本在您启动实例时运行一次。要在每次重新引导或启动实例时运行用户数据脚本，请将 <persist>true</persist> 添加到用户数据。

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<persist>true</persist>
```

EC2Launch v2 代理

要在 EC2Launch v2 executeScript 任务处于 UserData 阶段时将 XML 用户数据脚本作为分离的进程运行，请向用户数据添加 <detach>true</detach>。

> **Note**
>
> 以前的启动代理不支持 detach 标签。

```
<script>
    echo Current date and time >> %SystemRoot%\Temp\test.log
    echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<detach>true</detach>
```

Windows PowerShell 脚本的语法

Amazon Windows AMI 包括 Amazon Tools for Windows PowerShell，因此您可在用户数据中指定这些 cmdlet。如果您将一个 IAM 角色与实例相关联，那么您不必为 cmdlet 指定凭证，因为实例上运行的应用程序可以使用该角色的凭证访问Amazon资源（例如 Amazon S3 存储桶）。

使用 <powershell> 标签指定 Windows PowerShell 脚本。使用换行符分隔命令。<powershell> 标签区分大小写。

例如：

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
```

默认情况下，用户数据脚本会在您启动实例时运行一次。要在每次重新引导或启动实例时运行用户数据脚本，请将 <persist>true</persist> 添加到用户数据。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

您可以使用 <powershellArguments> 标签来指定一个或多个 PowerShell 参数。如果未传递任何参数，则默认情况下，EC2Launch 和 EC2Launch v2 会添加以下参数：-ExecutionPolicy Unrestricted。

示例：

```
<powershell>
    $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

EC2Launch v2 代理

要在 EC2Launch v2 executeScript 任务处于 UserData 阶段时将 XML 用户数据脚本作为分离的进程运行，请向用户数据添加 <detach>true</detach>。

> **Note**
>
> 以前的启动代理不支持 detach 标签。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

YAML 配置脚本的语法

如果使用 EC2Launch v2 运行脚本，则可以使用 YAML 格式。要查看 EC2Launch v2 的配置任务、详细信息和示例，请参阅 EC2Launch v2 任务配置。

使用 executeScript 任务指定 YAML 脚本。

运行 PowerShell 脚本的 YAML 语法示例

```
version: 1.0
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
    content: |-
      $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
      New-Item $file -ItemType file
```

运行批处理脚本的 YAML 语法示例

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: batch
    runAs: localSystem
    content: |-
      echo Current date and time >> %SystemRoot%\Temp\test.log
      echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
```

Base64 编码

如果使用的是 Amazon EC2 API 或不对用户数据执行 base64 编码的工具，则您必须自行对用户数据进行编码。否则，系统会记录找不到要运行的 script 或 powershell 标签的错误。下面是使用 Windows PowerShell 进行编码的示例。

```
$UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

下面是一个使用 PowerShell 进行解码的示例。

```
$Script =
 [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData))
```

有关 base64 编码的更多信息，请参阅 https://www.ietf.org/rfc/rfc4648.txt。

压缩后的用户数据

EC2Launch v2 支持使用压缩用户数据的方法，来提交超过 16 KB 这一 IMDS 规定限制的用户数据。要使用此功能，请将用户数据脚本压缩为 `.zip` 存档，然后将其传递给 EC2 实例。检测到压缩用户数据时，EC2Launch v2 会自动将压缩后的用户数据脚本解压缩，然后再运行该脚本。

对于标准用户数据，如果使用的是 Amazon EC2 API 或不对用户数据执行 base64 编码的工具，则必须自行对压缩用户数据进行编码。有关用户数据大小限制和 base64 编码的更多信息，请参阅访问 EC2 实例的实例元数据。

用户数据执行

默认情况下，所有 Amazon Windows AMI 均已为初次启动启用了用户数据执行。您可以指定用户数据脚本在下次实例重新引导或重启时运行。此外，您可以指定用户数据脚本在每次实例重新引导或重启时运行。

> **Note**
>
> 默认情况下不会使用户数据在首次启动后运行。要使用户数据在实例重启或启动时运行，请参阅在后续重启或启动期间运行脚本。

生成随机密码之后，从本地管理员账户运行用户数据脚本。否则，从系统账户运行用户数据脚本。

实例启动脚本

实例用户数据中的脚本在实例的初次启动期间运行。如果找到 `persist` 标签，则为后续重新引导或启动启用用户数据执行。EC2Launch v2、EC2Launch 和 EC2Config 的日志文件包含源自标准输出和标准错误流的输出。

EC2Launch v2

EC2Launch v2 的日志文件为 `C:\ProgramData\Amazon\EC2Launch\log\agent.log`。

> **Note**
>
> `C:\ProgramData` 文件夹可能已隐藏。要查看该文件夹，您必须显示隐藏的文件和文件夹。

执行用户数据时将记录以下信息：

- Info: Converting user-data to yaml format – 如果用户数据是以 XML 格式提供的
- Info: Initialize user-data state – 用户数据执行开始
- Info: Frequency is: always – 如果每次启动时都运行用户数据任务
- Info: Frequency is: once – 如果用户数据任务只运行一次
- Stage: postReadyUserData execution completed – 用户数据执行结束

EC2Launch

EC2Launch 的日志文件为 C:\ProgramData\Amazon\EC2-Windows\Launch\Log\UserdataExecution.log。

C:\ProgramData 文件夹可能已隐藏。要查看该文件夹，您必须显示隐藏的文件和文件夹。

执行用户数据时将记录以下信息：

- Userdata execution begins – 用户数据执行开始
- <persist> tag was provided: true – 找到持久标签时
- Running userdata on every boot – 找到持久标签时
- <powershell> tag was provided.. running powershell content – 如果找到 PowerShell 标签
- <script> tag was provided.. running script content – 如果找到脚本标签
- Message: The output from user scripts – 如果运行用户数据脚本，则会记录其输出

EC2Config

EC2Config 的日志文件为 C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2Config.log。执行用户数据时将记录以下信息：

- Ec2HandleUserData: Message: Start running user scripts – 用户数据执行开始
- Ec2HandleUserData: Message: Re-enabled userdata execution – 找到持久标签时
- Ec2HandleUserData: Message: Could not find <persist> and </persist> – 如果未找到持久标签
- Ec2HandleUserData: Message: The output from user scripts – 如果运行用户数据脚本，则会记录其输出

在后续重启或启动期间运行脚本

在更新实例用户数据后，更新后的用户将在下次重启或启动实例时自动反映在实例元数据中。但根据安装的启动代理不同，可能需要额外的配置才能将用户数据脚本配置为在后续重启或启动时运行。

如果选择了使用 Sysprep 关闭选项，则用户数据脚本将在下一次实例启动或重启时运行，即使没有为后续重启或启动启用用户数据执行，也是如此。

有关启用用户数据执行的说明，请选择与启动代理相匹配的选项卡。

EC2Launch v2

与 EC2Launch v1 不同，EC2Launch v2 在每次启动时都会对用户数据任务进行评估。无需手动计划用户数据任务。用户数据基于包含的频率或持久性选项运行。

对于 XML 用户数据脚本

要在每次启动时运行用户数据脚本，请在用户数据中添加 `<persist>true</persist>` 标志。如果未包含持久性标志，则用户数据脚本仅在首次启动时运行。

对于 YAML 用户数据

- 要在首次启动时运行用户数据中的任务，请将任务 `frequency` 设置为 `once`。
- 要在每次启动时都运行用户数据中的任务，请将任务 `frequency` 设置为 `always`。

EC2Launch

1. 连接到您的 Windows 实例。
2. 打开 PowerShell 命令窗口，并运行以下命令之一：

   运行一次

   要在下次启动时运行一次用户数据，请使用 `-Schedule` 标志。

   ```
   C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
   Schedule
   ```

   在所有后续启动时运行

   要在所有后续启动时运行用户数据，请使用 `-SchedulePerBoot` 标志。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

3. 从 Windows 实例断开连接。要在下次实例启动时运行更新后的脚本，请停止实例并更新用户数据。

EC2Config

1. 连接到您的 Windows 实例。
2. 打开 `C:\Program Files\Amazon\Ec2ConfigService\Ec2ConfigServiceSetting.exe`。
3. 对于用户数据，选择 Enable UserData execution for next service start (为下次服务启动启用用户数据执行)。
4. 从 Windows 实例断开连接。要在下次实例启动时运行更新后的脚本，请停止实例并更新用户数据。

用户数据和 Tools for Windows PowerShell

您可以使用 Tools for Windows PowerShell 指定、修改和查看实例的用户数据。有关使用实例元数据从实例查看用户数据的信息，请参阅访问 EC2 实例的实例元数据。有关用户数据和 Amazon CLI 的信息，请参阅 用户数据和 Amazon CLI。

示例：启动时指定实例用户数据

使用实例用户数据创建一个文本文件。要在每次重新引导或启动实例时运行用户数据脚本，请添加 `<persist>true</persist>`，如下例中所示。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

要在启动实例时指定实例用户数据，请使用 New-EC2Instance 命令。该命令不会对用户数据进行 base64 编码。使用以下命令在名为 `script.txt` 的文本文件中对用户数据进行编码。

```
PS C:\> $Script = Get-Content -Raw script.txt
```

```
PS C:\> $UserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

使用 -UserData 参数可将用户数据传递到 New-EC2Instance 命令。

```
PS C:\> New-EC2Instance -ImageId ami-abcd1234 -MinCount 1 -MaxCount 1 -
InstanceType m3.medium \
    -KeyName my-key-pair -SubnetId subnet-12345678 -SecurityGroupIds sg-1a2b3c4d \
    -UserData $UserData
```

示例：更新已停止实例的实例用户数据

您可以使用 Edit-EC2InstanceAttribute 命令修改已停止的实例的用户数据。

使用新脚本创建一个文本文件。使用以下命令在名为 new-script.txt 的文本文件中对用户数据进行编码。

```
PS C:\> $NewScript = Get-Content -Raw new-script.txt
PS C:\> $NewUserData =
 [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($NewScript))
```

使用 -UserData 和 -Value 参数可指定用户数据。

```
PS C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute userData -
Value $NewUserData
```

示例：查看实例用户数据

要检索实例的用户数据，请使用 Get-EC2InstanceAttribute 命令。

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute
 userData).UserData
```

下面是示例输出。请注意，用户数据已编码。

```
PHBvd2Vyc2hlbGw
+DQpSZW5hbWUtQ29tcHV0ZXIgLU5ld05hbWUgdXNlci1kYXRhLXRlc3QNCjwvcG93ZXJzaGVsbD4=
```

使用以下命令可将已编码的用户数据存储在变量中，然后对其进行编码。

```
PS C:\> $UserData_encoded = (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -
Attribute userData).UserData
PS C:
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData_encoded)
```

下面是示例输出。

```
<powershell>
    $file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
    New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

示例：重命名实例以匹配标签值

要读取标签值、在首次启动时重命名实例以匹配标签值并重新启动，您可以使用 Get-EC2Tag 命令。要成功运行此命令，您必须将具有 ec2:DescribeTags 权限的角色附加到实例上，因为标签信息通过 API 调用检索。有关使用 IAM 角色的设置权限的更多信息，请参阅 将 IAM 角色附加到实例。

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
    $instanceId = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" =
 $token} -Method GET -Uri 'http://169.254.169.254/latest/meta-data/instance-id' -
UseBasicParsing
 $nameValue = (Get-EC2Tag -Filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

IMDSv1

```
<powershell>
 $instanceId = (Invoke-WebRequest http://169.254.169.254/latest/meta-data/instance-
id -UseBasicParsing).content
 $nameValue = (Get-EC2Tag -Filter @{Name="resource-id";Value=
$instanceid},@{Name="key";Value="Name"}).Value
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

您还可以使用实例元数据中的标签对实例进行重命名，前提是您的实例配置为从实例元数据访问标签。有关更多信息，请参阅 使用实例元数据来查看 EC2 实例的标签。

IMDSv2

```
<powershell>
    [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -
UseBasicParsing
 $nameValue = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}
 -Method GET -Uri 'http://169.254.169.254/latest/meta-data/tags/instance/Name' -
UseBasicParsing
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
```

```
    {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

IMDSv1

```
<powershell>
 $nameValue = Get-EC2InstanceMetadata -Path /tags/instance/Name
 $pattern = "^(?![0-9]{1,15}$)[a-zA-Z0-9-]{1,15}$"
 #Verify Name Value satisfies best practices for Windows hostnames
 If ($nameValue -match $pattern)
     {Try
         {Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
     Catch
         {$ErrorMessage = $_.Exception.Message
         Write-Output "Rename failed: $ErrorMessage"}}
 Else
     {Throw "Provided name not a valid hostname. Please ensure Name value is between
 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>
```

识别在单个请求中启动的每个实例

本示例演示如何使用用户数据和实例元数据来配置 Amazon EC2 实例。

> **Note**
>
> 本部分中的示例使用 IMDS 的 IPv4 地址：`169.254.169.254`。如果要通过 IPv6 地址检索 EC2 实例的实例元数据，请确保启用并改用 IPv6 地址：`[fd00:ec2::254]`。IMDS 的 IPv6 地址与 IMDSv2 命令兼容。IPv6 地址仅可在支持 IPv6 的子网（双栈或仅 IPv6）中基于 Nitro 的实例上访问。

Alice 想要启动她最喜欢的数据库 AMI 的四个实例，第一个实例用作原始实例，其余三个用作副本。当她启动它们时，她想为每个副本添加有关复制策略的用户数据。她知道这些数据将对所有四个实例都可用，因此她所采用的用户数据构建方式必须能够让每个实例识别出哪些部分适用于自己。她可通过 `ami-launch-index` 实例元数据值来实现这一点，该值对每个实例都是唯一的。如果她同时启动多个实例，则 `ami-launch-index` 表示实例启动的顺序。第一个启动的实例的值是 `0`。

以下是 Alice 构建的用户数据。

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

`replicate-every=1min` 数据定义第一个副本的配置，`replicate-every=5min` 定义第二个副本的配置，以此类推。Alice 决定以 ASCII 字符串形式提供这些数据，用竖线符号 (|) 来分隔每个实例的数据。

Alice 使用 run-instances 命令启动 4 个实例，并指定用户数据。

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 4 \
    --instance-type t2.micro \
    --user-data "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

实例启动之后，所有实例都有以下用户数据和常用元数据的副本：

- AMI ID：ami-0abcdef1234567890
- 预留 ID：r-1234567890abcabc0
- 公有密钥：无
- 安全组名称：默认值
- 实例类型：t2.micro

但是，每个实例都有唯一的元数据，如下表所示。

元数据	值
instance-id	i-1234567890abcdef0
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

元数据	值
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

元数据	值
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

元数据	值
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226

元数据	值
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice 可以使用 ami-launch-index 值确定用户数据的哪个部分适用于特定实例。

1. 她连接到其中一个实例并检索该实例的 ami-launch-index，以确保该实例是副本之一：

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/meta-data/api/
   token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
   data/ami-launch-index
   2
   ```

   对于以下步骤，IMDSv2 请求使用前面的 IMDSv2 命令中存储的令牌，并假设令牌尚未过期。

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
   2
   ```

2. 她将 ami-launch-index 保存为一个变量。

   IMDSv2

   ```
   [ec2-user ~]$ ami_launch_index=`curl -H "X-aws-ec2-metadata-token: $TOKEN"
    http://169.254.169.254/latest/meta-data/ami-launch-index`
   ```

   IMDSv1

   ```
   [ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-
   launch-index`
   ```

3. 她将用户数据保存为一个变量。

IMDSv2

```
[ec2-user ~]$ user_data=`curl -H "X-aws-ec2-metadata-token: $TOKEN"
 http://169.254.169.254/latest/user-data`
```

IMDSv1

```
[ec2-user ~]$ user_data=`curl http://169.254.169.254/latest/user-data`
```

4. 最后，Alice 使用 cut 命令提取适用于该实例的用户数据部分。

IMDSv2

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

IMDSv1

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

检测主机是否为 EC2 实例

您可能需要了解您的应用程序或网站是否在 EC2 实例上运行，尤其是在您拥有混合计算环境的情况下。您可以使用以下选项之一确定您的应用程序或网站的主机是否为 EC2 实例。

选项

- 检查 实例身份文档
- 检查系统 UUID
- 检查系统虚拟机生成标识符

检查 实例身份文档

每个实例都有一个签名的实例身份文档，您可以通过加密方式验证该文档。您可以使用实例元数据服务（IMDS）查找这些文档。

有关更多信息，请参阅 实例身份文档。

检查系统 UUID

您可以获取系统 UID 并查看 EC2 的 UID 开头的八位字节（在 Linux 中，这可能是小写的 ec2）。此方法速度快，但可能不准确，因为不是 EC2 实例的系统也有很小的几率使用以这些字符开头的 UUID。此外，某些版本的 SMBIOS 使用小端序格式，在 UUID 的开头不包含 EC2。对于使用 SMBIOS 2.4 for Windows 的 EC2 实例，或除 Amazon Linux（具有自己的 SMBIOS 实现）之外的 Linux 发行版，可能就是这种情况。

Linux 示例：从 DMI 获取 UUID（仅限 HVM AMI）

通过以下命令使用桌面管理界面 (DMI) 获取 UUID：

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

在以下示例输出中，UUID 以“EC2”开头，表示该系统可能是 EC2 实例。

```
EC2E1916-9099-7CAF-FD21-012345ABCDEF
```

在以下示例输出中，UUID 以 little-endian 格式表示。

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

或者，对于在 Nitro 系统上构建的实例，您可以使用以下命令：

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

如果输出是实例 ID，如下面的示例输出所示，则说明系统是 EC2 实例：

```
i-0af01c0123456789a
```

Linux 示例：从管理程序获取 UUID（仅限 PV AMI）

使用以下命令从管理程序获取 UUID：

```
[ec2-user ~]$ cat /sys/hypervisor/uuid
```

在以下示例输出中，UUID 以“ec2”开头，表示该系统可能是 EC2 实例。

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

Windows 示例：使用 WMI 或 Windows PowerShell 获取 UUID

使用如下 Windows Management Instrumentation 命令行 (WMIC)：

```
wmic path win32_computersystemproduct get uuid
```

或者，如果您使用 Windows PowerShell，则可使用 Get-WmiObject cmdlet，如下所示：

```
PS C:\> Get-WmiObject -query "select uuid from Win32_ComputerSystemProduct" | Select
 UUID
```

在以下示例输出中，UUID 以“EC2”开头，表示该系统可能是 EC2 实例。

```
EC2AE145-D1DC-13B2-94ED-012345ABCDEF
```

对于使用 SMBIOS 2.4 的实例，可能用 little-endian 格式表示 UUID，例如：

```
45E12AEC-DCD1-B213-94ED-012345ABCDEF
```

检查系统虚拟机生成标识符

虚拟机生成标识符由被解释为加密随机整数标识符的 128 位的唯一缓冲区组成。您可以检索虚拟机生成标识符来识别 Amazon Elastic Compute Cloud 实例。生成标识符通过 ACPI 表条目在实例的来宾操作系统中公开。如果您的计算机被克隆、复制或导入到 Amazon，例如使用 VM Import/Export，该值将改变。

示例：从 Linux 中检索虚拟机生成标识符

您可以使用以下命令从运行 Linux 的实例中检索虚拟机生成标识符。

Amazon Linux 2

1. 根据需要，使用以下命令更新现有软件包：

   ```
   sudo yum update
   ```

2. 如有必要，使用以下命令获取 busybox 软件包：

```
sudo curl https://www.rpmfind.net/linux/epel/next/8/Everything/x86_64/Packages/
b/busybox-1.35.0-2.el8.next.x86_64.rpm --output busybox.rpm
```

3. 如有必要，使用以下命令安装必备软件包：

```
sudo yum install busybox.rpm iasl -y
```

4. 运行以下 `iasl` 命令从 ACPI 表中生成输出：

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

5. 运行以下命令以查看 `iasl` 命令的输出：

```
cat SSDT2.dsl
```

输出应该产生检索虚拟机生成标识符所需的地址空间：

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
```

```
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
        })
    }
}
}
```

6. （可选）使用以下命令提升终端对剩余步骤的权限：

```
sudo -s
```

7. 使用以下命令存储以前收集的地址空间：

```
VMGN_ADDR=0xFED01000
```

8. 使用以下命令迭代地址空间并构建虚拟机生成标识符：

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

9. 使用以下命令从输出文件中检索虚拟机生成标识符：

```
cat vmgenid ; echo
```

您的输出应类似于以下内容：

```
EC2F335D979132C4165896753E72BD1C
```

Ubuntu

1. 根据需要，使用以下命令更新现有软件包：

```
sudo apt update
```

2. 如有必要，使用以下命令安装必备软件包：

```
sudo apt install busybox iasl -y
```

3. 运行以下 `iasl` 命令从 ACPI 表中生成输出：

```
sudo iasl -p ./SSDT2 -d /sys/firmware/acpi/tables/SSDT2
```

4. 运行以下命令以查看 `iasl` 命令的输出：

```
cat SSDT2.dsl
```

输出应该产生检索虚拟机生成标识符所需的地址空间：

```
Intel ACPI Component Architecture
ASL+ Optimizing Compiler/Disassembler version 20190509
Copyright (c) 2000 - 2019 Intel Corporation

File appears to be binary: found 32 non-ASCII characters, disassembling
Binary file appears to be a valid ACPI table, disassembling
Input file /sys/firmware/acpi/tables/SSDT2, Length 0x7B (123) bytes
```

```
ACPI: SSDT 0x0000000000000000 00007B (v01 AMAZON AMZNSSDT 00000001 AMZN
 00000001)
Pass 1 parse of [SSDT]
Pass 2 parse of [SSDT]
Parsing Deferred Opcodes (Methods/Buffers/Packages/Regions)

Parsing completed
Disassembly completed
ASL Output:    ./SSDT2.dsl - 1065 bytes
$
/*
* Intel ACPI Component Architecture
* AML/ASL+ Disassembler version 20190509 (64-bit version)
* Copyright (c) 2000 - 2019 Intel Corporation
*
* Disassembling to symbolic ASL+ operators
*
* Disassembly of /sys/firmware/acpi/tables/SSDT2, Tue Mar 29 16:15:14 2022
*
* Original Table Header:
*     Signature        "SSDT"
*     Length           0x0000007B (123)
*     Revision         0x01
*     Checksum         0xB8
*     OEM ID           "AMAZON"
*     OEM Table ID     "AMZNSSDT"
*     OEM Revision     0x00000001 (1)
*     Compiler ID      "AMZN"
*     Compiler Version 0x00000001 (1)
*/
DefinitionBlock ("", "SSDT", 1, "AMAZON", "AMZNSSDT", 0x00000001)
{
Scope (\_SB)
{
    Device (VMGN)
    {
        Name (_CID, "VM_Gen_Counter")  // _CID: Compatible ID
        Name (_DDN, "VM_Gen_Counter")  // _DDN: DOS Device Name
        Name (_HID, "AMZN0000")  // _HID: Hardware ID
        Name (ADDR, Package (0x02)
        {
            0xFED01000,
            Zero
        })
```

```
    }
}
}
```

5. （可选）使用以下命令提升终端对剩余步骤的权限：

```
sudo -s
```

6. 使用以下命令存储以前收集的地址空间：

```
VMGN_ADDR=0xFED01000
```

7. 使用以下命令迭代地址空间并构建虚拟机生成标识符：

```
for offset in 0x0 0x4 0x8 0xc; do busybox devmem $(($VMGN_ADDR + $offset)) | sed
 's/0x//' | sed -z '$ s/\n$//' >> vmgenid; done
```

8. 使用以下命令从输出文件中检索虚拟机生成标识符：

```
cat vmgenid ; echo
```

您的输出应类似于以下内容：

```
EC2F335D979132C4165896753E72BD1C
```

示例：从 Windows 中检索虚拟机生成标识符

您可以创建示例应用程序以从运行 Windows 的实例中检索虚拟机生成标识符。有关更多信息，请参阅 Microsoft 文档中的获取虚拟机生成标识符。

Amazon EC2 实例的实例身份文档

您启动的每个实例都有一个实例身份文档，用于提供有关实例自身的信息。您可以使用实例身份文档验证实例的属性。

实例身份文档在实例停止并启动、重新启动或启动时生成。您可以通过实例元数据服务（IMDS）访问实例的实例身份文档。有关说明，请参阅 检索实例身份文档。

实例身份文档使用纯文本 JSON 格式。其包含以下信息。

数据	说明
accountId	启动实例的 Amazon 账户的 ID。
architecture	启动实例所用 AMI 的架构 (i386 | x86_64 | arm64)。
availabilityZone	实例在其中运行的可用区的名称。例如 us-east-1 。请记住，可用区名称可能因 Amazon账户而异。
billingProducts	实例的计费产品。
devpayProductCodes	已淘汰。
imageId	用于启动实例的 AMI 的 ID。
instanceId	实例的 ID。
instanceType	实例的实例类型。
kernelId	与实例关联的内核的 ID（如果适用）。
marketplaceProductCodes	用于启动实例的 AMI 的 Amazon Web Services Marketplace 产品代码。
pendingTime	启动实例的日期和时间。
privateIp	实例的私有 IP 地址。对于仅限 IPv4 和双堆栈实例，此项包含 IPv4 地址。对于仅限 IPv6 实例，此项包含 IPv6 地址。
ramdiskId	与实例关联的 RAM 磁盘的 ID（如果适用）。
region	运行实例的区域。
version	实例身份文档格式的版本。

检索 EC2 实例的实例身份文档

Amazon EC2 实例的实例身份文档使用纯文本 JSON 格式。有关实例身份证件内容的描述，请参阅 the section called “实例身份文档”。

实例身份文件存储在实例的实例元数据中，属于 `instance-identity/document` 动态数据类别。您可以通过连接到实例并从实例元数据中进行检索来访问实例的实例身份文档。

您可以使用 IPv4 地址 169.254.169.254 或 IPv6 地址 fd00:ec2::254 访问实例元数据。这些是链路本地地址，表示只能从该实例进行访问。本页上的示例使用 IMDS 的 IPv4 地址：169.254.169.254。要通过 IPv6 检索 EC2 实例的实例元数据，请使用 fd00:ec2::254。

要在检索实例身份文档后验证其真实性，请参阅验证实例身份文档。

IMDSv2

Linux

在 Linux 实例上运行以下命令，来检索实例身份文档。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

在 Windows 实例上运行以下 cmdlet，来检索实例身份文档。

```
[string]$token = (Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token-ttl-seconds' = '21600'} `
    -Method PUT -Uri 'http://169.254.169.254/latest/api/token' -UseBasicParsing).Content
```

```
(Invoke-WebRequest -Headers @{'X-aws-ec2-metadata-token' = $token} `
    -Uri 'http://169.254.169.254/latest/dynamic/instance-identity/document' -UseBasicParsing).Content
```

IMDSv1

Linux

在 Linux 实例上运行以下命令，来检索实例身份文档。

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

Windows

在 Windows 实例上运行以下 cmdlet，来检索实例身份文档。

```
(Invoke-WebRequest http://169.254.169.254/latest/dynamic/instance-identity/
document).Content
```

下面是示例输出。

```
{
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
    "availabilityZone" : "us-west-2b",
    "privateIp" : "10.158.112.84",
    "version" : "2017-09-30",
    "instanceId" : "i-1234567890abcdef0",
    "billingProducts" : null,
    "instanceType" : "t2.micro",
    "accountId" : "123456789012",
    "imageId" : "ami-5fb8c835",
    "pendingTime" : "2016-11-19T16:32:11Z",
    "architecture" : "x86_64",
    "kernelId" : null,
    "ramdiskId" : null,
    "region" : "us-west-2"
}
```

验证 Amazon EC2 实例的实例身份文档

如果您打算将实例身份文档的内容用于重要用途，则应在使用前验证其内容和真实性。

明文实例身份文档附有三个经哈希处理的加密签名。您可以使用这些签名验证实例身份文档的来源和真实性以及其中包含的信息。其中提供以下签名：

- Base64 编码的签名 — 这是实例身份文档的 Base64 编码 SHA256 哈希值，使用 RSA 密钥对进行加密。
- PKCS7 签名 — 这是实例身份文档的 SHA1 哈希值，使用 DSA 密钥对进行加密。
- RSA-2048 签名 — 这是实例身份文档的 SHA256 哈希值，使用 RSA-2048 密钥对进行加密。

每个签名在实例元数据中的不同端点上可用。您可以根据哈希和加密要求使用这些签名中的任何一个。要验证签名，您必须使用对应的 Amazon 公有证书。

选项

- 选项 1：使用 PKCS7 签名验证实例身份文档
- 选项 2：使用 base64 编码的签名验证实例身份文档
- 选项 3：使用 RSA-2048 签名验证实例身份文档

选项 1：使用 PKCS7 签名验证实例身份文档

本主题说明如何使用 PKCS7 签名和 Amazon DSA 公有证书验证实例身份文档。

Linux 实例

使用 PKCS7 签名和 Amazon DSA 公有证书验证实例身份文档

1. 连接到实例。
2. 从实例元数据中检索 PKCS7 签名，并将其连同所需的页眉和页脚添加到名为 pkcs7 的新文件。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

   ```
   $ echo "-----BEGIN PKCS7-----" >> pkcs7 \
    && TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7 \
    && echo "" >> pkcs7 \
    && echo "-----END PKCS7-----" >> pkcs7
   ```

   IMDSv1

   ```
   $ echo "-----BEGIN PKCS7-----" >> pkcs7 \
    && curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7 \
    && echo "" >> pkcs7 \
    && echo "-----END PKCS7-----" >> pkcs7
   ```

3. 在 实例身份文档签名的 Amazon 公有证书 中找到您所在区域的 DSA 公有证书，并将内容添加到名为 certificate 的新文件。
4. 使用 OpenSSL smime 命令来验证签名。包括 -verify 选项以指示需要验证签名，而包括 -noverify 选项则指示不需要验证证书。

```
$ openssl smime -verify -in pkcs7 -inform PEM -certfile certificate -noverify | tee
 document
```

如果签名有效，则会显示 `Verification successful` 消息。

此命令还会将实例身份文档的内容写入一个名为 `document` 的新文件。您可以使用以下命令将来自实例元数据的实例身份文档内容与此文件的内容进行比较。

```
$ openssl dgst -sha256 < document
```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

如果无法验证签名，请联系 Amazon Web Services 支持。

Windows 实例

先决条件

此过程需要 `System.Security` Microsoft .NET Core 类。要将该类添加到 PowerShell 会话中，请运行以下命令。

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> 此命令仅将该类添加到当前 PowerShell 会话。如果您启动新会话，则必须再次运行该命令。

使用 PKCS7 签名和 Amazon DSA 公有证书验证实例身份文档

1. 连接到实例。
2. 从实例元数据中检索 PKCS7 签名，将其转换为字节数组并添加到名为 `$Signature` 的变量。根据实例使用的 IMDS 版本，使用以下命令之一。

IMDSv2

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-ec2-
metadata-token-ttl-seconds' = '21600'} -Uri http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} -Uri http://169.254.169.254/latest/
dynamic/instance-identity/pkcs7).Content)
```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Uri
 http://169.254.169.254/latest/dynamic/instance-identity/pkcs7).Content)
```

3. 从实例元数据中检索纯文本实例身份文档，将其转换为字节数组并添加到名为 $Document 的变量。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} -Uri http://169.254.169.254/latest/
   dynamic/instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. 在 实例身份文档签名的 Amazon 公有证书 中找到您所在区域的 DSA 公有证书，并将内容添加到名为 certificate.pem 的新文件。

5. 从证书文件中提取证书并将其存储在名为 $Store 的变量中。

   ```
   PS C:\> $Store =
    [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
   Path certificate.pem)))
   ```

6. 验证签名。

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

如果签名有效，则命令不返回任何输出。如果无法验证签名，则命令返回 `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`。如果无法验证您的签名，请联系 Amazon Web Services 支持。

7. 验证实例身份文档的内容。

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

如果实例身份文档的内容有效，则命令返回 `True`。如果无法验证实例身份文档，请联系 Amazon Web Services 支持。

选项 2：使用 base64 编码的签名验证实例身份文档

本主题说明如何使用 base64 编码的签名和 Amazon RSA 公有证书验证实例身份文档。

Linux 实例

使用 base64 编码的签名和 Amazon RSA 公有证书验证实例身份文档

1. 连接到实例。
2. 从实例元数据中检索 base64 编码的签名，将其转换为二进制文件，然后将其添加到名为 `signature` 的文件。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/signature | base64 -d >> signature
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/signature |
 base64 -d >> signature
```

3. 从实例元数据中检索明文 实例身份文档，并将其添加到名为 document 的文件。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

   ```
   $ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
   metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
   dynamic/instance-identity/document >> document
   ```

   IMDSv1

   ```
   $ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document
    >> document
   ```

4. 在 实例身份文档签名的 Amazon 公有证书 中找到您所在区域的 RSA 公有证书，并将内容添加到名为 certificate 的新文件。

5. 从 Amazon RSA 公有证书中提取公有密钥，并将其保存到名为 key 的文件。

   ```
   $ openssl x509 -pubkey -noout -in certificate >> key
   ```

6. 使用 OpenSSL dgst 命令来验证实例身份文档。

   ```
   $ openssl dgst -sha256 -verify key -signature signature document
   ```

   如果签名有效，则会显示 Verification successful 消息。

   此命令还会将实例身份文档的内容写入一个名为 document 的新文件。您可以使用以下命令将来自实例元数据的实例身份文档内容与此文件的内容进行比较。

   ```
   $ openssl dgst -sha256 < document
   ```

```
$ curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/document | openssl dgst -sha256
```

如果无法验证签名，请联系 Amazon Web Services 支持。

Windows 实例

使用 base64 编码的签名和 Amazon RSA 公有证书验证实例身份文档

1. 连接到实例。
2. 从实例元数据中检索 base64 编码的签名，将其转换为字节数组并添加到名为 $Signature 的变量。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

   ```
   PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
   ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
   token).Content
   ```

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/signature).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/signature).Content)
   ```

3. 从实例元数据中检索纯文本实例身份文档，将其转换为字节数组并添加到名为 $Document 的变量。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

IMDSv1

```
PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
```

4. 在 实例身份文档签名的 Amazon 公有证书 中找到您所在区域的 RSA 公有证书，并将内容添加到名为 `certificate.pem` 的新文件。
5. 验证实例身份文档。

```
PS C:\> [Security.Cryptography.X509Certificates.X509Certificate2]::new((Resolve-
Path certificate.pem)).PublicKey.Key.VerifyData($Document, 'SHA256', $Signature)
```

如果签名有效，则命令返回 `True`。如果无法验证签名，请联系 Amazon Web Services 支持。

选项 3：使用 RSA-2048 签名验证实例身份文档

本主题说明如何使用 RSA-2048 签名和 Amazon RSA-2048 公有证书验证实例身份文档。

Linux 实例

使用 RSA-2048 签名和 Amazon RSA-2048 公有证书验证实例身份文档。

1. 连接到实例。
2. 从实例元数据中检索 RSA-2048 签名，并将其连同所需的页眉和页脚添加到名为 `rsa2048` 的文件。根据实例使用的 IMDS 版本，使用以下命令之一。

IMDSv2

```
$ echo "-----BEGIN PKCS7-----" >> rsa2048 \
 && TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
dynamic/instance-identity/rsa2048 >> rsa2048 \
 && echo "" >> rsa2048 \
 && echo "-----END PKCS7-----" >> rsa2048
```

IMDSv1

```
$ echo "-----BEGIN PKCS7-----" >> rsa2048 \
```

```
&& curl -s http://169.254.169.254/latest/dynamic/instance-identity/rsa2048
>> rsa2048 \
&& echo "" >> rsa2048 \
&& echo "-----END PKCS7-----" >> rsa2048
```

3. 在 实例身份文档签名的 Amazon 公有证书 中找到您所在区域的 RSA-2048 公有证书，并将内容添加到名为 `certificate` 的新文件。
4. 使用 OpenSSL smime 命令来验证签名。包括 `-verify` 选项以指示需要验证签名，而包括 `-noverify` 选项则指示不需要验证证书。

```
$ openssl smime -verify -in rsa2048 -inform PEM -certfile certificate -noverify |
 tee document
```

如果签名有效，则会显示 `Verification successful` 消息。如果无法验证签名，请联系 Amazon Web Services 支持。

Windows 实例

先决条件

此过程需要 `System.Security` Microsoft .NET Core 类。要将该类添加到 PowerShell 会话中，请运行以下命令。

```
PS C:\> Add-Type -AssemblyName System.Security
```

> **Note**
>
> 此命令仅将该类添加到当前 PowerShell 会话。如果您启动新会话，则必须再次运行该命令。

使用 RSA-2048 签名和 Amazon RSA-2048 公有证书验证实例身份文档。

1. 连接到实例。
2. 从实例元数据中检索 RSA-2048 签名，将其转换为字节数组并添加到名为 `$Signature` 的变量。根据实例使用的 IMDS 版本，使用以下命令之一。

IMDSv2

```
PS C:\> [string]$token = (Invoke-WebRequest -Method Put -Headers @{'X-aws-
ec2-metadata-token-ttl-seconds' = '21600'} http://169.254.169.254/latest/api/
token).Content
```

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest -Headers
 @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
instance-identity/rsa2048).Content)
```

IMDSv1

```
PS C:\> $Signature = [Convert]::FromBase64String((Invoke-WebRequest
 http://169.254.169.254/latest/dynamic/instance-identity/rsa2048).Content)
```

3. 从实例元数据中检索纯文本实例身份文档，将其转换为字节数组并添加到名为 $Document 的变量。根据实例使用的 IMDS 版本，使用以下命令之一。

   IMDSv2

   ```
   PS C:\> $Document = [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest -Headers
    @{'X-aws-ec2-metadata-token' = $Token} http://169.254.169.254/latest/dynamic/
   instance-identity/document).Content)
   ```

   IMDSv1

   ```
   PS C:\> $Document =  [Text.Encoding]::UTF8.GetBytes((Invoke-WebRequest
    http://169.254.169.254/latest/dynamic/instance-identity/document).Content)
   ```

4. 在 实例身份文档签名的 Amazon 公有证书 中找到您所在区域的 RSA-2048 公有证书，并将内容添加到名为 certificate.pem 的新文件。

5. 从证书文件中提取证书并将其存储在名为 $Store 的变量中。

   ```
   PS C:\> $Store =
    [Security.Cryptography.X509Certificates.X509Certificate2Collection]::new([Security.Cryptog
   Path certificate.pem)))
   ```

6. 验证签名。

```
PS C:\> $SignatureDocument = [Security.Cryptography.Pkcs.SignedCms]::new()
```

```
PS C:\> $SignatureDocument.Decode($Signature)
```

```
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

如果签名有效，则命令不返回任何输出。如果无法验证签名，则命令返回 `Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer`。如果无法验证您的签名，请联系 Amazon Web Services 支持。

7. 验证实例身份文档的内容。

```
PS C:
\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

如果实例身份文档的内容有效，则命令返回 `True`。如果无法验证实例身份文档，请联系 Amazon Web Services 支持。

实例身份文档签名的 Amazon 公有证书

以下 Amazon 公有证书可用于验证实例的实例身份文档内容，如验证实例身份文档中所述。

请确保使用适用于所在区域，以及正在使用的验证程序的正确证书。如果要验证 PKCS7 签名，请使用 DSA 证书。如果要验证 base64 编码的签名，请使用 RSA 证书。如果要验证 RSA-2048 签名，请使用 RSA-2048 证书。

展开下面的每个区域，查看特定于区域的证书。

中国（北京）— cn-north-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
```

```
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAOtrM5XLDSjCMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTA4MTQx
MDAxNDJaGA8yMTk1MDExNzEwMDE0MlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
```

```
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAvVBz+WQNdPiM9S+aUULOQEriTmNDUrjLWLr7SfaOJScBzis5D5ju0jh1
+qJdkbuGKtFX5OTWTm8pWhInX+hIOoS3exC4BaANoa1A3o6quoG+Rsv72qQf8LLH
sgEi6+LMlCN9TwnRKOToEabmDKorss4zFl7VSsbQJwcBSfOcIwbdRRaW9Ab6uJHu
79L+mBR3Ea+G7vSDrVIA8goAPkae6jY9WGw9KxsOrcvNdQoEkqRVtHo4bs9fMRHU
Etphj2gh4ObXlFN92VtvzD6QBs3CcoFWgyWGvzg+dNG5VCbsiiuRdmii3kcijZ3H
Nv1wCcZoEAqH72etVhsuvNRC/xAP8wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQA8
ezx5LRjzUU9EYWYhyYIEShFlP1qDHs7F4L46/5lc4pL8FPoQm5CZuAF31DJhYi/b
fcV7i3n++/ymQbCLC6kAg8DUB7NrcROll5ag8d/JXGzcTCnlDXLXx1905fPNa+jI
0q5quTmdmiSi0taeaKZmyUdhrB+a7ohWdSdlokEIOtbH1P+g5yll3bI2leYE6Tm8
LKbyfK/532xJPqO9abx4Ddn89ZEC6vvWVNDgTsxERg992Wi+/xoSw3XxkgAryIv1
zQ4dQ6irFmXwCWJqc6kHg/M5W+z60S/94+wGTXmp+19U6Rkq5jVMLh16XJXrXwHe
4KcgIS/aQGVgjM6wivVA
-----END CERTIFICATE-----
```

中国（宁夏）— cn-northwest-1

DSA

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzANBgkqhkiG9w0BAQsFADBcMQswCQYDVQQGEwJV
UzEZMBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnZpY2VzIExMQzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
gsw9+QSnEJeYWnmivJWOBdn9CyDpN7cpHVmeGgNJL2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2zO8xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulpo1uJ31
jsTAPKZ3p1/sxPXBBAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJslFcl90TZbQbgxIi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7zl+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GEtqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSEO1JX2q5jhcFP90rDOtZjlpYv57N/Z9iQ+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+61lMVv/qRjnyKfS9
c96nE98sYFj0ZVBzXw8Sq4Gh8FiVmFHbQp1peGC19idOUqxPxWsasWxQXO0azYsP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

RSA

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJALSOMbOoU2svMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAeFw0yMzA3MDQw
ODM1MzlaFw0yODA3MDIwODM1MzlaMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBX
YXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6
b24gV2ViIFNlcnZpY2VzIExMQzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
uhhUNlqAZdcWWB/OSDVDGk3OA99EFzOn/mJlmciQ/Xwu2dFJWmSCqEAE6gjufCjQ
q3voxAhC2CF+elKtJW/C0Sz/LYo60PUqd6iXF4h+upB9HkOOGuWHXsHBTsvgkgGA
1CGgel4U0Cdq+23eANr8N8m28UzljjSnTlrYCHtzN4sCAwEAAaOB1DCB0TALBgNV
HQ8EBAMCB4AwHQYDVR0OBBYEFBkZu3wT27NnYgrfH+xJz4HJaNJoMIGOBgNVHSME
gYYwgYOAFBkZu3wT27NnYgrfH+xJz4HJaNJooWCkXjBcMQswCQYDVQQGEwJVUzEZ
MBcGA1UECBMQV2FzaGluZ3RvbiBTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEgMB4G
A1UEChMXQW1hem9uIFdlYiBTZXJ2aWNlcyBMTEOCCQC0jjGzqFNrLzASBgNVHRMB
Af8ECDAGAQH/AgEAMA0GCSqGSIb3DQEBCwUAA4GBAECji43p+oPkYqmzll7e8Hgb
oADS0ph+YUz5P/bUCm61wFjlxaTfwKcuTR3ytj7bFLoW5Bm7Sa+TCl3lOGb2taon
2h+9NirRK6JYk87LMNvbS40HGPFumJL2NzEsGUeK+MRiWu+Oh5/lJGii3qw4YByx
SUDlRyNy1jJFstEZjOhs
-----END CERTIFICATE-----
```

RSA-2048

```
-----BEGIN CERTIFICATE-----
MIIDOzCCAiOgAwIBAgIJAPu4ssY3BlzcMA0GCSqGSIb3DQEBCwUAMFwxCzAJBgNV
BAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xNTEyMDMy
MTI5MzJaGA8yMTk1MDUwODIxMjkzMlowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAgT
EFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0Ft
YXpvbiBXZWIgU2VydmljZXMgTExDMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsOiGi4A6+YTLzCdIyP8b8SCT2M/6PGKwzKJ5XbSBoL3gsnSWiFYqPg9c
uJPNbiy9wSA9vlyfWMd90qvTfiNrT6vewP813QdJ3EENZOx4ERcf/Wd22tV72kxD
yw1Q3I1OMH4bOItGQAxU5OtXCjBZEEUZooOkU8RoUQOU2Pql4NTiUpzWacNutAn5
HHS7MDc4lUlsJqbN+5QW6fFrcNG/0Mrib3JbwdFUNhrQ5j+Yq5h78HarnUivnX/3
Ap+oPbentv1qd7wvPJu556LZuhfqI0TohiIT1Ah+yUdN5osoaMxTHKKtf/CsSJ1F
w3qXqFJQA0VWsqjFyHXFI32I/GOupwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCn
Um00QHvUsJSN6KATbghowLynHn3wZSQsuS8E0COpcFJFxP2SV0NYkERbXu0n/Vhi
yq5F8v4/bRA2/xpedLWmvFs7QWlomuXhSnYFkd33Z5gnXPb9vRkLwiMSw4uXls35
qQraczUJ9EXDhrv7VmngIk9H3YsxYrlDGEqh/oz4Ze4ULOgnfkauanHikk+BUEsg
/jsTD+7e+niEzJPihHdsvKFDlud5pakEzyxovHwNJ1GS2I//yxrJFIL91mehjqEk
RLPdNse7N6UvSnuXcOokwu6l6kfzigGkJBxkcq4gre3szZFdCQcUioj7Z4xtuTL8
YMqfiDtN5cbD8R8ojw9Y
```

```
-----END CERTIFICATE-----
```

EC2 实例的 STIG 合规性

安全技术实施指南 (STIG) 是 Defense Information Systems Agency (DISA) 创建的配置强化标准，用于保护信息系统和软件。为使您的系统符合 STIG 标准，您必须安装、配置和测试多种安全设置。

STIG 的应用基于以下漏洞分类。在创建 STIG 强化映像或将 STIG 设置应用于现有实例时，您可以选择应用到哪个级别进行强化。较高的级别包含所有较低级别的 STIG 设置。例如，高（第一类）级别包括“中等”和“低”类别的设置。

法规遵从性级别

- 高（第一类）

  最严重的风险。包括可能导致机密性、可用性或完整性丢失的任何漏洞。
- 中等（第二类）

  包括可能导致机密性、可用性或完整性丢失的任何漏洞，但可以减轻风险。
- 低（第三类）

  包括任何会降级用于防止机密性、可用性或完整性丢失的措施的漏洞。

Amazon EC2 提供以下方法来创建 STIG 强化实例：

- 您可以从专门的公共 Amazon Windows AMI 启动预先配置为 STIG 强化的 Windows 实例。有关更多信息，请参阅《Amazon Windows AMI reference》中的 STIG Hardened Amazon Windows Server AMIs。
- 您可以创建自定义 AMI 来启动使用 EC2 Image Builder STIG 强化组件构建的预配置实例。有关更多信息，请参阅《Image Builder User Guide》中的 STIG hardening components。
- 您可以使用 AWSEC2-ConfigureSTIG Systems Manager 命令文档将 STIG 设置应用于现有 EC2 实例。Systems Manager STIG 命令文档可扫描配置错误，运行补救脚本以安装和更新 DoD 证书，以及删除不必要的证书以维持 STIG 合规性。使用 STIG Systems Manager 文档无需额外付费。

主题

- EC2 实例的 STIG 强化设置

- STIG 强化脚本下载
- 使用 Amazon Systems Manager 将 STIG 设置应用到您的实例

EC2 实例的 STIG 强化设置

STIG 设置因实例操作系统和您选择的合规性级别而异。以下内容包含在各合规级别应用的 STIG 的系统特定详细信息。

内容

- EC2 Linux 实例的 STIG 设置
- EC2 Windows 实例的 STIG 强化设置

EC2 Linux 实例的 STIG 设置

本节包含有关 Amazon EC2 支持的 Linux STIG 强化设置的信息。强化脚本将支持的 STIG 设置应用于基于 Linux 发行版的基础架构。如果 Linux 发行版没有自己的 STIG 强化设置，Amazon EC2 将使用 RHEL 设置，如下所示。

- Red Hat Enterprise Linux (RHEL) 7 设置
  - RHEL 7
  - CentOS 7
  - Amazon Linux 2 (AL2)
- RHEL 8 STIG 设置
  - RHEL 8
  - CentOS 8
- RHEL 9 STIG 设置
  - RHEL 9
  - CentOS Stream 9

Linux STIG 低（第三类）

以下列表包含 Amazon EC2 支持的适合您的基础架构的 STIG 设置。如果支持的设置不适用于您的基础架构，Amazon EC2 会跳过该设置并继续。例如，某些 STIG 强化设置可能不适用于独立服务器。特定于组织的策略也有可能影响哪些设置适用，如针对管理员查看文档设置的要求。

有关完整列表，请参阅 STIG 文档库。有关如何查看完整列表的信息，请参阅 STIG 查看工具。

RHEL 7 STIG 版本 3 发行版 15

- RHEL 7/CentOS 7/AL2

  V-204452、V-204576 和 V-204605

RHEL 8 STIG 版本 2 发行版 6

- RHEL 8/CentOS 8

  V-230241、V-230269、V-230270、V-230281、V-230285、V-230346、V-230381、V-230395、V-230468 和 V-244527

RHEL 9 STIG 版本 2 发行版 7

- RHEL 9/CentOS Stream 9

  V-257782、V-257795、V-257796、V-257824、V-257880、V-257946、V-257947、V-258037、V-258067 和 V-258173

RHEL 10 STIG 版本 1 发行版 1

V-280995、V-281203 和 V-281292

Amazon Linux 2023 STIG 版本 1 发行版 1

V-274141

SLES 12 STIG 版本 3 发行版 4

V-217108、V-217113、V-217140、V-217198、V-217209、V-217211、V-217212、V-217213、V-217214、和 V-255915

SLES 15 STIG 版本 2 发行版 6

V-234811、V-234850、V-234868、V-234873、V-234905、V-234907、V-234908、V-234909、V-234933、和 V-255921

Ubuntu 18.04 STIG 版本 2 发行版 15

V-219163、V-219164、V-219165、V-219172、V-219173、V-219174、V-219175、V-219178、V-219179、和 V-219333

Ubuntu 20.04 STIG 版本 2 发行版 4

V-238202、V-238203、V-238221、V-238222、V-238223、V-238224、V-238226、V-238234、V-238235、和 V-238373

Ubuntu 22.04 STIG 版本 2 发行版 7

V-260472、V-260476、V-260479、V-260480、V-260481、V-260520、V-260521、V-260549、V-260550、和 V-260596

Ubuntu 24.04 STIG 版本 1 发行版 4

V-270645、V-270646、V-270664、V-270677、V-270690、V-270695、V-270706、V-270710、V-270734、和 V-270820

Linux STIG 中等（第二类）

以下列表包含 Amazon EC2 支持的适合您的基础架构的 STIG 设置。如果支持的设置不适用于您的基础架构，Amazon EC2 会跳过该设置并继续。例如，某些 STIG 强化设置可能不适用于独立服务器。特定于组织的策略也有可能影响哪些设置适用，如针对管理员查看文档设置的要求。

有关完整列表，请参阅 STIG 文档库。有关如何查看完整列表的信息，请参阅 STIG 查看工具。

> **Note**
>
> Linux STIG 中等类别包括所有列出的适用于 Linux STIG 低（第三类）的 STIG 强化设置，此外还包括Amazon EC2 支持的针对第二类漏洞的 STIG 强化设置。

RHEL 7 STIG 版本 3 发行版 15

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

- RHEL 7/CentOS 7/AL2

  V-204405、V-204406、V-204407、V-204408、V-204409、V-204410、V-204411、V-204412、V-204413和 V-256970

RHEL 8 STIG 版本 2 发行版 6

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

- RHEL 8/CentOS 8

  V-230222、V-230228、V-230231、V-230233、V-230236、V-230237、V-230238、V-230239、V-230240 和 V-279931

RHEL 9 STIG 版本 2 发行版 7

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

- RHEL 9/CentOS Stream 9

  V-257780、V-257781、V-257783、V-257786、V-257788、V-257790、V-257791、V-257792、V-257793 和 V-279936

RHEL 10 STIG 版本 1 发行版 1

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-280094、V-280942、V-280943、V-280945、V-280946、V-280947、V-280948、V-280950、V-280952、和 V-281365

Amazon Linux 2023 STIG 版本 1 发行版 2

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-273995、V-274000、V-274001、V-274002、V-274003、V-274004、V-274005、V-274006、V-274008、和 V-274187

SLES 12 STIG 版本 3 发行版 4

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-217102、V-217105、V-217105、V-217106、V-217106、V-217110、V-217116、V-217117、V-217118、和 V-256981

SLES 15 STIG 版本 2 发行版 6

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-234802、V-234807、V-234808、V-234809、V-234813、V-234815、V-234817、V-234821、V-234822、和 V-274879

Ubuntu 18.04 STIG 版本 2 发行版 15

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-219149、V-219155、V-219156、V-219160、V-219166、V-219168、V-219176、V-219181、V-219184、和 V-255906

Ubuntu 20.04 STIG 版本 2 发行版 4

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-238200、V-238205、V-238207、V-238209、V-238210、V-238211、V-238212、V-238213、V-238220、和 V-274853

Ubuntu 22.04 STIG 版本 2 发行版 7

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-260471、V-260473、V-260474、V-260475、V-260475、V-260477、V-260478、V-260485、V-260486、和 V-274866

Ubuntu 24.04 STIG 版本 1 发行版 4

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-270649、V-270651、V-270652、V-270653、V-270654、V-270656、V-270657、V-270659、V-270660、和 V-274873

Linux STIG 高（第一类）

以下列表包含 Amazon EC2 支持的适合您的基础架构的 STIG 设置。如果支持的设置不适用于您的基础架构，Amazon EC2 会跳过该设置并继续。例如，某些 STIG 强化设置可能不适用于独立服务器。特定于组织的策略也有可能影响哪些设置适用，如针对管理员查看文档设置的要求。

有关完整列表，请参阅 STIG 文档库。有关如何查看完整列表的信息，请参阅 STIG 查看工具。

> **Note**
>
> Linux STIG High 类别包括所有列出的适用于 Linux STIG 中低类别的 STIG 强化设置，以及 Amazon EC2 支持的 I 类漏洞的 STIG 强化设置。

RHEL 7 STIG 版本 3 发行版 15

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

- RHEL 7/CentOS 7/AL2

  V-204424、V-204425、V-204442、V-204443、V-204447、V-204448、V-204455、V-204462、V-204497 和 V-204621

RHEL 8 STIG 版本 2 发行版 6

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

- RHEL 8/CentOS 8

  V-230223、V-230264、V-230283、V-230284、V-230487、V-230492、V-230533、V-230558、V-244540 和 V-230531

RHEL 9 STIG 版本 2 发行版 7

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

- RHEL 9/CentOS Stream 9

  V-257820、V-257821、V-257826、V-257835、V-257955、V-257956、V-258059、V-258230、V-258238 和 V-257785

RHEL 10 版本 1 发行版 1

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-280932、V-280933、V-280934、V-280944、V-280949、V-280951、V-280978、V-281007、V-281014、和 V-281307

Amazon Linux 2023 STIG 版本 1 发行版 2

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-273996、V-273997、V-273999、V-274007、V-274038、V-274039、V-274046、V-274052、V-274057 和 V-274153

SLES 12 STIG 版本 3 发行版 4

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-217101、V-217139、V-217141、V-217142、V-217159、V-217160、V-217164、V-217264、V-217268、和 V-251721

SLES 15 STIG 版本 2 发行版 6

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-234800、V-234804、V-234818、V-234852、V-234859、V-234860、V-234898、V-234984、V-234985、和 V-251725

Ubuntu 18.04 STIG 版本 2 发行版 15

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-219157、V-219158、V-219177、V-219212、V-219308、V-219314、V-219316 和 V-251507

Ubuntu 20.04 STIG 版本 2 发行版 4

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-238201、V-238218、V-238219、V-238326、V-238327、V-238380 和 V-251504

Ubuntu 22.04 STIG 版本 2 发行版 7

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-260469、V-260482、V-260483、V-260523、V-260524、V-260526、V-260529、V-260539、V-260570、和 V-279937

Ubuntu 24.04 STIG 版本 1 发行版 4

包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-270647、V-270648、V-270665、V-270666、V-270708、V-270711、V-270712、V-270713、V-270714、和 V-279938

EC2 Windows 实例的 STIG 强化设置

Amazon EC2 支持以下 Windows 操作系统版本的 STIG 基准：

- Windows Server 2025

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

本节列出了 Amazon EC2 支持的、适合您的 Windows 基础架构当前 STIG 设置，以及版本历史记录日志。

您可以应用低、中或高 STIG 设置。

Windows STIG 低（第三类）

以下列表包含 Amazon EC2 支持的适合您的基础架构的 STIG 设置。如果支持的设置不适用于您的基础架构，Amazon EC2 会跳过该设置并继续。例如，某些 STIG 强化设置可能不适用于独立服务器。特定于组织的策略也有可能影响哪些设置适用，如针对管理员查看文档设置的要求。

有关当前 Windows STIG 的完整列表，请参阅 STIG 文档库。有关如何查看完整列表的信息，请参阅 STIG 查看工具。

- Windows Server 2025 STIG 版本 1 发行版 1

  V-278082、V-278083、V-278084、V-278085、V-278098、V-278104、V-278110 和 V-278231
- Windows Server 2022 STIG 版本 2 发行版 7

  V-254335、V-254336、V-254337、V-254338、V-254351、V-254357、V-254363 和 V-254481
- Windows Server 2019 STIG 版本 3 发行版 7

  V-205691、V-205819、V-205858、V-205859、V-205860、V-205870、V-205871 和 V-205923
- Windows Server 2016 STIG 版本 2 发行版 10

  V-224916、V-224917、V-224918、V-224919、V-224931、V-224942 和 V-225060
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5

  V-225250、V-225318、V-225319、V-225324、V-225327、V-225328、V-225330、V-225331、V-225332 和 V-225537
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 7

  STIG 设置未应用于第三类漏洞的 Microsoft .NET Framework。

- Windows Firewall STIG 版本 2 发行版 2

  V-241994、V-241995、V-241996、V-241999、V-242000、V-242001、V-242006、V-242007 和 V-242008
- Internet Explorer 11 STIG 版本 2 发行版 6

  V-223056 和 V-223078
- Microsoft Edge STIG 版本 2 发行版 4（仅限 Windows Server 2022 和 2025）

  V-235727、V-235731、V-235751、V-235752 和 V-235765

Windows STIG 中等（第二类）

以下列表包含 Amazon EC2 支持的适合您的基础架构的 STIG 设置。如果支持的设置不适用于您的基础架构，Amazon EC2 会跳过该设置并继续。例如，某些 STIG 强化设置可能不适用于独立服务器。特定于组织的策略也有可能影响哪些设置适用，如针对管理员查看文档设置的要求。

有关当前 Windows STIG 的完整列表，请参阅 STIG 文档库。有关如何查看完整列表的信息，请参阅 STIG 查看工具。

> **Note**
>
> Windows STIG 中等类别包括所有列出的适用于 Windows STIG Low（第三类）的 STIG 强化设置，此外还包括 Amazon EC2 支持的针对第二类漏洞的 STIG 强化设置。

- Windows Server 2025 STIG 版本 1 发行版 1

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-278015、V-278016、V-278019、V-278020、V-278021、V-278022、V-278023、V-278024、V-278025 和 V-279923
- Windows Server 2022 STIG 版本 2 发行版 7

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-254247、V-254269、V-254270、V-254271、V-254272、V-254273、V-254274、V-254275、V-254276 和 V-278949
- Windows Server 2019 STIG 版本 3 发行版 7

包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-205625、V-205626、V-205627、V-205629、V-205630、V-205633、V-205634、V-205635、V-205636 和 V-278941

- Windows Server 2016 STIG 版本 2 发行版 10

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-224850、V-224851、V-224852、V-224853、V-224854、V-224855、V-224856、V-224857、V-224858 和 V-257502

- Windows Server 2012 R2 MS STIG 版本 3 发行版 5

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-225239、V-225259、V-225260、V-225261、V-225263、V-225264、V-225265、V-225266、V-225267 和 V-225574

- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 7

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-225223、V-225230、V-225235、V-225238

- Windows Firewall STIG 版本 2 发行版 2

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-241989、V-241990、V-241991、V-241993、V-241998、V-242003、V-242004 和 V-242005

- Internet Explorer 11 STIG 版本 2 发行版 6

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

  V-223015、V-223017、V-223018、V-223019、V-223020、V-223021、V-223022、V-223023、V-223024 和 V-250541

- Microsoft Edge STIG 版本 2 发行版 4（仅限 Windows Server 2022 和 2025）

  V-235720、V-235721、V-235723、V-235724、V-235725、V-235726、V-235728、V-235729、V-235730 和 V-246736

- Microsoft Defender STIG 版本 2 发行版 7

  包括 Amazon EC2 支持的针对第 III 类（低）漏洞的所有 STIG 强化设置，以及：

V-213427、V-213429、V-213430、V-213431、V-213432、V-213433、V-213434、V-213435、V-213436 和 V-278863

Windows STIG 高（第一类）

以下列表包含 Amazon EC2 支持的适合您的基础架构的 STIG 设置。如果支持的设置不适用于您的基础架构，Amazon EC2 会跳过该设置并继续。例如，某些 STIG 强化设置可能不适用于独立服务器。特定于组织的策略也有可能影响哪些设置适用，如针对管理员查看文档设置的要求。

有关当前 Windows STIG 的完整列表，请参阅 STIG 文档库。有关如何查看完整列表的信息，请参阅 STIG 查看工具。

> **Note**
>
> Windows STIG 高类别包括所有列出的适用于 Windows STIG 中低类别的 STIG 强化设置，此外还包括 Amazon EC2 支持的针对第一类漏洞的 STIG 强化设置。

- Windows Server 2025 STIG 版本 1 发行版 1

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-278040、V-278099、V-278100、V-278101、V-278121、V-278125、V-278128、V-278196、V-278215 和 V-278250
- Windows Server 2022 STIG 版本 2 发行版 7

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-254250、V-254293、V-254352、V-254353、V-254354、V-254374、V-254378、V-254381、V-254446 和 V-254500
- Windows Server 2019 STIG 版本 3 发行版 7

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-205653、V-205654、V-205663、V-205711、V-205713、V-205724、V-205725、V-205750、V-205753 和 V-205919
- Windows Server 2016 STIG 版本 2 发行版 10

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

V-224831、V-224874、V-224932、V-224933、V-224934、V-224954、V-224958、V-224961、V-225025 和 V-225091

- Windows Server 2012 R2 MS STIG 版本 3 发行版 5

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-225274、V-225354、V-225364、V-225365、V-225366、V-225390、V-225396、V-225399、V-225444 和 V-225556

- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 7

  包括 Amazon EC2 支持的针对 Microsoft .NET Framework 第二类和第三类（中等和低）漏洞的所有 STIG 强化设置。第一类漏洞未应用额外 STIG 设置。

- Windows Firewall STIG 版本 2 发行版 2

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-241992、V-241997 和 V-242002

- Internet Explorer 11 STIG 版本 2 发行版 6

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-252910

- Microsoft Edge STIG 版本 2 发行版 4（仅限 Windows Server 2022 和 2025）

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-235758 和 V-235759

- Microsoft Defender STIG 版本 2 发行版 7

  包括 Amazon EC2 支持的针对第一类和第三类（中等和低）漏洞的所有 STIG 强化设置，以及：

  V-213426、V-213428、V-213452、V-213453

STIG 强化脚本下载

Amazon 将 STIG 强化脚本组合成适合每个版本的操作系统相关捆绑包。捆绑包是适用于其下载和运行的目标操作系统的归档文件。Linux 脚本捆绑包存储为 TAR 文件（文件扩展名为 .tgz）。Windows 脚本捆绑包存储为 ZIP 文件（文件扩展名为 .zip）。

Amazon 将脚本捆绑包存储在每个 Amazon Web Services 区域的 EC2 Windows S3 STIG 存储桶中。Linux 捆绑包没有单独的存储桶。使用 SSL/TLS 与 Amazon 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。

主题

- STIG 下载捆绑包详细信息
- Linux STIG 版本历史记录
- Windows STIG 版本历史

STIG 下载捆绑包详细信息

> **Important**
>
> 除少数例外情况外，Systems Manager 文档下载的 STIG 强化脚本不会安装第三方软件包。如果实例上已经安装了第三方软件包，或 `InstallPackage` 参数设置为 `yes`，则会应用 Amazon EC2 支持该软件包的相关 STIG。

当您 Amazon CLI 从中运行以下命令时，Amazon S3 会从存储桶中下载最新的 STIG 强化脚本捆绑包文件。

```
aws s3 cp s3://aws-windows-downloads-region/STIG/operating system/Latest/bundle-
name destination-directory
```

示例：下载到临时目录

此示例显示下载到 `/tmp` 目录的 Linux 捆绑包

```
aws s3 cp s3://aws-windows-downloads-us-east-1/STIG/Linux/
Latest/LinuxAWSConfigureSTIG.tgz /tmp
```

下载文件存储路径和捆绑包文件名的模式和示例如下所示：

下载文件存储路径

```
s3://aws-windows-downloads-<region>/STIG/<operating system>/Latest/<bundle
file name>
```

下载路径变量

region

Amazon Web Services 区域（每个区域都有自己的下载存储桶。）

operating system

应用 STIG 的实例的操作系统平台：`Linux` 或 `Windows`。

bundle file name

格式为 *`<os bundle name>.<file extension>`*。

os bundle name

操作系统捆绑包的标准名称前缀为 `LinuxAWSConfigureSTIG` 或 `AWSConfigureSTIG`。为保持向后兼容性，Windows 版的下载不包含平台前缀。

file extension

压缩文件格式 `tgz` (Linux) 或 `zip` (Windows)。

捆绑包文件名示例

- `LinuxAWSConfigureSTIG.tgz`
- `AWSConfigureSTIG.zip`

Linux STIG 版本历史记录

本节记录每季度更新的 Linux 脚本捆绑包的版本历史记录。要查看一个季度的变化和发布的版本，请选择其标题以展开信息。如果该季度没有任何变化，您将在标题中看到相应标注。

2026 年第一季度更改：2026 年 4 月 7 日：

添加了对 RHEL 10 操作系统的支持，更新了以下 STIG 版本，并对 2026 年第一季度发行版的所有合规性级别（低/中/高）应用了 STIG：

STIG-Build-Linux 版本 1.0.x

- RHEL 8 STIG 版本 2 发行版 6
- RHEL 9 STIG 版本 2 发行版 7
- RHEL 10 STIG 版本 1 发行版 1
- Amazon Linux 2023 STIG 版本 1 发行版 2

- SLES 12 STIG 版本 3 发行版 4
- SLES 15 STIG 版本 2 发行版 6
- Ubuntu 20.04 STIG 版本 2 发行版 4
- Ubuntu 22.04 STIG 版本 2 发行版 7
- Ubuntu 24.04 STIG 版本 1 发行版 4

2025 年第三季度更改 - 2025 年 9 月 4 日：

添加了对 SUSE Linux Enterprise Server（SLES）操作系统和 Amazon Linux 2023 的支持。更新了以下 STIG 版本，并对 2025 年第三季度发行版的所有合规性级别（低/中/高）应用了 STIG：

STIG-Build-Linux 版本 1.0.x

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 4
- RHEL 9 STIG 版本 2 发行版 5
- Amazon Linux 2023 STIG 版本 1 发行版 1
- SLES 12 STIG 版本 3 发行版 3
- SLES 15 STIG 版本 2 发行版 5
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 3
- Ubuntu 22.04 STIG 版本 2 发行版 5
- Ubuntu 24.04 STIG 版本 1 发行版 2

2025 年第二季度更改 - 2025 年 6 月 26 日：

更新了以下 STIG 版本，并对 2025 年第二季度发行版应用了 STIG：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 3
- RHEL 9 STIG 版本 2 发行版 4
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 2

- Ubuntu 22.04 STIG 版本 2 发行版 4
- Ubuntu 24.04 STIG 版本 1 发行版 1

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 3
- RHEL 9 STIG 版本 2 发行版 4
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 2
- Ubuntu 22.04 STIG 版本 2 发行版 4
- Ubuntu 24.04 STIG 版本 1 发行版 1

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 3
- RHEL 9 STIG 版本 2 发行版 4
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 2
- Ubuntu 22.04 STIG 版本 2 发行版 4
- Ubuntu 24.04 STIG 版本 1 发行版 1

2025 年第一季度更改 - 2025 年 4 月 11 日：

更新了以下 STIG 版本，对 2025 年第一季度发行版应用了 STIG，并添加了对 Ubuntu 24.04 的支持：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 2
- RHEL 9 STIG 版本 2 发行版 3
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 2

- Ubuntu 22.04 STIG 版本 2 发行版 3
- Ubuntu 24.04 STIG 版本 1 发行版 1

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 2
- RHEL 9 STIG 版本 2 发行版 3
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 2
- Ubuntu 22.04 STIG 版本 2 发行版 3
- Ubuntu 24.04 STIG 版本 1 发行版 1

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 2
- RHEL 9 STIG 版本 2 发行版 3
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 2
- Ubuntu 22.04 STIG 版本 2 发行版 3
- Ubuntu 24.04 STIG 版本 1 发行版 1

2024 年第四季度更改 - 2024 年 12 月 10 日：

更新了以下 STIG 版本，对 2024 年第四季度发行版应用了 STIG，并添加了有关 Linux 组件的两个新输入参数的信息：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 1
- RHEL 9 STIG 版本 2 发行版 2
- Ubuntu 18.04 STIG 版本 2 发行版 15

- Ubuntu 20.04 STIG 版本 2 发行版 1
- Ubuntu 22.04 STIG 版本 2 发行版 2

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 1
- RHEL 9 STIG 版本 2 发行版 2
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 1
- Ubuntu 22.04 STIG 版本 2 发行版 2

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 15
- RHEL 8 STIG 版本 2 发行版 1
- RHEL 9 STIG 版本 2 发行版 2
- Ubuntu 18.04 STIG 版本 2 发行版 15
- Ubuntu 20.04 STIG 版本 2 发行版 1
- Ubuntu 22.04 STIG 版本 2 发行版 2

2024 年第三季度更改 - 2024 年 10 月 4 日（无更改）：

2024 年第三季度发行版的 Linux 组件 STIG 无更改。

2024 年第二季度更改 - 2024 年 5 月 10 日：

更新了 STIG 版本，并对 2024 年第二季度发行版应用了 STIG。还新增了对 RHEL 9、CentOS Stream 9 和 Ubuntu 22.04 的支持，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 14
- RHEL 8 STIG 版本 1 发行版 14
- RHEL 9 STIG 版本 1 发行版 3
- Ubuntu 18.04 STIG 版本 2 发行版 14

- Ubuntu 20.04 STIG 版本 1 发行版 12
- Ubuntu 22.04 STIG 版本 1 发行版 1

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 14
- RHEL 8 STIG 版本 1 发行版 14
- RHEL 9 STIG 版本 1 发行版 3
- Ubuntu 18.04 STIG 版本 2 发行版 14
- Ubuntu 20.04 STIG 版本 1 发行版 12
- Ubuntu 22.04 STIG 版本 1 发行版 1

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 14
- RHEL 8 STIG 版本 1 发行版 14
- RHEL 9 STIG 版本 1 发行版 3
- Ubuntu 18.04 STIG 版本 2 发行版 14
- Ubuntu 20.04 STIG 版本 1 发行版 12
- Ubuntu 22.04 STIG 版本 1 发行版 1

2024 年第一季度更改 - 2024 年 2 月 6 日：

更新了 STIG 版本，并对 2024 年第一季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 14
- RHEL 8 STIG 版本 1 发行版 13
- Ubuntu 18.04 STIG 版本 2 发行版 13
- Ubuntu 20.04 STIG 版本 1 发行版 11

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 14

- RHEL 8 STIG 版本 1 发行版 13
- Ubuntu 18.04 STIG 版本 2 发行版 13
- Ubuntu 20.04 STIG 版本 1 发行版 11

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 14
- RHEL 8 STIG 版本 1 发行版 13
- Ubuntu 18.04 STIG 版本 2 发行版 13
- Ubuntu 20.04 STIG 版本 1 发行版 11

2023 年第四季度更改 - 2023 年 12 月 7 日：

更新了 STIG 版本，并对 2023 年第四季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 13
- RHEL 8 STIG 版本 1 发行版 12
- Ubuntu 18.04 STIG 版本 2 发行版 12
- Ubuntu 20.04 STIG 版本 1 发行版 10

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 13
- RHEL 8 STIG 版本 1 发行版 12
- Ubuntu 18.04 STIG 版本 2 发行版 12
- Ubuntu 20.04 STIG 版本 1 发行版 10

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 13
- RHEL 8 STIG 版本 1 发行版 12
- Ubuntu 18.04 STIG 版本 2 发行版 12
- Ubuntu 20.04 STIG 版本 1 发行版 10

2023 年第三季度更改 — 2023 年 10 月 4 日：

更新了 STIG 版本，并对 2023 年第三季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 12
- RHEL 8 STIG 版本 1 发行版 11
- Ubuntu 18.04 STIG 版本 2 发行版 11
- Ubuntu 20.04 STIG 版本 1 发行版 9

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 12
- RHEL 8 STIG 版本 1 发行版 11
- Ubuntu 18.04 STIG 版本 2 发行版 11
- Ubuntu 20.04 STIG 版本 1 发行版 9

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 12
- RHEL 8 STIG 版本 1 发行版 11
- Ubuntu 18.04 STIG 版本 2 发行版 11
- Ubuntu 20.04 STIG 版本 1 发行版 9

2023 年第二季度更改 — 2023 年 5 月 3 日：

更新了 STIG 版本，并对 2023 年第二季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 11
- RHEL 8 STIG 版本 1 发行版 10
- Ubuntu 18.04 STIG 版本 2 发行版 11
- Ubuntu 20.04 STIG 版本 1 发行版 8

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 11
- RHEL 8 STIG 版本 1 发行版 10
- Ubuntu 18.04 STIG 版本 2 发行版 11
- Ubuntu 20.04 STIG 版本 1 发行版 8

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 11
- RHEL 8 STIG 版本 1 发行版 10
- Ubuntu 18.04 STIG 版本 2 发行版 11
- Ubuntu 20.04 STIG 版本 1 发行版 8

2023 年第一季度更改 — 2023 年 3 月 27 日：

更新了 STIG 版本，并对 2023 年第一季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 10
- RHEL 8 STIG 版本 1 发行版 9
- Ubuntu 18.04 STIG 版本 2 发行版 10
- Ubuntu 20.04 STIG 版本 1 发行版 7

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 10
- RHEL 8 STIG 版本 1 发行版 9
- Ubuntu 18.04 STIG 版本 2 发行版 10
- Ubuntu 20.04 STIG 版本 1 发行版 7

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 10

- RHEL 8 STIG 版本 1 发行版 9
- Ubuntu 18.04 STIG 版本 2 发行版 10
- Ubuntu 20.04 STIG 版本 1 发行版 7

2022 年第四季度更改 — 2023 年 2 月 1 日：

更新了 STIG 版本，并对 2022 年第四季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 9
- RHEL 8 STIG 版本 1 发行版 8
- Ubuntu 18.04 STIG 版本 2 发行版 9
- Ubuntu 20.04 STIG 版本 1 发行版 6

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 9
- RHEL 8 STIG 版本 1 发行版 8
- Ubuntu 18.04 STIG 版本 2 发行版 9
- Ubuntu 20.04 STIG 版本 1 发行版 6

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 9
- RHEL 8 STIG 版本 1 发行版 8
- Ubuntu 18.04 STIG 版本 2 发行版 9
- Ubuntu 20.04 STIG 版本 1 发行版 6

2022 年第三季度更改 — 2022 年 9 月 30 日（无更改）：

2022 年第三季度发行版的 Linux 组件 STIGS 无更改。

2022 年第二季度更改 — 2022 年 8 月 2 日：

引入了 Ubuntu 支持，更新了 STIG 版本，并对 2022 年第二季度发行版应用了 STIGS，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 7
- RHEL 8 STIG 版本 1 发行版 6
- Ubuntu 18.04 STIG 版本 2 发行版 6（全新）
- Ubuntu 20.04 STIG 版本 1 发行版 4（全新）

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 7
- RHEL 8 STIG 版本 1 发行版 6
- Ubuntu 18.04 STIG 版本 2 发行版 6（全新）
- Ubuntu 20.04 STIG 版本 1 发行版 4（全新）

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 7
- RHEL 8 STIG 版本 1 发行版 6
- Ubuntu 18.04 STIG 版本 2 发行版 6（全新）
- Ubuntu 20.04 STIG 版本 1 发行版 4（全新）

2022 年第一季度更改 — 2022 年 4 月 26 日：

已重构以包括对容器更好的支持。将之前的 AL2 脚本与 RHEL 7 结合起来。更新了 STIG 版本，并对 2022 年第一季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 6
- RHEL 8 STIG 版本 1 发行版 5

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 6
- RHEL 8 STIG 版本 1 发行版 5

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 6
- RHEL 8 STIG 版本 1 发行版 5

2021 年第四季度更改 — 2021 年 12 月 20 日：

更新了 STIG 版本，并对 2021 年第四季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 5
- RHEL 8 STIG 版本 1 发行版 4

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 5
- RHEL 8 STIG 版本 1 发行版 4

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 5
- RHEL 8 STIG 版本 1 发行版 4

2021 年第三季度更改 — 2021 年 9 月 30 日：

更新了 STIG 版本，并对 2021 年第三季度发行版应用了 STIG，如下所示：

Linux STIG 低（第三类）

- RHEL 7 STIG 版本 3 发行版 4
- RHEL 8 STIG 版本 1 发行版 3

Linux STIG 中等（第二类）

- RHEL 7 STIG 版本 3 发行版 4
- RHEL 8 STIG 版本 1 发行版 3

Linux STIG 高（第一类）

- RHEL 7 STIG 版本 3 发行版 4
- RHEL 8 STIG 版本 1 发行版 3

Windows STIG 版本历史

本节记录每季度更新的 Windows 脚本捆绑包的版本历史记录。要查看一个季度的变化和发布的版本，请选择其标题以展开信息。如果该季度没有任何变化，您将在标题中看到相应标注。

2026 年第一季度更改：2026 年 4 月 7 日：

添加了对 Windows Server 2025 操作系统的支持，更新了以下 STIG 版本，并对 2026 年第一季度发行版的所有合规性级别（低/中/高）应用了 STIG：

STIG-Build-Windows-Low 版本 1.0.x

- Windows Server 2025 STIG 版本 1 发行版 1
- Windows Server 2022 STIG 版本 2 发行版 7
- Windows Server 2019 STIG 版本 3 发行版 7
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 7
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 6
- Microsoft Edge STIG 版本 2 发行版 4（仅限 Windows Server 2022 和 2025）

2025 年第四季度更改：2025 年 12 月 10 日（无更改）：

在 2025 年第四季度发行版中，Windows 组件 STIGS 无更改。

2025 年第三季度更改 - 2025 年 9 月 4 日（无更改）：

2025 年第三季度发行版的 Windows 组件 STIG 无更改。

2025 年第二季度更改 - 2025 年 6 月 26 日：

更新了 STIG 版本并对 2025 年第二季度发行版应用了 STIG，如下所示：

Windows STIG 低（第三类）

- Windows Server 2022 STIG 版本 2 发行版 4

- Windows Server 2019 STIG 版本 3 发行版 4
- Windows Server 2016 STIG 版本 2 发行版 10
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 6
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 5
- Microsoft Edge STIG 版本 2 发行版 2（仅限 Windows Server 2022）

Windows STIG 中等（第二类）

- Windows Server 2022 STIG 版本 2 发行版 4
- Windows Server 2019 STIG 版本 3 发行版 4
- Windows Server 2016 STIG 版本 2 发行版 10
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 6
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 5
- Microsoft Edge STIG 版本 2 发行版 2（仅限 Windows Server 2022）
- Defender STIG 版本 2 发行版 4

Windows STIG 高（第一类）

- Windows Server 2022 STIG 版本 2 发行版 4
- Windows Server 2019 STIG 版本 3 发行版 4
- Windows Server 2016 STIG 版本 2 发行版 10
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 6
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 5
- Microsoft Edge STIG 版本 2 发行版 2（仅限 Windows Server 2022）
- Defender STIG 版本 2 发行版 4

2025 年第一季度更改 - 2025 年 5 月 4 日：

更新了针对 Internet Explorer 11 STIG 版本 2 发行版 5 的 STIG，适用于 2025 年第一季度发行版的所有 STIG 组件。

2024 年第四季度更改 - 2024 年 12 月 10 日：

更新了 STIG 版本并对 2024 年第四季度发行版应用了 STIG，如下所示：

Windows STIG 低（第三类）

- Windows Server 2022 STIG 版本 2 发行版 2
- Windows Server 2019 STIG 版本 3 发行版 2
- Windows Server 2016 STIG 版本 2 发行版 9
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 2
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 5
- Microsoft Edge STIG 版本 2 发行版 2（仅限 Windows Server 2022）

Windows STIG 中等（第二类）

- Windows Server 2022 STIG 版本 2 发行版 2
- Windows Server 2019 STIG 版本 3 发行版 2
- Windows Server 2016 STIG 版本 2 发行版 9
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 2
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 5
- Microsoft Edge STIG 版本 2 发行版 2（仅限 Windows Server 2022）
- Defender STIG 版本 2 发行版 4

Windows STIG 高（第一类）

- Windows Server 2022 STIG 版本 2 发行版 2

- Windows Server 2019 STIG 版本 3 发行版 2
- Windows Server 2016 STIG 版本 2 发行版 9
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 2
- Windows Firewall STIG 版本 2 发行版 2
- Internet Explorer 11 STIG 版本 2 发行版 5
- Microsoft Edge STIG 版本 2 发行版 2（仅限 Windows Server 2022）
- Defender STIG 版本 2 发行版 4

2024 年第三季度更改 - 2024 年 10 月 4 日（无更改）：

2024 年第三季度发行版的 Windows 组件 STIG 无更改。

2024 年第二季度更改 - 2024 年 5 月 10 日（无更改）：

在 2024 年第二季度发行版中，Windows 组件 STIGS 无更改。

2024 年第一季度更改 - 2024 年 2 月 23 日（无更改）：

在 2024 年第一季度发行版中，Windows 组件 STIGS 无更改。

2023 年第四季度更改 - 2023 年 12 月 7 日（无更改）：

在 2023 年第四季度发行版中，Windows 组件 STIGS 无更改。

2023 年第三季度更改 — 2023 年 10 月 4 日（无更改）：

在 2023 年第三季度发行版中，Windows 组件 STIGS 无更改。

2023 年第二季度更改 — 2023 年 5 月 3 日（无更改）：

在 2023 年第二季度发行版中，Windows 组件 STIGS 无更改。

2023 年第一季度更改 — 2023 年 3 月 27 日（无更改）：

在 2023 年第一季度发行版中，Windows 组件 STIGS 无更改。

2022 年第四季度更改 — 2023 年 2 月 1 日：

更新了 STIG 版本并对 2022 年第四季度发行版应用了 STIG，如下所示：

Windows STIG 低（第三类）

- Windows Server 2022 STIG 版本 1 发行版 1
- Windows Server 2019 STIG 版本 2 发行版 5
- Windows Server 2016 STIG 版本 2 发行版 5
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 2
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 2 发行版 3
- Microsoft Edge STIG 版本 1 发行版 6（仅限 Windows Server 2022）

Windows STIG 中等（第二类）

- Windows Server 2022 STIG 版本 1 发行版 1
- Windows Server 2019 STIG 版本 2 发行版 5
- Windows Server 2016 STIG 版本 2 发行版 5
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 2
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 2 发行版 3
- Microsoft Edge STIG 版本 1 发行版 6（仅限 Windows Server 2022）
- Defender STIG 版本 2 发行版 4（仅限 Windows Server 2022）

Windows STIG 高（第一类）

- Windows Server 2022 STIG 版本 1 发行版 1
- Windows Server 2019 STIG 版本 2 发行版 5
- Windows Server 2016 STIG 版本 2 发行版 5
- Windows Server 2012 R2 MS STIG 版本 3 发行版 5
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 2
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 2 发行版 3

- Microsoft Edge STIG 版本 1 发行版 6（仅限 Windows Server 2022）
- Defender STIG 版本 2 发行版 4（仅限 Windows Server 2022）

2022 年第三季度更改 — 2022 年 9 月 30 日（无更改）：

在 2022 年第三季度发行版中，Windows 组件 STIGS 无更改。

2022 年第二季度更改 — 2022 年 8 月 2 日：

更新了 STIG 版本，并对 2022 年第二季度发行版应用了 STIG。

Windows STIG 低（第三类）

- Windows Server 2019 STIG 版本 2 发行版 4
- Windows Server 2016 STIG 版本 2 发行版 4
- Windows Server 2012 R2 MS STIG 版本 3 发行版 3
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 1 发行版 19

Windows STIG 中等（第二类）

- Windows Server 2019 STIG 版本 2 发行版 4
- Windows Server 2016 STIG 版本 2 发行版 4
- Windows Server 2012 R2 MS STIG 版本 3 发行版 3
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 1 发行版 19

Windows STIG 高（第一类）

- Windows Server 2019 STIG 版本 2 发行版 4
- Windows Server 2016 STIG 版本 2 发行版 4
- Windows Server 2012 R2 MS STIG 版本 3 发行版 3
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1

- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 1 发行版 19

2022 年第一季度更改 — 2022 年 2 月 8 日（无更改）：

在 2022 年第一季度发行版中，Windows 组件 STIGS 无更改。

2021 年第四季度更改 — 2021 年 12 月 20 日：

更新了 STIG 版本，并对 2021 年第四季度发行版应用了 STIG。

Windows STIG 低（第三类）

- Windows Server 2019 STIG 版本 2 发行版 3
- Windows Server 2016 STIG 版本 2 发行版 3
- Windows Server 2012 R2 MS STIG 版本 3 发行版 3
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 1 发行版 19

Windows STIG 中等（第二类）

- Windows Server 2019 STIG 版本 2 发行版 3
- Windows Server 2016 STIG 版本 2 发行版 3
- Windows Server 2012 R2 MS STIG 版本 3 发行版 3
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 1 发行版 19

Windows STIG 高（第一类）

- Windows Server 2019 STIG 版本 2 发行版 3
- Windows Server 2016 STIG 版本 2 发行版 3
- Windows Server 2012 R2 MS STIG 版本 3 发行版 3
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1

- Windows Firewall STIG 版本 2 发行版 1
- Internet Explorer 11 STIG 版本 1 发行版 19

2021 年第三季度更改 — 2021 年 9 月 30 日：

更新了 STIG 版本，并对 2021 年第三季度发行版应用了 STIG。

Windows STIG 低（第三类）

- Windows Server 2019 STIG 版本 2 发行版 2
- Windows Server 2016 STIG 版本 2 发行版 2
- Windows Server 2012 R2 MS STIG 版本 3 发行版 2
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 1 发行版 7
- Internet Explorer 11 STIG 版本 1 发行版 19

Windows STIG 中等（第二类）

- Windows Server 2019 STIG 版本 2 发行版 2
- Windows Server 2016 STIG 版本 2 发行版 2
- Windows Server 2012 R2 MS STIG 版本 3 发行版 2
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 1 发行版 7
- Internet Explorer 11 STIG 版本 1 发行版 19

Windows STIG 高（第一类）

- Windows Server 2019 STIG 版本 2 发行版 2
- Windows Server 2016 STIG 版本 2 发行版 2
- Windows Server 2012 R2 MS STIG 版本 3 发行版 2
- Microsoft .NET Framework 4.0 STIG 版本 2 发行版 1
- Windows Firewall STIG 版本 1 发行版 7
- Internet Explorer 11 STIG 版本 1 发行版 19

使用 Amazon Systems Manager 将 STIG 设置应用到您的实例

您可以使用 AWSEC2-ConfigureSTIG Systems Manager 命令文档将 STIG 设置应用于现有 EC2 实例。您必须从其更新的实例运行命令文档。命令文档根据其所运行实例的操作系统和配置应用相应的设置。

本页包含有关 AWSEC2-ConfigureSTIG 命令文档的详细信息，包括输入参数以及如何在 Systems Manager 控制台中或在 Amazon CLI 中使用 send-command 运行该文档。

AWSEC2-ConfigureSTIG 输入参数

您可以提供以下输入参数来指定命令文档应如何将 STIG 设置应用到您的实例。

Level（字符串，必填）

指定要应用的 STIG 严重性类别。有效值包括：

- 高
- 中
- 低

如果您未指定值，系统将默认该值为 High。

InstallPackages（字符串，可选 – 仅限 Linux）

如果值为 No，则脚本不安装任何其他软件包。如果值为 Yes，则脚本会安装实现最大合规性所需的其他软件包。默认值为 No。

SetDoDConsentBanner（字符串，可选 – 仅限 Linux）

如果值为 No，则当您连接到安装了 Linux STIG 脚本的实例时不会显示 DoD 同意横幅。如果值为 Yes，则当您连接到安装了 STIG Linux 脚本之一的实例时，会在您登录之前显示 DoD 同意横幅。您必须先确认该横幅，然后才能登录。默认值为 No。

有关同意横幅的示例，请参阅当您访问 DLA 文档服务网站时显示的 Disclaimer Department of Defense Privacy and Consent Notice。

运行 AWSEC2-ConfigureSTIG 命令文档

要运行 AWSEC2-ConfigureSTIG 文档，请按照适用于您的首选环境的步骤操作。

Console

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 从导航窗格中选择运行命令。这将显示您的账户中当前正在运行的命令的列表（如果适用）。
3. 选择 Run command（运行命令）。这将打开运行命令对话框，并显示您有权访问的命令文档列表。
4. 从命令文档列表中选择 AWSEC2-ConfigureSTIG。为了简化结果，您可以输入完整或部分文档名称。也可以按所有者、平台类型或标签进行筛选。

   当选择命令文档时，将在列表下方填充详细信息。
5. 从文档版本列表中选择 Default version at runtime。
6. 配置命令参数以定义 AWSEC2-ConfigureSTIG 如何安装脚本包并运行它来更新实例。有关参数的详细信息，请参阅 AWSEC2-ConfigureSTIG 输入参数。
7. 对于目标选择，请指定标签或者手动选择实例，以确定要在其上执行此操作的实例。

   > **Note**
   >
   > 如果手动选择实例，而且要查看的实例未包含在列表中，请参阅我的实例在哪里？以获得故障排除提示。

8. 要获得用来定义 Systems Manager Run Command 行为（例如速率控制）的其他参数，请输入从控制台运行命令中所述的值。
9. 选择运行。

   如果成功，命令文档将安装脚本并配置您的实例。如果命令执行失败，请查看 Systems Manager 命令输出，了解执行失败的详细原因。

Amazon CLI

示例 1：使用默认值运行

运行以下命令安装 STIG 脚本并使用默认值运行该脚本。有关输入参数的更多信息，请参阅AWSEC2-ConfigureSTIG 输入参数。

```
aws ssm send-command \
 --document-name "AWSEC2-ConfigureSTIG" \
```

```
 --instance-ids "i-1234567890abcdef0"'
```

示例 2：在您的实例上配置“中等”级别的 STIG 设置

运行以下命令安装 STIG 脚本，并将 `Level` 输入参数设置为 `Medium` 后运行该脚本。有关输入参数的更多信息，请参阅AWSEC2-ConfigureSTIG 输入参数。

```
aws ssm send-command \
 --document-name "AWSEC2-ConfigureSTIG" \
 --instance-ids "i-1234567890abcdef0"
 --parameters '{"Level":"Medium"}'
```

如果成功，命令文档将安装脚本并配置您的实例。如果命令执行失败，请查看命令输出，了解执行失败的详细原因。

PowerShell

示例 1：使用默认值运行

运行以下命令安装 STIG 脚本并使用默认值运行该脚本。有关输入参数的更多信息，请参阅AWSEC2-ConfigureSTIG 输入参数。

```
Send-SSMCommand -DocumentName "AWSEC2-ConfigureSTIG" -InstanceId
 "i-1234567890abcdef0"}
```

示例 2：在您的实例上配置“中等”级别的 STIG 设置

运行以下命令安装 STIG 脚本，并将 `Level` 输入参数设置为 `Medium` 后运行该脚本。有关输入参数的更多信息，请参阅AWSEC2-ConfigureSTIG 输入参数。

```
Send-SSMCommand -DocumentName "AWSEC2-ConfigureSTIG" -InstanceId
 "i-1234567890abcdef0" -Parameter @{'Level'='Medium'}
```

如果成功，命令文档将安装脚本并配置您的实例。如果命令执行失败，请查看命令输出，了解执行失败的详细原因。

在 EC2 实例上实现精确时钟和时间同步

对于许多服务器任务和进程来说，Amazon EC2 实例上准确一致的时间参考是非常重要的。系统日志中的时间戳在识别问题发生的时间和事件的时间顺序方面起着至关重要的作用。使用 Amazon CLI 或

Amazon SDK 从您的实例发送请求时，这些工具会以您的名义签署请求。如果实例的日期和时间设置不准确，可能会导致签名中的日期和请求的日期之间存在差异，进而导致 Amazon 拒绝请求。

为解决这一重要问题，Amazon 提供了 Amazon Time Sync Service，该服务可从所有 EC2 实例访问，并由各种 Amazon Web Services 服务 使用。该服务在每个 Amazon Web Services 区域 中使用一组与卫星连接的原子参考时钟，以提供准确的协调世界时（UTC）全球标准的当前时间读数。

为了获得最佳性能，建议在 EC2 实例上使用本地 Amazon Time Sync Service。要备份到实例上的本地 Amazon Time Sync Service，或者将 Amazon EC2 之外的资源连接到 Amazon Time Sync Service，您可以使用位于 `time.amazonaws.cn` 的公共 Amazon Time Sync Service。与本地 Amazon Time Sync Service 一样，公共 Amazon Time Sync Service 会自动涂抹添加到 UTC 中的任何闰秒。公共 Amazon Time Sync Service 由每个 Amazon Web Services 区域 中一组与卫星连接的原子参考时钟在全球范围内提供支持。

硬件数据包时间戳

您可以在实例上启用硬件数据包时间戳，为每个传入的网络数据包添加 64 位纳秒精度的时间戳。由于硬件数据包时间戳发生在硬件级别（在数据包到达内核、套接字或应用程序层之前），因此您可以绕过软件时间戳增加的任何延迟。硬件时间戳的基础参考时钟是 Amazon Time Sync Service PTP 硬件时钟。

优势

硬件数据包时间戳具有以下优点：

- 改进事件排序，这也可用于确定数据包到达 EC2 实例的实际顺序，确保公平的数据包处理。
- 衡量单向网络延迟。
- 与大多数本地解决方案相比，以更高的精度和准确度提高分布式事务速度。

先决条件和配置

要启用硬件数据包时间戳，您的实例必须满足以下先决条件：

- 必须是 Linux 实例。
- 满足支持 PTP 硬件时钟的要求。

有关配置说明，请参阅 GitHub 上弹性网络适配器 (ENA) 的 Linux 内核驱动程序系列页面上的 Hardware Packet Timestamping。

闰秒

闰秒的概念于 1972 年引入，是偶尔对 UTC 时间进行一秒钟的调整，将地球自转中的不规则性考虑在内，以适应国际原子时间（TAI）和太阳时（Ut1）之间的差异。为了代表客户管理闰秒，我们在 Amazon Time Sync Service 中设计了闰秒涂抹。有关更多信息，请参阅 Look Before You Leap – The Coming Leap Second and Amazon。

闰秒即将消失，我们完全支持在第 27 届国际计量大会上做出的在 2035 年或之前取消闰秒的决定。

为了支持这种过渡，当通过本地 NTP 连接或我们的公共 NTP 池（`time.amazonaws.cn`）访问 Amazon Time Sync Service 时，我们仍计划在闰秒事件期间涂抹时间。但是，PTP 硬件时钟不提供涂抹时间选项。如果出现闰秒，PTP 硬件时钟将按照 UTC 标准添加闰秒。在大多数情况下，闰秒涂抹和闰秒时间源是相同的。但是，由于其在闰秒事件中有所不同，因此我们不建议在闰秒事件期间在时间客户端配置中同时使用已涂抹和未涂抹的时间源。

主题

- 在 EC2 实例上设置时间参考，以使用本地 Amazon Time Sync Service
- 在您的 EC2 实例或任何连接互联网的设备上设置时间参考，以使用公共 Amazon Time Sync Service
- 比较您 Linux 实例的时间戳
- 更改实例的时区

相关资源

- Amazon 计算博客：It's About Time: Microsecond-Accurate Clocks on Amazon EC2 Instances
- Amazon Cloud Operations & Migrations Blog：使用 Amazon Time Sync Service 和 Amazon CloudWatch 管理 Amazon EC2 实例时钟精度 – 第 1 部分
- （Linux）https://chrony-project.org/

在 EC2 实例上设置时间参考，以使用本地 Amazon Time Sync Service

本地 Amazon Time Sync Service 使用网络时间协议（NTP），或者在支持的实例上提供本地精确时间协议（PTP）硬件时钟。PTP 硬件时钟支持 NTP 连接（Linux 和 Windows 实例）或者直接 PTP 连接

（仅限 Linux 实例）。NTP 连接和直接 PTP 连接使用相同的高精度时间源，但直接 PTP 连接比 NTP 连接更准确。与 Amazon Time Sync Service 的 NTP 连接支持闰秒涂抹，而与 PTP 硬件时钟的 PTP 连接不涂抹时间。有关更多信息，请参阅 闰秒。

您的实例可以访问本地 Amazon Time Sync Service，如下所示：

- 通过 NTP 在以下 IP 地址端点进行访问：
  - IPv4：`169.254.169.123`
  - IPv6：`fd00:ec2::123`（只能在基于 Nitro的实例上访问。）
- （仅限 Linux）通过直接 PTP 连接连接到本地 PTP 硬件时钟进行访问：
  - `PHC0`

Amazon Linux AMI、Windows AMI 和大多数合作伙伴 AMI 都将您的实例配置为默认使用 NTP IPv4 端点。这是大多数客户工作负载的建议设置。除非您希望使用 IPv6 端点或直接连接到 PTP 硬件时钟，否则从这些 AMI 启动的实例无需进行进一步配置。

NTP 连接和 PTP 连接不需要更改任何 VPC 配置，您的实例也不需要访问互联网。

注意事项

- 对于使用本地链路地址的服务，每秒数据包数（PPS）限制为 1024 个。此限制是 Route 53 Resolver DNS 查询、实例元数据服务（IMDS）请求、Amazon Time Service 网络时间协议（NTP）请求和 Windows 许可服务（适用于基于 Microsoft Windows 的实例）请求的总和。
- 只有 Linux 实例可以使用直接 PTP 连接来连接到本地 PTP 硬件时钟。Windows 实例使用 NTP 连接到本地 PTP 硬件时钟。

内容

- 连接到 Amazon Time Sync Service 的 IPv4 端点
- 连接到 Amazon Time Sync Service 的 IPv6 端点
- 连接到 PTP 硬件时钟

连接到 Amazon Time Sync Service 的 IPv4 端点

您的 AMI 可能已经默认配置了 Amazon Time Sync Service。否则请按照以下过程配置实例，从而通过 IPv4 端点使用本地 Amazon Time Sync Service。

有关故障排除问题的帮助，请参阅排查 Linux 实例上的 NTP 同步问题或排查 Windows 实例上的时间问题。

Amazon Linux

默认情况下，AL2023 以及最新版本的 Amazon Linux 2 配置为使用 Amazon Time Sync Service IPv4 端点。如果确认您的实例已经配置，则可以跳过下面的过程。

验证 chrony 是否已配置为使用 IPv4 端点

运行如下命令。输出中以 ^* 开始的行会指示首选的时间源。

```
chronyc sources -v | grep -F ^*
^* 169.254.169.123               3   4   377    14    +12us[+9653ns] +/-  290us
```

将 chrony 配置为连接到旧版 Amazon Linux 2 上的 IPv4 端点

1. 连接到您的实例并卸载 NTP 服务。

   ```
   [ec2-user ~]$ sudo yum erase 'ntp*'
   ```

2. 安装 chrony 软件包。

   ```
   [ec2-user ~]$ sudo yum install chrony
   ```

3. 使用任何文本编辑器（如 nano 或 vim）打开 /etc/chrony.conf 文件。在该文件中可能存在的其他 server 或 pool 语句前面添加以下行，然后保存更改：

   ```
   server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
   ```

4. 重启 chrony 守护程序 (chronyd)。

   ```
   [ec2-user ~]$ sudo service chronyd restart
   ```

   ```
   Starting chronyd:                                          [  OK  ]
   ```

   > **Note**
   >
   > 在 RHEL 和 CentOS (最高版本为 6) 上，服务名称是 chrony 而不是 chronyd。

5. 要配置 chronyd 为在每次系统启动时启动，请使用 chkconfig 命令。

```
[ec2-user ~]$ sudo chkconfig chronyd on
```

6. 确认 chrony 使用 169.254.169.123 IPv4 端点来同步时间。

```
[ec2-user ~]$ chronyc sources -v | grep -F ^*
```

输出中的 ^* 会指示首选的时间源。

```
^* 169.254.169.123               3   6    17    43    -30us[ -226us] +/-  287us
```

7. 验证 chrony 报告的时间同步指标。

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID    : A9FEA97B (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed Nov 22 13:18:34 2017
System time     : 0.000000626 seconds slow of NTP time
Last offset     : +0.002852759 seconds
RMS offset      : 0.002852759 seconds
Frequency       : 1.187 ppm fast
Residual freq   : +0.020 ppm
Skew            : 24.388 ppm
Root delay      : 0.000504752 seconds
Root dispersion : 0.001112565 seconds
Update interval : 64.4 seconds
Leap status     : Normal
```

Ubuntu

将 chrony 配置为连接到 Ubuntu 上的 IPv4 端点

1. 连接到您的实例并使用 apt 安装 chrony 软件包。

```
ubuntu:~$ sudo apt install chrony
```

> **Note**
>
> 如有必要，请先运行 `sudo apt update` 以更新您的实例。

2. 使用任何文本编辑器（如 nano 或 vim）打开 `/etc/chrony/chrony.conf` 文件。在该文件中已包含的任何其他 `server` 或 `pool` 语句前面添加以下行，然后保存您的更改：

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

3. 重新启动 chrony 服务。

```
ubuntu:~$ sudo /etc/init.d/chrony restart
```

```
Restarting chrony (via systemctl): chrony.service.
```

4. 确认 chrony 使用 169.254.169.123 IPv4 端点来同步时间。

```
ubuntu:~$ chronyc sources -v | grep -F ^*
```

输出中以 ^* 开始的行会指示首选的时间源。

```
^* 169.254.169.123               3   6    17    12    +15us[  +57us] +/-  320us
```

5. 验证 chrony 报告的时间同步指标。

```
ubuntu:~$ chronyc tracking
```

```
Reference ID    : 169.254.169.123 (169.254.169.123)
Stratum         : 4
Ref time (UTC)  : Wed Nov 29 07:41:57 2017
System time     : 0.000000011 seconds slow of NTP time
Last offset     : +0.000041659 seconds
RMS offset      : 0.000041659 seconds
Frequency       : 10.141 ppm slow
Residual freq   : +7.557 ppm
Skew            : 2.329 ppm
Root delay      : 0.000544 seconds
Root dispersion : 0.000631 seconds
Update interval : 2.0 seconds
```

```
Leap status     : Normal
```

SUSE Linux

从 SUSE Linux Enterprise Server 15 开始，chrony 是 NTP 的默认实现。

将 chrony 配置为连接到 SUSE Linux 上的 IPv4 端点

1. 使用任何文本编辑器（如 /etc/chrony.conf 或 vim）打开 nano 文件。
2. 确认该文件包含以下行：

   ```
   server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
   ```

   如果此行不存在，请添加它。
3. 注释掉任何其他服务器或池行。
4. 打开 YaST 并启用 chrony 服务。

Windows

从 2018 年 8 月的发行版开始，Windows AMI 默认情况下使用 Amazon Time Sync Service。从这些 AMI 启动的实例无需进一步配置，您可以跳过以下过程。

如果您使用的 AMI 默认未配置 Amazon Time Sync Service，则请先验证您当前的 NTP 配置。如果您的实例已经在使用 Amazon Time Sync Service的 IPv4 端点，则无需进行进一步配置。如果您的实例未使用 Amazon Time Sync Service，则请完成将 NTP 服务器更改为使用 Amazon Time Sync Service 的过程。

验证 NTP 配置

1. 从实例打开命令提示符窗口。
2. 通过键入以下命令获取当前 NTP 配置：

   ```
   w32tm /query /configuration
   ```

   该命令返回 Windows 实例的当前配置设置，并将显示您是否已连接到 Amazon Time Sync Service。
3. (可选) 通过键入以下命令获取当前配置的状态：

```
w32tm /query /status
```

该命令返回实例与 NTP 服务器同步的最后时间和轮询间隔等信息。

更改 NTP 服务器以使用 Amazon Time Sync Service

1. 从命令提示符窗口运行以下命令：

```
w32tm /config /manualpeerlist:169.254.169.123 /syncfromflags:manual /update
```

2. 使用以下命令验证新设置：

```
w32tm /query /configuration
```

在返回的输出中，请确认 NtpServer 显示 169.254.169.123 IPv4 端点。

Amazon Windows AMI 的默认 NTP 设置

亚马逊机器映像（AMI）通常符合现成的默认值，但在需要更改以便在 EC2 基础设施上正常工作的情况除外。以下设置已确定可在虚拟环境中正常工作，此外，还可将任何时间偏差保持在一秒的准确率内：

- 更新间隔 – 控制时间服务调整系统时间准确性的频率。Amazon 将更新间隔配置为每两分钟发生一次。
- NTP 服务器 – 从 2018 年 8 月版本开始，AMI 默认使用 Amazon Time Sync Service。此时间服务可通过位于 169.254.169.123 IPv4 端点的任何 Amazon Web Services 区域 访问。此外，0x9 标记指示时间服务充当客户端，并使用 SpecialPollInterval 来确定与所配置的时间服务器核查时间的频率。
- 类型 –“NTP”指示服务将充当独立的 NTP 客户端而不是作为域的一部分。
- 已启用和 InputProvider – 时间服务已启用并向操作系统提供时间。
- 特殊轮询间隔：对照所配置的 NTP 服务器，每 900 秒（即 15 分钟）检查一次。

> **Note**
>
> 对于 Windows Server 2025 AMI，`SpecialPollInterval` 值为 1024 秒而不是 900 秒。

注册表路径	键名称	数据
HKLM:\System\CurrentControlSet\services\w32time\Config	UpdateInterval	120
HKLM:\System\CurrentControlSet\services\w32time\Parameters	NtpServer	169.254.169.123,0x9
HKLM:\System\CurrentControlSet\services\w32time\Parameters	类型	NTP
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	启用	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	InputProvider	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	SpecialPollInterval	900（Windows Server 2016、2019 和 2022）或 1024（Windows Server 2025）

连接到 Amazon Time Sync Service 的 IPv6 端点

本节介绍如果您将实例配置为通过 IPv6 端点使用本地 Amazon Time Sync Service，在 连接到 Amazon Time Sync Service 的 IPv4 端点 中所述的步骤有何不同。它没有解释整个 Amazon Time Sync Service 配置流程。

IPv6 端点只能在基于 Nitro 的实例上访问。

不建议同时使用 IPv4 和 IPv6 端点条目。IPv4 和 IPv6 NTP 数据包来自您的实例的同一个本地服务器。没有必要同时配置 IPv4 和 IPv6 端点，这样做也不会提高实例上时间的准确性。

Linux

根据您使用的 Linux 发行版，当您到达编辑 `chrony.conf` 文件的步骤时，您将使用 Amazon Time Sync Service 的 IPv6 端点 (`fd00:ec2::123`) 而不是 IPv4 端点 (`169.254.169.123`)：

```
server fd00:ec2::123 prefer iburst minpoll 4 maxpoll 4
```

保存文件并验证 chrony 是否使用 `fd00:ec2::123` IPv6 端点来同步时间：

```
[ec2-user ~]$ chronyc sources -v
```

在输出中，如果您看到 `fd00:ec2::123` IPv6 端点，则配置完成。

Windows

当您到达将 NTP 服务器更改为使用 Amazon Time Sync Service 的步骤时，您将使用 Amazon Time Sync Service 的 IPv6 端点（`fd00:ec2::123`），而不是 IPv4 端点（`169.254.169.123`）：

```
w32tm /config /manualpeerlist:fd00:ec2::123 /syncfromflags:manual /update
```

验证您的新设置是否使用 `fd00:ec2::123` IPv6 端点同步时间：

```
w32tm /query /configuration
```

在输出中，请确认 `NtpServer` 显示 `fd00:ec2::123` IPv6 端点。

连接到 PTP 硬件时钟

PTP 硬件时钟是 Amazon Nitro System 的一部分，因此无需使用任何客户资源即可在支持的裸机和虚拟化 EC2 实例上直接访问。

PTP 硬件时钟的 NTP 端点与常规 Amazon Time Sync Service 的端点相同。如果您的实例具有 PTP 硬件时钟并且配置了 NTP 连接（到 IPv4 或 IPv6 端点），则您的实例时间将自动通过 NTP 从 PTP 硬件时钟获取。

对于 Linux 实例，您可以配置直接 PTP 连接，这将为您提供比 NTP 连接更准确的时间。Windows 实例仅支持与 PTP 硬件时钟的 NTP 连接。

要求

当满足以下要求时，PTP 硬件时钟在实例上可用：

- 支持的 Amazon Web Services 区域：美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、亚太地区（马来西亚）、亚太地区（泰国）、亚太地区（东京）和欧洲地区（斯德哥尔摩）
- 支持的 Local Zones：美国东部（纽约市）
- 支持的实例系列：
  - 通用型：M7a、M7g、M7i
  - 内存优化型：R7a、R7g、R7i
  - 存储优化型：I8g、I8ge
- （仅限 Linux）支持的操作系统上已安装 ENA 驱动程序版本 2.10.0 或更高版本。有关支持的操作系统的更多信息，请参阅 GitHub 上的驱动程序先决条件。

（仅限 Linux）配置与 PTP 硬件时钟的直接 PTP 连接

本节介绍如何使用直接 PTP 连接通过 PTP 硬件时钟将您的 Linux 实例配置为使用本地 Amazon Time Sync Service。需要在 `chrony` 配置文件中添加 PTP 硬件时钟的服务器条目。

要配置到 PTP 硬件时钟的直接 PTP 链接（仅限 Linux 实例）

1. 安装必备组件

   连接到您的 Linux 实例并执行以下操作：

   a. 安装弹性网络适配器（ENA）版本 2.10.0 或更高版本的 Linux kernel driver。

b. 启用 PTP 硬件时钟。

有关安装说明，请参阅 GitHub 上的 Linux kernel driver for Elastic Network Adapter (ENA) family。

2. 验证 ENA PTP 设备

验证 ENA PTP 硬件时钟设备是否显示在您的实例上。

```
[ec2-user ~]$ for file in /sys/class/ptp/*; do echo -n "$file: "; cat "$file/clock_name"; done
```

预期输出

```
/sys/class/ptp/ptp<index>: ena-ptp-<PCI slot>
```

其中：

- *index* 是内核注册的 PTP 硬件时钟索引。
- *PCI slot* 是 ENA 以太网控制器 PCI 插槽。这与 `lspci | grep ENA` 中所示的插槽相同。

输出示例

```
/sys/class/ptp/ptp0: ena-ptp-05
```

如果输出中没有 `ena-ptp-<PCI slot>`，则说明未正确安装 ENA 驱动程序。查看此安装驱动程序过程中的步骤 1。

3. 配置 PTP 符号链接

PTP 设备的名称通常命名为 `/dev/ptp0`、`/dev/ptp1`，以此类推，其索引取决于硬件初始化顺序。创建符号链接可确保无论索引如何更改，chrony 这类应用程序始终能引用正确的设备。

最新的 Amazon Linux 2023 AMI 包括一条创建 `/dev/ptp_ena` 符号链接的 udev 规则，该规则指向与 ENA 主机关联的正确 `/dev/ptp` 条目。

首先，通过运行以下命令，检查是否已存在符号链接。

```
[ec2-user ~]$ ls -l /dev/ptp*
```

输出示例

```
crw------- 1 root root 245, 0 Jan 31 2025 /dev/ptp0
lrwxrwxrwx 1 root root      4 Jan 31 2025 /dev/ptp_ena -> ptp0
```

其中：

- /dev/ptp*<index>* 是 PTP 设备的路径。
- /dev/ptp_ena 是常量符号链接，它指向同一 PTP 设备。

如果存在 /dev/ptp_ena 符号链接，请跳至此过程中的步骤 4。如果缺少符号链接，请执行以下操作之一：

a. 添加以下 udev 代码。

```
[ec2-user ~]$ echo "SUBSYSTEM==\"ptp\", ATTR{clock_name}==\"ena-ptp-*\",
 SYMLINK += \"ptp_ena\"" | sudo tee -a /etc/udev/rules.d/53-ec2-network-
interfaces.rules
```

b. 通过重启实例或通过运行以下命令来重新加载 udev 规则。

```
[ec2-user ~]$ sudo udevadm control --reload-rules && udevadm trigger
```

4. 配置 chrony

必须将 chrony 配置为使用 /dev/ptp_ena 符号链接而不是直接引用 /dev/ptp*<index>*。

a. 使用文本编辑器编辑 /etc/chrony.conf，并将以下行添加到文件中的任何位置。

```
refclock PHC /dev/ptp_ena poll 0 delay 0.000010 prefer
```

b. 重新启动 chrony。

```
[ec2-user ~]$ sudo systemctl restart chronyd
```

5. 验证 chrony 配置

验证 chrony 是否使用 PTP 硬件时钟来同步此实例上的时间。

```
[ec2-user ~]$ chronyc sources
```

预期输出

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
#* PHC0                          0   0    377    1   +2ns[ +1ns] +/-   5031ns
```

在返回的输出中，* 表示首选时间源。PHC0 对应于 PTP 硬件时钟。重启 chrony 后，您可能需要等待几秒钟才能显示星号。

在您的 EC2 实例或任何连接互联网的设备上设置时间参考，以使用公共 Amazon Time Sync Service

您可以将您的实例或任何连网设备（例如您的本地计算机或本地服务器）设置为使用公共 Amazon Time Sync Service，该服务可通过 `time.amazonaws.cn` 在互联网上进行访问。您可以使用公共 Amazon Time Sync Service 作为本地 Amazon Time Sync Service 的备份，并将 Amazon 以外的资源连接到 Amazon Time Sync Service。

> **Note**
>
> 为了获得最佳性能，建议在您的实例上使用本地 Amazon Time Sync Service，并且仅使用公共 Amazon Time Sync Service 作为备份。

使用适用于您的实例或设备操作系统的说明。

Linux

使用 chrony 或 ntpd 将您的 Linux 实例或设备设置为使用公共 Amazon Time Sync Service

1. 使用文本编辑器编辑 `/etc/chrony.conf`（如您使用 chrony）或 `/etc/ntp.conf`（如您使用 ntpd），如下所示：

   a. 为防止您的实例或设备尝试混用已涂抹和未涂抹的服务器，请移除或注释掉以 `server` 开头的行，但与本地 Amazon Time Sync Service 的任何现有连接除外。

> **Important**
>
> 如果您要将 EC2 实例设置为连接到公共 Amazon Time Sync Service，请勿删除以下将您的实例设置为连接到本地 Amazon Time Sync Service 的行。本地 Amazon Time Sync Service 是一种更直接的连接，可以提供更好的时钟准确度。公共 Amazon Time Sync Service 只能用作备份。
>
> ```
> server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
> ```

b. 添加以下行以连接到公共 Amazon Time Sync Service。

```
pool time.amazonaws.cn iburst
```

2. 使用以下命令之一重启进程守护程序。

- chrony

```
sudo service chronyd force-reload
```

- ntpd

```
sudo service ntp reload
```

macOS

将您的 macOS 实例或设备设置为使用公共 Amazon Time Sync Service

1. 打开 System Preferences (系统首选项)。
2. 选择 Date & Time（日期和时间），然后选择 Date & Time（日期和时间）选项卡。
3. 要进行更改，请选择锁定图标，并在出现提示时输入密码。
4. 对于 Set date and time automatically（自动设置日期和时间），请输入 **`time.amazonaws.cn`**。

Windows

将您的 Windows 实例或设备设置为使用公共 Amazon Time Sync Service

1. 打开 Control Panel（控制面板）。

2. 选择 Date and Time（日期和时间）图标。
3. 选择 Internet Time（互联网时间）选项卡。如果您的 PC 是域的一部分，此选项卡将不可用。在这种情况下，您的 PC 将与域控制器同步时间。您可以将控制器配置为使用公共 Amazon Time Sync Service。
4. 选择 Change settings（更改设置）。
5. 选中与互联网时间服务器同步复选框。
6. 在 Server（服务器）旁边输入 **`time.amazonaws.cn`**。

将您的 Windows Server 实例或设备设置为使用公共 Amazon Time Sync Service

- 按照 Microsoft 的说明更新注册表。

比较您 Linux 实例的时间戳

如果您使用的是 Amazon Time Sync Service，就可以将 Amazon EC2 Linux 实例上的时间戳与 ClockBound 进行比较，以确定事件的真实时间。ClockBound 可以测定 EC2 实例的时钟的准确性，并允许您检查一个给定的时间戳是早于还是晚于实例的当前时钟。在确定整个 EC2 实例中的事件与事务的顺序和一致性方面，这些信息十分重要，并且不会收到各个实例的地理位置的影响。

ClockBound 是一个开源守护进程和开源库。要了解关于 ClockBound 的详情，包括安装说明，请参阅 GitHub 上的 ClockBound。

ClockBound 仅支持 Linux 实例。

如果您使用与 PTP 硬件时钟的直接 PTP 连接，则您的时间进程守护程序（例如 chrony）将低估时钟误差范围。这是因为 PTP 硬件时钟不会像 NTP 那样将正确的误差范围信息传递给 chrony。因此，您的时钟同步进程守护程序假定时钟精确到 UTC，因此误差范围为 `0`。为了测量完整的误差范围，Nitro 系统会计算 PTP 硬件时钟的误差范围，并通过 ENA 驱动程序 `sysfs` 文件系统将其提供给您的 EC2 实例。您可以将其直接读取为以纳秒为单位的值。

检索 PTP 硬件时钟错误绑定

1. 首先使用以下命令之一获取 PTP 硬件时钟设备的正确位置。命令中的路径因用于启动实例的 AMI 而异。

   - 对于 Amazon Linux 2：

     ```
     cat /sys/class/net/eth0/device/uevent | grep PCI_SLOT_NAME
     ```

- 对于 Amazon Linux 2023：

```
cat /sys/class/net/ens5/device/uevent | grep PCI_SLOT_NAME
```

输出是 PCI 插槽名称，也就是 PTP 硬件时钟设备的位置。在此例中，该位置为 0000:00:03.0。

```
PCI_SLOT_NAME=0000:00:03.0
```

2. 要检索 PTP 硬件时钟错误绑定，请运行以下命令。请包括上一步中的 PCI 插槽名称。

```
cat /sys/bus/pci/devices/0000:00:03.0/phc_error_bound
```

输出是 PTP 硬件时钟的时钟误差范围（以纳秒为单位）。

在使用直接 PTP 连接到 PTP 硬件时钟时，要计算特定时间点的正确时钟误差范围，必须在 chrony 轮询 PTP 硬件时钟时添加来自 chrony 或 ClockBound 的时钟误差范围。有关测量和监控时钟准确度的更多信息，请参阅 Manage Amazon EC2 instance clock accuracy using Amazon Time Sync Service and Amazon CloudWatch – Part 1。

更改实例的时区

Amazon EC2 实例默认设置为 UTC（协调世界时）时区。您可以将实例上的时间更改为本地时区或网络中的其他时区。

请使用适用于您实例操作系统的说明。

Linux

> **Important**
>
> 此信息适用于 Amazon Linux。有关其他发布版本的信息，请参阅特定于该版本的文档。

在 Amazon Linux 上更改时区

1. 查看系统的当前时区设置。

```
[ec2-user ~]$ timedatectl
```

2. 列出可用的时区。

```
[ec2-user ~]$ timedatectl list-timezones
```

3. 设置选定的时区。

```
[ec2-user ~]$ sudo timedatectl set-timezone America/Vancouver
```

4. （可选）通过运行 timedatectl 命令，确认当前时区已更新为新时区。

```
[ec2-user ~]$ timedatectl
```

Windows

更改 Windows 实例上的时区

1. 从实例打开命令提示符窗口。
2. 确定将在实例上使用的时区。要获取时区的列表，请使用以下命令：

```
tzutil /l
```

该命令采用以下格式返回所有可用时区的列表：

```
display name
time zone ID
```

3. 查找要分配给该实例的时区 ID。
4. 示例：分配 UTC 时区：

```
tzutil /s "UTC"
```

示例：分配太平洋标准时间：

```
tzutil /s "Pacific Standard Time"
```

在更改 Windows 实例上的时间时，必须确保该时区在系统重启后仍然保留。否则，当实例重新启动时，它会恢复使用 UTC 时间。可通过添加 RealTimeIsUniversal 注册表项来保留时区设置。默认情况下，会在所有当前一代实例上设置此注册表项。若要验证是否设置了 RealTimeIsUniversal 注册表项，请参阅以下过程中的步骤 3。如果未设置该注册表项，请从头开始执行以下步骤。

设置 RealTimeIsUniversal 注册表项

1. 从实例打开命令提示符窗口。
2. 使用以下命令添加注册表项：

```
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /
v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

3. (可选) 验证该实例是否使用以下命令成功保存了该注册表项：

```
reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
\TimeZoneInformation" /s
```

此命令返回 TimeZoneInformation 注册表项的子项。您应在列表底部看到 RealTimeIsUniversal 项，类似于下文：

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation
    Bias                            REG_DWORD     0x1e0
    DaylightBias                    REG_DWORD     0xffffffc4
    DaylightName                    REG_SZ        @tzres.dll,-211
    DaylightStart                   REG_BINARY    00000300020002000000000000000000
    StandardBias                    REG_DWORD     0x0
    StandardName                    REG_SZ        @tzres.dll,-212
    StandardStart                   REG_BINARY    00000B0001000200000000000000000
    TimeZoneKeyName                 REG_SZ        Pacific Standard Time
    DynamicDaylightTimeDisabled     REG_DWORD     0x0
    ActiveTimeBias                  REG_DWORD     0x1a4
    RealTimeIsUniversal             REG_DWORD     0x1
```

管理 EC2 实例的设备驱动程序

设备驱动程序是与 Amazon EC2 实例的虚拟化硬件通信的软件组件。为防止系统错误、性能问题和其他意外行为，务必让驱动程序保持在最新状态。对于可能因使用情况对系统性能产生重大影响的驱动程

序（例如网络、显卡和存储设备驱动程序），则尤其如此。新的驱动程序版本可能包含缺陷修复或引入扩展功能，您可能想为当前运行的实例利用这些功能。

网络驱动程序

Linux 发行版可以在内核中加入弹性网络适配器（ENA）或 Elastic Fabric Adapter（EFA）。但是，不同发行版中实现内核驱动程序功能的时间可能会有所差异。

ENA 和 EFA Linux 内核驱动程序可从 Amazon Drivers GitHu 存储库中获得。有关更多信息和可用驱动程序的链接，请参阅 GitHub 上的 Amazon Drivers。

有关 ENA 驱动程序的更多信息，请参阅在 EC2 实例上使用 ENA 启用增强联网功能。有关 EFA 驱动程序的更多信息，请参阅本指南 适用于 Amazon EC2 上 AI/ML 和 HPC 工作负载的 Elastic Fabric Adapter 部分的入门主题。

要在 Windows 实例上安装或更新网络驱动程序，请参阅下列主题：

- 在 Windows 上安装 ENA 驱动程序
- 安装最新的 Amazon PV 驱动程序

  有关更多信息，请参阅 Windows 实例的半虚拟化驱动程序。

> **Note**
>
> EFA 在 Windows 实例上不受支持。

显卡驱动程序

要安装或更新显卡驱动程序，请参阅下列主题：

- EC2 实例的 AMD 驱动程序
- Amazon EC2 实例的 NVIDIA 驱动程序

存储设备驱动程序

要安装或更新存储驱动程序，请参阅下列主题：

- 若为 Linux 实例，请参阅《Amazon EBS User Guide》中的 Install or upgrade the NVMe driver。
- 若为 Windows 实例，请参阅Amazon NVMe 驱动程序。

EC2 实例的 AMD 驱动程序

已挂载 AMD GPU 的实例（如 G4ad 实例）必须安装相应的 AMD 驱动程序。根据您的要求，您可以使用带有预装驱动程序的 AMI，也可以从 Amazon S3 下载驱动程序。

要在挂载 NVIDIA GPU 的实例（例如，G4dn 实例）上安装 NVIDIA 驱动程序，则请参阅NVIDIA 驱动程序。

目录

- AMD Radeon Pro Software for Enterprise Driver
- 安装了 AMD 驱动程序的 AMI
- AMD 驱动程序下载

AMD Radeon Pro Software for Enterprise Driver

The AMD Radeon Pro Software for Enterprise Driver 旨在为专业级图形使用案例提供支持。使用驱动程序，您可以将实例配置为每个 GPU 两个 4K 显示。

支持的 API

- OpenGL、OpenCL
- Vulkan
- AMD 高级媒体框架
- 视频加速 API
- DirectX 9 及更高版本
- 微软硬件媒体平台变换

安装了 AMD 驱动程序的 AMI

Amazon 提供安装的 AMD 驱动程序随附的不同亚马逊机器映像（AMI）。打开使用 AMD 驱动程序的 Marketplace 产品。

AMD 驱动程序下载

如果您使用的不是安装了 AMD 驱动程序的 AMI，则可以下载 AMD 驱动程序并将其安装在您的实例上。仅以下操作系统版本支持 AMD 驱动程序：

- 内核版本 5.4 的 Amazon Linux 2
- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

这些下载仅供 Amazon 客户使用。下载即表明您同意仅将下载的软件用于开发在 AMD Radeon Pro V520 硬件上使用的 AMIs。安装软件时，您需要遵循 AMD 最终用户许可协议的条款。

在 Amazon Linux 2 Linux 实例上安装 AMD 驱动程序

1. 连接到 Linux 实例。
2. 在 Linux 实例上安装 Amazon CLI 并配置默认凭证。有关更多信息，请参阅 Amazon Command Line Interface 用户指南中的安装 Amazon CLI。

   > **Important**
   >
   > 用户或角色必须具有包含 AmazonS3ReadOnlyAccess 策略的授予权限。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Amazon 托管式策略：AmazonS3ReadOnlyAccess。

3. 安装内核 5.4

   ```
   $ sudo amazon-linux-extras disable kernel-5.10
   $ sudo amazon-linux-extras enable kernel-5.4
   $ sudo yum install -y kernel
   ```

4. 安装 gcc 和 make（如果尚未安装）。

   ```
   $ sudo yum install gcc make
   ```

5. 更新软件包缓存并获取实例软件包更新。

```
$ sudo amazon-linux-extras install epel -y
$ sudo yum update -y
```

6. 重启实例。

```
$ sudo reboot
```

7. 在实例重启后重新连接到实例。
8. 下载最新的 AMD 驱动程序。

```
$ aws s3 cp --recursive s3://ec2-amd-linux-drivers/latest/ .
```

9. 解压文件。

```
$ tar -xf amdgpu-pro-*rhel*.tar.xz
```

10. 切换到提取的驱动程序所在的文件夹。
11. 运行自我安装脚本以安装完整的图形堆栈。

```
$ ./amdgpu-pro-install -y --opencl=pal,legacy
```

12. 重启实例。

```
$ sudo reboot
```

13. 确认驱动程序正常运行。

```
$ sudo dmesg | grep amdgpu
```

该响应应当与以下内容相似：

```
Initialized amdgpu
```

在 Ubuntu Linux 实例上安装 AMD 驱动程序

1. 连接到 Linux 实例。
2. 更新软件包缓存并获取实例软件包更新。

```
$ sudo apt-get update --fix-missing && sudo apt-get upgrade -y
```

3. 安装 gcc 和 make（如果尚未安装）。

```
$ sudo apt install build-essential -y
```

4. 安装 Linux 固件和内核模块

```
$ sudo apt install linux-firmware linux-modules-extra-aws -y
```

5. 重启实例

```
$ sudo reboot
```

6. 在实例重启后重新连接到实例。
7. 安装 AMD Linux 驱动程序包

   - 对于 Ubuntu 20.04：

```
$ wget https://repo.radeon.com/.preview/afe3e25b8f1beff0bb312e27924d63b5/amdgpu-install/5.4.02.01/ubuntu/focal/amdgpu-install_5.4.02.01.50402-1_all.deb
$ sudo dpkg --add-architecture i386
$ sudo apt install ./amdgpu-install_5.4.02.01.50402-1_all.deb
```

   - 对于更高的 Ubuntu 版本，请访问 AMD Radeon™ 显卡 Linux® 驱动程序，下载最新版本的 Ubuntu 程序包并进行安装。

```
$ sudo apt install ./amdgpu-install_{version-you-downloaded}.deb
```

8. 运行自我安装脚本以安装完整的图形堆栈。

```
$ amdgpu-install --usecase=workstation --vulkan=pro -y
```

9. 重启实例。

```
$ sudo reboot
```

10. 确认驱动程序正常运行。

```
$ sudo dmesg | grep amdgpu
```

该响应应当与以下内容相似：

```
Initialized amdgpu
```

在 Windows 实例上安装 AMD 驱动程序

1. 连接到您的 Windows 实例并打开 PowerShell 窗口。
2. 在 Windows 实例上配置 Amazon Tools for Windows PowerShell 的默认凭证。有关更多信息，请参阅 Amazon Tools for Windows PowerShell 用户指南中的 Amazon Tools for PowerShell 入门

> **Important**
>
> 用户或角色必须具有包含 AmazonS3ReadOnlyAccess 策略的授予权限。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Amazon 托管式策略：AmazonS3ReadOnlyAccess。

3. 根据您的 Windows 版本设置密钥前缀：

   - Windows 10 和 Windows 11

     ```
     $KeyPrefix = "latest/AMD_GPU_WINDOWS10"
     ```

   - Windows Server 2016

     ```
     $KeyPrefix = "archives"
     ```

   - Windows Server 2019

     ```
     $KeyPrefix = "latest/AMD_GPU_WINDOWS_2K19" # use "archives" for Windows Server
      2016
     ```

   - Windows Server 2022

     ```
     $KeyPrefix = "latest/AMD_GPU_WINDOWS_2K22"
     ```

4. 使用以下 PowerShell 命令将驱动程序从 Amazon S3 下载到桌面。

```
$Bucket = "ec2-amd-windows-drivers"
$LocalPath = "$home\Desktop\AMD"
```

```
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
    $LocalFilePath = Join-Path $LocalPath $LocalFileName
    Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
    }
}
```

5. 解压下载的驱动程序文件并使用以下 PowerShell 命令运行安装程序。

```
Expand-Archive $LocalFilePath -DestinationPath "$home\Desktop\AMD\$KeyPrefix" -
Verbose
```

然后请检查新目录的内容。可以使用 `Get-ChildItem` PowerShell 命令检索目录名称。

```
Get-ChildItem "$home\Desktop\AMD\$KeyPrefix"
```

此输出应当类似于如下所示：

```
Directory: C:\Users\Administrator\Desktop\AMD\latest

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----       10/13/2021  12:52 AM                210414a-365562C-Retail_End_User.2
```

安装驱动程序：

```
pnputil /add-driver $home\Desktop\AMD\$KeyPrefix\*.inf /install /subdirs
```

6. 按照说明安装驱动程序并根据需要重启实例。
7. 要验证 GPU 是否正常工作，请检查设备管理器。您应看到“AMD Radeon Pro V520 MxGPU”作为显示适配器列出。
8. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

Amazon EC2 实例的 NVIDIA 驱动程序

已挂载 NVIDIA GPU 的实例（例如 P 系列或 G 系列实例类型）必须安装相应的 NVIDIA 驱动程序。根据实例类型，您可以下载公有 NVIDIA 驱动程序、从仅对 Amazon 客户可用的 Amazon S3 下载驱动程序或者使用预安装了驱动程序的 Amazon AMI。

要在挂载 AMD GPU 的实例（例如，G4ad 实例）上安装 AMD 驱动程序，则请参阅AMD 驱动程序。

主题

- NVIDIA 驱动程序的类型
- 按实例类型列出的可用驱动程序
- 安装选项
- 使用包含 NVIDIA 驱动程序的 AMI
- 安装 NVIDIA 公共驱动程序
- 安装 NVIDIA GRID 驱动程序（G7e、G6、Gr6、G6e、G6f、Gr6f、G5、G4dn 和 G3 实例）
- 安装 NVIDIA 游戏驱动程序（G7e、G6、G6e、G5 和 G4dn 实例）

NVIDIA 驱动程序的类型

以下是可与基于 GPU 的实例一起使用的 NVIDIA 驱动程序的主要类型。

Tesla 驱动程序

这些驱动程序主要用于将 GPU 用于计算任务的计算工作负载，例如适用于机器学习的并行浮点计算，以及适用于高性能计算应用的快速傅里叶变换。

GRID 驱动程序

这些驱动程序经过认证，可针对渲染 3D 模型或高分辨率视频等内容的专业可视化应用程序，提供最佳性能。您可以将 GRID 驱动程序配置为支持两种模式。Quadro 虚拟工作站的每个 GPU 可支持四个 4K 显示器。GRID vApps 提供 RDSH App 托管功能。

Gaming 驱动程序

这些驱动程序包含针对游戏的优化，并经常更新以提供性能增强。它们支持每个 GPU 单个 4K 显示器。

已配置模式

在 Windows 上，Tesla 驱动程序配置为在 Tesla Compute Cluster (TCC) 模式下运行。GRID 和 Gaming 驱动程序配置为在 Windows Display Driver Model (WDDM) 模式下运行。在 TCC 模式下，该卡专用于计算工作负载。在 WDDM 模式下，该卡同时支持计算和图形工作负载。

NVIDIA 控制面板

GRID 和游戏驱动程序支持 NVIDIA 控制面板。Tesla 驱动程序不支持此控制面板。

支持 Tesla、GRID 和游戏驱动程序的 API

- OpenCL、OpenGL 和 Vulkan
- NVIDIA CUDA 和相关库（例如，cuDNN、TensorRT、nvJPEG 和 cuBLAS）
- 用于视频编码的 NVENC 和用于视频解码的 NVDEC
- 仅限 Windows 的 API：DirectX、Direct2D、DirectX Video Acceleration、DirectX Raytracing

按实例类型列出的可用驱动程序

下表总结了各种 GPU 实例类型支持的 NVIDIA 驱动程序。

实例类型	Tesla 驱动程序	GRID 驱动程序	Gaming 驱动程序
G3	是	是	否
G4dn	是	是	是
G5	支持	是	是
G5g	是[1]	否	否
G6	支持	是	是
G6e	支持	是	是
G6f	否	是	否
Gr6	支持	是	否
Gr6f	否	是	否
G7e	支持	是	是

实例类型	Tesla 驱动程序	GRID 驱动程序	Gaming 驱动程序
P3	是	否	否
P4d	是	否	否
P4de	是	否	否
P5	是	否	否
P5e	是	否	否
P5en	是	否	否
P6-B200	是	否	否
P6e-GB200	是	否	否
P6-B300	是	否	否

[1] 此 Tesla 驱动程序还支持特定于 ARM64 平台的优化图形应用程序

安装选项

使用以下选项之一获取 GPU 实例所需的 NVIDIA 驱动程序。

Options

1. 使用包含 NVIDIA 驱动程序的 AMI
2. 安装 NVIDIA 公共驱动程序
3. 安装 NVIDIA GRID 驱动程序（G7e、G6、Gr6、G6e、G6f、Gr6f、G5、G4dn 和 G3 实例）
4. 安装 NVIDIA 游戏驱动程序（G7e、G6、G6e、G5 和 G4dn 实例）

使用包含 NVIDIA 驱动程序的 AMI

Amazon 和 NVIDIA 随已安装的 NVIDIA 驱动程序提供了不同的亚马逊机器映像（AMI）。

- 具备 Tesla 公共驱动程序的 Marketplace 产品
- 具备 GRID 驱动程序的 Marketplace 产品

- 具备 Gaming 驱动程序的 Marketplace 产品

要查看取决于操作系统（OS）平台的注意事项，请选择适用于您的 AMI 的选项卡。

Linux

要使用这些 AMI 之一更新已安装的驱动程序版本，您必须从实例中卸载 NVIDIA 软件包以避免版本冲突。使用此命令卸载 NVIDIA 软件包：

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

Amazon 提供的 CUDA 工具包安装包对 NVIDIA 驱动程序有依赖性。卸载 NVIDIA 软件包也会删除 CUDA 工具包。必须在安装 NVIDIA 驱动程序之后重新安装 CUDA 工具包。

Windows

如果您使用 Amazon Web Services Marketplace 产品之一创建自定义 Windows AMI，则 AMI 必须是使用 Windows Sysprep 创建的标准化映像，以确保 GRID 驱动程序正常工作。有关更多信息，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。

安装 NVIDIA 公共驱动程序

如果使用包含 NVIDIA 驱动程序的 AMI中所述的 Amazon Web Services Marketplace AMI 不适合您的使用案例，则可以安装公共驱动程序和自带许可。安装选项包含以下内容：

- 选项 1：仅安装驱动程序
- 选项 2：使用 CUDA 工具包安装（建议用于 Linux 发行版）

P6-B200 和 P6-B300 实例类型注意事项

P6-B200 和 P6-B300 平台的独特之处在于它们将 Mellanox ConnectX 网络接口卡 (NIC) 作为 PCIe 设备公开给实例。这些 NIC 不充当典型的网络接口，而是充当 NVSwitch 网桥，提供控制路径来初始化和配置 NVFabric（即 GPU 互连的 NVLink 拓扑）。

要完全初始化系统，NVIDIA Fabric Manager 必须配置 `NVFabric` 并建立 NVSwitch 拓扑。这使得 InfiniBand 内核模块能够与 Mellanox ConnectX NIC 通信。

NVIDIA Fabric Manager 包含在 CUDA 工具包中。对于此实例类型，我们建议使用 选项 2：使用 CUDA 工具包安装。

选项 1：仅安装驱动程序

要安装特定驱动程序，请登录您的实例并从 http://www.nvidia.com/Download/Find.aspx 下载适合实例类型的 64 位 NVIDIA 公共驱动程序。对于产品类型、产品系列和产品，请使用下表中所示的选项。

然后，按照《NVIDIA 驱动程序安装指南》中的本地存储库安装说明进行操作。

> **Note**
>
> P6-B200 和 P6-B300 实例类型需要安装和配置与 NVIDIA CUDA Toolkit 捆绑在一起的附加包。有关更多信息，请参阅 选项 2：使用 CUDA 工具包安装 中有关 Linux 发行版的说明。

实例	产品类型	产品系列	产品	最低驱动程序版本
G3	Tesla	M-Class	M60	--
G4dn	Tesla	T 系列	T4	--
G5	Tesla	A 系列	A10	470.00 或更高版本
G5g[1]	Tesla	T 系列	T4G	470.82.01 或更高版本
G6	Tesla	L 系列	L4	525.0 或更高版本
G6e	Tesla	L 系列	L40S	535.0 或更高版本
Gr6	Tesla	L 系列	L4	525.0 或更高版本
G7e	Tesla	RTX 系列	RTX PRO 6000 Blackwell	575.0 或更高版本

实例	产品类型	产品系列	产品	最低驱动程序版本
P3	Tesla	V 系列	V100	--
P4d	Tesla	A 系列	A100	--
P4de	Tesla	A 系列	A100	--
P5	Tesla	H 系列	H100	530 或更高版本
P5e	Tesla	H 系列	H200	550 或更高版本
P5en	Tesla	H 系列	H200	550 或更高版本
P6-B200[2]	Tesla	HGX 系列	B200	570 或更高版本
P6e-GB200	Tesla	HGX 系列	B200	570 或更高版本
P6-B300[2]	Tesla	HGX 系列	B300	580 或更高版本

[1] G5g 实例的操作系统是 Linux aarch64。

[2] 对于 P6-B200 和 P6-B300 实例类型，配置 NVIDIA Fabric Manager 有额外的安装要求。

选项 2：使用 CUDA 工具包安装

安装说明因操作系统而稍有不同。要使用 NVIDIA CUDA 工具包在您的实例上安装公共驱动程序，请按照有关您的实例操作系统的说明进行操作。对于此处未显示的实例操作系统，请按照 NVIDIA 开发者网站上有关您的操作系统和实例类型架构的说明进行操作。有关更多信息，请参阅 CUDA 工具包下载。

有关实例类型架构或其他规范，请参阅 Amazon EC2 实例类型参考中的加速计算规范。

Amazon Linux 2023

本节介绍在 Amazon Linux 2023 实例上安装的 NVIDIA CUDA 工具包。本节中的命令示例基于 x86_64 架构。

有关 `arm64-sbsa` 命令，请参阅 CUDA 工具包下载并选择适用于您的发行版的选项。您做出最终选择之后会显示说明。

先决条件

在安装工具包和驱动程序之前，请运行以下命令，以确保安装了正确版本的内核标头和开发包。

```
[ec2-user ~]$ sudo dnf install kernel-devel-$(uname -r) kernel-headers-$(uname -r) -y
```

下载工具包和驱动程序

选择要用于您的实例的安装类型，然后按照相关步骤进行操作。

AL2023 NVIDIA repository

Amazon Linux 2023 通过 Amazon 维护的专用存储库提供 NVIDIA GPU 驱动程序和 CUDA 工具包软件包。Amazon 使用 AL2023 候选版本对该存储库进行认证，并通过 Amazon Linux 安全中心提供安全公告。建议 AL2023 实例使用此选项，因为它可以简化安装，并通过标准的 `dnf update` 工作流更新驱动程序。

有关说明，请参阅《Amazon Linux 2023 用户指南》中的 NVIDIA 驱动程序。

RPM local installation

您可以按照以下说明，将 CUDA 工具包安装程序存储库捆绑包下载到您的实例，然后提取并注册指定的捆绑包。

要查看 NVIDIA 开发者网站上的说明，请参阅 CUDA 工具包下载。

```
[ec2-user ~]$ wget https://developer.download.nvidia.com/compute/cuda/13.0.0/
local_installers/cuda-repo-amzn2023-13-0-local-13.0.0_580.65.06-1.x86_64.rpm
[ec2-user ~]$ sudo rpm -i cuda-repo-amzn2023-13-0-
local-13.0.0_580.65.06-1.x86_64.rpm
```

RPM network installation

您可以按照以下说明，在实例上使用程序包管理器注册 CUDA 存储库。运行安装步骤时，程序包管理器仅会下载所需的程序包。

要查看 NVIDIA 开发者网站上的说明，请参阅 CUDA 工具包下载。

```
[ec2-user ~]$ sudo dnf config-manager --add-repo https://
developer.download.nvidia.com/compute/cuda/repos/amzn2023/x86_64/cuda-amzn2023.repo
```

RPM 本地安装和 RPM 网络安装的其余步骤相同。

1. 完成 CUDA 工具包安装

```
[ec2-user ~]$ sudo dnf clean all
[ec2-user ~]$ sudo dnf install cuda-toolkit -y
```

2. 安装驱动程序的开放内核模块变体

```
[ec2-user ~]$ sudo dnf module install nvidia-driver:open-dkms -y
```

3. 安装 GPUDirect Storage 和 Fabric 管理器

```
[ec2-user ~]$ sudo dnf install nvidia-gds -y
[ec2-user ~]$ sudo dnf install nvidia-fabric-manager -y
```

4. 启用 Fabric 管理器和驱动程序持久性

```
[ec2-user ~]$ sudo systemctl enable nvidia-fabricmanager
[ec2-user ~]$ sudo systemctl enable nvidia-persistenced
```

5. （仅 P6-B200 和 P6-B300）这些实例类型需要安装和配置与 NVIDIA CUDA Toolkit 捆绑在一起的附加包。

   1. 安装 NVIDIA Link Subnet Manager 和 `ibstat`。

      ```
      [ec2-user ~]$ sudo dnf install nvlink5
      ```

   2. 启用启动时自动加载 Infiniband 模块。

      ```
      [ec2-user ~]$ echo "ib_umad" | sudo tee -a /etc/modules-load.d/modules.conf
      ```

6. 重启实例

```
[ec2-user ~]$ sudo reboot
```

Ubuntu 24.04

本节介绍在 Ubuntu 24.04 实例上安装的 NVIDIA CUDA 工具包。本节中的命令示例基于 x86_64 架构。

有关 `arm64-sbsa` 命令，请参阅 CUDA 工具包下载并选择适用于您的发行版的选项。您做出最终选择之后会显示说明。

先决条件

在安装工具包和驱动程序之前，请运行以下命令，以确保安装了正确版本的内核标头和开发包。

```
$ apt install linux-headers-$(uname -r)
```

下载工具包和驱动程序

选择要用于您的实例的安装类型，然后按照相关步骤进行操作。

Deb local installation

您可以按照以下说明，将 CUDA 工具包安装程序存储库捆绑包下载到您的实例，然后提取并注册指定的捆绑包。

要查看 NVIDIA 开发者网站上的说明，请参阅 CUDA 工具包下载。

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/
x86_64/cuda-ubuntu2404.pin
$ sudo mv cuda-ubuntu2404.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/13.0.0/
local_installers/cuda-repo-ubuntu2404-13-0-local_13.0.0-580.65.06-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2404-13-0-local_13.0.0-580.65.06-1_amd64.deb
$ sudo cp /var/cuda-repo-ubuntu2404-13-0-local/cuda-*-keyring.gpg /usr/share/
keyrings/
```

Deb network installation

您可以按照以下说明，在实例上使用程序包管理器注册 CUDA 存储库。运行安装步骤时，程序包管理器仅会下载所需的程序包。

要查看 NVIDIA 开发者网站上的说明，请参阅 CUDA 工具包下载。

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/
cuda-keyring_1.1-1_all.deb
$ sudo dpkg -i cuda-keyring_1.1-1_all.deb
```

本地安装和网络安装的其余步骤相同。

1. 完成 CUDA 工具包安装

```
$ sudo apt update
```

```
$ sudo apt install cuda-toolkit -y
```

2. 安装驱动程序的开放内核模块变体

```
$ sudo apt install nvidia-open -y
```

3. 安装 GPUDirect Storage 和 Fabric 管理器

```
$ sudo apt install nvidia-gds -y
$ sudo apt install nvidia-fabricmanager -y
```

4. 启用 Fabric 管理器和驱动程序持久性

```
$ sudo systemctl enable nvidia-fabricmanager
$ sudo systemctl enable nvidia-persistenced
```

5. （仅 P6-B200 和 P6-B300）这些实例类型需要安装和配置与 NVIDIA CUDA Toolkit 捆绑在一起的附加包。

   1. 安装最新的 Infiniband 特定设备驱动程序和诊断实用程序。

      ```
      $ sudo apt install linux-modules-extra-$(uname -r) -y
      $ sudo apt install infiniband-diags -y
      ```

   2. 安装 NVIDIA Link Subnet Manager。

      ```
      $ sudo apt install nvlsm -y
      ```

6. 重启实例

```
sudo reboot
```

7. 更新您的路径并添加以下环境变量。

```
$ export PATH=${PATH}:/usr/local/cuda-13.0/bin
$ export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/cuda-13.0/lib64
```

Windows 操作系统

要在 Windows 上安装 NVIDIA 驱动程序，请执行以下步骤：

1. 打开在其中下载驱动程序的文件夹，然后启动安装文件。按照说明安装驱动程序并根据需要重启实例。
2. 使用设备管理器禁用标有警告图标的名为 Microsoft Basic Display Adapter 的显示适配器。安装这些 Windows 功能：Media Foundation 和 Quality Windows Audio Video Experience。

   > **Important**
   >
   > 切勿禁用名为 Microsoft Remote Display Adapter 的显示适配器。如果禁用 Microsoft Remote Display Adapter，连接可能会中断，并且可能无法在实例重启后成功连接到实例。

3. 检查设备管理器以验证 GPU 正常工作。
4. 为实现 GPU 的最佳性能，请完成优化 Amazon EC2 实例上的 GPU 设置中的优化步骤。

安装 NVIDIA GRID 驱动程序（G7e、G6、Gr6、G6e、G6f、Gr6f、G5、G4dn 和 G3 实例）

这些下载仅供 Amazon 客户使用。一旦开始下载，即表示您同意按照 NVIDIA GRID 云终端用户许可协议（EULA）中提及的 Amazon 解决方案的要求，仅将下载的软件用来开发 AMI，以用于 NVIDIA L4、NVIDIA L40S、NVIDIA A10G、NVIDIA Tesla T4 或 NVIDIA Tesla M60 硬件。可使用 GRID 驱动程序在 Amazon 环境中创建和使用 AMI。安装软件后，您将受 NVIDIA GRID 云最终用户许可协议的条款约束。有关适用于您操作系统的 NVIDIA GRID 驱动程序版本的信息，请参阅 NVIDIA 网站上的 NVIDIA Virtual GPU (vGPU) Software。

注意事项

- 对于 Linux，G7e 实例需要 GRID 19.1 或更高版本，对于 Windows，需要 19.4（582.16）或更高版本。
- G6f 和 Gr6f 实例需要 GRID 18.4 至 GRID 19.5 版本。
- G6e 实例需要 GRID 17.4 或更高版本。
- G6 和 Gr6 实例需要 GRID 17.1 或更高版本。
- G5 实例需要 GRID 13.1 或更高版本（或 GRID 12.4 或更高版本）。
- 要让 GRID 许可正常运行，G3 实例需要 Amazon 提供的 DNS 解析。
- IMDSv2 仅在 14.0 或更高版本的 NVIDIA 驱动程序上受支持。

- 对于 Windows 实例，如果您使用自定义 Windows AMI 启动实例，则 AMI 必须是使用 Sysprep 创建的标准化映像，以确保 GRID 驱动程序正常工作。有关更多信息，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。
- GRID 17.0 及更高版本不支持 Windows Server 2019。
- GRID 14.2 及更高版本不支持 Windows Server 2016。
- G3 实例不支持 GRID 17.0 及更高版本。
- 对于 Linux 实例，如果 NVIDIA 安装程序失败并显示错误消息，则可能需要安装或更新软件包（例如 gcc）。具体情况取决于操作系统和内核的版本。有关更多信息，请参阅 NVIDIA 企业支持门户。

先决条件

- (Linux) 验证 Amazon CLI 是否已安装在您的实例上并配置了默认凭证。有关更多信息，请参阅《Amazon Command Line Interface 用户指南》中的安装 Amazon CLI。
- (Windows) 在您的实例上配置 Amazon Tools for Windows PowerShell 的默认凭证。有关更多信息，请参阅《Amazon Tools for PowerShell 用户指南》中的 Amazon Tools for Windows PowerShell 入门。
- 用户或角色必须具有包含 AmazonS3ReadOnlyAccess 策略的授予权限。

Amazon Linux 2023

在实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo dnf update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo dnf install gcc make
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 安装内核标头软件包。

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra
```

6. 使用以下命令下载 GRID 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ . --
region us-east-1 --no-sign-request
```

此存储桶中存储了多个版本的 GRID 驱动程序。您可以使用以下命令查看所有可用的版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/  --region us-
east-1 --no-sign-request
```

7. 使用以下命令添加权限以运行驱动程序安装实用程序。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所示运行自安装脚本，安装您下载的 GRID 驱动程序。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

9. 确认驱动程序正常运行。以下命令的响应列出已安装的 NVIDIA 驱动程序版本和有关 GPU 的详细信息。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 实例上使用 NVIDIA vGPU 软件版本 14.x 或更高版本，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

12. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

    a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

    b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

Amazon Linux 2

在实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install gcc make
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 为运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 使用以下命令下载 GRID 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ . --
region us-east-1 --no-sign-request
```

此存储桶中存储了多个版本的 GRID 驱动程序。您可以使用以下命令查看所有可用的版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/  --region us-
east-1 --no-sign-request
```

7. 使用以下命令添加权限以运行驱动程序安装实用程序。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所示运行自安装脚本，安装您下载的 GRID 驱动程序。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

如果您使用的是具有 5.10 内核版本的 Amazon Linux 2，请使用以下命令安装 GRID 驱动程序。

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

9. 确认驱动程序正常运行。以下命令的响应列出已安装的 NVIDIA 驱动程序版本和有关 GPU 的详细信息。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 实例上使用 NVIDIA vGPU 软件版本 14.x 或更高版本，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

12. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

    a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

    b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

CentOS 7 和 Red Hat Enterprise Linux 7

在实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 为您运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 禁用 NVIDIA 显卡的 nouveau 开源驱动程序。

   a. 将 nouveau 添加到 /etc/modprobe.d/blacklist.conf 黑名单文件。复制下面的代码块并将其粘贴到终端中。

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   b. 编辑 /etc/default/grub 文件并添加以下行：

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

   c. 重新生成 Grub 配置。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 使用以下命令下载 GRID 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ . --
region us-east-1 --no-sign-request
```

此存储桶中存储了多个版本的 GRID 驱动程序。您可以使用以下命令查看所有可用的版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/  --region us-
east-1 --no-sign-request
```

8. 使用以下命令添加权限以运行驱动程序安装实用程序。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

9. 如下所示运行自安装脚本，安装您下载的 GRID 驱动程序。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

10. 确认驱动程序正常运行。以下命令的响应列出已安装的 NVIDIA 驱动程序版本和有关 GPU 的详细信息。

```
[ec2-user ~]$ nvidia-smi -q | head
```

11. 如果您在 G4dn、G5 或 G5g 实例上使用 NVIDIA vGPU 软件版本 14.x 或更高版本，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

12. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

13. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

c. 安装 GUI 桌面/工作站程序包。

```
[ec2-user ~]$ sudo yum groupinstall -y "Server with GUI"
```

CentOS Stream 8 和 Red Hat Enterprise Linux 8

在实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 为您运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
devel-$(uname -r)
```

6. 使用以下命令下载 GRID 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ . --
region us-east-1 --no-sign-request
```

此存储桶中存储了多个版本的 GRID 驱动程序。您可以使用以下命令查看所有可用的版本。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/  --region us-
east-1 --no-sign-request
```

7. 使用以下命令添加权限以运行驱动程序安装实用程序。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

8. 如下所示运行自安装脚本，安装您下载的 GRID 驱动程序。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

   系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

9. 确认驱动程序正常运行。以下命令的响应列出已安装的 NVIDIA 驱动程序版本和有关 GPU 的详细信息。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 实例上使用 NVIDIA vGPU 软件版本 14.x 或更高版本，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

12. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

    a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

    b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

    c. 安装 GUI 工作站程序包。

```
[ec2-user ~]$ sudo dnf groupinstall -y workstation
```

Rocky Linux 8

在 Linux 实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

   ```
   [ec2-user ~]$ sudo yum update -y
   ```

2. 安装 gcc 和 make（如果尚未安装）。

   ```
   [ec2-user ~]$ sudo yum install -y gcc make
   ```

3. 重启实例以加载最新内核版本。

   ```
   [ec2-user ~]$ sudo reboot
   ```

4. 重启之后重新连接到实例。
5. 为您运行的内核版本安装内核标头软件包。

   ```
   [ec2-user ~]$ sudo dnf install -y elfutils-libelf-devel libglvnd-devel kernel-
   devel-$(uname -r)
   ```

6. 使用以下命令下载 GRID 驱动程序安装实用程序：

   ```
   [ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ . --
   region us-east-1 --no-sign-request
   ```

   此存储桶中存储了多个版本的 GRID 驱动程序。您可以使用以下命令查看所有可用的版本。

   ```
   [ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/  --region us-
   east-1 --no-sign-request
   ```

7. 使用以下命令添加权限以运行驱动程序安装实用程序。

   ```
   [ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
   ```

8. 如下所示运行自安装脚本，安装您下载的 GRID 驱动程序。例如：

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

9. 确认驱动程序正常运行。以下命令的响应列出已安装的 NVIDIA 驱动程序版本和有关 GPU 的详细信息。

```
[ec2-user ~]$ nvidia-smi -q | head
```

10. 如果您在 G4dn、G5 或 G5g 实例上使用 NVIDIA vGPU 软件版本 14.x 或更高版本，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

11. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

12. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

    a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

    b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

Ubuntu 和 Debian

在实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
$ sudo apt-get update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
$ sudo apt-get install -y gcc make
```

3. （Ubuntu）升级 linux-aws 程序包以接收最新版本。

```
$ sudo apt-get upgrade -y linux-aws
```

（Debian）升级程序包以接收最新版本。

```
$ sudo apt-get upgrade -y
```

4. 重启实例以加载最新内核版本。

```
$ sudo reboot
```

5. 重启之后重新连接到实例。
6. 为您当前运行的内核版本安装内核标头软件包。

```
$ sudo apt-get install -y linux-headers-$(uname -r) linux-modules-extra-$(uname -r)
```

7. 禁用 NVIDIA 显卡的 nouveau 开源驱动程序。

   a. 将 nouveau 添加到 /etc/modprobe.d/blacklist.conf 黑名单文件。复制下面的代码块并将其粘贴到终端中。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   b. 编辑 /etc/default/grub 文件并添加以下行：

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

   c. 重新生成 Grub 配置。

```
$ sudo update-grub
```

8. 使用以下命令下载 GRID 驱动程序安装实用程序：

```
$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ . --region us-east-1
 --no-sign-request
```

此存储桶中存储了多个版本的 GRID 驱动程序。您可以使用以下命令查看所有可用的版本。

```
$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/  --region us-east-1 --no-
sign-request
```

9. 使用以下命令添加权限以运行驱动程序安装实用程序。

```
$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 如下所示运行自安装脚本，安装您下载的 GRID 驱动程序。例如：

```
$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

11. 确认驱动程序正常运行。以下命令的响应列出已安装的 NVIDIA 驱动程序版本和有关 GPU 的详细信息。

```
$ nvidia-smi -q | head
```

12. 如果您在 G4dn、G5 或 G5g 实例上使用 NVIDIA vGPU 软件版本 14.x 或更高版本，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

13. 重启实例。

```
$ sudo reboot
```

14. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

c. 安装 GUI 桌面/工作站程序包。

```
$ sudo apt-get install -y lightdm ubuntu-desktop
```

Windows 操作系统

在 Windows 实例上安装 NVIDIA GRID 驱动程序

1. 连接到您的 Windows 实例并打开 PowerShell 窗口。
2. 使用以下 PowerShell 命令，将驱动程序和 NVIDIA GRID Cloud 最终用户许可协议从 Amazon S3 下载到您的桌面。

```
$Bucket = "ec2-windows-nvidia-drivers"
$KeyPrefix = "latest"
$LocalPath = "$home\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region cn-
north-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
$LocalFilePath = Join-Path $LocalPath $LocalFileName
Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region cn-north-1
}
}
```

此存储桶中存储了多个版本的 NVIDIA GRID 驱动程序。您可以通过删除 `-KeyPrefix $KeyPrefix` 选项下载桶中的所有可用 Windows 版本。有关适用于您操作系统的 NVIDIA GRID 驱动程序版本的信息，请参阅 NVIDIA 网站上的 NVIDIA Virtual GPU (vGPU) Software。

从 GRID 版本 11.0 开始，您可以对 G3 和 G4dn 实例使用 `latest` 下的驱动程序。我们不会将 11.0 之后的版本添加到 `g4/latest` 中，但会将版本 11.0 和特定于 G4dn 的早期版本保留在 `g4/latest` 下。

G5 实例需要 GRID 13.1 或更高版本（或 GRID 12.4 或更高版本）。

3. 导航到桌面，双击安装文件以启动它 (选择与您的实例操作系统版本对应的驱动程序版本)。按照说明安装驱动程序并根据需要重启实例。要验证 GPU 是否正常工作，请检查设备管理器。
4. （可选）使用以下命令可在控制面板中禁用授权页面以防止用户意外更改产品类型（默认情况下，将启用 NVIDIA GRID 虚拟工作站）。有关更多信息，请参阅 GRID 许可用户指南。

   PowerShell

   运行以下 PowerShell 命令以创建注册表值，从而在控制面板中禁用授权页面。Amazon Windows AMI 中的 Amazon Tools for PowerShell 默认为 32 位版本，并且此命令将失败。相反，将使用操作系统附带的 64 位版本的 PowerShell。

   ```
   New-Item -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name GridLicensing
   New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" -
   Name "NvCplDisableManageLicensePage" -PropertyType "DWord" -Value "1"
   ```

   命令提示符

   运行以下注册表命令以创建注册表值，从而在控制面板中禁用授权页面。您可以使用命令提示符窗口或 64 位版本的 PowerShell 运行它。

   ```
   reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" /v
    NvCplDisableManageLicensePage /t REG_DWORD /d 1
   ```

5. （可选）根据您的用例，您可能会完成以下可选步骤。如果您不需要此功能，请不要完成这些步骤。

   a. 请设置高性能显示协议 Amazon DCV，从而有助于利用四个高达 4K 分辨率的显示器的优势。

   b. NVIDIA Quadro 虚拟工作站模式默认处于启用状态。要为 RDSH 应用程序托管功能激活 GRID 虚拟应用程序，请完成在基于 Amazon EC2 GPU 的实例上激活 NVIDIA GRID 虚拟应用程序中的 GRID 虚拟应用程序激活步骤。

安装 NVIDIA 游戏驱动程序（G7e、G6、G6e、G5 和 G4dn 实例）

这些驱动程序仅供 Amazon 客户使用。下载即表明您同意仅将下载的软件用于开发在 RTX PRO 6000 Blackwell、NVIDIA L4、NVIDIA L40S、NVIDIA A10G、NVIDIA Tesla T4 或 NVIDIA Tesla M60 硬

件上使用的 AMI。可使用 GRID 驱动程序在 Amazon 环境中创建和使用 AMI。安装软件后，您将受 NVIDIA GRID 云最终用户许可协议的条款约束。

注意事项

- 要让 GRID 许可正常运行，G3 实例需要 Amazon 提供的 DNS 解析。
- IMDSv2 仅在 495.x 或更高版本的 NVIDIA 驱动程序上受支持。

先决条件

- (Linux) 验证 Amazon CLI 是否已安装在您的实例上并配置了默认凭证。有关更多信息，请参阅《Amazon Command Line Interface 用户指南》中的安装 Amazon CLI。
- 用户或角色必须具有包含 AmazonS3ReadOnlyAccess 策略的授予权限。

Amazon Linux 2023

在实例上安装 NVIDIA Gaming 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo dnf update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo dnf install gcc make
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启后重新连接到您的实例。
5. 安装内核标头软件包。

```
[ec2-user ~]$ sudo dnf install -y kernel-devel kernel-modules-extra kernel-devel-
$(uname -r) kernel-headers-$(uname -r) dkms
```

6. 使用以下命令下载 Gaming 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ . --region us-east-1 --no-sign-request
```

此存储桶中存储了多个版本的游戏驱动程序。您可以使用以下命令查看所有可用的版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/ --region us-east-1 --no-sign-request
```

7. 从下载的 .zip 归档中提取 Gaming 驱动程序安装实用程序。

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. 使用以下命令添加权限以运行驱动程序安装实用程序：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令运行安装程序：

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项 (您可以接受默认选项)。

10. 使用以下命令创建所需的配置文件。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下载并重命名认证文件。

- 对于 590.48 及更高版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 对于 460.39 到 580.95 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 对于 440.68 到 445.48 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 对于早期版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

12. 如果您在 G4dn、G5 或 G5g 实例上使用 510.x 或更高版本的 NVIDIA 驱动程序，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /
etc/modprobe.d/nvidia.conf
```

13. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

14. 请使用以下命令验证 NVIDIA Gaming 许可证。

```
[ec2-user ~]$ nvidia-smi.exe -q
```

在输出中，搜索 `vGPU Software Licensed Product`。

15. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。

Amazon Linux 2

在实例上安装 NVIDIA Gaming 驱动程序

1. 连接到您的实例。更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

2. 安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install gcc make
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启后重新连接到您的实例。
5. 为您当前运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo yum install -y kernel-devel-$(uname -r)
```

6. 使用以下命令下载 Gaming 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ . --region us-east-1 --no-sign-request
```

此存储桶中存储了多个版本的游戏驱动程序。您可以使用以下命令查看所有可用的版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/ --region us-east-1 --no-sign-request
```

7. 从下载的 `.zip` 归档中提取 Gaming 驱动程序安装实用程序。

```
[ec2-user ~]$ unzip latest-driver-name.zip -d nvidia-drivers
```

8. 使用以下命令添加权限以运行驱动程序安装实用程序：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令运行安装程序：

```
[ec2-user ~]$ sudo ./nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

如果您使用的是具有 5.10 内核版本的 Amazon Linux 2，请使用以下命令安装 NVIDIA 游戏驱动程序。

```
[ec2-user ~]$ sudo CC=/usr/bin/gcc10-cc ./NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项（您可以接受默认选项）。

10. 使用以下命令创建所需的配置文件。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下载并重命名认证文件。

    - 对于 590.48 及更高版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
      ```

    - 对于 460.39 到 580.95 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
      ```

    - 对于 440.68 到 445.48 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
      ```

    - 对于早期版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
      ```

12. 如果您在 G4dn、G5 或 G5g 实例上使用 510.x 或更高版本的 NVIDIA 驱动程序，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

    ```
    [ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
    ```

    ```
    [ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
    ```

13. 重启实例。

    ```
    [ec2-user ~]$ sudo reboot
    ```

14. 请使用以下命令验证 NVIDIA Gaming 许可证。

```
[ec2-user ~]$ nvidia-smi.exe -q
```

在输出中，搜索 vGPU Software Licensed Product。

15. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。

CentOS 7 和 Red Hat Enterprise Linux 7

在实例上安装 NVIDIA Gaming 驱动程序

1. 连接到 Linux 实例。安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

2. 更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 为您当前运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. 禁用 NVIDIA 显卡的 nouveau 开源驱动程序。

   a. 将 nouveau 添加到 /etc/modprobe.d/blacklist.conf 黑名单文件。复制下面的代码块并将其粘贴到终端中。

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
```

```
EOF
```

b. 编辑 /etc/default/grub 文件并添加以下行：

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

c. 重新生成 Grub 配置。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. 使用以下命令下载 Gaming 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ . --region us-
east-1 --no-sign-request
```

此存储桶中存储了多个版本的游戏驱动程序。您可以使用以下命令查看所有可用的版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/ --region us-east-1 --
no-sign-request
```

8. 从下载的 .zip 归档中提取 Gaming 驱动程序安装实用程序。

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

9. 使用以下命令添加权限以运行驱动程序安装实用程序：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

10. 使用以下命令运行安装程序：

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项 (您可以接受默认选项)。

11. 使用以下命令创建所需的配置文件。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

12. 使用以下命令下载并重命名认证文件。

- 对于 590.48 及更高版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 对于 460.39 到 580.95 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 对于 440.68 到 445.48 版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 对于早期版本：

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

13. 如果您在 G4dn、G5 或 G5g 实例上使用 510.x 或更高版本的 NVIDIA 驱动程序，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
```

14. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

15. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。如果您不需要此功能，请勿完成此步骤。

CentOS Stream 8 和 Red Hat Enterprise Linux 8

在实例上安装 NVIDIA Gaming 驱动程序

1. 连接到 Linux 实例。安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

2. 更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 为您当前运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo yum install -y unzip kernel-devel-$(uname -r)
```

6. 使用以下命令下载 Gaming 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ . --region us-
east-1 --no-sign-request
```

此存储桶中存储了多个版本的游戏驱动程序。您可以使用以下命令查看所有可用的版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/ --region us-east-1 --
no-sign-request
```

7. 从下载的 `.zip` 归档中提取 Gaming 驱动程序安装实用程序。

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. 使用以下命令添加权限以运行驱动程序安装实用程序：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令运行安装程序：

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项 (您可以接受默认选项)。

10. 使用以下命令创建所需的配置文件。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下载并重命名认证文件。

    - 对于 590.48 及更高版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
      ```

    - 对于 460.39 到 580.95 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
      ```

    - 对于 440.68 到 445.48 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
      ```

    - 对于早期版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
      ```

12. 如果您在 G4dn、G5 或 G5g 实例上使用 510.x 或更高版本的 NVIDIA 驱动程序，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

    ```
    [ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
    ```

    ```
    [ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
    ```

13. 重启实例。

    ```
    [ec2-user ~]$ sudo reboot
    ```

14. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。

Rocky Linux 8

在实例上安装 NVIDIA Gaming 驱动程序

1. 连接到 Linux 实例。安装 gcc 和 make（如果尚未安装）。

```
[ec2-user ~]$ sudo yum install -y gcc make
```

2. 更新软件包缓存并获取实例软件包更新。

```
[ec2-user ~]$ sudo yum update -y
```

3. 重启实例以加载最新内核版本。

```
[ec2-user ~]$ sudo reboot
```

4. 重启之后重新连接到实例。
5. 为您当前运行的内核版本安装内核标头软件包。

```
[ec2-user ~]$ sudo dnf install -y unzip elfutils-libelf-devel libglvnd-devel
 kernel-devel-$(uname -r)
```

6. 使用以下命令下载 Gaming 驱动程序安装实用程序：

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ . --region us-
east-1 --no-sign-request
```

此存储桶中存储了多个版本的游戏驱动程序。您可以使用以下命令查看所有可用的版本：

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/ --region us-east-1 --
no-sign-request
```

7. 从下载的 `.zip` 归档中提取 Gaming 驱动程序安装实用程序。

```
[ec2-user ~]$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

8. 使用以下命令添加权限以运行驱动程序安装实用程序：

```
[ec2-user ~]$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

9. 使用以下命令运行安装程序：

```
[ec2-user ~]$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项 (您可以接受默认选项)。

10. 使用以下命令创建所需的配置文件。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

11. 使用以下命令下载并重命名认证文件。

    - 对于 590.48 及更高版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
      ```

    - 对于 460.39 到 580.95 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
      ```

    - 对于 440.68 到 445.48 版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
      ```

    - 对于早期版本：

      ```
      [ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
      ```

12. 如果您在 G4dn、G5 或 G5g 实例上使用 510.x 或更高版本的 NVIDIA 驱动程序，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
[ec2-user ~]$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
[ec2-user ~]$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/modprobe.d/nvidia.conf
```

13. 重启实例。

```
[ec2-user ~]$ sudo reboot
```

14. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。

Ubuntu 和 Debian

在实例上安装 NVIDIA Gaming 驱动程序

1. 连接到 Linux 实例。安装 gcc 和 make（如果尚未安装）。

```
$ sudo apt-get install -y gcc make build-essential
```

2. 更新软件包缓存并获取实例软件包更新。

```
$ sudo apt-get update -y
```

3. 升级 `linux-aws` 程序包以接收最新版本。

```
$ sudo apt-get upgrade -y linux-aws
```

4. 重启实例以加载最新内核版本。

```
$ sudo reboot
```

5. 重启之后重新连接到实例。
6. 为您当前运行的内核版本安装内核标头软件包。

```
$ sudo apt install -y unzip dkms linux-headers-$(uname -r)
```

7. 禁用 NVIDIA 显卡的 nouveau 开源驱动程序。

   a. 将 nouveau 添加到 `/etc/modprobe.d/blacklist.conf` 黑名单文件。复制下面的代码块并将其粘贴到终端中。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
```

```
blacklist rivatv
EOF
```

b. 编辑 /etc/default/grub 文件并添加以下行：

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

c. 重新生成 Grub 配置。

```
$ sudo update-grub
```

8. 使用以下命令下载 Gaming 驱动程序安装实用程序：

```
$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ . --region us-east-1 --no-
sign-request
```

此存储桶中存储了多个版本的游戏驱动程序。您可以使用以下命令查看所有可用的版本：

```
$ aws s3 ls --recursive s3://nvidia-gaming/linux/ --region us-east-1 --no-sign-
request
```

9. 从下载的 .zip 归档中提取 Gaming 驱动程序安装实用程序。

```
$ unzip *Gaming-Linux-Guest-Drivers.zip -d nvidia-drivers
```

10. 使用以下命令添加权限以运行驱动程序安装实用程序：

```
$ chmod +x nvidia-drivers/NVIDIA-Linux-x86_64*-grid.run
```

11. 使用以下命令运行安装程序：

```
$ sudo nvidia-drivers/NVIDIA-Linux-x86_64*.run
```

系统提示时，接受许可协议并根据需要指定安装选项 (您可以接受默认选项)。

12. 使用以下命令创建所需的配置文件。

```
$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

13. 使用以下命令下载并重命名认证文件。

- 对于 590.48 及更高版本：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert_2026_03_02.cert"
```

- 对于 460.39 到 580.95 版本：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertLinux_2024_02_22.cert"
```

- 对于 440.68 到 445.48 版本：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- 对于早期版本：

```
$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

14. 如果您在 G4dn、G5 或 G5g 实例上使用 510.x 或更高版本的 NVIDIA 驱动程序，请使用以下命令禁用 GSP。有关为什么需要这样做的更多信息，请访问 NVIDIA 文档。

```
$ sudo touch /etc/modprobe.d/nvidia.conf
```

```
$ echo "options nvidia NVreg_EnableGpuFirmware=0" | sudo tee --append /etc/
modprobe.d/nvidia.conf
```

15. 重启实例。

```
$ sudo reboot
```

16. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。如果您不需要此功能，请勿完成此步骤。

Windows 操作系统

在实例上安装 NVIDIA 游戏驱动程序之前，除了所有游戏驱动程序提及的注意事项之外，您还必须确保满足以下先决条件。

- 如果您使用自定义 Windows AMI 启动 Windows 实例，则 AMI 必须是使用 Windows Sysprep 创建的标准化映像，以确保游戏驱动程序正常运行。有关更多信息，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。
- 在 Windows 实例上配置 Amazon Tools for Windows PowerShell 的默认凭证。有关更多信息，请参阅 Amazon Tools for Windows PowerShell 用户指南中的 Amazon Tools for PowerShell 入门

在 Windows 实例上安装 NVIDIA Gaming 驱动程序

1. 连接到您的 Windows 实例并打开 PowerShell 窗口。
2. 配置您的 Amazon 配置文件以按如下方式使用全球区域：

```
$ aws configure
AWS Access Key ID [None]: enter a Global region access key
AWS Secret Access Key [None]: enter a Global region secret access key
Default region name [None]: us-east-1
Default output format [None]: json
```

3. 使用以下 PowerShell 命令下载并安装 Gaming 驱动程序。

```
$Bucket = "nvidia-gaming"
$KeyPrefix = "windows/latest"
$LocalPath = "$home\Desktop\NVIDIA"
$Objects = Get-S3Object -BucketName $Bucket -KeyPrefix $KeyPrefix -Region us-east-1
foreach ($Object in $Objects) {
$LocalFileName = $Object.Key
if ($LocalFileName -ne '' -and $Object.Size -ne 0) {
$LocalFilePath = Join-Path $LocalPath $LocalFileName
Copy-S3Object -BucketName $Bucket -Key $Object.Key -LocalFile $LocalFilePath -
Region us-east-1
}
}
```

此 S3 存储桶中存储了多个版本的 NVIDIA GRID 驱动程序。如果将 `$KeyPrefix` 变量的值从“windows/latest”更改为“windows”，则可以下载桶中的所有可用版本。

4. 导航到桌面，双击安装文件以启动它 (选择与您的实例操作系统版本对应的驱动程序版本)。按照说明安装驱动程序并根据需要重启实例。要验证 GPU 是否正常工作，请检查设备管理器。
5. 使用以下方法之一注册驱动程序。

Version 527.27 or above

使用 64 位版本的 PowerShell 或命令提示符窗口创建以下注册表项。

键：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global

名称：vGamingMarketplace

类型：DWord

值：2

PowerShell

运行以下 PowerShell 命令创建此注册表值。Amazon Windows AMI 中的 Amazon Tools for PowerShell 默认为 32 位版本，并且此命令将失败。相反，将使用操作系统附带的 64 位版本的 PowerShell。

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global"
 -Name "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

命令提示符

运行以下注册表命令创建此注册表值。您可以使用命令提示符窗口或 64 位版本的 PowerShell 运行它。

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global" /v
 vGamingMarketplace /t REG_DWORD /d 2
```

Earlier versions

使用 64 位版本的 PowerShell 或命令提示符窗口创建以下注册表项。

键：HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global

名称：vGamingMarketplace

类型：DWord

值：2

PowerShell

运行以下 PowerShell 命令创建此注册表值。Amazon Windows AMI 中的 Amazon Tools for PowerShell 默认为 32 位版本，并且此命令将失败。相反，将使用操作系统附带的 64 位版本的 PowerShell。

```
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name
 "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

命令提示符

运行以下注册表命令以使用命令提示符窗口创建此注册表项。您也可以在 64 位版本的 PowerShell 中使用此命令。

```
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global" /v vGamingMarketplace /t
 REG_DWORD /d 2
```

6. 在 PowerShell 中运行以下命令。这将下载认证文件，重命名文件 `GridSwCert.txt`，然后将该文件移至系统驱动器上的 Public Documents（公共文档）文件夹。通常，该文件夹的路径为 `C:\Users\Public\Documents`。

   - 对于 591.59 及更高版本：

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCert_2026_03_02.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 对于 460.39 到 581.80 版本：

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCertWindows_2024_02_22.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 对于 445.87 版本：

     ```
     Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
     Archive/GridSwCert-Windows_2020_04.cert" -OutFile "$Env:PUBLIC\Documents
     \GridSwCert.txt"
     ```

   - 对于早期版本：

```
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-
Archive/GridSwCert-Windows_2019_09.cert" -OutFile "$Env:PUBLIC\Documents
\GridSwCert.txt"
```

如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

7. 重新启动您的实例。
8. 在实例上找到 `nvidia-smi.exe` 文件。

```
Get-ChildItem -Path C:\ -Recurse -Filter "nvidia-smi.exe"
```

请使用以下命令验证 NVIDIA Gaming 许可证。将 *path* 替换为上一个命令输出中的文件夹名称。

```
C:\Windows\System32\DriverStore\FileRepository\path\nvidia-smi.exe -q
```

该输出值应该类似于以下内容。

```
vGPU Software Licensed Product
Product Name              : NVIDIA Cloud Gaming
License Status            : Licensed (Expiry: N/A)
```

9. （可选）请设置高性能显示协议 Amazon DCV，从而有助于利用单个高达 4K 分辨率的显示器的优势。如果您不需要此功能，请勿完成此步骤。

在 EC2 Windows 实例上安装 ENA 驱动程序

如果实例并非基于 Amazon 提供的最新 Windows 亚马逊机器映像（AMI）中的一种，请使用以下步骤在您的实例上安装当前 ENA 驱动程序。您应该在方便重启实例的时候执行该更新。如果安装脚本没有自动重启实例，我们建议您最后再重启实例。

如果您在实例运行期间使用实例存储卷来存储数据，则当您停止该实例时，该数据将被擦除。在停止实例之前，请首先确认您是否已将所需数据从实例存储卷复制到持久性存储，例如 Amazon EBS 或 Amazon S3。

先决条件

要安装或升级 ENA 驱动程序，Windows 实例必须满足以下先决条件：

- 已安装 PowerShell 3.0 或更高版本。
- 本节中显示的命令必须在 64 位版本的 PowerShell 中运行。不要使用 x86 版本的 PowerShell。这是 32 位版本的 shell，不支持这些命令。

第 1 步：备份数据

我们建议您创建备份 AMI，以备在无法通过设备管理器回滚更改时使用。要使用 Amazon Web Services 管理控制台创建备份 AMI，请执行以下步骤：

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择需要升级驱动程序的实例，然后从实例状态菜单中选择停止实例。
4. 在实例停止后再次选择该实例。要创建备份，请从操作菜单中选择映像和模板，然后选择创建映像。
5. 要重启实例，请从实例状态菜单中选择启动实例。

第 2 步：安装或升级 ENA 驱动程序

您可以使用 Amazon Systems Manager Distributor 或 PowerShell cmdlet 安装或升级 ENA 驱动程序。有关进一步的说明，请选择您要使用的方法对应的选项卡。

Systems Manager Distributor

您可以使用 Systems Manager Distributor 功能将程序包部署到 Systems Manager 托管式节点。使用 Systems Manager Distributor 时，您可以一次性安装 ENA 驱动程序包，也可以通过计划更新来安装。有关如何使用 Systems Manager Distributor 安装 ENA 驱动程序包 (`AwsEnaNetworkDriver`) 的更多信息，请参阅《Amazon Systems Manager 用户指南》中的 安装或更新程序包。

PowerShell

本节介绍了如何在实例上使用 PowerShell cmdlet 下载和安装 ENA 驱动程序包。

选项 1：下载并解压缩最新版本

1. 连接到您的实例并以本地管理员身份登录。
2. 使用 invoke-webrequest cmdlet 下载最新驱动程序包：

```
PS C:\> invoke-webrequest https://ec2-windows-drivers-
downloads.s3.amazonaws.com/ENA/Latest/AwsEnaNetworkDriver.zip -
outfile $env:USERPROFILE\AwsEnaNetworkDriver.zip
```

> **Note**
>
> 如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```
>
> 您还可以在实例上通过浏览器窗口下载最新驱动程序包。

3. 使用 expand-archive cmdlet 将您下载的 zip 存档文件提取到您的实例上：

```
PS C:\> expand-archive $env:userprofile\AwsEnaNetworkDriver.zip -
DestinationPath $env:userprofile\AwsEnaNetworkDriver
```

选项 2：下载并解压缩特定版本

1. 连接到您的实例并以本地管理员身份登录。
2. 利用 ENA Windows 驱动程序版本历史记录 表中的版本链接下载所需特定版本的 ENA 驱动程序包。
3. 将 zip 存档文件提取到您的实例。

使用 PowerShell 安装 ENA 驱动程序

无论您下载的是最新版本还是特定版本的驱动程序，安装步骤都一样。要安装 ENA 驱动程序，请按照以下步骤操作。

1. 要安装驱动程序，请从实例上的 `AwsEnaNetworkDriver` 目录运行 `install.ps1` PowerShell 脚本。如果出现错误，请确保您使用的是 PowerShell 3.0 或更高版本。
2. 如果安装程序没有自动重启实例，请运行 Restart-Computer PowerShell cmdlet。

```
PS C:\> Restart-Computer
```

第 3 步（可选）：安装后验证 ENA 驱动程序版本

为确保已在实例上成功安装 ENA 驱动程序包，您可以按如下方式验证新版本：

1. 连接到您的实例并以本地管理员身份登录。
2. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 `devmgmt.msc`。
3. 选择确定。这将打开 Device Manager（设备管理器）窗口。
4. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。
5. 选择名称或打开 Amazon Elastic Network Adapter 的上下文菜单，然后选择 Properties（属性）。这将打开 Amazon 弹性网络适配器属性对话框。

> **Note**
>
> 所有 ENA 适配器都使用相同的驱动程序。如果您有多个 ENA 适配器，选择其中任何一个即可更新所有 ENA 适配器的驱动程序。

6. 要验证已安装的当前版本，请打开驱动程序选项卡，然后检查驱动程序版本。如果当前版本与您的目标版本不符，请参阅 对弹性网络适配器 Windows 驱动程序进行问题排查。

回滚 ENA 驱动程序安装

如果安装出现任何问题，则可能需要回滚驱动程序。按照以下步骤回滚在实例上安装的先前版本的 ENA 驱动程序。

1. 连接到您的实例并以本地管理员身份登录。
2. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 `devmgmt.msc`。
3. 选择确定。这将打开 Device Manager（设备管理器）窗口。
4. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。

5. 选择名称或打开 Amazon Elastic Network Adapter 的上下文菜单，然后选择 Properties（属性）。这将打开 Amazon 弹性网络适配器属性对话框。

> **Note**
>
> 所有 ENA 适配器都使用相同的驱动程序。如果您有多个 ENA 适配器，选择其中任何一个即可更新所有 ENA 适配器的驱动程序。

6. 要回滚驱动程序，请打开驱动程序选项卡，然后选择回滚驱动程序。这将打开驱动程序包回滚窗口。

> **Note**
>
> 如果驱动程序选项卡未显示回滚驱动程序操作，或者该操作不可用，则表示实例上的驱动程序存储不包含之前安装的驱动程序包。要解决此问题，请参阅 诊断场景 并展开安装了意外的 ENA 驱动程序版本部分。有关设备驱动程序包选择过程的更多信息，请参阅《Microsoft 文档网站》上的 Windows 如何为设备选择驱动程序包。

跟踪 ENA Windows 驱动程序发行版

Windows AMI 包含 ENA 驱动程序，用于启用增强联网功能。

对于 Windows Server 2016 及以上版本，我们建议您使用最新的驱动程序版本。对于早期版本的 Windows Server，请参阅下表来确定要使用哪个 ENA 驱动程序版本。

Windows Server 版本	ENA 驱动程序版本
Windows Server 2012 R2	2.6.0 及更早版本
Windows Server 2012	2.6.0 及更早版本
Windows Server 2008 R2	2.2.3 及更早版本

ENA Windows 驱动程序版本历史记录

下表总结了每个版本的变更。

驱动程序版本	详细信息	发行日期
2.11.0	新特征 • 增加了对动态队列分配的支持，最多可扩展至 64 个 I/O 队列，具体视实例类型的限制而定。 错误修复 • 修复了从设备检索 DMA 物理地址宽度时未处理错误情况的问题。	2025 年 8 月 1 日
2.10.0	错误修复 • 修复了队列初始化期间内存分配失败处理不当的问题，从而防止意外重启。 • 修复了可能导致忽略设备功能的网络卸载配置错误。	2025 年 6 月 24 日
2.9.0	新特征 • 增加了对由设备发起的异步重置请求的支持。 • 增加了对处理由设备提供的大型 LLQ 最大深度值的支持。 • 在 Windows 事件查看器中添加了事件 ID 58001，以更好地了解因设备配置错误导致的意外电源状态转换。 错误修复 •	2024 年 12 月 12 日

驱动程序版本	详细信息	发行日期
	修复了设备初始化期间内存分配失败处理不当的问题，从而防止意外重启。 • 修复了中断服务例程中可能会在设备停止期间将 DPC 排入队列的问题，从而防止意外重启。	
2.8.0	错误修复 • 修复了完整出口网络缓冲区列表（NBL）处理流程中的争用条件，该条件可能尝试释放已经释放的 NBL，从而导致内存损坏。 • 修复了禁用所有 LSO 时错误检测 L3 协议的问题，以及可能会导致意外行为的校验和卸载问题。	2024 年 9 月 30 日

驱动程序版本	详细信息	发行日期
2.7.0	新特征 • 已删除对 Windows Server 2012（Windows 8）和 Windows Server 2012 R2（Windows 8.1）的支持。这些操作系统版本已终止对 Amazon 的支持。在 Windows Server 2012 及更早版本上安装驱动程序将失败。 • 添加了对将 IPv6 Tx 校验和计算分载到设备的支持。 • 添加了广泛的低延迟队列（LLQ）支持。这将根据设备推荐动态启用。您可以使用新的“WideLLQ”注册表项覆盖此设置。 • 添加了因 Rx 溢出导致的数据包丢失报告，这表明 Rx 环中没有足够的空间容纳传入的数据包。 • 增加了对来自设备的次优配置通知的支持。在 Windows 事件查看器中查看事件 ID 59000。 错误修复 • 避免由于 Tx 数据包的标头超过最大低延迟队列（LLQ）标头大小而导致的不必要设备重置。	2024 年 5 月 1 日

驱动程序版本	详细信息	发行日期

2.6.0	新特征
	• 为支持 ENA Express 的实例类型添加了以下网络性能指标。
	  • ena_srd_mode
	  • ena_srd_tx_pkts
	  • ena_srd_eligible_tx_pkts
	  • ena_srd_rx_pkts
	  • ena_srd_resource_utilization
	• 为基于 Nitro 的实例类型添加了 conntrack_allowance_available 网络性能指标。
	• 添加了由于检测到 RX 数据损坏而导致的新适配器重置原因。
	• 更新了驱动程序日志记录基础设施。
	
	错误修复
	• 防止在 CPU 不足导致网络性能指标更新失败时重置适配器。
	• 防止错误检测到设备检测信号中断。
	• 修复了驱动程序安装脚本，可支持降级操作。
	• 修复了接收错误计数统计数据。
		2023 年 6 月 20 日

驱动程序版本	详细信息	发行日期
2.5.0	公告 由于无法在 Windows 域控制器上初始化，ENA Windows 驱动程序版本 2.5.0 已回滚。Windows 客户端和 Windows Server 不受影响。	2023 年 2 月 17 日
2.4.0	新特征 • 添加了对 Windows Server 2022 的支持。 • 删除了对 Windows Server 2008 R2 的支持。 • 将低延迟队列 (LLQ) 设置为始终开启，以提高第六代 Amazon EC2 实例的性能。 错误修复 • 修复了无法将网络性能指标发布到适用于 Windows 的性能计数器 (PCW) 系统的故障。 • 修复了注册表项读取操作期间的内存泄漏问题。 • 在适配器重置过程中，如果出现不可恢复的错误，可防止无限重置循环。	2022 年 4 月 28 日

驱动程序版本	详细信息	发行日期
2.2.4	公告 由于第六代 EC2 实例的潜在性能下降，ENA Windows 驱动程序版本 2.2.4 已回滚。建议您使用以下方法之一降级驱动程序： • 安装旧版本 1. 从此表中的链接下载旧版本软件包（版本 2.2.3）。 2. 运行 install.ps1 PowerShell 安装脚本。 有关安装前和安装后步骤的详细信息，请参阅 在 Windows 上启用增强联网。 • 使用 Amazon EC2 Systems Manager 进行批量更新 • 通过 SSM 文档 `AWS-ConfigureAWSPackage` 和如下参数执行批量更新： • Name（名称）：AwsEnaNetworkDriver • Version（版本）：2.2.3	2021 年 10 月 26 日

驱动程序版本	详细信息	发行日期
2.2.3	新特征 • 增加了对新 Nitro 卡的支持，实例联网速度高达 400 Gbps。 错误修复 • 修复了 ENA 驱动程序的系统时间更改和系统时间查询之间的竞争条件，这会导致硬件无响应的误报检测。 公告 Windows ENA 驱动程序版本 2.2.3 是支持 Windows Server 2008 R2 的最终版本。Windows Server 2008 R2 将继续支持当前可用的使用 ENA 的实例类型，并且驱动程序可以通过下载获得。未来的实例类型不会支持 Windows Server 2008 R2，且您无法启动、导入或将 Windows Server 2008 R2 映像迁移到未来的实例类型。	2021 年 3 月 25 日
2.2.2	新特征 • 添加了对使用 CloudWatch 和适用于 Windows 使用者的性能计数器查询网络适配器性能指标的支持。 错误修复 • 修复了裸机实例的性能问题。	2020 年 12 月 21 日

驱动程序版本	详细信息	发行日期
2.2.1	新特征 • 添加一种方法，以允许主机查询 Elastic Network Adapter 来获取网络性能指标。	2020 年 10 月 1 日
2.2.0	新特征 • 添加了对下一代硬件类型的支持。 • 提高了从停止休眠恢复后的实例启动时间，并消除误报 ENA 错误消息。 性能优化 • 优化入站流量的处理。 • 改进了低资源环境中的共享内存管理。 错误修复 • 在驱动程序无法重置的极少数情况下，避免在移除 ENA 设备时系统崩溃。	2020 年 8 月 12 日
2.1.5	错误修复 • 修复了裸机实例上偶尔出现的网络适配器初始化失败的问题。	2020 年 6 月 23 日

驱动程序版本	详细信息	发行日期
2.1.4	错误修复 • 防止来自网络堆栈的损坏 LSO 数据包元数据导致的连接问题。 • 防止由罕见的竞争情况导致的系统崩溃，这种情况导致访问已释放的数据包内存。	2019 年 11 月 25 日
2.1.2	新特征 • 添加了对供应商 ID 报告的支持，以允许操作系统生成基于 MAC 的 UUID。 错误修复 • 改进了初始化期间的 DHCP 网络配置性能。 • 当最大传输单位 (MTU) 超过 4K 时，正确计算入站 IPv6 流量的 L4 校验和。 • 驱动程序稳定性的一般改进和较小错误修复。	2019 年 11 月 4 日
2.1.1	错误修复 • 防止丢弃从操作系统到达的高度分段的 TCP LSO 数据包。 • 在 IPv6 网络中正确处理在 IPSec 内封装安全负载 (ESP) 协议。	2019 年 9 月 16 日

驱动程序版本	详细信息	发行日期

2.1.0	ENA Windows 驱动程序 2.1 版引入了新的 ENA 设备功能，提升了性能，添加了新的功能并包含多项稳定性改进。
	
	新功能
	
	• 在巨型帧配置中使用标准化的 Windows 注册表项。
	• 允许通过 ENA 驱动程序属性 GUI 配置 VLAN ID 设置。
	• 改进的恢复流程
	  • 改进的故障识别机制。
	  • 添加了对可调恢复参数的支持。
	• 对于具有超过 8 个 vCPU 的较新 EC2 实例，最多支持 32 个 I/O 队列。
	• 将驱动程序内存占用量减少约 90%。
	
	性能优化
	
	• 减少了传输路径延迟。
	• 支持接收校验和分载。
	• 优化负载较高的系统的性能（优化了锁定机制使用）。
	• 进一步增强以降低 CPU 利用率，并提高了具有较高负载的系统的响应速度。
		2019 年 7 月 1 日

驱动程序版本	详细信息	发行日期
	错误修复 • 修复由于不连续 Tx 标头解析无效而导致崩溃的问题。 • 修复在裸机实例上分离弹性网络接口期间驱动程序 1.5 版崩溃的问题。 • 修复 IPv6 上的 LSO 伪标头校验和计算错误。 • 修复在初始化失败时的潜在内存资源泄漏问题。 • 禁用 IPv4 分段的 TCP/UDP 校验和分载。 • 修复 VLAN 配置。在只应禁用 VLAN 优先级时，错误地禁用了 VLAN。 • 使事件查看器能够正确解析自定义驱动程序消息。 • 修复由于时间戳处理无效而无法初始化驱动程序的问题。 • 修复数据处理和 ENA 设备禁用之间的争用情况。	
1.5.0	更新 • 改进了稳定性，修复了性能问题。 • 接收缓冲区现在可在 ENA NIC 的“Advanced Properties (高级属性)”中配置为高达 8192 的值。 • 默认接收缓冲区为 1k。	2018 年 10 月 4 日

驱动程序版本	详细信息	发行日期
1.2.3	包括可靠性修复，并且统一了对 Windows Server 2008 R2 到 Windows Server 2016 的支持。	2018 年 2 月 13 日
1.0.8	首次发布。包含在适用于 Windows Server 2008 R2、Windows Server 2012 RTM、Windows Server 2012 R2 和 Windows Server 2016 的 AMI 中。	2016 年 7 月

订阅来自 Amazon SNS 的 ENA Windows 驱动程序发布通知

Amazon SNS 可在 EC2 Windows 驱动程序的新版本发布时向您发送通知。使用以下过程订阅这些通知。

订阅 EC2 通知

1. 通过以下网址打开 Amazon SNS 控制台：https://console.aws.amazon.com/sns/v3/home。
2. 如果需要，可在导航栏中将区域更改为美国东部（弗吉尼亚北部）。您必须选择此区域，因为您订阅的 SNS 通知是在此区域中创建的。
3. 在导航窗格中，选择 Subscriptions。
4. 选择 Create subscription。
5. 在 Create subscription 对话框中，执行以下操作：
   a. 对于 TopicARN，复制以下 Amazon Resource Name (ARN)：

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers

   b. 对于 Protocol，选择 `Email`。
   c. 对于端点，输入您希望向其发送通知的电子邮件地址。
   d. 选择 Create subscription。
6. 您将收到一封确认电子邮件。打开电子邮件，然后按照说明操作以完成订阅。

每当发布新的 EC2 Windows 驱动程序时，我们都会向订户发送通知。如果您不希望再收到这些通知，请通过以下步骤取消订阅。

从 Amazon EC2 Windows 驱动程序通知中取消订阅

1. 通过以下网址打开 Amazon SNS 控制台：https://console.aws.amazon.com/sns/v3/home。
2. 在导航窗格中，选择 Subscriptions。
3. 选中订阅的复选框，然后选择 Actions、Delete subscriptions。当系统提示进行确认时，选择 Delete。

Windows 实例的半虚拟化驱动程序

Windows AMI 包含一系列驱动程序，以允许访问虚拟化硬件。Amazon EC2 会使用这些驱动程序将实例存储和 Amazon EBS 卷映射到其设备。下表显示了不同驱动程序之间的主要区别。

	Red Hat 半虚拟化	Citrix PV 驱动程序	Amazon PV
实例类型	并非对所有实例类型都支持。如果您指定了不受支持的实例类型，实例将受损。	支持 Xen 实例类型。	支持 Xen 实例类型。
附加的卷	支持最多 16 个附加的卷。	支持超过 16 个附加的卷。	支持超过 16 个附加的卷。
Network	此驱动程序在高负载下 (例如，快速 FTP 文件传输) 重置网络连接时存在已知问题。		若采用兼容的实例类型，此驱动程序会在网络适配器上自动配置巨型帧。在实例位于集群置放群组时，这会在集群置放群组中的实例之间提供更好的网络性能。

	Red Hat 半虚拟化	Citrix PV 驱动程序	Amazon PV
			有关更多信息，请参阅 Amazon EC2 实例的置放群组。

下表显示了应在 Amazon EC2 上每个版本的 Windows Server 上运行的 PV 驱动程序。

Windows Server 版本	PV 驱动程序版本
Windows Server 2025	不支持
Windows Server 2022	Amazon PV 最新版本
Windows Server 2019	Amazon PV 最新版本
Windows Server 2016	Amazon PV 最新版本
Windows Server 2012 R2	Amazon PV 版本 8.4.3
Windows Server 2012	Amazon PV 版本 8.4.3
Windows Server 2008 R2	Amazon PV 版本 8.3.5
Windows Server 2008	Citrix PV 驱动程序 5.9
Windows Server 2003	Citrix PV 驱动程序 5.9

目录

- Amazon PV 驱动程序
- Citrix PV 驱动程序
- Red Hat PV 驱动程序
- 在 EC2 Windows 实例上升级半虚拟化驱动程序
- 排查 Windows 实例上的 PV 驱动程序问题

Amazon PV 驱动程序

Amazon PV 驱动程序存储在 %ProgramFiles%\Amazon\Xentools 目录中。此目录还包含公有符号和一个命令行工具 xenstore_client.exe，使用该工具可以访问 XenStore 中的条目。例如，以下 PowerShell 命令从虚拟机监控程序返回当前时间：

```
PS C:\> [DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl
 AWSXenStoreBase).XenTime).ToString("hh:mm:ss")
11:17:00
```

Amazon PV 驱动程序在 Windows 注册表中列于 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services 之下。这些驱动程序组件如下所示：xenbus、xeniface、xennet、xenvbd 和 xenvif。

Amazon PV 驱动程序还有一个名为 LiteAgent 的 Windows 服务，它在用户模式下运行。它会处理相关任务，例如在 Xen 代实例上通过 Amazon API 进行的关闭和重启事件。您可以通过从命令行运行 Services.msc 来访问和管理服务。在 Nitro 代实例上运行时，Amazon PV 驱动程序将不会使用，并且 LiteAgent 服务将自动停止启动使用驱动程序版本 8.2.4。更新至最新的 Amazon PV 驱动程序也将更新 LiteAgent 并提高所有实例世代上的可靠性。

安装最新的 Amazon PV 驱动程序

Amazon Windows AMI 包含一系列驱动程序，以允许访问虚拟化硬件。Amazon EC2 会使用这些驱动程序将实例存储和 Amazon EBS 卷映射到其设备。我们建议您安装最新的驱动程序来提高您的 EC2 Windows 实例的稳定性和性能。

安装选项

- 使用 Amazon Systems Manager 自动更新 PV 驱动程序。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的演练：在 EC2 Windows 实例上自动更新 PV 驱动程序。
- 下载驱动程序包并手动运行安装程序。请务必检查 readme.txt 文件以了解系统要求。有关下载和安装 Amazon PV 驱动程序，或者升级域控制器的信息，请参阅 手动升级 Windows Server 实例 (Amazon PV 升级)。

Amazon PV 驱动程序包历史记录

下表显示了每个驱动程序版本中的 Amazon PV 驱动程序更改。

程序包版本	详细信息	发行日期
8.6.0	• 有关 XenStore 交互的稳定性修复。	2025 年 5 月 27 日
8.5.0	• 稳定性修复解决了网络设备分离期间罕见的崩溃情况。 • 稳定性修复可解决 EBS 卷分离期间崩溃的少数问题。 • 修复了包安装程序中的错误。 • 更新了 PV 安装程序以使用 `Pnputil`。	2024 年 10 月 31 日
8.4.3	为改善升级体验，修复了软件包安装程序中的错误。这是最后一个可以在 Windows Server 2012 和 2012 R2 上运行的版本。此版本可供下载，但由于 Windows Server 2012 和 2012 R2 已终止支持，因此不再受支持。	2023 年 1 月 24 日
8.4.2	稳定性修复以解决争用情况。	2022 年 4 月 13 日
8.4.1	改进了软件包安装程序。	2022 年 1 月 7 日
8.4.0	• 稳定性修复可解决磁盘 IO 卡住的少数问题。 • 稳定性修复可解决 EBS 卷分离期间崩溃的少数问题。 • 增加了跨多个核心分配负载的功能，用于利用 20,000 多个 IOPS 以及由于瓶颈而发生降级的工作负载。要启用此功能，请参阅 利用 20,000 多个磁盘 IOPS 由于 CPU 瓶颈而发生降级的工作负载。	2021 年 3 月 2 日
8.3.5	改进了软件包安装程序。 这是最后一个可以在 Windows Server 2008 R2 上运行的版本。此版本可供下载，但不再受支持。Windows Server 2008 R2 的生命周期已经终止，Microsoft 不再提供支持。	2022 年 1 月 7 日
8.3.4	提高了网络设备连接的可靠性。	2020 年 8 月 4 日

程序包版本	详细信息	发行日期
8.3.3	• 更新面向 XenStore 的组件，以防止在错误处理路径期间进行错误检查。 • 更新存储组件，以避免在提交无效 SRB 时崩溃。 要在 Windows Server 2008 R2 实例上更新此驱动程序，您必须首先验证安装了相应的修补程序，以解决以下 Microsoft 安全公告：Microsoft Security Advisory 3033929。	2020 年 2 月 4 日
8.3.2	增强了网络组件的可靠性。	2019 年 7 月 30 日
8.3.1	提高了存储组件的性能和可靠性。	2019 年 6 月 12 日
8.2.7	提高效率以支持迁移到最新一代实例类型。	2019 年 5 月 20 日
8.2.6	提高了故障转储路径的效率。	2019 年 1 月 15 日
8.2.5	更多安全增强。 PowerShell 安装程序现在在程序包中提供。	2018 年 12 月 12 日
8.2.4	可靠性改进。	2018 年 10 月 2 日
8.2.3	缺陷修复和性能改进。 将 EBS 卷 ID 报告为 EBS 卷的磁盘序列号。这可实现集群方案，如 S2D。	2018 年 5 月 29 日
8.2.1	网络和存储性能改进以及多项可靠性修复。 要验证是否已安装此版本，请参阅以下 Windows 注册表值：`HKLM\Software\Amazon\PVDriver\Version 8.2.1`。	2018 年 3 月 8 日

程序包版本	详细信息	发行日期
7.4.3	增加了适用于 Windows Server 2016 的支持。 对所有受支持的 Windows 操作系统版本的稳定性修复。 *Amazon PV 驱动程序版本 7.4.3 的签名将于 2019 年 3 月 29 日过期。我们建议更新到最新的 Amazon PV 驱动程序。	2016 年 11 月 18 日
7.4.2	对 X1 实例类型支持的稳定性修复。	2016 年 8 月 2 日
7.4.1	• Amazon PV 存储驱动程序的性能改进。 • 对 Amazon PV 存储驱动程序进行了稳定性修复：修复了实例在遇到错误代码 0x0000DEAD 时出现系统崩溃的问题。 • Amazon PV 网络驱动程序的稳定性修复。 • 增加了适用于 Windows Server 2008R2 的支持。	2016 年 7 月 12 日
7.3.2	• 改进了日志记录和诊断。 • Amazon PV 存储驱动程序的稳定性修复。在某些情况下，将磁盘重新附加到实例后，磁盘可能不显示在 Windows 中。 • 增加了适用于 Windows Server 2012 的支持。	2015 年 6 月 24 日
7.3.1	TRIM 更新：与 TRIM 请求相关的修复。该修复实现了实例的稳定性并提高了管理大量 TRIM 请求时的实例性能。	
7.3.0	TRIM 支持：Amazon PV 驱动程序现在将 TRIM 请求发送到管理程序。假如基础存储支持 TRIM (SSD)，临时磁盘将妥善处理 TRIM 请求。请注意，自 2015 年 3 月起，基于 EBS 的存储不再支持 TRIM。	

程序包版本	详细信息	发行日期
7.2.5	• Amazon PV 存储驱动程序中的稳定性修复：在某些情况下，Amazon PV 驱动程序可能解除引用无效的内存并导致系统故障。 • 生成故障转储时的稳定性修复：在某些情况下，在写入故障转储时，Amazon PV 驱动程序可能受阻于竞争条件。在此版本之前，该问题只能通过强制驱动程序停止并重启来解决，这样做会丢失内存转储。	
7.2.4	设备 ID 持久性：此驱动程序修复将掩盖平台 PCI 设备 ID 并强制系统始终显示同一设备 ID，即使实例发生移动也是如此。更普遍的情况是，修复将影响管理程序显示虚拟设备的方式。修复还包括对 Amazon PV 驱动程序的辅助安装程序的修改，因此系统保留了映射的虚拟设备。	
7.2.2	• 在目录服务还原模式 (DSRM) 中加载 Amazon PV 驱动程序：目录服务还原模式是 Windows Server 域控制器的安全模式启动选项。 • 在重新附加虚拟网络适配器设备时保留设备 ID：此修复将强制系统检查 MAC 地址映射并保留设备 ID。此修复将确保重新附加适配器时适配器保留其静态设置。	
7.2.1	• 在安全模式下运行：修复了驱动程序在安全模式下无法加载的问题。以前，Amazon PV 驱动程序只会在正常运行的系统中进行实例化。 • 将磁盘添加到 Microsoft Windows 存储池：以前，我们汇总了第 83 页的查询。此修复禁用了第 83 页的支持。请注意，这不会影响在集群环境中使用的存储池，因为半虚拟化磁盘不是有效的集群磁盘。	
7.2.0	基础：Amazon PV 基础版本。	

Citrix PV 驱动程序

Citrix 半虚拟驱动程序存储在 `%ProgramFiles%\Citrix\XenTools` (32 位实例) 和 `%ProgramFiles(x86)%\Citrix\XenTools` (64 位实例) 目录中。

Citrix PV 驱动程序在 Windows 注册表中列于 HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\services 之下。这些驱动程序组件如下：xenevtchn、xeniface、xennet、Xennet6、xensvc、xenvbd 和 xenvif。

Citrix 还有一个名为 XenGuestAgent 的驱动程序组件，它以 Windows 服务的形式运行。它会处理相关任务，例如通过 API 进行的关闭和重启事件。您可以通过从命令行运行 Services.msc 来访问和管理服务。

如果您在执行某些工作负载时遇到联网错误，可能需要禁用 Citrix PV 驱动程序的 TCP 分载功能。有关更多信息，请参阅 TCP 分载。

Red Hat PV 驱动程序

支持对旧实例使用 Red Hat 驱动程序，但不建议对具有 12 GB 以上的 RAM 的较新实例使用该驱动程序，因为存在驱动程序限制。运行 Red Hat 驱动程序并具有 12 GB 以上的 RAM 的实例可能无法启动并变得无法访问。我们建议将 Red Hat 驱动程序升级为 Citrix PV 驱动程序，然后将 Citrix PV 驱动程序升级为 Amazon PV 驱动程序。

Red Hat 驱动程序的源文件在 %ProgramFiles%\RedHat（32 位实例）或 %ProgramFiles(x86)%\RedHat（64 位实例）目录中。这两种驱动程序分别为 Red Hat 半虚拟化网络驱动程序 rhelnet 和 Red Hat SCSI 微端口驱动程序 rhelscsi。

在 EC2 Windows 实例上升级半虚拟化驱动程序

我们建议您安装最新的半虚拟化驱动程序来提高您的 EC2 Windows 实例的稳定性和性能。本页上的说明可以帮助您下载驱动程序包并运行安装程序。

验证您的 Windows 实例使用哪个驱动程序

打开设备管理器，查看网络适配器。验证 PV 驱动程序是否是以下一种：

- Amazon PV 网络设备
- Citrix PV 以太网适配器
- Red Hat PV NIC 驱动程序

系统要求

请务必在下载中检查该 readme.txt 文件以了解系统要求。

内容

- 使用分发服务器升级 Windows Server 实例（Amazon PV 升级）
- 手动升级 Windows Server 实例(Amazon PV 升级)
- 升级域控制器 (Amazon PV 升级)
- 升级 Windows Server 2008 和 2008 R2 实例（Red Hat 到 Citrix PV 的升级）
- 升级 Citrix Xen 代理客户服务

使用分发服务器升级 Windows Server 实例（Amazon PV 升级）

您可以使用分发服务器（Amazon Systems Manager 的一项功能）安装或升级 Amazon PV 驱动程序包。安装或升级可以执行一次，也可以按计划进行安装或升级。此分发服务器软件包不支持安装类型的 `In-place update` 选项。

> **Important**
>
> 如果您的实例是域控制器，请参阅升级域控制器 (Amazon PV 升级)。这些域控制器实例的升级过程与标准版本的 Windows 的升级过程不同。

1. 如果需要回滚更改，我们建议您创建备份。

   > **Tip**
   >
   > 您可以使用 `AWS-CreateImage` 运行手册，通过 Systems Manager Automation 创建 AMI，而不是从 Amazon EC2 控制台创建 AMI。有关更多信息，请参阅 Amazon Systems Manager Automation 运行手册参考用户指南中的 AWS-CreateImage。

   a. 当您停止某个实例时，任何实例存储卷上的数据都将被擦除。在停止实例之前，请确认您是否已将所需数据从实例存储卷复制到持久性存储，例如 Amazon EBS 或 Amazon S3。
   b. 在导航窗格中，选择实例。
   c. 选择需要升级驱动程序的实例，然后依次选择 Instance state (实例状态)、Stop instance (停止实例)。
   d. 实例停止后，选择实例，依次选择 Actions (操作)、Image and templates (映像和模板)，然后选择 Create image (创建映像)。
   e. 依次选择实例状态、启动实例。

2. 使用远程桌面来连接到实例。有关更多信息，请参阅 the section called “使用 RDP 客户端进行连接”。
3. 在执行此升级之前，建议您将所有非系统磁盘离线，然后记录磁盘管理中映射到辅助磁盘的所有盘符。如果您要对 Amazon PV 驱动程序执行就地升级，则不需要此步骤。我们还建议在 Services 控制台中将不必要的服务设置为 Manual 启动。
4. 有关如何使用分发服务器安装或升级 Amazon PV 驱动程序包的说明，请参阅 Amazon Systems Manager 用户指南中的安装或更新软件包。
5. 在名称中，选择 AWSPVDriver。
6. 在安装类型中，选择卸载并重新安装。
7. 根据需要为软件包配置其他参数，然后使用 Step 4 中提及的过程运行安装或升级。

   在运行分发服务器软件包后，实例将自动重启，然后升级驱动程序。实例将有长达 15 分钟的时间不可用。
8. 升级完成且实例通过 Amazon EC2 控制台的两项运行状况检查后，使用远程桌面连接实例，验证新驱动程序是否已安装。
9. 连接后，请运行以下 PowerShell 命令：

   ```
   Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
   ```
10. 确认驱动程序版本与驱动程序版本历史记录表中列出的最新版本相同。有关更多信息，请参阅 Amazon PV 驱动程序包历史记录。打开磁盘管理以审核所有脱机辅助卷，并按照 Step 3 中记录的对应盘符将其联机。

如果您之前使用 Netsh 为 Citrix PV 驱动程序禁用了 TCP 分载，建议您在升级到 Amazon PV 驱动程序后重新启用此功能。Citrix 驱动程序的 TCP 分载问题在 Amazon PV 驱动程序中不会出现。因此，通过使用 Amazon PV 驱动程序，TCP 分载可提供更高的性能。

如果您之前已将一个静态 IP 地址或 DNS 配置应用于网络接口，则可能需要在升级 Amazon PV 驱动程序后重新应用该静态 IP 地址或 DNS 配置。

手动升级 Windows Server 实例(Amazon PV 升级)

使用以下过程对 Amazon PV 驱动程序执行就地升级，或在 Windows Server 2008 R2、Windows Server 2012、Windows Server 2012 R2、Windows Server 2016、Windows Server 2019 或 Windows Server 2022 上从 Citrix PV 驱动程序升级到 Amazon PV 驱动程序。此升级不适用于 Red Hat 驱动程序或 Windows Server 的其他版本。

某些较旧版本的 Windows Server 无法使用最新的驱动程序。要知道您的操作系统应该使用哪个驱动程序版本，请参阅 Windows 实例的半虚拟化驱动程序 页面。

> **Important**
>
> 如果您的实例是域控制器，请参阅升级域控制器 (Amazon PV 升级)。这些域控制器实例的升级过程与标准版本的 Windows 的升级过程不同。

手动升级 Amazon PV 驱动程序

1. 如果需要回滚更改，我们建议您创建备份。

   > **Tip**
   >
   > 您可以使用 `AWS-CreateImage` 运行手册，通过 Systems Manager Automation 创建 AMI，而不是从 Amazon EC2 控制台创建 AMI。有关更多信息，请参阅 Amazon Systems Manager Automation 运行手册参考用户指南中的 AWS-CreateImage。

   a. 当您停止某个实例时，任何实例存储卷上的数据都将被擦除。在停止实例之前，请确认您是否已将所需数据从实例存储卷复制到持久性存储，例如 Amazon EBS 或 Amazon S3。

   b. 在导航窗格中，选择实例。

   c. 选择需要升级驱动程序的实例，然后依次选择 Instance state (实例状态)、Stop instance (停止实例)。

   d. 实例停止后，选择实例，依次选择 Actions (操作) 、Image and templates (映像和模板)，然后选择 Create image (创建映像)。

   e. 依次选择实例状态、启动实例。

2. 使用远程桌面来连接到实例。

3. 在执行此升级之前，建议您将所有非系统磁盘离线，然后记录磁盘管理中映射到辅助磁盘的所有盘符。如果您要对 Amazon PV 驱动程序执行就地升级，则不需要此步骤。我们还建议在 Services 控制台中将不必要的服务设置为 Manual 启动。

4. 使用以下选项之一将驱动程序下载到您的实例：

   - 浏览器 - 将最新的驱动程序包下载到实例并解压缩 zip 存档。
   - PowerShell – 运行以下命令：

```
Invoke-WebRequest https://s3.cn-north-1.amazonaws.com.cn/ec2-windows-
drivers-downloads-cn/AWSPV/Latest/AWSPVDriver.zip -outfile $env:USERPROFILE
\pv_driver.zip
Expand-Archive $env:userprofile\pv_driver.zip -DestinationPath $env:userprofile
\pv_drivers
```

如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

5. 运行 `AWSPVDriverSetup.msi`。

在运行 MSI 后，实例将自动重启，然后升级驱动程序。实例将有长达 15 分钟的时间不可用。在升级完成并且实例通过了 Amazon EC2 控制台中的两项运行状况检查后，您可以通过使用远程桌面连接到实例，然后运行以下 PowerShell 命令，来验证新驱动程序是否已安装：

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

确认驱动程序版本与驱动程序版本历史记录表中列出的最新版本相同。有关更多信息，请参阅 Amazon PV 驱动程序包历史记录。打开磁盘管理以审核所有脱机辅助卷，并按照 Step 3 中记录的对应盘符将其联机。

如果您之前使用 Netsh 为 Citrix PV 驱动程序禁用了 TCP 分载，建议您在升级到 Amazon PV 驱动程序后重新启用此功能。Citrix 驱动程序的 TCP 分载问题在 Amazon PV 驱动程序中不会出现。因此，通过使用 Amazon PV 驱动程序，TCP 分载可提供更高的性能。

如果您之前已将一个静态 IP 地址或 DNS 配置应用于网络接口，则可能需要在升级 Amazon PV 驱动程序后重新应用该静态 IP 地址或 DNS 配置。

升级域控制器 (Amazon PV 升级)

使用以下过程在域控制器上对 Amazon PV 驱动程序执行就地升级，或者从 Citrix PV 驱动程序升级到 Amazon PV 驱动程序。为确保您的 FSMO 角色在升级期间保持可操作性，我们建议您在开始升级之前将这些角色转移到其他域控制器。有关更多信息，请参阅 Microsoft Learn 网站上的 How to view and transfer FSMO roles。

升级域控制器

1. 如果需要回滚更改，我们建议您创建域控制器的备份。不支持使用 AMI 作为备份。有关更多信息，请参阅 Microsoft 文档中的备份和还原注意事项。
2. 运行以下命令，将 Windows 配置为启动进入目录服务还原模式 (DSRM)。

   > **Warning**
   >
   > 在运行该命令之前，请确认您知道 DSRM 密码。您需要此信息，这样您可以在升级完成后登录您的实例，并且实例会自动重启。

   ```
   bcdedit /set {default} safeboot dsrepair
   ```

   PowerShell：

   ```
   PS C:\> bcdedit /set "{default}" safeboot dsrepair
   ```

   由于升级实用工具删除 Citrix PV 存储驱动程序以便安装 Amazon PV 驱动程序，因此系统必须启动进入 DSRM 模式。因此，建议您记录磁盘管理中的所有盘符和文件夹对辅助磁盘的映射。没有 Citrix PV 存储驱动程序时，将不会检测到第二个驱动器。使用第二个驱动器上的 NTDS 文件夹的域控制器将不会启动，因为不会检测到第二个磁盘。

   > **Warning**
   >
   > 运行该命令后，请勿手动重启系统。系统将无法访问，因为 Citrix PV 驱动程序不支持 DSRM。

3. 运行以下命令添加 **`DisableDCCheck`** 到注册表：

   ```
   reg add HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck /t
    REG_SZ /d true
   ```

4. 将最新的驱动程序包下载到实例并解压缩 zip 存档。
5. 运行 `AWSPVDriverSetup.msi`。

   在运行 MSI 后，实例将自动重启，然后升级驱动程序。实例将有长达 15 分钟的时间不可用。

6. 在升级完成并且实例在 Amazon EC2 控制台中通过了两项健康检查后，使用远程桌面连接到实例。打开磁盘管理以审核所有脱机辅助卷，并按照之前记录的对应盘符和文件夹映射将其联机。

   您必须按以下格式指定用户名来连接到实例：hostname\administrato。例如，Win2k12TestBox\administrator。

7. 运行以下命令删除 DSRM 启动配置：

   ```
   bcdedit /deletevalue safeboot
   ```

8. 重启实例。
9. 要完成升级过程，请验证是否安装了新驱动程序。在设备管理器中的存储控制器下，找到 Amazon PV 存储主适配器。确认驱动程序版本与驱动程序版本历史记录表中列出的最新版本相同。有关更多信息，请参阅Amazon PV 驱动程序包历史记录。
10. 运行以下命令，从注册表中删除 **`DisableDCCheck`**：

    ```
    reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck
    ```

> **Note**
>
> 如果您之前使用 Netsh 为 Citrix PV 驱动程序禁用了 TCP 分载，建议您在升级到 Amazon PV 驱动程序后重新启用此功能。Citrix 驱动程序的 TCP 分载问题在 Amazon PV 驱动程序中不会出现。因此，通过使用 Amazon PV 驱动程序，TCP 分载可提供更高的性能。

升级 Windows Server 2008 和 2008 R2 实例（Red Hat 到 Citrix PV 的升级）

开始将 Red Hat 驱动程序升级为 Citrix PV 驱动程序之前，请务必执行以下操作：

- 安装最新版本的 EC2Config 服务。有关更多信息，请参阅安装最新版的 EC2Config。
- 验证您是否安装了 Windows PowerShell 3.0。要验证已安装的版本，请在 PowerShell 窗口中运行以下命令：

  ```
  PS C:\> $PSVersionTable.PSVersion
  ```

  Windows PowerShell 3.0 捆绑在 Windows 管理框架 (WMF) 3.0 版安装包中。如果您需要安装 Windows PowerShell 3.0，请参阅 Microsoft 下载中心中的 Windows Management Framework 3.0。

- 备份实例上的重要信息或者从实例创建一个 AMI。有关创建 AMI 的更多信息，请参阅创建 Amazon EBS-backed AMI。

> **Tip**
>
> 您可以使用 `AWS-CreateImage` 运行手册，通过 Systems Manager Automation 创建 AMI，而不是从 Amazon EC2 控制台创建 AMI。有关更多信息，请参阅 Amazon Systems Manager Automation 运行手册参考用户指南中的 AWS-CreateImage。

如果您要创建 AMI，请务必执行以下操作：

- 写下您的密码。
- 请勿手动运行 Sysprep 工具或使用 EC2Config 服务。
- 将您的以太网适配器设置为使用 DHCP 自动获取 IP 地址。

升级 Red Hat 驱动程序

1. 连接到您的实例并以本地管理员身份登录。有关连接到实例的更多信息，请参阅使用 RDP 连接到 Windows 实例。
2.
3. 将升级包的内容提取到您所选的位置。
4. 双击 Upgrade.bat 文件。如果您收到安全警告，请选择 Run。
5. 在 Upgrade Drivers 对话框中，查看此信息并选择 Yes (如果您已开始升级)。
6. 在 Red Hat Paravirtualized Xen Drivers for Windows uninstaller 对话框中，选择是，删除 Red Hat 软件。您的实例会重新启动。

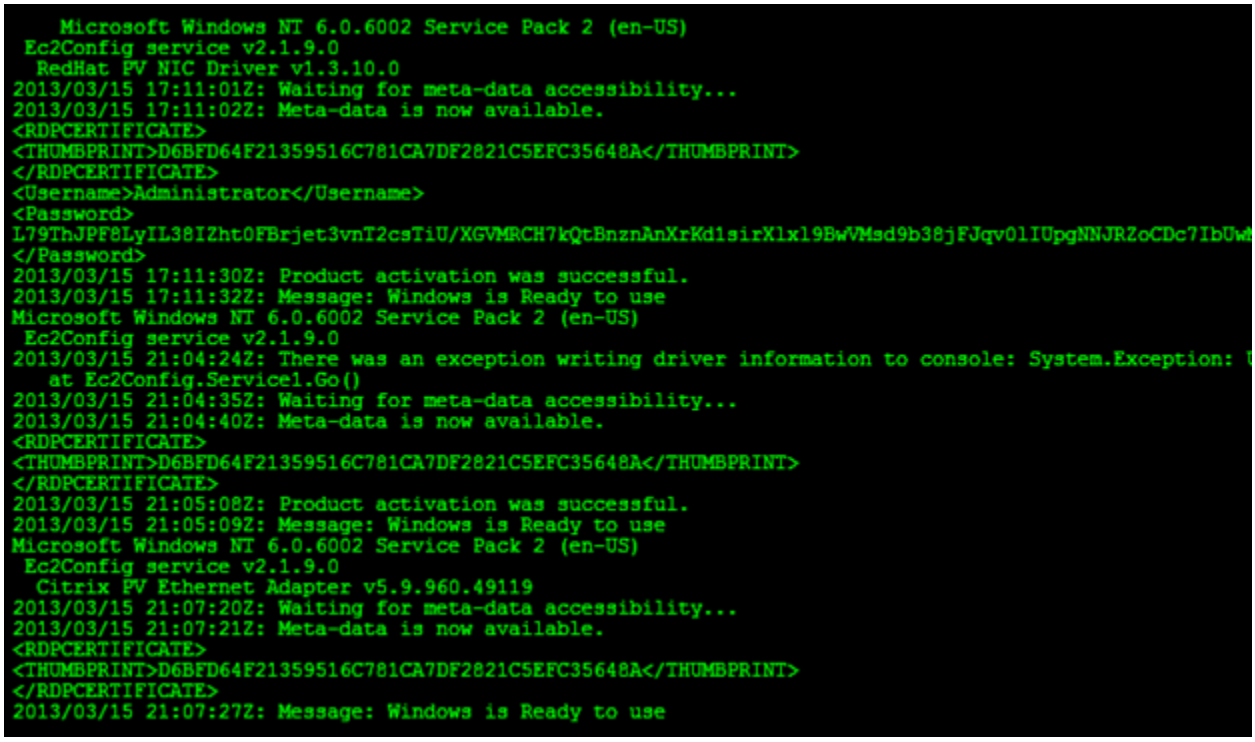
> **Note**
>
> 如果您没有看到卸载程序对话框，请选择 Windows 任务栏中的 Red Hat Paravirtualize。

7. 检查实例是否已重新启动并且可以使用。

   a. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

b. 在实例页面上，依次选择操作、监控和排查，然后选择获取系统日志。

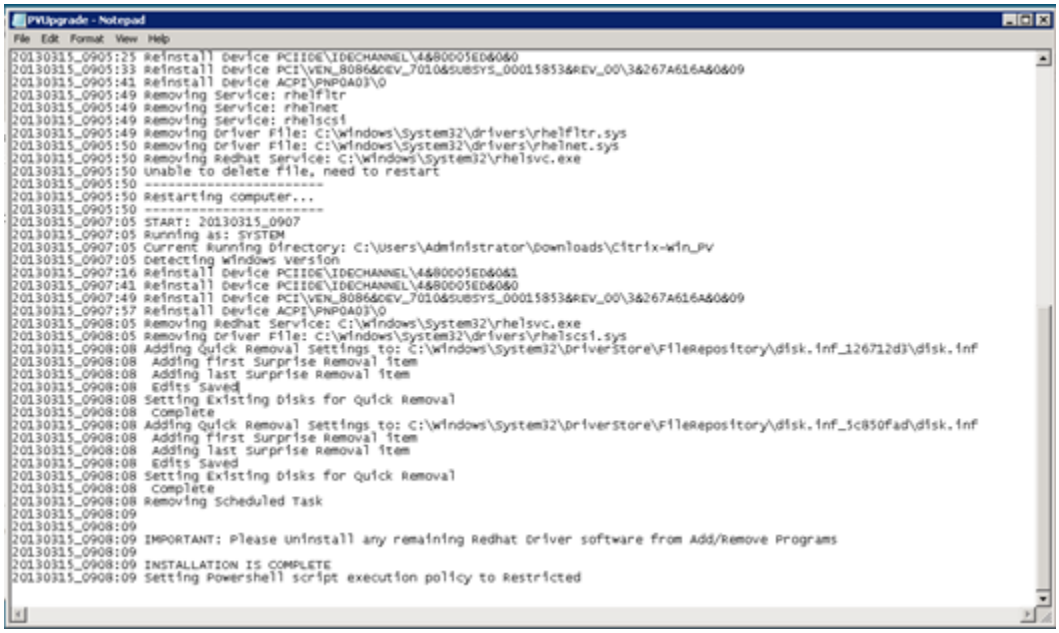
c. 升级操作应该已重新启动服务器 3 到 4 次。您可以在日志文件中看到此信息 (依据 `Windows is Ready to use` 的显示次数)。



8. 连接到您的实例并以本地管理员身份登录。

9. 关闭 Red Hat Paravirtualized Xen Drivers for Windows uninstaller 对话框。

10. 确认安装已完成。导航至您之前提取的 `Citrix-WIN_PV` 文件夹，打开 `PVUpgrade.log` 文件，然后检查是否有文本 `INSTALLATION IS COMPLETE`。



升级 Citrix Xen 代理客户服务

如果您要在 Windows Server 上使用 Citrix PV 驱动程序，则可以升级 Citrix Xen 来宾代理服务。此 Windows 服务会处理相关任务，例如通过 API 进行的关闭和重启事件。您可以在任意版本的 Windows Server 上运行此升级包，前提是该实例正在运行 Citrix PV 驱动程序。

> **Important**
>
> 对于 Windows Server 2008 R2 及更高版本，我们建议您升级到包含 Guest Agent 更新的 Amazon PV 驱动程序。

升级您的驱动程序之前，请确保您已经备份实例上的重要信息，或者从实例创建了一个 AMI。有关创建 AMI 的更多信息，请参阅创建 Amazon EBS-backed AMI。

> **Tip**
>
> 您可以使用 `AWS-CreateImage` 运行手册，通过 Systems Manager Automation 创建 AMI，而不是从 Amazon EC2 控制台创建 AMI。有关更多信息，请参阅 Amazon Systems Manager Automation 运行手册参考用户指南中的 AWS-CreateImage。

如果您要创建 AMI，请务必执行以下操作：

- 不在 EC2Config 服务中启用 Sysprep 工具。
- 写下您的密码。
- 将您的以太网适配器设为 DHCP。

升级 Citrix Xen 代理客户服务

1. 连接到您的实例并以本地管理员身份登录。有关连接到实例的更多信息，请参阅使用 RDP 连接到 Windows 实例。
2.
3. 将升级包的内容提取到您所选的位置。
4. 双击 Upgrade.bat 文件。如果您收到安全警告，请选择 Run。
5. 在 Upgrade Drivers 对话框中，查看此信息并选择 Yes (如果您已开始升级)。
6. 升级完成后，`PVUpgrade.log` 文件会打开并包含文本 `UPGRADE IS COMPLETE`。
7. 重新启动您的实例。

排查 Windows 实例上的 PV 驱动程序问题

以下是您可能遇到的有关较旧的 Amazon EC2 映像和 PV 驱动程序的问题的解决方案。

目录

- Windows Server 2012 R2 在实例重启后丢失网络和存储连接
- TCP 分载
- 时间同步
- 利用 20,000 多个磁盘 IOPS 由于 CPU 瓶颈而发生降级的工作负载

Windows Server 2012 R2 在实例重启后丢失网络和存储连接

> **Important**
>
> 仅 2014 年 9 月之前提供的 AMI 会出现此问题。

在 2014 年 9 月 10 日之前提供的 Windows Server 2012 R2 亚马逊机器映像（AMI）可能在实例重启后丢失网络和存储连接。Amazon Web Services 管理控制台系统日志中的错误指明：“Difficulty

detecting PV driver details for Console Output”。连接丢失是由即插即用清理功能导致。此功能每 30 天扫描一次不活动的系统设备并禁用它们。此功能错误地将 EC2 网络设备识别为不活动状态并将其从系统中删除。出现这种情况时，实例在重启后会丢失网络连接。

对于您怀疑可能受此问题影响的系统，您可以下载并运行就地驱动程序升级。如果您无法执行就地驱动程序升级，则可以运行帮助程序脚本。该脚本将确定您的实例是否受影响。如果实例受影响，并且 Amazon EC2 网络设备尚未移除，该脚本将禁用即插即用清理扫描。如果网络设备已移除，该脚本将修复设备，禁用即插即用清理扫描，并可让您的实例重新启动并启用网络连接。

目录

- 选择如何解决问题
- 方法 1 - 增强联网
- 方法 2 - 注册表配置
- 运行修正脚本

选择如何解决问题

有两种方法可以恢复与受此问题影响的实例的网络和存储连接。选择以下方法之一：

方法	先决条件	过程概述
方法 1 - 增强联网	增强联网仅在需要 C3 实例类型的 Virtual Private Cloud (VPC) 中可用。如果服务器当前不使用 C3 实例类型，则必须临时更改它。	您应将服务器实例类型更改为 C3 实例。增强联网随后可让您连接到受影响的实例并修复问题。修复问题后，您将实例更改回原始的实例类型。此方法通常比方法 2 更快，导致用户错误的可能性也小。只要 C3 实例运行，您就会产生额外费用。
方法 2 - 注册表配置	能够创建或访问辅助服务器。能够更改注册表设置。	您将根卷从受影响的实例分离，将其附加到其他实例，连接该实例，然后在注册表中进行更改。只要额外的服务器运行，您就会产生额外费用。此方法比方法 1 慢一些，但此方

方法	先决条件	过程概述
		法在方法 1 无法解决问题的情况下有效。

方法 1 - 增强联网

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 找到受影响的实例。选择实例并选择 Instance state (实例状态)，然后选择 Stop instance (停止实例)。

   > **Warning**
   >
   > 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

4. 在实例停止后创建备份。选择实例，再依次选择 Actions (操作)、Image and templates (映像和模板)，然后选择 Create image (创建映像)。
5. 将实例类型更改为任何 C3 实例类型。
6. 启动实例。
7.
8. 提取文件夹的内容，然后运行 `AWSPVDriverSetup.msi`。

   在运行 MSI 后，实例将自动重启，然后升级驱动程序。实例将有长达 15 分钟的时间不可用。
9. 在升级已完成并且实例在 Amazon EC2 控制台中通过了两项健康检查后，请使用远程桌面连接到实例并验证新驱动程序是否已安装。在设备管理器中的存储控制器下，找到 Amazon PV 存储主适配器。确认驱动程序版本与驱动程序版本历史记录表中列出的最新版本相同。有关更多信息，请参阅Amazon PV 驱动程序包历史记录。
10. 停止实例并将实例更改回原始的实例类型。
11. 启动实例并恢复正常使用。

方法 2 - 注册表配置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择实例。
3. 找到受影响的实例。选择实例，再选择 Instance state (实例状态)，然后选择 Stop instance （停止实例)。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

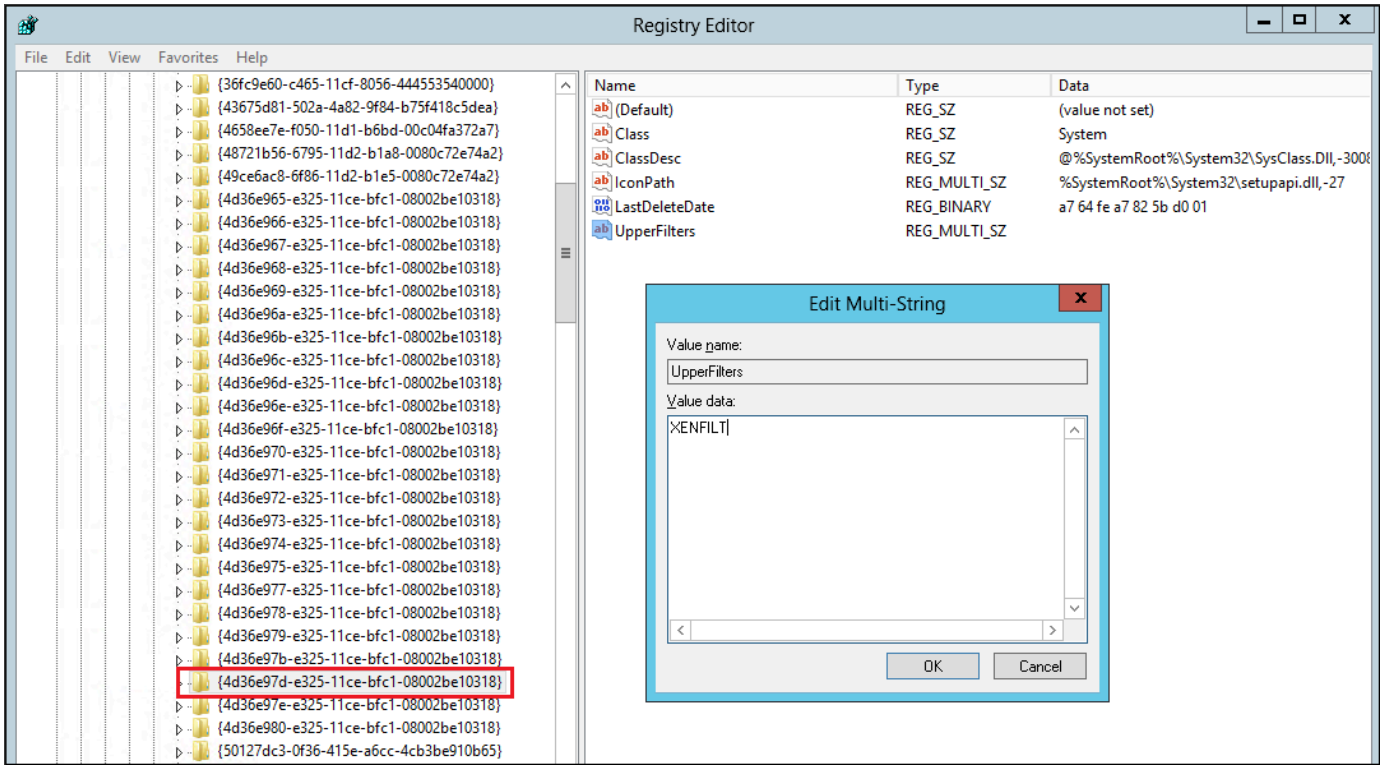
4. 选择 Launch Instance (启动实例) 并在与受影响实例相同的可用区中创建临时的 Windows Server 2008 或 Windows Server 2012 实例。不要创建 Windows Server 2012 R2 实例。

> **Important**
>
> 如果您未在与受影响实例相同的可用区中创建该实例，则无法将受影响的实例的根卷附加到新实例。

5. 在导航窗格中，选择 Volumes。
6. 找到受影响的实例的根卷。分离卷并向之前创建的临时实例附加卷。使用默认设备名称 (xvdf) 附加该卷。
7. 使用远程桌面连接临时实例，然后通过磁盘管理实用工具使该卷可供使用。
8. 在临时实例上，打开运行对话框，键入 **`regedit`**，然后按 Enter。
9. 在注册表编辑器导航窗格中，选择 HKEY_Local_Machine，然后从 File（文件）菜单中选择 Load Hive（加载 Hive）。
10. 在 Load Hive（加载 Hive）对话框中，导航到Affected Volume（受影响的卷）\Windows\System32\config\System 并在 Key Name（项名称）对话框中键入临时名称。例如，输入 OldSys。
11. 在注册表编辑器的导航窗格中，找到以下项：

    HKEY_LOCAL_MACHINE*#####*\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

    HKEY_LOCAL_MACHINE*#####*\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318
12. 对于每个项，双击 UpperFilters，输入值 XENFILT，然后选择 OK。



13. 找到以下项：

    HKEY_LOCAL_MACHINE*#####*\ControlSet001\Services\XENBUS\Parameters

14. 使用名称 ActiveDevice 和以下值创建新字符串 (REG_SZ)：

    PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. 找到以下项：

    HKEY_LOCAL_MACHINE*#####*\ControlSet001\Services\XENBUS

16. 将 Count 从 0 更改为 1。
17. 找到并删除以下项：

    HKEY_LOCAL_MACHINE*#####*\ControlSet001\Services\xenvbd\StartOverride

    HKEY_LOCAL_MACHINE *#####*\ControlSet001\Services\xenfilt\StartOverride

18. 在注册表编辑器导航窗格中，选择您在首次打开注册表编辑器时创建的临时项。
19. 从 File（文件）菜单中，选择 Unload Hive（卸载 Hive）。
20. 在磁盘管理实用工具中，选择您之前附加的驱动器，打开上下文（右键单击）菜单，然后选择脱机。

21. 在 Amazon EC2 控制台中，将受影响的实例从临时实例分离，然后将其重新附加到您的具有设备名称 /dev/sda1 的 Windows Server 2012 R2 实例。您必须指定此设备名称才能将卷指派为根卷。
22. 启动实例。
23.
24. 提取文件夹的内容，然后运行 `AWSPVDriverSetup.msi`。

    在运行 MSI 后，实例将自动重启，然后升级驱动程序。实例将有长达 15 分钟的时间不可用。
25. 在升级已完成并且实例在 Amazon EC2 控制台中通过了两项健康检查后，请使用远程桌面连接到实例并验证新驱动程序是否已安装。在设备管理器中的存储控制器下，找到 Amazon PV 存储主适配器。确认驱动程序版本与驱动程序版本历史记录表中列出的最新版本相同。有关更多信息，请参阅Amazon PV 驱动程序包历史记录。
26. 删除或停止您在本过程中创建的临时实例。

运行修正脚本

如果您无法执行就地驱动程序升级或无法迁移到较新的实例，则可以运行修正脚本来修复即插即用清理任务导致的问题。

运行修正脚本

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择要为其运行修正脚本的实例。选择 Instance state (实例状态)，然后选择 Stop instance （停止实例)。

   > **Warning**
   >
   > 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

4. 在实例停止后创建备份。选择实例，再依次选择 Actions (操作)、Image and templates (映像和模板)，然后选择 Create image (创建映像)。
5. 选择 Instance state (实例状态)，然后选择 Start instance （开始实例)。
6.
7. 提取文件夹的内容。

8. 根据 Readme.txt 文件中的指示运行修正脚本。该文件位于您提取 RemediateDriverIssue.zip 的文件夹中。

TCP 分载

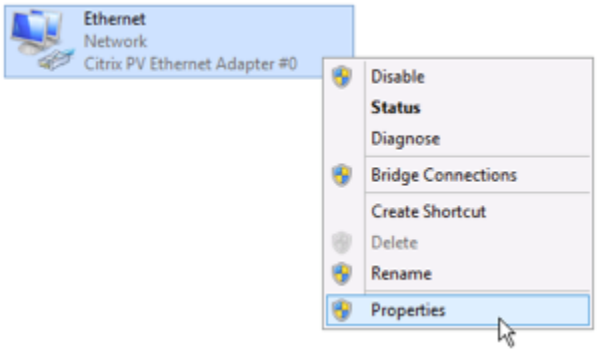
> **Important**
>
> 此问题不适用于运行 Amazon PV 或 Intel 网络驱动程序的实例。

默认情况下，会为 Windows AMI 中的 Citrix PV 驱动程序启用 TCP 分载。如果您遇到传输级错误或数据包传输错误（在 Windows 性能监视器上可见，例如在运行特定 SQL 工作负载时），可能需要禁用此功能。

> **Warning**
>
> 禁用 TCP 分载可能会降低实例的网络性能。

为 Windows Server 2012 和 2008 禁用 TCP 分载

1. 连接到您的实例并以本地管理员身份登录。
2. 如果您使用 Windows Server 2012，请按 Ctrl+Esc 访问 Start 屏幕，然后选择 Control Panel。如果您使用 Windows Server 2008，请选择 Start 并选择 Control Panel。
3. 依次选择 Network and Internet 和 Network and Sharing Center。
4. 选择 Change adapter settings。
5. 右键单击 Citrix PV Ethernet Adapter #0 (Citrix PV 以太网适配器 #0)，然后选择 Properties (属性)。



6. 在 Local Area Connection Properties 对话框中，选择 Configure 以打开 Citrix PV Ethernet Adapter #0 Properties 对话框。
7. 在 Advanced 选项卡上，禁用每个属性，但 Correct TCP/UDP Checksum Value 除外。要禁用一个属性，请从 Property 中选择该属性，然后从 Value 中选择 Disabled。
8. 选择 OK (确定)。
9. 从命令提示符窗口运行以下命令。

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

10. 重启实例。

时间同步

在 2013.02.13 Windows AMI 版本前，Citrix Xen 代理客户无法正确设置系统时间。这可能导致您的 DHCP 租约过期。如果您在连接到您的实例时遇到问题，可能需要更新代理。

要确定您是否有更新的 Citrix Xen 客户代理，请检查 `C:\Program Files\Citrix\XenGuestAgent.exe` 文件的日期是否为 2013 年 3 月。如果此文件的日期早于这个时间，则请更新 Citrix Xen 代理客户服务。有关更多信息，请参阅升级 Citrix Xen 代理客户服务。

利用 20,000 多个磁盘 IOPS 由于 CPU 瓶颈而发生降级的工作负载

如果您使用的是运行 Amazon PV 驱动程序的 Windows 实例（利用 20000 多个 IOPS），则可能会受到此问题的影响，并且会遇到错误代码 `0x9E: USER_MODE_HEALTH_MONITOR`。

Amazon PV 驱动程序中的磁盘读取和写入 (IO) 分两个阶段进行：IO 准备和 IO 完成。默认情况下，准备阶段在单个任意核心上运行。完成阶段在核心 `0` 上运行。处理 IO 所需的计算量因其大小和其他属性而异。有些 IO 在准备阶段使用更多计算，其他 IO 则在完成阶段使用更多计算。当实例驱动超过 20,000 个 IOPS 时，准备阶段或完成阶段可能会导致瓶颈，其中运行实例的 CPU 达到 100% 容量。准备阶段或完成阶段是否成为瓶颈取决于应用程序使用的 IO 的属性。

从 Amazon PV 驱动程序 8.4.0 开始，准备阶段和完成阶段的负载可以跨多个核心分配，从而消除瓶颈。每个应用程序使用不同的 IO 属性。因此，应用以下配置之一可能会提高、降低或不影响应用程序的性能。应用这些配置中的任何一个之后，监控应用程序以验证它是否符合所需的性能。

1. 先决条件

   在开始此故障排除过程之前，请验证以下先决条件：

   - 您的实例使用 Amazon PV 驱动程序 8.4.0 或更高版本。要升级，请参阅 在 EC2 Windows 实例上升级半虚拟化驱动程序。
   - 您对实例拥有 RDP 访问权限。使用 RDP 连接到 Windows 实例的步骤，请参阅 使用 RDP 客户端连接到 Windows 实例。
   - 您对实例拥有管理员访问权限。

2. 观察实例的 CPU 负载

   您可以使用 Windows 任务管理器查看每个 CPU 的负载，以确定磁盘 IO 的潜在瓶颈。

   1. 验证应用程序是否正在运行并处理与生产工作负载类似的流量。
   2. 使用 RDP 连接到您的实例。
   3. 选择实例上的开始菜单。
   4. 进入开始菜单中的 `Task Manager` 以打开任务管理器。
   5. 如果任务管理器显示摘要视图，请选择更多详细信息以展开详细视图。
   6. 选择 Performance (性能) 选项卡。
   7. 在左侧窗格中，选择 CPU。
   8. 右键单击主窗格中的图表，然后选择将图表更改为>逻辑处理器以显示每个单独的核心。
   9. 根据实例上的核心数量，您可能会看到显示随时间变化的 CPU 负载的行，也可能只看到一个数字。
      - 如果您看到显示随时间变化的负载的图表，请查找显示框几乎完全变成阴影的 CPU。
      - 如果您在每个核心上看到一个数字，请查找始终显示 95% 或更高的核心。

   10注意核心 `0` 或者另一个核心是否正在经历重负载。

3. 选择要应用的配置

配置名称	应用此配置的时间	备注
Default configuration	工作负载驱动的 IOPS 不到 20,000 个，或者其他配置没有提高性能或稳定性。	对于此配置，IO 发生在几个核心上，这可能会通过增加缓存区域和减少上下文切换而使较小的工作负载受益。

配置名称	应用此配置的时间	备注
Allow driver to choose whether to distribute completion	工作负载驱动了 20,000 多个 IOPS，核心 0 上观察到中等或高负载。	无论是否遇到问题，建议使用 PV 8.4.0 或更高版本且利用 20,000 多个 IOPS 的所有 Xen 实例使用此配置。
Distribute both preparation and completion	工作负载驱动了 20,000 多个 IOPS，要么是允许驱动程序选择分配并没有提高性能，要么是 0 之外的其他核心正在经历高负载。	此配置支持 IO 准备和 IO 完成的分配。

> **Note**
>
> 我们建议您在没有同时分配 IO 完成的情况下不要分配 IO 准备（设置 DpcRedirection 而不设置 NotifierDistributed），因为在准备阶段并行运行时，完成阶段对准备阶段过载敏感。

注册表项的值

- NotifierDistributed

  值 0 或不存在 — 完成阶段将在核心 0 上运行。

  值 1 — 驱动程序选择运行完成阶段或核心 0 或者每个附加磁盘一个额外核心。

  值 2 — 驱动程序在每个附加磁盘的额外核心上运行完成阶段。

- DpcRedirection

  值 0 或不存在 — 准备阶段将在单个、任意的核心上运行。

  值 1 — 准备阶段跨多个核心分配。

默认配置

对于 8.4.0 之前的 Amazon PV 驱动程序版本，或者如果在应用本节中的一个其他配置后观察到性能或稳定性降级，则应用默认配置。

1. 使用 RDP 连接到您的实例。
2. 以管理员身份打开新的 PowerShell 命令提示符。
3. 运行以下命令以删除 `NotifierDistributed` 和 `DpcRedirection` 注册表项。

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name NotifierDistributed
```

```
Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Name DpcRedirection
```

4. 重新启动您的实例。

允许驱动程序选择是否分配完成

设置 `NotiferDistributed` 注册表项，允许 PV 存储驱动程序选择是否分配 IO 完成。

1. 使用 RDP 连接到您的实例。
2. 以管理员身份打开新的 PowerShell 命令提示符。
3. 运行以下命令以设置 `NotiferDistributed` 注册表项。

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name NotifierDistributed
```

4. 重新启动您的实例。

分配准备和完成

设置 `NotifierDistributed` 和 `DpcRedirection` 注册表项以始终分配准备阶段和完成阶段。

1. 使用 RDP 连接到您的实例。

2. 以管理员身份打开新的 PowerShell 命令提示符。
3. 运行以下命令以设置 `NotifierDistributed` 和 `DpcRedirection` 注册表项。

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000002 -Name NotifierDistributed
```

```
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd
\Parameters -Value 0x00000001 -Name DpcRedirection
```

4. 重新启动您的实例。

Amazon NVMe 驱动程序

在基于 Nitro 的实例上，Amazon EBS 卷和实例存储卷显示为 NVMe 块储存设备。要为作为 NVMe 块设备公开的卷充分利用 Amazon EBS 功能的性能和功能，实例必须安装了 Amazon NVMe 驱动程序。所有当前一代 Amazon Windows AMI 都默认安装了 Amazon NVMe 驱动程序。

有关 EBS 和 NVMe 的更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 和 NVMe。有关 SSD 实例存储和 NVMe 的更多信息，请参阅适用于 EC2 实例的 SSD 实例存储卷。

Linux 实例

以下 AMI 包含所需的 NVMe 驱动程序：

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 或更高版本（采用 `linux-aws` 内核）

> **Note**
>
> Amazon 基于 Graviton 的实例类型需要 Ubuntu 18.04 或更高版本（采用 `linux-aws` 内核）

- Red Hat Enterprise Linux 7.4 或更高版本
- SUSE Linux Enterprise Server 12 SP2 或更高版本
- CentOS 7.4.1708 或更高版本
- FreeBSD 11.1 或更高版本

- Debian GNU/Linux 9 或更高版本

确认实例具有 NVMe 驱动程序

您可以使用以下命令确认您的实例是否具有 NVMe 驱动程序。

- Amazon Linux、RHEL、CentOS 和 SUSE Linux Enterprise Server

```
$ modinfo nvme
```

  如果实例具有 NVMe 驱动程序，则该命令将返回有关该驱动程序的信息。

- Amazon Linux 2 和 Ubuntu

```
$ ls /sys/module/ | grep nvme
```

  如果实例具有 NVMe 驱动程序，该命令将返回已安装的驱动程序。

更新 NVMe 驱动程序

如果您的实例具有 NVMe 驱动程序，您可以使用以下过程将驱动程序更新到最新版本。

1. 连接到您的 实例。
2. 更新程序包缓存以获取必需程序包更新，如下所示。
   - 对于 Amazon Linux 2、Amazon Linux、CentOS 和 Red Hat Enterprise Linux：

     ```
     [ec2-user ~]$ sudo yum update -y
     ```

   - 对于 Ubuntu 和 Debian：

     ```
     [ec2-user ~]$ sudo apt-get update -y
     ```

3. Ubuntu 16.04 及更高版本包含 `linux-aws` 程序包，该程序包包含基于 Nitro 的实例所需的 NVMe 和 ENA 驱动程序。升级 `linux-aws` 程序包以接收最新版本，如下所示：

   ```
   [ec2-user ~]$ sudo apt-get install --only-upgrade -y linux-aws
   ```

   对于 Ubuntu 14.04，您可以安装最新的 `linux-aws` 程序包，如下所示：

```
[ec2-user ~]$ sudo apt-get install linux-aws
```

4. 重启实例以加载最新内核版本。

```
sudo reboot
```

5. 重启之后重新连接到实例。

Windows 实例

PowerShell

如果您没有从 Amazon 提供的最新 Amazon Windows AMI 之一启动实例，请按照以下过程在实例上安装当前的 Amazon NVMe 驱动程序。此安装需要重启。安装脚本将重新引导实例，或者您必须在最后一步中重新引导实例。

先决条件

- 已安装 PowerShell 3.0 或更高版本。
- 本节中显示的命令必须在 64 位版本的 PowerShell 中运行。不要使用 x86 版本的 PowerShell。这是 32 位版本的 shell，不支持这些命令。

下载并安装最新 Amazon NVMe 驱动程序

1. 如果需要回滚更改，我们建议您按如下方式创建 AMI 作为备份。
   a. 当您停止某个实例时，任何实例存储卷上的数据都将被擦除。在停止实例之前，请确认您是否已将所需数据从实例存储卷复制到持久性存储，例如 Amazon EBS 或 Amazon S3。
   b. 在导航窗格中，选择实例。
   c. 选择需要升级驱动程序的实例，然后依次选择 Instance state (实例状态)、Stop instance (停止实例)。
   d. 实例停止后，选择实例，依次选择 Actions (操作) 、Image and templates (映像和模板)，然后选择 Create image (创建映像)。
   e. 依次选择实例状态、启动实例。
2. 连接到您的实例并以本地管理员身份登录。
3. 使用以下选项之一将驱动程序下载到您的实例：

- 浏览器 - 将最新的驱动程序包下载到实例并解压缩 zip 存档。
- PowerShell – 运行以下命令：

```
Invoke-WebRequest https://s3.cn-north-1.amazonaws.com.cn/ec2-windows-
drivers-downloads-cn/NVMe/Latest/AWSNVMe.zip -outfile $env:USERPROFILE
\nvme_driver.zip
Expand-Archive $env:userprofile\nvme_driver.zip -DestinationPath
 $env:userprofile\nvme_driver
```

如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：

```
[Net.ServicePointManager]::SecurityProtocol =
 [Net.SecurityProtocolType]::Tls12
```

4. 通过运行 nvme_driver 目录 (.\install.ps1) 中的 install.ps1 PowerShell 脚本，将驱动程序安装到您的实例中 如果出现错误，请确保您使用的是 PowerShell 3.0 或更高版本。

   a. （可选）从 Amazon NVMe 版本 1.5.0 开始，Windows Server 2016 及更高版本支持小型计算机系统接口（SCSI）永久预留。此功能增加了对带有共享 Amazon EBS 存储的 Windows Server 故障转移集群的支持。默认情况下，安装期间不启用此功能。

   在运行 install.ps1 脚本以安装驱动程序时，您可以通过指定值为 $true 的 EnableSCSIPersistentReservations 参数来启用该功能。

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $true
```

   在运行 install.ps1 脚本以安装驱动程序时，您可以通过指定值为 $false 的 EnableSCSIPersistentReservations 参数来禁用该功能。

```
PS C:\> .\install.ps1 -EnableSCSIPersistentReservations $false
```

   b. 从 Amazon NVMe 1.5.0 开始，install.ps1 脚本始终安装带有驱动程序的 ebsnvme-id 工具。

   （可选）对于版本 1.4.0、1.4.1 和 1.4.2，install.ps1 脚本允许您指定 ebsnvme-id 工具是否应与驱动程序一起安装。

i. 如需安装 ebsnvme-id 工具，请指定 InstallEBSNVMeIdTool ‘Yes’。

ii. 如果不希望安装此工具，请指定 InstallEBSNVMeIdTool ‘No’。

如果您未指定 InstallEBSNVMeIdTool，且 C:\ProgramData\Amazon\Tools 中已存在该工具，则软件包将默认升级该工具。如果该工具不存在，则 install.ps1 将默认不升级该工具。

如果您不想将该工具作为软件包的一部分进行安装，而想稍后再安装，则可以在驱动程序包中找到最新版本或该工具。或者，您可以从 Amazon S3 下载版本 1.0.0：

下载 ebsnvme-id 工具。

5. 如果安装程序没有重新引导实例，请重新引导该实例。

Distributor

您可以使用分发服务器（Amazon Systems Manager 的一项功能）一次性安装 NVMe 驱动程序包，也可以使用计划更新来安装。

安装最新的 Amazon NVMe 驱动程序

1. 有关如何使用分发服务器安装 NVMe 驱动程序包的说明，请参阅《Amazon EC2 Systems Manager 用户指南》中的安装或更新软件包。
2. 在安装类型中，选择卸载并重新安装。
3. 在名称中，选择 AWSNVMe。
4. （可选）在其他参数,中，通过指定值来自定义安装。必须使用有效的 JSON 语法来格式化这些值。有关如何为 aws configure 软件包传递其他参数的示例，请参阅命令文档插件参考。

   a. 从 Amazon NVMe 1.5.0 开始，该驱动程序支持 Windows Server 2016 及更高版本的 SCSI 永久预留。默认情况下，安装期间不启用此功能。

      - 要启用此功能，请指定 {"SSM_EnableSCSIPersistentReservations": "true"}。
      - 如果您不想启用此功能，请指定 {"SSM_EnableSCSIPersistentReservations": "false"}。

   b. 从 Amazon NVMe 1.5.0 开始，install.ps1 脚本将始终安装 ebsnvme-id 工具。

（可选）对于版本 1.4.0、1.4.1 和 1.4.2，install.ps1 脚本允许您指定 ebsnvme-id 工具是否应与驱动程序一起安装。

- 要安装 ebsnvme-id 工具，请指定 {"SSM_InstallEBSNVMeIdTool": "Yes"}。
- 如果不希望安装此工具，请指定 {"SSM_InstallEBSNVMeIdTool": "No"}。

  如果您未指定 Additional Arguments（其它参数）为 SSM_InstallEBSNVMeIdTool，且 C:\ProgramData\Amazon\Tools 中已存在该工具，则软件包将默认升级该工具。如果该工具不存在，则软件包将默认不升级该工具。

  如果您不想将该工具作为软件包的一部分进行安装，而想稍后再安装，则可以在驱动程序包中找到该工具的最新版本。或者，您可以从 Amazon S3 下载版本 1.0.0：

  下载 ebsnvme-id 工具。

5. 如果安装程序没有重新引导实例，请重新引导该实例。

为 Windows 实例配置 SCSI 永久预留

安装 Amazon NVMe 驱动程序版本 1.5.0 或更高版本后，您可以使用适用于 Windows Server 2016 及更高版本的 Windows 注册表启用或禁用 SCSI 永久预留。您必须先重启数据库实例，然后这些注册表更改才会生效。

您可以使用以下命令启用 SCSI 永久预留，该命令将 EnableSCSIPersistentReservations 的值设置为 1。

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 1
```

您可以使用以下命令禁用 SCSI 永久预留，该命令将 EnableSCSIPersistentReservations 的值设置为 0。

```
PS C:\> $registryPath = "HKLM:\SYSTEM\CurrentControlSet\Services\AWSNVMe\Parameters
\Device"
Set-ItemProperty -Path $registryPath -Name EnableSCSIPersistentReservations -Value 0
```

Amazon NVMe Windows 驱动程序版本历史记录

下表显示在 Amazon EC2 上每个版本的 Windows Server 上运行的 Amazon NVMe 驱动程序。

Windows Server 版本	Amazon NVME 驱动程序版本
Windows Server 2025	最新版本
Windows Server 2022	最新版本
Windows Server 2019	最新版本
Windows Server 2016	最新版本
Windows Server 2012 R2	版本 1.5.1 及更早版本
Windows Server 2012	版本 1.5.1 及更早版本
Windows Server 2008 R2	版本 1.3.2 及更早版本
Windows Server 2008	版本 1.3.2 及更早版本

下表说明已发行的 Amazon NVMe 驱动程序版本。

程序包版本	驱动程序版本	详细信息	发行日期
1.8.1	1.8.1	• 修复了磁盘序列号生成方面的向后兼容性问题。	2026 年 2 月 24 日
1.8.0	1.8.0	• 增加了对 IOCTL 的支持，以便在实现动态命名空间管理的设备上运行。有关更多信息，请参阅 Microsoft 文档中的 IOCTL_SCSI_MINIPORT IOCTL。 • 修复了“获取日志页面”命令和序列号生成中的错误。	2026 年 1 月 16 日
1.7.0	1.7.0	• 添加了对 NVMe 获取日志页面命令的支持。 • 添加了对 EBS 和 EC2 实例存储卷详细性能统计数据的支持。 •	2025 年 9 月 17 日

程序包版本	驱动程序版本	详细信息	发行日期
		修复了识别命名空间命令的错误。	
1.6.0	1.6.0	• 更新了安装脚本以使用 PnPUtil。 • 更新了 ebsnvme-id.exe 以使用 NVMe IOCTL。	2024 年 10 月 25 日
1.5.1	1.5.0	修复了安装脚本，以便在 `ebsnvme-id` 工具不存在时为其创建文件夹。	2023 年 11 月 17 日
1.5.0	1.5.0	为运行 Windows Server 2016 及更高版本的实例增加对小型计算机系统接口（SCSI）永久保留的支持。ebsnvme-id 工具（`ebsnvme-id.exe` ）现在已默认安装。	2023 年 8 月 31 日
1.4.2	1.4.2	修复了 Amazon NVMe 驱动程序 不支持 D3 实例上实例存储卷的错误。	2023 年 3 月 16 日
1.4.1	1.4.1	报告支持此可选 NVMe 功能的 EBS 卷的命名空间首选写入粒度（NPGW）。有关更多信息，请参阅 NVMe 基本规范版本 1.4 中的第 8.25 部分“通过 I/O 大小和对齐依从性改进性能”。	2022 年 5 月 20 日

程序包版本	驱动程序版本	详细信息	发行日期
1.4.0	1.4.0	• 增加了对 IOCTL 的支持，允许应用程序与 NVMe 设备进行交互。此支持允许应用程序从 NVMe 设备处获得 `IdentifyController` 、`IdentifyNamespace` 和 `NameSpace` 的列表。有关更多信息，请参阅 Microsoft 文档中的协议特定的查询。 • 1.4.0 驱动程序版本和最新 `ebsnvme-id` 工具 (`ebsnvme-id.exe`) 组合包含于同一个软件包中。这种组合使您能够通过单个软件包即可同时安装驱动程序和工具。有关更多详细信息，请参阅Amazon NVMe 驱动程序。 • 错误修复与可靠性改进。	2021 年 11 月 23 日
1.3.2	1.3.2	修复了与修改主动处理 IO 的 EBS 卷有关的问题，此问题可能导致数据损坏。不修改在线 EBS 卷（例如，调整大小或更改类型）的客户不会受到影响。 这是最后一个可以在 Windows Server 2008 和 2008 R2 上运行的版本。此版本可供下载，但不再受支持。Windows Server 2008 和 2008 R2 的生命周期已经终止，Microsoft 不再提供支持。	2019 年 9 月 10 日
1.3.1	1.3.1	可靠性改进。	2019 年 5 月 21 日
1.3.0	1.3.0	设备优化改进。	2018 年 8 月 31 日
1.2.0	1.2.0	对所有受支持的实例（包括裸机实例）上的 Amazon NVMe 设备的性能和可靠性改进。	2018 年 6 月 13 日
>1.0.0	>1.0.0	运行 Windows Server 的受支持实例类型的 Amazon NVMe 驱动程序。	2018 年 2 月 12 日

订阅 通知

Amazon SNS 可在 EC2 Windows 驱动程序的新版本发布时向您发送通知。使用以下过程订阅这些通知。

从控制台订阅 EC2 通知

1. 通过以下网址打开 Amazon SNS 控制台：https://console.aws.amazon.com/sns/v3/home。
2. 如果需要，可在导航栏中将区域更改为美国东部（弗吉尼亚北部）。您必须选择此区域，因为您订阅的 SNS 通知是在此区域中创建的。
3. 在导航窗格中，选择 Subscriptions。
4. 选择 Create subscription。
5. 在 Create subscription 对话框中，执行以下操作：
   a. 对于 TopicARN，复制以下 Amazon Resource Name (ARN)：

      arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers
   b. 对于 Protocol，选择 `Email`。
   c. 对于 Endpoint，键入可用于接收通知的电子邮件地址。
   d. 选择 Create subscription。
6. 您将收到一封确认电子邮件。打开电子邮件，然后按照说明操作以完成订阅。

每当发布新的 EC2 Windows 驱动程序时，我们都会向订户发送通知。如果您不希望再收到这些通知，请通过以下步骤取消订阅。

从 Amazon EC2 Windows 驱动程序通知中取消订阅

1. 通过以下网址打开 Amazon SNS 控制台：https://console.aws.amazon.com/sns/v3/home。
2. 在导航窗格中，选择 Subscriptions。
3. 选中订阅的复选框，然后选择 Actions、Delete subscriptions。当系统提示进行确认时，选择 Delete。

使用Amazon CLI 订阅 EC2 通知

要使用 Amazon CLI 订阅 EC2 通知，请使用以下命令。

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers --
protocol email --notification-endpoint YourUserName@YourDomainName.ext
```

使用 Amazon Tools for Windows PowerShell 订阅 EC2 通知

要使用 Amazon Tools for Windows PowerShell 订阅 EC2 通知，请使用以下命令。

```
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-
drivers' -Protocol email -Region us-east-1 -Endpoint 'YourUserName@YourDomainName.ext'
```

配置您的 Amazon EC2 Windows 实例

启动 Windows 实例后，您可以以管理员身份登录，以执行 Windows 功能和系统设置的其他配置。EC2 Windows 故障排除实用程序 可以帮助您排查实例上的问题。

您可以按如下方式配置 Windows 启动代理和其他 Windows 特定功能。

Windows 启动代理

每个 Amazon Windows AMI（以及 Amazon Web Services Marketplace 上提供的许多其他 AMI）都包含一个已预先配置默认设置的 Windows 启动代理。启动代理在实例启动期间执行任务，并可在实例停止并稍后启动或重新启动时运行。

适用于 Windows 的 EC2 快速启动

必须对每个 Amazon EC2 Windows 实例执行标准的 Windows 操作系统（OS）启动步骤，其中包括多次重启，通常需要 15 分钟或更长时间才能完成。启用了 EC2 Fast Launch 功能的 Amazon EC2 Windows Server AMI 会提前完成其中一些步骤并重启，以便缩短启动实例所需的时间。

Windows 特定的系统设置

以下列表包括一些仅适用于 Windows 操作系统的系统设置：

更改 Windows 管理员密码

在连接到 Windows 实例时，您必须指定有权访问该实例的用户账户以及密码。首次连接到实例时，必须使用管理员账户并提供默认密码。在首次连接到实例后，建议您更改管理员密码的默认值。

添加 Windows 系统组件

Windows Server 操作系统包括许多可选组件。在每个 Amazon Windows Server AMI 中包含所有可选组件是不实际的。作为替代，我们提供安装介质 EBS 快照，这些快照具有在您的 Windows 实例上配置或安装组件所需的文件。

在 Windows 上安装 WSL

适用于 Linux 的 Windows 子系统（WSL）可以免费下载并在 Windows 实例上安装。安装 WSL 后，除传统的 Windows 桌面系统外，您可以直接在 Windows 实例上运行原生的 Linux 命令行工具，并使用 Linux 工具来编写脚本。您可以在单个 Windows 实例上轻松切换 Linux 和 Windows 系统，这在开发环境中可能非常实用。

适用于 Windows 实例的 Amazon 设备驱动程序

可以为 Windows 实例更新 Amazon 设备驱动程序。有关更多信息，请参阅 the section called “管理设备驱动程序”。

下表按 Windows 版本总结了基于 Nitro 的实例支持的驱动程序。

版本	存储驱动程序	增强的网络驱动程序
Windows Server 2025	Amazon NVMe 最新版本	ENA 最新版本
Windows Server 2022	Amazon NVMe 最新版本	ENA 最新版本
Windows Server 2019	Amazon NVMe 最新版本	ENA 最新版本
Windows Server 2016	Amazon NVMe 最新版本	ENA 最新版本
Windows Server 2012 R2	Amazon NVMe 版本 1.5.1	ENA 版本 2.6.0
Windows Server 2008 R2	Amazon NVMe 版本 1.3.2	ENA 版本 2.2.3

下表按 Windows 版本总结了基于 Xen 的实例支持的驱动程序。

版本	存储驱动程序	增强的网络驱动程序
Windows Server 2022	Amazon PV 最新版本	

版本	存储驱动程序	增强的网络驱动程序
		• ENA 最新版本 [1] • Intel VF [2] • Amazon PV 最新版本 [3]
Windows Server 2019	Amazon PV 最新版本	• ENA 最新版本 [1] • Intel VF [2] • Amazon PV 最新版本 [3]
Windows Server 2016	Amazon PV 最新版本	• ENA 最新版本 [1] • Intel VF [2] • Amazon PV 最新版本 [3]
Windows Server 2012 R2	Amazon PV 版本 8.4.3	• ENA 版本 2.6.0 [1] • Intel VF [2] • Amazon PV 版本 8.4.3 [3]
Windows Server 2008 R2	Amazon PV 版本 8.3.5	• ENA 版本 2.2.3 [1] • Intel VF [2] • Amazon PV 版本 8.3.5 [3]

[1] 适用于实例类型 G3、H1、I3、`m4.16xlarge`、P3、P3dn 和 R4。

[2] 适用于实例类型 C3、C4、D2、I2、M4（`m4.16xlarge` 除外）和 R3。

[3] 适用于实例类型 C1、M1、M2、M3、T1、T2、X1 和 X1e。

Amazon EC2 Windows 实例上的 Windows 启动代理

每个 Amazon Windows AMI 都包含一个已预配置默认设置的 Windows 启动代理。启动代理在实例启动期间执行任务，并可在实例停止并稍后启动或重新启动时运行。有关特定代理的信息，请参阅以下列表中的详细信息页面。

有关 Amazon Windows AMI 的更多信息，请参阅 AmazonWindows AMI 参考。

- 使用 EC2Launch v2 代理在 EC2 Windows 实例启动期间执行任务
- 使用 EC2Launch v1 代理在 EC2 Windows 实例启动期间执行任务
- 使用 EC2Config 服务在 EC2 旧版 Windows 操作系统实例启动期间执行任务

内容

- 比较 Amazon EC2 启动代理
- 为 EC2 Windows 启动代理配置 DNS 后缀
- 订阅 EC2 Windows 启动代理通知
- EC2Launch v2 和 EC2Config 代理的 Windows 服务管理

比较 Amazon EC2 启动代理

下表显示了 EC2Config、EC2Launch v1 和 EC2Launch v2 之间的主要功能差异。

功能	EC2Config	EC2Launch v1	EC2Launch v2
Run as (运行方式)	Windows 服务	PowerShell 脚本	Windows 服务
支持	仅限传统操作系统	Windows Server 版本： • 2016 • 2019（LTSC 和 SAC）	Windows Server 版本： • 2016 • 2019（LTSC 和 SAC） • 2022 • 2025
配置文件	XML	JSON	JSON/YAML
设置管理员用户名	否	否	是
压缩后的用户数据	否	否	是
本地用户数据录入到 AMI 上	否	否	是的，可配置

功能	EC2Config	EC2Launch v1	EC2Launch v2
用户数据中的任务配置	否	否	是
可配置壁纸	否	否	是
自定义任务运行顺序	否	否	是
可配置任务	15	9	在启动时为 20
支持 Windows 事件查看器	是	否	是
事件查看器事件类型的数量	2	0	30

> **Note**
>
> EC2Config 文档仅供历史参考之用。Microsoft 不再支持它运行的操作系统版本。强烈建议升级到最新的启动服务。

为 EC2 Windows 启动代理配置 DNS 后缀

借助 Amazon EC2 启动代理，您可以配置 Windows 实例用于域名解析的 DNS 后缀列表。启动代理将以下值添加到 DNS 后缀搜索列表中，从而覆盖 `System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` 注册表项中的标准 Windows 设置：

- 实例的域
- 因实例域传递而产生的后缀
- NV 域
- 每个网络接口卡指定的域

所有启动代理都支持 DNS 后缀配置。有关更多信息，请参阅您的特定启动代理版本：

- 有关 `setDnsSuffix` 任务以及如何在 EC2Launch v2 中配置 DNS 后缀的信息，请参阅 setDnsSuffix。
- 有关 DNS 后缀列表设置以及如何为 EC2Launch v1 启用或禁用传递的信息，请参阅 在 Windows 实例上配置 EC2Launch v1 代理。
- 有关 DNS 后缀列表设置以及如何为 EC2Config 启用或禁用传递的信息，请参阅 EC2Config 设置文件。

域名传递

域名传递是一种 Active Directory 行为，该行为允许子域中的计算机在不使用完全限定的域名的情况下访问父域中的资源。默认情况下，域名传递会一直持续到域名进度中只剩下两个节点。

如果实例已连接到某个域，则启动代理会对域名执行传递，并将结果添加到 **System\CurrentControlSet\Services\Tcpip\Parameters\SearchList** 注册表项中维护的 DNS 后缀搜索列表中。代理使用以下注册表项中的设置来确定传递行为。

- **System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution**
  - 未设置时，禁用传递
  - 设置为 1 时，启用传递（默认）
  - 设置为 0 时，禁用传递
- **System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel**
  - 未设置时，使用等级 2（默认）
  - 设置为 3 或更大值时，使用该值来设置等级

当您禁用传递或将传递设置更改为更高级别时，`System\CurrentControlSet\Services\Tcpip\Parameters\SearchList` 注册表项仍包含先前添加的后缀。这些后缀不会被自动移除。您可以手动更新列表，也可以清除列表并让代理完成设置新列表的过程。

> **Note**
>
> 要从注册表中清除 DNS 后缀列表，可以运行以下命令。

```
PS C:\> Invoke-CimMethod -ClassName Win32_NetworkAdapterConfiguration -
MethodName "SetDNSSuffixSearchOrder" -Arguments @{ DNSDomainSuffixSearchOrder =
 $null } | Out-Null
```

传递示例

以下示例显示传递过程中的域名进度。

`corp.example.com`

- 进展到 `example.com`

`locale.region.corp.example.com`

1. 进展到 `region.corp.example.com`
2. 进展到 `corp.example.com`
3. 进展到 `example.com`

具有 `DomainNameDevolutionLevel=3` 设置的 `locale.region.corp.example.com`

1. 进展到 `region.corp.example.com`
2. 进展到 `corp.example.com`。由于等级设置，进度到此停止。

订阅 EC2 Windows 启动代理通知

Amazon SNS 可在 EC2 启动代理的新版本发布时向您发送通知。使用以下过程订阅这些通知。

订阅 EC2Config 通知

1. 通过以下网址打开 Amazon SNS 控制台：https://console.aws.amazon.com/sns/v3/home。
2. 如果需要，可在导航栏中将区域更改为美国东部（弗吉尼亚北部）。您必须选择此区域，因为您订阅的 SNS 通知是在此区域中创建的。
3. 在导航窗格中，选择 Subscriptions。
4. 选择 Create subscription。

5. 在 Create subscription 对话框中，执行以下操作：

   a. 对于主题 ARN，请使用以下 Amazon 资源名称（ARN），该名称与您想要接收通知的代理相匹配：

      - EC2Launch v2：

        ```
        arn:aws:sns:cn-north-1:512218236690:amazon-ec2launch-v2
        ```

      - EC2Launch 或 EC2Config：

        ```
        arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config
        ```

   b. 对于 Protocol（协议），选择 `Email`。

   c. 对于端点，输入您要接收通知的电子邮件地址。

   d. 选择 Create subscription。

6. 您将收到要求您确认订阅的电子邮件。打开电子邮件，然后按照说明操作以完成订阅。

当启动代理的新版本发布时，我们会向订阅用户发送通知。如果您不希望再收到这些通知，请通过以下步骤取消订阅。

取消订阅启动代理通知

1. 打开 Amazon SNS 控制台。
2. 在导航窗格中，选择 Subscriptions。
3. 选择订阅，然后依次选择操作、删除订阅。当系统提示进行确认时，选择 Delete。

EC2Launch v2 和 EC2Config 代理的 Windows 服务管理

如果您以具有管理权限的用户身份登录实例，则您可以像管理任何其他 Windows 服务一样管理 EC2Launch v2 和 EC2Config 启动代理。EC2Launch v1 是一组 PowerShell 脚本，默认情况下通过计划任务进行管理。本节介绍 EC2Launch v2 和 EC2Config 的服务管理。

要将更新的设置应用到您的实例，您可以从 Microsoft 管理控制台（MMC）服务界面停止并重启 EC2Launch v2 代理或 EC2Config 服务启动代理。同样，安装新版本的启动代理时，必须先停止代理，然后在安装完成后重启代理。

> **Note**
>
> 您必须以管理员身份打开 MMC 服务界面才能选择这些操作。为此，您可以从上下文菜单中选择以管理员身份运行。或者，要使用键盘打开界面，请执行以下步骤：
>
> 1. 使用 Tab 键或箭头键从管理工具菜单中选择服务菜单项。
> 2. 使用以下键盘组合以管理员身份打开：Ctrl + Shift + Enter。

以下过程列出了在实例上停止和启动启动代理的步骤。

停止启动代理

1. 启动并连接到您的 Windows 实例。
2. 从 Windows 的开始菜单选择管理工具。
3. 以管理员身份打开服务控制台，如本节开头所述。
4. 在服务列表中，选择在您实例上运行的代理（EC2Launch 或 EC2Config），然后从操作菜单中选择停止。或者，您可以使用上下文菜单停止代理。

重启启动代理

1. 启动并连接到您的 Windows 实例。
2. 从 Windows 的开始菜单选择管理工具。
3. 以管理员身份打开服务控制台，如本节开头所述。
4. 在服务列表中，选择在您实例上运行的代理（EC2Launch 或 EC2Config），然后从操作菜单中选择启动或重启。或者，您可以使用上下文菜单重启代理。

如果您不需要更新配置设置、创建自己的 AMI 或使用 Amazon Systems Manager，则可以删除并卸载启动代理。

删除

删除服务会删除它的注册表子项。

卸载

卸载服务会删除与此服务相关的文件、注册表子项和所有快捷方式。

删除启动代理

1. 启动并连接到您的 Windows 实例。
2. 启动 Windows 命令提示符窗口。
3. 运行以下命令之一以删除启动代理。

   - 运行以下命令以删除 C2Launch 或 EC2Launch v2：

     ```
     sc delete ec2launch
     ```

   - 运行以下命令以删除 EC2Config 服务：

     ```
     sc delete ec2config
     ```

卸载启动代理

1. 启动并连接到您的 Windows 实例。
2. 选择 Windows 系统，然后从 Windows 的开始菜单中选择控制面板。
3. 选择程序和功能以打开您的实例上安装的程序列表。
4. 从列表中选择您的启动代理（Amazon EC2Launch 或 EC2ConfigService），然后从文件菜单中选择卸载。或者，您可以使用上下文菜单。

   > **Note**
   >
   > 您可以在版本列中查看安装的启动代理版本。

使用 EC2Launch v2 代理在 EC2 Windows 实例启动期间执行任务

默认情况下，从 Amazon Windows Server 2022 和 Windows Server 2025 AMI 启动的所有受支持的 Amazon EC2 实例包含 EC2Launch v2 启动代理 (`EC2Launch.exe`)。我们还提供 Windows Server 2016 和 2019 AMI，其中安装了 EC2Launch v2 作为默认的启动代理。除了包含 EC2Launch v1 的 Windows Server 2016 和 2019 AMI 之外，还提供了这些 AMI。您可以在 Amazon EC2 控制台 AMI 页面的搜索中输入以下前缀来搜索默认包含 EC2Launch v2 的 Windows AMI：`EC2LaunchV2-Windows_Server-*`。

要比较启动代理版本的功能，请参阅 比较 Amazon EC2 启动代理。

EC2Launch v2 在实例启动期间执行任务，并可在实例停止并稍后启动或重新启动时运行。EC2Launch v2 还可按需执行任务。部分任务可自动启用，而其他任务必须手动启用。EC2Launch v2 服务支持所有 EC2Config 和 EC2Launch 功能。

此服务使用配置文件来控制其操作。您可以使用图形工具或直接将其作为单个 .yml 文件 (`agent-config.yml`) 进行编辑，以此来更新配置文件。有关文件位置的更多信息，请参阅 EC2Launch v2 目录结构。

EC2Launch v2 发布 Windows 事件日志，帮助您进行故障排除和设置触发器。有关更多信息，请参阅 Windows 事件日志。

支持的操作系统版本

EC2Launch v2 代理支持以下 Windows Server 操作系统（OS）版本：

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019（长期服务渠道和半年期服务渠道）
- Windows Server 2016

默认运行的任务

EC2Launch v2 代理在初始实例启动期间默认仅运行以下任务一次。任务按照其在启动阶段的运行顺序进行组织。

`Boot` 阶段

- extendRootPartition

`PreReady` 阶段

- activateWindows
- setDnsSuffix
- setAdminAccount
- setWallpaper

`PostReady` 阶段

- startSsm

EC2Launch v2 概念

在考虑使用 EC2Launch v2 时，以下概念很有用。

agent-config

`agent-config` 是位于 EC2Launch v2 的配置文件夹中的文件。它包括用于 boot、network、PreReady 和 PostReady 阶段的配置。此文件用于指定在 AMI 首次引导或以后引导时，应运行任务的实例配置。

默认情况下，EC2Launch v2 安装会安装一个 `agent-config` 文件，其中包含在标准 Amazon Windows AMI 中使用的推荐配置。您可以更新配置文件以更改 EC2Launch v2 为您的 AMI 指定的默认引导体验。有关文件位置的更多信息，请参阅 EC2Launch v2 目录结构。

频率

任务频率决定任务的运行时间，具体取决于引导上下文。大多数任务的频率只有一次。您可以指定 `executeScript` 任务的频率。

您将在 EC2Launch v2 任务配置 中看到以下频率。

- 一次 — 任务在 AMI 首次引导时（完成了 Sysprep）运行一次。
- 始终 — 任务在每次启动代理运行时都会运行。启动代理在以下情况运行：
  - 实例启动或重新启动时
  - EC2Launch 服务运行时
  - `EC2Launch.exe run` 被调用

阶段

阶段是 EC2Launch v2 代理运行的任务的逻辑分组。某些任务只能在特定阶段运行。另一些任务可以分多个阶段运行。使用 `agent-config.yml` 时，必须指定阶段列表以及每个阶段内运行的任务列表。

该服务按以下顺序运行阶段：

第 1 阶段：启动

第 2 阶段：网络

第 3 阶段：PreReady

Windows 准备就绪

PreReady 阶段完成后，该服务会将 `Windows is ready` 消息发送到 Amazon EC2 控制台。

第 4 阶段：PostReady

用户数据在 PostReady 阶段运行。有些脚本版本在 `agent-config.yml` 文件 PostReady 阶段之前运行，有些则在之后运行，如下所示：

在 `agent-config.yml` 之前：

- YAML 用户数据版本 1.1
- XML 用户数据

在 `agent-config.yml` 之后

- YAML 用户数据版本 1.0（用于向后兼容的旧版本）

有关示例阶段和任务，请参阅 示例：agent-config.yml。

使用用户数据时，必须指定要运行的启动代理的任务列表。阶段是默认的。有关示例任务，请参阅 示例：用户数据。

EC2Launch v2 会按照您在 `agent-config.yml` 和用户数据中指定的顺序运行任务列表。阶段按顺序运行。上一阶段完成后，开始下一阶段。任务也按顺序运行。

Task

您可以调用任务来对实例执行操作。您可以在 `agent-config.yml` 文件中或通过用户数据配置任务。有关 EC2Launch v2 的可用任务列表，请参阅 EC2Launch v2 任务。有关任务配置架构和详细信息，请参阅 EC2Launch v2 任务配置。

用户数据

用户数据是在启动实例时可配置的数据。您可以更新用户数据以动态更改如何配置自定义 AMI 或快速启动 AMI。EC2Launch v2 支持 60 kB 用户数据输入长度。用户数据仅包括 UserData 阶段，因此在 `agent-config` 文件之后运行。您可以在使用启动实例向导启动实例时输入用户数据，也可以从 EC2 控制台修改用户数据。有关使用用户的更多信息，请参阅Amazon EC2 如何处理 Windows 实例的用户数据。

EC2Launch v2 任务概览

EC2Launch v2 可以在每次引导时执行以下任务：

- 设置新的（可选）自定义壁纸，以呈现有关实例的信息。
- 为在本地计算机上创建的管理员账户设置属性。
- 将 DNS 后缀添加到搜索后缀列表中。只有尚不存在的后缀才会添加到列表中。
- 为任何附加卷设置驱动器盘符，并扩展它们以使用可用空间。
- 将配置中的文件写入磁盘。
- 运行 EC2Launch v2 配置文件或 `user-data` 中指定的脚本。`user-data` 的脚本可以是纯文本，也可以是压缩文件，并以 base64 格式提供。
- 使用给定参数运行程序。
- 设置计算机名称。
- 将实例信息发送到 Amazon EC2 控制台。
- 向 Amazon EC2 控制台发送 RDP 证书指纹。
- 请动态扩展操作系统分区，以包含所有未分区的空间。
- 运行用户数据。有关指定用户数据的更多信息，请参阅EC2Launch v2 任务配置。
- 设置非持久静态路由以访问元数据服务和 Amazon KMS 服务器。
- 将非引导分区设置为 `mbr` 或 `gpt`。
- 在 Sysprep 之后启动 Systems Manager 服务。
- 优化 ENA 设置。
- 为更高版本的 Windows 启用 OpenSSH。
- 启用巨型帧。
- 将 Sysprep 设置为与 EC2Launch v2 一起运行。
- 发布 Windows 事件日志。

EC2Launch v2 目录结构

EC2Launch v2 应安装在以下目录中：

- 服务二进制文件：`%ProgramFiles%\Amazon\EC2Launch`
- 服务数据（设置、日志文件和状态文件）：`%ProgramData%\Amazon\EC2Launch`

> **Note**
>
> 默认情况下，Windows 会隐藏 C:\ProgramData 下的文件和文件夹。要查看 EC2Launch v2 目录和文件，必须在 Windows 资源管理器中输入路径，或者更改文件夹属性以显示隐藏的文件和文件夹。

%ProgramFiles%\Amazon\EC2Launch 目录包含二进制文件和支持库。它包括以下子目录：

- settings
  - EC2LaunchSettingsUI.exe：用于修改 agent-config.yml 文件的用户界面
  - YamlDotNet.dll：用于支持用户界面中部分操作的 DLL
- tools
  - ebsnvme-id.exe：用于检查实例上 EBS 卷的元数据的工具
  - AWSAcpiSpcrReader.exe：用于确定要使用的正确 COM 端口的工具
  - EC2LaunchEventMessage.dll：用于支持 EC2Launch 的 Windows 事件日志记录的 DLL
- service
  - EC2LaunchService.exe：当启动代理作为一项服务运行时所启动的 Windows 服务可执行文件
- EC2AgentTelemetry.dll：用于支持 EC2 代理遥测的 DLL
- EC2Launch.exe：主要 EC2Launch 可执行文件
- EC2LaunchAgentAttribution.txt：EC2Launch 中使用的代码的归因

%ProgramData%\Amazon\EC2Launch 目录包含以下子目录。服务生成的所有数据（包括日志、配置和状态）都存储在此目录中。

- config – 配置

  服务配置文件作为 agent-config.yml 存储在此目录中。此文件可以进行更新以修改、添加或删除由服务运行的原定设置任务。在此目录中创建文件的权限仅限于管理员账户，以防止特权提升。

- log – 实例日志

  服务 (agent.log) 和遥测 (telemetry.log) 的日志都存储在此目录中。当 agent.log 大小达到 1 MB 时，它会自动轮换，并使用时间戳格式创建备份文件（例如，agent-2026-03-02T18-56-39.188.log）。一次只能维护一个备份日志文件。

- state – 服务状态数据

服务用于确定应运行哪些任务的状态存储在此处。其中有一个 `.run-once` 文件，指示服务是否已在 Sysprep 之后运行（因此，频率为一次的任务在下次运行时将跳过）。此子目录包含 `state.json` 和 `previous-state.json`，用于跟踪每个任务的状态。

- `sysprep` – Sysprep

  此目录包含的文件用于确定 Sysprep 在创建可重复使用的自定义 Windows AMI 时，需要执行哪些操作。

- `wallpaper` – 墙纸

  此墙纸图像存储在此目录中。

遥测

遥测是附加信息，可帮助 Amazon 更好地了解您的需求、诊断问题并提供功能，以改善 Amazon Web Services 服务 带给您的体验。

EC2Launch v2 版本 `2.1.592` 及更高版本会收集遥测，例如使用情况指标和错误。此数据是从运行 EC2Launch v2 的 Amazon EC2 实例收集的。这包括 Amazon 拥有的所有 Windows AMI。

EC2Launch v2 收集以下类型的遥测：

- 使用情况信息 — 代理命令、安装方法和计划的运行频率。
- 错误和诊断信息 — 代理安装错误代码、运行错误代码和错误调用堆栈。

从 2.0.592 到 2.1.1 版本收集的数据示例：

```
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentCommandErrorCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AdminPasswordTypeCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: IpConflictDetectionCode=0
2025/07/18 22:38:52Z: EC2LaunchTelemetry: AgentErrorLocation=addroutes.go:49
```

从 2.2.63 版本开始，EC2 代理遥测数据被格式化为 JSON 对象：

```
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2" ... }
```

预设情况下，遥测处于启用状态。您可以随时禁用遥测收集。

在实例上禁用遥测

要禁用单个实例的遥测，您可以设置系统环境变量，也可以使用 MSI 修改安装。

要通过设置系统环境变量来禁用遥测，请以管理员身份运行以下命令。

```
setx /M EC2LAUNCH_TELEMETRY 0
```

要使用 MSI 禁用遥测，请在下载 MSI 后运行以下命令。

```
msiexec /i ".\AmazonEC2Launch.msi" Remove="Telemetry" /q
```

EC2Launch v2 的更多主题

- 安装最新版本的 EC2Launch v2
- 配置 Windows 实例的 EC2Launch v2 设置
- EC2Launch v2 启动任务的任务定义
- 排查 EC2Launch v2 代理的问题
- EC2Launch v2 版本历史记录

安装最新版本的 EC2Launch v2

您可以使用以下方法之一，在您的 EC2 实例上安装 EC2Launch v2 代理：

- 从 Amazon S3 下载代理并使用 Windows PowerShell 进行安装。有关下载 URL，请参阅 在 Amazon S3 上下载 EC2Launch v2。
- 使用 SSM 分发服务器安装。
- 创建自定义映像时，从 EC2 Image Builder 组件安装。
- 从预安装了 EC2Launch v2 的 AMI 启动您的实例。

> **Warning**
>
> AmazonEC2Launch.msi 不会卸载先前版本的 EC2 启动服务，如 EC2Launch (v1) 和 EC2Config。

有关安装步骤，请选择与您的首选方法相匹配的选项卡。

PowerShell

要使用 Windows PowerShell 安装最新版本的 EC2Launch v2 代理，请按照以下步骤操作。

1. 创建本地目录。

```
New-Item -Path "$env:USERPROFILE\Desktop\EC2Launchv2" -ItemType Directory
```

2. 设置下载位置的 URL。使用您将使用的 Amazon S3 URL 运行以下命令。有关下载 URL，请参阅 在 Amazon S3 上下载 EC2Launch v2

```
$Url = "Amazon S3 URL/AmazonEC2Launch.msi"
```

3. 使用以下混合命令下载代理并运行安装

```
$DownloadFile = "$env:USERPROFILE\Desktop\EC2Launchv2\" + $(Split-Path -Path
 $Url -Leaf)
Invoke-WebRequest -Uri $Url -OutFile $DownloadFile
msiexec /i "$DownloadFile"
```

> **Note**
>
> 如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

4. msiexec 命令在 Windows Server 实例上的以下位置安装 EC2Launch v2：`%ProgramFiles%\Amazon\EC2Launch`。要验证安装是否已运行，可以检查实例上的本地文件系统。

Amazon Systems Manager Distributor

要使用 Amazon Systems Manager 快速设置功能为 EC2Launch v2 配置自动更新，请参阅 使用 Distributor 快速设置功能自动安装和更新 EC2Launch v2。

也可以从 Amazon Systems Manager Distributor 处一次性安装 `AWSEC2Launch-Agent` 软件包。有关如何从系统管理器分发服务器安装软件包的说明，请参阅《Amazon Systems Manager 用户指南》中的安装或更新软件包。

EC2 Image Builder component

您可以在使用 EC2 Image Builder 构建自定义映像时安装 `ec2launch-v2-windows` 组件。有关如何使用 EC2 Image Builder 构建自定义映像的说明，请参阅 EC2 Image Builder 用户指南中的使用 EC2 Image Builder 控制台创建映像管道向导。

AMI

EC2Launch v2 默认预安装在 Windows Server 2022 及更高版本操作系统的 AMI 上：

- Windows_Server-##-English-Full-Base
- Windows_Server-##-English-Core-Base
- Windows_Server-##-English-Core-EKS_Optimized
- 具有所有其他语言的 Windows Server ##
- 安装了 SQL 的 Windows Server ##

以下 Windows Server AMI 上也预装了 EC2Launch v2。您可以在 Amazon EC2 控制台上查找这些 AMI，也可以通过使用以下搜索前缀查找：`EC2LaunchV2-` 在 Amazon CLI 中

- EC2LaunchV2-Windows_Server-2019-English-Core-Base
- EC2LaunchV2-Windows_Server-2019-English-Full-Base
- EC2LaunchV2-Windows_Server-2016-English-Core-Base
- EC2LaunchV2-Windows_Server-2016-English-Full-Base

使用 Amazon Systems Manager Distributor 快速设置功能自动安装和更新 EC2Launch v2

使用 Amazon Systems Manager Distributor 快速设置功能，可以为 EC2Launch v2 设置自动更新。以下过程可在您的实例上设置 Systems Manager 关联，该关联以您指定的频率自动更新 EC2Launch v2 代理。Distributor 快速设置功能创建的关联可包括 Amazon Web Services 账户 和区域内的实例，也可以包括 Amazon 组织内的实例。有关设置组织的更多信息，请参阅《Amazon Organizations 用户指南》中的教程：创建和配置组织。

开始之前，请确保实例满足所有先决条件。

先决条件

要使用 Distributor 快速设置功能设置自动更新，实例必须满足以下先决条件。

- 至少有一个支持 EC2Launch v2 的正在运行的实例。请参阅 EC2Launch v2 支持的操作系统。
- 已在实例上执行了 Systems Manager 设置任务。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的设置 Systems Manager。
- EC2Launch v2 必须是实例上安装的唯一启动代理。如果安装了多个启动代理，则 Distributor 快速设置功能配置将失败。在使用 Distributor 快速设置功能配置 EC2Launch v2 之前，请卸载 EC2Config 或 EC2Launch v1 启动代理（如果存在）。

为 EC2Launch v2 配置 Distributor 快速设置功能

要使用 Distributor 快速设置功能为 EC2Launch v2 创建配置，请在完成 Distributor 包部署步骤时使用以下设置：

- 软件包：Amazon EC2Launch v2 代理。
- 更新频率：从列表中选择一个频率。
- 目标：从可用的部署选项中进行选择。

要检查配置的状态，请导航至 Amazon Web Services 管理控制台 中的 Systems Manager 快速设置功能配置选项卡。

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 在导航窗格中，选择快速设置。
3. 在配置选项卡中，选择与您创建的配置关联的行。配置选项卡列出了您的配置，并包括关键详细信息的摘要，例如区域、部署状态和关联状态。

   > **Note**
   >
   > 每个 EC2Launch v2 Distributor 配置的关联名称都以以下前缀开头：`AWS-QuickSetup-Distributor-EC2Launch-Agent-`。

4. 要查看详细信息，请选择配置，然后选择查看详细信息。

有关更多信息和故障排除步骤，请参阅《Amazon Systems Manager 用户指南》中的对快速设置结果进行故障排除。

在 Amazon S3 上下载 EC2Launch v2

要安装最新版本的 EC2Launch v2，请从以下位置下载安装程序：

- https://s3.cn-north-1.amazonaws.com.cn/amazon-ec2launch-v2/windows/amd64/latest/AmazonEC2Launch.msi

配置安装选项

安装或升级 EC2Launch v2 时，可以使用 EC2Launch v2 安装对话框或命令行 shell 中的 msiexec 命令配置安装选项。

EC2Launch v2 安装程序首次在实例上运行时，它会按照以下方式初始化实例上的启动代理设置：

- 它会创建本地路径并将启动代理文件写入其中。这有时被称为净安装。
- 它创建了 EC2LAUNCH_TELEMETRY 环境变量（如果还不存在），然后根据您的配置进行设置。

有关配置详细信息，请选择与您将使用的配置方法相匹配的选项卡。

Amazon EC2Launch Setup dialog

安装或升级 EC2Launch v2 时，可以使用 EC2Launch v2 安装对话框配置以下安装选项。

基本安装选项

发送遥测

当您在设置对话框中包含此功能时，安装程序会将 EC2LAUNCH_TELEMETRY 环境变量的值设置为 1。如果您禁用发送遥测，安装程序会将环境变量的值设置为 0。

当 EC2Launch v2 代理运行时，它会读取 EC2LAUNCH_TELEMETRY 环境变量，用于确定是否上传遥测数据。如果值等于 1，它会上传数据。否则，它不会上传。

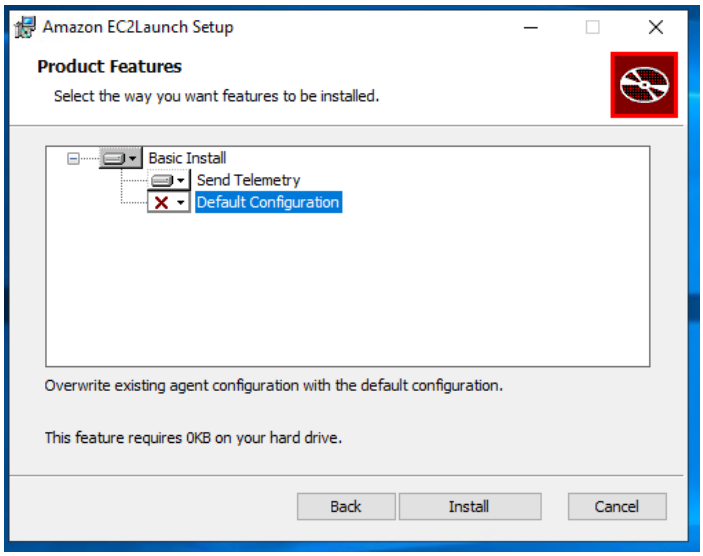
默认配置

EC2Launch v2 的默认配置为：如果本地启动代理已经存在，则将其覆盖。首次在实例上运行安装时，默认配置会执行净安装。如果您在初次安装时禁用默认配置，则安装将失败。

如果您在实例上再次运行安装程序，则可以禁用默认配置，以执行不取代 %ProgramData%/Amazon/EC2Launch/config/agent-config.yml 文件的升级。

示例：使用遥测升级 EC2Launch v2

以下示例显示了配置为升级当前安装并启用遥测的 EC2Launch v2 设置对话框。此配置在不替换代理配置文件的情况下执行安装程序，并将 EC2LAUNCH_TELEMETRY 环境变量设置为值 1。



Command line

安装或升级 EC2Launch v2 时，可以使用命令行 Shell 中的 msiexec 命令配置以下安装选项。

ADDLOCAL参数值

基本（必需）

安装启动代理。如果此值不存在于 ADDLOCAL 参数，安装将结束。

清除

当您将 Clean 值包含在 ADDLOCAL 参数中，安装程序会将代理配置文件写入以下位置：%ProgramData%/Amazon/EC2Launch/config/agent-config.yml。如果代理配置文件已经存在，它会覆盖该文件。

当您留下 ADDLOCAL 参数中的 Clean 值，安装程序执行的升级不会替换代理配置文件。

遥测

当您在 ADDLOCAL 中包含 Telemetry 值，安装程序会将 EC2LAUNCH_TELEMETRY 环境变量的值设置为 1。

当您留下 ADDLOCAL 参数中的 Telemetry 值，安装程序会将环境变量的值设置为 0。

当 EC2Launch v2 代理运行时，它会读取 EC2LAUNCH_TELEMETRY 环境变量，用于确定是否上传遥测数据。如果值等于 1，它会上传数据。否则，它不会上传。

示例：安装带有遥测功能的 EC2Launch v2

```
& msiexec /i "C:\Users\Administrator\Desktop\EC2Launchv2\AmazonEC2Launch.msi"
 ADDLOCAL="Basic,Clean,Telemetry" /q
```

验证 EC2Launch v2 版本

使用以下过程之一验证您的实例上安装的 EC2Launch v2 版本。

PowerShell

使用 Windows PowerShell 验证已安装的 EC2Launch v2 版本，如下所示。

1. 从 AMI 启动实例并连接到该实例。
2. 使用以下 PowerShell 中的命令验证已安装的 EC2Launch v2 版本：

```
& "C:\Program Files\Amazon\EC2Launch\EC2Launch.exe" version
```

Windows Control Panel

验证 Windows 控制面板中安装的 EC2Launch v2 版本，如下所示。

1. 从 AMI 启动实例并连接到该实例。
2. 打开 Windows 控制面板，然后选择程序和功能。
3. 在已安装程序的列表中查找 Amazon EC2Launch。其版本号会显示在 Version (版本) 列中。

要查看 Amazon Windows AMI 的最近更新，请参阅《Amazon Windows AMI 参考》中的 Windows AMI version history。

有关 EC2Launch v2 的最新版本，请参阅 EC2Launch v2 版本历史记录。

您可在 EC2Launch v2 服务的新版本发布时收到通知。有关更多信息，请参阅 订阅 EC2 Windows 启动代理通知。

配置 Windows 实例的 EC2Launch v2 设置

本节包含有关如何配置 EC2Launch v2 的设置的信息。

主题包含：

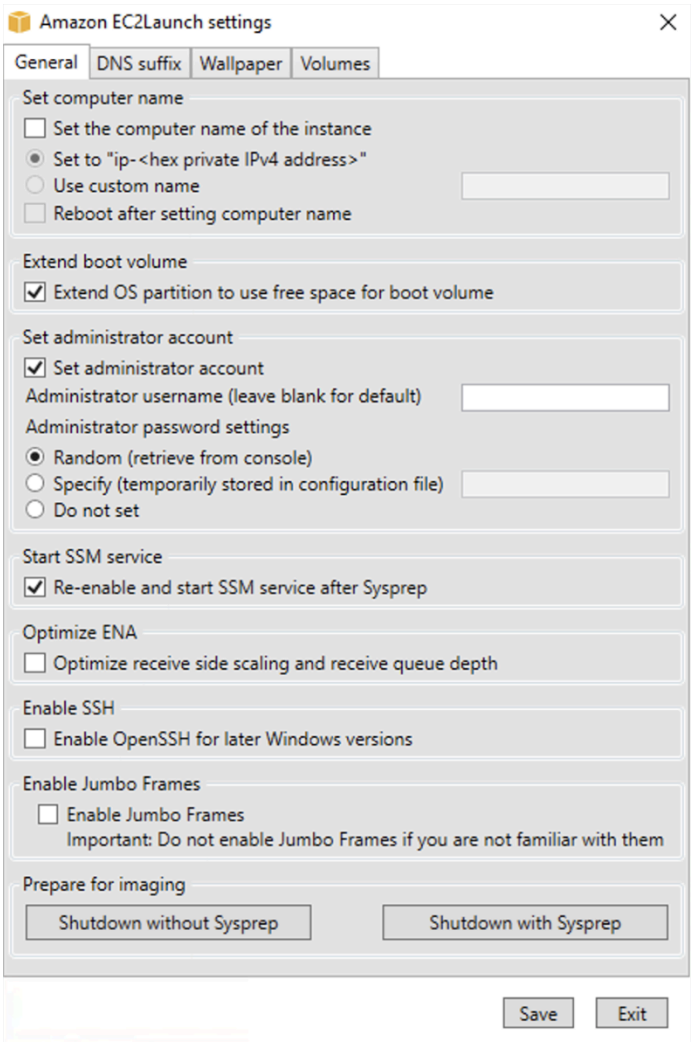
- 使用 EC2Launch v2 设置对话框更改设置
- 使用 CLI 配置 EC2Launch v2
- EC2Launch v2 任务配置
- EC2Launch v2 退出代码并重新启动
- EC2Launch v2 和 Sysprep

使用 EC2Launch v2 设置对话框更改设置

以下过程介绍如何使用 EC2Launch v2 设置对话框来启用或禁用设置。

> **Note**
>
> 如果您在 agent-config.yml 文件中适当配置了自定义任务，并尝试打开 Amazon EC2Launch settings（Amazon EC2Launch 设置）对话框，则会收到错误消息。有关示例架构，请参阅 示例：`agent-config.yml`。

1. 启动并连接到您的 Windows 实例。
2. 从“开始”菜单，选择所有程序，然后导航到 EC2Launch 设置。在选择使用 Sysprep 关机或不使用 Sysprep 关机之前，请务必保存要在运行关机时应用的所有更改。



3. 在 EC2Launch 设置 对话框的常规选项卡上，您可以启用或禁用以下设置。

   a. Set Computer Name（设置电脑名称）

      如果启用此设置（默认情况下禁用），则每次引导时会将当前主机名与所需主机名进行比较。如果主机名不匹配，则会重置主机名，然后系统可以选择重新启动以接受新的主机名。如果未指定自定义主机名，则该主机名会以十六进制格式的私有 IPv4 地址生成，例如：`ip-AC1F4E6`。若要防止修改现有主机名，请不要启用该设置。

   b. 扩展引导卷

      此设置可动态扩展 `Disk 0/Volume 0` 以包含所有未分区的空间。从具有自定义大小的根卷启动实例时，此设置非常有用。

   c. 设置管理员帐户

启用后，您可以为在本地计算机上创建的管理员账户设置用户名和密码属性。如果未启用此功能，则在 Sysprep 之后，不会在系统上创建管理员账户。仅当 `adminPassword` 为 `adminPasswordtype` 时，在 `Specify` 中提供密码。

密码类型定义如下：

i. `Random`

EC2Launch 生成一个密码并使用用户的密钥对其进行加密。系统会在实例启动后禁用此设置，以便在重新启动或停止再启动实例后该密码仍然存在。

ii. `Specify`

EC2Launch 使用您在 `adminPassword` 中指定的密码 如果密码不满足系统要求，EC2Launch 会生成随机密码。该密码以明文方式存储在 `agent-config.yml` 文件中，并且在 Sysprep 设置管理员密码时会被删除。EC2Launch 使用用户的密钥对密码进行加密。

iii. `Do not set`

EC2Launch 使用您在 unattend.xml 文件中指定的密码。如果未在 unattend.xml 中指定密码，管理员账户会被禁用。

d. 启动 SSM 服务

如果选择此选项，Systems Manager 服务将启用，以便在 Sysprep 之后启动。EC2Launch v2 执行前面所述的所有任务，而 SSM Agent 处理 Run Command 和 State Manager 这样的 Systems Manager 功能的请求。

您可以使用 Run Command 升级现有实例，以便使用最新版本的 EC2Launch v2 服务和 SSM Agent。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的使用 Run Command 更新 SSM Agent。

e. 优化 ENA

在选中时，将配置 ENA 设置以确保针对 Amazon 优化了 ENA 接收端扩展和接收队列深度设置。有关更多信息，请参阅配置接收端缩放 CPU 关联。

f. 启用 SSH

此设置为更高版本的 Windows 启用 OpenSSH，以允许远程系统管理。

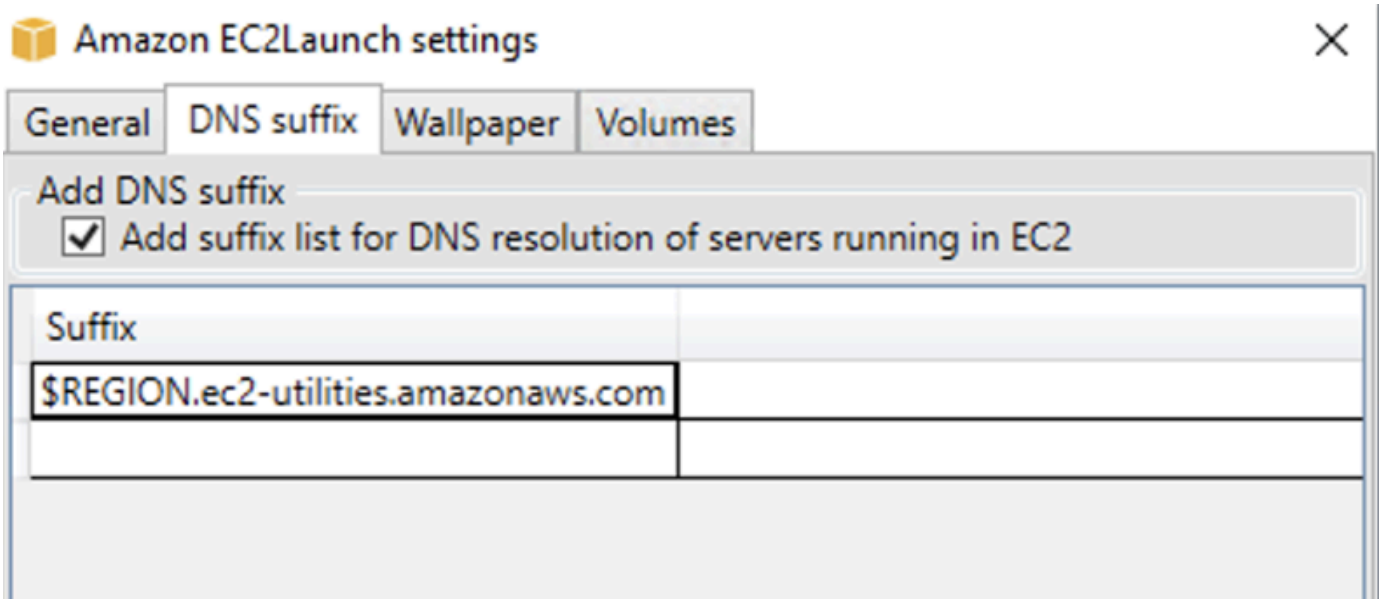
g. 启用巨型帧

选择此项可启用巨型帧。巨型帧可能会对您的网络通信造成意外的影响，因此在启用之前，请确保您了解巨型帧将如何影响系统。有关巨型帧的详细信息，请参阅巨型帧 (9001 MTU)。

h. 准备映像

选择您希望在使用或不使用 Sysprep 的情况下关闭 EC2 实例。如果您希望将 Sysprep 与 EC2Launch v2 配合使用，请选择 Shutdown with Sysprep (使用 Sysprep 关闭)。

4. 在 DNS 后缀选项卡上，您可以选择是否要添加 DNS 后缀列表，供在 EC2 中运行的服务器的 DNS 解析使用，而不提供完全限定的域名。DNS 后缀可以包含变量 `$REGION` 和 `$AZ`。只有尚不存在的后缀才会添加到列表中。



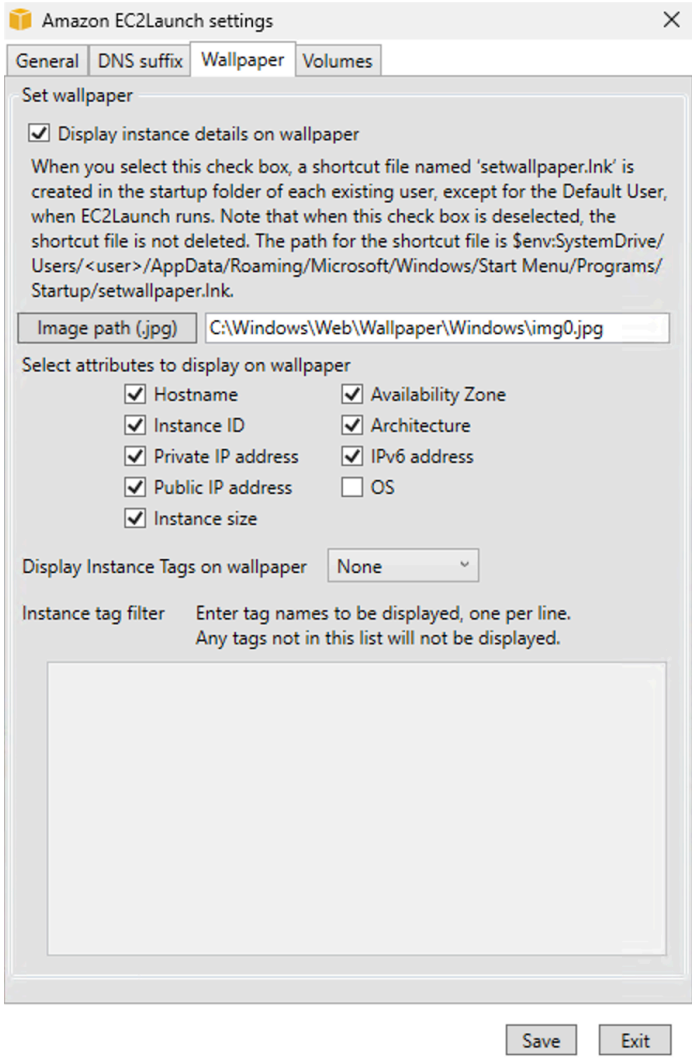
5. 在墙纸选项卡上，您可以使用背景图像配置实例墙纸，并为要显示的墙纸指定实例详细信息。Amazon EC2 会在您每次登录时生成详细信息。

您可以使用以下控件配置墙纸。

- 在墙纸上显示实例详细信息 – 此复选框会启用或禁用在墙纸上显示实例详细信息。
- 图像路径 (.jpg) – 指定用作墙纸背景的图像路径。
- 选择要在墙纸上显示的属性 – 选中要在墙纸上显示的实例详细信息的复选框。清除之前选中的要从墙纸中删除的实例详细信息的复选框。
- 在墙纸上显示实例标签 – 选择以下设置之一，以在墙纸上显示实例标签：
  - 无 – 不在墙纸上显示任何实例标签。
  - 显示全部 – 在墙纸上显示所有实例标签。

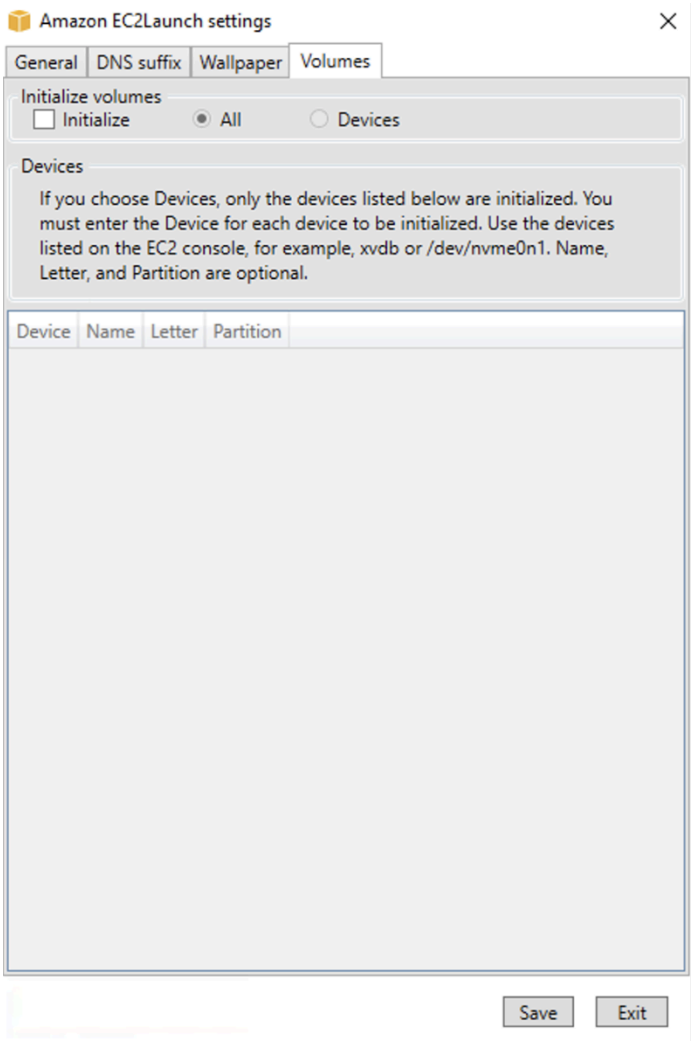
- 显示筛选部分 – 在墙纸上显示指定的实例标签。当您选择此设置时，您可以在实例标签筛选条件框中添加要在墙纸上显示的实例标签。

> **Note**
>
> 必须启用元数据中的标签，才能在墙纸上显示标签。有关实例标签和元数据的更多信息，请参阅 使用实例元数据来查看 EC2 实例的标签。



6. 在卷选项卡上，选择是否要初始化附加到实例的卷。启用时，可为任何附加卷设置驱动器盘符，并扩展它们以使用可用空间。如果选择全部，则初始化所有存储卷。如果选择设备，则只初始化列表中指定的设备。您必须为要初始化的每个设备进行输入。使用 EC2 控制台上列出的设备，例如，xvdb 或 /dev/nvme0n1。下拉列表显示已附加到实例的存储卷。要输入未附加到实例的设备，请在文本字段中输入该设备。

名称、盘符和分区是可选字段。如果没有为分区指定值，则使用 gpt 分区类型初始化大于 2 TB 的存储卷，使用 mbr 分区类型初始化小于 2 TB 的存储卷。如果配置了设备，并且非 NTFS 设备包含分区表，或者磁盘的前 4 KB 包含数据，则跳过磁盘并记录操作。



使用 CLI 配置 EC2Launch v2

您可以使用命令行界面 (CLI) 配置 EC2Launch 设置和管理服务。以下部分包含您可用于管理 EC2Launch v2 的 CLI 命令的描述和使用信息。

命令

- collect-logs
- get-agent-config
- list-volumes

- 重置
- 运行
- status
- sysprep
- 验证
- 版本
- wallpaper

collect-logs

收集 EC2Launch 的日志文件，压缩文件，并将其放置在指定的目录中。

示例

```
ec2launch collect-logs -o C:\Mylogs.zip
```

用法

`ec2launch collect-logs [flags]`

Flags

`-h, --help`

`collect-logs` 的帮助

`-o, --output string`

压缩输出日志文件的路径

get-agent-config

以指定的格式（JSON 或 YAML）输出 `agent-config.yml`。如果未指定格式，则会以先前指定的格式输出 `agent-config.yml`。

示例

```
ec2launch get-agent-config -f json
```

用法

`ec2launch get-agent-config [flags]`

Flags

`-h, --help`

`get-agent-config` 的帮助

`-f, --format string`

`agent-config` 文件的输出格式：`json`、`yaml`

list-volumes

列出附加到实例的所有存储卷，包括临时卷和 EBS 卷。

示例

```
ec2launch list-volumes
```

用法

`ec2launch list-volumes`

Flags

`-h, --help`

`list-volumes` 的帮助

重置

此任务的主要目标是在代理下次运行时对其进行重置 为此，reset 命令从本地 `EC2Launch` 目录中删除 EC2Launch v2 的所有代理状态数据（请参阅 EC2Launch v2 目录结构）。重置可选择性地删除服务和 Sysprep 日志。

脚本行为取决于代理在哪种模式下运行脚本：内联或分离。

内联（默认）

EC2Launch v2 代理一次只能运行一个脚本（`detach: false`）。这是默认设置。

> **Note**
>
> 当您的内联脚本发出 reset 或 sysprep 命令时，它会立即运行并重置代理。当前任务完成，然后代理在不运行任何其他任务的情况下关闭。
> 例如，如果发出命令的任务之后会有 `startSsm` 任务（默认情况下在用户数据运行后包括在内），则该任务不会运行，Systems Manager 服务也不会启动。

已分离

EC2Launch v2 代理与其他任务同时运行脚本（`detach: true`）。

> **Note**
>
> 当分离的脚本发出 reset 或 sysprep 命令时，这些命令会等待代理完成后再运行。executeScript 之后的任务仍将运行。

示例

```
ec2launch reset -c
```

用法

```
ec2launch reset [flags]
```

Flags

`-c, --clean`

在`reset`前清除实例日志

`-h, --help`

`reset` 的帮助

运行

运行 EC2Launch v2。

示例

```
ec2launch run
```

用法

```
ec2launch run [flags]
```

Flags

`-h, --help`

`run` 的帮助

status

获取 EC2Launch v2 代理的状态。也可选择在代理完成运行之前阻止该进程。进程退出代码决定了代理状态：

- `0` – 代理已运行并且成功。
- `1` – 代理已运行但失败。
- `2` – 代理仍在运行中。
- `3` – 代理处于未知状态。代理状态为未运行或已停止。
- `4` – 尝试检索代理状态时出现错误。
- `5` – 代理未运行，并且上次已知运行的状态未知。这可能意味着以下之一：
  - `state.json` 和 `previous-state.json` 都将被删除。
  - `previous-state.json` 已损坏。

  这是运行 `reset` 命令后的代理状态。

示例：

```
ec2launch status -b
```

用法

```
ec2launch status [flags]
```

Flags

`-b,--block`

在代理完成运行前阻止才进程

`-h,--help`

`status` 的帮助

sysprep

此任务的主要目标是在代理下次运行时对其进行重置 为此，sysprep 命令会重置代理状态、更新 `unattend.xml` 文件、禁用 RDP 并运行 Sysprep。

脚本行为取决于代理在哪种模式下运行脚本：内联或分离。

内联（默认）

EC2Launch v2 代理一次只能运行一个脚本（`detach: false`）。这是默认设置。

> **Note**
>
> 当您的内联脚本发出 reset 或 sysprep 命令时，它会立即运行并重置代理。当前任务完成，然后代理在不运行任何其他任务的情况下关闭。
> 例如，如果发出命令的任务之后会有 `startSsm` 任务（默认情况下在用户数据运行后包括在内），则该任务不会运行，Systems Manager 服务也不会启动。

已分离

EC2Launch v2 代理与其他任务同时运行脚本（`detach: true`）。

> **Note**
>
> 当分离的脚本发出 reset 或 sysprep 命令时，这些命令会等待代理完成后再运行。executeScript 之后的任务仍将运行。

示例：

```
ec2launch sysprep
```

用法

`ec2launch sysprep [flags]`

Flags

-c,--clean

在sysprep前清除实例日志

-h,--help

Sysprep 的帮助

-s,--shutdown

sysprep之后关闭实例

验证

验证 agent-config 文件 C:\ProgramData\Amazon\EC2Launch\config\agent-config.yml。

示例

```
ec2launch validate
```

用法

ec2launch validate [flags]

Flags

-h ，--help

validate 的帮助

版本

获取可执行版本。

示例

```
ec2launch version
```

用法

ec2launch version [flags]

Flags

`-h, --help`

`version` 的帮助

wallpaper

将新壁纸设置为提供的壁纸路径（.jpg 文件），并显示选定的实例详细信息。

语法

```
ec2launch wallpaper ^
--path="C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg" ^
--all-tags ^
--
attributes=hostName,instanceId,privateIpAddress,publicIpAddress,ipv6Address,instanceSize,availa
```

输入

参数

--allowed-tags [*tag-name-1*、*tag-name-n*]

（可选）要显示在墙纸上的实例标签名称的 JSON 数组（采用 Base64 编码）。您可以使用此标签或 `--all-tags`，但不能同时使用两者。

--attributes *attribute-string-1*、*attribute-string-n*

（可选）以逗号分隔的 `wallpaper` 属性字符串列表，用于将设置应用于墙纸。

[--path | -p] *path-string*

（必需）指定 `wallpaper` 背景图像文件路径。

Flags

--all-tags

（可选）显示墙纸上的所有实例标签。您可以使用此标签或 `--allowed-tags`，但不能同时使用两者。

[--help | -h]

显示 wallpaper 命令的帮助信息。

EC2Launch v2 任务配置

本节包括 agent-config.yml 和用户数据的配置架构、任务、详细信息和示例。

任务和示例

- 架构：agent-config.yml
- 配置在启动或重启期间运行的 EC2Launch v2 用户数据脚本

架构：**agent-config.yml**

agent-config.yml 文件的结构如下所示。请注意，不能在同一阶段重复一个任务。有关任务属性，请参阅随后的任务说明。

文档结构：agent-config.yml

JSON

```
{
 "version": "1.1",
 "config": [
  {
   "stage": "string",
   "tasks": [
    {
     "task": "string",
     "inputs": {
      ...
     }
    },
    ...
   ]
  },
  ...
 ]
}
```

YAML

```
version: 1.1
config:
- stage: string
  tasks:
```

```
  - task: string
 inputs:
   ...
  ...
...
```

示例：**agent-config.yml**

以下示例显示了 agent-config.yml 配置文件的设置。

```
version: 1.1
config:
- stage: boot
  tasks:
  - task: extendRootPartition
- stage: preReady
  tasks:
  - task: activateWindows
    inputs:
      activation:
        type: amazon
  - task: setDnsSuffix
    inputs:
      suffixes:
      - $REGION.ec2-utilities.amazonaws.com
  - task: setAdminAccount
    inputs:
      password:
        type: random
  - task: setWallpaper
    inputs:
      path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
      attributes:
      - hostName
      - instanceId
      - privateIpAddress
      - publicIpAddress
      - instanceSize
      - availabilityZone
      - architecture
- stage: postReady
  tasks:
  - task: startSsm
```

配置在启动或重启期间运行的 EC2Launch v2 用户数据脚本

以下 JSON 和 YAML 示例显示了用户数据的文档结构。Amazon EC2 解析您在文档中指定的 `tasks` 数组中命名的每个任务。每个任务都有自己的一组属性和要求。有关详细信息，请参阅 EC2Launch v2 启动任务的任务定义。

> **Note**
>
> 任务只能在用户数据任务数组中出现一次。

文档结构：用户数据

JSON

```
{
 "version": "1.1",
 "tasks": [
  {
   "task": "string",
   "inputs": {
    ...
   },
  },
  ...
 ]
}
```

YAML

```
version: 1.1
tasks:
- task: string
  inputs:
    ...
...
```

示例：用户数据

有关用户数据的更多信息，请参阅 Amazon EC2 如何处理 Windows 实例的用户数据。

以下 YAML 文档示例显示了 EC2Launch v2 作为用户数据运行的 PowerShell 脚本，用于创建文件。

```
version: 1.1
tasks:
- task: executeScript
  inputs:
  - frequency: always
    type: powershell
    runAs: localSystem
    content: |-
      New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

您可以对用户数据使用与先前版本的启动代理兼容的 XML 格式。EC2Launch v2 在 `UserData` 阶段将脚本作为 `executeScript` 任务运行。为了符合 EC2Launch v1 和 EC2Config 的行为，默认情况下，用户数据脚本以附加/内联进程运行。

您可以添加可选标签来自定义脚本的运行方式。例如，要在实例重新启动时运行用户数据脚本以及在实例启动时运行一次用户数据脚本，您可以使用以下标签：

`<persist>true</persist>`

示例：

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

您可以使用 `<powershellArguments>` 标签来指定一个或多个 PowerShell 参数。如果未传递任何参数，则默认情况下，EC2Launch v2 会添加以下参数：`-ExecutionPolicy Unrestricted`。

示例：

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<powershellArguments>-ExecutionPolicy Unrestricted -NoProfile -NonInteractive</
powershellArguments>
```

要将 XML 用户数据脚本作为分离的进程运行，请在您的用户数据中添加以下标签。

```
<detach>true</detach>
```

示例：

```
<powershell>
  $file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
  New-Item $file -ItemType file
</powershell>
<detach>true</detach>
```

> **Note**
>
> 以前的启动代理不支持分离标签。

更改日志：用户数据

下表列出了用户数据更改，并将其交叉引用至适用的 EC2Launch v2 代理版本。

用户数据版本	Details	已引入
1.1	• 用户数据任务在代理配置文件中的 `PostReady` 阶段之前运行。 • 在启动 Systems Manager Agent 之前运行用户数据（与 EC2Launch v1 和 EC2Config 的行为相同）。*	EC2Launch v2 版本 2.0.1245
1.0	• 将弃用。 • 用户数据任务在代理配置文件中的 `PostReady` 阶段之后运行。这不与 EC2Launch v1 向后兼容。 • 受 Systems Manager Agent 启动和用户数据任务之间争用情况的影响。	EC2Launch v2 版本 2.0.0

* 与默认 `agent-config.yml` 文件结合使用时。

EC2Launch v2 退出代码并重新启动

您可以使用 EC2Launch v2 定义脚本如何处理退出代码。默认情况下，脚本中运行的最后一个命令的退出代码将报告为整个脚本的退出代码。例如，如果脚本包含三个命令，而第一个命令失败，但以下命令成功，则运行状态将报告为 `success`，因为最后一个命令成功。

如果您希望脚本重启实例，则必须在脚本中指定 `exit 3010`，即使重启是脚本中的最后一步，也必须在脚本中指定。`exit 3010` 指示 EC2Launch v2 重新启动实例并再次调用脚本，直到它返回不是 `3010` 的退出代码，或者直到达到最大重启计数为止。EC2Launch v2 允许每个任务最多重启 5 次。如果您尝试使用其他机制（例如）从脚本重启实例 `Restart-Computer`，则脚本运行状态将不一致。例如，它可能会陷入重启循环或不执行重启。

如果您使用的是与旧代理兼容的 XML 用户数据格式，则用户数据的运行时间可能会超过您预期的运行时间。有关更多信息，请参阅排查部分中的服务多次运行用户数据。

EC2Launch v2 和 Sysprep

EC2Launch v2 服务会运行 Sysprep，借助该 Microsoft 工具您可以创建可重复使用的自定义 Windows AMI。在 EC2Launch v2 调用 Sysprep 时，会使用 `%ProgramData%\Amazon\EC2Launch` 中的设置文件来确定要执行的操作。您可以使用 EC2Launch 设置对话框间接编辑这些文件，也可以直接使用 YAML 编辑器或文本编辑器直接编辑。然而，有些高级设置并未包含在 EC2Launch 设置 对话框中，因此您必须直接编辑这些条目。

如果您在更新某实例的设置之后从中创建了一个 AMI，则新设置会应用到所有从新 AMI 启动的实例。有关创建 AMI 的信息，请参阅创建 Amazon EBS-backed AMI。

EC2Launch v2 启动任务的任务定义

EC2Launch v2 在启动期间运行的每个任务都有自己的一组属性和要求。任务详细信息包括任务运行频率的设置（一次或始终）、运行代理启动过程的哪个阶段、语法和 YAML 文档示例。有关更多信息，请查看本参考中显示的任务详细信息。

EC2Launch v2 任务

- activateWindows
- enableJumboFrames
- enableOpenSsh
- executeProgram
- executeScript
- extendRootPartition

- initializeVolume
- optimizeEna
- setAdminAccount
- setDnsSuffix
- setHostName
- setWallpaper
- startSsm
- sysprep
- writeFile

activateWindows

针对一组 Amazon KMS 服务器激活 Windows。如果实例被检测为自带许可（BYOL），则会跳过激活过程。

Frequency – 一次

AllowedStages – `[PreReady]`

输入 –

`activation`：（映射）

`type`：（字符串）使用的激活类型，设置为 `amazon`

示例

```
task: activateWindows
  inputs:
    activation:
    type: amazon
```

enableJumboFrames

启用巨型帧，这会增加网络适配器的最大传输单位 (MTU)。有关更多信息，请参阅巨型帧 (9001 MTU)。

Frequency – 始终

AllowedStages – `[PostReady, UserData]`

Inputs – 无

示例

```
task: enableJumboFrames
```

enableOpenSsh

启用 Windows OpenSSH 并将实例的公有密钥添加到授权密钥文件夹中。

Frequency – 一次

AllowedStages – `[PreReady, UserData]`

Inputs – 无

示例

以下示例说明如何在实例上启用 OpenSSH，以及如何将实例的公有密钥添加到授权的密钥文件夹中。此配置仅适用于运行 Windows Server 2019 及更高版本的实例。

```
task: enableOpenSsh
```

executeProgram

使用可选参数并以指定的频率运行程序。

阶段：您可以在 `PreReady`、`PostReady` 和 `UserData` 阶段运行 `executeProgram` 任务。

频率：可配置，参见输入。

输入

本节包含一个或多个供 executeProgram 任务运行的程序（输入）。每个输入可能包括以下可配置设置：

频率（字符串）

（必填）确切指定以下默认值中的一项：

- `once`
- `always`

路径（字符串）

（必填）要运行的可执行文件的路径。

参数（字符串列表）

（可选）以逗号分隔的参数列表，作为输入提供给程序。

运行为（字符串）

（必填）必须设置为 localSystem

Output

所有任务都会将日志文件条目写入 agent.log 文件。executeProgram 任务的其他输出会单独存储在一个动态命名的文件夹中，如下所示：

%LocalAppData%\Temp*EC2Launch#########**outputfilename.tmp*

输出文件的确切路径包含在 agent.log 文件中，例如：

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\ExecuteProgramInputs.tmp
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

executeProgram 任务的输出文件

ExecuteProgramInputs.tmp

包含可执行文件的路径以及 executeProgram 任务将在运行时传递给该可执行文件的所有输入参数。

Output.tmp

包含 executeProgram 任务运行的程序的运行时系统输出。

Err.tmp

包含 executeProgram 任务运行的程序的运行时系统错误消息。

示例

以下示例演示了如何使用 executeProgram 任务从实例的本地目录运行可执行文件。

示例 1：带有一个参数的安装程序可执行文件

此示例演示了一个在静默模式下运行安装程序可执行文件的 executeProgram 任务。

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\Users\Administrator\Desktop\setup.exe
      arguments: ['-quiet']
```

示例 2：带有两个参数的 VLC 可执行文件

此示例演示了一个运行具有两个参数的 VLC 可执行文件的 executeProgram 任务，该任务将这两个参数作为输入参数传递。

```
task: executeProgram
  inputs:
    - frequency: always
      path: C:\vlc-3.0.11-win64.exe
      arguments: ['/L=1033','/S']
      runAs: localSystem
```

executeScript

使用可选参数和指定的频率运行脚本。脚本行为取决于代理在哪种模式下运行脚本：内联或分离。

内联（默认）

EC2Launch v2 代理一次只能运行一个脚本（detach: false）。这是默认设置。

> **Note**
>
> 当您的内联脚本发出 reset 或 sysprep 命令时，它会立即运行并重置代理。当前任务完成，然后代理在不运行任何其他任务的情况下关闭。
> 例如，如果发出命令的任务之后会有 startSsm 任务（默认情况下在用户数据运行后包括在内），则该任务不会运行，Systems Manager 服务也不会启动。

已分离

EC2Launch v2 代理与其他任务同时运行脚本（detach: true）。

> **Note**
>
> 当分离的脚本发出 reset 或 sysprep 命令时，这些命令会等待代理完成后再运行。executeScript 之后的任务仍将运行。

阶段：您可以在 `PreReady`、`PostReady` 和 `UserData` 阶段运行 `executeScript` 任务。

频率：可配置，参见输入。

输入

本节包含一个或多个供 executeScript 任务运行的脚本（输入）。每个输入可能包括以下可配置设置：

频率（字符串）

（必填）确切指定以下默认值中的一项：

- `once`
- `always`

类型（字符串）

（必填）确切指定以下默认值中的一项：

- `batch`
- `powershell`

参数（字符串列表）

（可选）要传递给 Shell（但不是 PowerShell 脚本）的字符串参数的列表。`type: batch` 不支持此参数。如果未传递任何参数，则默认情况下，EC2Launch v2 会添加以下参数：`-ExecutionPolicy Unrestricted`。

内容（字符串）

（必填）脚本内容。

运行为（字符串）

（必填）确切指定以下默认值中的一项：

- `admin`
- `localSystem`

分离（布尔值）

（可选）EC2Launch v2 代理默认每次运行一个脚本（detach: false）。要与其他任务同时运行脚本，请将此值设置为 true (detach: true)。

> **Note**
>
> 如果将 detach 设置为 true，则脚本退出代码（包括 3010）不产生作用。

Output

所有任务都会将日志文件条目写入 agent.log 文件。executeScript 任务运行的脚本的其他输出会单独存储在一个动态命名的文件夹中，如下所示：

%LocalAppData%\Temp*EC2Launch#########**outputfilename.ext*

输出文件的确切路径包含在 agent.log 文件中，例如：

```
Program file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\UserScript.ps1
Output file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Output.tmp
Error file is created at: C:\Windows\system32\config\systemprofile\AppData\Local
\Temp\EC2Launch123456789\Err.tmp
```

executeScript 任务的输出文件

UserScript.*ext*

包含 executeScript 任务运行的脚本。文件扩展名取决于您在 executeScript 任务的 type 参数中指定的脚本类型，如下所示：

- 如果类型为 batch，则文件扩展名为 .bat。
- 如果类型为 powershell，则文件扩展名为 .ps1。

Output.tmp

包含 executeScript 任务运行的脚本的运行时系统输出。

Err.tmp

包含 executeScript 任务运行的脚本的运行时系统错误消息。

示例

以下示例演示了如何使用 executeScript 任务运行内联脚本。

示例 1：Hello world 输出文本文件

此示例演示了一个运行 PowerShell 脚本，以在 C: 驱动器上创建一个“Hello world”文本文件的 executeScript 任务。

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: admin
      content: |-
        New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
        Set-Content 'C:\PowerShellTest.txt' "Hello world"
```

示例 2：运行两个脚本

此示例说明了 executeScript 任务可以运行多个脚本，并且脚本类型不必一致。

第一个脚本 (type: powershell) 会将当前在实例上运行的进程的摘要写入 C: 驱动器上的一个文本文件。

第二个脚本 (batch) 会将系统信息写入 Output.tmp 文件。

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |
        Get-Process | Out-File -FilePath C:\Process.txt
    - frequency: always
      type: batch
      runAs: localSystem
      content: |
        systeminfo
```

示例 3：重启后的幂等系统配置

此示例显示了一个运行一个幂等脚本来执行以下系统配置，并在每个步骤之间重新启动的 executeScript 任务：

- 将计算机重命名。
- 将计算机加入到域中。
- 启用 Telnet。

该脚本会确保每个操作仅运行一次，从而防止循环重启并确保脚本的幂等性。

```
task: executeScript
  inputs:
    - frequency: always
      type: powershell
      runAs: localSystem
      content: |-
        $name = $env:ComputerName
        if ($name -ne $desiredName) {
          Rename-Computer -NewName $desiredName
          exit 3010
        }
        $domain = Get-ADDomain
        if ($domain -ne $desiredDomain)
        {
          Add-Computer -DomainName $desiredDomain
          exit 3010
        }
        $telnet = Get-WindowsFeature -Name Telnet-Client
        if (-not $telnet.Installed)
        {
          Install-WindowsFeature -Name "Telnet-Client"
          exit 3010
        }
```

extendRootPartition

扩展根卷以使用磁盘上的所有可用空间。

Frequency – 一次

AllowedStages – [Boot]

Inputs – 无

示例

```
task: extendRootPartition
```

initializeVolume

初始化附加到实例的空卷，以便对它们进行激活和分区。如果启动代理检测到卷不是空卷，则会跳过初始化。如果卷的前 4KiB 为空，或者卷没有 Windows 可识别的驱动器布局，则该卷被视为空卷。

无论驱动器是否已初始化，此任务运行时始终应用 `letter` 输入参数。

`initializeVolume` 任务执行以下操作。

- 将磁盘属性 `offline` 和 `readonly` 设置为 false。
- 创建分区。如果在 `partition` 输入参数中未指定分区类型，则应用以下默认值：
  - 如果磁盘大小小于 2TB，则将分区类型设置为 `mbr`。
  - 如果磁盘大小为 2TB 或更大，则将分区类型设置为 `gpt`。
- 将卷格式设置为 NTFS。
- 按如下方式设置卷标：
  - 使用 `name` 输入参数的值（如已指定）。
  - 如果卷是临时卷，并且未指定名称，请将卷标设置为 `Temporary Storage Z`。
- 如果卷是临时卷（SSD 或 HDD，而不是 Amazon EBS），请在卷的根目录下创建一个包含以下内容的 `Important.txt` 文件：

  ```
  This is an 'Instance Store' disk and is provided at no additional charge.

  *This disk offers increased performance since it is local to the host
  *The number of Instance Store disks available to an instance vary by instance type
  *DATA ON THIS DRIVE WILL BE LOST IN CASES OF IMPAIRMENT OR STOPPING THE INSTANCE.
   PLEASE ENSURE THAT ANY IMPORTANT DATA IS BACKED UP FREQUENTLY

  For more information, please refer to: ### EC2 ############.
  ```

- 将驱动器号设置为 `letter` 输入参数中指定的值。

阶段：您可以在 `PostReady` 和 `UserData` 阶段运行 `initializeVolume` 任务。

频率：始终。

输入

您可将运行时系统参数配置如下：

设备（映射列表）

（如有条件）启动代理初始化的每个设备的配置。当 initialize 输入参数设置为 devices 时，则为必需。

- 设备（字符串，必需）- 在创建实例期间标识设备。例如，xvdb、xvdf 或 \dev \nvme0n1。
- 字母（字符串，可选）- 一个字符。要分配的驱动器号。
- 名称（字符串，可选）- 要分配的卷名称。
- 分区（字符串，可选）- 为要创建的分区类型指定以下值之一，或者让启动代理根据卷大小进行默认设置：
  - mbr
  - gpt

初始化（字符串）

（必填）确切指定以下默认值中的一项：

- all
- devices

示例

以下示例显示了 initializeVolume 任务的示例输入配置。

示例 1：在一个实例上初始化两个卷

此示例显示了一个在实例上初始化两个辅助卷的 initializeVolume 任务。示例中名为 DataVolume2 的设备是临时的。

```
task: initializeVolume
inputs:
  initialize: devices
  devices:
  - device: xvdb
    name: DataVolume1
    letter: D
    partition: mbr
```

```
  - device: /dev/nvme0n1
    name: DataVolume2
    letter: E
    partition: gpt
```

示例 2：初始化附加到实例的 EBS 卷

此示例显示了一个对附加到实例的所有空 EBS 卷进行初始化的 initializeVolume 任务。

```
task: initializeVolume
inputs:
  initialize: all
```

optimizeEna

根据当前实例类型优化 ENA 设置；可能会重新引导实例。

Frequency – 始终

AllowedStages – [PostReady, UserData]

Inputs – 无

示例

```
task: optimizeEna
```

setAdminAccount

为在本地计算机上创建的管理员账户设置默认属性。

Frequency – 一次

AllowedStages – [PreReady]

输入 –

name：（字符串）管理员账户的名称

password：（映射）

type：（字符串）用于设置密码的策略，可为 static、random 或 doNothing

data：（字符串）存储数据，如果 type 字段为 static

示例

```
task: setAdminAccount
inputs:
  name: Administrator
  password:
  type: random
```

setDnsSuffix

将 DNS 后缀添加到搜索后缀列表中。只有尚不存在的后缀才会添加到列表中。有关启动代理如何设置 DNS 后缀的更多信息，请参阅 为 EC2 Windows 启动代理配置 DNS 后缀。

Frequency – 始终

AllowedStages – [PreReady]

输入 –

suffixes：（字符串列表）一个或多个有效 DNS 后缀的列表；有效替换变量是 $REGION 和 $AZ

示例

```
task: setDnsSuffix
inputs:
  suffixes:
  - $REGION.ec2-utilities.amazonaws.com
```

setHostName

将计算机的主机名设置为自定义字符串，如果未指定 hostName，则该值为私有 IPv4 地址。

Frequency – 始终

AllowedStages – [PostReady, UserData]

输入 –

hostName：（字符串）可选主机名，必须采用如下格式。

- 必须为 15 个字符或更少
- 必须仅包含字母数字字符（a-z、A-Z、0-9）和连字符 (-)。
- 不能完全由数字字符组成。

reboot：（布尔值）表示在更改主机名时是否允许重新引导

示例

```
task: setHostName
inputs:
  reboot: true
```

setWallpaper

在每个现有用户的启动文件夹中创建 setwallpaper.lnk 快捷方式文件，但 Default User 除外。当用户在实例启动后首次登录时，此快捷方式文件就会运行。使用显示实例属性的自定义壁纸设置实例。

快捷方式文件路径是：

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/
Startup/setwallpaper.lnk
```

> **Note**
>
> 删除 setWallpaper 任务不会删除此快捷方式文件。有关更多信息，请参阅 setWallpaper 任务未启用，但壁纸会在重新启动时重置。

阶段：您可以在 PreReady 和 UserData 阶段配置墙纸。

频率：always

墙纸配置

您可以使用以下设置配置墙纸。

输入

您提供的输入参数以及可设置以配置墙纸的属性：

路径（字符串）

（必需）用于墙纸图像的本地 .jpg 格式图像文件的文件名路径。

属性（字符串列表）

（可选）您可以向墙纸添加以下一项或多项属性：

- architecture
- availabilityZone
- hostName
- instanceId
- instanceSize
- privateIpAddress
- publicIpAddress
- ipv6Address

InstanceTags

（可选）您可以仅使用以下选项之一进行此设置。

- AllTags（字符串）– 将所有实例标签添加到墙纸。

```
instanceTags: AllTags
```

- InstanceTags（字符串列表）– 指定要添加到墙纸的实例标签名称列表。例如：

```
instanceTags:
  - Tag 1
  - Tag 2
```

示例

以下示例显示墙纸配置输入（可设置墙纸背景图像文件路径）、名为 Tag 1 和 Tag 2 的实例标签，以及实例的主机名、实例 ID 及私有和公有 IP 地址等属性。

```
task: setWallpaper
inputs:
  path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
  attributes:
  - hostName
  - instanceId
  - privateIpAddress
  - publicIpAddress
  instanceTags:
  - Tag 1
  - Tag 2
```

> **Note**
>
> 必须启用元数据中的标签，才能在墙纸上显示标签。有关实例标签和元数据的更多信息，请参阅 使用实例元数据来查看 EC2 实例的标签。

startSsm

在 Sysprep 之后启动 Systems Manager (SSM) 服务。

Frequency – 始终

AllowedStages – [PostReady, UserData]

Inputs – 无

示例

```
task: startSsm
```

sysprep

重置服务状态、更新 unattend.xml、禁用 RDP 并运行 Sysprep。此任务仅在所有其他任务完成后才会运行。

Frequency – 一次

AllowedStages – [UserData]

输入 –

clean：（布尔值）在运行 Sysprep 之前清理实例日志

shutdown：（布尔值）在运行 Sysprep 后关闭实例

示例

```
task: sysprep
inputs:
clean: true
shutdown: true
```

writeFile

将文件写入目标。

Frequency – 请参阅 Inputs

AllowedStages – [PostReady, UserData]

输入 –

frequency：（字符串）once 或 always 之一

destination：（字符串）写入内容的路径

content：（字符串）要写入目标的文本

示例

```
task: writeFile
inputs:
  - frequency: once
  destination: C:\Users\Administrator\Desktop\booted.txt
  content: Windows Has Booted
```

排查 EC2Launch v2 代理的问题

本节介绍 EC2Launch v2 的常见故障排除场景以及有关查看 Windows 事件日志及控制台日志输出和消息的信息。

故障排除主题

- 常见故障排除场景
- Windows 事件日志
- EC2Launch v2 控制台日志输出

常见故障排除场景

本节介绍常见的故障排除场景和解决步骤。

方案

- 服务无法设置壁纸

- 服务无法运行用户数据
- 服务只运行一次任务
- 服务无法运行任务
- 服务多次运行用户数据
- EC2Launch v1 中的计划任务在迁移到 EC2Launch v2 后无法运行
- 服务初始化不为空的 EBS 卷
- setWallpaper 任务未启用，但壁纸会在重新启动时重置
- 服务停留在运行状态
- 无效的 agent-config.yml 会阻止打开 EC2Launch v2 设置对话框
- task:executeScript should be unique and only invoked once

服务无法设置壁纸

解决方案

1. 确保存在 `%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\setwallpaper.lnk` 。
2. 检查 `%ProgramData%\Amazon\EC2Launch\log\agent.log` 是否发生任何错误。

服务无法运行用户数据

可能的原因：服务可能在运行用户数据之前失败。

解决方案

1. 检查 `%ProgramData%\Amazon\EC2Launch\state\previous-state.json`。
2. 查看 `boot`、`network`、`preReady` 和 `postReadyLocalData` 是否全部标记为成功。
3. 如果其中一个阶段失败，请检查 `%ProgramData%\Amazon\EC2Launch\log\agent.log` 中的具体错误。

服务只运行一次任务

解决方案

1. 检查任务的频率。

2. 如果服务已在 Sysprep 之后运行，并且任务频率设置为 `once`，则该任务不会再次运行。
3. 如果您希望任务在每次运行 EC2Launch v2 时运行，请将任务的频率设置为 `always`。

服务无法运行任务

解决方案

1. 检查 `%ProgramData%\Amazon\EC2Launch\log\agent.log` 中的最新条目。
2. 如果没有发生错误，请尝试从 `"%ProgramFiles%\Amazon\EC2Launch\EC2Launch.exe" run` 手动运行服务以查看任务是否成功。

服务多次运行用户数据

解决方案

用户数据在 EC2Launch v1 和 EC2Launch v2 之间的处理方式不同。将 `persist` 设置为 `true` 时，EC2Launch v1 在实例上将用户数据作为计划任务运行。如果 `persist` 设置为 `false`，则即使重新启动后退出或运行时中断，也不会计划该任务。

EC2Launch v2 将用户数据作为代理任务运行并跟踪其运行状态。如果用户数据发出计算机重启或用户数据在运行时中断，则运行状态将保持不变，`pending` 并且用户数据将在下次实例启动时再次运行。如果要阻止用户数据脚本多次运行，请使脚本具有幂等性。

以下示例幂极脚本设置计算机名称并加入域。

```
<powershell>
  $name = $env:computername
  if ($name -ne $desiredName) {
 Rename-Computer -NewName $desiredName
  }
  $domain = Get-ADDomain
  if ($domain -ne $desiredDomain)
  {
 Add-Computer -DomainName $desiredDomain
  }
  $telnet = Get-WindowsFeature -Name Telnet-Client
  if (-not $telnet.Installed)
  {
 Install-WindowsFeature -Name "Telnet-Client"
  }
```

```
</powershell>
<persist>false</persist>
```

EC2Launch v1 中的计划任务在迁移到 EC2Launch v2 后无法运行

解决方案

迁移工具不会检测任何链接到 EC2Launch v1 脚本的计划任务；因此，它不会在 EC2Launch v2 中自动设置这些任务。要配置这些任务，请编辑 `agent-config.yml` 文件或使用 EC2Launch v2 设置对话框。例如，如果实例的计划任务正在运行 `InitializeDisks.ps1`，则在运行迁移工具之后，您必须在 EC2Launch v2 设置对话框中指定要初始化的卷。请参阅 使用 EC2Launch v2 设置对话框更改设置 程序的步骤 6。

服务初始化不为空的 EBS 卷

解决方案

在初始化卷之前，EC2Launch v2 会尝试检测该卷是否为空。如果卷不为空，则会跳过初始化。不初始化任何检测到不为空的卷。如果卷的前 4 KiB 为空，或者卷没有 Windows 可识别的驱动器布局，则该卷被视为空。在 Linux 系统上初始化和格式化的卷没有 Windows 可识别的驱动程序布局，例如 MBR 或 GPT。因此，它将被视为空并已初始化。如果要保留此数据，请不要依赖 EC2Launch v2 空驱动器检测。相反，请指定要在 EC2Launch v2 设置对话框（请参阅步骤 6）或 `agent-config.yml` 中初始化的卷。

`setWallpaper` 任务未启用，但壁纸会在重新启动时重置

`setWallpaper` 任务在每个现有用户的启动文件夹中创建 `setwallpaper.lnk` 快捷方式文件。当用户在实例启动后首次登录时，此快捷方式文件就会运行。使用显示实例属性的自定义壁纸设置实例。删除 `setWallpaper` 任务不会删除此快捷方式文件。您必须手动删除此文件或使用脚本将其删除。

快捷方式路径是：

```
$env:SystemDrive/Users/<user>/AppData/Roaming/Microsoft/Windows/Start Menu/
Programs/Startup/setwallpaper.lnk
```

解决方案

手动删除此文件，或使用脚本将其删除。

删除快捷方式文件的 PowerShell 脚本示例

```
foreach ($userDir in (Get-ChildItem "C:\Users" -Force -Directory).FullName)
{
 $startupPath = Join-Path $userDir -ChildPath "AppData\Roaming\Microsoft\Windows\Start
 Menu\Programs\Startup"
 if (Test-Path $startupPath)
 {
  $wallpaperSetupPath = Join-Path $startupPath -ChildPath "setwallpaper.lnk"
  if (Test-Path $wallpaperSetupPath)
  {
   Remove-Item $wallpaperSetupPath -Force -Confirm:$false
  }
 }
}
```

服务停留在运行状态

描述

EC2Launch v2 被阻止，日志消息（`agent.log`）与以下类似：

```
2022-02-24 08:08:58 Info:
 ********************************************************************************
2022-02-24 08:08:58 Info: EC2Launch Service starting
2022-02-24 08:08:58 Info: Windows event custom log exists: Amazon EC2Launch
2022-02-24 08:08:58 Info: ACPI SPCR table not supported. Bailing Out
2022-02-24 08:08:58 Info: Serial port is in use. Waiting for Serial Port...
2022-02-24 08:09:00 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:02 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:04 Info: ACPI SPCR table not supported. Use default console port.
2022-02-24 08:09:06 Info: ACPI SPCR table not supported. Use default console port.
```

可能的原因

SAC 已启用并使用串行端口。有关更多信息，请参阅使用 SAC 排查 Windows 实例的问题。

解决方案

请尝试以下步骤来解决该问题：

- 禁用使用串行端口的服务。
- 如果您希望服务继续使用串行端口，请编写自定义脚本来执行启动代理任务，并将其作为计划的任务调用。

无效的 **agent-config.yml** 会阻止打开 EC2Launch v2 设置对话框

说明

EC2Launch v2 设置在打开对话框之前尝试解析 agent-config.yml 文件。如果 YAML 配置文件不遵循支持的架构，则对话框将显示以下错误：

```
Unable to parse configuration file agent-config.yml. Review configuration file. Exiting application.
```

解决方案

1. 验证配置文件是否遵循支持的架构。
2. 如果想从头开始，请将默认配置文件复制到 agent-config.yml 中。您可以使用“任务配置”部分中提供的示例 agent-config.yml。
3. 您也可以通过删除 agent-config.yml 重新开始。EC2Launch v2 设置会生成空配置文件。

task:executeScript should be unique and only invoked once

说明

不能在同一阶段重复一个任务。

解决方案

有些任务必须作为数组输入，例如 executeScript 和 executeProgram。有关如何将脚本编写为数组的示例，请参阅 executeScript。

Windows 事件日志

EC2Launch v2 发布重要事件的 Windows 事件日志，例如服务启动、Windows 已就绪以及任务成功和失败。事件标识符唯一标识特定事件。每个事件都包含阶段、任务和级别信息以及描述。您可以使用事件标识符为特定事件设置触发器。

事件 ID 提供有关事件的信息，并唯一标识某些事件。事件 ID 的最低有效位表示事件的严重性。

活动	最低有效位
Success	. . .0
Informational	. . .1

活动	最低有效位
Warning	. . .2
Error	. . .3

当服务启动或停止时生成的与服务相关的事件，包括个位数事件标识符。

活动	个位数标识符
Success	0
Informational	1
Warning	2
Error	3

EC2LaunchService.exe 事件的事件消息以 Service: 开头。EC2Launch.exe 事件的事件消息不以 Service: 开头。

四位数的事件 ID 包括有关事件阶段、任务和严重性的信息。

主题

- 事件 ID 格式
- 事件 ID 示例
- Windows 事件日志架构

事件 ID 格式

下表显示了 EC2Launch v2 事件标识符的格式。

3	2 1	0
S	T	L

3	2 1	0

表中的字母和数字代表以下事件类型和定义。

Event type	定义
S（阶段）	0 – 服务级别消息 1 – Boot 2 – Network 3 – PreReady 5 – Windows 准备就绪 6 – Postready 7 – UserData
T（任务）	由相应的两个值表示的任务在每个阶段都不同。要查看事件的完整列表，请参阅 Windows 事件日志架构。
L（事件级别）	0 – 成功 1 – 信息性 2 – 警告 3 – 错误

事件 ID 示例

以下是示例事件 ID。

- 5000 – Windows 已准备好使用

- 3010 – PreReady 阶段中的激活 Windows 任务已成功执行
- 6013 – Postready 本地数据阶段中的设置壁纸任务遇到错误

Windows 事件日志架构

消息/事件 ID	事件消息
. . .0	Success
. . .1	Informational
. . .2	Warning
. . .3	Error
x	EC2Launch service-level logs
0	EC2Launch service exited successfully
1	EC2Launch service informational logs
2	EC2Launch service warning logs
3	EC2Launch service error logs
10	Replace state.json with previous-state.json
100	Serial Port
200	Sysprep
300	PrimaryNic
400	Metadata
x000	Stage (1 digit), Task (2 digits), Status (1 digit)

消息/事件 ID	事件消息
1000	Boot
1010	Boot - extend_root_partition
2000	Network
2010	Network - add_routes
3000	PreReady
3010	PreReady - activate_windows
3020	PreReady - install_egpu_manager
3030	PreReady - set_monitor_on
3040	PreReady - set_hibernation
3050	PreReady - set_admin_account
3060	PreReady - set_dns_suffix
3070	PreReady - set_wallpaper
3080	PreReady - set_update_schedule
3090	PreReady - output_log
3100	PreReady - enable_open_ssh
5000	Windows is Ready to use
6000	PostReadyLocalData
7000	PostReadyUserData
6010/7010	PostReadyLocal/UserData - set_wallpaper

消息/事件 ID	事件消息
6020/7020	PostReadyLocal/UserData - set_update_schedule
6030/7030	PostReadyLocal/UserData - set_hostname
6040/7040	PostReadyLocal/UserData - execute_program
6050/7050	PostReadyLocal/UserData - execute_script
6060/7060	PostReadyLocal/UserData - manage_package
6070/7070	PostReadyLocal/UserData - initialize_volume
6080/7080	PostReadyLocal/UserData - write_file
6090/7090	PostReadyLocal/UserData - start_ssm
7100	PostReadyUserData - enable_open_ssh
6110/7110	PostReadyLocal/UserData - enable_jumbo_frames

EC2Launch v2 控制台日志输出

本部分包含 EC2Launch v2 的示例控制台日志输出，并列出所有 EC2Launch v2 控制台日志错误消息，以帮助您排查问题。有关实例控制台输出及如何进行访问的更多信息，请参阅 the section called “实例控制台输出”。

输出

- EC2Launch v2 控制台日志输出
- EC2Launch v2 控制台日志消息

EC2Launch v2 控制台日志输出

以下是 EC2Launch v2 的示例控制台日志输出。本示例中的某些值被用大括号括起来的代表性文本替换。

```
2025/07/22 21:26:53Z: Windows sysprep configuration complete.
2025/07/22 21:26:53Z: Message: Waiting for access to metadata...
2025/07/22 21:26:53Z: Message: Meta-data is now available.
2025/07/22 21:26:53Z: AMI Origin Version: 2024.12.13
2025/07/22 21:26:53Z: AMI Origin Name: Windows_Server-2022-English-Full-Base
2025/07/22 21:26:53Z: OS: Microsoft Windows NT 10.0.20348
2025/07/22 21:26:53Z: OsVersion: 10.0
2025/07/22 21:26:53Z: OsProductName: Windows Server 2022 Datacenter
2025/07/22 21:26:53Z: OsBuildLabEx: 20348.1.amd64fre.fe_release.210507-1500
2025/07/22 21:26:53Z: OsCurrentBuild: 20348
2025/07/22 21:26:53Z: OsReleaseId: 2009
2025/07/22 21:26:53Z: Language: en-US
2025/07/22 21:26:53Z: TimeZone: UTC
2025/07/22 21:26:53Z: Offset: UTC +0000
2025/07/22 21:26:53Z: Launch: EC2 Launch v2.2.63
2025/07/22 21:26:53Z: AMI-ID: ami-1234567890abcdef1
2025/07/22 21:26:53Z: Instance-ID: i-1234567890abcdef0
2025/07/22 21:26:54Z: Instance Type: t3.xlarge
2025/07/22 21:26:54Z: Driver: AWS NVMe Driver v1.6.0.35
2025/07/22 21:26:54Z: SubComponent: 1.6.0.35; EnableSCSIPersistentReservations: 0
2025/07/22 21:26:54Z: Driver: AWS PV Driver Package v8.5.0
2025/07/22 21:26:55Z: Driver: Amazon Elastic Network Adapter v2.8.0.0
2025/07/22 21:26:55Z: Driver: EC2 Windows Utility Device v3.1.0.25
2025/07/22 21:26:55Z: HOSTNAME: EC2AMAZ-9FJG5CC
2025/07/22 21:26:55Z: RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
2025/07/22 21:26:55Z: RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
2025/07/22 21:26:56Z: SSM: Amazon SSM Agent v3.3.2746.0
2025/07/22 21:26:57Z: User data format: no_user_data
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IsTelemetryEnabled=true
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentOsArch=windows_amd64
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentCommandErrorCode=1
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AdminPasswordTypeCode=0
2025/07/22 21:26:57Z: EC2LaunchTelemetry: AgentErrorLocation=execute_windows.go:410
```

```
2025/07/22 21:26:57Z: EC2LaunchTelemetry: IpConflictDetectionCode=0
2025/07/22 21:26:57Z: Message: Windows is Ready to use
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2", ...}
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2", ...}
```

EC2Launch v2 控制台日志消息

以下是所有 EC2Launch v2 控制台日志消息的列表。

```
Error EC2Launch service is stopping. {error message}
```

停止服务错误详情：

- Error setting up EC2Launch agent folders
- See instance logs for detail
- Error stopping service
- Error initializing service

```
Windows sysprep configuration complete
```

```
Invalid administrator username: {invalid username}
```

```
Invalid administrator password
Username: {username}
Password: <Password>{encrypted password}</Password>
```

以下消息是包含 AMI 详细信息的信息块：

```
AMI Origin Version: {amiVersion}
AMI Origin Name: {amiName}
Microsoft Windows NT {currentVersion}.{currentBuildNumber}
OsVersion: {currentVersion}
OsProductName: {productName}
OsBuildLabEx: {buildLabEx}
OsCurrentBuild: {currentBuild}
OsReleaseId: {releaseId}
Language: {language}
```

```
TimeZone: {timeZone}
Offset: UTC {offset}
Launch agent: EC2Launch {BuildVersion}
AMI-ID: {amiId}
Instance-ID: {instanceId}
Instance Type: {instanceType}
HOSTNAME: {computer name}
RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
SqlServerBilling: {sql billing}
SqlServerInstall: {sql patch leve, edition type}
Driver: Amazon NVMe Driver {version}
Driver: Inbox NVMe Driver {version}
Driver: Amazon PV Driver Package {version}
Driver: EC2 Windows Utility Device {version}
SSM: Amazon SSM Agent {version}
Amazon VSS Version: {version}
```

```
Windows sysprep configuration complete.
Windows is being configured. 'SysprepState is {state}'
Windows is still being configured. 'SysprepState is {state}'
Windows is Ready to use
Waiting for access to metadata...
Meta-data is now available.
Metadata is not available for this instance.
Timed out waiting for access to metadata.
User data format: {format}
```

EC2Launch v2 遥测消息包括启动遥测属性值。从 2.2.63 版本开始，EC2 代理遥测数据被格式化为 JSON 对象。

```
EC2LaunchTelemetry: {telemetry property}
```

```
{"type":"EC2AgentTelemetry","agentId":"WindowsLaunchAgentV2" ... }
```

EC2Launch v2 版本历史记录

版本历史记录

- EC2Launch v2 版本历史记录

EC2Launch v2 版本历史记录

要确保安装了最新的启动代理，请参阅安装最新版本的 EC2Launch v2。您可在 EC2Launch v2 代理的新版本发布时收到通知。有关更多信息，请参阅 订阅 EC2 Windows 启动代理通知。

下表说明已发行的 EC2Launch v2 版本。

版本	详细信息	发行日期
2.5.1	• 改进了启用休眠时的错误检测。	2026 年 6 月 3 日
2.5.0	• 更新了串行端口逻辑，以便在串行端口不可用时继续执行代理任务。 • 实例控制台输出中新增了 EC2 Windows 实用程序设备驱动程序版本。 • 移除了未使用的 bench.log 文件。 • 优化了写入实例控制台输出时的驱动程序查找。	2026 年 5 月 4 日
2.4.0	• 通过删除冗余 `console.log` 、`err.log` 文件来简化代理日志记录。 • 通过添加日志轮换、将日志大小限制为 1 MB 并将时间精度提高到毫秒，增强了代理日志记录功能。 • 移除了 Hyper-V 状态查询和相关的实例控制台输出。	2026 年 3 月 5 日
2.3.237	• 修复了因某些 Visual C++ 运行时版本与代理遥测函数不兼容导致的代理崩溃问题。	2026 年 2 月 25 日
2.3.108	• 在壁纸上增加了 IPv6 地址显示功能。有关内存和网络属性的信息，请参阅 Amazon 文档。	2025 年 12 月 29 日

版本	详细信息	发行日期
2.3.56	• 修复了以下问题：EC2Launch v2 未正确处理预配置的永久网络路由。	2025 年 11 月 4 日
2.3.5	• 在添加路由时使用 IMDS 改进了主网络接口检测。	2025 年 9 月 15 日
2.2.63	• 更新启动代理以改进 遥测。	2025 年 7 月 29 日
2.1.1	• 增加了对 IPv6 的全面支持。	2025 年 5 月 14 日
2.0.2107	• 增加了增强路由功能，用于处理接口上没有 IPv4 或 IPv6 地址可用的情况。	2025 年 3 月 27 日
2.0.2081	• 修复了未正确检索或验证 RDP 证书信息的问题。添加了在需要时自动启动远程桌面服务的功能。 • 调整了 EC2Launch v2 服务权限以修复查询服务状态时出现的问题。	2025 年 2 月 4 日
2.0.2046	• 更新了 `agent-config.yml` 文件中的壁纸路径，以使用默认的操作系统壁纸路径。 • 增加了遥测功能，用于监控代理出现错误的位置。 • 更新了代理日志消息收发。	2024 年 10 月 3 日
2.0.1981	• 更新了非管理员用户的 `EC2Launch.exe` CLI 命令错误消息。	2024 年 8 月 6 日

版本	详细信息	发行日期
2.0.1948	• 添加了遥测来监控管理员密码选项的使用情况。 • 修改了 EC2Launch 权限。	2024 年 7 月 1 日
2.0.1924	• 更新了 EC2Launch 设置 UI。 • 更新了壁纸 CLI 命令。 • 更新了 EC2Launch 安装程序。	2024 年 6 月 10 日
2.0.1914	• 添加具有未指定网关地址的路由（IPv4 为 `0.0.0.0`，或 IPv6 为 `::`）。 • 务必同时添加 IPv4 和 IPv6 路由。 • 修复了未指定 `Administrator` 用户名时将其添加到 `agent-config.yml` 文件中的问题。 • 修改了 EC2Launch v2 权限。	2024 年 6 月 5 日
2.0.1881	• 在 `setAdminAccount` 任务中添加了加密密码选项。 • 添加了 CLI 命令来加密 agent-config.yml 中的加密静态密码。 • 修复了 XML 用户数据在使用管理员权限运行时无法添加 PowerShell 参数的问题。有关更多详细信息，请参阅Amazon EC2 如何处理 Windows 实例的用户数据。 • 调整了 `executeScript` 任务和用户数据脚本使用 `LocalSystem` 权限运行时的 PowerShell 参数。参数为空时，代理使用以下默认值：`-ExecutionPolicy Unrestricted` 。 • 已阻止将重复的驱动程序版本打印到控制台日志。	2024 年 5 月 8 日

版本	详细信息	发行日期
2.0.1815	• 在 sysprep 之前，将错误处理调整为在关键设置问题上失败。 • 修复了壁纸和主机名任务在多个 IP 地址分配给主网络接口的实例上使用错误 IP 地址的问题。 • 壁纸和主机名任务更改为先从 IMDS 获取私有 IP，如果禁用 IMDS，则故障恢复到 WMI。 • 修复了 `initializeVolume` 任务中由于暂时性错误而导致 `sc1` 卷无法初始化的问题。	2024 年 3 月 6 日
2.0.1739	• 修复了以 Windows 管理员用户身份运行的 `executeScript` 任务无法捕获退出代码的问题。	2024 年 1 月 17 日
2.0.1702	• 对于标准用户，将 `Telemetry.log` 权限限制为仅 `read-execute` 。 • 将 EC2Launch Windows 服务配置为在启动失败时重新启动。 • 通过记录 `route.exe` `stderr` 输出，使 `add-routes` 故障变得可操作。 • 修复了路由指标超出范围 [1, 9999] 时出现的问题。 • 为几种新实例类型添加了壁纸支持。 • 修复了以 Windows 管理员用户身份运行并将输出发送到 `stderr` 的用户数据脚本引起的问题。	2024 年 1 月 4 日

版本	详细信息	发行日期
2.0.1643	• 已将 `ebsnvme-id.exe` 工具更新为版本 1.1.0.7。 • 修复了以“metal-*”开头的裸机实例类型（例如 metal-48x1）的接收端缩放（RSS）和接收队列深度设置的问题。 • 删除了报告阻止代理的 XML UserData 命令的遥测事件。 • 更新了 `setDnsSuffix` 任务以基于以下注册表条目限制域名传递：`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel` 。 • 添加了公共任务和添加网络路由的 CLI。 • 注意：这是最后一个正式支持 Windows Server 2012 的版本。 • 注意：这是最后一个正式支持 32 位操作系统的版本。	2023 年 10 月 4 日
2.0.1580	• 更改在您修改日志文件权限时启动代理处理错误的方式。 • 添加连接串行端口的超时时间。如果正在使用串行端口，则超时允许启动代理继续运行。	2023 年 9 月 5 日

版本	详细信息	发行日期

2.0.1521	• 已弃用 EC2Launch.exe reset 和 sysprep 命令的 –block 标记。
	• 已更新 EC2Launch.exe 以检测和处理内联 executeScript 任务中使用的 reset 和 sysprep 命令。这些命令会导致代理在 executeScript 任务运行它们后停止运行。
	• 更新了 XML 用户数据脚本，使其在默认情况下以内联方式运行。
	• 启用 XML 用户数据脚本以便与新的 detach 标签分离运行。有关更多详细信息，请参阅用户数据脚本。
	• 对代理日志进行了以下更改。
	  • 更新了代理日志消息。
	  • 从代理日志中删除了 executeScript 内容和输出。
	  • 从代理日志中删除了 executeProgram 参数和输出。
	• 对控制台日志进行了以下更改。
	  • 为控制台日志增加了 EnableSCSIPersistentReservations 值。
		2023 年 7 月 3 日

版本	详细信息	发行日期
2.0.1303	• 增加了添加网络路由时的额外错误处理和日志行。 • 允许在 PreReady 阶段的 `executeScript` 和 `executeProgram` 任务。 • 更新了 `executeProgram` 任务，以生成与 executeScript 任务输出类似的输出文件。有关更多信息，请参阅 executeProgram。 • 增加了遥测功能，以用于监控 XML 用户数据中阻止代理命令的使用情况。	2023 年 5 月 3 日
2.0.1245	• 以明文形式记录崩溃调用堆栈，从而提高了崩溃可见性。 • 添加了 EventLog 服务作为启动依赖项，以修复 Amazon EC2Launch 服务启动速度快于 EventLog 服务时发生的崩溃。 • 代理配置文件（如 EC2Launch v1 和 EC2Config）中的 XML 用户数据将在 PostReady 阶段之前运行。 • 添加了 YAML 用户数据版本 1.1，代理配置文件中的用户数据将在 PostReady 阶段之前运行（代理配置文件的 YAML 用户数据版本 1.0 则会在 PostReady 阶段之后运行）。	2023 年 3 月 8 日
2.0.1173	• 新增了在墙纸上显示实例标签的可选功能。有关更多信息，请参阅 setWallpaper。 • 新增了 Elastic Graphics 的安全组设置有误时的错误处理功能。 • 修复了未启用实例元数据服务时的超时问题。	2023 年 2 月 6 日

版本	详细信息	发行日期
2.0.1121	• 修复了未分配公有 IPv4 地址时将 404 错误打印到墙纸的问题。 • 修复了卷的文件系统在其设备驱动器号设为 D 时，格式化为 RAW 而不是 NTFS 的问题。 • 修复了将 NVMe SSD 卷错误识别为 EBS 卷的问题。 • 修复了禁用 IMDS 时激活 Windows 的错误。	2023 年 1 月 4 日
2.0.1082	• 修复了禁用 IMDS 时 `setWallpaper` ：`privateIp Address` 字段为空的问题。 • 修复了禁用 IMDS 时将主机名设置为私有 IPv4 地址的问题。 • 修复了在 Windows Server 2012 上初始化卷的问题。 • 修复了设置巨型帧的问题。 • 修复了实例启动时未指定 SSH 密钥时的错误。 • 修复了当 Windows 没有“ReleaseId”注册表项时 Windows Server 2012 上的错误。	2022 年 12 月 7 日
2.0.1011	• 修复了 PnPDeviceID 为空时查找网络适配器的逻辑。	2022 年 11 月 11 日
2.0.1009	• 使用 PCI 分段信息选择控制台端口。	2022 年 11 月 8 日

版本	详细信息	发行日期
2.0.982	• 添加重试逻辑以获取 RDP 信息。 • 修复了在 d2.8xlarge 实例上进行卷初始化期间的错误。 • 修复了重启后可以选择不正确的网络适配器的问题。 • 移除 ACPI SPCR 不可用时的误报错误消息。	2022 年 10 月 31 日
2.0.863	• 更新了的 IMDS 等待逻辑以仅发出 IMDSv2 请求。 • 添加了可将驱动器号分配给已初始化但尚未装入的卷的逻辑。 • 当密钥对类型不受支持时，会显示更具体的错误消息。 • 修复了 3010 重启代码错误。 • 添加了对无效 base64 编码用户数据的检查。	2022 年 7 月 6 日
2.0.698	• 修复了执行脚本时日志输出中的错别字。	2022 年 1 月 30 日
2.0.674	• 遥测上载启用/禁用的隐私控制。 • 修复 index out of bounds 错误。 • 在 sysprep 期间删除壁纸快捷键。	2021 年 11 月 15 日
2.0.651	• 添加逻辑以在 EC2Launch v2 安装期间卸载旧版代理。 • 修复根卷未列为卷 0 时出现的 list-volume CLI 问题。	2021 年 10 月 7 日

版本	详细信息	发行日期
2.0.592	• 修复错误以正确报告阶段状态。 • 删除日志文件关闭时的错误告警的错误信息。 • 添加遥测。	2021 年 8 月 31 日
2.0.548	• 为十六进制 IP 主机名添加前导零。 • 修复了 `enableOpenSsh` 任务的文件权限。 • 修复了 sysprep 命令崩溃。	2021 年 8 月 4 日
2.0.470	• 修复了网络阶段等待 DHCP 向实例分配 IP 的错误。 • 修复了 `SearchList` 注册表项不存在时，`setDnsSuffix` 的错误。 • 修复了 `setDnsSuffix` 中的 DNS 传递逻辑中的错误。 • 中途重新启动后添加网络路由。 • 允许 `initializeVolume` 重新编写现有卷的字母。 • 从版本子命令中删除了额外信息。	2021 年 7 月 20 日
2.0.285	• 增加了在已分离的进程中运行用户脚本的选项。 • 旧用户数据（XML 用户数据）现在在一个分离的进程中运行，这是与之前的启动代理类似的行为。 • 将 CLI 标志添加到 `sysprep` 和 `reset` 命令，这允许它们在服务停止之前阻塞。 • 限制配置文件夹的权限。	2021 年 3 月 8 日

版本	详细信息	发行日期
2.0.207	• 向 hostName 任务添加可选 setHostName 字段。 • 修复了重启错误。重启任务 executeScript 和 executeProgram 将标记为“running”。 • 向状态命令添加更多返回代码。 • 添加引导服务以修复在 t2.nano 实例类型上运行时的启动问题。 • 修复了净安装模式以删除安装程序未跟踪的文件。	2021 年 2 月 2 日
2.0.160	• 修复 validate 命令以检测无效的阶段名称。 • 在 w32tm resync 任务中添加 addroutes 命令。 • 修复了更改 DNS 后缀搜索顺序的问题。 • 添加检查条件以更好地报告无效用户数据。	2020 年 12 月 4 日
2.0.153	在 UserData 中添加 Sysprep 功能。	2020 年 11 月 3 日

版本	详细信息	发行日期
2.0.146	• 修复了非英语 AMI 上的 RootExtend 的问题。 • 授予用户组对日志文件的写入权限。 • 为 GPT 卷创建 MS 预留分区。 • 在 Amazon EC2Launch 设置中添加 list-volumes 命令和卷下拉列表。 • 添加了 get-agent-config 命令，用于按 yaml 或 json 格式打印 agent-config.yml 文件。 • 如果未检测到公有密钥，则清除静态密码。	2020 年 10 月 6 日
2.0.124	• 添加了在壁纸上显示操作系统版本的选项。 • 初始化加密的 EBS 卷。 • 为没有本地 DNS 名称的 VPC 添加路由。	2020 年 9 月 10 日
2.0.104	• 创建 DNS 后缀搜索列表（如果不存在）。 • 如果没有请求，则跳过休眠状态。	2020 年 8 月 12 日
2.0.0	首次发布。	2020 年 6 月 30 日

使用 EC2Launch v1 代理在 EC2 Windows 实例启动期间执行任务

适用于 Windows Server 2016 和 2019 的 Amazon 托管 AMI 包含一组名为 EC2Launch 的 Windows Powershell 脚本。EC2Launch 在初始实例启动期间执行各种任务。有关 Amazon Windows AMI 中包含的 EC2Launch 版本的信息，请参阅 Amazon Windows AMI Reference。

> **Note**
>
> Windows Server 2016 及更高版本操作系统的最新启动代理是 EC2Launch v2，它取代了 EC2Config 和 EC2Launch，并预安装在名称以 `EC2LaunchV2-Windows_Server-*` 开头的 Amazon Windows Server 2016 和 2019 AMI 上。您也可以在 Windows Server 2016 和 2019 上手动安装和配置代理。有关更多信息，请参阅 安装 EC2Launch v2。
> 要将 EC2Launch 与 IMDSv2 一起使用，版本必须为 1.3.2002730 或更高版本。

您可以使用以下 Windows PowerShell 命令验证已安装的 EC2Launch 版本。

```
Test-ModuleManifest -Path "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1" | Select Version
```

EC2Launch 任务

在初始实例启动期间，默认情况下 EC2Launch 执行以下任务：

- 设置用来呈现实例相关信息的新墙纸。
- 将计算机名称设置为实例的私有 IPv4 地址。
- 向 Amazon EC2 控制台发送实例信息。
- 向 EC2 控制台发送 RDP 证书指纹。
- 设置管理员账户的随机密码。
- 添加 DNS 后缀。
- 动态扩展操作系统分区以包含所有未分区的空间。
- 执行用户数据 (如果指定)。有关指定用户数据的更多信息，请参阅在启动包含用户数据输入的 EC2 实例时运行命令。
- 设置持久静态路由以访问元数据服务和 Amazon KMS 服务器。

> **Important**
>
> 如果从此实例创建了自定义 AMI，则捕获这些路由作为操作配置的一部分，并且从 AMI 启动的任意新实例将保留相同的路由，而不论其子网放置如何。要更新路由，请参阅在启动自定义 AMI 时更新 Server 2016 及更高版本的元数据/KMS 路由。

以下任务有助于保持与 EC2Config 服务的向后兼容。在启动期间，您还可以配置 EC2Launch 来执行以下任务：

- 初始化辅助 EBS 卷。
- 将 Windows 事件日志发送到 EC2 控制台日志。
- 将 Windows 已可供使用 消息发送到 EC2 控制台。

EC2Launch 目录结构

默认情况下，EC2Launch 安装在根目录 `C:\ProgramData\Amazon\EC2-Windows\Launch` 中的 Windows Server 2016 及更高版本 AMI 上。

> **Note**
>
> 默认情况下，Windows 会隐藏 `C:\ProgramData` 下的文件和文件夹。要查看 EC2Launch 目录和文件，必须在 Windows 资源管理器中键入路径，或者更改文件夹属性以显示隐藏的文件和文件夹。

`Launch` 目录包含以下子目录。

- `Scripts` — 包含组成 EC2Launch 的 PowerShell 脚本。
- `Module` — 包含用于生成与 Amazon EC2 相关的脚本的模块。
- `Config` — 包含您可以自定义的脚本配置文件。
- `Sysprep` — 包含 Sysprep 资源。
- `Settings` — 包含一个适用于 Sysprep 图形用户界面的应用程序。
- `Library` — 包含 EC2 启动代理的共享库。
- `Log` — 包含脚本的子目录和脚本生成的日志文件。

遥测

遥测是附加信息，可帮助 Amazon 更好地了解您的需求、诊断问题并提供功能，以改善 Amazon 服务带给您的体验。

EC2Launch 版本 `1.3.2003498` 及更高版本会收集遥测，例如使用情况指标和错误。此数据是从运行 EC2Launch 的 Amazon EC2 实例收集的。这包括 Amazon 拥有的所有 Windows AMI。

EC2Launch 收集以下类型的遥测：

- 使用情况信息 — 代理命令、安装方法和计划的运行频率。
- 错误和诊断信息 — 代理安装和运行错误代码。

收集的数据示例：

```
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsAgentScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: IsUserDataScheduledPerBoot=true
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandCode=1
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentCommandErrorCode=5
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallCode=2
2021/07/15 21:44:12Z: EC2LaunchTelemetry: AgentInstallErrorCode=0
```

预设情况下，遥测处于启用状态。您可以随时禁用遥测收集。如果启用了遥测，EC2Launch 会发送遥测数据，无需额外的客户通知。

已收集您启用或禁用遥测的选择。

您可以选择或取消遥测收集。收集您选择或取消遥测的选择，以确保我们能够遵守您的遥测选项。

遥测可见性

启用遥测后，它将显示在 Amazon EC2 控制台输出中，如下所示：

```
2021/07/15 21:44:12Z: Telemetry: <Data>
```

在实例上禁用遥测

要通过设置系统环境变量来禁用遥测，请以管理员身份运行以下命令：

```
setx /M EC2LAUNCH_TELEMETRY 0
```

如需在安装过程中禁用遥测，请按照如下所示运行 `install.ps1`：

```
. .\install.ps1 -EnableTelemetry:$false
```

EC2Launch 的更多主题

- 在 Windows 实例上配置 EC2Launch v1 代理

- EC2Launch 版本历史记录

在 Windows 实例上配置 EC2Launch v1 代理

在您的实例首次初始化后，您可以配置 EC2Launch 以再次运行并执行不同的启动任务。

任务

- 配置初始化文件
- 计划 EC2Launch 在每次启动时运行
- 初始化驱动器并映射盘符
- 将 Windows 事件日志发送到 EC2 控制台
- 在成功启动后发送 Windows 已准备就绪消息

配置初始化文件

指定 `LaunchConfig.json` 文件中的设置以启用或禁用以下初始化任务：

- 将计算机名称设置为实例的私有 IPv4 地址。
- 将显示器设置为始终保持打开状态。
- 设置新墙纸。
- 添加 DNS 后缀列表。

> **Note**
>
> 这会为以下域添加 DNS 后缀查找，并配置其他标准后缀。有关启动代理如何设置 DNS 后缀的更多信息，请参阅 为 EC2 Windows 启动代理配置 DNS 后缀。
>
> ```
> region.ec2-utilities.amazonaws.com
> ```

- 扩展引导卷大小。
- 设置管理员密码。

配置初始化设置

1. 在要配置的实例上，在文本编辑器中打开以下文件：`C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json`。

2. 根据需要更新以下设置并保存您的更改。仅当 adminPassword 为 adminPasswordtype 时，在 Specify 中提供密码。

```
{
 "setComputerName": false,
 "setMonitorAlwaysOn": true,
 "setWallpaper": true,
 "addDnsSuffixList": true,
 "extendBootVolumeSize": true,
 "handleUserData": true,
 "adminPasswordType": "Random | Specify | DoNothing",
 "adminPassword": "password that adheres to your security policy (optional)"
}
```

密码类型定义如下：

Random

EC2Launch 生成一个密码并使用用户的密钥对其进行加密。系统会在实例启动后禁用此设置，以便在重新启动或停止再启动实例后该密码仍然存在。

Specify

EC2Launch 使用您在 adminPassword 中指定的密码。如果密码不满足系统要求，EC2Launch 会生成随机密码。该密码以明文方式存储在 LaunchConfig.json 文件中，并且在 Sysprep 设置管理员密码时会被删除。EC2Launch 使用用户的密钥对密码进行加密。

DoNothing

EC2Launch 使用您在 unattend.xml 文件中指定的密码。如果未在 unattend.xml 中指定密码，管理员账户会被禁用。

3. 在 Windows PowerShell 中运行以下命令，以便安排脚本作为 Windows 计划任务运行。该脚本将在下次启动期间执行一次，然后禁止这些任务再次运行。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

计划 EC2Launch 在每次启动时运行

您可以计划 EC2Launch 在每次启动时运行，而不是仅仅在首次启动时运行。

要允许 EC2Launch 在每次启动时运行，请执行以下操作：

1. 打开 Windows PowerShell 并运行以下命令：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
SchedulePerBoot
```

2. 或者，使用以下命令运行可执行文件：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe
```

然后选择 Run EC2Launch on every boot。您可以为 EC2 实例指定 Shutdown without Sysprep 或 Shutdown with Sysprep。

> **Note**
>
> 当您允许 EC2Launch 在每次启动时运行，下次 EC2Launch 运行时将发生以下情况：
>
> - 如果 AdminPasswordType 仍设置为 Random，则 EC2Launch 将在下次启动时生成新密码。在启动之后，AdminPasswordType 会自动设置为 DoNothing，以防止 EC2Launch 在后续启动时生成新密码。要防止 EC2Launch 在第一次启动时生成新密码，请手动将 AdminPasswordType 设置为 DoNothing，然后再重新启动。
> - 除非用户数据的 HandleUserData 设置为 false，否则 persist 将设置回 true。有关更多信息，请参阅 the section called “用户数据脚本”。

初始化驱动器并映射盘符

在 DriveLetterMappingConfig.json 文件中指定设置以将盘符映射到您的 EC2 实例上的卷。该脚本用于初始化尚未初始化和分区的驱动器。有关在 Windows 中获取卷详细信息的更多信息，请参阅 Microsoft 文档中的 Get-Volume。

将盘符映射到卷

1. 在文本编辑器中打开 C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json 文件。
2. 指定以下卷设置并保存您的更改：

```
{
 "driveLetterMapping": [
```

```
  {
   "volumeName": "sample volume",
   "driveLetter": "H"
  }
 ]
}
```

3. 打开 Windows PowerShell 并使用以下命令来运行初始化磁盘的 EC2Launch 脚本：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

要在每次实例启动时初始化磁盘，请添加 `-Schedule` 标记，如下所示：

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -Schedule
```

将 Windows 事件日志发送到 EC2 控制台

在 `EventLogConfig.json` 文件中指定设置，以便将 Windows 事件日志发送到 EC2 控制台日志。

配置设置来发送 Windows 事件日志

1. 在实例上，在文本编辑器中 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogConfig.json` 打开文件。
2. 配置以下日志设置并保存您的更改：

```
{
 "events": [
  {
   "logName": "System",
   "source": "An event source (optional)",
   "level": "Error | Warning | Information",
   "numEntries": 3
  }
 ]
}
```

3. 在 Windows PowerShell 中运行以下命令，以便每当实例启动时系统都安排脚本作为 Windows 计划任务运行。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendEventLogs.ps1 -Schedule
```

日志可能需要三分钟或更长时间才会显示在 EC2 控制台日志中。

在成功启动后发送 Windows 已准备就绪消息

在每次启动后，EC2Config 服务会向 EC2 控制台发送“Windows 已准备就绪”消息。EC2Launch 仅在初次启动后发送此消息。为了实现与 EC2Config 服务的向后兼容，您可以安排 EC2Launch 在每次启动时发送此消息。在实例上，打开 Windows PowerShell 并运行以下命令。系统会安排脚本作为 Windows 计划任务运行。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -Schedule
```

EC2Launch 版本历史记录

下表说明已发行的 EC2Launch 版本。

版本	详细信息	发行日期
1.4.299	• 优化了检索 EC2WinUtil 驱动程序版本的逻辑。 • 通过移除 Hyper-V 状态查询，缩短了 Windows 准备就绪所需的时间。	2026 年 3 月 3 日
1.4.183	• 更新了安装程序逻辑，以防止在 Windows Server 2022 上安装不受支持。 • 更新为将 EC2WinUtil 驱动程序版本打印到实例控制台日志中。	2026 年 2 月 4 日
1.4.6	• 更新了代理日志消息收发。	2025 年 10 月 13 日
1.3.2005119	• 修复了在没有任何参数的情况下调用 `Invoke-Userdata` 时会失败的问题。	2025 年 2 月 11 日
1.3.2005065	• 修复了未正确检索或验证 RDP 证书信息的问题。添加了在需要时自动启动远程桌面服务的功能。	2024 年 10 月 22 日

版本	详细信息	发行日期
1.3.2005008	• 更新 Set-Wallpaper 以在找不到默认墙纸图像时恢复纯色背景。	2024 年 8 月 6 日
1.3.2004959	• 更新了安装程序逻辑，以防止在 Windows Server 2025 或更高版本上安装不受支持。	2024 年 7 月 2 日
1.3.2004891	• 修复 HandleUserData 未按预期设置为 false 的问题。 • 在 LaunchConfig.json 中添加 Encrypted 密码选项。 • 更改默认情况下对用户指定的密码进行加密的 Settings UI 行为。 • 添加 SetAdminPasswordConfig.ps1 以将 Specify 密码选项转换为代理配置文件中的 Encrypted 密码选项。	2024 年 5 月 31 日
1.3.2004617	• 修复了设置壁纸时的错误。	2024 年 1 月 15 日

版本	详细信息	发行日期
1.3.2004592	• 更新了 install.ps1 为 `%ProgramData%\Amazon\EC2-Windows\Launch` 设置的访问权限。 • 对于标准用户账户，将 EC2Launch 文件夹/文件的访问权限限制为仅读取执行。 • 将座席更改为在实例未启用 IMDS 时停止等待实例元数据服务（IMDS）初始化。 • 在等待 IMDS 初始化时增加了五分钟超时时间。 • 将座席更改为在 `Windows is Ready` 消息之前（而不是之后）将遥测数据写入实例控制台日志。 • 为几种新实例类型添加了壁纸支持。 有关 EC2Launch 目录的访问权限和用户账户权限的更多信息，请参阅 the section called “EC2Launch 目录结构”。	2024 年 1 月 2 日
1.3.2004491	• 添加了遥测以监控指定管理员密码选项的使用情况。	2023 年 11 月 9 日
1.3.2004462	• 每次写入串行控制台后都添加了刷新。	2023 年 10 月 18 日

版本	详细信息	发行日期
1.3.2004438	• 基于以下注册表条目限制域名传递：HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters\DomainNameDevolutionLevel 。 • 将 UserdataExecution.log 权限限制为仅限 Administrators 。 • 在 Windows Event Log 中添加了日志初始化失败时的错误消息。	2023 年 10 月 4 日
1.3.2004256	• 为控制台日志增加了 EnableSCSIPersistentReservations 值。 • 为 Get-ConsolePort 增加了重试功能。	2023 年 7 月 7 日
1.3.2004052	• 修复了实例启动时未指定 SSH 密钥时发生的错误。 • 已更新为在出现故障时重试启动 AmazonSSMAgent Windows 服务。 • 已更新为在 BeforeSysprep.cmd 出现故障并出现非零的退出代码时，SysprepInstance.ps1 将失败。	2023 年 3 月 8 日
1.3.2003975	• 修复了影响 Packer AMI 生成的问题（SysprepInstance.ps1 将 $LastErrorCode 返回为 1）。	2022 年 12 月 24 日

版本	详细信息	发行日期
1.3.2003961	• 修复了在快速启动的实例上明确指定的管理员密码被随机密码覆盖的问题。 • 修复了 SSM Agent 无法在较小的实例类型上启动的问题。 • 修复了实例控制台日志包含 RDPCERTIFICATE-THUMBPRINT: 00000000000000000000000000 而不是有效的 RDP 证书指纹值的问题。	2022 年 12 月 6 日
1.3.2003923	• 修复了 PnPDeviceID 为空时查找网络适配器的逻辑。	2022 年 11 月 9 日
1.3.2003919	• 更新了 Get-ConsolePort 以使用 PCI 分段信息。 • 修复了重启后可以选择不正确的网络适配器的问题。 • 修复了 start-SSM-Agent 超时逻辑。 • 修复了 Send-AdminCredentials 函数别名的向后兼容性。	2022 年 11 月 8 日
1.3.2003857	• 选中主网络适配器时，为具有默认网关的适配器确定优先顺序。 • 扩展了内存密码加密。	2022 年 10 月 3 日
1.3.2003824	• 修复了 `setComputerName` 过程中出现的错误。 • 增加了在检测到 BYOL 账单代码时跳过 Windows 激活的逻辑。 • 增加了内存密码加密。 • 修复了 `m6id.4xlarge` 上卷初始化过程中出现的错误。	2022 年 8 月 30 日

版本	详细信息	发行日期
1.3.2003691	• 更新了的 IMDS 等待逻辑以仅发出 IMDSv2 请求。 • 修复了影响 eGPU 安装的错误。	2022 年 6 月 21 日
1.3.2003639	• 已添加网络适配器等待逻辑以防止在初始化之前使用。 • 修复了一些小问题。	2022 年 5 月 10 日
1.3.2003498	• 添加了遥测。 • 向设置 UI 添加了快捷方式。 • 格式化了 PowerShell 脚本。 • 修复了在 BeforeSysprep.cmd 完成之前发生的关机问题。	2022 年 1 月 31 日
1.3.2003411	更改的密码生成逻辑，以排除复杂性较低的密码。	2021 年 8 月 4 日
1.3.2003364	更新了 Install-EgpuManager 的 IMDSv2 支持。	2021 年 6 月 7 日
1.3.2003312	• 添加了 `setMonitorAlwaysOn` 设置前后的日志行。 • 将 Amazon Nitro Enclaves 程序包版本添加到了控制台日志中。	2021 年 5 月 4 日
1.3.2003284	通过将用于存储用户数据的位置更新为 `LocalAppData` 改进了权限模型。	2021 年 3 月 23 日

版本	详细信息	发行日期
1.3.2003236	• 更新了在 Set-AdminAccount 和 Randomize-LocalAdminPassword 中设置用户密码的方法。 • 修复了 InitializeDisks 以便在将磁盘设置为可写之前检查磁盘是否设置为只读。	2021 年 2 月 11 日
1.3.2003210	install.ps1 的本地化修复。	2021 年 1 月 7 日
1.3.2003205	用于更新 install.ps1 目录权限的 %ProgramData%Amazon EC2-WindowsLaunchModuleScripts 安全修复程序。	2020 年 12 月 28 日
1.3.2003189	添加路线后添加 w32tm resync。	2020 年 12 月 4 日
1.3.2003155	更新了实例类型信息。	2020 年 8 月 25 日
1.3.2003150	已将 OsCurrentBuild 和 OsReleaseId 添加到控制台输出。	2020 年 4 月 22 日
1.3.2003040	修复了 IMDS 版本 1 回退逻辑。	2020 年 4 月 7 日
1.3.2002730	添加了对 IMDS V2 的支持。	2020 年 3 月 3 日
1.3.2002240	修复了一些小问题。	2019 年 10 月 31 日
1.3.2001660	修复了首次执行 Sysprep 后用户不使用密码自动登录的问题。	2019 年 7 月 2 日
1.3.2001360	修复了一些小问题。	2019 年 3 月 27 日

版本	详细信息	发行日期
1.3.2001220	所有 PowerShell 脚本已签名。	2019 年 2 月 28 日
1.3.2001200	修复了其中在 Windows Server 故障失效转移集群中的节点上运行该脚本的 InitializeDisks.ps1 将在其盘符与本地盘符匹配的远程节点上格式化驱动器的问题。	2019 年 2 月 27 日
1.3.2001160	修复了 Windows 2019 中缺失壁纸的问题。	2019 年 2 月 22 日
1.3.2001040	• 增加了用于将监视器设置为永不关闭以修复 ACPI 问题的插件。 • 将 SQL Server 版本写入到了控制台中。	2019 年 1 月 21 日
1.3.2000930	修复了在启用了 ipv6 的 ENI 上向元数据添加路由的问题。	2019 年 1 月 2 日
1.3.2000760	• 添加了 RSS 的默认配置以及 ENA 设备的接收队列设置。 • Sysprep 期间禁用休眠。	2018 年 12 月 5 日
1.3.2000630	• 添加了 DNS 服务器的路由 169.254.169.253/32。 • 添加了设置管理用户筛选条件。 • 对实例休眠进行了改进。 • 添加了计划 EC2Launch 在每次启动时运行的选项。	2018 年 11 月 9 日

版本	详细信息	发行日期
1.3.2000430.0	• 将路由 169.254.169.123/32 添加到 AMZN 时间服务。 • 将路由 169.254.169.249/32 添加到 GRID 许可证服务。 • 添加了尝试启动 Systems Manager 时的 25 秒超时。	2018 年 9 月 19 日
1.3.200039.0	• 修复了 EBS NVME 卷不正确的盘符。 • 添加 NVME 驱动程序版本的其他日志记录。	2018 年 8 月 15 日
1.3.2000080	修复了一些小问题。	
1.3.610	修复了将输出和错误从用户数据重定向至文件的问题。	
1.3.590	• 在墙纸中增加了缺失的实例类型。 • 修复了盘符映射和磁盘安装的问题。	
1.3.580	• 修复了 `Get-Metadata` 以对 Web 请求使用默认系统代理设置。 • 在磁盘初始化中添加了针对 NVMe 的特殊案例。 • 修复了一些小问题。	
1.3.550	添加了 `-NoShutdown` 选项以在不关机的情况下启用 Sysprep。	
1.3.540	修复了一些小问题。	
1.3.530	修复了一些小问题。	
1.3.521	修复了一些小问题。	

版本	详细信息	发行日期
1.3.0	• 修复了更改计算机名称时的十六进制长度问题。 • 修复了更改计算机名称时可能出现循环重启的问题。 • 修复了壁纸设置中的问题。	
1.2.0	• 更新以在 EC2 系统日志中显示有关已安装操作系统 (OS) 的信息。 • 更新以在 EC2 系统日志中显示 EC2Launch 和 SSM Agent 版本。 • 修复了一些小问题。	
1.1.2	• 更新以在 EC2 系统日志中显示 ENA 驱动程序信息。 • 更新以从主 NIC 筛选条件逻辑中排除 Hyper-V。 • 向注册表项中添加了 Amazon KMS 服务器和端口以进行 KMS 激活。 • 改进了多用户的壁纸设置。 • 更新以从持久存储清除路由。 • 更新以从 DNS 后缀列表中的可用区删除 z。 • 更新以解决用户数据中有关 <runAsLocalSystem> 标记的问题。	
1.1.1	初始版本。	

使用 EC2Config 服务在 EC2 旧版 Windows 操作系统实例启动期间执行任务

> **Note**
>
> 对 EC2Config 的支持已终止。Microsoft 不再支持其运行的操作系统版本。强烈建议升级到最新的启动代理。
>
> Windows Server 2022 及更高版本操作系统的最新启动代理是 EC2Launch v2，它取代了 EC2Config 和 EC2Launch，并预装在 Amazon Windows Server 2022 和 2025 AMI 上。您也可以在 Windows Server 2016 和 2019 上手动安装和配置代理。有关更多信息，请参阅 安装 EC2Launch v2。

适用于 Windows Server 2016 之前的 Windows Server 版本的 Windows AMI 包括一项可选服务，即 EC2Config 服务 (`EC2Config.exe`)。EC2Config 在实例启动时启动，并且在启动期间以及您每次停止或启动实例时执行相关任务。EC2Config 还可按需执行任务。部分任务可自动启用，而其他任务必须手动启用。尽管此服务是可选的，但可让您访问否则无法访问的高级功能。此服务可使用 LocalSystem 账户运行。

EC2Config 服务会运行 Sysprep，借助该 Microsoft 工具您可以创建可重复使用的自定义 Windows AMI。在 EC2Config 调用 Sysprep 时，会使用 `%ProgramFiles%\Amazon\EC2ConfigService\Settings` 中的设置文件来确定要执行的操作。您可以使用 EC2 服务属性系统对话框间接编辑这些文件，也可以使用 XML 编辑器或文本编辑器直接进行编辑。然而，有些高级设置并未包含在 Ec2 服务属性系统对话框中，因此您必须直接编辑这些条目。

如果您在更新某实例的设置之后从中创建了一个 AMI，则新设置会应用到所有从新 AMI 启动的实例。有关创建 AMI 的信息，请参阅创建 Amazon EBS-backed AMI。

EC2Config 会使用设置文件来控制它的操作。您可以通过图形工具或直接编辑 XML 文件来更新这些设置文件。服务二进制文件和其他文件均包含在 `%ProgramFiles%\Amazon\EC2ConfigService` 目录中。

内容

- EC2Config 和 Amazon Systems Manager
- EC2Config 任务
- EC2Config 设置文件
- 安装最新版的 EC2Config
- 配置 EC2Config 服务的 .NET 代理设置

- 在 EC2 Windows 实例上通过系统对话框设置 EC2Config 服务属性
- 排查 EC2Config 启动代理的问题
- EC2Config 版本历史记录

EC2Config 和 Amazon Systems Manager

对于在 2016 年 11 月之前发布的早于 Windows Server 2016 的 Windows Server 版本，EC2Config 服务可以处理从包含这些版本的 AMI 所创建实例上的 Systems Manager 请求。

对于在 2016 年 11 月之后发布的早于 Windows Server 2016 的 Windows Server 版本，从包含这些版本的 AMI 创建的实例包含 EC2Config 服务和 SSM Agent。EC2Config 执行前面所述的所有任务，而 SSM Agent 处理 Run Command 和 State Manager 这样的 Systems Manager 功能的请求。

您可以使用 Run Command 更新现有实例，以便使用最新版本的 EC2Config 服务和 SSM Agent。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的使用 Run Command 更新 SSM Agent。

EC2Config 任务

当系统首次启动实例后又将其禁用时，EC2Config 会运行初始启动任务。要再次运行这些任务，您必须在关闭实例之前明确启用它们，或者手动运行 Sysprep。这些任务如下所示：

- 设置管理员账户的随机加密密码。
- 生成和安装远程桌面连接使用的主机证书。
- 请动态扩展操作系统分区，以包含所有未分区的空间。
- 执行指定的用户数据 (和 Cloud-Init，如果已安装)。有关指定用户数据的更多信息，请参阅在启动包含用户数据输入的 EC2 实例时运行命令。

每次实例启动时，EC2Config 会执行以下任务：

- 更改主机名称以与十六进制表示法的私有 IP 地址匹配 (此任务在默认情况下处于禁用状态，必须启用才能在实例启动时运行)。
- 配置密钥管理服务器 (Amazon KMS)，检查 Windows 激活状态并根据需要激活 Windows。
- 挂载所有 Amazon EBS 卷和实例存储卷，并且将卷名称映射到驱动器盘符。
- 将事件日志条目写入控制台以帮助排查问题 (此任务在默认情况下处于禁用状态，必须进行启用才能在实例启动时运行)。

- 将 Windows 已准备就绪写入控制台。
- 在附加单个或多个 NIC 的情况下，向主网络适配器添加自定义路由以启用以下 IP 地址：169.254.169.250、169.254.169.251 和 169.254.169.254。当您访问实例元数据时，Windows 激活会使用这些地址。

> **Note**
>
> 如果 Windows 操作系统配置为使用 IPv4，则可以使用这些 IPv4 链路本地地址。如果 Windows 操作系统禁用了 IPv4 网络协议堆栈并改为使用 IPv6，则请添加 [fd00:ec2::250] 来代替 169.254.169.250 和 169.254.169.251。然后添加 [fd00:ec2::254] 来代替 169.254.169.254。

每次用户登录时，EC2Config 都会执行以下任务：

- 在桌面背景中显示墙纸信息。

实例运行时，您可以请求 EC2Config 按需执行以下任务：

- 运行 Sysprep 并关闭实例，以便您可以从实例中创建一个 AMI。有关更多信息，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。

EC2Config 设置文件

设置文件会控制 EC2Config 服务的操作。这些文件位于 C:\Program Files\Amazon\Ec2ConfigService\Settings 目录下：

- ActivationSettings.xml：使用密钥管理服务器 (Amazon KMS) 控制产品激活。
- AWS.EC2.Windows.CloudWatch.json — 控制要发送到 CloudWatch 的性能计数器以及要发送到 CloudWatch Logs 的日志。
- BundleConfig.xml — 控制 EC2Config 为 AMI 创建准备实例存储支持的实例的方式。
- Config.xml — 控制基本设置。
- DriveLetterConfig.xml — 控制盘符映射。
- EventLogConfig.xml — 控制实例启动时控制台上显示的事件日志信息。
- WallpaperSettings.xml — 控制桌面背景上显示的信息。

ActivationSettings.xml

此文件包含用于控制产品激活的设置。当 Windows 启动时，EC2Config 服务会检查是否已激活 Windows。如果尚未激活 Windows，它会通过搜索指定的 Amazon KMS 服务器激活 Windows。

- `SetAutodiscover` – 指示是否自动检测 Amazon KMS。
- `TargetKMSServer`：存储 Amazon KMS 的私有 IP 地址。Amazon KMS 必须位于您的实例所在的同一区域。
- `DiscoverFromZone` – 发现指定 DNS 区域中的 Amazon KMS 服务器。
- `ReadFromUserData` – 获取 UserData 中的 Amazon KMS 服务器。
- `LegacySearchZones` – 发现指定 DNS 区域中的 Amazon KMS 服务器。
- `DoActivate` — 尝试使用本部分中的指定设置进行激活。该值可以是 `true` 或 `false`。
- `LogResultToConsole` — 在控制台中显示结果。

BundleConfig.xml

此文件包含用于控制 EC2Config 为 AMI 创建过程准备实例的方式的设置。

- `AutoSysprep`—指明是否自动使用 Sysprep。将值更改为 `Yes` 可使用 Sysprep。
- `SetRDPCertificate` — 为远程桌面服务器设置自签名证书。这样一来，您可以将 RDP 安全应用到实例中。如果新实例应该具有该证书，请将该值改为 `Yes`。

  此设置不用于操作系统版本早于 Windows Server 2016 的实例，因为它们可以生成自己的证书。
- `SetPasswordAfterSysprep`—在新启动的实例上设置随机密码，使用用户启动密钥对其加密并且将该加密密码输出到控制台中。如果新实例不应设置为随机加密密码，请将该设置的值改为 No。

Config.xml

插件

- `Ec2SetPassword` — 在您每次启动实例时生成随机加密密码。默认情况下，首次启动之后该功能会被禁用，因此重新启动该实例并不会更改用户设置的密码。每次您启动实例时，请将该设置改为 `Enabled`，以继续生成密码。

  如果您计划从自己的实例创建一个 AMI，则该设置非常重要。
- `Ec2SetComputerName` — 根据实例的 IP 地址将实例的主机名设置为唯一名称，并且重新启动实例。若要设置您自己的主机名或防止修改您的现有主机名，必须禁用此设置。

- Ec2InitializeDrives — 在启动期间初始化和格式化所有卷。该功能已默认启用。
- Ec2EventLog — 在控制台中显示事件日志条目。默认情况下，会显示来自系统事件日志的最近三个错误条目。要指定将显示的事件日志条目，请编辑 EventLogConfig.xml 目录中的 EC2ConfigService\Settings 文件。有关此文件中设置的信息，请参阅事件日志键。
- Ec2ConfigureRDP — 在实例上设置自签名证书，以便用户可以使用远程桌面安全访问实例。此设置不用于操作系统版本早于 Windows Server 2016 的实例，因为它们可以生成自己的证书。
- Ec2OutputRDPCert — 在控制台中显示远程桌面证书信息，以便用户可以根据指纹对其进行验证。
- Ec2SetDriveLetter — 根据用户定义的设置，设置已挂载卷的盘符。默认情况下，Amazon EBS 卷附加到实例之后，系统会根据实例上的盘符进行挂载。要指定您的盘符映射，请编辑 DriveLetterConfig.xml 目录中的 EC2ConfigService\Settings 文件。
- Ec2WindowsActivate — 该插件处理 Windows 激活。它检查 Windows 是否已激活。如果未激活，则它会更新 Amazon KMS 客户端设置，然后激活 Windows。

  要修改 Amazon KMS 设置，请编辑 ActivationSettings.xml 目录中的 EC2ConfigService\Settings 文件。
- Ec2DynamicBootVolumeSize — 扩展磁盘 0/卷 0 以包含所有未分区的空间。
- Ec2HandleUserData—创建脚本并在运行 Sysprep 后首次启动实例时运行用户创建的脚本。脚本标签包含的命令会保存到批处理文件中，而 PowerShell 标签包含的命令会保存到 .ps1 文件中（对应于“Ec2 服务属性”系统对话框中的“用户数据”复选框）。
- Ec2ElasticGpuSetup — 如果实例与 Elastic GPU 关联，请安装 Elastic GPU 软件包。
- Ec2FeatureLogging — 将 Windows 功能安装和相应服务状态发送到控制台。仅对 Microsoft Hyper-V 功能和相应 vmms 服务提供支持。

全局设置

- ManageShutdown — 确保从 Amazon S3 支持的 AMI 启动的实例不会在运行 Sysprep 时终止。
- SetDnsSuffixList — 为 Amazon EC2 设置网络适配器的 DNS 后缀。这允许 Amazon EC2 中运行的服务器的 DNS 解析，无需提供完全限定域名。

> **Note**
>
> 这会为以下域添加 DNS 后缀查找，并配置其他标准后缀。有关启动代理如何设置 DNS 后缀的更多信息，请参阅 为 EC2 Windows 启动代理配置 DNS 后缀。

```
region.ec2-utilities.amazonaws.com
```

- `WaitForMetaDataAvailable` — 确保 EC2Config 服务会在继续启动之前等待元数据处于可访问状态并且网络可用。此检查确保了 EC2Config 可以从激活和其他插件的元数据获取信息。
- `ShouldAddRoutes` — 在附加多个 NIC 的情况下，向主网络适配器添加自定义路由以启用以下 IP 地址：169.254.169.250、169.254.169.251 和 169.254.169.254。当您访问实例元数据时，Windows 激活会使用这些地址。
- `RemoveCredentialsfromSyspreponStartup` — 在下一次启动服务时从 `Sysprep.xml` 中删除管理员密码。要确保该密码持久保留，请编辑该设置。

DriveLetterConfig.xml

此文件包含用于控制盘符映射的设置。默认情况下，一个卷可映射到任何可用的盘符。您可按以下方式将卷挂载为特定的盘符。

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- `VolumeName` — 卷标签。例如：*`My Volume`*。要为实例存储卷指定映射，请使用标签 `Temporary Storage X`，其中 `X` 为 0 到 25 之间的数字。
- `DriveLetter` — 盘符。例如：*`M:`*。如果盘符已在使用，则映射会失败。

EventLogConfig.xml

此文件包含用于控制实例启动时显示在控制台上的事件日志信息的设置。默认情况下，我们会显示来自系统事件日志的最近三个错误条目。

- Category — 监视器的事件日志密钥。
- ErrorType — 事件类型（例如 、Error、Warning）Information。
- NumEntries — 此类别中所存储事件的数目。
- LastMessageTime — 为防止重复推送相同的消息，服务会在每次推送消息时更新此值。
- AppName — 事件源或记录事件的应用程序。

WallpaperSettings.xml

此文件包含用于控制在桌面背景上显示的信息的设置。默认情况下显示以下信息。

- Hostname — 显示计算机名称。
- Instance ID — 显示实例的 ID。
- Public IP Address — 显示实例的公有 IP 地址。
- Private IP Address — 显示实例的私有 IP 地址。
- Availability Zone — 显示实例在其中运行的可用区。
- Instance Size — 显示实例的类型。
- Architecture — 显示 PROCESSOR_ARCHITECTURE 环境变量的设置。

您可以通过删除条目删除默认显示的任何信息。您可以按如下所示添加其他要显示的实例元数据。

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

您可以按如下所示添加其他要显示的系统环境变量。

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

此文件包含控制 EC2Config 如何初始化驱动器的设置。

默认情况下，EC2Config 用操作系统初始化未联机的驱动器。您可以自定义插件，如下所示。

```
<InitializeDrivesSettings>
    <SettingsGroup>setting</SettingsGroup>
</InitializeDrivesSettings>
```

使用设置组指定要如何初始化驱动器：

FormatWithTRIM

在驱动器格式化时启用 TRIM 命令。在驱动器格式化和初始化后，系统还原 TRIM 配置。

从 EC2Config 版本 3.18 开始，在磁盘格式化操作过程中默认禁用 TRIM 命令。这将缩短格式化时间。对于 EC2Config 版本 3.18 及更高版本，在磁盘格式化操作期间使用此设置启用 TRIM。

FormatWithoutTRIM

在格式化驱动器时禁用 TRIM 命令并在 Windows 中改进格式化时间。在驱动器格式化和初始化后，系统还原 TRIM 配置。

DisableInitializeDrives

对新驱动器禁用格式化。使用此设置可手动初始化驱动器。

安装最新版的 EC2Config

> **Note**
>
> Windows Server 2022 及更高版本操作系统的最新启动代理是 EC2Launch v2，其取代了 EC2Config 和 EC2Launch。EC2Launch v2 预装在 Amazon Windows Server 2022 和 2025 AMI 中。您也可以在 Windows Server 2016 和 2019 上手动安装和配置代理。有关更多信息，请参阅 安装 EC2Launch v2。

有关如何接收 EC2Config 更新通知的信息，请参阅订阅 EC2 Windows 启动代理通知。有关每个版本中的更改的信息，请参阅EC2Config 版本历史记录。

开始前的准备工作

- 确认您有 .NET Framework 3.5 SP1 或更高版本。
- 默认情况下，安装程序会在安装期间用默认设置文件替换您的设置文件，并在安装完成时重启 EC2Config 服务。如果您更改了 EC2Config 服务设置，请从 `config.xml` 目录中复制 `%Program`

Files%\Amazon\Ec2ConfigService\Settings 文件。在更新 EC2Config 服务后，可以还原此文件来保留您的配置更改。

验证 EC2Config 版本

使用以下过程验证您的实例上安装的 EC2Config 版本。

验证安装的 EC2Config 版本

1. 从 AMI 启动实例并连接到该实例。
2. 在控制面板中，选择 Programs and Features (程序和功能)。
3. 在已安装程序的列表中查找 Ec2ConfigService。其版本号会显示在 Version (版本) 列中。

配置 EC2Config 服务的 .NET 代理设置

您可以使用下列方法之一将 EC2Config 服务配置为通过代理进行通信：适用于 .NET 的 Amazon 开发工具包、system.net 元素或 Microsoft 组策略和 Internet Explorer。使用 Amazon SDK for .NET 时可以指定登录凭证，因此，请优先使用此方法。

Methods

- 使用 适用于 .NET 的 Amazon SDK 配置代理设置（首选）
- 使用 system.net 元素配置代理设置
- 使用 Microsoft 组策略及 Microsoft Internet Explorer 配置代理设置

使用 适用于 .NET 的 Amazon SDK 配置代理设置（首选）

您可以通过在 proxy 文件中指定 Ec2Config.exe.config 元素来配置 EC2Config 服务的代理设置。有关更多信息，请参阅适用于 .NET 的 Amazon 开发工具包的配置文件参考。

在 Ec2Config.exe.config 文件中指定代理元素

1. 在需要 EC2Config 服务通过代理进行通信的实例上编辑 Ec2Config.exe.config 文件。默认情况下，该文件位于以下目录中：%ProgramFiles%\Amazon\Ec2ConfigService。
2. 将以下 aws 元素添加到 configSections。不要将此添加到任何现有 sectionGroups 中。

   适用于 EC2Config 3.17 版本或更早版本

   ```
   <configSections>
   ```

```
   <section name="aws" type="Amazon.AWSSection, AWSSDK"/>
</configSections>
```

适用于 EC2Config 3.18 版本或更新版本

```
<configSections>
     <section name="aws" type="Amazon.AWSSection, AWSSDK.Core"/>
</configSections>
```

3. 在 aws 文件中添加下面的 Ec2Config.exe.config 元素。

```
<aws>
   <proxy
     host="string value"
     port="string value"
     username="string value"
     password="string value" />
</aws>
```

4. 保存您的更改。

使用 system.net 元素配置代理设置

您可以在 system.net 文件的 Ec2Config.exe.config 元素中指定代理设置。有关更多信息，请参阅 defaultProxy 元素（网络设置）。

在 Ec2Config.exe.config 文件中指定 system.net 元素

1. 在需要 EC2Config 服务通过代理进行通信的实例上编辑 Ec2Config.exe.config 文件。默认情况下，该文件位于以下目录中：%ProgramFiles%\Amazon\Ec2ConfigService。
2. 将 defaultProxy 条目添加 system.net 中。有关更多信息，请参阅 defaultProxy 元素（网络设置）。

   例如，下面的配置将路由所有流量并使用当前为 Internet Explorer 配置的代理，但元数据和授权流量除外 (这两者将绕过该代理)。

```
<defaultProxy>
    <proxy usesystemdefault="true" />
    <bypasslist>
        <add address="169.254.169.250" />
        <add address="169.254.169.251" />
```

```
        <add address="169.254.169.254" />
        <add address="[fd00:ec2::250]" />
        <add address="[fd00:ec2::254]" />
    </bypasslist>
</defaultProxy>
```

3. 保存您的更改。

使用 Microsoft 组策略及 Microsoft Internet Explorer 配置代理设置

EC2Config 服务以本地系统用户账户运行。更改实例上的组策略设置后，您可以在 Internet Explorer 中为此账户指定实例级代理设置。

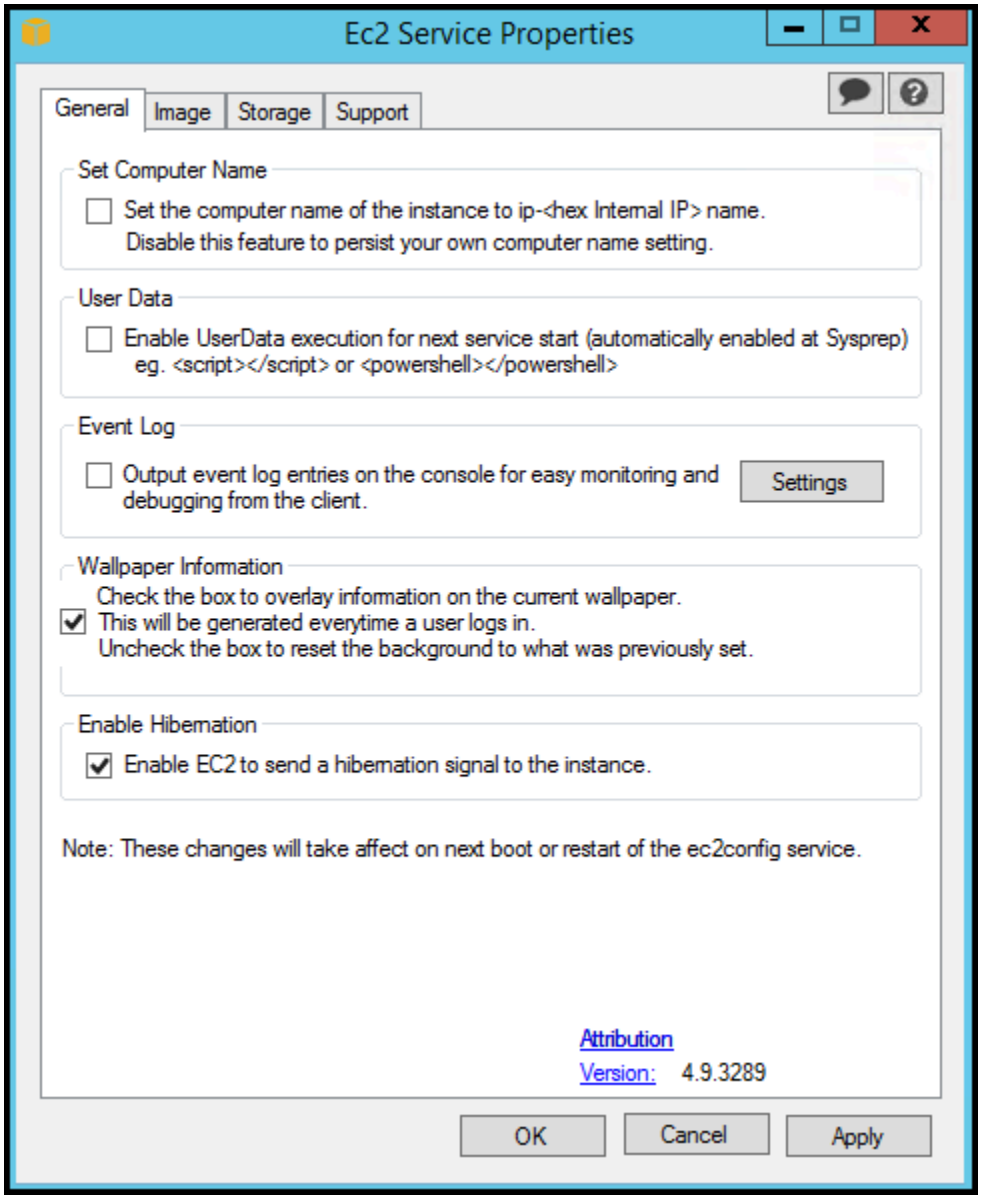
使用组策略及 Internet Explorer 配置代理设置

1. 在需要 EC2Config 服务通过代理进行通信的实例上，以管理员身份打开命令提示符，键入 **`gpedit.msc`**，然后按 Enter。
2. 在本地组策略编辑器中，依次选择本地计算机策略下的计算机配置、管理模板、Windows 组件、Internet Explorer。
3. 在右侧窗格中，选择按计算机进行代理服务器设置(不是按用户)，然后选择编辑策略设置。
4. 选择启用，然后选择应用。
5. 打开 Internet Explorer，然后选择工具按钮。
6. 选择 Internet 选项，然后选择连接选项卡。
7. 选择局域网设置。
8. 在代理服务器下，选择为 LAN 使用代理服务器选项。
9. 指定地址和端口信息，然后选择确定。

在 EC2 Windows 实例上通过系统对话框设置 EC2Config 服务属性

以下过程介绍如何使用 EC2 服务属性系统对话框来启用或禁用设置。

1. 启动并连接到您的 Windows 实例。
2. 在 Start (开始) 菜单中，单击 All Programs (所有程序)，然后单击 EC2ConfigService Settings (EC2ConfigService 设置)。



3. 在 EC2 服务属性系统对话框的常规选项卡上，您可以启用或禁用以下设置。

   Set Computer Name（设置电脑名称）

   如果此设置已启用 (默认情况下已启用)，则会在每次启动时将主机名与当前内部 IP 地址进行比较；如果主机名和内部 IP 地址不匹配，则重置主机名以包含内部 IP 地址，随后系统重启以接受新主机名。若要设置您自己的主机名或防止修改现有主机名，请不要启用该设置。

   用户数据

   利用用户数据执行，您可以在实例元数据中指定脚本。默认情况下，这些脚本在初次启动期间运行。您还可以配置它们在下次重新引导或启动实例时运行，或者在每次重新引导或启动实例时运行。

如果您有很大的脚本，我们建议您使用用户数据来下载该脚本，然后再运行它。

有关更多信息，请参阅用户数据执行。

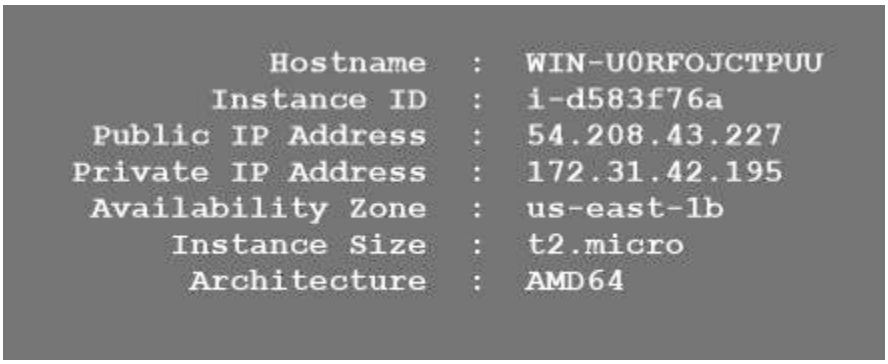
Event Log (事件日志)

使用此设置可在启动期间在控制台上显示事件日志条目，以便轻松监控和调试。

单击 Settings (设置)，为发送到控制台的日志条目指定筛选条件。默认筛选条件会将来自系统事件日志的最近三个错误条目发送到控制台。

Wallpaper Information (壁纸信息)

使用此设置可在桌面背景中显示系统信息。以下是桌面背景上所显示信息的示例。

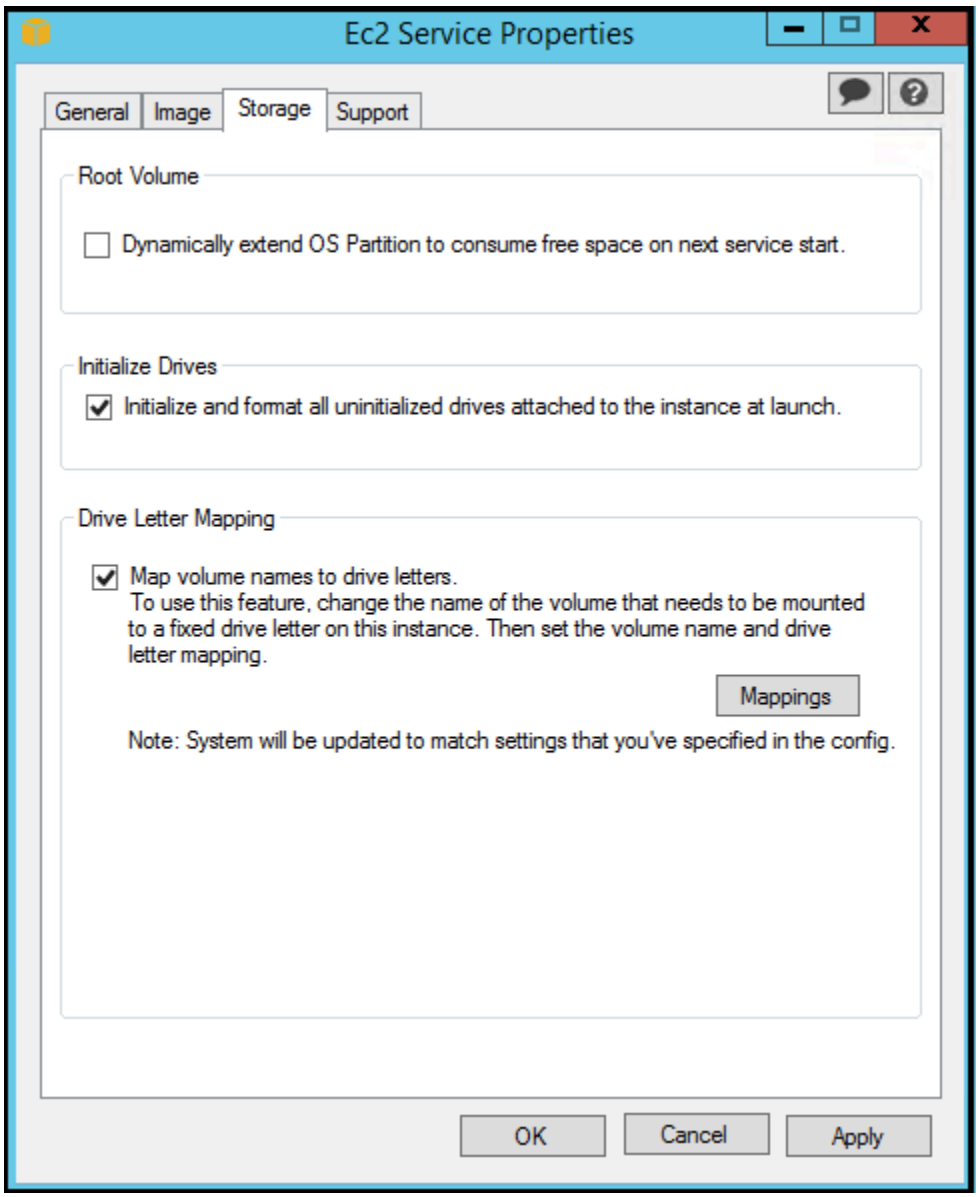


桌面背景上显示的信息受设置文件 `EC2ConfigService\Settings\WallpaperSettings.xml` 控制。

Enable Hibernation (启用休眠)

使用此设置可允许 EC2 向操作系统发出信号来执行休眠。

4. 单击 Storage 标签。您可以启用或禁用以下设置。



Root Volume

此设置可将磁盘 0/卷 0 动态扩展以包含所有未分区的空间。从具有自定义大小的根卷启动实例时，此设置非常有用。

Initialize Drives

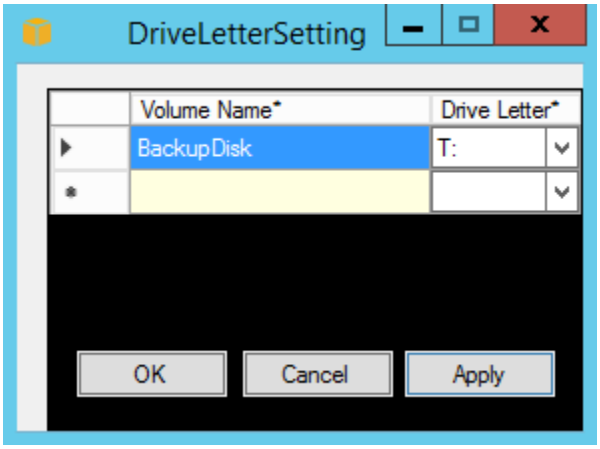
此设置可格式化并附加在启动期间附加到实例的所有卷。

Drive Letter Mapping (盘符映射)

系统会将附加到实例的卷映射到盘符。对于 Amazon EBS 卷，默认以从 D: 到 Z: 的顺序分配驱动器盘符。对于实例存储卷，默认设置则取决于驱动程序。AmazonPV 驱动程序和 Citrix PV

驱动程序以从 Z: 到 A: 的顺序分配实例存储卷驱动器盘符。Red Hat 驱动程序按 D: 到 Z: 的顺序分配实例存储卷盘符。

要为您的卷选择盘符，请单击 Mappings (映射)。在 DriveLetterSetting 对话框中，为每个卷指定 Volume Name (卷名称) 和 Drive Letter (驱动器盘符) 值，单击 Apply (应用)，然后单击 OK (确定)。我们建议您选择诸如字母表中间的字母作为盘符，以避免与可能在使用的盘符发生冲突。



在指定盘符映射并附加与您指定的某个卷名称具有相同标签的卷之后，EC2Config 会自动为该卷分配您指定的盘符。然而，如果盘符已在使用，则盘符映射会失败。请注意，在您指定盘符映射之后，EC2Config 不会更改已挂载卷的盘符。

5. 要保存设置并在稍后继续对其进行处理，请单击确定以关闭 EC2 服务属性系统对话框。如果您已完成实例自定义并要从该实例创建 AMI，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。

排查 EC2Config 启动代理的问题

以下信息可帮助您解决与 EC2Config 服务相关的问题。

更新无法访问的实例上的 EC2Config

通过执行以下过程，使用远程桌面更新无法访问的 Windows Server 实例上的 EC2Config 服务。

更新您无法连接到的 Amazon EBS 支持的 Windows 实例上的 EC2Config

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 找到受影响的实例。选择实例并选择 Instance state (实例状态)，然后选择 Stop instance (停止实例)。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

4. 选择 Launch instances (启动实例) 并在受影响实例所在的同一可用区中创建一个临时 `t2.micro` 实例。使用与用于启动受影响实例的 AMI 不同的 AMI。

> **Important**
>
> 如果您未在与受影响实例相同的可用区中创建该实例，则无法将受影响的实例的根卷附加到新实例。

5. 在 EC2 控制台中，选择 Volumes。
6. 找到受影响的实例的根卷。分离卷并向之前创建的临时实例附加卷。使用默认设备名称 (xvdf) 附加该卷。
7. 使用远程桌面连接临时实例，然后通过磁盘管理实用工具使该卷可供使用。
8.
9. 在临时实例上，打开“运行”对话框，键入 **`regedit`**，然后按 Enter。
10. 选择 HKEY_LOCAL_MACHINE。从 File 菜单中，选择 Load Hive（加载 Hive）。选择驱动器，然后导航至以下文件并打开：`Windows\System32\config\SOFTWARE`。当系统提示时，请指定密钥名称。
11. 选择刚加载的键并导航至 `Microsoft\Windows\CurrentVersion`。选择 `RunOnce` 密钥。如果此密钥不存在，则从上下文 (右键单击) 菜单中选择 `CurrentVersion`，然后选择 New 和 Key。为密钥 `RunOnce` 命名。
12. 从上下文（右键单击）菜单中，依次选择 `RunOnce` 密钥、New 和 String Value。输入 `Ec2Install` 作为名称并输入 `C:\Temp\Ec2Install.exe /quiet` 作为数据。
13. 选择 HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT \CurrentVersion\Winlogon 密钥。从上下文（右键单击）菜单中选择 New，然后选择 String Value。输入 **`AutoAdminLogon`** 作为名称并输入 **`1`** 作为值数据。
14. 选择 HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT \CurrentVersion\Winlogon> 密钥。从上下文（右键单击）菜单中选择 New，然后选择 String Value。输入 **`DefaultUserName`** 作为名称并输入 **`Administrator`** 作为值数据。

15. 选择 HKEY_LOCAL_MACHINE*specified key name*\Microsoft\Windows NT\CurrentVersion\Winlogon 密钥。从上下文（右键单击）菜单中选择 New，然后选择 String Value。键入 **DefaultPassword** 作为名称，然后在值数据中输入密码。
16. 在注册表编辑器导航窗格中，选择在首次打开注册表编辑器时创建的临时项。
17. 从 File（文件）菜单中，选择 Unload Hive（卸载 Hive）。
18. 在磁盘管理实用工具中，选择以前附加的驱动器，打开上下文 (右键单击) 菜单，然后选择脱机。
19. 在 Amazon EC2 控制台中，将受影响的卷与临时实例分离，然后将其重新附加到具有设备名称 /dev/sda1 的实例。您必须指定此设备名称才能将卷指派为根卷。
20. 启动和停止 Amazon EC2 实例实例。
21. 在实例启动后，检查系统日志并确认看到消息 Windows is ready to use。
22. 打开注册表编辑器并选择 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon。删除您之前创建的字符串值密钥：AutoAdminLogon、DefaultUserName 和 DefaultPassword。
23. 删除或停止您在本过程中创建的临时实例。

EC2Config 版本历史记录

下表说明已发行的 EC2Config 版本。有关 SSM Agent 更新的信息，请参阅 Systems Manager SSM Agent 发布说明。

> **Important**
>
> 对 EC2Config 的支持已终止。仅最新版本的 EC2Config 代理可供下载。之前的版本将标记为私有。

版本	详细信息	发行日期
4.9.5777	• 修复了某些实例类型的 RSS 配置设置不正确的问题。 • SSM Agent 3.3.484.0 的新版本。	2024 年 6 月 17 日
4.9.5554	• 基于以下注册表条目限制域名传递：HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\D	2023 年 10 月 4 日

版本	详细信息	发行日期
	nscache\Parameters\DomainNameDevolutionLevel 。 • SSM Agent 3.2.1630.0 的新版本。	
4.9.5467	• 添加了用于发现控制台端口的重试功能。 • SSM Agent 3.1.2282.0 的新版本。	2023 年 8 月 1 日
4.9.5288	• 已将 Amazon Core SDK 更新为版本 3.7.103.23 。 • 修复了 AWS-UpdateEC2Config SSM 文档无法在仅启用 IMDSv2 的实例上更新 EC2Config 的问题。 • SSM Agent 3.1.2144.0 的新版本。	2023 年 3 月 8 日
4.9.5231	• SSM Agent 的新版本 3.1.1927.0。	2023 年 2 月 14 日
4.9.5103	• 修复了在 r5d 和 i4i 实例系列上错误识别临时卷的问题。 • SSM Agent 的新版本 3.1.1856.0。	2022 年 12 月 5 日
4.9.5064	• 更新为使用 PCI 分段信息选择控制台端口。 • 签署了 PowerShell 脚本并添加了版权标头。 • 修复了主网络适配器选择逻辑。 • SSM Agent 的新版本 3.1.1732.0。	2022 年 11 月 16 日
4.9.4588	• 更新了的 IMDS 等待逻辑以仅发出 IMDSv2 请求。 • 添加了 libec2launch.dll 启动代理共享库。 • SSM Agent 的新版本 3.1.1188.0。	2022 年 5 月 31 日

版本	详细信息	发行日期
4.9.4556	• 增加了等待逻辑，以确保在使用之前完成 NIC 的完全初始化。 • 新版本的 Log4Net 2.0.14.0 包含了安全补丁。 • 新版本的 SSM Agent 3.1.1045.0 包含了安全补丁。	2022 年 3 月 1 日
4.9.4536	• 修复了 Temp 文件夹丢失时 UserData 崩溃的问题。 • SSM Agent 的新版本 (3.1.804.0)。	2022 年 1 月 31 日
4.9.4508	• 修复了正确计算 diskpart 脚本路径的问题。 • SSM Agent 的新版本 (3.1.338.0)。	2021 年 10 月 6 日
4.9.4500	• 已更新具有 IMDS v2 支持的 `Install-EgpuManagerConfig` 。 • 已更新网站链接以使用 https。 • SSM Agent 的新版本 3.1.282.0	2021 年 9 月 7 日
4.9.4419	• 修复了 IMDS 版本 1 回退逻辑 • 将 Windows 临时目录的所有用法更新到 EC2Config 临时目录 • SSM Agent 的新版本 (3.0.1124.0)	2021 年 6 月 2 日
4.9.4381	• 在 EC2ConfigUpdater 中增加了对 SSM 文档架构版本 2.2 的支持 • 将 Amazon Nitro Enclaves 程序包版本添加到了控制台日志中。 • SSM Agent 的新版本 (3.0.529.0)	2021 年 5 月 4 日

版本	详细信息	发行日期
4.9.4326	• 已删除设置 UI 中的所有链接 • 这是最后一个支持 Windows Server 2008 的 EC2Config 版本。	2021 年 3 月 3 日
4.9.4279	• 修复了与 `Ec2ConfigMonitor` 计划任务相关的安全问题 • 修复了盘符映射问题和不正确的临时磁盘计数 • 已将 `OsCurrentBuild` 和 `OsReleaseId` 添加到控制台输出 • SSM Agent 的新版本 (2.3.871.0)	2020 年 12 月 11 日
4.9.4222	• 修复了 IMDS 版本 1 回退逻辑 • SSM Agent 的新版本 (2.3.842.0)	2020 年 4 月 7 日
4.9.4122	• 添加了对 IMDS v2 的支持 • SSM Agent 的新版本 (2.3.814.0)	2020 年 3 月 4 日
4.9.3865	• 解决了在裸机实例上检测到 Windows Server 2008 R2 的 COM 端口的问题 • SSM Agent 的新版本 (2.3.722.0)	2019 年 10 月 31 日
4.9.3519	• 新的 SSM Agent 版本 2.3.634.0	2019 年 6 月 18 日
4.9.3429	• 新的 SSM Agent 版本 2.3.542.0	2019 年 4 月 25 日
4.9.3289	• 新的 SSM Agent 版本 2.3.444.0	2019 年 2 月 11 日
4.9.3270	• 增加了用于将监视器设置为永不关闭以修复 ACPI 问题的插件 • 将 SQL Server 版本写入到了控制台中 • 新的 SSM Agent 版本 2.3.415.0	2019 年 1 月 22 日

版本	详细信息	发行日期
4.9.3230	• 更新了盘符映射说明以更好地符合相应功能 • 新的 SSM Agent 版本 2.3.372.0	2019 年 1 月 10 日
4.9.3160	• 增加了主 NIC 的等待时间 • 添加了 RSS 的默认配置以及 ENA 设备的接收队列设置 • Sysprep 期间禁用休眠 • 新的 SSM Agent 版本 2.3.344.0 • 将 Amazon 开发工具包升级到 3.3.29.13	2018 年 12 月 15 日
4.9.3067	• 对实例休眠进行了改进 • 新的 SSM Agent 版本 2.3.235.0	2018 年 11 月 8 日
4.9.3034	• 添加了 DNS 服务器的路由 169.254.169.253/32 • 新的 SSM Agent 版本 2.3.193.0	2018 年 10 月 24 日
4.9.2986	• 为所有 EC2Config 相关库添加了签名 • 新的 SSM Agent 版本 2.3.136.0	2018 年 10 月 11 日
4.9.2953	新的 SSM Agent 版本 (2.3.117.0)	2018 年 10 月 2 日
4.9.2926	新的 SSM Agent 版本 (2.3.68.0)	2018 年 9 月 18 日
4.9.2905	• 新的 SSM Agent 版本 (2.3.50.0) • 将路由 169.254.169.123/32 添加到 AMZN 时间服务 • 将路由 169.254.169.249/32 添加到 GRID 许可证服务 • 修复导致 EBS NVMe 卷被标记为短暂存储的问题	2018 年 9 月 17 日
4.9.2854	新的 SSM Agent 版本 (2.3.13.0)	2018 年 8 月 17 日
4.9.2831	新的 SSM Agent 版本 (2.2.916.0)	2018 年 8 月 7 日

版本	详细信息	发行日期
4.9.2818	新的 SSM Agent 版本 (2.2.902.0)	2018 年 7 月 31 日
4.9.2756	新的 SSM Agent 版本 (2.2.800.0)	2018 年 6 月 27 日
4.9.2688	新的 SSM Agent 版本 (2.2.607.0)	2018 年 5 月 25 日
4.9.2660	新的 SSM Agent 版本 (2.2.546.0)	2018 年 5 月 11 日
4.9.2644	新的 SSM Agent 版本 (2.2.493.0)	2018 年 4 月 26 日
4.9.2586	新的 SSM Agent 版本 (2.2.392.0)	2018 年 3 月 28 日
4.9.2565	• 新的 SSM Agent 版本 (2.2.355.0) • 修复了 M5 和 C5 实例的问题 (找不到半虚拟化驱动程序) • 添加有关实例类型、最新半虚拟化驱动程序和 NVMe 驱动程序的控制台日志记录	2018 年 3 月 13 日
4.9.2549	新的 SSM Agent 版本 (2.2.325.0)	2018 年 3 月 8 日
4.9.2461	新的 SSM Agent 版本 (2.2.257.0)	2018 年 2 月 15 日
4.9.2439	新的 SSM Agent 版本 (2.2.191.0)	2018 年 2 月 6 日
4.9.2400	新的 SSM Agent 版本 (2.2.160.0)	2018 年 1 月 16 日

版本	详细信息	发行日期
4.9.2327	• 新的 SSM Agent 版本 (2.2.120.0) • Amazon EC2 裸机实例上增加了 COM 端口发现 • Amazon EC2 裸机实例上增加了 Hyper-V 状态日志	2018 年 1 月 2 日
4.9.2294	新的 SSM Agent 版本 (2.2.103.0)	2017 年 12 月 4 日
4.9.2262	新的 SSM Agent 版本 (2.2.93.0)	2017 年 11 月 15 日
4.9.2246	新的 SSM Agent 版本 (2.2.82.0)	2017 年 11 月 11 日
4.9.2218	新的 SSM Agent 版本 (2.2.64.0)	2017 年 10 月 29 日
4.9.2212	新的 SSM Agent 版本 (2.2.58.0)	2017 年 10 月 23 日
4.9.2203	新的 SSM Agent 版本 (2.2.45.0)	2017 年 10 月 19 日
4.9.2188	新的 SSM Agent 版本 (2.2.30.0)	2017 年 10 月 10 日
4.9.2180	• 新的 SSM Agent 版本 (2.2.24.0) • 添加了适用于 GPU 实例的 Elastic GPU 插件	2017 年 10 月 5 日
4.9.2143	新的 SSM Agent 版本 (2.2.16.0)	2017 年 10 月 1 日
4.9.2140	新的 SSM Agent 版本 (2.1.10.0)	
4.9.2130	新的 SSM Agent 版本 (2.1.4.0)	
4.9.2106	新的 SSM Agent 版本 (2.0.952.0)	

版本	详细信息	发行日期
4.9.2061	新的 SSM Agent 版本 (2.0.922.0)	
4.9.2047	新的 SSM Agent 版本 (2.0.913.0)	
4.9.2031	新的 SSM Agent 版本 (2.0.902.0)	
4.9.2016	• 新的 SSM Agent 版本 (2.0.879.0) • 修复了 Windows Server 2003 的 CloudWatch Logs 目录路径	
4.9.1981	• 新的 SSM Agent 版本 (2.0.847.0) • 修复了与使用在 EBS 卷中生成的 `important.txt` 有关的问题。	
4.9.1964	新的 SSM Agent 版本 (2.0.842.0)	
4.9.1951	• 新的 SSM Agent 版本 (2.0.834.0) • 修复了不为临时驱动器从 Z: 映射驱动器盘符的问题。	
4.9.1925	• 新的 SSM Agent 版本 (2.0.822.0) • [错误] 此版本不是从 SSM Agent v4.9.1775 进行更新的有效目标。	
4.9.1900	新的 SSM Agent 版本 (2.0.805.0)	
4.9.1876	• 新的 SSM Agent 版本 (2.0.796.0) • 修复了与管理用户数据执行的输出/错误重定向有关的问题。	

版本	详细信息	发行日期
4.9.1863	• 新的 SSM Agent 版本 (2.0.790.0) • 修复了与将多个 EBS 卷附加到 Amazon EC2 实例有关的问题。 • 改进了 CloudWatch 以获取配置路径，保持向后兼容性。	
4.9.1791	新的 SSM Agent 版本 (2.0.767.0)	
4.9.1775	新的 SSM Agent 版本 (2.0.761.0)	
4.9.1752	新的 SSM Agent 版本 (2.0.755.0)	
4.9.1711	新的 SSM Agent 版本 (2.0.730.0)	
4.8.1676	新的 SSM Agent 版本 (2.0.716.0)	
4.7.1631	新的 SSM Agent 版本 (2.0.682.0)	
4.6.1579	• 新的 SSM Agent 版本 (2.0.672.0) • 修复了 v4.3、v4.4 和 v4.5 中的代理更新问题	
4.5.1534	新的 SSM Agent 版本 (2.0.645.1)	
4.4.1503	新的 SSM Agent 版本 (2.0.633.0)	
4.3.1472	新的 SSM Agent 版本 (2.0.617.1)	
4.2.1442	新的 SSM Agent 版本 (2.0.599.0)	
4.1.1378	新的 SSM Agent 版本 (2.0.558.0)	

版本	详细信息	发行日期
4.0.1343	• Run Command、State Manager、CloudWatch 代理以及域加入支持已转移到称为 SSM Agent 的另一个代理中。SSM Agent 将作为 EC2Config 升级的一部分进行安装。有关更多信息，请参阅EC2Config 和 Amazon Systems Manager。 • 如果您在 EC2Config 中设置了代理，则您需要更新 SSM Agent 的代理设置才可以升级。如果您不更新代理设置，则将无法使用 Run Command 来管理实例。要避免此情况出现，请先参阅以下信息，然后再更新到更新版本：Amazon Systems Manager 用户指南中的在 Windows 实例上安装和配置 SSM Agent。 • 如果您之前使用本地配置文件 (`AWS.EC2.Windows.CloudWatch.json`) 在实例上启用了 CloudWatch 集成，则您需要对文件进行配置，才能与 SSM Agent 配合使用。	
3.19.1153	• 已为具有旧 Amazon KMS 配置的实例重新启用激活插件。跳过 BYOL 用户的激活过程。 • 已更改在磁盘格式化操作期间要禁用的默认 TRIM 行为，并且增加了 FormatWithTRIM 来使用 UserData 覆盖 InitializeDisks 插件。	
3.18.1118	• 进行了修复，现在能够可靠地将路由添加到主网络适配器。 • 进行了更新来改善对 Amazon 服务的支持。	
3.17.1032	• 修复了当筛选条件设置为相同类别时出现重复系统日志的问题。 • 进行了修复，可防止在磁盘初始化期间挂起。	

版本	详细信息	发行日期
3.16.930	增加了对在启动时将“Windows 已可供使用”事件记录到 Windows 事件日志的支持。	
3.15.880	进行了修复，允许将 Systems Manager Run Command 输出上传到名称中带有“.”字符的 S3 存储桶。	
3.14.786	增加了对覆盖 InitializeDisks 插件设置的支持。例如：为了加快 SSD 磁盘初始化，您可以通过在 UserData 中指定的方式来临时禁用 TRIM： <InitializeDrivesSettings><SettingsGroup>FormatWithoutTRIM</SettingsGroup></InitializeDrivesSettings	
3.13.727	Systems Manager Run Command - 进行了修复，能够在 Windows 重新启动后可靠地处理命令。	
3.12.649	• 进行了修复，能够在运行命令/脚本时正常处理重新启动操作。 • 进行了修复，能够可靠地取消正在运行的命令。 • 增加了对（可选）在通过 Systems Manager Run Command 安装应用程序时将 MSI 日志上传到 S3 的支持。	
3.11.521	• 进行了修复，可启用适用于 Windows Server 2003 的 RDP 指纹生成。 • 进行了修复，以便在 EC2Config 日志行中包括时区和 UTC 偏移量。 • 对并行运行 Run Command 命令的 Systems Manager 支持。 • 回滚上一个更改以使分区磁盘联机。	

版本	详细信息	发行日期
3.10.442	• 修复了在安装 MSI 应用程序时的 Systems Manager 配置故障。 • 进行了修复，使存储磁盘能够可靠地联机。 • 进行了更新来改善对 Amazon 服务的支持。	
3.9.359	• 修复了后处理 Sysprep 脚本，将 Windows 更新配置保留为默认状态。 • 修复了密码生成插件，改善了在获取 GPO 密码策略设置时的可靠性。 • 将 EC2Config/SSM 日志文件夹权限限制为本地管理员组。 • 进行了更新来改善对 Amazon 服务的支持。	
3.8.294	• 修复了不在主驱动器上时阻止日志上传的 CloudWatch 问题。 • 通过添加重试逻辑改善了磁盘初始化过程。 • 增加了在创建 AMI 期间 SetPassword 插件偶尔发生故障时的改进的错误处理方式。 • 进行了更新来改善对 Amazon 服务的支持。	

版本	详细信息	发行日期
3.7.308	• 改进了用于实例内的测试和故障排除的 ec2config-cli 实用程序。 • 避免在 OpenVPN 适配器上为 Amazon KMS 和元数据服务添加静态路由。 • 修复了 UserData 执行未遵循“persist”标签的问题。 • 改进了 EC2 控制台日志记录不可用时的错误处理。 • 进行了更新来改善对 Amazon 服务的支持。	
3.6.269	• 修复了 Windows 激活可靠性，首先使用链路本地地址 169.254.0.250/251 来通过 Amazon KMS 激活 Windows • 改进了用于 Systems Manager、Windows 激活和域加入方案的代理处理方式 • 修复了将重复的用户账户行添加到 Sysprep 回复文件的问题	
3.5.228	• 解决了 CloudWatch 插件在读取 Windows 事件日志时可能会消耗过多的 CPU 和内存的情况 • 新增了一个指向 EC2Config 设置 UI 中的 CloudWatch 配置文档的链接	
3.4.212	• 修复了 EC2Config 与 VM-Import 结合使用时的问题。 • 修复了 WiX 安装程序中的服务命名问题。	

版本	详细信息	发行日期
3.3.174	• 改进了 Systems Manager 和域加入故障的异常处理。 • 进行了更改，以便支持 Systems Manager SSM 架构版本控制。 • 修复了 Win2K3 的临时磁盘格式化问题。 • 进行了更改，以便支持配置超过 2TB 的磁盘大小。 • 通过将 GC 模式设置为默认值来减少虚拟内存使用量。 • 支持从 `aws:psModule` 和 `aws:application` 插件中的 UNC 路径下载构件。 • 改进了 Windows 激活插件的日志记录。	
3.2.97	• 通过延迟加载 Systems Manager SSM 程序集改进了性能。 • 改进了格式错误的 sysprep2008.xml 的异常处理。 • 对 Systems Manager 的“Apply”配置的命令行支持。 • 进行了更改，以支持当存在暂挂计算机重命名操作时加入域。 • 支持 `aws:applications` 插件中的可选参数。 • 支持 `aws:psModule` 插件中的命令数组。	

版本	详细信息	发行日期
3.0.54	• 启用面向 Systems Manager 的支持。 • 自动通过 Systems Manager 将 EC2 Windows 实例加入到 Amazon目录域。 • 通过 Systems Manager 配置并上传 CloudWatch Logs/指标。 • 通过 Systems Manager 安装 PowerShell 模块。 • 通过 Systems Manager 安装 MSI 应用程序。	
2.4.233	• 增加了计划任务来从服务启动故障中恢复 EC2Config。 • 改进了控制台日志错误消息。 • 进行了更新来改善对 Amazon 服务的支持。	
2.3.313	• 修复了当启用 CloudWatch Logs 功能时某些情况下消耗大量内存的问题。 • 修复了升级错误，使得低于 2.1.19 的 ec2config 版本现在可以升级到最新版本。 • 更新了日志中的 COM 端口打开异常，使之可读性更好，更有帮助。 • Ec2configServiceSettings UI 禁用了大小调整，并修复了 UI 中的属性和版本显示位置。	
2.2.12	• 已处理在查询注册表项来确定 Windows Sysprep 状态时偶尔返回空值的情况下的 NullPointerException。 • 释放 finally 代码块中的未托管资源。	

版本	详细信息	发行日期
2.2.11	修复了 CloudWatch 插件处理空日志行的问题。	
2.2.10	• 删除了通过 UI 配置 CloudWatch Logs 设置的功能。 • 允许用户定义 `%ProgramFiles%\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` 文件中的 CloudWatch Logs 设置，以便在未来做出改进。	
2.2.9	修复了未处理异常，增加了日志记录。	
2.2.8	• 修复了 EC2Config 安装程序用来支持 Windows Server 2003 SP1 以及更高版本的 Windows 操作系统版本检查。 • 修复了读取与更新 Sysprep 配置文件相关的注册表项时的空值处理。	
2.2.7	• 增加了在针对 Windows 2008 及更高版本执行 Sysprep 期间运行 EC2Config 的支持。 • 改进了异常处理和日志记录，以实现更好的诊断	
2.2.6	• 减少了在上传日志事件时实例以及 CloudWatch Logs 上的负载。 • 解决了 CloudWatch Logs 插件未始终保持启用状态的升级问题	

版本	详细信息	发行日期
2.2.5	• 增加了对将日志上传到 CloudWatch 日志服务的支持。 • 修复了 Ec2OutputRDPCert 插件中的竞用情况问题 • 更改了 EC2Config 服务恢复选项以便从 TakeNoAction 重新启动 • 增加了 EC2Config 崩溃时的更多异常信息	
2.2.4	• 修复了 PostSysprep.cmd 中的拼写错误 • 修复了 EC2Config 不将自身固定到 OS2012+ 开始菜单的错误	
2.2.3	• 增加了安装 EC2Config 但不在安装后立即启动服务的选项。要使用该选项，可从命令提示符运行“Ec2Install.exe start=false” • 在墙纸插件中增加了参数来控制对墙纸的添加/删除。要使用该选项，可从命令提示符运行“Ec2WallpaperInfo.exe set”或“Ec2WallpaperInfo.exe revert” • 增加了对 RealTimeIsUniversal 键的检查，将不正确的 RealTimeIsUniveral 注册表项设置输出到控制台 • 删除了 Windows 临时文件夹的 EC2Config 依赖项 • 删除了 .Net 3.5 的 UserData 执行依赖项	
2.2.2	• 增加了对服务停止行为的检查来检查是否正在释放资源 • 修复了在加入域时长时间执行的问题	

版本	详细信息	发行日期
2.2.1	• 更新了安装程序，允许从较旧的版本升级 • 修复了纯 .Net4.5 环境中的 Ec2WallpaperInfo 错误 • 修复了间歇性的驱动程序检测错误 • 增加了无提示安装选项。使用“-q”选项执行 Ec2Install.exe。例如：“Ec2Install.exe -q”	
2.2.0	• 增加了对纯 .Net4 和 .Net4.5 环境的支持 • 更新了安装程序	
2.1.19	• 增加了使用 Intel 网络驱动程序时的临时磁盘标记支持 (例如 C3 实例类型)。有关更多信息，请参阅Amazon EC2 实例上的增强联网功能。 • 增加了对控制台输出的 AMI 原始版本和 AMI 源名称支持 • 对控制台输出进行了更改，以实现一致的格式/分析 • 更新了帮助文件	
2.1.18	• 对完成通知增加了 EC2Config WMI 对象 (-Namespace root\Amazon -Class EC2_ConfigService) • 改进了大事件日志的启动 WMI 查询的性能；此类查询可能会在初始执行期间导致长时间的高 CPU 占用率	

版本	详细信息	发行日期
2.1.17	• 修复了与标准输出以及标准错误缓冲区填充有关的 UserData 执行问题 • 修复了在 w2k8 及更高版本的操作系统上有时会出现的不正确的 RDP 指纹 • 控制台输出现在对于 Windows 2008 及更高版本的操作系统包含“RDPCERTIFICATE-SubjectName:”，其中带有计算机名称值 • 在盘符映射下拉列表中增加了 D:\ • 将“帮助”按钮移到了右上角，并更改了外观 • 在右上角增加了反馈调查链接	

版本	详细信息	发行日期
2.1.16	• “常规”选项包含指向新版本的 EC2Config 下载页面的链接 • 重叠桌面墙纸现在存储在用户本地的 Appdata 文件夹而不是“我的文档”中，以便支持 MyDoc 重定向 • MSSQLServer 名称在后处理 Sysprep 脚本中与系统同步 (2008+) • 对应用程序文件夹重新排序 (将文件移到插件目录并删除了重复的文件) • 更改了系统日志输出 (控制台)： • *转变为日期、名称、值的格式，以便于分析 (请开始将依赖项迁移到新格式) • *增加了“Ec2SetPassword”插件状态 • *增加了 Sysprep 开始和结束时间 • 修复了在非英语操作系统中临时磁盘未标记为“临时存储”的问题 • 修复了运行 Sysprep 后出现的 EC2Config 卸载故障	

版本	详细信息	发行日期
2.1.15	• 优化了对元数据服务的请求 • 元数据现在绕过代理设置 • 临时磁盘标记为“临时存储”，并且 Important.txt 放在找到的卷上 (仅 Citrix PV 驱动程序)。有关更多信息，请参阅在 EC2 Windows 实例上升级半虚拟化驱动程序。 • 向临时磁盘分配从 A 到 Z 的盘符 (仅 Citrix PV 驱动程序) - 这一分配可通过盘符映射插件使用卷标签“Temporary Storage X” (其中 x 是 0 到 25 之间的数字) 来覆盖 • UserData 现在会在“Windows 已准备就绪”后立即运行	
2.1.14	桌面墙纸修复	
2.1.13	• 桌面墙纸在默认情况下将显示主机名称 • 删除了 Windows 时间服务的依赖项 • 在多个 IP 分配到一个接口的情况下增加了路由	
2.1.11	• 对 Ec2Activation 插件进行了更改 • - 每 30 天验证激活状态 • - 如果宽限期还剩下 90 天 (共 180 天)，则重新尝试激活	

版本	详细信息	发行日期
2.1.10	• 重叠桌面墙纸不再随 Sysprep 保留，或者在无 Sysprep 的情况下关机时不再保留 • 每次使用 <persist>true</persist> 启动服务时运行的 Userdata 选项 • 将 /DisableWinUpdate.cmd 的位置和名称更改为 /Scripts/PostSysprep.cmd • 管理员密码默认情况下设置为在 /Scripts/PostSysprep.cmd 中不过期 • 卸载程序将从 c:\windows\setup\script\CommandComplete.cmd 中删除 EC2Config PostSysprep 脚本 • 增加路由支持自定义接口指标	
2.1.9	UserData 执行不再限制为 3851 个字符	

版本	详细信息	发行日期
2.1.7	• 将操作系统版本和语言标识符写入到控制台中 • 将 EC2Config 版本写入到控制台中 • 将半虚拟化驱动程序版本写入到控制台中 • 检测错误检查，并在找到错误的情况下在下次启动时输出到控制台 • 在 config.xml 中增加了选项来保存 Sysprep 凭证 • 增加在启动时 ENI 不可用的情况下执行的路由重试逻辑 • UserData 执行 PID 写入到控制台中 • 从 GPO 检索生成的最短密码长度 • 将服务启动设置为重试 3 次 • 向 /Scripts 文件夹增加了 S3_DownloadFile.ps1 和 S3_Uploadfile.ps1 示例	

版本	详细信息	发行日期
2.1.6	• 在“常规”选项卡中增加了版本信息 • 将“服务包”选项卡重命名为“映像” • 简化了指定密码的过程，并且将密码相关的 UI 从“常规”选项卡转移到“映像”选项卡 • 将“磁盘设置”选项卡重命名为“存储” • 增加了提供常见故障诊断工具的“支持”选项卡 • 设置 Windows Server 2003 `sysprep.ini` 以在默认情况下扩展操作系统分区 • 向墙纸中添加私有 IP 地址 • 在墙纸上显示私有 IP 地址 • 为控制台输出增加了重试逻辑 • 修复了元数据可访问性的 Com 端口异常 -- 该异常会导致 EC2Config 在显示控制台输出前终止 • 每次启动时检查激活状态 -- 必要时会激活 • 修复了相对路径问题 -- 在从启动文件夹手动执行墙纸快捷方式时会导致该问题；指向管理员/日志 • 修复了 Windows Server 2003 用户的默认背景颜色 (管理员除外)	

版本	详细信息	发行日期
2.1.2	• 采用 UTC (祖鲁) 的控制台时间戳 • “Sysprep”选项卡上不再显示超链接 • 增加了在 Windows 2008 和更高版本上首次启动时动态扩展根卷的功能 • 当启用“设置密码”时，现在自动允许 EC2Config 设置密码 • EC2Config 会在运行 Sysprep 之前检查激活状态 (如果未激活，则发出警告) • Windows Server 2003 `Sysprep.xml` 现在默认采用 UTC 时区而不是太平洋时区 • 随机化的激活服务器 • 将“磁盘映射”选项卡重命名为“磁盘设置” • 已将初始化驱动器 UI 项从“常规”选项卡移动到“磁盘设置”选项卡 • “帮助”按钮现在指向 HTML 帮助文件 • 使用更改对 HTML 帮助文件进行了更新 • 更新了盘符映射的“注释”文本 • 向 /Scripts 文件夹增加了 InstallUpdates.ps1，在执行 Sysprep 之前执行自动修补和清除	

版本	详细信息	发行日期
2.1.0	• 默认情况下，桌面墙纸在第一次登录 (不是断开/重新连接)时显示实例信息 • 通过将代码放在 <powershell></powershell> 之内，PowerShell 可以从 UserData 中运行	

对 Windows 实例使用 EC2 Fast Launch

为 EC2 Fast Launch 配置 Windows Server AMI 时，Amazon EC2 会创建一组预置的快照，用于加快启动速度，如下所示。

1. Amazon EC2 会根据您的设置启动一组临时 t3 实例。
2. 当每个临时实例完成标准启动步骤时，Amazon EC2 会创建该实例的预置快照。其将快照存储在您的 Amazon S3 桶中。
3. 快照准备就绪后，Amazon EC2 会终止关联的 t3 实例，从而尽可能地降低资源成本。
4. 下次 Amazon EC2 从启用了 EC2 Fast Launch 的 AMI 启动实例时，将使用其中的一个快照来显著缩短启动时间。

Amazon EC2 会自动补充您手头的快照，因为它会使用这些快照从启用了 EC2 Fast Launch 的 AMI 启动实例。

任何有权访问启用了 EC2 Fast Launch 的 AMI 的账户都可以从缩短的启动时间中受益。当 AMI 所有者授予您启动实例的权限时，预置快照将来自 AMI 所有者的账户。

如果与您共享了支持 EC2 Fast Launch 的 AMI，则可以在共享的 AMI 上启用或禁用快速启动功能。如果为 EC2 Fast Launch 启用共享的 AMI，Amazon EC2 会直接在账户中创建预置快照。如果您耗尽账户中的快照，您仍然可以使用 AMI 所有者账户中的快照。

> **Note**
>
> EC2 Fast Launch 会在启动使用预置快照后立即将其删除，从而最大限度地降低存储成本并防止重用。但是，如果已删除的快照符合保留规则，则回收站会自动将其保留。我们建议您查看回收站保留规则的范围，以免发生这种情况。有关更多信息，请参阅《Amazon EBS User Guide》中的 Recycle Bin。

此功能与 EBS 快速快照还原不同。您必须逐个显式启用 EBS 快速快照还原，并且会产生相关的成本。

以下视频演示了如何配置 Windows AMI 以加快启动速度，并快速概述了相关关键术语及其定义：在 Amazon 上启动 EC2 Windows 实例的速度最多可提高 65%。

资源成本

为 EC2 Fast Launch 配置 Windows AMI 无需支付服务费。但是，标准定价适用于 Amazon EC2 使用的任何底层 Amazon 资源。要了解有关相关资源成本以及如何进行管理的更多信息，请参阅 管理 EC2 Fast Launch 底层资源的成本。

内容

- 关键术语
- Windows 的 EC2 Fast Launch 先决条件
- 为 Amazon EC2 Windows Server AMI 配置 EC2 Fast Launch 设置
- 查看启用了 EC2 Fast Launch 的 AMI
- 管理 EC2 Fast Launch 底层资源的成本
- 监控 EC2 Fast Launch
- 用于 EC2 Fast Launch 的服务相关角色
- 排查 Windows EC2 Fast Launch 问题

关键术语

EC2 Fast Launch 功能使用以下关键术语：

预置的快照

满足以下条件的实例的快照：已从启用了 EC2 Fast Launch 的 Windows AMI 启动，完成了下列 Windows 启动步骤，并根据需要重启。

- Sysprep 专门化
- Windows 全新体验（OOBE）

完成这些步骤后，EC2 Fast Launch 会停止实例并创建一张快照，这样即可根据您的配置缩短以后从 AMI 启动的时间。

启动频率

控制 Amazon EC2 可在指定时间范围内启动的预置快照的数量。为 AMI 启用 EC2 Fast Launch 时，Amazon EC2 会在后台创建一组初始的预置快照。例如，如果启动频率设置为每小时启动五次（默认值），则 EC2 Fast Launch 会创建一组初始的预置快照（共五张）。

Amazon EC2 从启用了 EC2 Fast Launch 的 AMI 启动实例时，它将使用其中的一张预置快照来缩短启动时间。使用快照时，它们会自动补充，最多不超过启动频率指定的数量。

如果您预计从 AMI 启动的实例数量会飙升（例如，在特殊活动期间），则可以提前提高启动频率以覆盖所需的其他实例。当启动速率恢复正常时，您可以重新调低频率。

当您遇到的启动次数超过预期时，您可能会耗尽可用的所有预置快照。这不会导致任何启动失败。但是，它可能会导致某些实例经历标准启动过程，直到可以补充快照为止。

目标资源计数

为启用了 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI 预留的预置快照数量。

最大并行启动数

控制 Amazon EC2 可以同时启动多少个实例，以便为 EC2 Fast Launch 创建预置快照。如果您的目标资源数高于您配置的最大并行启动数，Amazon EC2 将启动最大并行启动数指定的实例数量以开始创建快照。当这些实例完成该过程时，Amazon EC2 会创建快照并停止实例。然后，它会继续启动更多实例，直至可用快照总数达到目标资源计数。最大并行启动数的值必须大于等于 6。

Windows 的 EC2 Fast Launch 先决条件

在设置 EC2 Fast Launch 之前，请确认您已满足在 Amazon Web Services 账户 中为 AMI 创建快照所需的以下先决条件：

- 如果在配置 EC2 Fast Launch 时提供了自定义启动模板，则该服务将使用您在启动模板中定义的 VPC 和其他配置设置。有关更多信息，请参阅 在设置 EC2 Fast Launch 时使用启动模板。
- 如果不使用自定义启动模板来配置设置，则在启用 EC2 Fast Launch 之前，必须将 EC2FastLaunchFullAccess 策略附加到当前 IAM 角色。然后，该服务会自动创建一个 Amazon CloudFormation 堆栈，其中包含您的 Amazon Web Services 账户中的以下资源。
  - 虚拟私有云（VPC）
  - 跨多个可用区的若干私有子网
  - 一个使用实例元数据服务版本 2（IMDSv2）配置的启动模板

  - 一个没有任何入站或出站规则的安全组
- 专用 EC2 Fast Launch AMI 必须支持执行用户数据脚本。
- 要为 AMI 配置 EC2 Fast Launch，您必须使用 Sysprep 和关机选项创建该 AMI。EC2 Fast Launch 功能目前不支持从正在运行的实例创建的 AMI。

  要使用 Sysprep 创建 AMI，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。
- 要为使用客户自主管理型密钥进行加密的加密 AMI 启用 EC2 Fast Launch，您必须授予 EC2 Fast Launch 的服务相关角色使用 CMK 的权限。有关更多信息，请参阅 the section called “访问客户自主管理型密钥”。
- 涵盖 Amazon Web Services 账户 中所有 AMI 的最大并行启动数默认配额为每个区域 40 次。您可以请求增加账户的服务限额，方法如下。

  1. 访问 https://console.aws.amazon.com/servicequotas/，打开服务配额控制台。
  2. 在导航窗格中，请选择 Amazon Web Services 服务。
  3. 在搜索栏中，输入 **`EC2 Fast Launch`**，然后选择结果。
  4. 选择并行实例启动的链接以打开服务配额详细信息页面。
  5. 选择请求增加账户配额。

  有关更多信息，请参阅《Service Quotas 用户指南》中的请求增加配额。

为 Amazon EC2 Windows Server AMI 配置 EC2 Fast Launch 设置

您可以为拥有的 Windows AMI 或通过 Amazon Web Services 管理控制台、API、开发工具包、CloudFormation 或 Amazon Command Line Interface（Amazon CLI）与您共享的 AMI 配置 EC2 Fast Launch。在配置 EC2 Fast Launch 之前，请验证您的 AMI 是否满足创建预置快照所需的所有先决条件。有关更多信息，请参阅 Windows 的 EC2 Fast Launch 先决条件。

为 Windows 实例启用快速启动时，Amazon EC2 会进行检查，确保您拥有从指定 AMI 和启动模板（如果提供）启动实例所需的权限，包括加密 AMI 的权限。为防止实例启动过程中出现错误，该服务会在启用 EC2 Fast Launch 之前验证您的权限。如果您没有所需的权限，则服务会返回错误，并且不会启用 EC2 Fast Launch。

EC2 Fast Launch 与 EC2 Image Builder 集成，可帮助您在启用 EC2 Fast Launch 时创建自定义映像。有关更多信息，请参阅《EC2 Image Builder 用户指南》中的在 EC2 快速启动启用时创建 Windows AMI 的分配设置（Amazon CLI）。

启用 EC2 Fast Launch

要更改这些设置，请确保您的 AMI 和运行所在的区域满足所有 Windows 的 EC2 Fast Launch 先决条件。

Console

启用 EC2 Fast Launch

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，在映像下选择 AMI。
3. 通过选择 Name（名称）旁边的复选框来选择要更新的 AMI。
4. 从 AMI 列表上方的操作菜单中，选择配置快速启动。这将打开配置快速启动页面，您可以在此页面中配置 EC2 Fast Launch 设置。
5. 要开始使用预置快照以更快地从 Windows AMI 启动实例，请选择为 Windows 启用快速启动复选框。
6. 从 Set anticipated launch frequency（设置预期的启动频率）下拉列表中，选择一个值以指定为覆盖预期实例启动卷而创建和维护的快照数量。
7. 完成更改后，选择 Save changes（保存更改）。

> **Note**
>
> 如果需要使用启动模板来指定某个 VPC，或为 IMDSv2 配置元数据设置，请参阅在设置 EC2 Fast Launch 时使用启动模板。

Amazon CLI

启用 EC2 Fast Launch

使用以下 enable-fast-launch 命令，为指定的 AMI 启用 EC2 Fast Launch，这会启动六个并行实例来进行预调配。

```
aws ec2 enable-fast-launch \
    --image-id ami-0abcdef1234567890  \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

下面是示例输出。

```
{
 "ImageId": "ami-0abcdef1234567890",
 "ResourceType": "snapshot",
 "SnapshotConfiguration": {
     "TargetResourceCount": 10
 },
 "LaunchTemplate": {},
 "MaxParallelLaunches": 6,
 "OwnerId": "0123456789123",
 "State": "enabling",
 "StateTransitionReason": "Client.UserInitiated",
 "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

PowerShell

启用 EC2 Fast Launch

使用 Enable-EC2FastLaunch cmdlet，为指定的 AMI 启用 EC2 Fast Launch，这会启动六个并行实例来进行预调配。

```
Enable-EC2FastLaunch `
 -ImageId ami-0abcdef1234567890 `
 -MaxParallelLaunch 6 `
 -Region us-west-2 `
 -ResourceType snapshot
```

下面是示例输出。

```
ImageId               : ami-0abcdef1234567890
 LaunchTemplate        :
 MaxParallelLaunches   : 6
 OwnerId               : 0123456789123
 ResourceType          : snapshot
 SnapshotConfiguration : Amazon.EC2.Model.FastLaunchSnapshotConfigurationResponse
 State                 : enabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime   : 2/25/2022 12:24:11 PM
```

禁用 EC2 Fast Launch

要更改这些设置，请确保您的 AMI 和运行所在的区域满足所有 Windows 的 EC2 Fast Launch 先决条件。

Console

禁用 EC2 Fast Launch

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，在映像下选择 AMI。
3. 通过选择 Name（名称）旁边的复选框来选择要更新的 AMI。
4. 从 AMI 列表上方的操作菜单中，选择配置快速启动。这将打开配置快速启动页面，您可以在此页面中配置 EC2 Fast Launch 设置。
5. 清除为 Windows 启用快速启动复选框，以禁用 EC2 Fast Launch 并删除预置快照。这将导致 AMI 向前为每个实例使用标准启动流程。

> **Note**
>
> 禁用 Windows 镜像优化后，任何现有的预置快照都将自动删除。必须先完成此步骤，然后才能再次开始使用该功能。

6. 完成更改后，选择 Save changes（保存更改）。

Amazon CLI

禁用 EC2 Fast Launch

使用以下 disable-fast-launch 命令，在指定的 AMI 上禁用 EC2 Fast Launch，并清理现有的预调配快照。

```
aws ec2 disable-fast-launch --image-id ami-01234567890abcedf
```

下面是示例输出。

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
```

```
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
}
```

PowerShell

禁用 EC2 Fast Launch

使用 Disable-EC2FastLaunch cmdlet，在指定的 AMI 上禁用 EC2 Fast Launch，并清理现有的预调配快照。

```
Disable-EC2FastLaunch -ImageId ami-0abcdef1234567890
```

下面是示例输出。

```
ImageId               : ami-0abcdef1234567890
 LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
 MaxParallelLaunches   : 6
 OwnerId               : 0123456789123
 ResourceType          : snapshot
 SnapshotConfiguration :
 State                 : disabling
 StateTransitionReason : Client.UserInitiated
 StateTransitionTime   : 2/25/2022 1:10:08 PM
```

在设置 EC2 Fast Launch 时使用启动模板

您可以使用启动模板配置一组启动参数，Amazon EC2 每次从该模板启动实例时都会使用这些参数。您可以指定用于基础映像的 AMI、实例类型、存储、网络设置等内容。

启动模板是可选的，但以下特定情况除外：在配置更快启动时，必须为 Windows AMI 使用启动模板：

- 您必须使用启动模板为 Windows AMI 指定现有的 VPC。如果您的 Amazon Web Services 账户使用默认 VPC，则不适用这一要求。
- 如果账户包含对 Amazon EC2 实例强制实施 IMDSv2 的策略，您必须创建一个启动模板来指定元数据配置以强制实施 IMDSv2。

  使用包含 EC2 控制台的元数据配置的启动模板，或在 Amazon CLI 中运行 enable-fast-launch 命令，或调用 EnableFastLaunch API 操作。

使用启动模板时，Amazon EC2 EC2 Fast Launch 不支持以下配置。如果您使用启动模板来设置 EC2 Fast Launch，则不得指定以下任何内容：

- 用户数据脚本
- 终止保护
- 禁用元数据
- 竞价型选项
- 终止实例的关闭行为
- 网络接口、弹性图形或竞价型实例请求的资源标签

指定 VPC

步骤 1：创建启动模板

创建一个启动模板，指定 Windows 实例的以下详细信息：

- VPC 子网。
- 实例类型为 `t3.xlarge`。

有关更多信息，请参阅 创建 Amazon EC2 启动模板。

步骤 2：为 EC2 Fast Launch AMI 指定启动模板

Console

为 EC2 Fast Launch 指定启动模板

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，在映像下选择 AMI。

3. 通过选择 Name（名称）旁边的复选框来选择要更新的 AMI。
4. 从 AMI 列表上方的操作菜单中，选择配置快速启动。这将打开配置快速启动页面，您可以在此页面中配置 EC2 Fast Launch 设置。
5. Launch template（启动模板）框执行筛选搜索，在当前区域的账户中查找与您输入的文本相匹配的启动模板。在框中输入完整的启动模板名称或 ID 或其一部分，以显示匹配的启动模板列表。例如，如果您在框中输入 `fast`，Amazon EC2 会在当前区域中查找您的账户中名称包含“fast”的所有启动模板。

   要创建启动模板，请选择 Create launch template（创建启动模板）。
6. 当您选择某个启动模板时，Amazon EC2 会在 Source template version（源模板版本）框中显示该模板的默认版本。要指定其他版本，请突出显示要替换的默认版本，然后在框中输入所需的版本号。
7. 完成更改后，选择 Save changes（保存更改）。

Amazon CLI

为 EC2 Fast Launch 指定启动模板

使用 enable-fast-launch 命令和 `--launch-template` 选项，并指定启动模板的名称或 ID。

```
--launch-template LaunchTemplateName=my-launch-template
```

PowerShell

为 EC2 Fast Launch 指定启动模板

使用 Enable-EC2FastLaunch cmdlet 以及 `-LaunchTemplate_LaunchTemplateId` 或 `-LaunchTemplate_LaunchTemplateName` 参数。

```
-LaunchTemplate_LaunchTemplateName my-launch-template
```

有关 EC2 启动模板的更多信息，请参阅在 Amazon EC2 启动模板中存储实例启动参数。

查看启用了 EC2 Fast Launch 的 AMI

您可以在 Amazon CLI 中使用 describe-fast-launch-images 命令，或者 Get-EC2FastLaunchImage Tools for PowerShell Cmdlet，获取启用了 EC2 Fast Launch 的 AMI 的详细信息。

Amazon EC2 会提供结果中返回的每个 Windows AMI 的以下详细信息：

- 启用了 EC2 Fast Launch 的 AMI 的映像 ID。
- 用于预调配关联 Windows AMI 的资源类型。支持的值为：`snapshot`。
- 快照配置，这是用于为使用快照的关联 Windows AMI 配置预调配的一组参数。
- 启动模板信息，其中包括关联的 AMI 从预调配快照启动 Windows 实例时使用的启动模板的 ID、名称和版本。
- 可为创建资源而同时启动的实例的最大数量。
- 关联 AMI 的所有者 ID。对于与您共享的 AMI，不会填充此内容。
- 关联 AMI 的当前 EC2 Fast Launch 状态。支持的值包括：`enabling | enabling-failed | enabled | enabled-failed | disabling | disabling-failed`。

> **Note**
>
> 您还可在 EC2 控制台中的 Manage image optimization（管理映像优化）页面上看到，当前的状态为 Image optimization state（映像优化状态）。

- 关联 AMI 的 EC2 Fast Launch 变为当前状态的原因。
- 关联 AMI 的 EC2 Fast Launch 变为当前状态的时间。

Amazon CLI

查找配置了 EC2 Fast Launch 的 AMI

使用以下 describe-fast-launch-images 命令，描述账户中配置了 EC2 Fast Launch 的每个 AMI 的详细信息。在此示例中，账户中只有一个 AMI 配置为 EC2 Fast Launch。

```
aws ec2 describe-fast-launch-images
```

下面是示例输出。

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
```

```
                "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
                "Version": "1"
            },
            "MaxParallelLaunches": 6,
            "OwnerId": "0123456789123",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated",
            "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
        }
    ]
}
```

PowerShell

查找配置了 EC2 Fast Launch 的 AMI

使用以下 Get-EC2FastLaunchImage cmdlet，描述账户中配置了 EC2 Fast Launch 的每个 AMI 的详细信息。在此示例中，账户中只有一个 AMI 配置为 EC2 Fast Launch。

```
Get-EC2FastLaunchImage -ImageId ami-0abcdef1234567890
```

下面是示例输出。

```
ImageId               : ami-0abcdef1234567890
LaunchTemplate        :
 Amazon.EC2.Model.FastLaunchLaunchTemplateSpecificationResponse
MaxParallelLaunches   : 6
OwnerId               : 012345678912
ResourceType          : snapshot
SnapshotConfiguration :
State                 : enabled
StateTransitionReason : Client.UserInitiated
StateTransitionTime   : 2/25/2022 12:54:43 PM
```

管理 EC2 Fast Launch 底层资源的成本

为 EC2 Fast Launch 配置 Windows AMI 无需支付服务费。但是，当您为 Amazon EC2 Windows AMI 启用 EC2 Fast Launch 时，标准定价适用于 Amazon EC2 用于准备和存储预置快照的底层 Amazon 资源。您可以配置成本分配标签，帮助跟踪和管理与 EC2 Fast Launch 资源相关的成本。有关如何配置成本分配标签的更多信息，请参阅 在账单上跟踪 EC2 Fast Launch 成本。

以下示例演示如何分配与 EC2 Fast Launch 快照成本相关的成本。

示例场景：AtoZ Example 公司有一个具有 50 GiB EBS 根卷的 Windows AMI。它们可以为其 AMI 启用 EC2 Fast Launch，并将目标资源计数设置为五。在一个月的期限内，为 AMI 使用 EC2 Fast Launch 的成本约为 5.00 美元，成本细目如下：

1. 当 AtoZ Example 启用 EC2 Fast Launch 时，Amazon EC2 会启动五个小型实例。每个实例都会完成 Sysprep 和 OOBE Windows 启动步骤，并根据需要重启。每个实例需要几分钟时间（时长可能有所不同，具体取决于该区域或可用区（AZ）的忙闲状况以及 AMI 的大小）。

   成本

   - 实例运行时成本（或最小运行时 [如适用]）：五个实例
   - 卷成本：五个 EBS 根卷
2. 预置过程完成后，Amazon EC2 会创建实例的快照并将其存储在 Simple Storage Service（Amazon S3）中。快照通常会存储 4-8 小时，然后才会被启动消耗。在这种情况下，每张快照的成本大约为 0.02 至 0.05 美元。

   成本

   - 快照存储 [Simple Storage Service（Amazon S3）]：五张快照
3. Amazon EC2 创建快照后，会停止实例。此时，实例不再累积成本。但是，EBS 卷的成本会继续累积。

   成本

   - EBS 卷：关联的 EBS 根卷的成本会继续增加。

> **Note**
>
> 此处显示的成本仅用于演示目的。成本可能有所不同，具体取决于您的 AMI 配置和定价套餐。

在账单上跟踪 EC2 Fast Launch 成本

成本分配标签有利于整理 Amazon 账单，从而反映与 EC2 Fast Launch 相关的成本。Amazon EC2 在为 EC2 Fast Launch 准备和存储预置快照时，会将以下标签添加到其创建的资源中。您可以使用该标签：

标签键：`CreatedBy`，值：`EC2 Fast Launch`

在 Billing and Cost Management 控制台中激活这些标签并设置详细账单报告后，报告中会显示 `user:CreatedBy` 列。该列包含来自所有服务的值。但是，如果您下载了 CSV 文件，则可以将数据导入电子表格，然后在值中筛选 `EC2 Fast Launch`。标签激活后，此信息也会显示在 Amazon 成本和使用情况报告 中。

步骤 1：激活用户定义的成本分配标签

要在成本报告中包含资源标签，您必须先在 Billing and Cost Management 控制台中激活这些标签。有关更多信息，请参阅《Amazon 账单与成本管理 用户指南》中的激活用户定义的成本分配标签。

> **Note**
>
> 激活时间可能长达 24 小时。

步骤 2：设置成本报告

如果您已经设置了成本报告，则在激活完成后，下次运行报告时会显示您的标签列。若是首次设置成本报告，请选择以下选项之一。

- 请参阅 Amazon 账单与成本管理 用户指南中的设置月度成本分配报告。
- 请参阅 Amazon 成本和使用情况报告 用户指南中的创建成本和使用情况报告。

> **Note**
>
> Amazon 最多可能需要 24 小时才能开始将报告传输到 S3 存储桶。

您可以为拥有的 Windows AMI 或通过 Amazon EC2 控制台、API、开发工具包、CloudFormation 或 Amazon CLI 中的 ec2 命令与您共享的 AMI 配置 EC2 Fast Launch。以下部分介绍了 Amazon EC2 控制台和 Amazon CLI 的配置步骤。

您还可以使用 EC2 Image Builder 创建为 EC2 Fast Launch 配置的自定义 Windows AMI。有关更多信息，请参阅 Create distribution settings for a Windows AMI with EC2 Fast Launch enabled (Amazon CLI)。

监控 EC2 Fast Launch

本节介绍如何监控账户中已启用 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI。

使用 EventBridge 监控 EC2 Fast Launch 的状态更改

如果启用 EC2 Fast Launch 的 Windows AMI 状态发生更改，Amazon EC2 就会生成 `EC2 Fast Launch State-change Notification` 事件。然后，Amazon EC2 会将状态更改事件发送到 Amazon EventBridge（以前称为 Amazon CloudWatch Events）。

您可以创建 EventBridge 规则来触发一个或多个操作以响应状态更改事件。例如，您可以创建 EventBridge 规则，检测何时启用了 EC2 Fast Launch，并执行以下操作：

- 将消息发送到 Amazon SNS 主题以通知订阅者。
- 调用执行某些操作的 Lambda 函数。
- 将状态更改数据发送到 Amazon Data Firehose 以进行分析。

有关更多信息，请参阅《Amazon EventBridge 用户指南》中的创建对事件作出反应的 Amazon EventBridge 规则。

状态更改事件

EC2 Fast Launch 功能会尽力发出 JSON 格式的状态更改事件。Amazon EC2 会近乎实时地将事件发送到 EventBridge。本部分介绍了事件字段并显示了事件格式的示例。

`EC2 Fast Launch State-change Notification`

imageId

识别 EC2 Fast Launch 状态发生更改的 AMI。

resourceType

用于预调配资源类型。支持的值为：`snapshot`。默认值为 `snapshot`。

状态

指定 AMI 的 EC2 Fast Launch 功能的当前状态。有效值包括：

- 正在启用 – 您已为该 AMI 启用了 EC2 Fast Launch 功能，并且 Amazon EC2 已开始为预置过程创建快照。
- 启用失败 – 首次为 AMI 启用 EC2 Fast Launch 功能时出现错误，导致预置过程失败。这种情况可能在预调配过程中随时发生。
- 已启用 – EC2 Fast Launch 功能已启用。Amazon EC2 为新启用的 EC2 Fast Launch AMI 创建首张预置快照后，状态会立即变为 `enabled`。如果 AMI 已启用并再次执行预调配过程，状态会立即更改。

- 启用失败 – 仅当 EC2 Fast Launch AMI 并非首次经历预置过程时，才适用此状态。如果禁用 EC2 Fast Launch 功能后重新启用，或者在首次完成预置后发生配置更改或出现其他错误，就可能会出现这种情况。
- 正在禁用 – AMI 所有者已为该 AMI 关闭 EC2 Fast Launch 功能，并且 Amazon EC2 已开始执行清理过程。
- 已禁用 – EC2 Fast Launch 功能已禁用。Amazon EC2 完成清除过程后，状态会更改为 `disabled`。
- 禁用失败 – 出现错误，导致清除过程失败。这意味着某些预调配快照可能仍保留在账户中。

stateTransitionReason

EC2 Fast Launch AMI 的状态发生更改的原因。

> **Note**
>
> 此事件消息中的所有字段均为必填字段。

以下示例演示了一个新启用的 EC2 Fast Launch AMI，其已启动首个实例来启动预置过程。此时，状态为 `enabling`。Amazon EC2 创建首个预调配快照后，状态更改为 `enabled`。

```
{
 "version": "0",
 "id": "01234567-0123-0123-0123-012345678901",
 "detail-type": "EC2 Fast Launch State-change Notification",
 "source": "aws.ec2",
 "account": "123456789012",
 "time": "2022-08-31T20:30:12Z",
 "region": "us-east-1",
 "resources": [
  "arn:aws:ec2:us-east-1:123456789012:image/ami-123456789012"
 ],
 "detail": {
  "imageId": "ami-123456789012",
  "resourceType": "snapshot",
  "state": "enabling",
  "stateTransitionReason": "Client.UserInitiated"
 }
}
```

使用 CloudWatch 监控 EC2 Fast Launch 指标

启用了 EC2 Fast Launch 的 Amazon EC2 AMI 会向 Amazon CloudWatch 发送指标。可使用 Amazon Web Services 管理控制台、Amazon CLI 或 API 列出 EC2 Fast Launch 发送到 CloudWatch 的指标。AWS/EC2 命名空间包括以下 EC2 Fast Launch 指标：

指标	描述
NumberOfAvailableFastLaunchSnapshots	每个启用 EC2 Fast Launch 的 AMI 可用的预置快照数量。
NumberOfInstancesFastLaunched	每个启用 EC2 Fast Launch 的 AMI 从预置快照启动的实例数。
NumberOfInstancesNotFastLaunched	由于启动时缺乏可用预置快照而导致冷启动的每个启用 EC2 Fast Launch 的 AMI 实例数量。
FastLaunchSnapshotUsedToRefillStartTime	为了在使用现有快照后创建另一个快照，Amazon EC2 从启用了 EC2 Fast Launch 的 AMI 中启动新映像的时间戳。
FastLaunchSnapshotCreationTime	测量 Amazon EC2 启动实例并为启用了 EC2 Fast Launch 的 AMI 创建快照所花费的时间。

用于 EC2 Fast Launch 的服务相关角色

Amazon EC2 使用服务相关角色获取代表您调用其他 Amazon Web Services 服务 所需的权限。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Web Services 服务 直接相关。服务相关角色提供了一种将权限委托给 Amazon Web Services 服务 的安全方式，因为只有相关服务才能担任服务相关角色。有关 Amazon EC2 如何使用 IAM 角色的更多信息，请参阅 适用于 Amazon EC2 的 IAM 角色。

Amazon EC2 使用名为 AWSServiceRoleForEC2FastLaunch 的服务相关角色创建和管理一组预置快照，从而减少从 Windows AMI 启动实例所需的时间。

AWSServiceRoleForEC2FastLaunch 授予的权限

AWSServiceRoleForEC2FastLaunch 服务相关角色仅信任以下服务来担任该角色：

- ec2fastlaunch.amazonaws.com

Amazon EC2 使用 EC2FastLaunchServiceRolePolicy 托管策略完成以下操作：

- Amazon CloudFormation：允许 EC2 Fast Launch 获取关联的 CloudFormation 堆栈的描述。
- Amazon CloudWatch：将与 EC2 Fast Launch 关联的指标数据发布到 Amazon EC2 命名空间。
- Amazon EC2：授予 EC2 Fast Launch 执行以下操作的权限：
  - 从启用了 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI 启动实例，以便执行预置步骤。此外，还需要指定资源模式以允许对关联到 License Manager 的 AMI 执行 `ec2:RunInstances` 操作。
  - 在 EC2 Fast Launch 创建预调配快照后，停止并终止由其启动的实例。
  - 描述用于从启用了 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI 启动实例的映像和实例类型资源，并利用这些资源创建快照。
  - 描述启动模板资源并从启动模板启动实例。
  - 描述实例、实例属性和实例状态、卷和卷属性。
  - 描述网络接口。
  - 删除 EC2 Fast Launch 创建的资源，包括快照、启动模板、卷和网络接口。
  - 标记 EC2 Fast Launch 为了启动和预调配 Windows 实例的资源，并创建快照以便在最终启动过程中使用。
- Amazon EventBridge：包含创建 EventBridge 事件规则以及检索所创建规则的详细信息或删除规则的权限。EC2 Fast Launch 还可获取接收基于事件规则转发的 EC2 Fast Launch 事件的目标服务列表，以及将目标服务添加到由其创建的事件规则或将其从所创建的规则中移除。
- IAM：允许 EC2 Fast Launch 创建 `EC2FastLaunchServiceRolePolicy` 服务相关角色，获取和使用名称中包含 `ec2fastlaunch` 的实例配置文件，以及使用启动模板中的实例配置文件代表您启动实例。
- Amazon KMS：包含创建授权以及列出由 EC2 Fast Launch 创建并且可以停用的授权。还用于描述或使用密钥来加密或解密挂载到 EC2 Fast Launch 所创建实例的卷，以及生成非明文数据密钥。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 EC2FastLaunchServiceRolePolicy。

有关 Amazon EC2 使用托管式策略的更多信息，请参阅 Amazon EC2 的 Amazon 托管式策略。

创建服务相关角色

您无需手动创建该服务相关角色。开始为 AMI 使用 EC2 Fast Launch 时，如果服务相关角色尚不存在，Amazon EC2 将为您创建一个。

如果从账户中删除了该服务相关角色，则可以为另一个 Windows AMI 启用 EC2 Fast Launch，以便在账户中重新创建此角色。或者，您还可以为当前 AMI 禁用 EC2 Fast Launch，然后将其重新启用。但是，禁用该功能会导致您的 AMI 对所有新实例使用标准启动流程，而 Amazon EC2 会删除所有预置快照。删除所有预置快照之后，您可以重新为 AMI 启用 EC2 Fast Launch。

访问客户自主管理型密钥

要为使用客户自主管理型密钥进行加密的加密 AMI 启用 EC2 Fast Launch，您必须授予 AWSServiceRoleForEC2FastLaunch 角色使用 CMK 的权限。为此，请调用 create-grant 命令。对于 `--grantee-principal`，请为账户中的 AWSServiceRoleForEC2FastLaunch 角色指定 ARN。对于 `--operations`，请指定 `CreateGrant`。

```
aws kms create-grant \
    --key-id arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2FastLaunch \
    --operations CreateGrant
```

编辑服务相关角色

Amazon EC2 不允许您编辑 AWSServiceRoleForEC2FastLaunch 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色说明。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除服务相关角色

只有在先删除所有相关资源后，才能删除服务相关角色。这可确保您不会无意中删除相关资源的访问权限，从而保护与启用了 EC2 Fast Launch 的 Amazon EC2 Windows Server AMI 关联的 Amazon EC2 资源。

使用 IAM 控制台、Amazon CLI 或 Amazon API 删除 AWSServiceRoleForEC2FastLaunch 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

支持的区域：

Amazon EC2 在所有开放 Amazon EC2 服务的区域均支持 EC2 Fast Launch 服务相关角色。

排查 Windows EC2 Fast Launch 问题

故障排除场景

以下场景有助于诊断和解决在尝试启用 EC2 Fast Launch 时可能遇到的常见问题。

无法停止实例以创建快照

描述

启用 EC2 Fast Launch 后，该服务会启动一组实例，以用于创建预调配的快照。每个实例有 30 分钟的时间来完成此过程。如果任何实例成功完成，则该服务会将 AMI 的 Fast Launch 状态设置为 `Enabled`。但如果某个实例未能在分配的时间内完成该过程，并且所有其他实例均未完成该过程，则该服务将终止所有实例，将 AMI 的 Fast Launch 状态设置为 `enabling_failed`，并将 Fast Launch 状态原因设置为以下内容：

```
Unable to stop instance ID=i-1234567890abcdef0 for snapshot creation.
```

原因

发生这种情况时，最常见的原因是尝试启用 EC2 Fast Launch 的 Windows AMI 是从正在运行的实例创建的，或尝试启用 EC2 Fast Launch 的 AMI 不满足所有 EC2 Fast Launch 先决条件。

解决方案

确保您使用的 AMI 满足所有 EC2 Fast Launch 先决条件。

要为 AMI 配置 EC2 Fast Launch，您必须使用 Sysprep 和关机选项创建该 AMI。有关更多信息，请参阅 使用 Windows Sysprep 创建 Amazon EC2 AMI。

您已达到 VPC 限制

描述

如果您既未使用启动模板来指定现有 VPC，也没有为账户定义默认 VPC，则该服务会自动创建一个包含 VPC 和其他资源的 Amazon CloudFormation 堆栈，如 EC2 Fast Launch 先决条件中所述。

原因

您的 Amazon Web Services 账户已达到该区域允许的最大 VPC 数量，并且您尚未指定将某个现有 VPC 用于 EC2 Fast Launch。这会导致进程失败。

解决方案

您可以使用以下任一选项来解决此问题：

- 您可以请求提高配额

- 您可以提供一个指定了现有 VPC 的启动模板

要请求增加账户可为每个区域定义的 VPC 数量，请按照以下步骤操作：

1. 访问 https://console.aws.amazon.com/servicequotas/，打开服务配额控制台。
2. 在服务控制台控制面板中，选择 Amazon Virtual Private Cloud（Amazon VPC）。这将会打开 VPC 服务配额。
3. 筛选 `VPCs per Region` 即可直接进入该配额。
4. 选择每个区域的 VPC 数量，然后选择请求在账户级别增加。

如果需要紧急请求增加配额，或者增加配额的请求被拒绝，请联系 Support 寻求帮助。有关更多信息，请参阅《Service Quotas 用户指南》中的请求增加配额。

启用 EC2 Fast Launch 的权限不足

描述

在未指定启动模板的情况下首次启用 EC2 Fast Launch 时，EC2 Fast Launch 会创建一个由服务拥有的 Amazon CloudFormation 堆栈，其中包含服务默认资源。但如果您的 IAM 主体（角色或用户）缺少必要的权限，CloudFormation 模板将无法完成部署。

日志消息可能如以下所示：

```
Can't enable EC2 Fast Launch. The IAM credentials that you are using do not have
 sufficient permissions. Attach EC2FastLaunchFullAccess in the IAM console.
```

原因

您的 IAM 用户或角色缺少启用 EC2 Fast Launch 的必要权限。

解决方案

验证启用 EC2 Fast Launch 的 IAM 主体（用户或角色）是否附加了 `EC2FastLaunchFullAccess` 策略。此 Amazon 托管式策略会授予对所有 EC2 Fast Launch 资源的完全访问权限。要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 EC2FastLaunchFullAccess 策略。

更改 Amazon EC2 实例的 Windows 管理员密码

如果您从 Amazon Windows AMI 启动实例，则预安装的启动代理会按如下方式设置默认密码：

- 对于 Windows Server 2022 及更高版本，EC2Launch v2 会生成默认密码。
- 对于 Windows Server 2016 和 2019，EC2Launch 代理会生成默认密码。
- 对于 Windows Server 2012 R2 及更早版本，EC2Config 服务会生成默认密码。

> **Note**
>
> 对于 Windows Server 2016 及更高版本 AMI，对本地管理员禁用 `Password never expires`。对于 Windows Server 2016 之前的 AMI 版本，对本地管理员启用 `Password never expires`。

在连接后更改管理员密码

在首次连接到实例后，建议您更改管理员密码的默认值。执行以下步骤可更改 Windows 实例的管理员密码。

> **Important**
>
> 将新密码保存在安全位置。您无法使用 Amazon EC2 控制台检索新密码。控制台只能用于检索默认密码。如果您尝试在更改默认密码后使用该密码连接到实例，则会收到错误“Your credentials did not work”。

更改本地管理员密码

1. 连接到实例并打开命令提示符。
2. 运行以下命令。如果您的新密码包含特殊字符，请确保将密码放入双引号之中。

```
net user Administrator "new_password"
```

3. 将新密码保存在安全位置。

更改丢失或过期的密码

如果您丢失了密码或密码过期，则可以生成新密码。有关密码重置步骤，请参阅重置 Amazon EC2 Windows 实例的 Windows 管理员密码。

将可选 Windows Server 组件添加到 Amazon EC2 Windows 实例

要访问并安装可选组件，您必须找到适合您的 Windows Server 版本的正确 EBS 快照，从快照创建卷，并将卷附加到您的实例。

开始前的准备工作

使用 Amazon Web Services 管理控制台或命令行工具获取实例的实例 ID 和可用区。您必须在实例所在的可用区中创建 EBS 卷。

使用以下过程之一，将 Windows Server 组件添加到实例。

Console

向实例添加 Windows 组件

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Snapshots。
3. 从 Filter（筛选条件）栏中，选择 Public Snapshots（公有快照）。
4. 添加 Owner Alias（所有者别名）筛选条件，并选择 amazon。
5. 添加描述筛选条件并输入 **`Windows`**。
6. 按 Enter
7. 选择符合您的系统架构和语言首选项的快照。例如，如果您的实例运行的是 Windows Server 2019，请选择 Windows 2019 English Installation Media。
8. 选择 Actions（操作）、Create volume from snapshot（从快照中创建卷）。
9. 对于可用区，选择与您的 Windows 实例匹配的可用区。选择 Add tag（添加标签），然后为标签键输入 **`Name`**，为标签值指定描述性名称。选择创建卷。
10. 在 Successfully created volume（成功创建的卷）消息（绿色广告条）中，选择刚才创建的卷。
11. 选择 Actions（操作）、Attach Volume（附加卷）。
12. 从 Instance（实例）中，选择实例 ID。
13. 对于 Device name（设备名称），输入连接的设备名称。如果您需要设备名称的帮助，请参阅 Amazon EC2 实例上卷的设备名称。
14. 选择附加卷。
15. 连接到您的实例并将卷置于可用状态。有关更多信息，请参阅《Amazon EBS 用户指南》中的使 Amazon EBS 卷可供使用。

> **Important**
>
> 请勿初始化该卷。

16. 打开 Control Panel、Programs and Features。选择 Turn Windows features on or off。当系统提示安装介质时，则使用安装介质指定 EBS 卷。
17. （可选）完成介质安装后，可以断开卷。断开卷后，可以将其删除。

Amazon CLI

向实例添加 Windows 组件

1. 使用 describe-snapshots 命令以及 `owner-ids` 参数和 `description` 筛选条件，以获取可用安装介质快照的列表。

```
aws ec2 describe-snapshots \
    --owner-ids amazon \
    --filters Name=description,Values=Windows*
```

2. 在输出中，请注意与您的系统架构和语言偏好匹配的快照的 ID。例如：

```
{
    "Snapshots": [
    ...
        {
            "OwnerAlias": "amazon",
            "Description": "Windows 2019 English Installation Media",
            "Encrypted": false,
            "VolumeId": "vol-be5eafcb",
            "State": "completed",
            "VolumeSize": 6,
            "Progress": "100%",
            "StartTime": "2019-10-25T20:00:47.000Z",
            "SnapshotId": "snap-22da283e",
            "OwnerId": "123456789012"
        },
    ...
   ]
}
```

3. 使用 create-volume 命令从快照创建卷。指定与您的实例相同的可用区。

```
aws ec2 create-volume \
    --snapshot-id snap-0abcdef1234567890 \
    --volume-type gp2 \
    --availability-zone us-east-1a
```

4. 在输出中，记下卷 ID。

```
{
    "AvailabilityZone": "us-east-1a",
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-01234567890abcdef",
    "State": "creating",
    "Iops": 100,
    "SnapshotId": "snap-0abcdef1234567890",
    "CreateTime": "2017-04-18T10:33:10.940Z",
    "Size": 6
}
```

5. 使用 attach-volume 命令将卷附加到实例。

```
aws ec2 attach-volume \
    --volume-id vol-0c98b37f30bcbc290 \
    --instance-id i-01474ef662b89480 \
    --device xvdg
```

6. 连接到您的实例并将卷置于可用状态。有关更多信息，请参阅《Amazon EBS 用户指南》中的使 Amazon EBS 卷可供使用。

> **Important**
>
> 请勿初始化该卷。

7. 打开 Control Panel、Programs and Features。选择 Turn Windows features on or off。当系统提示安装介质时，则使用安装介质指定 EBS 卷。

8. （可选）完成介质安装后，使用 detach-volume 命令将卷与实例断开。断开卷后，您可以使用 delete-volume 命令删除该卷。

PowerShell

向实例添加 Windows 组件

1. 使用包含 Owner 和 description 筛选器 Get-EC2Snapshot 获取可用的安装介质快照的列表。

```
Get-EC2Snapshot `
    -Owner amazon `
    -Filter @{ Name="description"; Values="Windows*" }
```

2. 在输出中，请注意与您的系统架构和语言偏好匹配的快照的 ID。例如：

```
...
DataEncryptionKeyId :
Description         : Windows 2019 English Installation Media
Encrypted           : False
KmsKeyId            :
OwnerAlias          : amazon
OwnerId             : 123456789012
Progress            : 100%
SnapshotId          : snap-0abcdef1234567890
StartTime           : 10/25/2019 8:00:47 PM
State               : completed
StateMessage        :
Tags                : {}
VolumeId            : vol-01234567890abcdef
VolumeSize          : 6
...
```

3. 使用 New-EC2Volume cmdlet 从快照创建卷。指定与您的实例相同的可用区。

```
New-EC2Volume `
    -AvailabilityZone us-east-1a `
    -VolumeType gp2 `
    -SnapshotId snap-0abcdef1234567890
```

4. 在输出中，记下卷 ID。

```
Attachments      : {}
AvailabilityZone : us-east-1a
CreateTime       : 4/18/2017 10:50:25 AM
Encrypted        : False
```

```
Iops             : 100
KmsKeyId         :
Size             : 6
SnapshotId       : snap-0abcdef1234567890
State            : creating
Tags             : {}
VolumeId         : vol-01234567890abcdef
VolumeType       : gp2
```

5. 使用 Add-EC2Volume cmdlet 将卷附加到您的实例。

```
Add-EC2Volume `
    -InstanceId i-1234567890abcdef0 `
    -VolumeId vol-01234567890abcdef `
    -Device xvdh
```

6. 连接到您的实例并将卷置于可用状态。有关更多信息，请参阅《Amazon EBS 用户指南》中的使 Amazon EBS 卷可供使用。

> **Important**
>
> 请勿初始化该卷。

7. 打开 Control Panel、Programs and Features。选择 Turn Windows features on or off。当系统提示安装介质时，则使用安装介质指定 EBS 卷。
8. （可选）完成介质安装后，使用 Dismount-EC2Volume cmdlet 将卷与实例断开。断开卷后，您可以使用 Remove-EC2Volume cmdlet 删除该卷。

在 EC2 Windows 实例上安装适用于 Linux 的 Windows 子系统

适用于 Linux 的 Windows 子系统（WSL）是 Microsoft Windows 的一项功能。通过在 EC2 Windows 实例上安装 WSL，您可以直接在 Windows 实例上运行本机 Linux 工具。

适用于 Linux 的 Windows 子系统（WSL）有两个版本：WSL 1 和 WSL 2。有关更多信息，请参阅 Microsoft 网站上适用于 Linux 的 Windows 子系统文档。

要求

- 操作系统必须是 Windows Server 2019 或更高版本。
- 您只能在虚拟化的 Windows 实例上安装 WSL 1（实例大小不为 `.metal` 或不支持嵌套虚拟化）。

- 您可以在支持嵌套虚拟化并启用了 NestedVirtualization CPU 选项的实例上安装 WSL 1 或 WSL 2。
- 您可以在裸机实例（实例大小为 .metal）上安装 WSL 1 或 WSL 2。裸机实例默认为嵌套虚拟化提供所需的支持。

有关 EC2 的嵌套虚拟化的更多信息，请参阅 利用嵌套虚拟化在 Amazon EC2 实例中运行虚拟机监控程序。

在 Windows 实例上安装 WSL

安装 WSL 1

1. 安装 WSL。您将使用的过程取决于实例上运行的 Windows Server 版本。
   - Windows Server 2022 及更高版本：在您的 EC2 实例上运行以下标准安装命令。

     ```
     wsl --install --enable-wsl1 --no-launch
     ```

   - Windows Server 2019：启用 WSL，然后如 Microsoft 网站上的在早期版本的 Windows Server 上安装 WSL 中所述安装 WSL。
2. 重新启动您的 EC2 实例。

   ```
   shutdown -r -t 20
   ```

3. 要将 WSL 配置为使用 WSL 1，请在您的实例上运行以下命令。虚拟化实例需要执行此步骤（实例大小不为 .metal 或未针对嵌套虚拟化进行配置）。

   ```
   wsl --set-default-version 1
   ```

4. 安装默认发行版。

   ```
   wsl --install
   ```

要安装 WSL 2（.metal 或启用了嵌套虚拟化的实例）

在您的 EC2 实例上运行以下标准安装命令。默认情况下会安装 WSL 2。

```
wsl --install
```

EC2 Windows 故障排除实用程序

EC2WinUtil 驱动程序为 Windows 实例提供以下类型的故障排除支持。

崩溃调用堆栈

EC2WinUtil 从实例收集基本崩溃信息并将其写入串行控制台。以下列表包括实用程序写入控制台的一些关键详细信息。

- 识别产生故障的模块。
- 与事件关联的 Windows 错误代码。
- 最近调用的堆栈跟踪。

有了这些详细信息，可以执行初步根本原因分析并确定是否需要进一步分析。输出到串行控制台还使 Amazon 能够跟踪 Amazon EC2 驱动程序的崩溃趋势，并诊断大规模崩溃事件。

> **Note**
>
> EC2WinUtil 不会在其崩溃调用堆栈中收集任何客户数据。

有关驱动程序发行说明，请参阅 EC2 Windows 实用程序驱动程序版本历史记录

EC2 Windows 实用程序驱动程序版本历史记录

下表显示了哪些 EC2WinUtil 驱动程序在 Amazon EC2 上的每个 Windows Server 版本上运行。早期版本的操作系统使用在启动实例的 Amazon Windows Server AMI 上预安装的驱动程序。与您共享的 AMI 或您通过 Amazon Web Services Marketplace 订阅的 AMI 未预安装该驱动程序。

Windows Server 版本	EC2WinUtil 驱动程序版本
Windows Server 2025	最新版本
Windows Server 2022	最新版本
Windows Server 2019	最新版本
Windows Server 2016	最新版本

> **Note**
>
> 在驱动程序版本 3.0.0 之前，无法下载 EC2WinUtil 驱动程序进行手动安装。早期版本仅作为 Amazon Windows AMI 的预安装驱动程序提供。

下表介绍了 EC2WinUtil 驱动程序的发布版本。

程序包版本	驱动程序版本	详细信息	发行日期
3.2.0	3.2.0	用于 EC2Launch v2 代理的增强控制台日志记录机制。	2026 年 6 月 2 日
3.1.1	3.1.1	增加了登录到控制台输出时的调用堆栈长度。	2026 年 3 月 3 日
3.1.0	3.1.0	改进了电源管理事件处理。	2026 年 2 月 4 日
3.0.0	3.0.0	对适用于 Windows 10 的驱动程序进行了现代化改造，并添加了对作为原始驱动程序安装的支持。	2024 年 6 月 13 日
2.0.0	2.0.0	增加了对裸机实例类型的 MMIO 串行端口输出的支持。还改进了崩溃解析并更新了输出格式。	2018 年 8 月 23 日
1.0.1	1.0.1	由于命名空间与 Amazon Inspector 冲突，将驱动程序名称更改为 EC2WinUtil 。包括几个错误修复。	2018 年 3 月 1 日
1.0.0	1.0.0	初始版本。驱动程序最初名为 AwsAgent。	2017 年 11 月 28 日

将 EC2 Windows 实例升级到更高版本的 Windows Server

如果需要将 EC2 Windows 实例上的 Windows Server 操作系统从早期版本升级，则可以使用以下方法之一。

就地升级

在现有实例上执行就地升级。在此过程中，只有操作系统文件会受到影响，设置、服务器角色和数据会保持不变。

迁移（也称为并行升级）

迁移涉及捕获设置、配置和数据，并将这些内容移植到全新 EC2 Windows 实例上的新版操作系统。您可以通过在 Amazon Web Services Marketplace 中订阅的公有或私有 Windows AMI 启动实例，也可以通过与您共享的 AMI 启动实例。您还可以使用 EC2 Image Builder 创建自定义 AMI。有关更多信息，请参阅《Image Builder User Guide》。

> **Note**
>
> Amazon 为 EC2 实例上运行的 Windows Server 版本提供一组公开可用的亚马逊机器映像（AMI）。这些 AMI 按月更新。有关最新 Windows AMI 的信息，请参阅《AmazonWindows AMI Reference》。

传统上，Microsoft 会建议迁移到新版本的 Windows Server，而不是就地升级。虽然迁移所产生的升级错误或问题更少，但所需的时间可能多于就地升级所需的时间，因为需要预置新实例、规划和移植应用程序以及调整新实例上的配置设置。就地升级的速度更快，但软件不兼容可能会产生错误。

内容

- 在 EC2 Windows 实例上执行就地升级
- 使用自动化运行手册升级 EC2 Windows 实例
- 将 EC2 Windows 实例迁移到基于 Nitro 的实例类型
- 对 EC2 Windows 实例上的操作系统升级进行问题排查

在 EC2 Windows 实例上执行就地升级

在执行就地升级之前，您必须确定此实例正在运行的网络驱动程序。半虚拟化网络驱动程序使您能够使用远程桌面访问实例。实例使用 Amazon PV、Intel Network Adapter 或增强联网驱动程序。有关更多信息，请参阅 Windows 实例的半虚拟化驱动程序。

就地升级开始前的准备工作

开始就地升级之前，请完成以下任务并注意以下重要详细信息。

- 阅读 Microsoft 文档了解更新要求、已知问题和限制。还应参阅有关升级的正式说明。
  - Windows Server 2012 的升级选项
  - Windows Server 2012 R2 的升级选项
  - Windows Server 2016 及更高版本的升级和转换选项
  - Windows Server 升级
- 我们建议在具有至少 2 个 vCPU 和 4GB RAM 的实例上执行操作系统升级。如果需要，您可以将实例更改为相同类型的更大尺寸（例如，将 t2.small 更改为 t2.large），执行升级，然后将其调整为原始大小。如果您需要保留实例大小，则可以使用实例控制台屏幕截图监控进度。有关更多信息，请参阅Amazon EC2 实例类型更改。
- 确认 Windows 实例上的根卷有足够的可用磁盘空间。Windows 安装过程可能不会发出磁盘空间不足的警告。有关升级特定操作系统所需的磁盘空间的信息，请参阅 Microsoft 文档。如果卷没有足够空间，可将其扩展。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 弹性卷。
- 确定升级方法。您必须将操作系统升级到相同架构。例如，必须将 32 位系统升级到 32 位系统。Windows Server 2008 R2 及更高版本是纯粹的 64 位系统。
- 禁用反病毒和反间谍软件和防火墙。这些类型的软件可与升级程序发生冲突。在完成升级后，请重新启用反病毒和反间谍软件以及防火墙。
- 更新到最新驱动程序，如将 EC2 Windows 实例迁移到基于 Nitro 的实例类型中所述。
- 升级帮助程序服务仅支持运行 Citrix PV 驱动程序的实例。如果实例运行的是 Red Hat 驱动程序，您必须先手动升级这些驱动程序。

使用 Amazon PV、Intel Network Adapter 或增强联网驱动程序就地升级实例

通过以下步骤，使用 Amazon PV 、Intel Network Adapter 或增强联网驱动程序升级 Windows Server 实例。

执行就地升级

1. 出于备份或测试目的，创建计划升级的系统的 AMI。然后，您可以对副本执行升级以模拟测试环境。如果升级完成，就可以在停机时间很短的情况下将流量切换到此实例。如果升级失败，则可以转至备份。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。
2. 确保您的 Windows Server 实例使用最新网络驱动程序。
   a. 要更新您的 Amazon PV 驱动程序，请参阅 在 EC2 Windows 实例上升级半虚拟化驱动程序。
   b. 要更新您的 ENA 驱动程序，请参阅 在 EC2 Windows 实例上安装 ENA 驱动程序。
   c. 要更新您的 Intel 驱动程序，请参阅 通过 Intel 82599 VF 接口实现增强联网

3. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
4. 在导航窗格中，选择实例。找到该实例。记下该实例的实例 ID 和可用区。您在此过程的稍后部分需要此信息。
5. 如果您要从 Windows Server 2012 或 2012 R2 升级到 Windows Server 2016 或更高版本，请在继续操作之前在您的实例上执行以下操作。

   a. 卸载 EC2Config 服务。有关更多信息，请参阅 EC2Launch v2 和 EC2Config 代理的 Windows 服务管理。

   b. 安装 EC2Launch v1 或 EC2Launch v2 代理。

   c. 安装 Amazon Systems Manager SSM Agent 有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的在适用于 Windows Server 的 Amazon EC2 实例上手动安装 SSM Agent。
6. 从 Windows Server 安装介质快照创建新卷。

   a. 在左导航窗格中的 Elastic Block Store (弹性数据块存储) 下，选择 Snapshots (快照)。

   b. 从筛选条件栏中，选择公有快照。

   c. 在搜索栏中，指定以下筛选条件：

      - 依次选择拥有者别名、= 和 amazon。
      - 选择描述，然后开始键入 **`Windows`**。选择符合您要升级到的系统架构和语言首选项的 Windows 筛选条件。例如，选择 Windows 2019 English Installation Media 以升级到 Windows Server 2019。

   d. 选中符合要升级到的系统架构和语言首选项的快照旁的复选框，然后选择操作、从快照创建卷。

   e. 在创建卷页面中，选择与您的 Windows 实例匹配的可用区，然后选择创建卷。
7. 在页面顶部的已成功创建卷 vol-***`1234567890example`*** 横幅中，选择刚刚创建的卷的 ID。
8. 选择 Actions（操作）、Attach Volume（附加卷）。
9. 在挂载卷页面上，对于实例，选择您的 Windows 实例的实例 ID，然后选择挂载卷。
10. 按照使 Amazon EBS 卷可供使用中的步骤使新卷可供使用。

> **Important**
>
> 请勿对磁盘进行初始化，因为这样做将会删除现有的数据。

11. 在 Windows PowerShell 中，切换到新的卷驱动器。通过打开已附加到实例的安装介质卷，开始升级。

    a. 如果您升级到 Windows Server 2016 或更高版本，请执行以下操作：

    ```
    .\setup.exe /auto upgrade /dynamicupdate disable
    ```

    > **Note**
    >
    > 运行 setup.exe 并将 /dynamicupdate 选项设置为禁用，将会阻止 Windows 在 Windows Server 升级过程中安装更新，因为在升级过程中安装更新可能会导致失败。升级完成后，您可以使用 Windows 更新安装更新。

    如果您升级到较早的 Windows Server 版本，请运行以下操作：

    ```
    Sources\setup.exe
    ```

    b. 对于选择要安装的操作系统，选择您的 Windows Server 实例的完整安装选项，然后选择下一步。

    c. 对于 Which type of installation do you want? (您需要什么类型的安装?)，选择 Upgrade (升级)。

    d. 完成向导。

Windows Server 安装程序将复制并处理文件。几分钟后，远程桌面会话关闭。升级所用的时间取决于在 Windows Server 实例上运行的应用程序和服务器角色的数量。升级过程少则 40 分钟，多则数小时。在升级过程中，实例可能无法通过一项或两项状态检查。升级完成后，可以通过所有状态检查。您可以检查系统日志中的控制台输出，或使用磁盘和 CPU 活动的 Amazon CloudWatch 指标以确定升级是否正在进行。

> **Note**
>
> 如果升级到 Windows Server 2019，在升级完成后，如果需要，您可以手动更改桌面背景以删除以前的操作系统名称。

如果实例在数小时后还未通过所有状态检查，请参阅对 EC2 Windows 实例上的操作系统升级进行问题排查。

升级后的任务

1. 登录实例以启动 .NET Framework 的升级并在提示时重启系统。
2. 如果您尚未在之前的步骤中执行此操作，请安装 EC2Launch v1 或 EC2Launch v2 代理。
3. 如果您已升级到 Windows Server 2012 R2，我们建议您将 PV 驱动程序升级到 Amazon PV 驱动程序。如果您在基于 Nitro 的实例上进行升级，我们建议您安装或升级 NVME 和 ENA 驱动程序。有关更多信息，请参阅 Amazon NVMe 驱动程序 或 在 Windows 上启用增强联网。
4. 重新启用反病毒和反间谍软件以及防火墙。

使用自动化运行手册升级 EC2 Windows 实例

您可以使用 Amazon Systems Manager Automation 运行手册在 Amazon 上执行 Windows 和 SQL Server 实例的自动升级。

内容

- 相关服务
- 执行选项
- 升级 Windows Server
- 升级 SQL Server

相关服务

将在自动升级过程中使用以下 Amazon 服务：

- Amazon Systems Manager。Amazon Systems Manager 是一个功能强大的统一界面，可以集中管理您的 Amazon 资源。有关更多信息，请参阅 Amazon Systems Manager 用户指南。
- Amazon Systems Manager 代理（SSM Agent）是一个 Amazon 软件，可以在 Amazon EC2 实例、本地服务器或虚拟机 (VM) 上安装和配置。SSM Agent 让 Systems Manager 可以更新、管理和配置这些资源。代理在Amazon云中处理 Systems Manager 服务的请求，然后按照请求中的指定运行它们。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的使用 SSM Agent。
- Amazon Systems Manager SSM 运行手册。SSM 运行手册定义 Systems Manager 对您的托管实例执行的操作。SSM 运行手册使用 JavaScript Object Notation (JSON) 或 YAML，并包括您指定

的步骤和参数。本主题使用两个 Systems Manager SSM 自动化运行手册。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的 Amazon Systems Manager Automation 运行手册参考。

执行选项

在 Systems Manager 控制台上选择自动化时，请选择执行。在选择 Automation 文档后，将提示您选择自动化执行选项。您可以从以下选项中进行选择。在本主题后面提供的途径步骤中，我们使用简单执行选项。

简单执行

如果要更新单个实例，但不希望执行每个自动化步骤以审核结果，请选择该选项。在下面的升级步骤中更详细地介绍了该选项。

速率控制

如果要将升级应用于多个实例，请选择该选项。您可以定义以下设置。

- 参数

  该设置（也会在“多账户和区域”设置中设置）定义了如何完成自动化。
- 目标

  选择要将自动化应用到的目标。也会在“多账户和区域”设置中设置该设置。
- 参数值

  使用自动化文档参数中定义的值。
- 资源组

  在Amazon中，资源是您可以使用的实体。示例包括 Amazon EC2 实例、Amazon CloudFormation 堆栈或 Amazon S3 存储桶。如果您使用多个资源，这样做可能会很有帮助：将它们作为一个组进行管理，而不是在每个任务中将资源从一个Amazon服务移动到另一个服务。在某些情况下，您可能希望管理大量相关的资源，例如，构成应用程序层的 EC2 实例。在这种情况下，您可能需要同时对这些资源执行批量操作。
- 标签

  标签帮助您按不同的方式对 Amazon 资源进行分类，例如，按用途、所有者或环境进行分类。如果您具有很多相同类型的资源，这种分类是非常有用的。您可以使用分配的标签快速识别特定的资源。
- 速率控制

也会在“多账户和区域”设置中设置速率控制。在设置速率控制参数时，您可以按目标数或队列百分比定义要将自动化应用到的队列数量。

多账户和区域

除了在“速率控制”中指定的参数（也会在“多账户和区域”设置中使用）以外，还具有两个其他设置：

- 账户和组织单位 (OU)

  指定要在其中运行自动化的多个账户。
- Amazon Web Services 区域

  指定要在其中运行自动化的多个 Amazon Web Services 区域。

手动执行

该选项类似于简单执行，但允许您逐个执行每个自动化步骤并审核结果。

升级 Windows Server

AWSEC2-CloneInstanceAndUpgradeWindows 运行手册从您的账户中的 Windows Server 实例中创建亚马逊机器映像（AMI），并将该 AMI 升级到所选的支持版本。该多步骤过程最多可能需要两小时才能完成。

在自动升级过程中包含两个 AMI：

- 当前运行的实例。第一个 AMI 是当前运行的实例，不会升级该实例。该 AMI 用于启动另一个实例以运行就地升级。在该过程完成后，将从您的账户中删除该 AMI，除非您明确要求保留原始实例。该设置是由 `KeepPreUpgradeImageBackUp` 参数处理的（默认值为 `false`，这表示默认删除该 AMI）。
- 升级的 AMI。该 AMI 是自动化过程的结果。

最终结果是一个 AMI，它是升级的 AMI 实例。

在升级完成后，您可以在 Amazon VPC 中启动新的 AMI 以测试应用程序功能。在测试后，在执行另一个升级之前，请计划应用程序停机，然后再完全切换到升级的实例。

先决条件

要使用 Amazon Systems Manager Automation 文档自动进行 Windows Server 升级，您必须执行以下任务：

- 使用指定的 IAM policy 创建 IAM 角色，以允许 Systems Manager 在 Amazon EC2 实例上执行自动化任务，并验证您是否满足使用 Systems Manager 的先决条件。有关更多信息，请参阅 Amazon Identity and Access Management 用户指南中的创建向 Amazon 服务委派权限的角色。
- 选择您希望如何运行自动化的选项。执行选项包括简单执行、速率控制、多账户和区域以及手动执行。有关这些选项的详细信息，请参阅执行选项。
- 验证实例上是否安装了 SSM Agent。有关更多信息，请参阅在 Amazon EC2 实例中为 Windows Server 安装和配置 SSM Agent。
- 必须在您的实例上安装 Windows PowerShell 3.0 或更高版本。
- 对于加入到某个 Microsoft Active Directory 域的实例，建议指定一个没有连接到您的域控制器的 `SubnetId`，以帮助避免主机名冲突。
- 实例子网必须具有通向互联网的出站连接，这样可以访问 Amazon S3 等 Amazon Web Services 服务 以及从 Microsoft 下载补丁。如果子网是公有子网且实例具有公有 IP 地址，或者子网是私有子网并使用路由将互联网流量发送到公有 NAT 设备，即满足此要求。
- 此自动化适用于运行 Windows Server 2008 R2、Windows Server 2012 R2、Windows Server 2016 和 Windows Server 2019 的实例。
- 验证实例的启动盘具有 20 GB 的可用磁盘空间。
- 如果实例未使用 Amazon 提供的 Windows 许可证，请指定包含 Windows Server 2012 R2 安装介质的 Amazon EBS 快照 ID。要实现此目的，应按照以下步骤进行：
  1. 验证 Amazon EC2 实例运行的是否是 Windows Server 2012 或更高版本。
  2. 在运行实例的同一可用区中创建一个 6GB 的 Amazon EBS 卷。将卷附加到实例。例如，将其附加为驱动器 D。
  3. 例如，右键单击 ISO 并将其挂载为实例的驱动器 E。
  4. 将 ISO 的内容从驱动器 E:\ 复制到驱动器 D:\
  5. 为上面步骤 2 中创建的 6GB 卷创建 Amazon EBS 快照。

Windows Server 升级限制

此自动化程序不支持升级 Windows 域控制器、集群或 Windows 桌面操作系统。此外，该自动化程序不支持安装了以下角色的 Windows Server 的 Amazon EC2 实例：

- 远程桌面会话主机 (RDSH)
- 远程桌面连接代理 (RDCB)
- 远程桌面虚拟化主机 (RDVH)
- 远程桌面 Web 访问 (RDWA)

执行 Windows Server 自动升级的步骤

请按照以下步骤使用 Amazon Web Services EC2-CloneInstanceAndUpgradeWindows 自动化运行手册升级您的 Windows Server 实例。

1. 从Amazon管理控制台打开 Systems Manager。
2. 从左侧导航窗格中，在 Change Management（变更管理）下，选择 Automation。
3. 选择执行自动化。
4. 搜索名为 `AWSEC2-CloneInstanceAndUpgradeWindows` 的自动化文档。
5. 在显示该文档名称时，选择该文档。在选择该文档时，将显示文档详细信息。
6. 选择 Execute automation（执行自动化）以输入此文档的参数。在页面顶部选择简单执行。
7. 根据以下准则输入请求的参数。

   - `InstanceID`

     类型：字符串

     （必需）运行 Windows Server 2008 R2、2012 R2、2016 或 2019 且安装有 SSM Agent 的实例。

   - `InstanceProfile.`

     类型：字符串

     （必需）IAM 实例配置文件。这是用于针对 Amazon EC2 实例和 Amazon AMI 执行 Systems Manager 自动化的 IAM 角色。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的配置 EC2 实例权限。

   - `TargetWindowsVersion`

     类型：字符串

     （必需）选择目标 Windows 版本。

   - `SubnetId`

类型：字符串

（必需）这是执行升级过程的子网以及源 EC2 实例所在的位置。验证子网是否具有到 Amazon 服务（包括 Amazon S3）和 Microsoft 的出站连接（以下载补丁）。

- `KeepPreUpgradedBackUp`

  类型：字符串

  （可选）如果该参数设置为 `true`，自动化将保留从实例中创建的映像。默认设置为 `false`。

- `RebootInstanceBeforeTakingImage`

  类型：字符串

  （可选）默认值为 `false`（无重新引导）。如果该参数设置为 `true`，Systems Manager 将在创建升级 AMI 之前重新引导实例。

8. 在输入这些参数后，选择执行。在自动化开始时，您可以监控执行进度。
9. 在自动化完成时，您将看到 AMI ID。您可以启动 AMI 以验证是否升级了 Windows 操作系统。

> **Note**
>
> 自动化不需要运行所有步骤。根据自动化和实例行为，可以有条件地执行这些步骤。Systems Manager 可能会跳过一些不需要的步骤。
> 此外，一些步骤可能会超时。Systems Manager 尝试升级并安装所有最新的补丁。不过，根据给定步骤的可定义超时设置，补丁有时会超时。在发生这种情况时，Systems Manager Automation 继续执行下一步，以确保将内部操作系统升级到目标 Windows Server 版本。

10. 在自动化完成后，您可以使用 AMI ID 启动 Amazon EC2 实例以检查升级。有关如何从 Amazon AMI 中创建 Amazon EC2 实例的更多信息，请参阅如何从自定义 AMI 启动 EC2 实例？

升级 SQL Server

AWSEC2-CloneInstanceAndUpgradeSQLServer 脚本从您账户中运行 SQL Server 的 Amazon EC2 实例中创建一个 AMI，然后将该 AMI 升级到较新版本的 SQL Server。该多步骤过程最多可能需要两小时才能完成。

在该工作流程中，自动化从实例中创建一个 AMI，然后在您提供的子网中启动新的 AMI。接下来，自动化执行 SQL Server 的就地升级。在升级完成后，自动化创建新的 AMI，然后再终止升级的实例。

在自动升级过程中包含两个 AMI：

- 当前运行的实例。第一个 AMI 是当前运行的实例，不会升级该实例。该 AMI 用于启动另一个实例以运行就地升级。在该过程完成后，将从您的账户中删除该 AMI，除非您明确要求保留原始实例。该设置是由 `KeepPreUpgradeImageBackUp` 参数处理的（默认值为 `false`，这表示默认删除该 AMI）。
- 升级的 AMI。该 AMI 是自动化过程的结果。

最终结果是一个 AMI，它是升级的 AMI 实例。

在升级完成后，您可以在 Amazon VPC 中启动新的 AMI 以测试应用程序功能。在测试后，在执行另一个升级之前，请计划应用程序停机，然后再完全切换到升级的实例。

先决条件

要使用 Amazon Systems Manager Automation 文档自动进行 SQL Server 升级，您必须执行以下任务：

- 使用指定的 IAM policy 创建 IAM 角色，以允许 Systems Manager 在 Amazon EC2 实例上执行自动化任务，并验证您是否满足使用 Systems Manager 的先决条件。有关更多信息，请参阅《Amazon Identity and Access Management 用户指南》中的创建向 Amazon Web Services 服务 委派权限的角色。
- 选择您希望如何运行自动化的选项。执行选项包括简单执行、速率控制、多账户和区域以及手动执行。有关这些选项的详细信息，请参阅执行选项。
- Amazon EC2 实例必须使用 Windows Server 2008 R2 或更高版本和 SQL Server 2008 或更高版本。
- 验证实例上是否安装了 SSM Agent。有关更多信息，请参阅在适用于 Windows Server 的 Amazon EC2 实例上使用 SSM Agent。
- 确认实例有足够的可用磁盘空间：
  - 如果要从 Windows Server 2008 R2 升级到 2012 R2，或者从 Windows Server 2012 R2 升级到更高版本的操作系统，请确认实例启动磁盘中有 20GB 的可用磁盘空间。
  - 如果要从 Windows Server 2008 R2 升级到 2016 或更高版本，请确认实例启动磁盘中有 40GB 的可用磁盘空间。
- 对于使用自带许可 (BYOL) SQL Server 版本的实例，以下额外的先决条件适用：
  - 提供包含目标 SQL Server 安装介质的 Amazon EBS 快照 ID。要实现此目的，应按照以下步骤进行：

1. 验证 Amazon EC2 实例运行的是否是 Windows Server 2008 R2 或更高版本。
2. 在运行实例的同一可用区中创建一个 6GB 的 Amazon EBS 卷。将卷附加到实例。例如，将其附加为驱动器 D。
3. 例如，右键单击 ISO 并将其挂载为实例的驱动器 E。
4. 将 ISO 的内容从驱动器 E:\ 复制到驱动器 D:\
5. 为步骤 2 中创建的 6GB 卷创建 Amazon EBS 快照。

SQL Server 自动升级限制

使用 Amazon Web Services EC2-CloneInstanceAndUpgradeSQLServer 运行手册执行自动升级时适用以下限制：

- 只能在使用 Windows 身份验证的 SQL Server 上执行升级。
- 验证实例上没有待处理的安全补丁更新。打开控制面板，然后选择检查更新。
- 不支持 HA 和镜像模式下的 SQL Server 部署。

执行 SQL Server 自动升级的步骤

请按照以下步骤使用 Amazon Web Services EC2-CloneInstanceAndUpgradeSQLServer 自动化运行手册升级您的 SQL Server。

1. 如果尚未挂载，请下载 SQL Server 2016 .iso 文件并将其挂载到源服务器中。
2. 在挂载该 .iso 文件后，复制所有组件文件，并将其放在所选的任何卷上。
3. 拍摄该卷的 Amazon EBS 快照，并将快照 ID 复制到剪贴板以供以后使用。有关更多信息，请参阅《Amazon EBS User Guide》中的 Create Amazon EBS snapshots。
4. 将实例配置文件附加到 Amazon EC2 源实例。这允许 Systems Manager 与 EC2 实例通信，并在将其添加到 Amazon Systems Manager 服务后在其中运行命令。对于该示例，我们将角色命名为 `SSM-EC2-Profile-Role` 并将 `AmazonSSMManagedInstanceCore` 策略附加到该角色。
5. 在 Amazon Systems Manager 控制台的左侧导航窗格中，选择托管实例。验证您的 EC2 实例是否位于托管实例列表中。如果在几分钟后没有看到您的实例，请参阅 Amazon Systems Manager 用户指南中的我的实例在哪里？。
6. 在左侧导航窗格中，在 Change Management（变更管理）下，选择 Automation。
7. 选择执行自动化。
8. 搜索名为 `AWSEC2-CloneInstanceAndUpgradeSQLServer` 的自动化文档。

9. 选择 AWSEC2-CloneInstanceAndUpgradeSQLServer SSM 文档，然后选择 Next（下一步）。
10. 确保选择了简单执行选项。
11. 根据以下准则输入请求的参数。

    - InstanceId

      类型：字符串

      （必需）运行 SQL Server 2008 R2（或更高版本）的实例。

    - IamInstanceProfile

      类型：字符串

      （必需）IAM 实例配置文件。

    - SQLServerSnapshotId

      类型：字符串

      （必需）目标 SQL Server 安装介质的快照 ID。对于包含 SQL Server 许可证的实例，此参数不是必需的。

    - SubnetId

      类型：字符串

      （必需）这是执行升级过程的子网以及源 EC2 实例所在的位置。验证子网是否具有到 Amazon 服务（包括 Amazon S3）和 Microsoft 的出站连接（以下载补丁）。

    - KeepPreUpgradedBackUp

      类型：字符串

      （可选）如果该参数设置为 true，自动化将保留从实例中创建的映像。默认设置为 false。

    - RebootInstanceBeforeTakingImage

      类型：字符串

      （可选）默认值为 false（无重新引导）。如果该参数设置为 true，Systems Manager 将在创建升级 AMI 之前重新引导实例。

    - TargetSQLVersion

类型：字符串

（可选）目标 SQL Server 版本。默认为 2016。

12. 在输入这些参数后，选择执行。在自动化开始时，您可以监控执行进度。
13. 在 Execution status（执行状态）显示为 Success（成功）时，展开 Outputs（输出）以查看 AMI 信息。您可以使用 AMI ID 为所选的 VPC 启动 SQL Server 实例。
14. 打开 Amazon EC2 控制台。在左侧导航窗格中，选择 AMI。将会看到新的 AMI。
15. 要验证是否成功安装了新的 SQL Server 版本，请选择新的 AMI，然后选择 Launch（启动）。
16. 为该 AMI 选择所需的实例类型、要部署到的 VPC 和子网以及要使用的存储。由于您从 AMI 中启动新的实例，因此，将以选项形式向您提供这些卷以包含在启动的新 EC2 实例中。您可以删除其中的任何卷，也可以添加卷。
17. 添加标签以帮助您识别实例。
18. 将一个或多个安全组添加到实例中。
19. 选择 Launch Instance。
20. 选择实例的标签名称，然后在操作下拉列表中选择连接。
21. 验证 SQL Server 新版本是否为新实例上的数据库引擎。

将 EC2 Windows 实例迁移到基于 Nitro 的实例类型

Amazon Windows AMI 配置了 Microsoft 安装介质使用的默认设置，并进行一些自定义。自定义内容包括支持基于 Nitro的实例（例如 M5 和 C5）的驱动程序和配置。

在从基于 Xen 的实例迁移到基于 Nitro 的实例（包括裸机实例）时，对于下列情况，建议您按照本主题中的步骤操作：

- 从自定义 Windows AMI 启动实例
- 从 Amazon 提供的 Windows AMI（2018 年 8 月之前创建）启动实例

或者，您可以使用 AWSSupport-UpgradeWindowsAWSDrivers 自动化文档自动完成第 1 部分、第 2 部分和第 3 部分中所述的过程。如果您选择使用自动过程，请先阅读（替代方法）使用 Amazon 升级 Amazon Systems Manager PV、ENA 和 NVMe 驱动程序，然后再继续执行第 4 部分和第 5 部分。

> **Note**
>
> 可以在 Windows Server 2016 及更高版本上执行以下迁移过程。已终止使用的早期操作系统版本未经测试，可能与最新的实例类型不兼容。
> 要迁移 Linux 实例，请参阅 the section called “实例类型更改”。

目录

- 第 1 部分：安装和升级 Amazon PV 驱动程序
- 第 2 部分：安装和升级 ENA
- 第 3 部分：升级 Amazon NVMe 驱动程序
- 第 4 部分：更新 EC2Config 和 EC2Launch
- 第 5 部分：为裸机实例安装串行端口驱动程序
- 第 6 部分：更新电源管理设置
- 第 7 部分：为新实例类型更新 Intel 芯片组驱动程序
- （替代方法）使用 Amazon 升级 Amazon Systems Manager PV、ENA 和 NVMe 驱动程序

开始之前

此过程假定您拥有基于 Xen 的实例（例如 M4 或 C4），并且要迁移到基于 Nitro 的实例。

您必须使用 PowerShell 3.0 或更高版本才能成功执行升级。

> **Note**
>
> 迁移时，现有网络接口卡上的静态 IP 或自定义 DNS 网络设置可能会丢失，因为实例会默认使用新的增强联网适配器设备。

在按照本过程中的步骤操作之前，建议您先创建实例的备份。从 EC2 控制台中，选择需要迁移的实例，打开上下文（右键单击）菜单，然后依次选择 Instance State (实例状态) 和 Stop (停止)。

> **Warning**
>
> 当您停止某个实例时，任何实例存储卷上的数据都将被擦除。要保留实例存储卷上的数据，请确保将这些数据备份到持久性存储中。

在 EC2 控制台中打开实例的上下文（右键单击）菜单，选择 Image (映像)，然后选择 Create Image (创建映像)。

> **Note**
>
> 本说明的第 4 部分和第 5 部分可以在您迁移或更改实例类型后完成。但是，如果要迁移到裸机实例类型，则建议您在迁移前完成这些步骤。

第 1 部分：安装和升级 Amazon PV 驱动程序

尽管 Amazon PV 驱动程序在 Nitro 系统中未使用，但如果您使用的是早期版本的 Citrix PV 或 Amazon PV，应当仍升级它们。最新的 Amazon PV 驱动程序解决了在以下情况下可能在早期版本的驱动程序中出现的错误：您使用的是 Nitro 系统或者需要迁移回基于 Xen 的实例。作为最佳实践，我们建议始终针对 Amazon 上的 Windows 实例升级到最新驱动程序。

使用以下过程对 Amazon PV 驱动程序执行就地升级，或在 Windows Server 2008 R2、Windows Server 2012、Windows Server 2012 R2、Windows Server 2016 或 Windows Server 2019 上从 Citrix PV 驱动程序升级到 Amazon PV 驱动程序。有关更多信息，请参阅在 EC2 Windows 实例上升级半虚拟化驱动程序。

要升级域控制器，请参阅 升级域控制器 (Amazon PV 升级)。

执行升级或升级到 Amazon PV 驱动程序

1. 使用远程桌面连接到实例并进行实例升级准备。在执行该升级前使所有非系统磁盘脱机。如果您要对 Amazon PV 驱动程序执行就地升级，则不需要此步骤。在 Services 控制台中将不必要的服务设置为手动 启动。
2. 将最新驱动程序包下载到实例。
3. 提取文件夹的内容，然后运行 `AWSPVDriverSetup.msi`。

在运行 MSI 后，实例将自动重启，然后升级驱动程序。实例可能将有长达 15 分钟的时间不可用。

在升级完成并且实例在 Amazon EC2 控制台中通过了两项运行状况检查后，请使用远程桌面连接到实例并验证新驱动程序是否已安装。在设备管理器中的存储控制器下，找到 Amazon PV 存储主适配器。确认驱动程序版本与驱动程序版本历史记录表中列出的最新版本相同。有关更多信息，请参阅Amazon PV 驱动程序包历史记录。

第 2 部分：安装和升级 ENA

升级到最新的 Elastic Network Adapter 驱动程序以确保所有网络功能均受支持。如果您启动了实例，而该实例尚未启用增强联网，则必须在该实例上下载并安装所需的网络适配器驱动程序。然后，设置 enaSupport 实例属性以激活增强联网。仅当安装了 ENA 驱动程序时，才能在受支持的实例类型上启用此属性。有关更多信息，请参阅在 EC2 实例上使用 ENA 启用增强联网功能。

1. 将最新驱动程序下载到实例。
2. 提取 zip 存档。
3. 通过运行提取的文件夹中的 `install.ps1` PowerShell 脚本安装驱动程序。

   > **Note**
   >
   > 要避免安装错误，请以管理员身份运行 `install.ps1` 脚本。

4. 检查 AMI 是否已激活 enaSupport。如果未启用，请按照 在 EC2 实例上使用 ENA 启用增强联网功能 上的文档继续操作。

第 3 部分：升级 Amazon NVMe 驱动程序

Amazon NVMe 驱动程序用于与显示为 Nitro 系统中的 NVMe 块储存设备的 Amazon EBS 和 SSD 实例存储卷交互以提高性能。

> **Important**
>
> 以下说明专门针对以下情况进行了修改：您在基于 Xen 的实例上安装或升级 Amazon NVMe，以将实例迁移到基于 Nitro 的实例。

1. 将最新驱动程序包下载到实例。

   如果您需要先前版本的驱动程序，请参阅 NVMe Windows 驱动程序版本了解支持的版本。
2. 提取 zip 存档。
3. 如 `Readme.txt` 中所述安装驱动程序。
4. 打开 PowerShell 会话并运行以下命令：

```
PS C:\> start rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
```

> **Note**
>
> 要应用该命令，您必须以管理员身份运行 PowerShell 会话。PowerShell (x86) 版本将会导致错误。
> 此命令仅在设备驱动程序上运行 sysprep。它不会运行完整的 sysprep 准备。

5. 对于 Windows Server 2008 R2 和 Windows Server 2012，请关闭实例，更改实例类型并启动实例，然后继续完成第 4 部分。如果在迁移到基于 Nitro 的实例类型之前，您在基于 Xen 的实例类型上再次启动了实例，则将无法启动。对于其他支持的 Windows AMI，您可以在运行设备 sysprep 后的任何时间更改实例类型。

第 4 部分：更新 EC2Config 和 EC2Launch

对于 Windows 实例，当在 Nitro 系统上（包括在 EC2 裸机上）运行时，最新的 EC2Config 和 EC2Launch 实用工具将提供额外的功能和信息。默认情况下，EC2Config 服务包含在早于 Windows Server 2016 的 AMI 中。EC2Launch 将取代 Windows Server 2016 及更高版本 AMI 上的 EC2Config。

EC2Config 和 EC2Launch 服务更新后，Amazon 提供的新 Windows AMI 将包含最新版本的该服务。不过，您必须将自己的 Windows AMI 和实例更新为最新版本的 EC2Config 和 EC2Launch。

安装或更新 EC2Config

1. 下载并解压缩 EC2Config 安装程序。
2. 运行 `EC2Install.exe`。有关选项的完整列表，请使用 `EC2Install` 选项运行 `/?`。默认情况下，安装程序会显示提示。要运行该命令而不显示提示，请使用 `/quiet` 选项。

有关更多信息，请参阅安装最新版的 EC2Config。

安装或更新 EC2Launch

1. 如果已在实例上安装和配置 EC2Launch，请备份 EC2Launch 配置文件。安装过程不保留此文件中的更改。默认情况下，该文件位于以下 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` 目录中。
2. 将 EC2-Windows-Launch.zip 下载到实例上的一个目录中。
3. 将 install.ps1 下载到 `EC2-Windows-Launch.zip` 的下载目录中。

将 install.ps1 下载到 EC2-Windows-Launch.zip 的下载目录中。

4. 运行 install.ps1。

> **Note**
>
> 要避免安装错误，请以管理员身份运行 install.ps1 脚本。

5. 如果您对 EC2Launch 配置文件进行了备份，则将其复制到 C:\ProgramData\Amazon\EC2-Windows\Launch\Config 目录。

有关更多信息，请参阅使用 EC2Launch v1 代理在 EC2 Windows 实例启动期间执行任务。

第 5 部分：为裸机实例安装串行端口驱动程序

i3.metal 实例类型使用基于 PCI 的串行设备而不是基于 I/O 端口的串行设备。最新 Windows AMI 将自动使用基于 PCI 的串行设备并安装串行端口驱动程序。如果您没有使用从 Amazon 提供的 Windows AMI（日期为 2018 年 4 月 11 日或更晚）启动的实例，则必须安装串行端口驱动程序来启用串行设备的 EC2 功能（如密码生成和控制台输出）。最新的 EC2Config 和 EC2Launch 实用工具还支持 i3.metal 并提供额外的功能。按照第 4 部分中的步骤操作（如果尚未执行）。

安装串行端口驱动程序

1. 将串行驱动程序包下载到实例。
2. 提取文件夹的内容，打开 aws_ser.INF 的上下文（右键单击）菜单，然后选择 install (安装)。
3. 选择 Okay (确定)。

第 6 部分：更新电源管理设置

以下电源管理设置更新将显示器设置为从不关闭，从而允许在 Nitro 系统上正常关闭操作系统。Amazon 截至 2018 年 11 月 28 日提供的所有 Windows AMI 都已具有此默认配置。

1. 打开命令提示符或 PowerShell 会话。
2. 运行以下命令：

```
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
```

```
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f71556f20b4a 7516b95f-
f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbcb2b7e 0
```

第 7 部分：为新实例类型更新 Intel 芯片组驱动程序

`u-6tb1.metal`、`u-9tb1.metal` 和 `u-12tb1.metal` 实例类型使用的硬件需要先前未安装在 Windows AMI 上的芯片组驱动程序。如果不使用从 Amazon 提供的 Windows AMI（日期为 2018 年 11 月 19 日或更晚）启动的实例，则必须使用 Intel Chipset INF Utility 安装驱动程序。

安装芯片组驱动程序

1. 将芯片组 INF 实用程序下载到实例。
2. 提取文件。
3. 运行 `SetupChipset.exe`。
4. 接受 Intel 软件许可协议并安装芯片组驱动程序。
5. 重启实例。

（替代方法）使用 Amazon 升级 Amazon Systems Manager PV、ENA 和 NVMe 驱动程序

`AWSSupport-UpgradeWindowsAWSDrivers` 自动化文档自动完成第 1 部分、第 2 部分和第 3 部分中所述的步骤。此方法还可在驱动程序升级失败时修复实例。

`AWSSupport-UpgradeWindowsAWSDrivers` 自动化文档在指定的 EC2 实例上升级或修复存储及网络 Amazon 驱动程序。该文档尝试通过调用 Amazon 代理（SSM Agent），在线安装最新版本的 Amazon Systems Manager 驱动程序。如果无法与 SSM Agent 通信，则在明确要求时，该文档可以执行 Amazon 驱动程序的离线安装。

> **Note**
>
> 此过程将在域控制器上失败。要在域控制器上更新驱动程序，请参阅升级域控制器 (Amazon PV 升级)。

使用 Amazon 自动升级 Amazon Systems Manager PV、ENA 和 NVMe 驱动程序

1. 在 https://console.aws.amazon.com/systems-manager 处打开 Systems Manager 控制台。
2. 选择 Automation (自动化)、Execute Automation (执行自动化)。
3. 搜索并选择 AWSSupport-UpgradeWindowsAWSDrivers 自动化文档，然后选择执行自动化。
4. 在输入参数部分，配置以下选项：

   实例 ID

   输入要升级实例的唯一 ID。

   AllowOffline

   （可选）选择下列选项之一：

   - `True` — 选择此选项可执行离线安装。在升级过程中，实例将停止并重新启动。

     > **Warning**
     > 当您停止某个实例时，任何实例存储卷上的数据都将被擦除。要保留实例存储卷上的数据，请确保将这些数据备份到持久性存储中。

   - `False` —（默认值）要执行在线安装，请保持选中此选项。在升级过程中，实例将重新启动。

     > **Important**
     > 在尝试升级操作之前，在线和离线升级会创建一个 AMI。该 AMI 在自动化完成后仍将存在。请确保您可以访问该 AMI，或者在不再需要时将其删除。

   SubnetId

   （可选）输入以下值之一：

   - `SelectedInstanceSubnet` —（默认值）升级过程会在要升级实例所在的同一子网中启动帮助程序 实例。该子网必须允许与 Systems Manager 终端节点 (`ssm.*`) 进行通信。
   - `CreateNewVPC` — 升级过程会在新 VPC 中启动帮助程序 实例。如果您不确定目标实例的子网是否允许与 `ssm.*` 终端节点进行通信，请使用此选项。您的用户必须具有创建 VPC 的权限。

- 特定子网 ID — 指定要在其中启动帮助程序 实例的特定子网的 ID。该子网必须与要升级的实例位于同一可用区中，并且必须允许与 `ssm.*` 终端节点进行通信。

5. 选择执行。
6. 留出完成升级的时间。完成在线升级可能需要长达 10 分钟的时间，而完成离线升级可能需要长达 25 分钟的时间。

对 EC2 Windows 实例上的操作系统升级进行问题排查

Amazon 利用升级帮助程序服务 (一种可帮助您执行涉及 Citrix PV 驱动程序的就地升级的 Amazon 实用程序) 提供对升级问题的支持。

升级后，实例可能临时 遇到高于平均 CPU 使用率的情况，而 .NET 运行时优化服务将优化 .NET 框架。这是预期行为。

如果实例在数小时后未通过所有状态检查，请执行以下检查。

- 如果已升级到 Windows Server 2008，但在数小时后所有状态检查均失败，则升级可能已失败，这时会显示 Click OK 提示以确认回滚。因为在这种状态下无法访问控制台，所以无法单击该按钮。要解决此问题，请通过 Amazon EC2 控制台或 API 执行重启。重启需要 10 分钟或更长时间才能开始。实例可能在 25 分钟后变为可用。
- 从服务器删除应用程序或服务器角色，然后重试。

如果从服务器删除应用程序或服务器角色后实例未通过所有状态检查，请执行以下操作。

- 停止实例，将根卷附加到其他实例。有关更多信息，请参阅"等待元数据服务"中有关如何停止根卷并将其附加到其他实例的描述。
- 分析 Windows 安装程序日志文件和事件日志中有无失败。

如对操作系统升级或迁移有其他问题或疑问，我们建议您查看就地升级开始前的准备工作中所列的文章。

Tutorial: Connect an Amazon EC2 instance to an Amazon RDS database（教程：将 Amazon EC2 实例连接到 Amazon RDS 数据库）

教程目标

本教程的目标是学习如何使用 Amazon Web Services 管理控制台 配置 Amazon EC2 实例与 Amazon RDS 数据库之间的安全连接。

有不同的选项用于配置连接。在本教程中，我们将探讨以下三个选项：

- 选项 1：使用 EC2 控制台将实例自动连接到 RDS 数据库

  使用 EC2 控制台中的自动连接功能自动配置 EC2 实例与 RDS 数据库之间的连接，以允许 EC2 实例与 RDS 数据库之间的流量。
- 选项 2：使用 RDS 控制台将实例自动连接到 RDS 数据库

  使用 RDS 控制台中的自动连接功能自动配置 EC2 实例与 RDS 数据库之间的连接，以允许 EC2 实例与 RDS 数据库之间的流量。
- 选项 3：通过创建安全组，将实例手动连接到 RDS 数据库

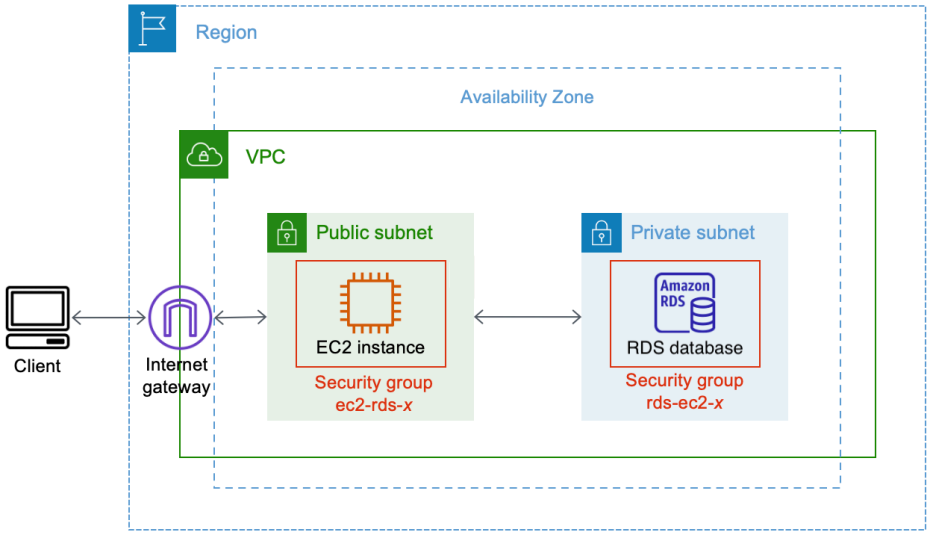
  通过手动配置和分配安全组来配置 EC2 实例与 RDS 数据库之间的连接，以重现由“选项 1”和“选项 2”中的自动连接功能自动创建的配置。

上下文

作为您为何要在 EC2 实例与 RDS 数据库之间配置连接的背景，让我们考虑以下场景：您的网站向您的用户提供一份表单以供填写。您需要在数据库中捕获表单数据。您可以在已配置为 Web 服务器的 EC2 实例上托管您的网站，也可以在 RDS 数据库中捕获该表单数据。EC2 实例和 RDS 数据库需要相互连接，以便表单数据可以从 EC2 实例传输到 RDS 数据库。本教程介绍如何配置该连接。请注意，这只是连接 EC2 实例与 RDS 数据库的使用案例的一个示例。

架构

下图显示了创建的资源以及完成本教程中的所有步骤后生成的架构配置。



该图说明了您将创建的以下资源：

- 您将在同一 Amazon Web Services 区域、VPC 和可用区中创建 EC2 实例及 RDS 数据库。
- 您将在公有子网中创建 EC2 实例。
- 您将在私有子网中创建 RDS 数据库。

  当您使用 RDS 控制台创建 RDS 数据库并自动连接 EC2 实例时，将自动选择数据库的 VPC、数据库（DB）子网组和公有访问设置。将在与 EC2 实例相同的 VPC 内的私有子网中自动创建 RDS 数据库。
- 互联网用户可以通过互联网网关使用 SSH 或 HTTP/HTTPS 连接到 EC2 实例。
- 互联网用户无法直接连接到 RDS 数据库；只有 EC2 实例连接到 RDS 数据库。
- 当您使用自动连接功能允许 EC2 实例与 RDS 数据库之间的流量时，将自动创建并添加以下安全组：
  - 将创建安全组 ec2-rds-***x*** 并将其添加到 EC2 实例中。它包含一条出站规则，引用 rds-ec2-***x*** 安全组作为其目标。这将允许来自 EC2 实例的流量通过 rds-ec2-***x*** 安全组到达 RDS 数据库。
  - 将创建安全组 rds-ec2-***x*** 并将其添加到 RDS 数据库中。它包含一条入站规则，引用 ec2-rds-***x*** 安全组作为其来源。这将允许来自 EC2 实例（具有 ec2-rds-***x*** 安全组）的流量到达 RDS 数据库。

  通过使用独立的安全组（一个用于 EC2 实例，一个用于 RDS 数据库），您可以更好地控制实例和数据库的安全性。如果您要在实例和数据库上使用同一安全组，然后将该安全组修改为仅适合（比如）数据库，则该修改将同时影响实例和数据库。换言之，如果您要使用一个安全组，则可能会无意中修改资源（实例或数据库）的安全性，因为您忘记了该安全组已连接到该资源。

自动创建的安全组也遵循最低权限，因为它们仅允许通过创建特定于工作负载的安全组对，在数据库端口上为此工作负载建立相互连接。

注意事项

在完成本教程中的任务时，请注意以下几点：

- 两个控制台 - 在本教程中，您将使用以下两个控制台：
  - Amazon EC2 控制台 - 您将使用 EC2 控制台启动实例，将 EC2 实例自动连接到 RDS 数据库，以及使用手动选项通过创建安全组来配置连接。
  - Amazon RDS 控制台 - 您将使用 RDS 控制台创建 RDS 数据库，然后将 EC2 实例自动连接到 RDS 数据库。
- 一个 VPC - 要使用自动连接功能，EC2 实例和 RDS 数据库必须位于同一 VPC 中。

  如果您要手动配置 EC2 实例与 RDS 数据库之间的连接，可以在一个 VPC 中启动 EC2 实例，在另一个 VPC 中启动 RDS 数据库；但是，您需要设置额外的路由和 VPC 配置。本教程中未涵盖此场景。
- 一个 Amazon Web Services 区域 - EC2 实例和 RDS 数据库必须位于同一区域中。
- 两个安全组 - EC2 实例与 RDS 数据库之间的连接由两个安全组配置 - 一个安全组用于 EC2 实例，一个安全组用于 RDS 数据库。

  当您使用 EC2 控制台或 RDS 控制台中的自动连接功能配置连接（本教程的“选项 1”和“选项 2”）时，将自动创建安全组并将其分配给 EC2 实例和 RDS 数据库。

  如果您不使用自动连接功能，则需要手动创建和分配安全组。您可以在本教程的“选项 3”中执行此操作。

完成教程所需的时间

30 分钟

您可以一口气完成整个教程，也可以一次完成一项任务。

成本

您在完成本教程过程中创建的 Amazon 资源可能会产生费用。

如果 EC2 实例和 RDS 数据库位于不同的可用区中，则需要支付数据传输费。为了避免产生这些费用，EC2 实例和 RDS 数据库必须位于同一可用区中。有关数据传输费的信息，请参阅“Amazon EC2 On-Demand Pricing”（Amazon EC2 按需定价）页面上的 Data Transfer（数据传输）。

为了防止在您完成教程后产生费用，请确保在不再需要资源时将其删除。有关删除资源的步骤，请参阅任务 4（可选）：清理。

选项 1：使用 EC2 控制台将实例自动连接到 RDS 数据库

“选项 1”的目标是探索 EC2 控制台中的自动连接功能，该功能可自动配置 EC2 实例与 RDS 数据库之间的连接，以允许从 EC2 实例到 RDS 数据库的流量。在“选项 3”中，您将学习如何手动配置连接。

任务

- 开始前的准备工作
- 任务 1：（可选）创建 RDS 数据库
- 任务 2：（可选）启动 EC2 实例
- 任务 3：将 EC2 实例自动连接到 RDS 数据库
- 任务 4：验证连接配置
- 任务 5（可选）：清理

开始前的准备工作

完成本教程需要做好以下准备：

- 与 EC2 实例位于同一 VPC 中的 RDS 数据库。您可以使用现有的 RDS 数据库，也可以按照“任务 1”中的步骤创建新的 RDS 数据库。
- 与 RDS 数据库位于同一 VPC 中的 EC2 实例。您可以使用现有的 EC2 实例，也可以按照“任务 2”中的步骤创建新的 EC2 实例。
- 调用以下操作的权限：
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`

- ec2:DescribeNetworkInterfaces
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:ModifyNetworkInterfaceAttribute
- ec2:RevokeSecurityGroupEgress

任务 1：（可选）创建 RDS 数据库

> **Note**
>
> 创建 Amazon RDS 数据库不是本教程的重点。如果您已拥有 RDS 数据库并希望在本教程中使用它，则可跳过此任务。
> 如果您使用现有的 RDS 数据库，请确保它与 EC2 实例位于同一 VPC 中，以便您可以使用自动连接功能。

此任务的目标是创建 RDS 数据库，以便您可以完成“任务 3”，在该任务中，您将配置 EC2 实例与 RDS 数据库之间的连接。此任务中的步骤将按以下方式配置 RDS 数据库：

- 引擎类型：MySQL
- 模板：免费套餐
- 数据库实例标识符：**tutorial-database-1**
- 数据库实例类：db.t3.micro

> **Important**
>
> 在生产环境中，您应该配置数据库，以满足您的特定需求。

创建 MySQL RDS 数据库

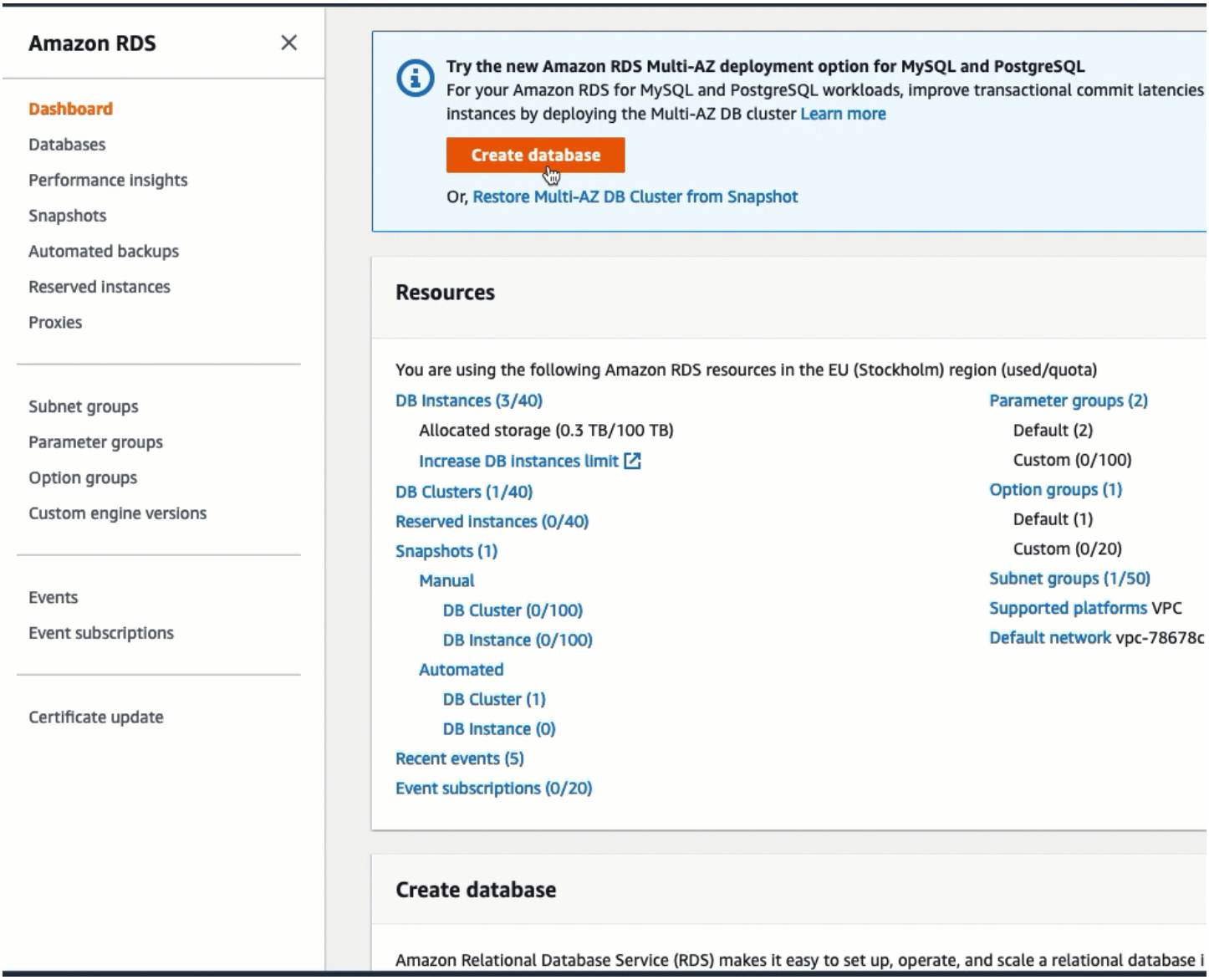
1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 从“Region selector”（区域选择器）（位于右上角）中选择一个 Amazon Web Services 区域。数据库和 EC2 实例必须位于同一区域中，才能使用 EC2 控制台中的自动连接功能。

3. 在控制面板中，选择 Create database（创建数据库）。
4. 在 Choose a database creation method（选择数据库创建方法）下，检查并确保选择 Standard create（标准创建）。如果您选择 Easy create（轻松创建），则 VPC 选择器将不可用。您必须确保您的数据库与 EC2 实例位于同一 VPC 中，才能使用 EC2 控制台中的自动连接功能。
5. 在 Engine options（引擎选项）下，对于 Engine type（引擎类型），选择 MySQL。
6. 在 Templates（模板）下，选择一个满足您需求的示例模板。对于本教程，选择 Free tier（免费套餐）以免费创建 RDS 数据库。但请注意，免费套餐仅适用于符合免费套餐资格的账户。您可以通过选择 Free tier（免费套餐）框中的 Info（信息）链接来阅读更多内容。
7. 在 Settings (设置) 下，执行下列操作：
   a. 对于 DB instance identifier（数据库实例标识符），输入数据库的名称。在本教程中，请输入 **`tutorial-database-1`**。
   b. 对于 Master username（主用户名），保留默认名称，即 **`admin`**。
   c. 对于 Master password（主密码），输入您在本教程中可以记住的密码，然后对于 Confirm password（确认密码），再次输入该密码。
8. 在实例配置下，对于数据库实例类，保留默认值，即 db.t3.micro。
9. 在 Connectivity（连接）下，对于 Compute resource（计算资源），选择 Don't connect to an EC2 compute resource（不连接到 EC2 计算资源），因为您稍后将在“任务 3”中连接 EC2 实例和 RDS 数据库。

   （稍后，在本教程的“选项 2”中，您将通过选择 Connect to an EC2 compute resource（连接到 EC2 计算资源），以试用 RDS 控制台中的自动连接功能。）
10. 对于 Virtual private cloud (VPC)（虚拟私有云（VPC）），选择一个 VPC。该 VPC 必须具有一个数据库子网组。要使用自动连接功能，EC2 实例和 RDS 数据库必须位于同一 VPC 中。
11. 对于此窗格中的其他字段，请保留所有默认值。
12. 选择创建数据库。

    在 Databases（数据库）屏幕上，在数据库准备就绪可供使用之前，新数据库的 Status（状态）为 Creating（正在创建）。当状态变为 Available（可用）时，您便可以连接到该数据库。根据数据库类和存储量，新数据库可能需要等待多达 20 分钟时间才可用。

观看动画：创建 RDS 数据库



任务 2：（可选）启动 EC2 实例

> **Note**
>
> 启动实例不是本教程的重点。如果您已拥有 Amazon EC2 实例并希望在本教程中使用它，则可跳过此任务。
>
> 如果您使用现有的 EC2 实例，请确保它与您的 RDS 数据库位于同一 VPC 中，以便您可以使用自动连接功能。

此任务的目标是启动 EC2 实例，以便您可以完成“任务 3”，在该任务中，您将配置 EC2 实例与 Amazon RDS 数据库之间的连接。此任务中的步骤将按以下方式配置 EC2 实例：

- 实例名称：**`tutorial-instance-1`**
- AMI：Amazon Linux 2
- 实例类型：`t2.micro`
- 自动分配公有 IP：已启用
- 具有以下三条规则的安全组：
  - 允许来自您的 IP 地址的 SSH
  - 允许来自任何地方的 HTTPS 流量
  - 允许来自任何地方的 HTTP 流量

> **Important**
>
> 在生产环境中，您应该配置实例，以满足您的特定需求。

启动 EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从“Region selector”（区域选择器）（位于右上角）中选择一个 Amazon Web Services 区域。实例与 RDS 数据库必须位于同一区域中，才能使用 EC2 控制台中的自动连接功能。
3. 在 EC2 Dashboard（EC2 控制面板）上，选择 Launch instance（启动实例）。
4. 在 Name and tags（名称与标签）下，对于 Name（名称），输入名称以标识您的实例。对于本教程，请将该实例命名为 **`tutorial-instance-1`**。虽然实例名称不是必填项，但是当您在 EC2 控制台中选择实例时，该名称将帮助您轻松识别它。
5. 在 Application and OS Images（应用程序和操作系统映像）下，选择满足您的 Web 服务器需求的 AMI。本教程使用 Amazon Linux 2。
6. 在 Instance type（实例类型）下，对于 Instance type（实例类型），选择满足您的 Web 服务器需求的实例类型。本教程使用的是 `t2.micro`。
7. 在 Key pair (login)（密钥对（登录））下，对于 Key pair name（密钥对名称），选择您的密钥对。
8. 在 Network settings（网络设置）下，执行以下操作：

a. 对于 Network（网络）和 Subnet（子网），如果您尚未更改默认 VPC 或子网，则可以保留默认设置。

   如果您对默认 VPC 或子网进行了更改，请检查以下各项：

   i. 实例必须与 RDS 数据库位于同一 VPC 中，才能使用自动连接功能。默认情况下，您只有一个 VPC。

   ii. 您要将实例启动到其中的 VPC 必须连接互联网网关，以便您能从互联网访问您的 Web 服务器。您的默认 VPC 将自动设置互联网网关。

   iii. 为了确保您的实例收到公有 IP 地址，对于 Auto-assign public IP（自动分配公有 IP），请检查并确保选择 Enable（启用）。如果选择 Disable（禁用），请选择 Edit（编辑）（位于 Network Settings（网络设置）右侧），然后对于 Auto-assign public IP（自动分配公有 IP），选择 Enable（启用）。

b. 要使用 SSH 连接到您的实例，您需要一条安全组规则，授权来自您计算机的公有 IPv4 地址的 SSH (Linux) 或 RDP (Windows) 流量。默认情况下，当您启动实例时，将使用允许来自任何地方的入站 SSH 流量的规则创建新的安全组。

   要确保只有您的 IP 地址才能连接到您的实例，请在防火墙（安全组）下，从允许入站 SSH 流量复选框旁边的下拉列表中，选择我的 IP。

c. 要允许流量从互联网流向您的实例，请选中以下复选框：

   - Allow HTTPs traffic from the internet（允许来自互联网的 HTTPs 流量）
   - Allow HTTP traffic from the internet（允许来自互联网的 HTTP 流量）

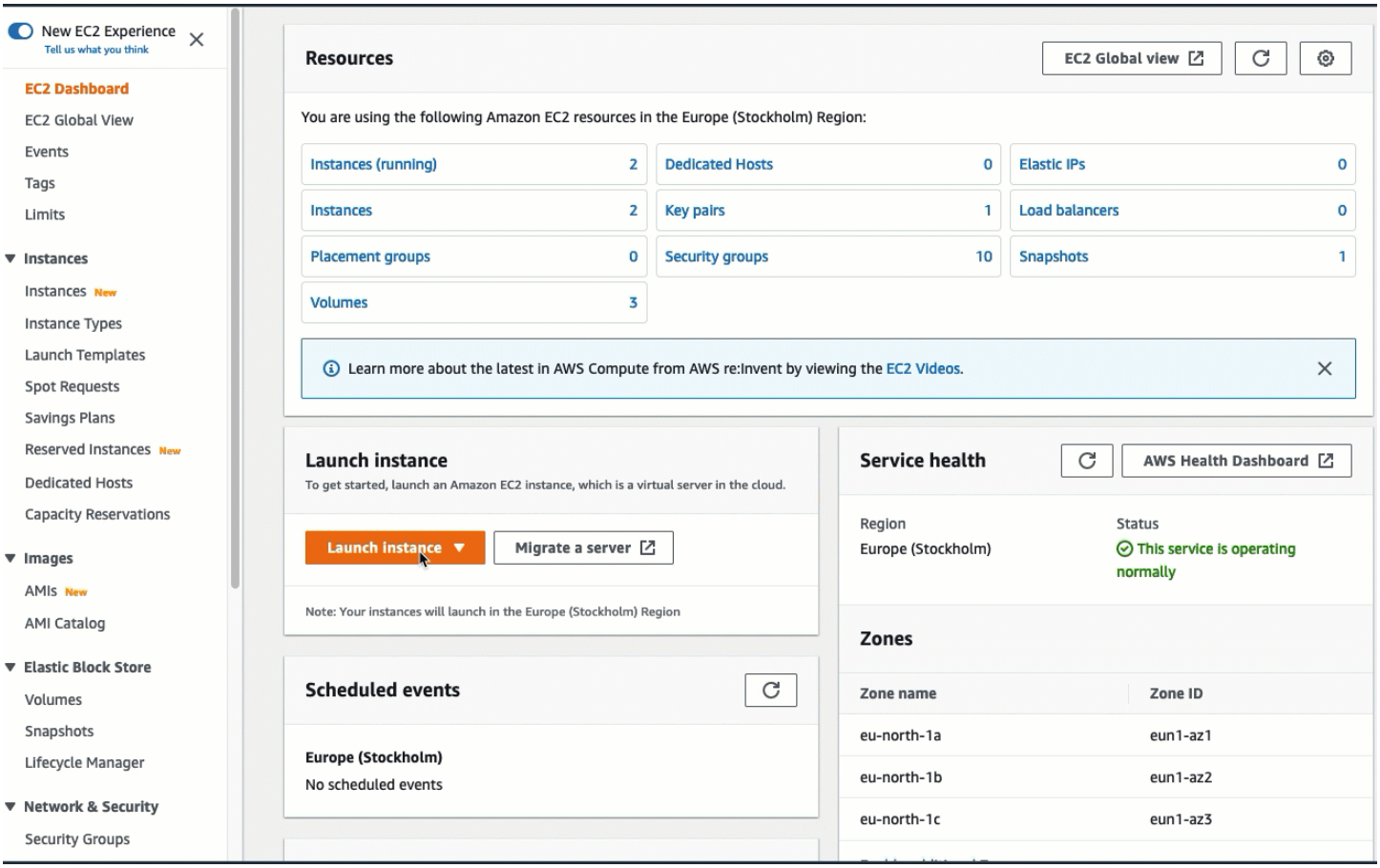
9. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。

10. 保持确认页面处于打开状态。对于下一个任务，当您将实例自动连接到数据库时，您将需要该页面。

    如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

有关启动实例的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

观看动画：启动 EC2 实例



任务 3：将 EC2 实例自动连接到 RDS 数据库

此任务的目标是使用 EC2 控制台中的自动连接功能自动配置 EC2 实例与 RDS 数据库之间的连接。

使用 EC2 控制台将 EC2 实例自动连接到 RDS 数据库

1. 在实例启动确认页面（它应从上一个任务中打开）上，选择 Connect an RDS database（连接 RDS 数据库）。

   如果您关闭了该确认页面，请按以下步骤操作：

   a. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

   b. 在导航窗格中，选择 Instances (实例)。

   c. 选择您刚刚创建的 EC2 实例，然后选择 Actions（操作）、Networking（联网）、Connect RDS database（连接 RDS 数据库）。

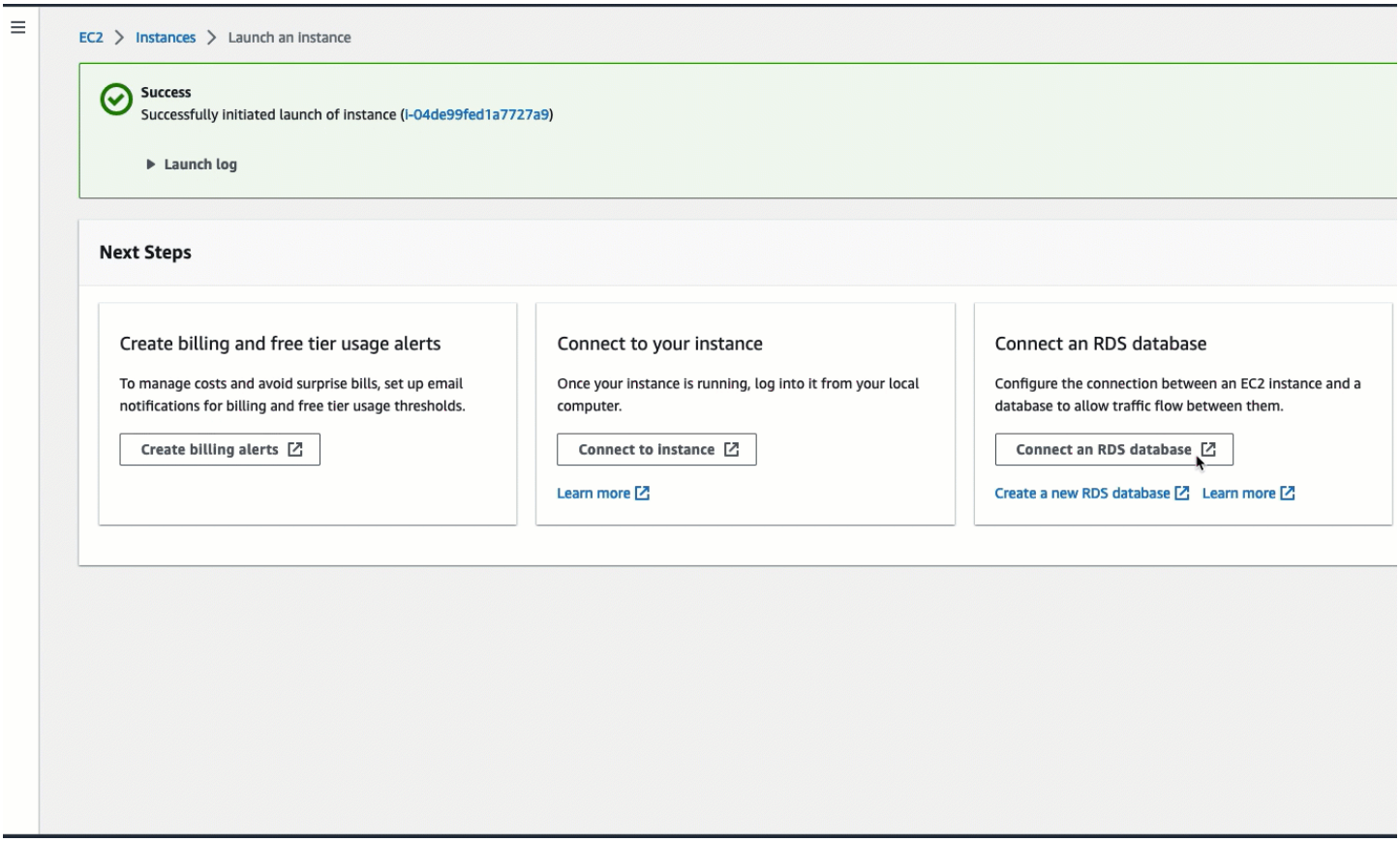
如果 Connect RDS database（连接 RDS 数据库）不可用，请检查该 EC2 实例是否处于 Running（正在运行）状态。

2. 对于 Database role（数据库角色），选择 Instance（实例）。本例中的实例是指数据库实例。
3. 对于 RDS database（RDS 数据库），选择您在“任务 1”中创建的 RDS 数据库。

> **Note**
>
> EC2 实例和 RDS 数据库必须位于同一 VPC 中，以便相互连接。

4. 选择连接。

观看动画：将新启动的 EC2 实例自动连接到 RDS 数据库



任务 4：验证连接配置

此任务的目标是验证两个安全组是否已创建并分配给实例和数据库。

当您使用控制台中的自动连接功能配置连接时，将自动创建安全组并将其分配给实例和数据库，如下所示：

- 将创建安全组 rds-ec2-*x* 并将其添加到 RDS 数据库中。它包含一条入站规则，引用 ec2-rds-*x* 安全组作为其来源。这将允许来自 EC2 实例（具有 ec2-rds-*x* 安全组）的流量到达 RDS 数据库。
- 将创建安全组 ec2-rds-*x* 并将其添加到 EC2 实例中。它包含一条出站规则，引用 rds-ec2-*x* 安全组作为其目标。这将允许来自 EC2 实例的流量通过 rds-ec2-*x* 安全组到达 RDS 数据库。

使用控制台验证连接配置

1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 在导航窗格中，选择 Databases（数据库）。
3. 选择您为本教程创建的 RDS 数据库。
4. 在连接和安全性选项卡的安全性、VPC 安全组下，验证并确保显示名为 rds-ec2-*x* 的安全组。
5. 选择 rds-ec2-*x* 安全组。EC2 控制台中的 Security Groups（安全组）屏幕打开。
6. 选择 rds-ec2-*x* 安全组以打开它。
7. 选择入站规则选项卡。
8. 验证并确保以下安全组规则存在，如下所示：

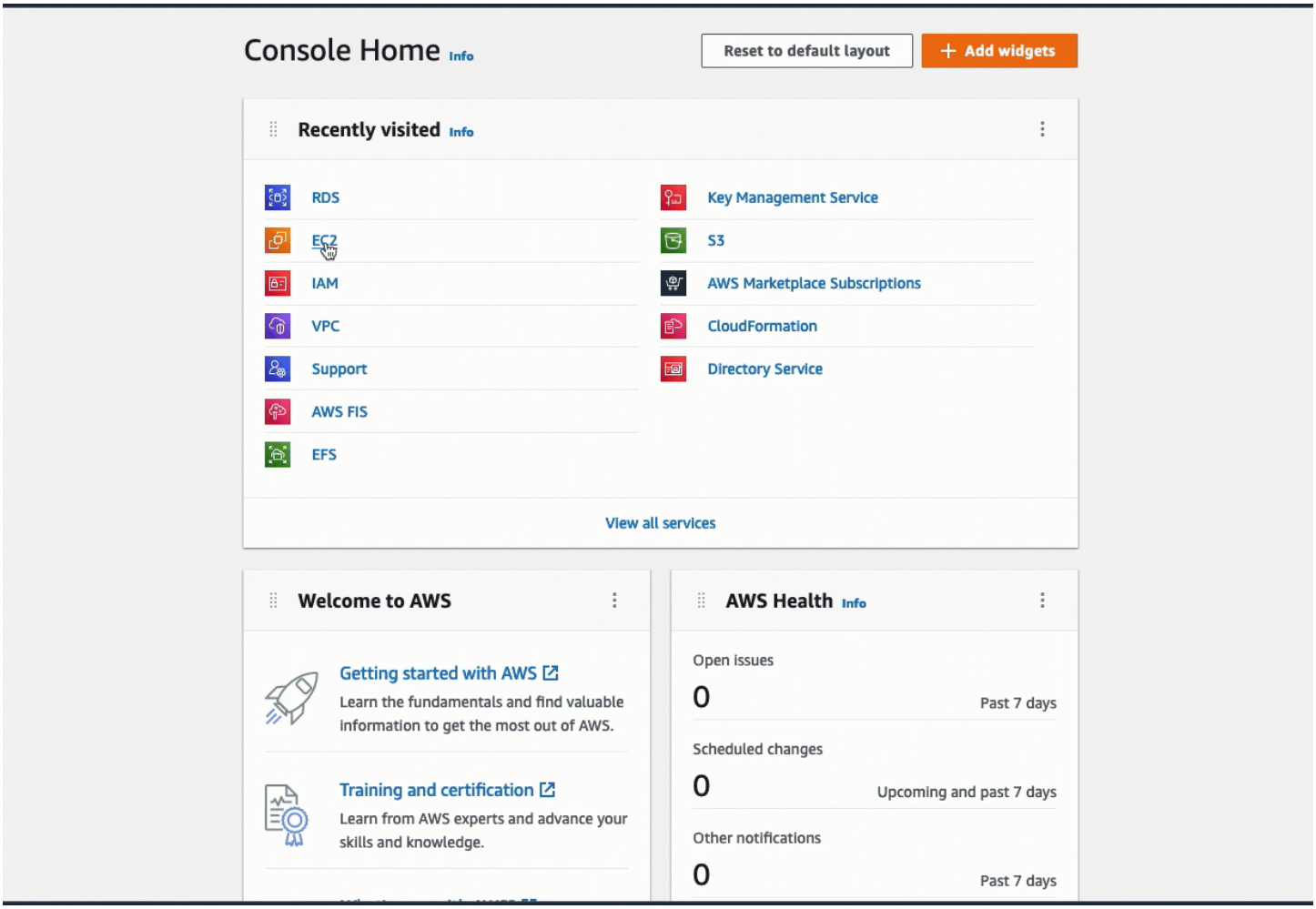
   - 类型：MYSQL/Aurora
   - 端口范围：3306
   - 来源：***sg-0987654321example*** / ec2-rds-*x* - 这是分配给您在前面的步骤中验证的 EC2 实例的安全组。
   - 描述：允许来自 EC2 实例的连接的规则，附有 ***sg-1234567890example***
9. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
10. 在导航窗格中，选择 Instances (实例)。
11. 选择您在上一个任务中选择要连接到 RDS 数据库的 EC2 实例，然后选择 Security（安全性）选项卡。
12. 在安全详细信息的安全组下，验证并确保列表中有名为 ec2-rds-*x* 的安全组。*x* 是一个数字。
13. 选择 ec2-rds-*x* 安全组以打开它。
14. 选择 Outbound rules（出站规则）选项卡。
15. 验证并确保以下安全组规则存在，如下所示：

    - 类型：MYSQL/Aurora

- 端口范围：3306
- 目标：*sg-1234567890example* / rds-ec2-*x*
- 描述：允许从此安全组所连接的任何实例到 **database-tutorial** 的连接的规则

通过验证这些安全组和安全组规则是否存在，以及它们是否已按照此过程中所述分配给 RDS 数据库和 EC2 实例，您可以使用自动连接功能验证是否已自动配置连接。

观看动画：验证连接配置



您已完成本教程的“选项 1”。现在，您可以完成“选项 2”，它将介绍如何使用 RDS 控制台将 EC2 实例自动连接到 RDS 数据库；您也可以完成“选项 3”，它将介绍如何手动配置在“选项 1”中自动创建的安全组。

任务 5（可选）：清理

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

现在您已完成该教程，最好清理（删除）所有您不再需要使用的资源。清理 Amazon 资源可以防止您的账户产生任何进一步的费用。

如果您专门为本教程启动了 EC2 实例，则可以将其终止，以停止产生任何与之相关的费用。

使用控制台终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您为本教程创建的实例，然后依次选择 Instance state（实例状态）、Terminate instance（终止实例）。
4. 当系统提示您确认时，选择终止。

如果您专门为本教程创建了 RDS 数据库，则可以将其删除，以停止产生任何与之相关的费用。

使用控制台删除 RDS 数据库

1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 在导航窗格中，选择 Databases (数据库)。
3. 选择您为本教程创建的 RDS 数据库，然后依次选择 Actions（操作）、Delete（删除）。
4. 在方框中输入 **`delete me`**，然后选择 Delete（删除）。

选项 2：使用 RDS 控制台将实例自动连接到 RDS 数据库

“选项 2”的目标是探索 RDS 控制台中的自动连接功能，该功能可自动配置 EC2 实例与 RDS 数据库之间的连接，以允许从 EC2 实例到 RDS 数据库的流量。在“选项 3”中，您将学习如何手动配置连接。

任务

- 开始前的准备工作
- 任务 1：（可选）启动 EC2 实例
- 任务 2：创建 RDS 数据库并将其自动连接到 EC2 实例
- 任务 3：验证连接配置
- 任务 4（可选）：清理

开始前的准备工作

完成本教程需要做好以下准备：

- 与 RDS 数据库位于同一 VPC 中的 EC2 实例。您可以使用现有的 EC2 实例，也可以按照“任务 1”中的步骤创建新实例。
- 调用以下操作的权限：
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

任务 1：（可选）启动 EC2 实例

> **Note**
>
> 启动实例不是本教程的重点。如果您已拥有 Amazon EC2 实例并希望在本教程中使用它，则可跳过此任务。

此任务的目标是启动 EC2 实例，以便您可以完成“任务 2”，在该任务中，您将配置 EC2 实例与 Amazon RDS 数据库之间的连接。此任务中的步骤将按以下方式配置 EC2 实例：

- 实例名称：**`tutorial-instance-2`**
- AMI：Amazon Linux 2
- 实例类型：`t2.micro`
- 自动分配公有 IP：已启用
- 具有以下三条规则的安全组：
  - 允许来自您的 IP 地址的 SSH
  - 允许来自任何地方的 HTTPS 流量
  - 允许来自任何地方的 HTTP 流量

> **Important**
>
> 在生产环境中，您应该配置实例，以满足您的特定需求。

启动 EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在 EC2 Dashboard（EC2 控制面板）上，选择 Launch instance（启动实例）。
3. 在 Name and tags（名称与标签）下，对于 Name（名称），输入名称以标识您的实例。对于本教程，请将该实例命名为 **`tutorial-instance-2`**。虽然实例名称不是必填项，但是当您在 RDS 控制台中选择实例时，该名称将帮助您轻松识别它。
4. 在 Application and OS Images（应用程序和操作系统映像）下，选择满足您的 Web 服务器需求的 AMI。本教程使用 Amazon Linux。

5. 在 Instance type（实例类型）下，对于 Instance type（实例类型），选择满足您的 Web 服务器需求的实例类型。本教程使用的是 `t2.micro`。
6. 在 Key pair (login)（密钥对（登录））下，对于 Key pair name（密钥对名称），选择您的密钥对。
7. 在 Network settings（网络设置）下，执行以下操作：

   a. 对于 Network（网络）和 Subnet（子网），如果您尚未更改默认 VPC 或子网，则可以保留默认设置。

      如果您对默认 VPC 或子网进行了更改，请检查以下各项：

      i. 实例必须与 RDS 数据库位于同一 VPC 中，才能使用自动连接配置。默认情况下，您只有一个 VPC。

      ii. 您要将实例启动到其中的 VPC 必须连接互联网网关，以便您能从互联网访问您的 Web 服务器。您的默认 VPC 将自动设置互联网网关。

      iii. 为了确保您的实例收到公有 IP 地址，对于 Auto-assign public IP（自动分配公有 IP），请检查并确保选择 Enable（启用）。如果选择 Disable（禁用），请选择 Edit（编辑）（位于 Network Settings（网络设置）右侧），然后对于 Auto-assign public IP（自动分配公有 IP），选择 Enable（启用）。

   b. 要使用 SSH 连接到您的实例，您需要一条安全组规则，授权来自您计算机的公有 IPv4 地址的 SSH (Linux) 或 RDP (Windows) 流量。默认情况下，当您启动实例时，将使用允许来自任何地方的入站 SSH 流量的规则创建新的安全组。

      要确保只有您的 IP 地址才能连接到您的实例，请在防火墙（安全组）下，从允许入站 SSH 流量复选框旁边的下拉列表中，选择我的 IP。

   c. 要允许流量从互联网流向您的实例，请选中以下复选框：

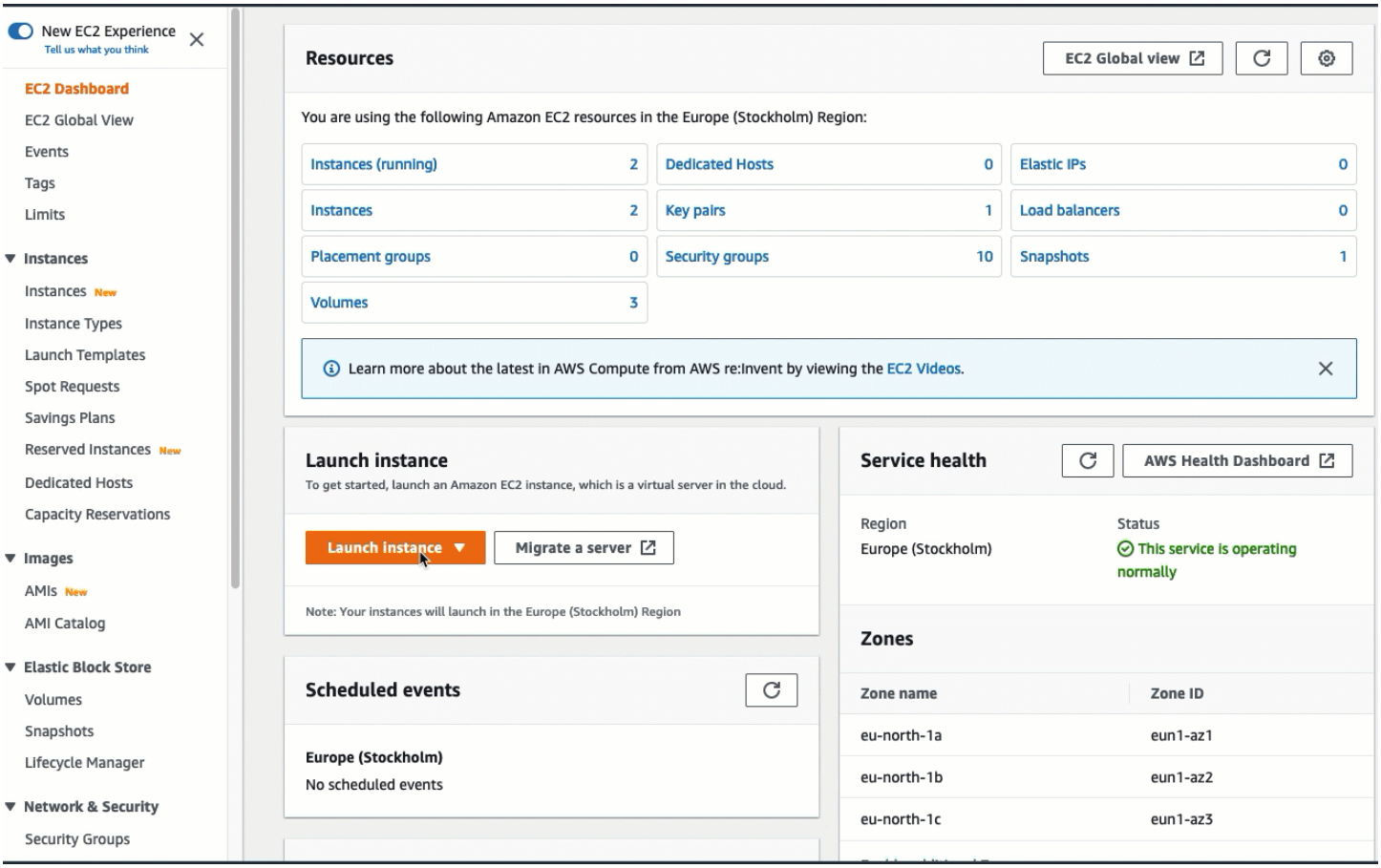
      - Allow HTTPs traffic from the internet（允许来自互联网的 HTTPs 流量）
      - Allow HTTP traffic from the internet（允许来自互联网的 HTTP 流量）

8. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。
9. 选择查看所有实例以关闭确认页面并返回控制台。实例将首先处于 `pending` 状态，然后进入 `running` 状态。

   如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

有关启动实例的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

观看动画：启动 EC2 实例



任务 2：创建 RDS 数据库并将其自动连接到 EC2 实例

此任务的目标是创建 RDS 数据库，然后使用 RDS 控制台中的自动连接功能自动配置 EC2 实例与 RDS 数据库之间的连接。此任务中的步骤将按以下方式配置数据库实例：

- 引擎类型：MySQL
- 模板：免费套餐
- 数据库实例标识符：**`tutorial-database`**
- 数据库实例类：`db.t3.micro`

> **Important**
>
> 在生产环境中，您应该配置实例，以满足您的特定需求。

创建 RDS 数据库并将其自动连接到 EC2 实例

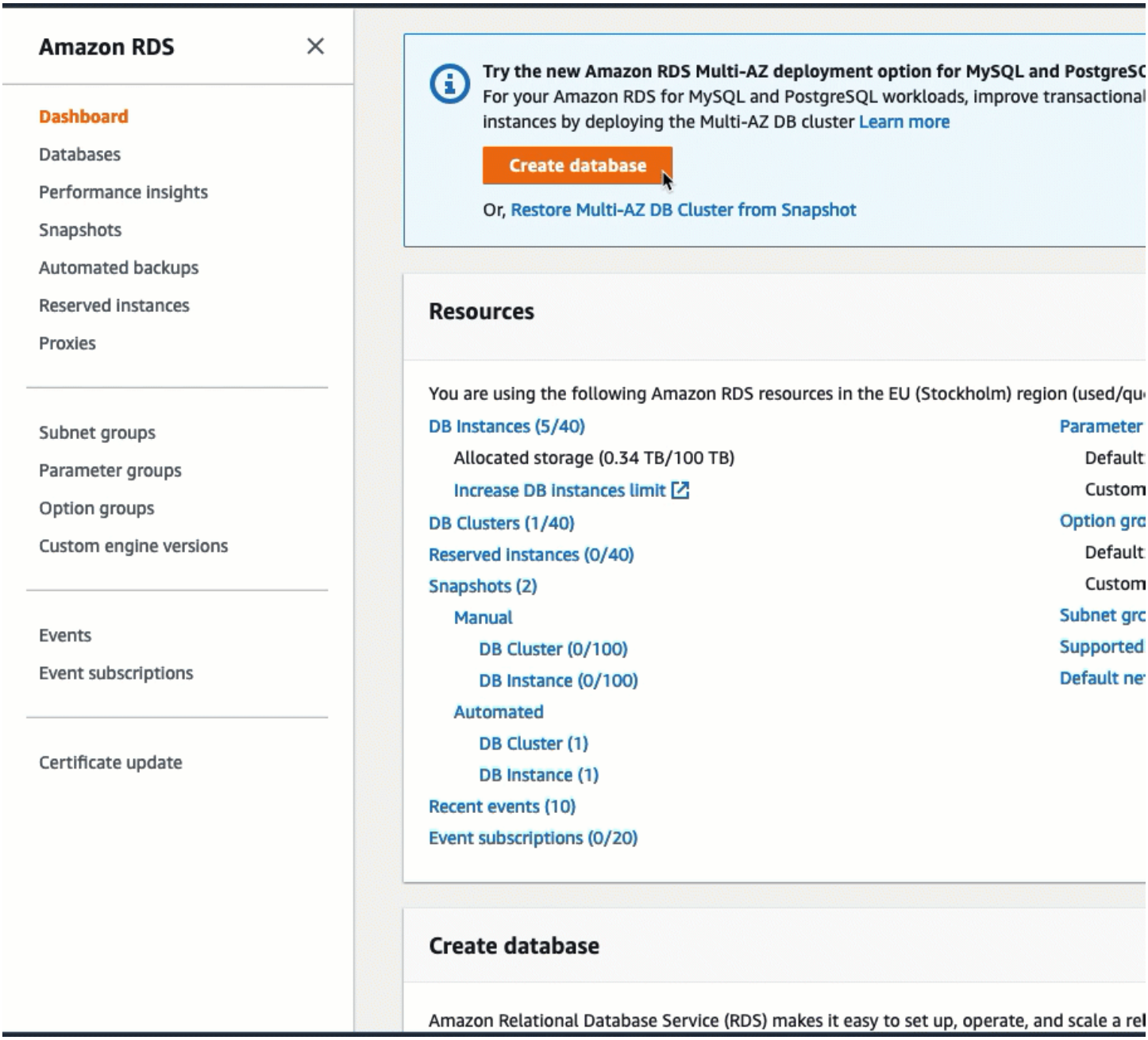
1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 从“Region selector”（区域选择器）（位于右上角）中，选择您在其中创建 EC2 实例的 Amazon Web Services 区域。EC2 实例和 RDS 数据库必须位于同一区域中。
3. 在控制面板中，选择 Create database（创建数据库）。
4. 在 Choose a database creation method（选择数据库创建方法）下，检查并确保选择 Standard create（标准创建）。如果选择 Easy create（轻松创建），则自动连接功能不可用。
5. 在 Engine options（引擎选项）下，对于 Engine type（引擎类型），选择 MySQL。
6. 在 Templates（模板）下，选择一个满足您需求的示例模板。对于本教程，选择 Free tier（免费套餐）以免费创建 RDS 数据库。但请注意，仅当您的账户符合免费套餐资格时，免费套餐才可用。您可以通过选择 Free tier（免费套餐）框中的 Info（信息）链接来阅读更多内容。
7. 在 Settings (设置) 下，执行下列操作：
   a. 对于 DB instance identifier（数据库实例标识符），输入数据库的名称。在本教程中，请输入 **`tutorial-database`**。
   b. 对于 Master username（主用户名），保留默认名称，即 **`admin`**。
   c. 对于 Master password（主密码），输入您在本教程中可以记住的密码，然后对于 Confirm password（确认密码），再次输入该密码。
8. 在实例配置下，对于数据库实例类，保留默认值，即 db.t3.micro。
9. 在 Connectivity（连接）下，对于 Compute resource（计算资源），选择 Connect to an EC2 compute resource（连接到 EC2 计算资源）。这是 RDS 控制台中的自动连接功能。
10. 对于 EC2 instance（EC2 实例），选择您要连接到的 EC2 实例。出于本教程的目的，您可以选择您在上一个任务中创建的实例（您已将其命名为 **`tutorial-instance`**），也可以选择另一个现有实例。如果您未在列表中看到您的实例，请选择 Connectivity（连接）右侧的刷新图标。

    当您使用自动连接功能时，将向此 EC2 实例中添加一个安全组，并将另一个安全组添加到 RDS 数据库中。会将这些安全组自动配置为允许 EC2 实例与 RDS 数据库之间的流量。在下一个任务中，您将验证安全组是否已创建并分配给 EC2 实例和 RDS 数据库。
11. 选择创建数据库。

    在 Databases（数据库）屏幕上，在数据库准备就绪可供使用之前，新数据库的 Status（状态）为 Creating（正在创建）。当状态变为 Available（可用）时，您便可以连接到该数据库。根据数据库类和存储量，新数据库可能需要等待多达 20 分钟时间才可用。

要了解更多信息，请参阅《Amazon RDS 用户指南》中的配置与 EC2 实例的自动网络连接。

观看动画：创建 RDS 数据库并将其自动连接到 EC2 实例



任务 3：验证连接配置

此任务的目标是验证两个安全组是否已创建并分配给实例和数据库。

当您使用控制台中的自动连接功能配置连接时，将自动创建安全组并将其分配给实例和数据库，如下所示：

- 将创建安全组 rds-ec2-*x* 并将其添加到 RDS 数据库中。它包含一条入站规则，引用 ec2-rds-*x* 安全组作为其来源。这将允许来自 EC2 实例（具有 ec2-rds-*x* 安全组）的流量到达 RDS 数据库。
- 将创建安全组 ec2-rds-*x* 并将其添加到 EC2 实例中。它包含一条出站规则，引用 rds-ec2-*x* 安全组作为其目标。这将允许来自 EC2 实例的流量通过 rds-ec2-*x* 安全组到达 RDS 数据库。

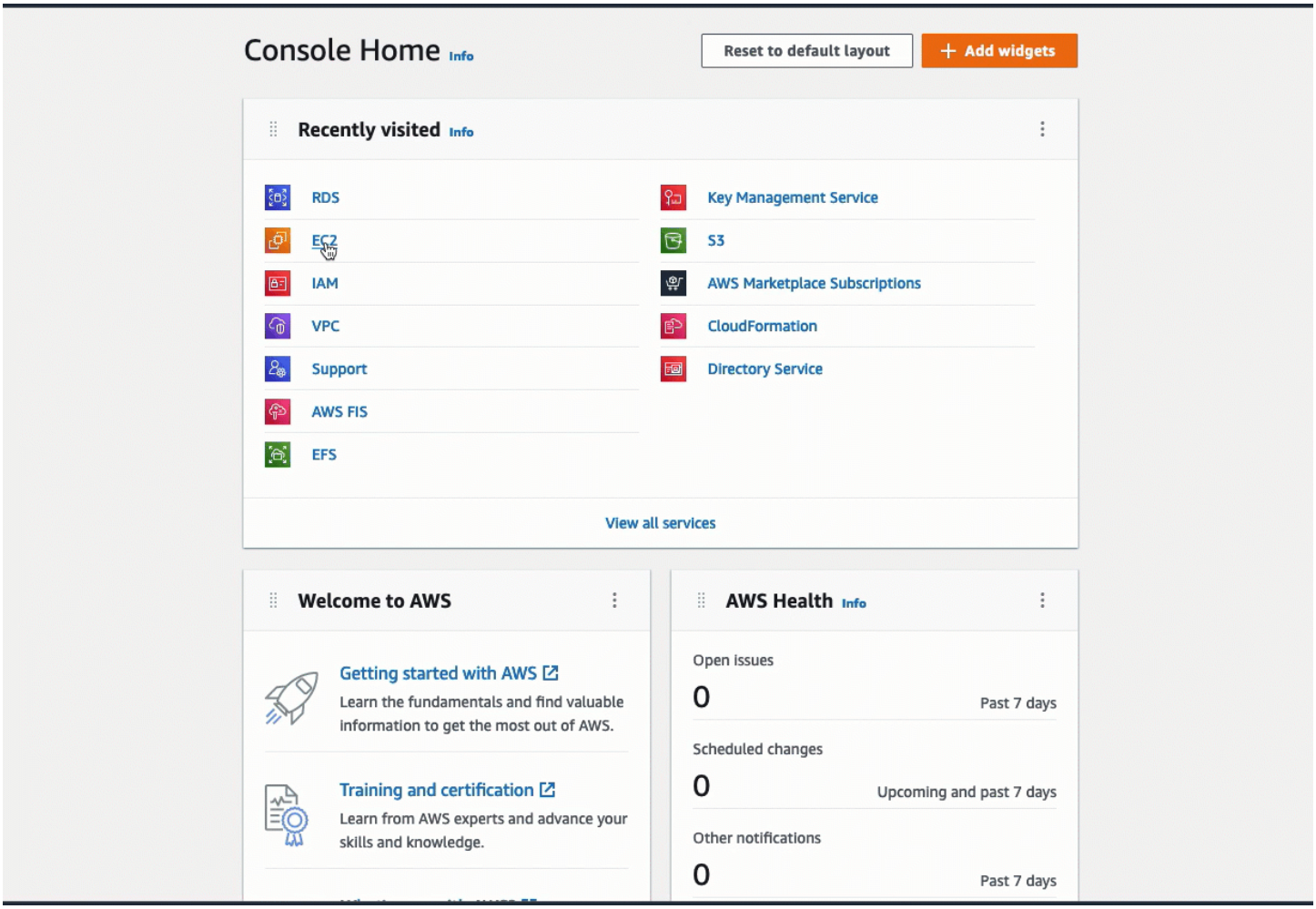
使用控制台验证连接配置

1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 在导航窗格中，选择 Databases（数据库）。
3. 选择您为本教程创建的 RDS 数据库。
4. 在连接和安全性选项卡的安全性、VPC 安全组下，验证并确保显示名为 rds-ec2-*x* 的安全组。
5. 选择 rds-ec2-*x* 安全组。EC2 控制台中的 Security Groups（安全组）屏幕打开。
6. 选择 rds-ec2-*x* 安全组以打开它。
7. 选择入站规则选项卡。
8. 验证并确保以下安全组规则存在，如下所示：
   - 类型：MYSQL/Aurora
   - 端口范围：3306
   - 来源：***sg-0987654321example*** / ec2-rds-*x* - 这是分配给您在前面的步骤中验证的 EC2 实例的安全组。
   - 描述：允许来自 EC2 实例的连接的规则，附有 ***sg-1234567890example***
9. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
10. 在导航窗格中，选择 Instances (实例)。
11. 选择您在上一个任务中选择要连接到 RDS 数据库的 EC2 实例，然后选择 Security（安全性）选项卡。
12. 在安全详细信息的安全组下，验证并确保列表中有名为 ec2-rds-*x* 的安全组。*x* 是一个数字。
13. 选择 ec2-rds-*x* 安全组以打开它。
14. 选择 Outbound rules（出站规则）选项卡。
15. 验证并确保以下安全组规则存在，如下所示：
    - 类型：MYSQL/Aurora
    - 端口范围：3306

- 目标：***sg-1234567890example*** / rds-ec2-***x***
- 描述：允许从此安全组所连接的任何实例到 **database-tutorial** 的连接的规则

通过验证这些安全组和安全组规则是否存在，以及它们是否已按照此过程中所述分配给 RDS 数据库和 EC2 实例，您可以使用自动连接功能验证是否已自动配置连接。

观看动画：验证连接配置



您已完成本教程的“选项 2”。现在，您可以完成“选项 3”，它将介绍如何手动配置在“选项 2”中自动创建的安全组。

任务 4（可选）：清理

> **Warning**
>
> 终止实例是永久且不可逆转的。

终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

现在您已完成该教程，最好清理（删除）所有您不再需要使用的资源。清理 Amazon 资源可以防止您的账户产生任何进一步的费用。

如果您专门为本教程启动了 EC2 实例，则可以将其终止，以停止产生任何与之相关的费用。

使用控制台终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您为本教程创建的实例，然后依次选择 Instance state（实例状态）、Terminate instance（终止实例）。
4. 当系统提示您确认时，选择终止。

如果您专门为本教程创建了 RDS 数据库，则可以将其删除，以停止产生任何与之相关的费用。

使用控制台删除 RDS 数据库

1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 在导航窗格中，选择 Databases (数据库)。
3. 选择您为本教程创建的 RDS 数据库，然后依次选择 Actions（操作）、Delete（删除）。
4. 在方框中输入 **`delete me`**，然后选择 Delete（删除）。

选项 3：通过创建安全组，将实例手动连接到 RDS 数据库

“选项 3”的目标是学习如何通过手动重现自动连接功能的配置，手动配置 EC2 实例与 RDS 数据库之间的连接。

任务

- 开始前的准备工作
- 任务 1：（可选）启动 EC2 实例

- 任务 2：（可选）创建 RDS 数据库
- 任务 3：通过创建安全组并将它们分配给实例，将 EC2 实例手动连接到 RDS 数据库
- 任务 4（可选）：清理

开始前的准备工作

完成本教程需要做好以下准备：

- 与 RDS 数据库位于同一 VPC 中的 EC2 实例。您可以使用现有的 EC2 实例，也可以按照“任务 1”中的步骤创建新实例。
- 与 EC2 实例位于同一 VPC 中的 RDS 数据库。您可以使用现有的 RDS 数据库，也可以按照“任务 2”中的步骤创建新数据库。
- 调用以下操作的权限：
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSecurityGroup`
  - `ec2:CreateSubnet`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

任务 1：（可选）启动 EC2 实例

> **Note**
>
> 启动实例不是本教程的重点。如果您已拥有 Amazon EC2 实例并希望在本教程中使用它，则可跳过此任务。

此任务的目标是启动 EC2 实例，以便您可以完成“任务 3”，在该任务中，您将配置 EC2 实例与 Amazon RDS 数据库之间的连接。此任务中的步骤将按以下方式配置 EC2 实例：

- 实例名称：**`tutorial-instance`**
- AMI：Amazon Linux 2
- 实例类型：`t2.micro`
- 自动分配公有 IP：已启用
- 具有以下三条规则的安全组：
  - 允许来自您的 IP 地址的 SSH
  - 允许来自任何地方的 HTTPS 流量
  - 允许来自任何地方的 HTTP 流量

> **Important**
>
> 在生产环境中，您应该配置实例，以满足您的特定需求。

启动 EC2 实例

1. 登录 Amazon Web Services 管理控制台，打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在 EC2 Dashboard（EC2 控制面板）上，选择 Launch instance（启动实例）。
3. 在 Name and tags（名称与标签）下，对于 Name（名称），输入名称以标识您的实例。对于本教程，请将该实例命名为 **`tutorial-instance-manual-1`**。虽然实例名称不是必填项，但该名称将帮助您轻松识别它。
4. 在 Application and OS Images（应用程序和操作系统映像）下，选择满足您的 Web 服务器需求的 AMI。本教程使用 Amazon Linux。
5. 在 Instance type（实例类型）下，对于 Instance type（实例类型），选择满足您的 Web 服务器需求的实例类型。本教程使用的是 `t2.micro`。
6. 在 Key pair (login)（密钥对（登录））下，对于 Key pair name（密钥对名称），选择您的密钥对。
7. 在 Network settings（网络设置）下，执行以下操作：
   a. 对于 Network（网络）和 Subnet（子网），如果您尚未更改默认 VPC 或子网，则可以保留默认设置。

如果您对默认 VPC 或子网进行了更改，请检查以下各项：

i. 实例必须与 RDS 数据库位于同一 VPC 中。默认情况下，您只有一个 VPC。

ii. 您要将实例启动到其中的 VPC 必须连接互联网网关，以便您能从互联网访问您的 Web 服务器。您的默认 VPC 将自动设置互联网网关。

iii. 为了确保您的实例收到公有 IP 地址，对于 Auto-assign public IP（自动分配公有 IP），请检查并确保选择 Enable（启用）。如果选择 Disable （禁用），请选择 Edit（编辑）（位于 Network Settings（网络设置）右侧），然后对于 Auto-assign public IP（自动分配公有 IP），选择 Enable（启用）。

b. 要使用 SSH 连接到您的实例，您需要一条安全组规则，授权来自您计算机的公有 IPv4 地址的 SSH (Linux) 或 RDP (Windows) 流量。默认情况下，当您启动实例时，将使用允许来自任何地方的入站 SSH 流量的规则创建新的安全组。

要确保只有您的 IP 地址才能连接到您的实例，请在防火墙（安全组）下，从允许入站 SSH 流量复选框旁边的下拉列表中，选择我的 IP。

c. 要允许流量从互联网流向您的实例，请选中以下复选框：

- Allow HTTPs traffic from the internet（允许来自互联网的 HTTPs 流量）
- Allow HTTP traffic from the internet（允许来自互联网的 HTTP 流量）

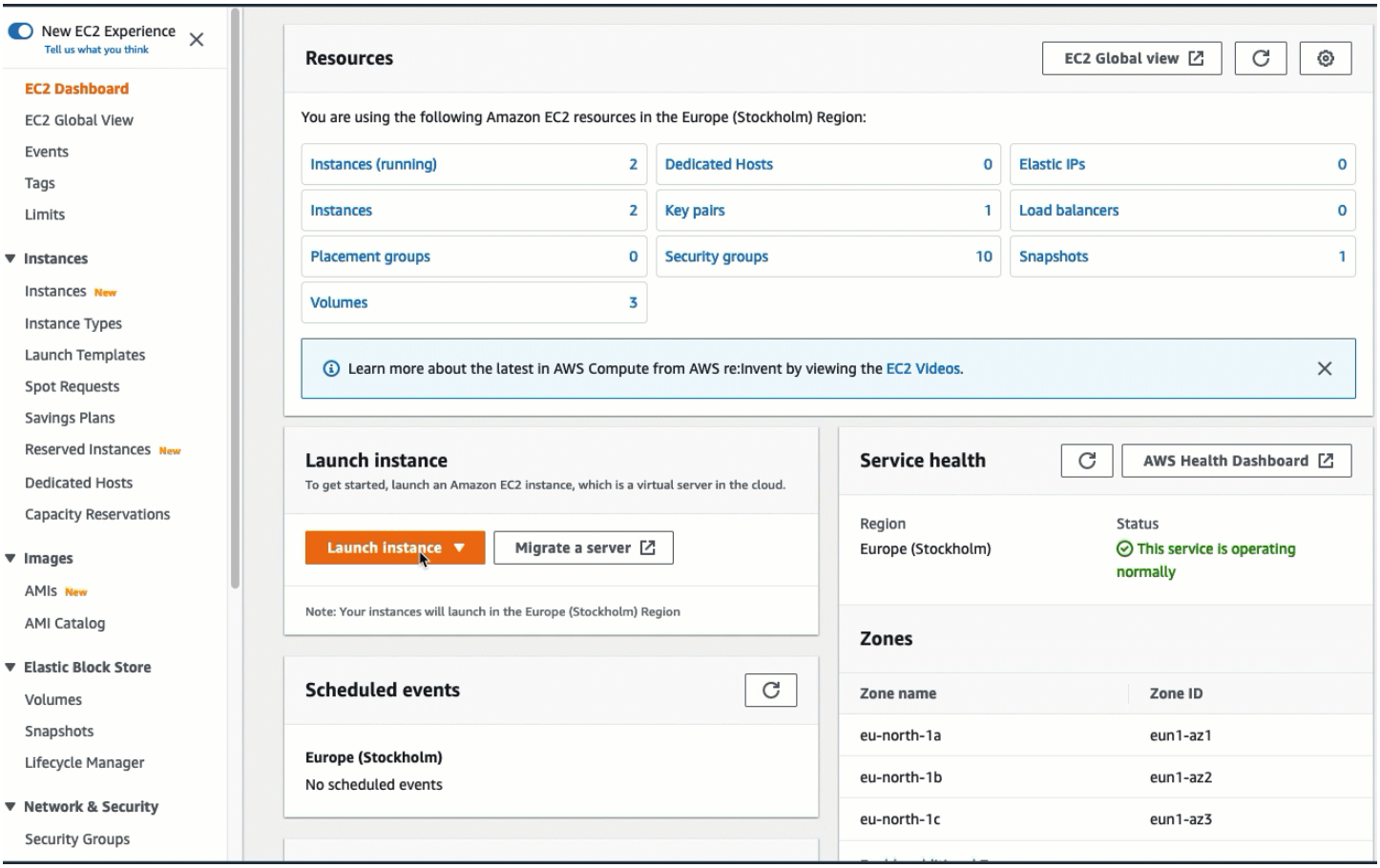
8. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。

9. 选择查看所有实例以关闭确认页面并返回控制台。实例将首先处于 `pending` 状态，然后进入 `running` 状态。

如果实例无法启动或状态立即转至 `terminated` 而非 `running`，请参阅 排查 Amazon EC2 实例启动问题。

有关启动实例的更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

观看动画：启动 EC2 实例



任务 2：（可选）创建 RDS 数据库

> **Note**
>
> 创建 RDS 数据库不是该教程这一部分的重点。如果您已拥有 RDS 数据库并希望将其用于本教程，则可跳过此任务。

此任务的目标是创建 RDS 数据库。在“任务 3”中，当您将此实例连接到 EC2 实例时，您将使用此实例。此任务中的步骤将按以下方式配置 RDS 数据库：

- 引擎类型：MySQL
- 模板：免费套餐
- 数据库实例标识符：**`tutorial-database-manual`**
- 数据库实例类：`db.t3.micro`

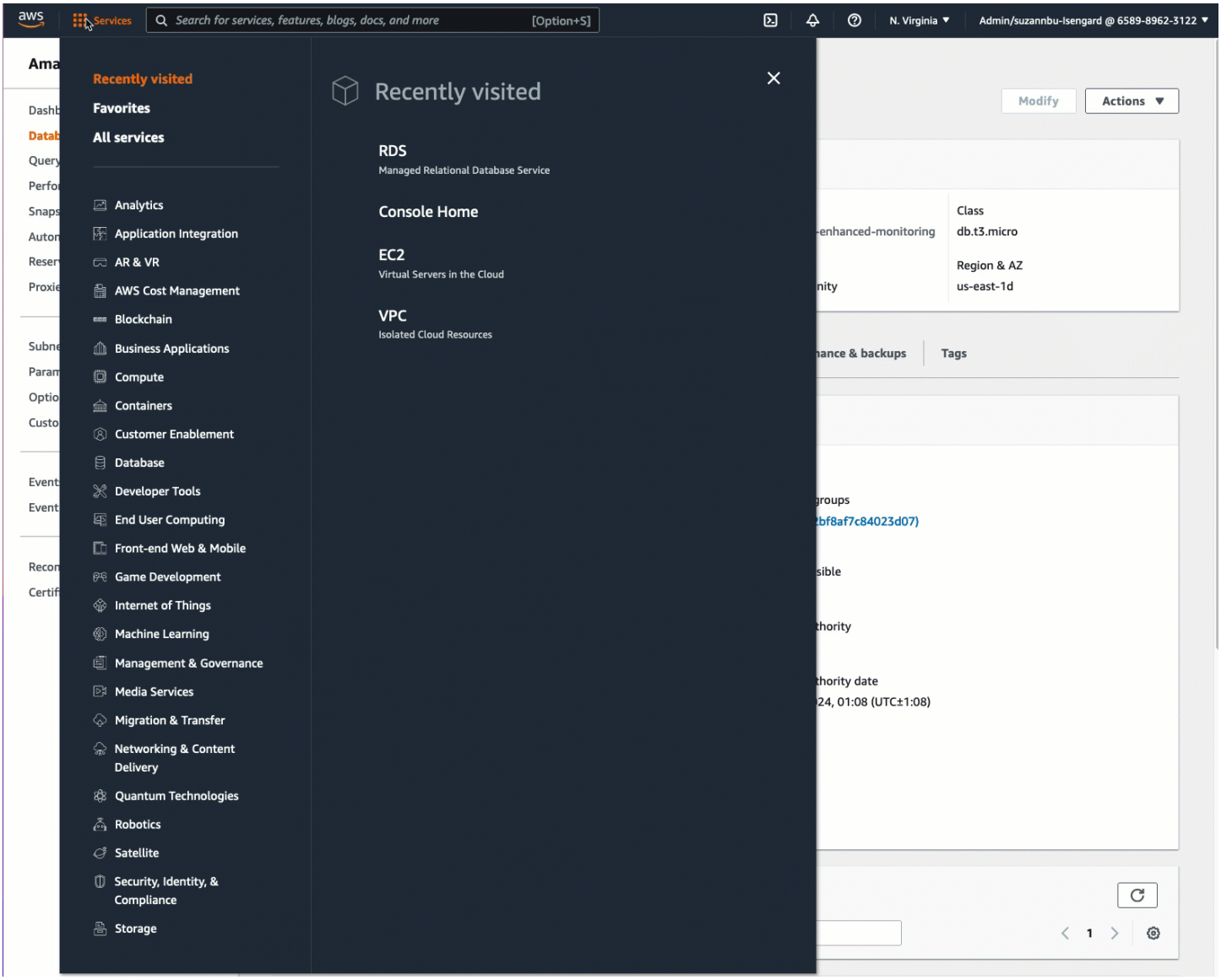
> **Important**
>
> 在生产环境中，您应该配置实例，以满足您的特定需求。

创建 MySQL 数据库实例

1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 从“Region selector”（区域选择器）（位于右上角）中，选择您在其中创建 EC2 实例的 Amazon Web Services 区域。EC2 实例和数据库实例必须位于同一区域中。
3. 在控制面板中，选择 Create database（创建数据库）。
4. 在 Choose a database creation method（选择数据库创建方法）下，选择 Easy create（轻松创建）。当您选择此选项时，用于自动配置连接的自动连接功能将不可用。
5. 在 Engine options（引擎选项）下，对于 Engine type（引擎类型），选择 MySQL。
6. 对于 数据库实例大小，选择 免费套餐。
7. 对于 DB instance identifier（数据库实例标识符），输入 RDS 数据库的名称。在本教程中，请输入 **`tutorial-database-manual`**。
8. 对于 Master username（主用户名），保留默认名称，即 **`admin`**。
9. 对于 Master password（主密码），输入您在本教程中可以记住的密码，然后对于 Confirm password（确认密码），再次输入该密码。
10. 选择创建数据库。

    在 Databases（数据库）屏幕上，在数据库实例准备就绪可供使用之前，新数据库实例的 Status（状态）为 Creating（正在创建）。当状态变为 Available（可用）时，您便可以连接到该数据库实例。根据数据库实例类和存储量，新实例可能需要等待 20 分钟时间才可用。

观看动画：创建数据库实例



任务 3：通过创建安全组并将它们分配给实例，将 EC2 实例手动连接到 RDS 数据库

此任务的目标是通过手动执行以下操作以重现自动连接功能的连接配置：创建两个新安全组，然后分别向 EC2 实例和 RDS 数据库添加一个安全组。

创建两个新安全组，然后分别向 EC2 实例和 RDS 数据库分配一个安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 首先创建要添加到 EC2 实例的安全组，如下所示：

    a. 在导航窗格中，选择 Security Groups（安全组）。

b. 选择 Create security group（创建安全组）。

c. 对于 Security group name（安全组名称），请输入一个描述性的安全组名称。在本教程中，请输入 **ec2-rds-manual-configuration**。

d. 对于 Description（描述），请输入一段简短描述。在本教程中，请输入 **EC2 instance security group to allow EC2 instance to securely connect to RDS database**。

e. 选择创建安全组。在创建 RDS 数据库安全组后，您将返回此安全组以添加出站规则。

3. 现在，创建要添加到 RDS 数据库的安全组，如下所示：

a. 在导航窗格中，选择 Security Groups（安全组）。

b. 选择 Create security group（创建安全组）。

c. 对于 Security group name（安全组名称），请输入一个描述性的安全组名称。在本教程中，请输入 **rds-ec2-manual-configuration**。

d. 对于 Description（描述），请输入一段简短描述。在本教程中，请输入 **RDS database security group to allow EC2 instance to securely connect to RDS database**。

e. 在 Inbound rules（入站规则）下，选择 Add rule（添加规则），然后执行以下操作：

i. 对于 Type（类型），选择 MYSQL/Aurora。

ii. 对于 Source（来源），选择您在此过程的“步骤 2”中创建的 EC2 实例安全组 ec2-rds-manual-configuration。

f. 选择创建安全组。

4. 编辑 EC2 实例安全组以添加出站规则，如下所示：

a. 在导航窗格中，选择安全组。

b. 选择 EC2 实例安全组（您已将其命名为 **ec2-rds-manual-configuration**），然后选择 Outbound rules（出站规则）选项卡。

c. 选择 Edit outbound rules（编辑出站规则）。

d. 选择 Add rule（添加规则），然后执行以下操作：

i. 对于 Type（类型），选择 MYSQL/Aurora。

ii. 对于目标，选择您在此过程的“步骤 3”中创建的 RDS 数据库安全组 rds-ec2-manual-configuration。

iii. 选择保存规则。

5. 将 EC2 实例安全组添加到 EC2 实例中，如下所示：

   a. 在导航窗格中，选择 Instances (实例)。

   b. 选择您的 EC2 实例，然后依次选择 Actions（操作）、Security（安全性）和 Change security groups（更改安全组）。

   c. 在 Associated security groups（关联安全组）下，选择 Select security groups（选择安全组）字段，选择您之前创建的 ec2-rds-manual-configuration，然后选择 Add security group（添加安全组）。

   d. 选择保存。

6. 将 RDS 数据库安全组添加到 RDS 数据库，如下所示：

   a. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。

   b. 在导航窗格中，选择 Databases（数据库），然后选择您的数据库。

   c. 选择 Modify(修改)。

   d. 在 Connectivity（连接）下，对于 Security group（安全组），选择您之前创建的 rds-ec2-manual-configuration，然后选择 Continue（继续）。

   e. 在 Scheduling of modifications（修改计划）下，选择 Apply immediately（立即应用）。

   f. 选择修改数据库实例。

   现在，您已完成模仿在使用自动连接功能时出现的自动步骤的手动步骤。

您已完成本教程的“选项 3”。如果您已完成“选项 1”、“选项 2”和“选项 3”，并且不再需要在本教程中创建的资源，则应删除这些资源，以防止产生不必要的成本。有关更多信息，请参阅 任务 4（可选）：清理。

任务 4（可选）：清理

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

现在您已完成该教程，最好清理（删除）所有您不再需要使用的资源。清理 Amazon 资源可以防止您的账户产生任何进一步的费用。

如果您专门为本教程启动了 EC2 实例，则可以将其终止，以停止产生任何与之相关的费用。

使用控制台终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您为本教程创建的实例，然后依次选择 Instance state（实例状态）、Terminate instance（终止实例）。
4. 当系统提示您确认时，选择终止。

如果您专门为本教程创建了 RDS 数据库，则可以将其删除，以停止产生任何与之相关的费用。

使用控制台删除 RDS 数据库

1. 通过以下网址打开 Amazon RDS 控制台：https://console.aws.amazon.com/rds/。
2. 在导航窗格中，选择 Databases (数据库)。
3. 选择您为本教程创建的 RDS 数据库，然后依次选择 Actions（操作）、Delete（删除）。
4. 在方框中输入 **`delete me`**，然后选择 Delete（删除）。

EC2 实例集和竞价型实例集

EC2 实例集和竞价型实例集非常实用，可在单次操作中启动几十、几百或几千个 Amazon EC2 实例的实例集。实例集中的每个实例由启动模板或您在启动时手动配置的启动参数集配置。

主题

- 功能和优势
- 哪种是实例集最佳使用方法？
- EC2 实例集或竞价型实例集配置选项
- 使用 EC2 实例集
- 使用竞价型实例集
- 监控 EC2 实例集或竞价型实例集
- EC2 实例集的教程
- EC2 实例集 CLI 配置示例
- 竞价型实例集 CLI 配置示例
- EC2 实例集和竞价型实例集的限额

功能和优势

实例集提供以下功能和优势，让您能够在多个 EC2 实例上运行应用程序时，最大限度地节省成本并优化可用性和性能。

多种实例类型

一个实例集可以启动多种实例类型，避免依赖一种实例类型的可用性。这可以提高实例集中实例的总体可用性。

在可用区之间分配实例

一个实例集可以启动到多个可用区，从而使您能够降低成本并提高可用性。如果您的实例集包含竞价型实例，则实例集会根据您对价格和中断的偏好自动选择可用区。

多种购买选项

一个实例集可以启动多种购买选项（竞价型实例和按需型实例），以便让您通过竞价型实例的使用来优化成本。您还可以将预留实例和节省计划折扣与实例集中的按需型实例结合使用，从而享受这些折扣。

自动替换竞价型实例

如果实例集包含竞价型实例，则当竞价型实例中断时，可以自动请求替换竞价型容量。通过容量再平衡功能，实例集还可以监控和主动替换中断风险较高的竞价型实例。

预留按需型容量

实例集可以使用按需容量预留来预留按需型容量。实例集还可以包含适用于机器学习的容量块，让您能在未来某个日期预留 GPU 实例，从而支持短期机器学习（ML）工作负载。

哪种是实例集最佳使用方法？

作为一般最佳实践，我们建议使用 Amazon EC2 Auto Scaling 启动竞价型实例和按需型实例实例集，因为其能够提供可用于管理实例集的其他功能。其他功能列表包括竞价型实例和按需型实例的自动运行状况检查替换、基于应用程序的运行状况检查，以及与弹性负载均衡器的集成，可确保将应用程序流量均匀分配到运行正常的实例。在使用 Amazon ECS、Amazon EKS（自管理式节点组）和 Amazon VPC Lattice 等 Amazon 服务时，也可以使用自动扩缩组。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南。

如果您无法使用 Amazon EC2 Auto Scaling，则可以考虑使用 EC2 实例集或竞价型实例集。EC2 实例集和竞价型实例集提供相同的核心功能。不过，EC2 实例集仅可通过命令行使用，不提供控制台支持。竞价型实例集提供控制台支持，但使用的是旧版 API，且暂无相关投资计划。

使用下表来确定要使用哪种实例集方法。

实例集方法	何时使用？	使用案例
Amazon EC2 Auto Scaling	• 您需要多个采用单一配置或混合配置的实例。 • 您想要自动执行实例的生命周期管理。	创建一个 Auto Scaling 组，以管理实例的生命周期，同时保持所需的实例数量。支持在指定的最小和最大限值之间的水平扩展（添加更多实例）。
EC2 Fleet	• 您需要多个采用单一配置或混合配置的实例。 • 您希望自行管理实例生命周期。	在单次操作中创建一个由按需型实例和竞价型实例组成的 `instant` 实例集，并具有多个因实例类型、AMI、可用区或子网而异的启动规范。竞价型

实例集方法	何时使用？	使用案例
	• 如果不需要自动扩缩，建议使用 instant 类型的 EC2 实例集。	实例分配策略默认按单位计算 lowest-price ，但建议将其改为 price-capacity-optimized 。
竞价型实例集	• 强烈反对使用竞价型实例集，因为其使用的是旧版 API，且暂无相关投资计划。 • 如果想要管理实例生命周期，请使用 EC2 实例集。 • 如果不想管理实例生命周期，请使用自动扩缩组。	仅在使用 EC2 实例集的用例中需要控制台支持时，才使用竞价型实例集。

EC2 实例集或竞价型实例集配置选项

规划 EC2 实例集或竞价型实例集时，建议在决定如何配置实例集时考虑以下选项。

配置选项	问题	文档
实例集请求类型	您想要提交对所需目标容量的一次性请求的实例集，还是随着时间推移维持目标容量的实例集？	EC2 实例集和竞价型实例集请求类型
竞价型实例	您是否计划在实例集中包含竞价型实例？ 查看竞价型实例的最佳实践，使用这些最佳实践规划您的实例集，以便以可能的最低价预置实例。	Amazon EC2 竞价型实例的最佳实践
实例集的支出限额	您是否想限制每小时为实例集支付的费用？	为 EC2 实例集或竞价型实例集设置支出限额
实例类型和基于属性的实例类型选择	您想要在实例集中指定实例类型，还是让 Amazon EC2 选择符合应用程序要求的实例类型？	指定 EC2 实例集或竞价型实例集的实例类型选择属性

配置选项	问题	文档
实例权重	您是否想为每种实例类型分配权重来表示其计算容量和性能，以便 Amazon EC2 可以选择任何可用实例类型的组合来满足所需的目标容量？	使用实例权重来管理 EC2 实例集或竞价型实例集的成本和性能
分配策略	您是否想决定是否针对可用容量、价格或实例类型进行优化，以便用于实例集中的竞价型实例和按需型实例？	使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量
容量再平衡	您是否想要实例集自动替换存在风险的竞价型实例？	在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例
按需容量预留	您是否想为实例集中的按需型实例预留容量？	使用容量预留来预留 EC2 实例集中的按需型容量

EC2 实例集和竞价型实例集请求类型

> **Note**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

EC2 实例集或竞价型实例集的请求类型决定了请求是同步还是异步的，以及是针对所需目标容量发出一次性请求，还是随着时间推移持续努力维持容量。配置实例集时，必须指定请求类型。

EC2 实例集和竞价型实例集都提供两种请求类型：`request` 和 `maintain`。此外，EC2 实例集还提供名为 `instant` 的第三种请求类型。

实例集请求类型

`instant`（仅适用于 EC2 实例集）

如果您将请求类型配置为 `instant`，EC2 队列会针对所需容量发出同步一次性请求。在 API 响应中，它返回已启动的实例并为那些无法启动实例提供错误。有关更多信息，请参阅 配置 instant 类型的 EC2 实例集。

`request`

如果将请求类型配置为 `request`，则实例集针对所需容量发出异步一次性请求。如果因竞价型实例中断导致容量减少，实例集不会尝试补充竞价型实例，也不会在容量不可用时于其他竞价型容量池中提交请求。在使用控制台创建类型为 `request` 的竞价型实例集时，请清除保持目标容量复选框。

`maintain`（默认）

如果将请求类型配置为 `maintain`，实例集会针对所需容量发出异步请求，并自动补充任何中断的竞价型实例来维持容量。在使用控制台创建类型为 `maintain` 的竞价型实例集时，请选中保持目标容量复选框

配置 instant 类型的 EC2 实例集

instant 类型的 EC2 队列是同步一次性请求，该请求只尝试启动所需容量一次。API 响应列出已启动的实例，以及那些无法启动实例的错误。使用 instant 类型的 EC2 队列有几个益处，本文会对其进行介绍。本文末尾提供了配置示例。

对于需要仅启动 API 来启动 EC2 实例的工作负载，您可以使用 RunInstances API。但是，通过 RunInstances，您只能启动按需实例或 Spot 实例，但不能在同一请求中启动两者。此外，当您使用 RunInstances 启动 Spot 实例时，您的 Spot 实例请求仅限于一个实例类型和一个可用区。此操作针对单个 Spot 容量池（一组具有相同实例类型和可用区的未使用实例）。如果 Spot 容量池没有足够的 Spot 实例容量来满足您的请求，则 RunInstances 调用将失败。

相比使用 RunInstances 启动 Spot 实例，我们建议您使用具有设为 `instant` 的 `type` 参数的 CreateFleet API，以获得以下益处：

- 在一个请求中启动按需实例和 Spot 实例。EC2 队列可以启动按需实例和/或 Spot 实例。如果具有可用的容量，并且您的请求的每小时最高价格超过 Spot 价格，则会满足竞价型实例请求。
- 提高 Spot 实例的可用性。通过使用类型为 `instant` 的 EC2 阶段，您可以根据 Spot 最佳实践启动 Spot 实例，获得以下益处：
  - Spot 最佳实践：灵活地选择实例类型和可用区。

    益处：通过指定多个实例类型和可用区，您可以增加 Spot 容量池的数量。这让 Spot 服务有更好的机会查找和分配所需的 Spot 计算容量。一条很好的经验法则是，对于每种工作负载，灵活地在至少 10 种实例类型之间进行选择，并确保所有可用区配置为在 VPC 中使用。
  - 竞价型实例最佳实践：使用 price-capacity-optimized 分配策略。

益处：`price-capacity-optimized` 分配策略可从可用性最佳的竞价型实例容量池中识别实例，然后在这些容量池中自动预置价格最低的实例。由于 Spot 实例容量来自于具有最佳容量的池，因此，这会降低在 Amazon EC2 需要回收容量时，将中断 Spot 实例的可能性。

- 获得更广泛功能集的访问权限。对于需要仅启动 API 的工作负载，并且您希望管理实例的生命周期，而不是让 EC2 队列为您进行管理，请使用类型为 `instant` 的 EC2 队列，而不是 RunInstances API。EC2 队列提供相比 RunInstances 更广泛的功能组，如以下示例所示。对于所有其他工作负载，您应该使用 Amazon EC2 Auto Scaling，因为它为各种工作负载（如 ELB 支持的应用程序、容器化工作负载和队列处理作业）提供了更全面的功能集。

您可以使用类型为 instant 的 EC2 实例集在容量块中启动实例。有关更多信息，请参阅 教程：将 EC2 实例集配置为将实例启动到容量块中。

Amazon EC2 Auto Scaling 和 Amazon EMR 等 Amazon 服务使用类型为 instant 的 EC2 队列启动 EC2 实例。

类型为 instant 的 EC2 队列的先决条件

有关创建 EC2 实例集的先决条件，请参阅 EC2 实例集先决条件。

instant EC2 队列如何工作

使用类型为 `instant` 的 EC2 队列时，事件序列如下所示：

1. 配置：将 CreateFleet 请求类型配置为 `instant`。有关更多信息，请参阅 创建 EC2 实例集。请注意，在进行 API 调用之后，您无法修改它。
2. 请求：当您进行 API 调用时，Amazon EC2 队列会针对所需容量发出同步一次性请求。
3. 响应：API 响应列出已启动的实例，以及那些无法启动实例的错误。
4. 描述：您可以描述 EC2 队列、列出与 EC2 队列关联的实例，以及查看 EC2 队列的历史记录。
5. 终止实例：您可以随时终止实例。
6. 删除实例集请求：可以手动或自动删除实例集请求：
   - 手动：您可以在实例启动后删除实例集请求。

     请注意，不支持包含正在运行实例的已删除 `instant` 实例集。当您删除 `instant` 实例集时，Amazon EC2 会自动终止其所有实例。对于具有超过 1000 个实例的 实例集，删除请求可能会失败。如果您的实例集有超过 1000 个实例，请先手动终止大多数实例，留下 1000 个或更少的实例。然后删除实例集，剩余的实例将自动终止。

- 自动：Amazon EC2 会在以下任一时间后删除实例集请求：
  - 所有实例均已终止。
  - 实例集未能启动任何实例。

示例

以下示例显示如何使用类型为 `instant` 的 EC2 队列，以获取不同使用案例。有关使用 EC2 CreateFleet API 参数的更多信息，请参阅 Amazon EC2 API 引用中的 CreateFleet。

示例

- 示例 1：使用容量优化的分配策略启动 Spot 实例
- 示例 2：使用容量优化分配策略启动单个 Spot 实例
- 示例 3：使用实例权重启动 Spot 实例
- 示例 4：在单个可用区内启动竞价型实例
- 示例 5：在单个可用区内启动单个实例类型的 Spot 实例
- 示例 6：仅当可以启动最小目标容量时才启动 Spot 实例
- 示例 7：仅当可在单个可用区中启动相同实例类型的最低目标容量时，才能启动 Spot 实例
- 示例 8：启动具有多个启动模板的实例
- 示例 9：启动基于按需实例的 Spot 实例
- 示例 10：基于使用容量预留和优先分配策略的按需实例，使用容量优化分配策略启动 Spot 实例
- 示例 11：使用容量优化优先级分配策略启动 Spot 实例
- 示例 12：指定 Systems Manager 参数而非 AMI ID

示例 1：使用容量优化的分配策略启动 Spot 实例

下面的示例指定了类型为 `instant` 的 EC2 队列中所需的参数：启动模板、目标容量、原定设置购买选项，并启动模板覆盖。

- 启动模板由其启动模板名称和版本号标识。
- 12 个启动模板覆盖指定了 4 个不同的实例类型和 3 个不同的子网，每个子网都位于单独的可用区中。每个实例类型和子网组合都定义了 Spot 容量池，从而生成 12 个 Spot 容量池。
- 队列的目标容量为 20 个实例。
- 原定设置购买选项为 `spot`，这会导致实例集尝试在 Spot 容量池中启动 20 个 Spot 实例，并为启动的实例数量提供最佳容量。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab"
            },
```

```
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 2：使用容量优化分配策略启动单个 Spot 实例

您可以通过进行类型为 `instant` 的多个 EC2 队列 API 调用，将 TotalTargetCapacity 设置为 1，以最佳方式一次启动一个 Spot 实例。

下面的示例指定了类型为 instant 的 EC2 队列中所需的参数：启动模板、目标容量、原定设置购买选项和启动模板覆盖。启动模板由其启动模板名称和版本号标识。12 个启动模板覆盖有 4 个不同的实例类型和 3 个不同的子网，每个子网位于单独的可用区域中。实例集的目标容量为 1 个实例，并且原定设置购买选项为 Spot，这导致实例集尝试根据容量优化分配策略，从 12 个 Spot 容量池之一启动 Spot 实例，从最可用的容量池启动 Spot 实例。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
```

```
        "LaunchTemplateSpecification":{
           "LaunchTemplateName":"ec2-fleet-lt1",
           "Version":"$Latest"
        },
        "Overrides":[
           {
              "InstanceType":"c5.large",
              "SubnetId":"subnet-fae8c380"
           },
           {
              "InstanceType":"c5.large",
              "SubnetId":"subnet-e7188bab"
           },
           {
              "InstanceType":"c5.large",
              "SubnetId":"subnet-49e41922"
           },
           {
              "InstanceType":"c5d.large",
              "SubnetId":"subnet-fae8c380"
           },
           {
              "InstanceType":"c5d.large",
              "SubnetId":"subnet-e7188bab"
           },
           {
              "InstanceType":"c5d.large",
              "SubnetId":"subnet-49e41922"
           },
           {
              "InstanceType":"m5.large",
              "SubnetId":"subnet-fae8c380"
           },
           {
              "InstanceType":"m5.large",
              "SubnetId":"subnet-e7188bab"
           },
           {
              "InstanceType":"m5.large",
              "SubnetId":"subnet-49e41922"
           },
           {
              "InstanceType":"m5d.large",
              "SubnetId":"subnet-fae8c380"
```

```
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
   ],
   "TargetCapacitySpecification": {
       "TotalTargetCapacity": 1,
       "DefaultTargetCapacityType": "spot"
   },
   "Type": "instant"
}
```

示例 3：使用实例权重启动 Spot 实例

以下示例使用实例权重，这意味着价格是每单位小时价格，而不是每实例小时价格。假定一个工作负载单位需要 15 GB 内存和 4 个 vCPU，每个启动配置列出不同的实例类型和不同的权重，具体取决于实例上可以运行的工作负载单位数。例如，m5.xlarge（4 个 vCPU 和 16 GB 内存）可以运行 1 个单位且权重为 1，m5.2xlarge（8 个 vCPUs 和 32 GB 内存）可以运行 2 个单位，且权重为 2，依此类推。总目标容量设置为 40 个单位。原定设置购买选项为 Spot，并且分配策略为容量优化，这会导致 40 个 m5.xlarge（40 除以 1），20 个 m5.2xlarge（40 除以 2），10 个 m5.4xlarge（40 除以 4），5 个 m5.8xlarge（40 除以 8），或基于容量优化分配策略，权重相加达到所需容量的实例类型的混合。

有关更多信息，请参阅 使用实例权重来管理 EC2 实例集或竞价型实例集的成本和性能。

```
{
   "SpotOptions":{
      "AllocationStrategy":"capacity-optimized"
   },
   "LaunchTemplateConfigs":[
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
```

```
            "InstanceType":"m5.xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":1
         },
         {
            "InstanceType":"m5.xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":1
         },
         {
            "InstanceType":"m5.xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":1
         },
         {
            "InstanceType":"m5.2xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":2
         },
         {
            "InstanceType":"m5.2xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":2
         },
         {
            "InstanceType":"m5.2xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":2
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":4
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":4
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":4
         },
```

```
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":8
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":8
            },
            {
               "InstanceType":"m5.8xlarge",
               "SubnetId":"subnet-49e41922",
               "WeightedCapacity":8
            }
         ]
      }
   ],
   "TargetCapacitySpecification":{
      "TotalTargetCapacity":40,
      "DefaultTargetCapacityType":"spot"
   },
   "Type":"instant"
}
```

示例 4：在单个可用区内启动竞价型实例

您可以通过将 Spot 选项 SingleAvailabilityZone 设置为 true（真），将实例集配置为启动单个可用区中的所有实例。

12 个启动模板覆盖具有不同的实例类型和子网（每个都位于单独的可用区），但权重容量相同。总目标容量为 20 个实例，原定设置购买选项为 Spot，并且 Spot 分配策略为容量优化。EC2 队列使用启动规范，从具有最佳容量的 Spot 容量池启动全部在单个可用区中的 20 个 Spot 实例。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
```

```
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
```

```
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 5：在单个可用区内启动单个实例类型的 Spot 实例

您可以通过将 SpotOptions SingleInstanceType 和 SingleAvailabilityZone 设置为 true（真），将队列配置为启动单个可用区中相同实例类型的所有实例。

12 个启动模板覆盖具有不同的实例类型和子网（每个都位于单独的可用区），但权重容量相同。总目标容量为 20 个实例，原定设置购买选项为 Spot，Spot 分配策略为容量优化。EC2 队列使用启动规范，从具有最佳容量的 Spot 容量池启动全部在单个可用区中相同实例类型的 20 个 Spot 实例。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
```

```
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5.4xlarge",
               "SubnetId":"subnet-49e41922"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"m5d.4xlarge",
               "SubnetId":"subnet-49e41922"
            }
```

```
        ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 6：仅当可以启动最小目标容量时才启动 Spot 实例

只有通过将 Spot 选项 MinTargetCapacity 设置为要同时启动的最小目标容量，以启动最小目标容量，您才可以将队列配置为启动实例。

指定 MinTargetCapacity 时，必须至少指定下列参数中的一个：SingleInstanceType 或 SingleAvailabilityZone。此示例中指定的是 SingleInstanceType，因此所有这 20 个实例必须使用相同的实例类型。

12 个启动模板覆盖具有不同的实例类型和子网（每个都位于单独的可用区），但权重容量相同。总目标容量和最小目标容量均设置为 20 个实例，默认购买选项为竞价型，竞价型分配策略为容量优化。只有在能够同时启动所有 20 个实例时，EC2 队列才能使用启动模板覆盖，从具有最佳容量的 Spot 容量池启动 20 个 Spot 实例。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "MinTargetCapacity": 20
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
```

```
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-49e41922"
         }
      ]
   }
```

```
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 7：仅当可在单个可用区中启动相同实例类型的最低目标容量时，才能启动 Spot 实例

只有通过将 Spot 选项 MinTargetCapacity，以及 SingleInstanceType 和 SingleAvailabilityZone 选项设置为要同时启动的最小目标容量，以在单个可用区中，以单个实例类型启动最小目标容量，您才可以将队列配置为启动实例。

覆盖启动模板的 12 个启动规范具有不同的实例类型和子网（每个都位于单独的可用区），但权重容量相同。总目标容量和最小目标容量均为 20 个实例，原定设置购买选项为 Spot，Spot 分配策略为容量优化，SingleInstanceType 和 SingleAvailabilityZone 均为 true（真）。只有在能够同时启动所有 20 个实例时，EC2 队列才能使用启动规范，从具有最佳容量的 Spot 容量池，启动全部在单个可用区中相同实例类型的 20 个 Spot 实例。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 20
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.4xlarge",
               "SubnetId":"subnet-e7188bab"
            },
            {
```

```
            "InstanceType":"c5.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5d.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5.4xlarge",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5d.4xlarge",
            "SubnetId":"subnet-49e41922"
         }
      ]
   }
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 20,
```

```
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 8：启动具有多个启动模板的实例

您可以通过指定多个启动模板来配置队列，以便为不同的实例类型或一组实例类型启动具有不同启动规范的实例。在此示例中，我们希望为不同实例类型提供不同的 EBS 卷大小，并且我们已经在启动模板 ec2-fleet-lt-4xl、ec2-fleet-lt-9xl 和 ec2-fleet-lt-18xl 中进行配置。

在此示例中，我们根据大小为 3 种实例类型使用了 3 种不同启动模板。所有启动模板上的启动规范覆盖使用基于实例类型上 vCPU 的实例权重。总目标容量为 144 个单位，原定设置购买选项为 Spot，并且 Spot 分配策略为容量优化。EC2 队列可以使用启动模板 ec2-fleet-4xl 启动 9 个 c5n.4xlarge（144 除以 16），或者使用启动模板 ec2-fleet-9xl 启动 4 个 c5n.9xlarge（144 除以 36），或者使用启动模板 ec2-fleet-18xl 启动 2 个 c5n.18xlarge（144 除以 72），或基于容量优化分配策略，权重相加达到所需容量的实例类型的混合。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt-18xl",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-fae8c380",
               "WeightedCapacity":72
            },
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-e7188bab",
               "WeightedCapacity":72
            },
            {
               "InstanceType":"c5n.18xlarge",
               "SubnetId":"subnet-49e41922",
```

```
            "WeightedCapacity":72
         }
      ]
   },
   {
      "LaunchTemplateSpecification":{
         "LaunchTemplateName":"ec2-fleet-lt-9xl",
         "Version":"$Latest"
      },
      "Overrides":[
         {
            "InstanceType":"c5n.9xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":36
         },
         {
            "InstanceType":"c5n.9xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":36
         },
         {
            "InstanceType":"c5n.9xlarge",
            "SubnetId":"subnet-49e41922",
            "WeightedCapacity":36
         }
      ]
   },
   {
      "LaunchTemplateSpecification":{
         "LaunchTemplateName":"ec2-fleet-lt-4xl",
         "Version":"$Latest"
      },
      "Overrides":[
         {
            "InstanceType":"c5n.4xlarge",
            "SubnetId":"subnet-fae8c380",
            "WeightedCapacity":16
         },
         {
            "InstanceType":"c5n.4xlarge",
            "SubnetId":"subnet-e7188bab",
            "WeightedCapacity":16
         },
         {
```

```
                "InstanceType":"c5n.4xlarge",
                "SubnetId":"subnet-49e41922",
                "WeightedCapacity":16
             }
          ]
       }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 144,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 9：启动基于按需实例的 Spot 实例

下面的示例为队列指定 20 个实例的总目标容量和 5 个按需实例的目标容量。原定设置购买选项为 Spot。队列按照指定的方式启动 5 个按需实例，但需要再启动 15 个实例以满足总目标容量要求。差值购买选项的计算公式为 TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType，这使得启动 15 个 Spot 实例的队列根据容量优化分配策略构成 12 个 Spot 容量池之一。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380"
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab"
            },
            {
               "InstanceType":"c5.large",
```

```
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"c5d.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5.large",
            "SubnetId":"subnet-49e41922"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-fae8c380"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-e7188bab"
         },
         {
            "InstanceType":"m5d.large",
            "SubnetId":"subnet-49e41922"
         }
      ]
   }
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 20,
    "OnDemandTargetCapacity": 5,
```

```
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 10：基于使用容量预留和优先分配策略的按需实例，使用容量优化分配策略启动 Spot 实例

您可以通过将容量预留的使用策略配置为 use-capacity-reservations-first（首先使用容量预留），来将队列配置为在以原定设置目标容量类型作为 Spot 启动按需实例的基础时，首先使用按需容量预留。此外，如果多个实例池具有未使用的 容量预留，则应用选定的按需分配策略。在此示例中，按需分配策略为优先级。

在该示例中，有 6 个可用的未使用容量预留。此数目少于队列的 10 个按需实例的目标按需容量。

账户在 2 个池中有以下 6 个未使用的容量预留。每个池中的容量预留数由 AvailableInstanceCount 指示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

以下队列配置仅显示该示例的相关配置。按需分配策略为优先级，容量预留的使用策略为 use-capacity-reservations-first（首先使用容量预留）。Spot 分配策略为容量优化。总目标容量为 20，按需目标容量为 10，原定设置目标容量类型为 Spot。

```
{
```

```
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "OnDemandOptions":{
       "CapacityReservationOptions": {
         "UsageStrategy": "use-capacity-reservations-first"
       },
       "AllocationStrategy":"prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 3.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 4.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 5.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922",
```

```
                "Priority": 6.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 7.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 8.0
            },
            {
                "InstanceType":"m5.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 9.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-fae8c380",
                "Priority": 10.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-e7188bab",
                "Priority": 11.0
            },
            {
                "InstanceType":"m5d.large",
                "SubnetId":"subnet-49e41922",
                "Priority": 12.0
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity": 10,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

在使用上述配置创建 instant 队列后，将启动下面的 20 个实例来满足目标容量：

- us-east-1a 中的 7 个 c5.large 按需实例 – us-east-1a 中的 c5.large 的优先级第一，并且有 3 个可用的未使用 c5.large 容量预留。首先使用容量预留启动 3 个按需实例，并根据按需分配策略（此示例中为优先级）启动另 4 个按需实例。
- us-east-1a 中的 3 个 m5.large 按需型实例：us-east-1a 中的 m5.large 实例优先级第二，并且有 3 个未使用的 m5.large 容量预留可用。
- 来自 12 个 Spot 容量池之一的 10 个 Spot 实例，该容量池根据容量优化的分配策略具有最佳容量。

在启动队列后，您可以运行 describe-capacity-reservations 来查看保留的未使用的 容量预留 的数目。在此示例中，您该看到以下响应，该响应指示已使用所有 c5.large 和 m5.large 容量预留。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 0
}
```

示例 11：使用容量优化优先级分配策略启动 Spot 实例

下面的示例指定了类型为 instant 的 EC2 队列中所需的参数：启动模板、目标容量、原定设置购买选项和启动模板覆盖。启动模板由其启动模板名称和版本号标识。覆盖启动模板的 12 个启动规范具有分配了优先级的 4 个不同的实例类型，以及 3 个不同的子网（每个都位于单独的可用区）。队列的目标容量是 20 个实例，原定设置购买选项为 Spot，这导致队列尝试根据容量优化优先级的分配策略，从 12 个 Spot 容量池之一启动 20 个 Spot 实例，该策略会尽最大努力实施优先级，但首先会针对容量进行优化。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized-prioritized"
    },
    "LaunchTemplateConfigs": [
      {
         "LaunchTemplateSpecification":{
```

```
            "LaunchTemplateName":"ec2-fleet-lt1",
            "Version":"$Latest"
         },
         "Overrides":[
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 1.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 2.0
            },
            {
               "InstanceType":"c5d.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 2.0
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 3.0
            },
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 3.0
            },
```

```
            {
               "InstanceType":"m5.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 3.0
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-fae8c380",
               "Priority": 4.0
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-e7188bab",
               "Priority": 4.0
            },
            {
               "InstanceType":"m5d.large",
               "SubnetId":"subnet-49e41922",
               "Priority": 4.0
            }
         ]
      }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 12：指定 Systems Manager 参数而非 AMI ID

以下示例使用启动模板来指定实例集中实例的配置。在本示例中，不是为 `ImageId` 指定 AMI ID，而是使用 System Manager 参数来引用 AMI。在实例启动时，Systems Manager 参数会解析为 AMI ID。

在本示例中，Systems Manager 参数以有效格式指定：`resolve:ssm:golden-ami`。Systems Manager 参数也有其他有效格式。有关更多信息，请参阅 使用 Systems Manager 参数而非 AMI ID。

> **Note**
>
> 队列类型必须为类型 `instant`。其他实例集类型不支持指定 Systems Manager 参数来代替 AMI ID。

```
{
    "LaunchTemplateData": {
        "ImageId": "resolve:ssm:golden-ami",
        "InstanceType": "m5.4xlarge",
        "TagSpecifications": [{
            "ResourceType": "instance",
            "Tags": [{
                "Key": "Name",
                "Value": "webserver"
            }]
        }]
    }
}
```

为 EC2 实例集或竞价型实例集设置支出限额

您可以设置每小时愿意为 EC2 实例集或竞价型实例集支出的限额。在达到支出限额时，即使尚未达到目标容量，实例集也会停止启动实例。

按需型实例和竞价型实例有单独的支出限额。

为 EC2 实例集中的按需型实例和竞价型实例配置支出限额

使用 create-fleet 命令和以下参数：

- 若为按需型实例：在 `OnDemandOptions` 结构中，在 `MaxTotalPrice` 字段中指定支出限额。
- 若为竞价型实例：在 `SpotOptions` 结构中，在 `MaxTotalPrice` 字段中指定支出限额。

为竞价型实例集中的按需型实例和竞价型实例配置支出限额

Amazon EC2 控制台和 Amazon CLI 都可用于配置支出限额。

（控制台）创建竞价型实例集时，请选中设置 Spot 实例的最大成本复选框，然后在设置您的最大成本(每小时)中输入一个值。有关更多信息，请参阅使用已定义的参数创建竞价型实例集请求中的步骤 6.e.。

（Amazon CLI）使用 request-spot-fleet 命令和以下参数：

- 若为按需型实例：在 `OnDemandMaxTotalPrice` 字段中指定支出限额。
- 若为竞价型实例：在 `SpotMaxTotalPrice` 字段中指定支出限额。

示例

以下示例显示了两个不同的方案。在第一个示例中，实例集在满足按需型实例的目标容量 (OnDemandTargetCapacity) 后停止启动按需型实例。在第二个示例中，实例集在达到每小时愿意为按需型实例支付的最高金额 (MaxTotalPrice) 时停止启动实例。

示例：在达到目标容量时停止启动按需型实例

假设发出 m4.large 按需实例请求，其中：

- 按需价格：每小时 0.10 美元
- OnDemandTargetCapacity: 10
- MaxTotalPrice：1.50 美元

实例集启动 10 个按需型实例，因为总费用 1.00 美元（10 个实例 x 0.10 美元）未超过实例的 MaxTotalPrice 1.50 美元。

示例：在达到最高总费用时停止启动按需型实例

假设发出 m4.large 按需实例请求，其中：

- 按需价格：每小时 0.10 美元
- OnDemandTargetCapacity: 10
- MaxTotalPrice：0.80 美元

如果实例集启动按需型目标容量（10 个按需型实例），则每小时的总成本为 1.00 美元。这超过了为按需实例的 MaxTotalPrice 指定的金额（0.80 美元）。为了防止支出超过您愿意支付的金额，实例集仅启动 8 个按需型实例（低于按需型目标容量），因为启动更多实例将超过按需型实例的 MaxTotalPrice。

具爆发能力的实例

如果您使用可突增性能的实例类型启动您的 Spot 实例，并打算立即并短时间使用可突增性能的 Spot 实例，且没有空闲时间来积累 CPU 积分，建议您以标准模式启动实例，以避免支付更高的成本。如果您以无限模式启动可突增性能的 Spot 实例并立即突增 CPU，您将会为突增花费超额积分。如果使用实例的时间很短，使得实例没有时间积累 CPU 积分来支付超额积分，则您将在终止实例时为超额积分付费。

仅当实例的运行时间较长，足以积累进行突增的 CPU 积分时，针对可突增性能的竞价型实例的无限模式才适用。否则，支付剩余积分会使可突增性能的竞价型实例比使用其他实例更加昂贵。有关更多信息，请参阅何时使用无限模式与固定 CPU。

通过提供足够的计算资源来配置实例，启动积分旨在为 T2 实例提供有成效的初始启动体验。不允许重复启动 T2 实例以访问新的启动积分。如果您需要持续的 CPU，您可以赚取积分（通过空转一段时间），将无限模式用于 T2 Spot 实例，或将实例类型和专用 CPU 一起使用。

指定 EC2 实例集或竞价型实例集的实例类型选择属性

创建 EC2 实例集或竞价型实例集时，必须指定一种或多种实例类型，以便配置实例集中的按需型实例和竞价型实例。除了手动指定实例类型外，还可以指定实例必须具有的属性，Amazon EC2 将使用这些属性标识所有实例类型。这称为基于属性的实例类型选择。例如，您可以指定实例所需的最小和最大 vCPU 数量，实例集将使用满足这些 vCPU 要求的任何可用实例类型启动实例。

基于属性的实例类型选择是工作负载和框架的理想选择，这些工作负载和框架可以灵活地确定所用的实例类型，例如在运行容器或 Web 机群、处理大数据以及实施持续集成和部署 (CI/CD) 工具等情况下。

优点

基于属性的实例类型选择具有以下优势：

- 轻松使用正确的实例类型 – 有如此多的实例类型可供使用，因此找到适用于您的工作负载的实例类型可能非常耗时。当您指定实例属性时，实例类型将自动具有工作负载所需的属性。
- 简化配置 – 要为实例集手动指定多种实例类型，您必须为每种实例类型创建单独的启动模板覆盖。但是，借助基于属性的实例类型选择，要提供多种实例类型，您只需在启动模板或启动模板覆盖中指定实例属性即可。
- 自动使用新实例类型 – 当您指定实例属性而不是实例类型时，您的实例集可以在新一代实例类型发布时使用这些实例类型，“满足未来”实例集配置的需求。
- 实例类型灵活性 – 指定实例属性而不是实例类型时，实例集可以从各种实例类型中进行选择来启动竞价型实例，从而遵循灵活选择竞价型实例类型的最佳实践。

主题

- 基于属性的实例类型选择的工作原理
- 价格保护
- 性能保护
- 注意事项

- 通过基于属性的实例类型选择创建 EC2 机群
- 通过基于属性的实例类型选择创建竞价型实例集
- 有效 EC2 实例集配置示例和无效 EC2 实例集配置示例
- 有效竞价型实例集配置示例和无效竞价型实例集配置示例
- 预览具有指定属性的实例类型

基于属性的实例类型选择的工作原理

要在机群配置中使用基于属性的实例类型选择，请将实例类型列表替换为实例所需的实例属性列表。EC2 实例集或竞价型实例集将在具有指定实例属性的任何可用实例类型上启动实例。

主题

- 实例属性的类型
- 在何处配置基于属性的实例类型选择
- EC2 实例集或竞价型实例集如何在预置实例集时使用基于属性的实例类型选择

实例属性的类型

您可以指定多个实例属性来表达计算需求，例如：

- vCPU 数量 - 每个实例的最小和最大 vCPU 数量。
- 内存 - 每个实例的最小和最大内存 GiB 数量。
- 本地存储 - 是使用 EBS 还是实例存储卷作为本地存储。
- 可突增性能 - 是否使用 T 实例系列（包括 T4g、T3a、T3 和 T2 类型）。

有关每个属性和默认值的描述，请参阅 Amazon EC2 API 参考中的 InstanceRequirements。

在何处配置基于属性的实例类型选择

根据您使用的是控制台还是 Amazon CLI，可以按如下方式为基于属性的实例类型选择指定实例属性：

在控制台中，您可以在以下实例集配置组件中指定实例属性：

- 在启动模板中，并在实例集请求中引用启动模板
- （仅适用于竞价型实例集）在实例集请求中

在 Amazon CLI 中，您可以在以下一个或全部机群配置组件中指定实例属性：

- 在启动模板中，并在实例集请求中引用启动模板
- 在启动模板覆盖中

  如果您想混合应用使用不同 AMI 的实例，则可以在多个启动模板覆盖中指定实例属性。例如，不同的实例类型可以使用 x86 和基于 ARM 的处理器。
- （仅适用于竞价型实例集）在启动规范中

EC2 实例集或竞价型实例集如何在预置实例集时使用基于属性的实例类型选择

EC2 实例集或竞价型实例集通过以下方式预置实例集：

- 标识具有指定属性的实例类型。
- 使用价格保护来确定要排除的实例类型。
- 根据具有匹配实例类型的 Amazon 区域或可用区确定将从中启动实例的容量池。
- 应用指定的分配策略来确定从中启动实例的容量池。

  请注意，基于属性的实例类型选择不会选择从中预置实例集的容量池；这是分配策略的任务。

  如果您指定分配策略，实例集将根据指定的分配策略启动实例。
  - 若为竞价型实例，基于属性的实例类型选择支持价格容量优化、容量优化和最低价格分配策略。请注意，不推荐使用最低价格竞价型分配策略，因为这对竞价型实例造成的中断风险最高。
  - 若为按需型实例，基于属性的实例类型选择支持最低价格分配策略。
- 如果没有适用于具有指定实例属性的实例类型的容量，则无法启动任何实例，并且机群返回错误。

价格保护

价格保护是一项功能，可以防止 EC2 实例集或竞价型实例集使用您认为成本过高的实例类型，即使它们恰好适合您指定的属性。要使用价格保护，您需要设置价格阈值。然后，当 Amazon EC2 选择具有您的属性的实例类型时，会排除定价高于阈值的实例类型。

Amazon EC2 计算价格阈值的方式如下：

- Amazon EC2 首先从与您的属性匹配的实例类型中识别出价格最低的实例类型。
- 然后，Amazon EC2 会使用您为价格保护参数指定的值（以百分比表示）乘以已识别的实例类型的价格。其结果就是用作价格阈值的价格。

按需型实例和竞价型实例有不同的价格阈值。

当您使用基于属性的实例类型选择创建实例集时，默认情况下会启用价格保护。您可以保留默认值，也可以指定自己的值。

您也可以关闭价格保护。要表明无价格保护阈值，请指定一个较高的百分比值，如 999999。

主题

- 价格最低的实例类型的识别方式
- 按需型实例价格保护
- 竞价型实例价格保护
- 指定价格保护阈值

价格最低的实例类型的识别方式

Amazon EC2 会从与您指定的属性相匹配的实例类型中识别价格最低的实例类型，从而确定价格阈值所依据的价格。它的做法具体如下：

- 它首先会从当前一代 C、M 或 R 实例类型中查找与您的属性相匹配的实例类型。如果找到匹配结果，它就会识别出价格最低的实例类型。
- 如果没有匹配结果，它会从当前一代的任何实例类型中查找与您的属性相匹配的实例类型。如果找到匹配结果，它就会识别出价格最低的实例类型。
- 如果没有匹配结果，它会从上一代的任何实例类型中查找与您的属性相匹配的实例类型，并识别出价格最低的实例类型。

按需型实例价格保护

按需型实例类型的价格保护阈值基于已识别的最低价格按需型实例类型 (`OnDemandMaxPricePercentageOverLowestPrice`)，按照高于其价格的百分比计算。您可以指定该高出的百分比，表示您原意支付的价格。如果您未指定此参数，则使用默认值 20 计算价格保护阈值(即比已识别的价格高出 20%)。

例如，如果识别的按需型实例价格为 `0.4271`，并且您指定了 25，则价格阈值将比 `0.4271` 高 25%。按如下方式计算：`0.4271 * 1.25 = 0.533875`。计算出的价格是您愿意为按需型实例支付的最高价格，在本示例中，Amazon EC2 将排除任何价格高于 `0.533875` 的按需型实例类型。

竞价型实例价格保护

默认情况下，Amazon EC2 将自动应用最佳的竞价型实例价格保护，从而始终如一地从各种实例类型中进行选择。您也可以自行手动设置价格保护。但是，让 Amazon EC2 为您代劳可以提高竞价型容量得到满足的可能性。

您可以使用以下选项之一，手动指定价格保护。如果您手动设置价格保护，则建议您使用第一个选项。

- 已识别的最低价格按需型实例类型的百分比 [MaxSpotPriceAsPercentageOfOptimalOnDemandPrice]

  例如，如果识别的按需型实例价格为 0.4271，并且您指定了 60，则价格阈值将是 0.4271 的 60%。按如下方式计算：0.4271 * 0.60 = 0.25626。计算出的价格是您愿意为竞价型实例支付的最高价格，在本示例中，Amazon EC2 将排除任何价格高于 0.25626 的竞价型实例类型。
- 比已识别的最低价格竞价型实例实例类型高出的百分比 [SpotMaxPricePercentageOverLowestPrice]

  例如，如果识别的竞价型实例价格为 0.1808，并且您指定了 25，则价格阈值将比 0.1808 高 25%。按如下方式计算：0.1808 * 1.25 = 0.226。计算出的价格是您愿意为竞价型实例支付的最高价格，在本示例中，Amazon EC2 将排除任何价格高于 0.266 的竞价型实例类型。我们不建议使用该参数，因为竞价型实例类型的价格可能会波动，因此您的价格保护阈值也可能会波动。

指定价格保护阈值

使用 Amazon CLI 指定价格保护阈值

使用 Amazon CLI 创建 EC2 实例集或竞价型实例集时，请配置实例集，以便使用基于属性的实例类型选择，然后执行以下操作：

- 若要指定按需型实例价格保护阈值，请在 JSON 配置文件的 InstanceRequirements 结构中，为 OnDemandMaxPricePercentageOverLowestPrice 输入以百分比表示的价格保护阈值。
- 要指定竞价型实例的价格保护阈值，请在 JSON 配置文件的 InstanceRequirements 结构中指定以下参数之一：
  - 对于 MaxSpotPriceAsPercentageOfOptimalOnDemandPrice，输入以百分比表示的价格保护阈值。
  - 对于 SpotMaxPricePercentageOverLowestPrice，输入以百分比表示的价格保护阈值。

有关更多信息，请参阅 通过基于属性的实例类型选择创建 EC2 机群 或 通过基于属性的实例类型选择创建竞价型实例集。

（仅适用于竞价型实例集）使用控制台指定价格保护阈值

在控制台中创建竞价型实例集时，请配置实例集，以便使用基于属性的实例类型选择，然后执行以下操作：

- 若要指定按需型实例价格保护阈值，请在其他实例属性下选择按需型价格保护，随后选择添加属性，然后输入以百分比表示的价格保护阈值。
- 若要指定竞价型实例价格保护阈值，请在其他实例属性下选择竞价型价格保护，随后选择添加属性，再选择作为价格依据的基值，然后输入以百分比表示的价格保护阈值。

> **Note**
>
> 创建实例集时，如果将 `TargetCapacityUnitType` 设置为 `vcpu` 或 `memory-mib`，则会基于每个 vCPU 或每个内存的价格，而不是每个实例的价格应用价格保护阈值。

性能保护

性能保护是一项功能，可确保您的 EC2 实例集或竞价型实例集使用的实例类型类似于或超过指定的性能基准。要使用性能保护，请指定一个实例系列作为基准参考。指定的实例系列的功能确立了可接受的最低性能水平。当 Amazon EC2 为您的实例集选择实例类型时，它会考虑您指定的属性和性能基准。低于性能基准的实例类型将被自动排除在选择范围之外，即使它们与您指定的其他属性相匹配。这可确保所有选定的实例类型都提供与指定实例系列建立的基准相似或更好的性能。Amazon EC2 使用此基准来指导实例类型选择，但并不能保证选定实例类型始终会超过每个应用程序的基准。

目前，此功能仅支持将 CPU 性能作为基准性能因素。指定的实例系列的 CPU 处理器的 CPU 性能充当性能基准，确保选定的实例类型与该基准相似或超过该基准。具有相同 CPU 处理器的实例系列会产生相同的筛选结果，即使其网络或磁盘性能有所不同。例如，将 `c6in` 或 `c6i` 指定为基准参考将产生相同的基于性能的筛选结果，因为两个实例系列使用相同的 CPU 处理器。

不支持的实例系列

以下实例系列不支持性能保护：

- 通用型：Mac1 | Mac2 | Mac2-m1ultra | Mac2-m2 | Mac2-m2pro | M1 | M2 | T1

- 计算优化型：C1
- 内存优化型：U-3tb1 | U-6tb1 | U-9tb1 | U-12tb1 | U-18tb1 | u-24tb1 | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb
- 加速计算型：G3 | G3s | P3dn | P4d | P5
- 高性能计算型：Hpc7g

如果您通过指定支持的实例系列来启用性能保护，则返回的实例类型将排除上述不受支持的实例系列。

如果您将不支持的实例系列指定为基准性能值，则 API 会针对 GetInstanceTypesFromInstanceRequirements 返回空响应，对于 CreateFleet、RequestSpotFleet、ModifyFleet 和 ModifySpotFleetRequest 返回异常。

示例：设置 CPU 性能基准

在以下示例中，实例要求是使用具有与 c6i 实例系列一样性能的 CPU 核心的实例类型启动。这将筛选掉 CPU 处理器性能较低的实例类型，即使它们满足您指定的其他实例要求（例如 vCPU 数量）。例如，如果您指定的实例属性包括 4 个 vCPU 和 16 GB 内存，则具有这些属性但 CPU 性能低于 c6i 的实例类型将被排除在选择范围之外。

```
"BaselinePerformanceFactors": {
        "Cpu": {
            "References": [
                {
                    "InstanceFamily": "c6i"
                }
            ]
        }
```

注意事项

- 您可以在 EC2 实例集或竞价型实例集中指定实例类型或实例属性，但不能同时指定两者。

  使用 CLI 时，启动模板覆盖将覆盖启动模板。例如，如果启动模板包含实例类型，而启动模板覆盖包含实例属性，则由实例属性标识的实例将覆盖启动模板中的实例类型。
- 使用 CLI 时，如果将实例属性指定为覆盖，也无法指定权重或优先级。
- 在一个请求配置中，最多可以指定四个 InstanceRequirements 结构。

通过基于属性的实例类型选择创建 EC2 机群

可以配置 EC2 实例集以使用基于属性的实例类型选择。无法通过 Amazon EC2 控制台创建 EC2 实例集。

基于属性的实例类型选择的属性在 `InstanceRequirements` 结构中指定。如果实例集配置中包含 `InstanceRequirements`，则必须排除 `InstanceType` 和 `WeightedCapacity`，因为它们无法同时确定实例集配置与实例属性。

在 Amazon CLI 和 PowerShell 示例中，指定了以下属性：

- `VCpuCount`：最少 2 个 vCPU，最多 4 个 vCPU。如果不需要最大限制，则可省略最大值。
- `MemoryMiB`：内存最小为 8 GiB，最大为 16 GiB。如果不需要最大限制，则可省略最大值。

此配置可识别具有 2 到 4 个 vCPU 和 8 到 16 GiB 内存的所有实例类型。然而，当 EC2 实例集预置实例集时，价格保护和分配策略可能会排除某些实例类型。

Amazon CLI

通过基于属性的实例类型选择创建 EC2 实例集

使用 create-fleet 命令创建 EC2 实例集。在 JSON 文件中指定机群配置。

```
aws ec2 create-fleet \
    --region us-east-1 \
    --cli-input-json file://file_name.json
```

以下示例 *file_name*.json 文件所含参数可配置 EC2 实例集以使用基于属性的实例类型选择。

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized"
    },
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
```

```
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

PowerShell

通过基于属性的实例类型选择创建 EC2 实例集

使用 New-EC2Fleet cmdlet。

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRangeRequest
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiBRequest
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirementsRequest
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object
 Amazon.EC2.Model.FleetLaunchTemplateSpecificationRequest
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.FleetLaunchTemplateOverridesRequest
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.FleetLaunchTemplateConfigRequest
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)
```

```
New-EC2Fleet `
    -SpotOptions_AllocationStrategy "price-capacity-optimized" `
    -LaunchTemplateConfig @($launchTemplateConfig) `
    -TargetCapacitySpecification_DefaultTargetCapacityType "spot" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -Type "instant"
```

通过基于属性的实例类型选择创建竞价型实例集

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

您可以配置实例集以使用基于属性的实例类型选择。

基于属性的实例类型选择的属性在 `InstanceRequirements` 结构中指定。如果实例集配置中包含 `InstanceRequirements`，则必须排除 `InstanceType` 和 `WeightedCapacity`，因为它们无法同时确定实例集配置与实例属性。

在 Amazon CLI 和 PowerShell 示例中，指定了以下属性：

- `VCpuCount`：最少 2 个 vCPU，最多 4 个 vCPU。如果不需要最大限制，则可省略最大值。
- `MemoryMiB`：内存最小为 8 GiB，最大为 16 GiB。如果不需要最大限制，则可省略最大值。

此配置可识别具有 2 到 4 个 vCPU 和 8 到 16 GiB 内存的所有实例类型。然而，当竞价型实例集预置实例集时，价格保护和分配策略可能会排除某些实例类型。

Console

配置竞价型实例集以使用基于属性的实例类型选择

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Spot Requests（竞价型请求），然后选择 Request Spot Instances（请求竞价型实例）。
3. 按照步骤创建竞价型实例集。有关更多信息，请参阅 使用已定义的参数创建竞价型实例集请求。

创建竞价型实例集时，按如下方式配置机群以使用基于属性的实例类型选择：

a. 对于 Instance type requirements（实例类型要求），选择 Specify instance attributes that match your compute requirements（指定符合计算要求的实例属性）。
b. 对于 vCPU，输入所需的最小和最大 vCPU 数。要指定没有限制，请选择 No minimum（没有最小值）和/或 No maximum（没有最大值）。
c. 对于 Memory (GiB)（内存 (GiB)），输入所需的最小和最大内存量。要指定没有限制，请选择 No minimum（没有最小值）和/或 No maximum（没有最大值）。
d. （可选）对于 Additional instance attributes（其它实例属性），您可以选择指定一个或多个属性以更详细地表达计算要求。每个额外属性都会进一步增加对您的请求的限制。
e. （可选）展开 Preview matching instance types（预览匹配的实例类型），查看具有指定属性的实例类型。

Amazon CLI

配置竞价型实例集以使用基于属性的实例类型选择

使用 request-spot-fleet 命令创建竞价型实例集。在 JSON 文件中指定机群配置。

```
aws ec2 request-spot-fleet \
    --region us-east-1 \
    --spot-fleet-request-config file://file_name.json
```

以下示例 *file_name*.json 文件所含参数可配置竞价型实例集以使用基于属性的实例类型选择。

```
{
    "AllocationStrategy": "priceCapacityOptimized",
    "TargetCapacity": 20,
    "Type": "request",
    "LaunchTemplateConfigs": [{
        "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1"
        },
        "Overrides": [{
            "InstanceRequirements": {
                "VCpuCount": {
                    "Min": 2,
                    "Max": 4
```

```
                },
                "MemoryMiB": {
                    "Min": 8192,
                    "Max": 16384
                }
            }
        }]
    }]
}
```

PowerShell

配置竞价型实例集以使用基于属性的实例类型选择

使用 Request-EC2SpotFleet cmdlet。

```
$vcpuCount = New-Object Amazon.EC2.Model.VCpuCountRange
$vcpuCount.Min = 2
$vcpuCount.Max = 4
$memoryMiB = New-Object Amazon.EC2.Model.MemoryMiB
$memoryMiB.Min = 8192
$memoryMiB.Max = 16384
$instanceRequirements = New-Object Amazon.EC2.Model.InstanceRequirements
$instanceRequirements.VCpuCount = $vcpuCount
$instanceRequirements.MemoryMiB = $memoryMiB

$launchTemplateSpec = New-Object Amazon.EC2.Model.FleetLaunchTemplateSpecification
$launchTemplateSpec.LaunchTemplateName = "my-launch-template"
$launchTemplateSpec.Version = "1"
$override = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
$override.InstanceRequirements = $instanceRequirements
$launchTemplateConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchTemplateConfig.LaunchTemplateSpecification = $launchTemplateSpec
$launchTemplateConfig.Overrides = @($override)

Request-EC2SpotFleet `
    -SpotFleetRequestConfig_AllocationStrategy "PriceCapacityOptimized" `
    -SpotFleetRequestConfig_TargetCapacity 20 `
    -SpotFleetRequestConfig_Type "Request" `
    -SpotFleetRequestConfig_LaunchTemplateConfig $launchTemplateConfig
```

有效 EC2 实例集配置示例和无效 EC2 实例集配置示例

如果您使用 Amazon CLI 创建 EC2 机群，必须确保机群配置有效。以下示例展示了有效配置和无效配置。

如果配置包含以下项，则视为无效：

- 单个 Overrides 结构，但同时包含 InstanceRequirements 和 InstanceType
- 两个 Overrides 结构，其中一个包含 InstanceRequirements，而另一个包含 InstanceType
- 两个 InstanceRequirements 结构，但在同一 LaunchTemplateSpecification 中具有重叠的属性值

示例配置

- 有效配置：具有覆盖的单个启动模板
- 有效配置：具有多个 InstanceRequirements 的单个启动模板
- 有效配置：两个启动模板，每个都具有覆盖
- 有效配置：仅指定 InstanceRequirements，且属性值不重叠
- 配置无效：Overrides 包含 InstanceRequirements 和 InstanceType
- 配置无效：两个 Overrides 包含 InstanceRequirements 和 InstanceType
- 配置无效：属性值重叠

有效配置：具有覆盖的单个启动模板

以下配置有效。包含一个启动模板和一个 Overrides 结构（包含一个 InstanceRequirements 结构）。下面是示例配置的文本说明。

```
{
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
```

```
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5000,
        "DefaultTargetCapacityType": "spot",
        "TargetCapacityUnitType": "vcpu"
        }
    }
}
```

InstanceRequirements

要使用基于属性的实例选择，您必须在机群配置中包含 InstanceRequirements 结构，并为机群中的实例指定所需的属性。

在上述示例中，指定了以下实例属性：

- VCpuCount – 实例类型的 vCPU 数量必须最少为 2 个，最多为 8 个。
- MemoryMiB – 实例类型的最大内存必须为 10240MiB。最小值 0 表示没有最低限制。
- MemoryGiBPerVCpu – 实例类型的每个 vCPU 内存最大值必须为 10000GiB。Min 参数是可选的。省略此属性表示没有最低限制。

TargetCapacityUnitType

TargetCapacityUnitType 参数为目标容量指定单位。在该示例中，目标容量是 5000，目标容量单位类型是 vcpu，它们共同指定了所需的 5000 个 vCPU 目标容量。EC2 机群将启动足够多的实例，以便机群中的 vCPU 总数为 5000 个 vCPU。

有效配置：具有多个 InstanceRequirements 的单个启动模板

以下配置有效。包含一个启动模板和一个 Overrides 结构（包含两个 InstanceRequirements 结构）。InstanceRequirements 中指定的属性有效，因为值不重叠，第一个 InstanceRequirements 结构指定 VCpuCount 为 0-2 个 vCPU，而第二个 InstanceRequirements 结构指定 4-8 个 vCPU。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
```

```
        }
    }
}
```

有效配置：两个启动模板，每个都具有覆盖

以下配置有效。包含两个启动模板，每个模板都有一个 Overrides 结构（包含一个 InstanceRequirements 结构）。此配置适用于在同一机群中支持 arm 和 x86 架构。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
```

```
                    }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

有效配置：仅指定 **InstanceRequirements**，且属性值不重叠

以下配置有效。包含两个 LaunchTemplateSpecification 结构，每个结构都有一个启动模板和一个 Overrides 结构（包含一个 InstanceRequirements 结构）。InstanceRequirements 中指定的属性有效，因为值不重叠，第一个 InstanceRequirements 结构指定 VCpuCount 为 0-2 个 vCPU，而第二个 InstanceRequirements 结构指定 4-8 个 vCPU。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
```

```
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

配置无效：**Overrides** 包含 **InstanceRequirements** 和 **InstanceType**

以下配置无效。Overrides 结构同时包含 InstanceRequirements 和 InstanceType。对于 Overrides，可以指定 InstanceRequirements 或 InstanceType，但不能同时指定两者。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
```

```
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

配置无效：两个 Overrides 包含 InstanceRequirements 和 InstanceType

以下配置无效。Overrides 结构同时包含 InstanceRequirements 和 InstanceType。您可以指定 InstanceRequirements 或 InstanceType，但不能同时指定两者，即使采用不同的 Overrides 结构也是如此。

```
{
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
```

```
            ]
        },
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "MyOtherLaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
            {
                "InstanceType": "m5.large"
            }
          ]
        }
    ],
     "TargetCapacitySpecification": {
        "TotalTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
  }
}
```

配置无效：属性值重叠

以下配置无效。两个 InstanceRequirements 结构，每个结构都包含 "VCpuCount": {"Min": 0, "Max": 2}。这些属性的值重叠，这将导致容量池重复。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "MyLaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
            {
                "InstanceRequirements": {
                    "VCpuCount": {
                        "Min": 0,
                        "Max": 2
                    },
                    "MemoryMiB": {
                        "Min": 0
                    }
```

```
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
                }
              ]
            }
        ],
         "TargetCapacitySpecification": {
            "TotalTargetCapacity": 1,
            "DefaultTargetCapacityType": "spot"
        }
    }
}
```

有效竞价型实例集配置示例和无效竞价型实例集配置示例

如果您使用 Amazon CLI 创建竞价型实例集，必须确保机群配置有效。以下示例展示了有效配置和无效配置。

如果配置包含以下项，则视为无效：

- 单个 `Overrides` 结构，但同时包含 `InstanceRequirements` 和 `InstanceType`
- 两个 `Overrides` 结构，其中一个包含 `InstanceRequirements`，而另一个包含 `InstanceType`
- 两个 `InstanceRequirements` 结构，但在同一 `LaunchTemplateSpecification` 中具有重叠的属性值

示例配置

- 有效配置：具有覆盖的单个启动模板
- 有效配置：具有多个 InstanceRequirements 的单个启动模板
- 有效配置：两个启动模板，每个都具有覆盖
- 有效配置：仅指定 InstanceRequirements，且属性值不重叠

- 配置无效：Overrides 包含 InstanceRequirements 和 InstanceType
- 配置无效：两个 Overrides 包含 InstanceRequirements 和 InstanceType
- 配置无效：属性值重叠

有效配置：具有覆盖的单个启动模板

以下配置有效。包含一个启动模板和一个 Overrides 结构（包含一个 InstanceRequirements 结构）。下面是示例配置的文本说明。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "My-launch-template",
                "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 2,
                            "Max": 8
                        },
                        "MemoryMib": {
                            "Min": 0,
                            "Max": 10240
                        },
                        "MemoryGiBPerVCpu": {
                            "Max": 10000
                        },
                        "RequireHibernateSupport": true
                    }
                }
            ]
        }
    ],
        "TargetCapacity": 5000,
```

```
            "OnDemandTargetCapacity": 0,
            "TargetCapacityUnitType": "vcpu"
    }
}
```

InstanceRequirements

要使用基于属性的实例选择，您必须在机群配置中包含 InstanceRequirements 结构，并为机群中的实例指定所需的属性。

在上述示例中，指定了以下实例属性：

- VCpuCount – 实例类型的 vCPU 数量必须最少为 2 个，最多为 8 个。
- MemoryMiB – 实例类型的最大内存必须为 10240MiB。最小值 0 表示没有最低限制。
- MemoryGiBPerVCpu – 实例类型的每个 vCPU 内存最大值必须为 10000GiB。Min 参数是可选的。省略此属性表示没有最低限制。

TargetCapacityUnitType

TargetCapacityUnitType 参数为目标容量指定单位。在该示例中，目标容量是 5000，目标容量单位类型是 vcpu，它们共同指定了所需的 5000 个 vCPU 目标容量。竞价型实例集将启动足够多的实例，以便机群中的 vCPU 总数为 5000 个 vCPU。

有效配置：具有多个 InstanceRequirements 的单个启动模板

以下配置有效。包含一个启动模板和一个 Overrides 结构（包含两个 InstanceRequirements 结构）。InstanceRequirements 中指定的属性有效，因为值不重叠，第一个 InstanceRequirements 结构指定 VCpuCount 为 0-2 个 vCPU，而第二个 InstanceRequirements 结构指定 4-8 个 vCPU。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
```

```
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

有效配置：两个启动模板，每个都具有覆盖

以下配置有效。包含两个启动模板，每个模板都有一个 Overrides 结构（包含一个 InstanceRequirements 结构）。此配置适用于在同一机群中支持 arm 和 x86 架构。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
```

```
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "armLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                },
                {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "x86LaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
```

```
}
```

有效配置：仅指定 **InstanceRequirements**，且属性值不重叠

以下配置有效。包含两个 LaunchTemplateSpecification 结构，每个结构都有一个启动模板和一个 Overrides 结构（包含一个 InstanceRequirements 结构）。InstanceRequirements 中指定的属性有效，因为值不重叠，第一个 InstanceRequirements 结构指定 VCpuCount 为 0-2 个 vCPU，而第二个 InstanceRequirements 结构指定 4-8 个 vCPU。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
```

```
                        "VCpuCount": {
                            "Min": 4,
                            "Max": 8
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    }
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

配置无效：**Overrides** 包含 **InstanceRequirements** 和 **InstanceType**

以下配置无效。Overrides 结构同时包含 InstanceRequirements 和 InstanceType。对于 Overrides，可以指定 InstanceRequirements 或 InstanceType，但不能同时指定两者。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
```

```
                            "Min": 0
                        }
                    }
                },
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

配置无效：两个 **Overrides** 包含 **InstanceRequirements** 和 **InstanceType**

以下配置无效。Overrides 结构同时包含 InstanceRequirements 和 InstanceType。您可以指定 InstanceRequirements 或 InstanceType，但不能同时指定两者，即使采用不同的 Overrides 结构也是如此。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
```

```
                        }
                    }
                }
              ]
            },
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyOtherLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceType": "m5.large"
                }
              ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

配置无效：属性值重叠

以下配置无效。两个 InstanceRequirements 结构，每个结构都包含 "VCpuCount": {"Min": 0, "Max": 2}。这些属性的值重叠，这将导致容量池重复。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "MyLaunchTemplate",
                    "Version": "1"
                },
                "Overrides": [
                {
                    "InstanceRequirements": {
```

```
                        "VCpuCount": {
                            "Min": 0,
                            "Max": 2
                        },
                        "MemoryMiB": {
                            "Min": 0
                        }
                    },
                    {
                      "InstanceRequirements": {
                          "VCpuCount": {
                              "Min": 0,
                              "Max": 2
                          },
                          "MemoryMiB": {
                              "Min": 0
                          }
                      }
                  }
                ]
            }
        ],
        "TargetCapacity": 1,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain"
    }
}
```

预览具有指定属性的实例类型

您可以使用 get-instance-types-from-instance-requirements 命令预览与您指定的属性匹配的实例类型。这对于在不启动任何实例的情况下确定要在请求配置中指定的属性尤其有用。请注意，该命令不考虑可用容量。

通过使用 Amazon CLI 指定属性来预览实例类型列表

1. （可选）要生成所有可以指定的可能属性，请使用 get-instance-types-from-instance-requirements 命令和 `--generate-cli-skeleton` 参数。您可以选择使用 `input > attributes.json` 将输出定向到某个文件以将其保存。

```
aws ec2 get-instance-types-from-instance-requirements \
```

```
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

预期输出

```
{
    "DryRun": true,
    "ArchitectureTypes": [
        "i386"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 0,
            "Max": 0
        },
        "MemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "CpuManufacturers": [
            "intel"
        ],
        "MemoryGiBPerVCpu": {
            "Min": 0.0,
            "Max": 0.0
        },
        "ExcludedInstanceTypes": [
            ""
        ],
        "InstanceGenerations": [
            "current"
        ],
        "SpotMaxPricePercentageOverLowestPrice": 0,
        "OnDemandMaxPricePercentageOverLowestPrice": 0,
        "BareMetal": "included",
        "BurstablePerformance": "included",
        "RequireHibernateSupport": true,
        "NetworkInterfaceCount": {
            "Min": 0,
            "Max": 0
```

```
        },
        "LocalStorage": "included",
        "LocalStorageTypes": [
            "hdd"
        ],
        "TotalLocalStorageGB": {
            "Min": 0.0,
            "Max": 0.0
        },
        "BaselineEbsBandwidthMbps": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorTypes": [
            "gpu"
        ],
        "AcceleratorCount": {
            "Min": 0,
            "Max": 0
        },
        "AcceleratorManufacturers": [
            "nvidia"
        ],
        "AcceleratorNames": [
            "a100"
        ],
        "AcceleratorTotalMemoryMiB": {
            "Min": 0,
            "Max": 0
        },
        "NetworkBandwidthGbps": {
            "Min": 0.0,
            "Max": 0.0
        },
        "AllowedInstanceTypes": [
            ""
        ]
    },
    "MaxResults": 0,
    "NextToken": ""
}
```

2. 使用上一步的输出创建 JSON 配置文件，然后按如下方式进行配置：

> **Note**
>
> 您必须提供 ArchitectureTypes、VirtualizationTypes、VCpuCount 和 MemoryMiB 的值。您可以省略其它属性；如果省略，则使用默认值。
> 有关每个属性及其默认值的描述，请参阅 get-instance-types-from-instance-requirements。

a. 对于 ArchitectureTypes，请指定一个或多个处理器架构类型。

b. 对于 VirtualizationTypes，请指定一个或多个虚拟化类型。

c. 对于 VCpuCount，请指定最小和最大 vCPU 数量。要指定没有最低限制，对于 Min，请指定 0。要指定没有最大限制，请省略 Max 参数。

d. 对于 MemoryMiB，请指定最小和最大内存量（以 MiB 为单位）。要指定没有最低限制，对于 Min，请指定 0。要指定没有最大限制，请省略 Max 参数。

e. 您可以选择指定一个或多个其他属性来进一步限制返回的实例类型列表。

3. 要预览具有您在 JSON 文件中所指定属性的实例类型，请使用 get-instance-types-from-instance-requirements 命令，然后使用 --cli-input-json 参数指定 JSON 文件的名称和路径。您可以选择将输出格式设置为以表格格式显示。

```
aws ec2 get-instance-types-from-instance-requirements \
    --cli-input-json file://attributes.json \
    --output table
```

示例 *attributes.json* 文件

在此示例中，JSON 文件包含所需属性。它们是 ArchitectureTypes、VirtualizationTypes、VCpuCount 和 MemoryMiB。此外，还包含可选属性 InstanceGenerations。请注意，对于 MemoryMiB，可以省略 Max 值，指示没有限制。

```
{

    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
```

```
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

输出示例

```
-----------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+---------------------------------------+
||             InstanceTypes             ||
|+-------------------------------------+|
||             InstanceType              ||
|+-------------------------------------+|
||  c4.xlarge                          ||
||  c5.xlarge                          ||
||  c5a.xlarge                         ||
||  c5ad.xlarge                        ||
||  c5d.xlarge                         ||
||  c5n.xlarge                         ||
||  d2.xlarge                          ||
...
```

4. 在确定满足您的需求的实例类型后，请记下您使用的实例属性，以便在配置机群请求时可以使用。

使用实例权重来管理 EC2 实例集或竞价型实例集的成本和性能

您可以借助实例权重为 EC2 实例集或竞价型实例集中的每种实例类型分配权重，来表示它们相对于彼此的计算容量和性能。实例集可以根据权重使用指定实例类型的任意组合，前提是可以满足所需的目标容量。这有助于管理实例集的成本和性能。

权重表示实例类型占总目标容量的容量单位。

示例：使用实例权重管理性能

假设实例集包含两种实例类型，并且每种实例类型都分配到了不同的权重，来反映实现相同性能所需的每种实例类型的数量，如下所示：

- m5.large – 权重：1
- m5.2xlarge – 权重：4

如此分配权重，即表明需要 4 个 m5.large 实例才能实现与 1 个 m5.2xlarge 实例相同的性能。

要计算给定目标容量所需的每种实例类型的实例数量，请使用以下公式：

```
target capacity / weight = number of instances
```

如果目标容量为 8 个单位，则实例集可以通过 m5.large 或 m5.2xlarge 或混合使用两者来满足目标容量，如下所示：

- 8 个 m5.large 实例（容量为 8/权重为 1，表示需要 8 个实例）
- 2 个 m5.2xlarge 实例（容量为 8/权重为 4，表示需要 2 个实例）
- 4 个 m5.large 和 1 个 m5.2xlarge

示例：使用实例权重管理成本

默认情况下，您指定的价格是每实例小时 价格。在使用实例权重功能时，您指定的价格是每单位小时价格。您可以将某种实例类型的价格除以它表示的单位数来计算每单位小时价格。实例集通过将目标容量除以实例权重，计算出要启动的实例数。如果结果不是整数，则 队列会将其向上舍入到下一个整数，以便队列的大小不低于其目标容量。队列可以选择您在启动规范中指定的任意池，即使所启动实例的容量超过请求的目标容量也是如此。

下表提供了用于确定目标容量为 10 的实例集每单位价格的计算示例。

实例类型	实例权重	目标容量	启动的实例数	每实例小时价格	每单位小时价格
r3.xlarge	2	10	5 (10 除以 2)	0.05 美元	0.025 美元 (0.05 除以 2)

实例类型	实例权重	目标容量	启动的实例数	每实例小时价格	每单位小时价格
r3.8xlarge	8	10	2 (10 除以 8，结果向上舍入)	0.10 美元	0.0125 美元 (0.10 除以 8)

按如下所示使用实例集实例权重，在履行时以最低的单位价格预置池中所需的目标容量：

1. 采用实例（默认设置）或所选单位（如 vCPU、内存、存储或吞吐量）为实例集设置目标容量。
2. 设置每单位价格。
3. 为每个启动规范指定权重，这是实例类型向目标容量提供的单位数。

实例权重示例

考虑具有以下配置的实例集请求：

- 目标容量为 24
- 一个实例类型为 r3.2xlarge 且权重为 6 的启动规范
- 一个实例类型为 c3.xlarge 且权重为 5 的启动规范

每个权重表示相应实例类型向目标容量提供的单位数。如果第一个启动规范提供了最低单位价格（r3.2xlarge 每实例小时价格除以 6），则实例集会启动 4 个这样的实例（24 除以 6）。

如果第二个启动规范提供了最低单位价格（c3.xlarge 每实例小时价格除以 5），则实例集会启动 5 个这样的实例（24 除以 5，结果向上取整）。

实例权重和分配策略

考虑具有以下配置的实例集请求：

- 目标容量为 30 个竞价型实例
- 一个实例类型为 c3.2xlarge 且权重为 8 的启动规范
- 一个实例类型为 m3.xlarge 且权重为 8 的启动规范
- 一个实例类型为 r3.xlarge 且权重为 8 的启动规范

实例集会启动四个实例（30 除以 8，结果向上取整）。使用 diversified 策略时，队列会在所有三个池中各启动一个实例，并在三个池中提供最低每单位价格的那个池中启动第四个实例。

使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量

在 EC2 实例集或竞价型实例集中使用多个容量池（每个容量池都包含一个实例类型和可用区）时，可以使用分配策略来管理 Amazon EC2 如何利用这些容量池满足竞价型和按需型容量。分配策略可以针对可用容量、价格和要使用的实例类型进行优化。竞价型实例和按需型实例有不同的分配策略。

主题

- 竞价型实例的分配策略
- 按需型实例的分配策略
- 选择合适的竞价型分配策略
- 维持竞价型实例的目标容量
- 针对按需容量优化实例类型

竞价型实例的分配策略

您的启动配置决定了 EC2 实例集或竞价型实例集可以从中启动竞价型实例的所有可能竞价型容量池（实例类型和可用区）。但在启动实例时，实例集将使用您指定的分配策略从所有可能的池中选择特定的池。

> **Note**
>
> （仅限 Linux 实例）如果将竞价型实例配置为启动并开启 AMD SEV-SNP，则您需要按小时支付额外的使用费，费率为所选实例类型按需小时费率的 10%。如果分配策略使用价格作为输入，则实例集不包括这笔额外费用；只使用竞价型价格。

您可以为竞价型实例指定以下分配策略之一：

价格容量优化（推荐）

实例集根据正在启动的实例数确定具有最高可用容量的池。这意味着我们将从我们认为短期内中断概率最低的池中请求竞价型实例。然后，实例集从这些池中价格最低的池请求竞价型实例。

价格容量优化分配策略是大多数竞价型工作负载的最佳选择，例如无状态容器化应用程序、微服务、Web 应用程序、数据和分析任务以及批处理。

如果使用的是 Amazon CLI，则 EC2 实例集的参数名称为 `price-capacity-optimized`，竞价型实例集的参数名称为 `priceCapacityOptimized`。

容量优化

实例集根据正在启动的实例数确定具有最高可用容量的池。这意味着我们将从我们认为短期内中断概率最低的池中请求竞价型实例。

使用竞价型实例，定价会根据长期供需趋势缓慢发生变化，但容量会实时波动。容量优化策略通过查看实时容量数据并预测可用性最高的池，自动在可用性最高的池中启动竞价型实例。这适用于因重启工作而导致中断成本更高的工作负载，例如长时间持续集成（CI）、图像和媒体渲染、深度学习以及高性能计算（HPC）工作负载，这些工作负载可能具有与重启工作相关的更高的中断成本。通过实现更低的中断可能性，容量优化策略可以降低工作负载的整体成本。

或者，您也可以使用容量优化优先级分配策略，该策略带有优先级参数，可从最高到最低优先级对实例类型进行排序。您可以为不同的实例类型设置相同的优先级。实例集首先会针对容量进行优化，但会尽力遵循实例类型的优先级（例如，如果遵循优先级不会显著影响实例集预置最佳容量的能力）。对于必须最大限度地减少中断可能性，同时对某些实例类型的偏好也很重要的工作负载来说，这是一个不错的选择。请注意，为竞价型容量的实例类型设置优先级时，如果按需型分配策略设置为优先，那么相同的优先级也会应用于按需型实例。对于竞价型实例集，仅当实例集使用启动模板时，才支持使用优先级。

如果使用的是 Amazon CLI，则 EC2 实例集的参数名称为 `capacity-optimized` 和 `capacity-optimized-prioritized`，竞价型实例集的参数名称为 `capacityOptimized` 和 `capacityOptimizedPrioritized`。

多元化

竞价型实例 分布在所有 Spot 容量池中。如果使用的是 Amazon CLI，则 EC2 实例集和竞价型实例集的参数名称皆为 `diversified`。

最低价格（不推荐）

> **Warning**
>
> 不推荐使用最低价格分配策略，因为这对竞价型实例造成的中断风险最高。

竞价型实例来自价格最低并且具有可用容量的池。这是使用 Amazon CLI 时的默认策略。不过，建议指定价格容量优化分配策略来覆盖默认值。

借助最低价格策略，如果价格最低的池没有可用容量，则竞价型实例将来自价格次低且具有可用容量的池。如果某个池在满足所需容量之前耗尽容量，则实例集将继续从价格次低的池中提取实例来满足请求。为确保达到所需容量，您可能会获得来自多个容量池的竞价型实例。

由于此策略仅考虑实例价格而不考虑容量可用性，因此可能会导致较高的中断率。

最低价格分配策略仅在使用 Amazon CLI 时可用。EC2 实例集的参数名称为 `lowest-price`，竞价型实例集的参数名称为 `lowestPrice`。

要使用的池数量

在其中分配您的目标 Spot 容量的 Spot 池数量。仅当分配策略设置为最低价格时有效。实例集选择价格最低的竞价型池，并在指定数量的竞价型池之间平均分配目标竞价型容量。

请注意，实例集会尽力根据指定的池数量提取竞价型实例。如果池在满足目标容量之前耗尽竞价型容量，则实例集将继续从价格次低的池中提取实例来满足请求。为确保达到目标容量，您可能会从超过指定数量的池接收竞价型实例。同样，如果大多数池没有 Spot 容量，则您可能会从少于指定数量的池接收完整目标容量。

此参数仅在指定最低价格分配策略且使用 Amazon CLI 时可用。EC2 实例集和竞价型实例集的参数名称皆为 `InstancePoolsToUseCount`。

按需型实例的分配策略

您的启动配置决定了 EC2 实例集或竞价型实例集可以从中启动按需型实例的所有可能容量池（实例类型和可用区）。但在启动实例时，实例集将使用您指定的分配策略从所有可能的池中选择特定的池。

您可以为按需型实例指定以下分配策略之一：

最低价格

按需型实例来自价格最低且具有可用容量的池。这是默认策略。

如果价格最低的池没有可用容量，则按需型实例将来自价格次低且具有可用容量的池。

如果某个池在满足所需容量之前耗尽容量，则实例集将继续从价格次低的池中提取实例来满足请求。为确保满足所需容量，您可能会获得来自多个容量池的按需型实例。

优先级

实例集使用为每个启动模板覆盖分配的优先级，先启动最高优先级的实例类型。此策略不能与基于属性的实例类型选择一起使用。有关如何使用此分配策略的示例，请参阅[针对按需容量优化实例类型]。

选择合适的竞价型分配策略

您可以选择适当的 Spot 分配策略，从而根据您的使用案例来优化实例集。

平衡最低价格和可用容量

为了在价格最低的竞价型容量池与具有最高可用容量的竞价型容量池之间取得平衡，建议使用价格容量优化分配策略。该策略根据池的价格以及这些池中竞价型实例的可用容量来决定向哪些池请求竞价型实例。这意味着将从我们认为短期内中断概率最低的池中请求竞价型实例，同时还要考虑价格。

如果实例集运行弹性和无状态的工作负载（包括容器化应用程序、微服务、Web 应用程序、数据和分析作业以及批量处理），则使用价格容量优化分配策略来实现最大的成本节约和可用容量。

如果实例集运行的工作负载可能会因重启工作而导致更高的中断成本，则应设置检查点，以便应用程序中断时可从该点重启。通过使用检查点，可以让价格容量优化分配策略变得非常适用于这些工作负载，因为其会从价格最低的池中分配容量，这些池同时还具有较低的竞价型实例中断率。

有关使用价格容量优化分配策略的 JSON 配置示例，请参阅以下内容：

- EC2 实例集 – 示例 10：在价格容量优化的实例集中启动竞价型实例
- 竞价型实例集 – 示例 11：在 priceCapacityOptimized 实例集中启动竞价型实例

当工作负载拥有较高中断成本时

如果您运行的工作负载使用价格相似的实例类型，或中断成本非常高，以至于任何成本节约都不足以负担一个轻微中断，则可以选择使用容量优化策略。此策略从可用容量最多的 Spot 容量池分配容量，这些容量池中断的可能性更低，从而可降低您工作负载的整体成本。

如果必须将中断的可能性降至最低，但某些实例类型的首选项又很紧要时，则可使用容量优化优先级分配策略来表达池的优先级，然后按从最高到最低优先级的顺序设置要使用的实例类型。

请注意，为容量优化优先级设置优先级时，如果按需型分配策略设置为优先，那么相同的优先级也会应用于按需型实例。另请注意，对于竞价型实例集，仅当实例集使用启动模板时，才支持使用优先级。

有关使用容量优化分配策略的 JSON 配置示例，请参阅以下内容：

- EC2 实例集 – 示例 8：在容量优化的队列中启动竞价型实例
- 竞价型实例集 – 示例 9：在容量优化的队列中启动竞价型实例

有关使用容量优化优先级分配策略的 JSON 配置示例，请参阅以下内容：

- EC2 实例集 – 示例 9：在具有优先级的容量优化队列中启动竞价型实例
- 竞价型实例集 – 示例 10：在具有优先级的容量优化队列中启动竞价型实例

当您的工作负载时间灵活且可用容量不是一个因素时

如果实例集较小或运行时间较短，则可以使用价格容量优化来最大限度地节省成本，同时还要考虑可用容量。

当您的实例集较大或运行时间较长时

如果实例集较大或运行时间较长，则可以使用多元化策略在多个池间分配竞价型实例来提高实例集的可用性。例如，如果实例集指定 10 个池，目标容量为 100 个实例，则实例集会在每个池中启动 10 个竞价型实例。如果某个池的 Spot 价格超过您在该池中的最高价，您的队列仅 10% 受到影响。使用此策略还可降低您的队列对单个池的 Spot 价格随时间上涨的敏感度。使用多元化策略时，实例集不会在竞价型价格等于或高于按需型价格的任何池中启动竞价型实例。

维持竞价型实例的目标容量

在竞价型实例因竞价型价格或竞价型容量池的可用容量发生变化而终止之后，`maintain` 类型的实例集会启动替换竞价型实例。分配策略确定从哪个池启动替换实例，如下所示：

- 如果分配策略为价格容量优化，则实例集在具有最高竞价型实例可用容量的池中启动替换实例，同时还要考虑价格，并确定价格最低且具有高可用容量的池。
- 如果分配策略为容量优化，则实例集在具有最高竞价型实例可用容量的池中启动替换实例。
- 如果分配策略为多元化，则实例集在其余池间分配替换竞价型实例。

针对按需容量优化实例类型

EC2 实例集或竞价型实例集尝试满足按需型容量时，默认首先启动价格最低的实例类型。如果按需型分配策略设置为优先，则实例集使用优先级确定在满足按需型容量时先使用哪个实例类型。优先级分配给启动模板覆盖，优先级最高的最先启动。

示例：确定实例类型的优先级

在本例中，您可以配置三个启动模板覆盖，每个覆盖具有不同的实例类型。

实例类型的按需价格在价格范围内。以下是本例中使用的实例类型，按价格顺序列出，从价格最低的实例类型开始：

- m4.large – 价格最低
- m5.large
- m5a.large

如果您不使用优先级来确定顺序，则实例集将从价格最低的实例类型开始满足按需容量。

但是，假设您有想要先使用的未用 m5.large 预留实例。您可以设置启动模板覆盖优先级，以便按优先级顺序使用实例类型，如下所示：

- m5.large – 优先级 1
- m4.large – 优先级 2
- m5a.large – 优先级 3

在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例

借助“容量再平衡”功能，EC2 实例集或竞价型实例集可以通过主动替换存在中断风险的竞价型实例来维持所需的竞价型容量。竞价型实例面临较高的中断风险时，Amazon EC2 会发送再平衡建议。如果启用了“容量再平衡”功能，则再平衡建议会在存在风险的实例中断之前，触发启动新的竞价型实例。

“容量再平衡”功能可在运行中的实例被 Amazon EC2 中断之前，主动使用新竞价型实例扩展实例集，从而帮助维持工作负载的可用性。

配置 EC2 实例集使用“容量再平衡”功能来启动替换竞价型实例

使用 create-fleet 命令和 MaintenanceStrategies 结构中的相关参数。有关 JSON 配置示例，请参阅示例 7：配置容量再平衡以启动替换竞价型实例。

配置竞价型实例集使用“容量再平衡”功能来启动替换竞价型实例

您可以使用 Amazon EC2 控制台或 Amazon CLI 来配置“容量再平衡”功能。

（控制台）创建竞价型实例集时，选中容量重新平衡复选框。有关更多信息，请参阅使用已定义的参数创建竞价型实例集请求中的步骤 6.d..

（Amazon CLI）使用 request-spot-fleet 命令和 SpotMaintenanceStrategies 结构中的相关参数。有关 JSON 配置示例，请参阅示例 8：配置容量再平衡以启动替换竞价型实例。

主题

- 限制
- 配置选项
- 注意事项

限制

- 容量再平衡仅适用于类型为 maintain 的机群。
- 队列运行时，您无法修改容量再平衡设置。要更改容量再平衡设置，您必须删除此队列并创建新队列。

配置选项

EC2 实例集和竞价型实例集的 ReplacementStrategy 支持以下两个值：

launch-before-terminate

Amazon EC2 可以在新的替换竞价型实例启动后终止接收再平衡通知的竞价型实例。如果指定 launch-before-terminate，还必须为 termination-delay 指定值。启动新的替换实例后，Amazon EC2 将等待 termination-delay 的持续时间，然后终止旧实例。对于 termination-delay，最短为 120 秒（2 分钟），最长为 7200 秒（2 小时）。

建议仅当您可以预测实例关闭过程完成所需的时长时才使用 launch-before-terminate。这将确保只有在关闭过程完成后才终止旧实例。请注意，Amazon EC2 可能会在 termination-delay 之前中断旧实例，并发出两分钟警告。

强烈建议不要将 lowest-price（EC2 实例集）或 lowestPrice（竞价型实例集）分配策略与 launch-before-terminate 结合使用，避免让替换竞价型实例面临较高的中断风险。

launch

针对现有竞价型实例发出再平衡通知时，Amazon EC2 将启动替换竞价型实例。Amazon EC2 不会终止接收再平衡通知的实例。您可以终止旧实例，也可以使其保持运行状态。在实例运行期间，您需要为它们付费。

注意事项

如果针对“容量再平衡”功能配置 EC2 实例集或竞价型实例集，请考虑以下事项：

在请求中提供尽可能多的 Spot 容量池

将实例集配置为使用多种实例类型和多个可用区。这提供了在各种 Spot 容量池中启动竞价型实例的灵活性。有关更多信息，请参阅 灵活地选择实例类型和可用区。

避免替换竞价型实例中断的风险升高

为避免中断风险增加，我们建议使用 `capacity-optimized` 或 `capacity-optimized-prioritized` 分配策略。这些策略确保替换竞价型实例在最佳的竞价型容量池中启动，因此在不久的将来不太可能被中断。有关更多信息，请参阅 使用价格和容量优化分配策略。

如果使用 `lowest-price` 分配策略，替换竞价型实例可能会面临较高的中断风险。这是因为，即使替换竞价型实例可能在启动后不久中断，Amazon EC2 始终会在当时具有可用容量的价格最低池中启动实例。

只有可用性相同或更好时，Amazon EC2 才会启动新实例

容量再平衡的目标之一是提高竞价型实例的可用性。如果现有竞价型实例收到再平衡建议，Amazon EC2 只有在新实例提供与现有实例相同或更好的可用性时才会启动新实例。如果新实例的中断风险比现有实例更高，那么 Amazon EC2 将不会启动新实例。但是，Amazon EC2 将继续评测 Spot 容量池，并在可用性改善时启动新实例。

如果 Amazon EC2 没有主动启动新实例，则您的现有实例可能会中断。发生这种情况时，无论新实例的中断风险是否较高，Amazon EC2 都会尝试启动新实例。

容量再平衡不会提高您的竞价型实例中断率

当您启用容量再平衡时，它不会提高您的竞价型实例中断率（在 Amazon EC2 需要收回容量时回收的竞价型实例的数量）。然而，如果容量再平衡检测到实例存在中断风险，Amazon EC2 将立即尝试启动新实例。结果是，与存在风险的实例中断后等待 Amazon EC2 启动新实例相比，可能会替换更多实例。

虽然在启用了容量再平衡的情况下您可以更换更多实例，但在中断您的实例之前，您会有更多时间采取行动，从而因为处于主动而非被动地位而受益。使用 Spot Instance interruption notice（竞价型实例中断通知），您通常最多只有两分钟的时间来正常关闭您的实例。借助容量再平衡提前启动新实例，您可以使现有流程更有可能在存在风险的实例上完成，您可以启动实例关闭程序，并防止在存在风险的实例上安排新工作。您还可以开始准备新启动的实例，以接管应用程序。借助容量再平衡的主动替换，您可以因绝佳的连续性而受益。

作为演示使用容量再平衡的风险和优点的理论示例，请考虑以下场景：

- 下午 2:00 – 收到针对实例 A 的再平衡建议，Amazon EC2 立即开始尝试启动替换实例 B，让您有时间启动关闭程序。*

- 下午 2:30 - 收到针对实例 B 的再平衡建议，该实例将被替换为实例 C，让您有时间启动关闭程序。*
- 下午 2:32 - 如果未启用容量再平衡，并且如果在下午 2:32 收到针对实例 A 的竞价型实例中断通知，则您最多只有两分钟时间采取行动，但实例 A 将一直运行到此时。

* 如果指定了 `launch-before-terminate`，Amazon EC2 将在替换实例上线后终止存在风险的实例。

Amazon EC2 可以启动新替换竞价型实例，直到已执行容量达到目标容量的两倍

针对“容量再平衡”功能配置实例集时，实例集会尝试为接收再平衡建议的每个竞价型实例启动新的替换竞价型实例。在竞价型实例收到再平衡建议后，它不再统计为已执行容量的一部分。根据替换策略，Amazon EC2 将在预配置的终止延迟后终止实例，或使其保持运行状态。这使您有机会对实例执行再平衡操作。

如果您的队列达到目标容量的两倍，即使替换实例本身收到再平衡建议，它也会停止启动新的替换实例。

例如，您创建的实例集目标容量为 100 个竞价型实例。所有竞价型实例都会收到再平衡建议，这将导致 Amazon EC2 启动 100 个替换竞价型实例。这使已执行的竞价型实例的数量增加到 200 个，是目标容量的两倍。一些替换实例会收到再平衡建议，但由于实例集不能超过其目标容量的两倍，因此不会再启动更多替换实例。

请注意，在实例运行期间，您需要为所有实例付费。

建议将实例集配置为终止接收再平衡建议的竞价型实例

如果针对“容量再平衡”功能配置实例集，建议仅在可以预测实例关闭过程完成所需的时长时，才选择 `launch-before-terminate` 和适当的终止延迟。这将确保只有在关闭过程完成后才终止旧实例。

如果您选择自行终止建议再平衡的实例，我们建议您监控机群中的竞价型实例接收的再平衡建议信号。通过监控信号，您可以在 Amazon EC2 中断受影响的实例之前快速对其执行再平衡操作，然后您可以手动终止它们。如果您不终止实例，则需要在实例运行期间继续为其付费。Amazon EC2 不会自动终止接收再平衡建议的实例。

您可以使用 Amazon EventBridge 或实例元数据设置通知。有关更多信息，请参阅 监控再平衡建议信号。

在计算横向缩减或横向扩展期间的已执行容量时，实例集会排除收到再平衡建议的实例

如果针对“容量再平衡”功能配置了实例集，并且将目标容量更改为横向缩减或横向扩展，则实例集不会将标记为再平衡的实例计算为已执行容量的一部分，如下所示：

- 横向缩减 – 如果您降低所需的目标容量，Amazon EC2 将终止未标记为再平衡的实例，直到达到所需容量。标记为再平衡的实例不计入已执行容量。

  例如，您创建的竞价型实例集目标容量为 100 个竞价型实例。10 个实例会收到再平衡建议，因此 Amazon EC2 会启动 10 个新的替换实例，从而获得 110 个实例的执行容量。然后，您将目标容量减少到 50 个（横向缩减），但已执行的容量实际上为 60 个实例，因为 Amazon EC2 不会终止标记为再平衡的 10 个实例。您需要手动终止这些实例，也可以让它们保持运行状态。

- 横向扩展 – 如果您增加所需的目标容量，Amazon EC2 将启动新实例，直到达到所需容量。标记为再平衡的实例不计入已执行容量。

  例如，您创建的实例集目标容量为 100 个竞价型实例。10 个实例会收到再平衡建议，因此实例集启动 10 个新的替换实例，从而获得 110 个实例的执行容量。然后，您将目标容量增加到 200（扩展），但已执行的容量实际上为 210 个实例，因为标记为再平衡的 10 个实例不被队列计入目标容量的一部分。您需要手动终止这些实例，也可以让它们保持运行状态。

使用容量预留来预留 EC2 实例集中的按需型容量

可以通过将 容量预留 的使用策略配置为 `use-capacity-reservations-first` 来将队列配置为在启动 按需实例 时首先使用 按需容量预留。可以将此设置与 按需实例 的分配策略（`lowest-price` 或 `prioritized`）结合使用。

在将未使用的容量预留用于实现按需容量时：

- 队列使用未使用的容量预留来实现按需容量，最多可达到目标按需容量。
- 如果多个实例池具有未使用的容量预留，则应用按需分配策略（`lowest-price` 或 `prioritized`）。
- 如果未使用的容量预留数少于按需目标容量，则将根据按需分配策略（`lowest-price` 或 `prioritized`）启动剩余的按需目标容量。

只能将未使用的 按需容量预留 用于 `instant` 类型的队列。

有关如何将队列配置为使用 容量预留 来实现按需容量的示例，请参阅 EC2 实例集 CLI 配置示例。有关更多信息，请参阅使用 EC2 按需容量预留来预留计算容量和按需容量预留常见问题。

使用 EC2 实例集

要开始使用 EC2 实例集，请创建一个请求，其中包含总目标容量、按需型容量、竞价型容量，以及指定实例集中实例配置的启动模板。您可以选择指定其他参数，也可以让实例集使用默认值。您还可以在创建实例集时标记实例集请求及其实例和卷。

实例集在有可用容量时启动按需型实例，在最高价超过 Spot 价格并且有可用容量时启动竞价型实例。

实例集启动后，即可描述实例集请求、实例集中的实例以及任何实例集事件。您也可以根据需要分配额外的标签。

如需更改任何实例集参数（例如总目标容量），则可以修改该实例集，前提是该实例集已配置为保持容量。提交一次性请求后，您无法修改其容量。

实例集请求在过期或您删除它之前一直有效。删除实例集请求时，您可以终止实例或让其保持运行状态。如果选择让实例保持运行，按需型实例会持续运行至您将其终止，竞价型实例会持续运行至遇到中断或您将其终止。

主题

- EC2 实例集请求状态
- EC2 实例集先决条件
- 创建 EC2 实例集
- 标记新的或现有的 EC2 实例集请求及其启动的实例和卷
- 描述 EC2 实例集、其实例及其事件
- 修改 EC2 实例集
- 删除 EC2 实例集请求和实例集中的实例

EC2 实例集请求状态

EC2 实例集请求可以处于多种状态之一，每种状态表示请求生命周期及其实例管理的不同阶段。

EC2 实例集请求可以处于以下某种状态：

`submitted`

EC2 实例集正在接受评估，并且 Amazon EC2 正准备启动目标数量的实例。如果请求超出实例集限制，则会被立即删除。

`active`

EC2 实例集请求已验证，并且 Amazon EC2 正在尝试使正在运行的实例保持目标数量。请求会保持这一状态，直到其被修改或删除。

`modifying`

正在修改 EC2 实例集请求。请求会保持这一状态，直到修改全部完成或请求被删除。只能修改 `maintain` 实例集类型。此状态不适用于其他请求类型。

`deleted_running`

已删除 EC2 实例集请求，且不启动其他竞价型实例。其现有实例将继续运行，直至中断或被手动终止。请求会保持此状态，直到所有实例都已中断或终止。删除 EC2 实例集请求后，只有类型 `maintain` 或 `request` 的 EC2 实例集才能有正在运行的实例。不支持包含正在运行实例的已删除 `instant` 实例集。此状态不适用于 `instant` 实例集。

`deleted_terminating`

EC2 实例集请求已删除，正在终止其实例。请求会保持此状态，直到所有实例都已终止。

`deleted`

EC2 实例集请求已删除，没有正在运行的实例。请求将在其实例终止两天后被删除。

EC2 实例集先决条件

要创建 EC2 实例集，必须满足以下先决条件：

- 启动模板
- EC2 实例集的服务相关角色
- 授予对用于加密的 AMI 和 EBS 快照的客户托管密钥的访问权限
- EC2 实例集用户的权限

启动模板

启动模板指定要启动的实例的相关配置信息，例如实例类型和可用区。有关启动模板的更多信息，请参阅在 Amazon EC2 启动模板中存储实例启动参数。

EC2 实例集的服务相关角色

`AWSServiceRoleForEC2Fleet` 角色授予 EC2 实例集权限以代表您请求、启动、终止和标记实例。Amazon EC2 使用此服务相关角色来完成以下操作：

- `ec2:RunInstances` – 启动实例。
- `ec2:RequestSpotInstances`：请求竞价型实例。
- `ec2:TerminateInstances` – 终止实例。
- `ec2:DescribeImages`：描述实例的亚马逊机器映像（AMI）。
- `ec2:DescribeInstanceStatus`：描述实例的状态。
- `ec2:DescribeSubnets`：描述实例的子网。
- `ec2:CreateTags`：将标签添加到 EC2 实例集、实例和卷中。

确保此角色存在，然后才使用 Amazon CLI 或 API 来创建 EC2 实例集。

> **Note**
>
> `instant` EC2 实例集不需要此角色。

要创建该角色，请如下使用 IAM 控制台。

为 EC2 实例集创建 AWSServiceRoleForEC2Fleet 角色

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）。
3. 选择创建角色。
4. 在选择受信任的实体页面上，请执行以下操作：
   a. 在可信实体类型中选择 Amazon 服务。
   b. 在使用案例下，对于服务或使用案例，选择 EC2 - 实例集。

      > **Tip**
      >
      > 请务必选择 EC2 - 实例集。如果您选择 EC2，则 EC2 - 实例集使用案例不会出现在使用案例列表中。EC2 - 实例集使用案例将自动创建具有所需 IAM 权限的策略，并建议将 AWSServiceRoleForEC2Fleet 作为角色名称。

   c. 选择下一步。
5. 在添加权限页面上，选择下一步。

6. 在命名、检查并创建页面上，选择创建角色。

如果您不再需要使用 EC2 实例集，我们建议您删除 AWSServiceRoleForEC2Fleet 角色。当此角色从您的账户中删除后，如果您创建其他实例集，可再次创建此角色。

有关更多信息，请参阅《IAM 用户指南》中的服务相关角色。

授予对用于加密的 AMI 和 EBS 快照的客户托管密钥的访问权限

如果在 EC2 实例集中指定加密的 AMI 或加密的 Amazon EBS 快照，并且您使用 Amazon KMS 密钥进行加密，则必须授予 AWSServiceRoleForEC2Fleet 角色使用客户托管式密钥的权限，以便 Amazon EC2 可以代表您启动实例。为此，您必须在客户托管密钥中添加授权，如以下过程中所示。

在提供权限时，授权是密钥策略的替代方法。有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的使用授权和在 Amazon KMS 中使用密钥策略。

为 AWSServiceRoleForEC2Fleet 角色授予使用客户托管密钥的权限

- 使用 create-grant 命令在客户托管密钥中添加授权，并指定授予权限的主体（AWSServiceRoleForEC2Fleet 服务相关角色），以执行授权允许的操作。客户托管密钥由 `key-id` 参数和客户托管密钥的 ARN 指定。主体是由 `grantee-principal` 参数和 AWSServiceRoleForEC2Fleet 服务相关角色的 ARN 指定的。

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Fleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

EC2 实例集用户的权限

如果用户打算创建或管理 EC2 实例集，请确保为其授予所需权限。

为 EC2 实例集创建策略

1. 打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择策略。

3. 选择 Create policy。
4. 在创建策略页面上，选择 JSON 选项卡，将文本替换为以下内容，并选择查看策略。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
              "iam:ListRoles",
              "iam:PassRole",
              "iam:ListInstanceProfiles"
            ],
            "Resource":"arn:aws:iam::123456789012:role/DevTeam*"
        }
    ]
}
```

   `ec2:*` 为用户授予调用所有 Amazon EC2 API 操作的权限。要将用户限制到特定 Amazon EC2 API 操作，请改为指定这些操作。

   用户必须具有相应权限，可以调用 `iam:ListRoles` 操作以枚举现有 IAM 角色、调用 `iam:PassRole` 操作以指定 EC2 实例集角色以及调用 `iam:ListInstanceProfiles` 操作以枚举现有实例配置文件。

   （可选）要允许用户使用 IAM 控制台创建角色或实例配置文件，还必须在策略中添加以下操作：

   - `iam:AddRoleToInstanceProfile`
   - `iam:AttachRolePolicy`
   - `iam:CreateInstanceProfile`

- `iam:CreateRole`
- `iam:GetRole`
- `iam:ListPolicies`

5. 在查看策略页面上，输入策略名称和描述，然后选择创建策略。
6. 要提供访问权限，请为您的用户、组或角色添加权限：

   - 通过身份提供者在 IAM 中托管的用户：

     创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。
   - IAM 用户：
     - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
     - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

创建 EC2 实例集

要创建 EC2 实例集，请在 JSON 文件中定义实例集配置，然后使用 create-fleet 命令引用该文件。在 JSON 文件中，您必须指定实例集的总目标容量、竞价型实例和按需型实例的单独目标容量，以及用于定义实例集中实例配置（例如 AMI、实例类型、子网、可用区、一个或多个安全组）的启动模板。您可以选择指定其他配置，例如用于覆盖启动模板配置的参数、从 EC2 容量池中选择竞价型实例和按需型实例的分配策略，以及愿意为实例集支付的最高金额。有关更多信息，请参阅 EC2 实例集或竞价型实例集配置选项。

EC2 实例集在有可用容量时启动按需型实例，在最高价超过竞价型实例价格并且有可用容量时启动竞价型实例。

如果实例集包含竞价型实例且类型为 `maintain`，Amazon EC2 会尝试在竞价型实例中断时维持实例集的目标容量。

EC2 实例集限制

以下限制适用于 EC2 实例集。

- EC2 实例集仅可通过 Amazon EC2 API、Amazon CLI、Amazon SDK 和 Amazon CloudFormation 创建。

- EC2 实例集请求不能跨 Amazon 区域。您需要为每个区域创建单独的 EC2 实例集。
- EC2 实例集请求不能跨同一可用区内的不同子网。

创建 EC2 实例集

要使用 EC2 实例集启动实例的实例集，只需在实例集请求中指定以下参数，实例集会使用其他参数的默认值：

- `LaunchTemplateId` 或 `LaunchTemplateName` – 指定要使用的启动模板（其中包含要启动的实例的参数，例如实例类型和可用区）
- `TotalTargetCapacity` – 指定机群的总目标容量
- `DefaultTargetCapacityType` – 指定默认购买选项是按需型还是竞价型

要覆盖启动模板中指定的参数，可指定一个或多个覆盖。每个覆盖可以有不同的实例类型、可用区、子网和最高价，并且可以包含不同的权重容量。

对于 `instant` 类型的 EC2 实例集，可以指定 Systems Manager 参数而非 AMI ID。您可以在覆盖或启动模板中指定 Systems Manager 参数。有关更多信息，请参阅 使用 Systems Manager 参数而非 AMI ID。

在 JSON 文件中指定实例集参数。有关可以指定的所有可能参数的信息，请参阅查看所有的 EC2 实例集配置选项。

有关实例集配置示例，请参阅 EC2 实例集 CLI 配置示例。

目前没有控制台支持创建 EC2 Fleet。

创建 EC2 实例集

使用 create-fleet 命令创建实例集，并指定包含实例集配置参数的 JSON 文件。

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

以下是 `request` 或 `maintain` 类型的实例集的示例输出。

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

以下是启动了目标容量的 instant 类型实例集的示例输出。

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ],
      "InstanceType": "c5.large",
      "Platform": null
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-5678901234abcdef0",
        "i-5432109876abcdef9"
      ]
  ]
}
```

以下是启动了部分目标容量并且出现“无法启动实例”错误的 instant 类型实例集的示例输出。

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientInstanceCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a"
        }
      },
      "Lifecycle": "on-demand",
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef9"
      ]
  ]
}
```

以下是未启动任何实例的 instant 类型实例集的示例输出。

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": ""
    },
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a",
        }
      },
      "Lifecycle": "on-demand",
      "ErrorCode": "InsufficientCapacity",
      "ErrorMessage": ""
    },
  ],
  "Instances": []
}
```


创建 EC2 实例集来替换运行状况不佳的竞价型实例

EC2 实例集每 2 分钟检查一次实例集中实例的运行状况。实例的运行状况为 `healthy` 或 `unhealthy`。

EC2 实例集将使用 Amazon EC2 提供的状态检查来确定实例的运行状况。如果在连续三次运行状况检查中，实例状态检查或系统状态检查的状态有任一项为 unhealthy，则确定该实例的运行状况为 impaired。有关更多信息，请参阅Amazon EC2 实例的状态检查。

您可以配置您的实例集以替换运行状况不佳的竞价型实例。将 ReplaceUnhealthyInstances 设置为 true 以后，若竞价型实例报告为 unhealthy 则将其替换。在替换运行状况不佳的竞价型实例时，实例集可能在几分钟内降至其目标容量之下。

要求

- 仅对保持目标容量的 EC2 实例集（类型 maintain 的实例集）支持运行状况检查替换，不支持类型为 request 或 instant 的实例集。
- 仅对竞价型实例支持运行状况检查替换。对于 按需型实例 不支持此功能。
- 您可以将 EC2 实例集配置为仅在您创建它时替换运行状况不佳的实例。
- 用户仅在其有权调用 ec2:DescribeInstanceStatus 操作时才能使用运行状况检查替换。

配置 EC2 实例集 以替换运行状况不佳的竞价型实例

1. 使用创建 EC2 实例集中用于创建 EC2 实例集的信息。
2. 要将实例集配置为替换运行状况不佳的竞价型实例，请在 JSON 文件中将 ReplaceUnhealthyInstances 指定 true。

查看所有的 EC2 实例集配置选项

要查看 EC2 机群配置参数的完整列表，可以生成 JSON 文件。有关每个参数的描述，请参阅 create-fleet。

通过所有可能的 EC2 实例集参数生成 JSON 文件

使用 create-fleet (Amazon CLI) 命令和 --generate-cli-skeleton 参数生成 EC2 机群 JSON 文件，并将输出定向到某个文件以将其保存。

```
aws ec2 create-fleet \
    --generate-cli-skeleton input > ec2createfleet.json
```

下面是示例输出。

```
{
    "DryRun": true,
```

```
    "ClientToken": "",
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch"
            }
        },
        "InstanceInterruptionBehavior": "hibernate",
        "InstancePoolsToUseCount": 0,
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        },
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MinTargetCapacity": 0,
        "MaxTotalPrice": ""
    },
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "",
                "LaunchTemplateName": "",
                "Version": ""
            },
            "Overrides": [
                {
                    "InstanceType": "r5.metal",
                    "MaxPrice": "",
                    "SubnetId": "",
                    "AvailabilityZone": "",
                    "WeightedCapacity": 0.0,
                    "Priority": 0.0,
                    "Placement": {
                        "AvailabilityZone": "",
                        "Affinity": "",
```

```
                    "GroupName": "",
                    "PartitionNumber": 0,
                    "HostId": "",
                    "Tenancy": "dedicated",
                    "SpreadDomain": "",
                    "HostResourceGroupArn": ""
                },
                "InstanceRequirements": {
                    "VCpuCount": {
                        "Min": 0,
                        "Max": 0
                    },
                    "MemoryMiB": {
                        "Min": 0,
                        "Max": 0
                    },
                    "CpuManufacturers": [
                        "amd"
                    ],
                    "MemoryGiBPerVCpu": {
                        "Min": 0.0,
                        "Max": 0.0
                    },
                    "ExcludedInstanceTypes": [
                        ""
                    ],
                    "InstanceGenerations": [
                        "previous"
                    ],
                    "SpotMaxPricePercentageOverLowestPrice": 0,
                    "OnDemandMaxPricePercentageOverLowestPrice": 0,
                    "BareMetal": "included",
                    "BurstablePerformance": "required",
                    "RequireHibernateSupport": true,
                    "NetworkInterfaceCount": {
                        "Min": 0,
                        "Max": 0
                    },
                    "LocalStorage": "excluded",
                    "LocalStorageTypes": [
                        "ssd"
                    ],
                    "TotalLocalStorageGB": {
                        "Min": 0.0,
```

```
                        "Max": 0.0
                    },
                    "BaselineEbsBandwidthMbps": {
                        "Min": 0,
                        "Max": 0
                    },
                    "AcceleratorTypes": [
                        "inference"
                    ],
                    "AcceleratorCount": {
                        "Min": 0,
                        "Max": 0
                    },
                    "AcceleratorManufacturers": [
                        "amd"
                    ],
                    "AcceleratorNames": [
                        "a100"
                    ],
                    "AcceleratorTotalMemoryMiB": {
                        "Min": 0,
                        "Max": 0
                    }
                }
            }
        ]
    }
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 0,
    "OnDemandTargetCapacity": 0,
    "SpotTargetCapacity": 0,
    "DefaultTargetCapacityType": "on-demand",
    "TargetCapacityUnitType": "memory-mib"
},
"TerminateInstancesWithExpiration": true,
"Type": "instant",
"ValidFrom": "1970-01-01T00:00:00",
"ValidUntil": "1970-01-01T00:00:00",
"ReplaceUnhealthyInstances": true,
"TagSpecifications": [
    {
        "ResourceType": "fleet",
        "Tags": [
```

```
                {
                    "Key": "",
                    "Value": ""
                }
            ]
        }
    ],
    "Context": ""
}
```

标记新的或现有的 EC2 实例集请求及其启动的实例和卷

要对 EC2 实例集请求及其启动的实例和卷进行分类与管理，可以使用自定义元数据标记它们。您可以在创建 EC2 实例集请求之时或之后为其分配标签。同理，可以在实例集启动实例和卷之时或之后为实例和卷分配标签。

在标记实例集请求时，不会自动标记实例集启动的实例和卷。您需要明确标记实例集启动的实例和卷。您可以选择仅将标签分配给实例集请求，仅分配给实例集启动的实例，仅分配给附加到实例集启动的实例的卷，或者分配给所有这三种内容。

> **Note**
>
> 对于 `instant` 实例集类型，您可以标记附加到按需型实例和竞价型实例的卷。对于 `request` 或 `maintain` 实例集类型，您只能标记附加到按需型实例的卷。

有关标签的工作原理的更多信息，请参阅标记 Amazon EC2 资源。

先决条件

授予用户标记资源的权限。有关更多信息，请参阅 示例：标记资源。

授予用户标记资源的权限

创建包含以下内容的 IAM policy：

- `ec2:CreateTags` 操作。这将授予用户创建标签的权限。
- `ec2:CreateFleet` 操作。这将授予用户创建 EC2 实例集请求的权限。
- 对于 `Resource`，我们建议您指定 `"*"`。这允许用户标记所有资源类型。

要提供访问权限，请为您的用户、组或角色添加权限：

- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。
- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

标记新的 EC2 实例集请求

要在创建时标记 EC2 实例集请求，请在用于创建该实例集的 JSON 文件中指定键值对。`ResourceType` 的值必须为 `fleet`。如果指定其他值，实例集请求失败。

标记 EC2 实例集启动的实例和卷

要在实例集启动实例和卷时标记它们，请在 EC2 实例集请求中引用的启动模板中指定标签。

> **Note**
>
> 您无法标记附加到 `request` 或 `maintain` 实例集类型启动的竞价型实例的卷。

标记现有的 EC2 实例集请求、实例和卷

使用 create-tags 命令标记现有的资源。

```
aws ec2 create-tags \
    --resources fleet-12a34b55-67cd-8ef9-
ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE \
    --tags Key=purpose,Value=test
```

描述 EC2 实例集、其实例及其事件

您可以描述 EC2 机群配置、EC2 机群中的实例，以及 EC2 机群的历史事件。

任务

- 描述 EC2 机群
- 描述 EC2 实例集中的所有实例

- 描述 EC2 实例集的事件历史记录

描述 EC2 机群

Amazon CLI

描述 EC2 实例集

使用 describe-fleets 命令。

```
aws ec2 describe-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

下面是示例输出。

```
{
    "Fleets": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2022-02-09T03:35:52+00:00",
            "FleetId": "fleet-364457cd-3a7a-4ed9-83d0-7b63e51bb1b7",
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "FulfilledCapacity": 2.0,
            "FulfilledOnDemandCapacity": 0.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "LaunchTemplateName": "my-launch-template",
                        "Version": "$Latest"
                    }
                }
            ],
            "TargetCapacitySpecification": {
                "TotalTargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "TerminateInstancesWithExpiration": false,
            "Type": "maintain",
            "ReplaceUnhealthyInstances": false,
```

```
            "SpotOptions": {
                "AllocationStrategy": "capacity-optimized",
                "InstanceInterruptionBehavior": "terminate"
            },
            "OnDemandOptions": {
                "AllocationStrategy": "lowestPrice"
            }
        }
    ]
}
```

PowerShell

描述 EC2 实例集

使用 Get-EC2FleetList cmdlet。

```
Get-EC2FleetList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

描述 EC2 实例集中的所有实例

正在运行的实例的返回列表将定期刷新，或可能过时。

Amazon CLI

描述指定 EC2 实例集中的实例

使用 describe-fleet-instances 命令。

```
aws ec2 describe-fleet-instances \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

下面是示例输出。

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-09cd595998cb3765e",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-86k84j6p"
```

```
        },
        {
            "InstanceId": "i-09cf95167ca219f17",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-dvxi7fsm"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

PowerShell

描述指定 EC2 实例集中的实例

使用 Get-EC2FleetInstanceList cmdlet。

```
Get-EC2FleetInstanceList `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

描述 EC2 实例集的事件历史记录

有关事件历史记录中事件的更多信息，请参阅EC2 队列 事件类型。

Amazon CLI

描述指定 EC2 实例集的事件

使用 describe-fleet-history 命令。

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2020-06-01T00:00:00Z
```

下面是示例输出。

```
{
    "HistoryRecords": [
        {
            "EventInformation": {
                "EventSubType": "submitted"
            },
```

```
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:05.000Z"
        },
        {
            "EventInformation": {
                "EventSubType": "active"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:15.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
                "EventSubType": "progress"
            },
            "EventType": "fleetRequestChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-083a1c446e66085d2"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        },
        {
            "EventInformation": {
                "EventDescription": "{\"instanceType\":\"t2.small\", ...}",
                "EventSubType": "launched",
                "InstanceId": "i-090db02406cc3c2d6"
            },
            "EventType": "instanceChange",
            "Timestamp": "2020-09-01T18:26:17.000Z"
        }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2020-06-01T00:00:00.000Z"
}
```

PowerShell

描述指定 EC2 实例集的事件

使用 Get-EC2FleetHistory cmdlet。

```
Get-EC2FleetHistory `
    -FleetId fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE `
    -UtcStartTime 2020-06-01T00:00:00Z
```

修改 EC2 实例集

您可以修改 EC2 实例集的总目标容量、竞价型容量和按需型容量。您还可以修改，如果新的总目标容量减少到实例集当前大小以下，是否应终止正在运行的实例。

注意事项

修改 EC2 实例集时请考虑以下几点：

- 实例集类型 – 您只能修改 `maintain` 类型的 EC2 实例集，不能修改 `request` 或 `instant` 类型的 EC2 实例集。
- 实例集参数 – 您可以修改 EC2 实例集的以下参数：
  - `target-capacity-specification` – 增加或减少以下目标容量：
    - `TotalTargetCapacity`
    - `OnDemandTargetCapacity`
    - `SpotTargetCapacity`
  - `excess-capacity-termination-policy`：当 EC2 实例集的总目标容量降到实例集的当前大小以下时是否应终止正在运行的实例。有效值为：
    - `no-termination`
    - `termination`
- 增加总目标容量时的实例集行为 – 在增加总目标容量时，EC2 实例集会根据为 `DefaultTargetCapacityType` 指定的实例购买选项（按需型实例或竞价型实例）和指定的分配策略，启动额外的实例。
- 减少竞价型目标容量时的实例集行为 – 在减少竞价型目标容量时，EC2 实例集会删除超过新目标容量的任何打开的请求。您可以请求实例集终止竞价型实例，直到实例集的大小达到新目标容量。当 EC2 实例集因目标容量下降而终止某个竞价型实例时，该实例将收到一条竞价型实例中断通知。

根据分配策略选择要终止的实例：

- `capacity-optimized`：终止可用容量最少的池中的实例。
- `price-capacity-optimized`：使用价格和可用容量的组合：终止可用容量最少且价格最高的池中的实例。
- `diversified`：终止所有池中的实例。
- `lowest-price`：终止价格最高的池中的实例。

您也可以请求 EC2 实例集保持实例集的当前大小，而不替换任何已中断的竞价型实例或者手动终止的实例。

- 实例集状态 – 您可以修改处于 `submitted` 或 `active` 状态的 EC2 实例集。当您修改实例集时，它会进入 `modifying` 状态。

修改 EC2 实例集的命令

Amazon CLI

修改 EC2 实例集的总目标容量

使用 modify-fleet 命令。

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=20
```

如果要减少目标容量，但希望保持实例集的当前大小，可按如下方式修改上一个示例。

```
aws ec2 modify-fleet \
    --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity-specification TotalTargetCapacity=10 \
    --excess-capacity-termination-policy no-termination
```

PowerShell

修改 EC2 实例集的总目标容量

使用 Edit-EC2Fleet cmdlet。

```
Edit-EC2Fleet `
```

```
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20
```

如果要减少目标容量，但希望保持实例集的当前大小，可按如下方式修改上一个示例。

```
Edit-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacitySpecification_TotalTargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

删除 EC2 实例集请求和实例集中的实例

如果不再需要某个 EC2 实例集请求，可以将其删除。删除实例集请求后，将取消与实例集关联的所有竞价型实例请求，避免启动任何新的竞价型实例。

删除 EC2 实例集请求后，您还必须指定是否要终止其所有实例。包括按需型实例和竞价型实例。对于 `instant` 实例集，EC2 实例集必须在删除实例集时终止实例。不支持包含正在运行实例的已删除 `instant` 实例集。

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

如果您指定在实例集请求删除后必须终止实例，则实例集请求会进入 `deleted_terminating` 状态。否则，实例集会进入 `deleted_running` 状态，并且实例会继续运行，直到遇到中断或您手动将其终止。

限制

- 在单次操作中，最多可以删除 25 个 `instant` 类型的实例集。
- 在单次操作中，最多可以删除 100 个 `maintain` 或 `request` 类型的实例集。
- 在单次操作中，最多可以删除 125 个实例集，前提是不超过上述指定的各类实例集的限额。
- 如果要删除的实例集超过指定数量，将不会删除任何实例集。

- 不支持包含正在运行实例的已删除 instant 实例集。当您删除 instant 实例集时，Amazon EC2 会自动终止其所有实例。对于具有超过 1000 个实例的 instant 实例集，删除请求可能会失败。如果您的实例集有超过 1000 个实例，请先手动终止大多数实例，留下 1000 个或更少的实例。然后删除实例集，剩余的实例将自动终止。

Amazon CLI

删除 EC2 实例集请求并终止其实例

使用带 --terminate-instances 选项的 delete-fleets 命令。

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

下面是示例输出。

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
        }
    ]
}
```

删除 EC2 实例集请求但不终止其实例

通过改用 --no-terminate-instances 选项修改上一个示例。请注意，instant 实例集不支持 --no-terminate-instances。

```
aws ec2 delete-fleets \
    --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

下面是示例输出。

```
{
```

```
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"
        }
    ]
}
```

PowerShell

删除 EC2 实例集请求并终止其实例

使用带 `-TerminateInstance` 参数的 Remove-EC2Fleet cmdlet。

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

删除 EC2 实例集请求但不终止其实例

通过更改 `-TerminateInstance` 参数的值修改上一个示例。

```
Remove-EC2Fleet `
    -FleetId "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

实例集无法删除时进行故障排除

如果 EC2 实例集请求无法删除，则输出中的 `UnsuccessfulFleetDeletions` 将返回 EC2 实例集请求的 ID、错误代码和错误消息。

错误代码为：

- `ExceededInstantFleetNumForDeletion`
- `fleetIdDoesNotExist`
- `fleetIdMalformed`
- `fleetNotInDeletableState`
- `NoTerminateInstancesNotSupported`

- UnauthorizedOperation
- unexpectedError

ExceededInstantFleetNumForDeletion 问题排查

如果您尝试在单个请求中删除超过 25 个 instant 实例集，则会返回 ExceededInstantFleetNumForDeletion 错误。下面是此错误的示例输出。

```
{
    "UnsuccessfulFleetDeletions": [
     {
          "FleetId": " fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
          "Error": {
                  "Message": "Can’t delete more than 25 instant fleets in a single
 request.",
                  "Code": "ExceededInstantFleetNumForDeletion"
           }
     },
     {
           "FleetId": "fleet-9a941b23-0286-5bf4-2430-03a029a07e31",
           "Error": {
                  "Message": "Can’t delete more than 25 instant fleets in a single
 request.",
                  "Code": "ExceededInstantFleetNumForDeletion"
            }
     }
     .
     .
     .
     ],
     "SuccessfulFleetDeletions": []
}
```

NoTerminateInstancesNotSupported 问题排查

如果您指定在删除 instant 实例集时不得终止该实例集中的实例，则会返回 NoTerminateInstancesNotSupported 错误。--no-terminate-instances 实例集不支持 instant。下面是此错误的示例输出。

```
{
      "UnsuccessfulFleetDeletions": [
```

```
            {
                  "FleetId": "fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
                  "Error": {
                          "Message": "NoTerminateInstances option is not supported for
 instant fleet",
                          "Code": "NoTerminateInstancesNotSupported"
                   }
            }
      ],
      "SuccessfulFleetDeletions": []
}
```

`UnauthorizedOperation` 问题排查

如果您没有终止实例的权限，则在删除必须终止其实例的实例集时会收到 `UnauthorizedOperation` 错误。下面是错误响应。

```
<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not
 authorized to perform this
operation. Encoded authorization failure message: VvuncIxj7Z_CPGNYXWqnuFV-
YjByeAU66Q9752NtQ-I3-qnDLWs6JLFd
KnSMMiq5s6cGqjjPtEDpsnGHzzyHasFHOaRYJpaDVravoW25azn6KNkUQQlFwhJyujt2dtNCdduJfrqcFYAjlEiRMkfDHt7
BHturzDK6A560Y2nDSUiMmAB1y9UNtqaZJ9SNe5sNxKMqZaqKtjRbk02RZu5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtJ76
VPiU5v2s-
UgZ7h0p2yth6ysUdhlONg6dBYu8_y_HtEI54invCj4CoK0qawqzMNe6rcmCQHvtCxtXsbkgyaEbcwmrm2m01-
EMhekLFZeJLr
DtYOpYcEl4_nWFX1wtQDCnNNCmxnJZAoJvb3VMDYpDTsxjQv1PxODZuqWHs23YXWVywzgnLtHeRf2o4lUhGBw17mXsS07k7
PT9vrHtQiILor5VVTsjSPWg7edj__1rsnXhwPSu8gI48ZLRGrPQqFq0RmKO_QIE8N8s6NWzCK4yoX-9gDcheurOGpkprPIC
</Message></Error></Errors><RequestID>89b1215c-7814-40ae-a8db-41761f43f2b0</
RequestID></Response>
```

为了解决错误，您必须将 `ec2:TerminateInstances` 操作添加到 IAM policy，如下例中所示。

使用竞价型实例集

> **Important**
>
> 强烈反对使用竞价型实例集，因为其使用的是旧版 API，且暂无相关投资计划。如果想要管理实例生命周期，请改用 EC2 实例集。如果不想管理实例生命周期，请改用自动扩缩组。仅在使用 EC2 实例集的使用案例需要控制台支持时，才使用竞价型实例集。有关更多信息，请参阅哪种是实例集最佳使用方法？。

要开始使用竞价型实例集，请创建一个请求，其中包含竞价型实例总目标容量、一个可选的按需部分，然后手动指定 AMI 和密钥对，或者指定包含实例集中实例配置的启动模板。您可以选择指定其他参数，也可以让实例集使用默认值。您还可以在创建实例集时标记实例集请求及其实例和卷。

实例集在有可用容量时启动按需型实例，在最高价超过 Spot 价格并且有可用容量时启动竞价型实例。

实例集启动后，即可描述实例集请求、实例集中的实例以及任何实例集事件。您也可以根据需要分配额外的标签。

如需更改任何实例集参数（例如总目标容量），则可以修改该实例集，前提是该实例集已配置为保持容量。提交一次性请求后，您无法修改其容量。

实例集请求在过期或您删除（删除）它之前一直有效。取消实例集请求时，您可以终止实例或让其保持运行状态。如果选择让实例保持运行，按需型实例会持续运行至您将其终止，竞价型实例会持续运行至遇到中断或您将其终止。

主题

- 竞价型实例集请求状态
- 竞价型实例集权限
- 创建竞价型实例集
- 标记新的或现有的竞价型实例集请求及其启动的实例和卷
- 描述竞价型实例集请求、其实例及事件历史记录
- 修改竞价型实例集请求
- 取消（删除）竞价型实例集请求
- 了解竞价型实例集的自动扩展

竞价型实例集请求状态

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

竞价型实例集请求可以处于多种状态之一，每种状态表示请求生命周期的不同阶段及其实例管理。

竞价型实例集请求可以处于以下某种状态：

`submitted`

正在评估竞价型实例集请求，并且 Amazon EC2 正准备启动目标数量的实例。如果请求超出竞价型实例集限额，则会被立即取消。

`active`

已验证竞价型实例集，并且 Amazon EC2 正在尝试维护目标数量的正在运行的竞价型实例。请求会保持此状态，直到修改或取消它。

`modifying`

竞价型实例集请求正在修改中。请求将保持该状态，直到完全处理修改或取消该请求。只能修改 `maintain` 实例集类型。此状态不适用于一次性 `request` 实例集类型。

`cancelled_running`

已取消（删除）竞价型实例集且不启动其他竞价型实例。其现有实例将继续运行，直至中断或被手动终止。请求会保持此状态，直到所有实例都已中断或终止。

`cancelled_terminating`

已取消（删除）竞价型实例集，正在终止其实例。请求会保持此状态，直到所有实例都已终止。

`cancelled`

已取消（删除）竞价型实例集，没有正在运行的实例。请求将在其实例终止两天后被删除。

竞价型实例集权限

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

如果用户打算创建或管理竞价型实例集，您需要为其授予所需权限。

如果您使用 Amazon EC2 控制台创建竞价型实例集，将创建两个名为 `AWSServiceRoleForEC2SpotFleet` 和 `AWSServiceRoleForEC2Spot` 的服务相关角色和一个名为 `aws-ec2-spot-fleet-tagging-role` 的角色，这些角色为竞价型实例集授予代表您请求、启动、终止和标记资源的权限。如果您使用 Amazon CLI 或 API，您必须确保这些角色存在。

按照以下说明授予所需的权限并创建角色。

权限和角色

- 向用户授予竞价型实例集权限
- 竞价型实例集的服务相关角色
- 竞价型实例的服务相关角色
- 用于标记竞价型实例集的 IAM 角色

向用户授予竞价型实例集权限

如果用户打算创建或管理竞价型实例集，请确保为其授予所需的权限。

为竞价型实例集创建策略

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在导航窗格中，选择 Policies、Create policy。
3. 在 Create policy (创建策略) 页面上，选择 JSON，然后将文本替换为以下内容。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:CreateTags",
                "ec2:RequestSpotFleet",
                "ec2:ModifySpotFleetRequest",
                "ec2:CancelSpotFleetRequests",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequestHistory"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/aws-ec2-spot-fleet-tagging-role"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "iam:ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

上述示例策略为用户授予多数竞价型实例集使用案例所需的权限。要将用户限制为特定的 API 操作，请仅指定这些 API 操作。

所需的 EC2 和 IAM API

必须在策略中包含以下 API：

- `ec2:RunInstances` – 需要其启动竞价型实例集中的实例
- `ec2:CreateTags` – 需要其标记竞价型实例集请求、实例或卷
- `iam:PassRole` – 需要其指定竞价型实例集角色
- `iam:CreateServiceLinkedRole` – 需要其创建服务相关角色
- `iam:ListRoles` – 需要其枚举现有的 IAM 角色
- `iam:ListInstanceProfiles` – 需要其枚举现有的实例配置文件

> **Important**
>
> 如果您在启动规范或启动模板中为 IAM 实例配置文件指定角色，则必须授予用户将该角色传递给服务的权限。为此，在 IAM policy 中，包括 "`arn:aws:iam::*:role/`*`IamInstanceProfile-role`*" 作为 `iam:PassRole` 操作的资源。有关更多信息，请参阅 IAM 用户指南中的向用户授予将角色传递给 Amazon 服务的权限。

竞价型实例集 API

根据需要，将以下竞价型实例集 API 操作添加到策略中：

- ec2:RequestSpotFleet
- ec2:ModifySpotFleetRequest
- ec2:CancelSpotFleetRequests
- ec2:DescribeSpotFleetRequests
- ec2:DescribeSpotFleetInstances
- ec2:DescribeSpotFleetRequestHistory

可选的 IAM API

（可选）要允许用户使用 IAM 控制台创建角色或实例配置文件，您必须在策略中添加以下操作：

- iam:AddRoleToInstanceProfile
- iam:AttachRolePolicy
- iam:CreateInstanceProfile
- iam:CreateRole
- iam:GetRole
- iam:ListPolicies

4. 选择查看策略。
5. 在查看策略页面上，输入策略名称和描述，然后选择创建策略。
6. 要提供访问权限，请为您的用户、组或角色添加权限：
   - 通过身份提供者在 IAM 中托管的用户：

     创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。
   - IAM 用户：
     - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
     - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

竞价型实例集的服务相关角色

Amazon EC2 使用服务相关角色获取代表您调用其他Amazon服务所需的权限。服务相关角色是一种独特类型的 IAM 角色，它与Amazon服务直接相关。服务相关角色提供了一种将权限委托给 Amazon 服务的安全方式，因为只有相关服务才能担任服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色。

Amazon EC2 使用名为 AWSServiceRoleForEC2SpotFleet 的服务相关角色代表您启动和管理实例。

> **Important**
>
> 如果在竞价型实例集中指定加密的 AMI 或加密的 Amazon EBS 快照，则必须为 AWSServiceRoleForEC2SpotFleet 角色授予使用该 CMK 的权限，以便 Amazon EC2 可以代表您启动实例。有关更多信息，请参阅授予对用于加密的 AMI 和 EBS 快照的 CMK 的访问权限。

AWSServiceRoleForEC2SpotFleet 授予的权限

AWSServiceRoleForEC2SpotFleet 角色授予竞价型实例集权限以代表您请求、启动、终止和标记实例。Amazon EC2 使用此服务相关角色来完成以下操作：

- `ec2:RequestSpotInstances`：请求竞价型实例
- `ec2:RunInstances` - 启动实例
- `ec2:TerminateInstances` - 终止实例
- `ec2:DescribeImages` – 描述实例的亚马逊机器映像（AMI）
- `ec2:DescribeInstanceStatus` - 监控实例的状态。
- `ec2:DescribeSubnets` - 描述实例的子网
- `ec2:CreateTags` – 在竞价型实例集请求、实例和卷中添加标签
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer` - 将指定的实例添加到指定的负载均衡器
- `elasticloadbalancing:RegisterTargets` - 向指定目标组注册指定目标

创建服务相关角色

在大多数情况下，无需手动创建服务相关角色。Amazon EC2 在您首次使用控制台创建竞价型实例集时创建 AWSServiceRoleForEC2SpotFleet 服务相关角色。

如果在 2017 年 10 月之前具有活动竞价型实例集请求（此时 Amazon EC2 开始支持该服务相关角色），则 Amazon EC2 在您的Amazon账户中创建 AWSServiceRoleForEC2SpotFleet 角色。有关更多信息，请参阅 IAM 用户指南中的我的Amazon账户中出现新角色。

如果您使用 Amazon CLI 或 API 来创建竞价型实例集，则必须确保此角色存在。

使用控制台为竞价型实例集创建 AWSServiceRoleForEC2SpotFleet 角色

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）。
3. 选择创建角色。
4. 在选择受信任的实体页面上，请执行以下操作：
   a. 在可信实体类型中选择 Amazon 服务。
   b. 在使用案例下，对于服务或使用案例，选择 EC2。
   c. 对于使用案例，选择 EC2 - 竞价型实例集。

   > **Note**
   >
   > EC2 - 竞价型实例集使用案例将自动创建具有所需 IAM 权限的策略，并建议将 AWSEC2SpotFleetServiceRolePolicy 作为角色名称。

   d. 选择下一步。
5. 在添加权限页面上，选择下一步。
6. 在命名、检查并创建页面上，选择创建角色。

使用 Amazon CLI 为竞价型实例集创建 AWSServiceRoleForEC2SpotFleet 角色

如下所示使用 create-service-linked-role 命令。

```
aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

如果您不再需要使用竞价型实例集，我们建议您删除 AWSServiceRoleForEC2SpotFleet 角色。从您的账户中删除该角色后，如果您使用控制台请求竞价型实例集，Amazon EC2 将再次创建该角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

授予对用于加密的 AMI 和 EBS 快照的 CMK 的访问权限

如果在竞价型实例集请求中指定加密的 AMI 或加密的 Amazon EBS 快照，并且您使用客户托管的密钥进行加密，则必须为 AWSServiceRoleForEC2SpotFleet 角色授予使用该 CMK 的权限，以便 Amazon EC2 可以代表您启动实例。为此，您必须在 CMK 中添加授权，如以下过程中所示。

在提供权限时，授权是密钥策略的替代方法。有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的使用授权和在 Amazon KMS 中使用密钥策略。

为 AWSServiceRoleForEC2SpotFleet 角色授予使用 CMK 的权限

- 使用 create-grant 命令在 CMK 中添加授权，并指定授予权限的主体（AWSServiceRoleForEC2SpotFleet 服务相关角色）以执行授权允许的操作。CMK 是由 `key-id` 参数和 CMK 的 ARN 指定的。主体是由 `grantee-principal` 参数和 AWSServiceRoleForEC2SpotFleet 服务相关角色的 ARN 指定的。

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
spotfleet.amazonaws.com/AWSServiceRoleForEC2SpotFleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
 "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
 "ReEncryptTo"
```

竞价型实例的服务相关角色

Amazon EC2 使用名为 AWSServiceRoleForEC2Spot 的服务相关角色代表您启动和管理竞价型实例。有关更多信息，请参阅竞价型实例请求的服务相关角色。

用于标记竞价型实例集的 IAM 角色

`aws-ec2-spot-fleet-tagging-role` IAM 角色授予竞价型实例集标记竞价型实例集请求、实例和卷的权限。有关更多信息，请参阅 标记新的或现有的竞价型实例集请求及其启动的实例和卷。

> **Important**
>
> 如果您选择在实例集中标记实例并选择保持目标容量（竞价型实例集请求属于类型 `maintain`），用户与 `IamFleetRole` 的权限差异可能会导致实例集中实例的标记行为不一

致。如果 IamFleetRole 不包含 CreateTags 权限，则可能不会标记由实例集启动的某些实例。虽然我们正在努力修复这种不一致性，但为了确保实例集启动的所有实例都被标记，我们建议您为 aws-ec2-spot-fleet-tagging-role 使用 IamFleetRole 角色。或者，要使用现有角色，请将 AmazonEC2SpotFleetTaggingRole Amazon 托管策略附加到现有角色。否则，您需要手动将 CreateTags 权限添加到现有策略。

创建用于标记竞价型实例集的 IAM 角色

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）。
3. 选择创建角色。
4. 在 Select trusted entity（选择可信实体）页面中，在 Trusted entity type（可信实体类型）下选择 service（Amazon 服务）。
5. 在使用案例下，从其他 Amazon 服务的使用案例中选择 EC2，然后选择 EC2 – 竞价型实例集标记。
6. 选择下一步。
7. 在添加权限页面上，选择下一步。
8. 在 Name, review, and create（命名、检查和创建）页面上，对于 Role name（角色名称），输入角色的名称（例如 **aws-ec2-spot-fleet-tagging-role**）。
9. 检查此页面上的信息，然后选择 Create role（创建角色）。

防止跨服务混淆座席

混淆代理问题是一个安全问题，即没有执行操作权限的实体可能会迫使更具权限的实体执行该操作。我们建议在 aws-ec2-spot-fleet-tagging-role 信任策略中使用 aws:SourceArn 和 aws:SourceAccount 全局条件上下文键，以限制竞价型实例集为其他服务提供的资源访问权限。

将 aws:SourceArn 和 aws:SourceAccount 条件键添加到 **aws-ec2-spot-fleet-tagging-role** 信任策略

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择角色。
3. 找到您之前创建的 aws-ec2-spot-fleet-tagging-role 并选择链接（而不是复选框）。

4. 在 Summary（摘要）下，选择 Trust relationships（信任关系）选项卡，然后选择 Edit trust policy（编辑信任策略）。
5. 在 JSON 语句中，添加一个包含您的 aws:SourceAccount 和 aws:SourceArn 全局条件上下文键的 Condition 元素，以防止混淆代理人问题，如下所示：

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/
sfr-*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
```

> **Note**
>
> 如果 aws:SourceArn 值包含账户 ID，并且您同时使用了这两个全局条件上下文键，则 aws:SourceAccount 值和 aws:SourceArn 值中的账户在同一策略语句中使用时，必须使用相同的账户 ID。

最终的信任策略将如下所示：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "spotfleet.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-
request/sfr-*"
      },
      "StringEquals": {
```

```
            "aws:SourceAccount": "111122223333"
        }
      }
    }
}
```

6. 选择更新策略。

下表提供了 aws:SourceArn 的潜在值，以根据不同的明确程度限制 aws-ec2-spot-fleet-tagging-role 的范围。

API 操作	调用的服务	范围	**aws:SourceArn**
RequestSpotFleet	Amazon STS (AssumeRole)	将 aws-ec2-spot-fleet-tagging-role 上的 AssumeRole 功能限定为指定账户中的竞价型实例集请求。	arn:aws:ec2:*: *123456789012* :spot-fleet-request/sfr-*
RequestSpotFleet	Amazon STS (AssumeRole)	将 aws-ec2-spot-fleet-tagging-role 上的 AssumeRole 功能限定为指定账户和指定区域中的竞价型实例集请求。请注意，此角色将不能在其他区域使用。	arn:aws:ec2: *us-east-1* :*123456789012* :spot-fleet-request/sfr-*
RequestSpotFleet	Amazon STS (AssumeRole)	将 aws-ec2-spot-fleet-tagging-role 上的 AssumeRole 功能限定为仅影响实例集 sfr-11111111-1111-1111-1111-1111	arn:aws:ec2: *us-east-1* :*123456789012* :spot-fleet-request/sfr-*11111111-1111-1111*

API 操作	调用的服务	范围	aws:SourceArn
		-111111111111 的操作。请注意，此角色可能无法用于其他竞价型实例集。此外，此角色不能用于通过请求竞价型实例集来启动任何新的竞价型实例集。	*-1111-1111111111*

创建竞价型实例集

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

使用 Amazon Web Services 管理控制台时，只需选择 AMI 和所需的目标总容量，即可快速创建竞价型实例集请求。Amazon EC2 将配置一个最符合您需求并遵循竞价型实例最佳实践的实例集。您也可以修改任何默认设置。

要在实例集中包含按需型实例，则必须在请求中指定启动模板以及所需的按需容量。

实例集在有可用容量时启动按需型实例，在最高价超过 Spot 价格并且有可用容量时启动竞价型实例。

如果实例集包含竞价型实例且类型为 `maintain`，Amazon EC2 会尝试在竞价型实例中断时维持实例集的目标容量。

所需的权限

有关更多信息，请参阅 the section called “竞价型实例集权限”。

任务

- 快速创建竞价型实例集请求
- 使用已定义的参数创建竞价型实例集请求
- 创建竞价型实例集来替换运行状况不佳的竞价型实例

快速创建竞价型实例集请求

使用 Amazon EC2 控制台，按照以下步骤快速创建竞价型实例集请求。

使用推荐设置创建竞价型实例集请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 如果您是首次接触 Spot，则会看到一个欢迎页面；请选择 Get started。否则，请选择创建竞价型实例集请求。
4. 在 Launch parameters（启动参数）下，选择 Manually configure launch parameters（手动配置启动参数）。
5. 对于 AMI，选择一个 AMI。
6. 在 Target capacity（目标容量）下，对于 Total target capacity（总目标容量），请指定要请求的单位数。对于单位类型，您可以选择实例、vCPU 或 内存（GiB）。
7. 对于您的实例集请求概览，查看您的实例集配置，然后选择启动。

使用已定义的参数创建竞价型实例集请求

您可以使用自己定义的参数创建竞价型实例集。

Console

使用已定义的参数创建竞价型实例集请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 如果您是首次接触 Spot，则会看到一个欢迎页面；请选择 Get started。否则，请选择创建竞价型实例集请求。
4. 对于启动参数，可以手动配置启动参数，也可以使用启动模板，如下所示：

   a. [手动配置] 要在 Amazon EC2 控制台中定义启动参数，请选择手动配置启动参数，再执行以下操作：

      i. 对于 AMI，选择 Amazon 提供的基本 AMI，或者选择 Search for AMI（搜索 AMI）以使用来自我们用户社群、Amazon Web Services Marketplace 或您自己的 AMI。

> **Note**
>
> 如果启动参数中指定的 AMI 已注销或禁用，则无法从该 AMI 启动任何新实例。对于设置为维持目标容量的实例集，目标容量将无法维持。

ii. （可选）对于 Key pair name (密钥对名称)，选择现有密钥对或新建一个密钥对。

[现有密钥对] 选择所需的密钥对。

[新密钥对] 选择创建新密钥对前往密钥对页面。完成之后，返回 Spot Requests（竞价型请求）页面并刷新列表。

iii. （可选）展开 Additional launch parameters（其它启动参数），然后执行以下操作：

A. （可选）要启用 Amazon EBS 优化，对于 EBS-optimized（EBS 优化），请选择 Launch EBS-optimized instances（启动 EBS 优化实例）。

B. （可选）要为实例添加临时性块级存储，请对 Instance store (实例存储) 选择 Attach at launch (启动时附加)。

C. （可选）要添加存储，请选择 Add new volume（添加新卷），然后根据实例类型指定其它实例存储卷或 Amazon EBS 卷。

D. (可选) 默认情况下，已为您的实例启用基本监控。要启用详细监控，对于 Monitoring（监控），请选择 Enable CloudWatch detailed monitoring（启用 CloudWatch 详细监控）。

E. （可选）要运行专用竞价型实例，请为租赁选择专用 – 运行专用实例。

F. （可选）对于 Security groups (安全组)，选择一个或多个安全组，或者新建一个。

[现有安全组] 选择一个或多个所需的安全组。

[新安全组] 选择 Create new security group（新建安全组）以前往 Security Groups（安全组）页面。完成之后，返回 Spot Requests（Spot 请求）并刷新列表。

G. （可选）要能够通过 Internet 访问实例，请对 Auto-assign IPv4 Public IP (自动分配 IPv4 公有 IP) 选择 Enable (启用)。

H. （可选）要使用 IAM 角色启动竞价型实例，请对 IAM 实例配置文件选择角色。

I. (可选) 要运行启动脚本，请将其复制到 User data。

J. （可选）要添加标签，请选择 Create tag（创建标签），输入标签的键和值，然后选择 Create（创建）。对每个标签重复此操作。

对于每个标签，要使用相同标签标记实例和竞价型实例集请求，请确保同时选择了 Instance（实例）和 Fleet（机群）。要仅标记由实例集启动的实例，请清除实例集。要仅标记竞价型实例集请求，请清除 Instance（实例）。

b. [启动模板] 要使用在启动模板中创建的配置，请选择使用启动模板，再为启动模板选择一个启动模板。

> **Note**
>
> 如果希望在竞价型实例集中包含按需容量，则必须指定启动模板。

5. 对于 Additional request details (其他请求详细信息)，执行以下操作：

a. 查看其他请求详细信息。要进行更改，请清除 Apply defaults (应用默认设置)。

b. （可选）对于 IAM 实例集角色，您可以使用默认角色或选择其他角色。要在更改角色后使用默认角色，请选择 Use default role (使用默认角色)。

c. (可选) 要创建仅在特定时间段内有效的请求，请编辑请求有效起始时间和请求有效截止时间。

d. （可选）默认情况下，Amazon EC2 在竞价型实例集会在请求过期时终止竞价型实例。要保持这些实例在请求过期后继续运行，请清除 Terminate the instances when the request expires (请求到期时终止实例)。

e. （可选）要向负载均衡器注册竞价型实例，请选择从一个或多个负载均衡器接收流量，然后选择一个或多个经典负载均衡器或目标组。

6. 对于 Target capacity（目标容量），执行以下操作：

a. 对于 Total target capacity（总目标容量），请指定要请求的单位数。对于单位类型，您可以选择 Instances（实例）、vCPU 或 Memory (MiB)（内存 (MiB)）。要将目标容量指定为 0 以便以后可增加容量，请选择维持目标容量。

b. （可选）对于 Include On-Demand base capacity（包括按需基本容量），请指定要请求的按需单位数。该数字必须小于 Total target capacity (总目标容量)。Amazon EC2 会计算差值，并将差值分配给要请求的 Spot 单位。

> **Important**
>
> 要指定可选的按需容量，您必须先选择一个启动模板。

c. （可选）默认情况下，在竞价型实例中断时，Amazon EC2 将终止这些实例。要保持目标容量，请选择保持目标容量。然后，您可以指定 Amazon EC2 在竞价型实例中断时终止、停止或休眠这些实例。为此，请从中断行为中选择相应的选项。

> **Note**
>
> 如果启动参数中指定的 AMI 已注销或禁用，则无法从该 AMI 启动任何新实例。在这种情况下，对于设置为维持目标容量的实例集，目标容量将无法维持。

d. （可选）要允许竞价型实例集在系统针对实例集中的现有竞价型实例发出实例再平衡通知时启动替换竞价型实例，请选择 Capacity rebalance（容量再平衡），然后选择实例替换策略。如果选择终止前启动，请指定 Amazon EC2 终止旧实例之前的延迟时间（以秒为单位）。有关更多信息，请参阅 在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例。

e. （可选）要控制每小时为实例集中所有的竞价型实例支付的金额，请选择设置 竞价型实例的最大成本，然后输入您每小时愿意支付的最大总额。当达到最大总额后，即使未达到目标容量，竞价型实例集也会停止启动竞价型实例。有关更多信息，请参阅 为 EC2 实例集或竞价型实例集设置支出限额。

7. 对于 Network（网络），执行以下操作：

a. 对于 Network (网络)，选择现有 VPC 或新建一个。

[现有 VPC] 选择所需的 VPC。

[新 VPC] 选择 Create new VPC (新建 VPC) 以前往 Amazon VPC 控制台。完成之后，请返回此屏幕并刷新列表。

b. （可选）对于可用区，让 Amazon EC2 为竞价型实例选择可用区，或者指定一个或多个可用区。

如果您在一个可用区中有多个子网，则请从 Subnet (子网) 中选择合适的子网。要添加子网，请选择 Create new subnet (新建子网) 以前往 Amazon VPC 控制台。完成之后，请返回此屏幕并刷新列表。

8. 对于 Instance type requirements（实例类型要求），可以指定实例属性，然后让 Amazon EC2 使用这些属性识别最佳实例类型，也可以指定实例列表。有关更多信息，请参阅 指定 EC2 实例集或竞价型实例集的实例类型选择属性。

   a. 如果选择 Specify instance attributes that match your compute requirements（指定符合计算要求的实例属性），请按如下方式指定实例属性：

      i. 对于 vCPU，请输入所需的最小和最大 vCPU 数。要指定没有限制，请选择没有最小值和/或没有最大值。

      ii. 对于 Memory (GiB)（内存 (GiB)），输入所需的最小和最大内存量。要指定没有限制，请选择没有最小值和/或没有最大值。

      iii. （可选）对于其他实例属性，您可以选择指定一个或多个属性来更详细地表达计算要求。每个额外属性都会进一步增加对您的请求的限制。您可以省略其它属性；如果省略，则使用默认值。有关每个属性及其默认值的描述，请参阅 get-spot-placement-scores。

      iv. （可选）要查看具有指定属性的实例类型，请展开 Preview matching instance types（预览匹配的实例类型）。要排除在请求中使用的实例类型，请选择实例，然后选择 Exclude selected instance types（排除选定的实例类型）。

   b. 如果选择 Manually select instance types（手动选择实例类型），竞价型实例集将提供实例类型默认列表。要选择更多实例类型，请选择 Add instance types（添加实例类型），选择要在请求中使用的实例类型，然后选择 Select（选择）。要删除实例类型，请选择实例类型，然后选择 Delete（删除）。

9. 对于分配策略，请选择符合需求的竞价型分配策略和按需型分配策略。有关更多信息，请参阅 使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量。

10. 对于 Your fleet request at a glance（您的机群请求概览），查看您的机群配置，并在必要时进行任何调整。

11. （可选）要下载一个启动配置副本以用于 Amazon CLI，请选择 JSON config (JSON 配置)。

12. 准备好启动竞价型实例集后，请选择启动。

    竞价型实例集请求类型为 `fleet`。执行请求后，系统会添加请求类型 `instance`，此时其状态为 `active` 和 `fulfilled`。

Amazon CLI

创建竞价型实例集请求

使用 request-spot-fleet 命令。

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

有关示例配置文件，请参阅竞价型实例集 CLI 配置示例。

PowerShell

创建竞价型实例集请求

使用 Request-EC2SpotFleet cmdlet。以下示例在容量优化实例集中启动竞价型实例。

```
Request-EC2SpotFleet `
    -SpotFleetRequestConfig_TargetCapacity 50 `
    -SpotFleetRequestConfig_AllocationStrategy "CapacityOptimized" `
    -SpotFleetRequestConfig_IamFleetRole "arn:aws:iam::123456789012:role/my-spot-
fleet-role" `
    -SpotFleetRequestConfig_LaunchTemplateConfig @($launchConfig)
```

按如下方式定义启动配置，设置启动模板并覆盖所需的属性。有关示例配置，请参阅 竞价型实例集 CLI 配置示例。

```
$lcSpec = Amazon.EC2.Model.FleetLaunchTemplateSpecification
# To do - Set FleetLaunchTemplateSpecification properties
$lcOverrides = New-Object Amazon.EC2.Model.LaunchTemplateOverrides
# To do - Set LaunchTemplateOverrides properties
$launchConfig = New-Object Amazon.EC2.Model.LaunchTemplateConfig
$launchConfig.LaunchTemplateSpecification $lcSpec
$launchConfig.Overrides @($lcOverrides)
```

创建竞价型实例集来替换运行状况不佳的竞价型实例

竞价型实例集每 2 分钟检查一次实例集中竞价型实例的运行状况。实例的运行状况为 healthy 或 unhealthy。

竞价型实例集将使用 Amazon EC2 提供的状态检查来确定实例的运行状况。如果在连续三次运行状况检查中，实例状态检查或系统状态检查的状态有任一项为 unhealthy，则可确定该实例的运行状况为 impaired。有关更多信息，请参阅 Amazon EC2 实例的状态检查。

您可以配置您的实例集以替换运行状况不佳的竞价型实例。启用运行状况检查替换后，如果竞价型实例被报告为 unhealthy，则会被替换。在替换运行状况不佳的竞价型实例时，实例集的容量可能在几分钟内降至其目标容量之下。

要求

- 只有在保持目标容量（类型 maintain 的实例集）的情况下竞价型实例集才支持运行状况检查替换，而不支持一次性竞价型实例集（类型 request 的实例集）。
- 仅对竞价型实例支持运行状况检查替换。对于 按需型实例 不支持此功能。
- 您可以将竞价型实例集配置为仅在您创建它时替换运行状况不佳的实例。
- 用户仅在其有权调用 ec2:DescribeInstanceStatus 操作时才能使用运行状况检查替换。

Console

配置竞价型实例集来替换运行不正常的竞价型实例

1. 按使用已定义的参数创建竞价型实例集请求中创建竞价型实例集的步骤操作。
2. 要将实例集配置为替换运行状况不佳的竞价型实例，请展开其他启动参数，然后在运行状况检查下选择替换运行状况不佳的实例。要启用此选项，您必须先选择 Maintain target capacity (保持目标容量)。

Amazon CLI

配置竞价型实例集来替换运行不正常的竞价型实例

使用 ReplaceUnhealthyInstances 属性为 SpotFleetRequestConfig 的 request-spot-fleet 命令。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "TargetCapacity": 10,
        "ReplaceUnhealthyInstances": true
    }
}
```

PowerShell

配置竞价型实例集请求来替换运行不正常的竞价型实例

使用带 `-SpotFleetRequestConfig_ReplaceUnhealthyInstance` 参数的 Request-EC2SpotFleet cmdlet。

```
-SpotFleetRequestConfig_ReplaceUnhealthyInstance $true
```

标记新的或现有的竞价型实例集请求及其启动的实例和卷

Important

竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

要对竞价型实例集请求及其启动的实例和卷进行分类与管理，可以使用自定义元数据标记它们。您可以在创建竞价型实例集请求时或之后为其分配标签。同理，可以在实例集启动实例和卷之时或之后为实例和卷分配标签。

在标记实例集请求时，不会自动标记实例集启动的实例和卷。您需要明确标记实例集启动的实例和卷。您可以选择仅将标签分配给实例集请求，仅分配给实例集启动的实例，仅分配给附加到实例集启动的实例的卷，或者分配给所有这三种内容。

Note

您只能标记附加到按需型实例的卷。您无法标记附加到竞价型实例的卷。

您可以使用 Amazon EC2 控制台或命令行工具分配标签。

有关标签的工作原理的更多信息，请参阅标记 Amazon EC2 资源。

内容

- 先决条件
- 标记新的竞价型实例集及其启动的实例和卷
- 标记现有的竞价型实例集

- 查看竞价型实例集请求标签

先决条件

授予用户标记资源的权限。有关更多信息，请参阅 示例：标记资源。

授予用户标记资源的权限

创建包含以下内容的 IAM policy：

- `ec2:CreateTags` 操作。这将授予用户创建标签的权限。
- `ec2:RequestSpotFleet` 操作。这将授予用户创建竞价型实例集请求的权限。
- 对于 `Resource`，您必须指定 `"*"`。这允许用户标记所有资源类型。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotFleetRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:RequestSpotFleet"
            ],
            "Resource": "*"
        }
    ]
}
```

Important

对于 `spot-fleet-request` 资源，我们目前不支持资源级权限。如果您指定 `spot-fleet-request` 以作为资源，在您尝试标记实例集时，将会发生未经授权异常。以下示例说明如何不设置策略。

```
{
```

```
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags",
        "ec2:RequestSpotFleet"
    ],
    "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/*"
}
```

要提供访问权限，请为您的用户、组或角色添加权限：

- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。
- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

标记新的竞价型实例集及其启动的实例和卷

使用控制台标记新的竞价型实例集请求及其启动的实例和卷

1. 按照使用已定义的参数创建竞价型实例集请求过程操作。
2. 添加标签的方式取决于您是手动配置实例集还是使用启动模板。
   - 如果手动配置了实例集，请执行以下操作：

     要添加标签，请展开其他启动参数，选择创建标签，然后输入标签的键和值。对每个标签重复此操作。

     对于每个标签，您可以使用相同标签标记竞价型实例集请求和实例。要对两者进行标记，请确保同时选择实例和实例集。要仅标记竞价型实例集请求，请清除 Instance（实例）。要仅标记由实例集启动的实例，请清除实例集。

> **Note**
>
> 手动配置实例集时，无法选择标记卷。仅附加到按需型实例的卷支持卷标签。手动配置实例集时，您无法指定按需型实例。

- 如果使用启动模板，请执行以下操作：

  要向实例集请求添加标签，请在标签下，选择创建标签，然后输入该标签的键和值。对每个标签重复此操作。

  要标记实例集中的资源，您必须在启动模板中指定标签。

标记新的竞价型实例集请求及其使用 Amazon CLI 启动的实例和卷

要在创建竞价型实例集请求时标记该请求，并在实例集启动实例和卷时标记它们，请按以下方式配置竞价型实例集请求配置：

竞价型实例集请求标签：

- 在 `SpotFleetRequestConfig` 中指定竞价型实例集请求的标签。
- 对于 `ResourceType`，请指定 `spot-fleet-request`。如果指定其他值，实例集请求将失败。
- 对于 `Tags`，请指定键值对。您可以指定多个键值对。

实例标签：

- 为 `LaunchSpecifications` 中的实例指定标签。
- 对于 `ResourceType`，请指定 `instance`。如果指定其他值，实例集请求将失败。
- 对于 `Tags`，请指定键值对。您可以指定多个键值对。

  或者，您可以在竞价型实例集请求中引用的启动模板中为实例指定标签。

卷标签：

- 在竞价型实例集请求中引用的启动模板中为卷指定标签。不支持 `LaunchSpecifications` 中的卷标记。

在以下示例中，使用两个标签来标记竞价型实例集请求：键=环境，值=生产；键=成本中心，值=123。由实例集启动的实例使用一个标签（与竞价型实例集请求的其中一个标签相同）进行标记：键=成本中心，值=123。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1,
        "TagSpecifications": [
            {
                "ResourceType": "spot-fleet-request",
                "Tags": [
                    {
                        "Key": "Environment",
                        "Value":"Production"
                    },
                    {
```

```
                        "Key": "Cost-Center",
                        "Value":"123"
                    }
                ]
            }
        ]
    }
}
```

使用 Amazon CLI 标记由竞价型实例集启动的实例

要在实例集启动实例时标记实例，您可以在竞价型实例集请求中引用的启动模板中指定标记，也可以在竞价型实例集请求配置中指定标记，如下所示：

- 为 LaunchSpecifications 中的实例指定标签。
- 对于 ResourceType，请指定 instance。如果指定其他值，实例集请求将失败。
- 对于 Tags，请指定键值对。您可以指定多个键值对。

在以下示例中，由实例集启动的实例使用一个标签进行标记：键=成本中心，值=123。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
```

```
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1
    }
}
```

使用 Amazon CLI 标记挂载至由竞价型实例集启动的按需型实例的卷

要在实例集创建卷时标记这些卷，您必须在竞价型实例集请求中引用的启动模板中指定标签。

> **Note**
>
> 仅附加到按需型实例的卷支持卷标签。您无法标记附加到竞价型实例的卷。
> 不支持 `LaunchSpecifications` 中的卷标记。

标记现有的竞价型实例集

使用控制台标记现有竞价型实例集请求

创建竞价型实例集请求后，您可以使用控制台向实例集请求添加标签。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择 Tags (标签) 选项卡，然后选择 Create Tag (创建标签)。

使用 Amazon CLI 标记现有竞价型实例集请求

可以使用 create-tags 命令标记现有的资源。在以下示例中，使用以下标签标记现有竞价型实例集请求：键=目的，值=测试。

```
aws ec2 create-tags \
    --resources sfr-11112222-3333-4444-5555-66666EXAMPLE \
```

```
    --tags Key=purpose,Value=test
```

查看竞价型实例集请求标签

使用控制台查看竞价型实例集请求标签

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择您的竞价型实例集请求并选择标签选项卡。

描述竞价型实例集请求标签

使用 describe-tags 命令可查看指定资源的标签。在以下示例中，您将描述指定竞价型实例集请求的标签。

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
    "Tags": [
        {
            "Key": "Environment",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Production"
        },
        {
            "Key": "Another key",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Another value"
        }
    ]
}
```

您还可以通过描述竞价型实例集请求来查看竞价型实例集请求的标签。

使用 describe-spot-fleet-requests 命令可查看指定的竞价型实例集请求的配置，其中包括为实例集请求指定的任何标签。

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```


```
{
    "SpotFleetRequestConfigs": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2020-02-13T02:49:19.709Z",
            "SpotFleetRequestConfig": {
                "AllocationStrategy": "capacityOptimized",
                "OnDemandAllocationStrategy": "lowestPrice",
                "ExcessCapacityTerminationPolicy": "Default",
                "FulfilledCapacity": 2.0,
                "OnDemandFulfilledCapacity": 0.0,
                "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-
tagging-role",
                "LaunchSpecifications": [
                    {
                        "ImageId": "ami-0123456789EXAMPLE",
                        "InstanceType": "c4.large"
                    }
                ],
                "TargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "Type": "maintain",
                "ReplaceUnhealthyInstances": false,
                "InstanceInterruptionBehavior": "terminate"
            },
            "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "SpotFleetRequestState": "active",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                },
                {
                    "Key": "Another key",
                    "Value": "Another value"
                }
            ]
        }
    ]
```

```
}
```

描述竞价型实例集请求、其实例及事件历史记录

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

您可以描述竞价型实例集配置、竞价型实例集中的实例以及竞价型实例集的事件历史记录。

Console

描述竞价型实例集

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。ID 以 sfr- 开头。要查看配置详细信息，请选择 Description (描述)。
4. 要列出竞价型实例集的竞价型实例，请选择实例。
5. 要查看竞价型实例集的历史记录，请选择历史记录。

Amazon CLI

描述竞价型实例集请求

使用 describe-spot-fleet-requests 命令。

```
aws ec2 describe-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

描述用于指定竞价型实例集请求的运行实例

使用 describe-spot-fleet-instances 命令。

```
aws ec2 describe-spot-fleet-instances \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

描述用于指定竞价型实例集请求的事件历史记录

使用 describe-spot-fleet-request-history 命令。

```
aws ec2 describe-spot-fleet-request-history \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --start-time 2024-05-18T00:00:00Z
```

PowerShell

描述竞价型实例集请求

使用 Get-EC2SpotFleetRequest cmdlet。

```
Get-EC2SpotFleetRequest
```

描述用于指定竞价型实例集请求的运行实例

使用 Get-EC2SpotFleetInstance cmdlet。

```
Get-EC2SpotFleetInstance `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
```

描述用于指定竞价型实例集请求的事件历史记录

使用 Get-EC2SpotFleetRequestHistory cmdlet。

```
Get-EC2SpotFleetRequestHistory `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -UtcStartTime 2024-05-18T00:00:00Z
```

修改竞价型实例集请求

Important

竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

您可以修改活动的竞价型实例集请求以完成以下任务：

- 增加总目标容量和按需部分

- 减少总目标容量和按需部分

在增加总目标容量时，竞价型实例集会根据其竞价型实例集请求的分配策略启动额外的竞价型实例。当您提高按需部分时，竞价型实例集会启动其他按需型实例。

在减少总目标容量时，竞价型实例集会取消超过新目标容量的任何打开的请求。您可以请求竞价型实例集终止竞价型实例，直到实例集的大小达到新目标容量。如果分配策略为 `diversified`，则竞价型实例集将在池间终止实例。或者，您可以请求竞价型实例集保持实例集当前的实例集大小，而不替换已中断的任何竞价型实例或者您手动终止的任何竞价型实例。

注意事项

- 您无法修改一次性竞价型实例集请求。您只能在创建竞价型实例集请求时选择了保持目标容量的情况下修改竞价型实例集请求。
- 当竞价型实例集因目标容量下降而终止某个实例时，该实例将收到一条竞价型实例中断通知。

Console

修改竞价型实例集请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 依次选择 Actions (操作) 和 Modify target capacity (修改目标容量)。
5. 在 Modify target capacity 中，执行以下操作：
   a. 输入新的目标容量和按需部分。
   b. （可选）如果您要减少目标容量，但是要使实例集保持其当前大小，请清除终止实例。
   c. 选择提交。

Amazon CLI

修改竞价型实例集请求

使用 modify-spot-fleet-request 命令可更新指定竞价型实例集请求的目标容量。

```
aws ec2 modify-spot-fleet-request \
```

```
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity 20
```

可以按如下所示修改前面的命令，以减少指定竞价型实例集的目标容量而不因此终止任何竞价型实例。

```
aws ec2 modify-spot-fleet-request \
    --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --target-capacity 10 \
    --excess-capacity-termination-policy NoTermination
```

PowerShell

修改竞价型实例集请求

使用 Edit-EC2SpotFleetRequest cmdlet 更新指定竞价型实例集请求的目标容量。

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20
```

可以按如下所示修改前面的命令，以减少指定竞价型实例集的目标容量而不因此终止任何竞价型实例。

```
Edit-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TargetCapacity 20 `
    -ExcessCapacityTerminationPolicy "NoTermination"
```

取消（删除）竞价型实例集请求

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

如果您不再需要竞价型实例集，则可以取消竞价型实例集请求，这将删除该请求。取消实例集请求后，与实例集关联的所有竞价型实例请求也将取消，从而不会启动任何新的竞价型实例。

取消竞价型实例集请求后，您还必须指定是否要终止其所有实例。包括按需型实例和竞价型实例。

> **Warning**
>
> 终止实例是永久且不可逆转的。
> 终止一个实例后，您将无法再连接到该实例，而且也无法对其进行恢复操作。被配置为在终止后删除的所有连接的 Amazon EBS 卷也将被永久删除并且无法恢复。实例存储卷中存储的所有数据将永久丢失。有关更多信息，请参阅 实例终止的工作原理。
> 在终止实例之前，请务必确保已将需要在终止后保留的所有数据备份到持久存储中。

如果您指定必须在取消实例集请求后终止实例，实例集请求会进入 `cancelled_terminating` 状态。否则，实例集会进入 `cancelled_running` 状态，并且实例会继续运行，直到遇到中断或您手动将其终止。

限制

- 在单个请求中，最多可以取消 100 个实例集。如果超过指定数量，将不会取消任何实例集。

Console

取消（删除）竞价型实例集请求

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 依次选择操作和取消请求。
5. 在取消竞价型实例请求对话框中，执行以下操作：
   a. 要在取消竞价型实例集请求的同时终止关联的实例，请将终止实例复选框保持为选中状态。要取消竞价型实例集请求而不终止关联的实例，请清除终止实例复选框。
   b. 选择确认。

Amazon CLI

取消（删除）竞价型实例集请求并终止其实例

使用带 `--terminate-instances` 选项的 cancel-spot-fleet-requests 命令。

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

下面是示例输出。

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

取消（删除）竞价型实例集请求而不终止其实例

通过改用 `--no-terminate-instances` 选项修改上一个示例。

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

下面是示例输出。

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

PowerShell

取消（删除）竞价型实例集请求并终止其实例

使用带 -TerminateInstance 参数的 Stop-EC2SpotFleetRequest cmdlet。

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $true
```

取消（删除）竞价型实例集请求而不终止其实例

通过更改 -TerminateInstance 参数的值修改上一个示例。

```
Stop-EC2SpotFleetRequest `
    -SpotFleetRequestId "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE" `
    -TerminateInstance $false
```

了解竞价型实例集的自动扩展

> **Important**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

自动扩展使您的竞价型实例集能够根据需求增加或减少其目标容量。凭借自动扩展，竞价型实例集可以响应一个或多个扩展策略，在指定范围内启动实例（横向扩展）或终止实例（横向缩减）。

通过结合使用 Amazon EC2、Amazon CloudWatch 和 Application Auto Scaling API 可实现竞价型实例集的自动扩展。通过 Amazon EC2 可创建竞价型实例集请求，通过 CloudWatch 可创建警报，通过 Application Auto Scaling 可创建扩展策略。

自动扩缩的类型

竞价型实例集支持以下类型的自动扩展：

- 目标跟踪扩展：针对特定指标的值，增加或减少实例集的当前容量。这与恒温器保持家里温度的方式类似：您选择所需的温度，恒温器将完成所有其他工作。
- 步进扩展：根据一组扩展调整，增加或减小实例集的当前容量，这些调整称为步进调整，将根据警报严重程度发生变化。
- 计划扩展：根据日期和时间增加或减少实例集的当前容量。

注意事项

对竞价型实例集使用自动扩展时，请注意以下事项：

- 实例权重：如果使用实例权重，请记住，竞价型实例集可以根据需要超出目标容量。执行容量可以是浮点数，但目标容量必须是整数，因此竞价型实例集向上舍入到下一个整数。在您查看触发警报时扩展策略的结果时，必须考虑这些行为。例如，假设目标容量为 30，执行容量为 30.1，扩展策略减 1。当触发报警时，自动扩展过程将 30.1 减 1 得到 29.1，然后将其向上取整为 30，因此不执行扩展操作。再如，假设您选择的实例权重为 2、4 和 8，目标容量为 10，但没有权重 2 实例可用，因此为执行容量为 12 的竞价型实例集预置权重为 4 和 8 的实例。如果扩展策略将目标容量减少 20% 并触发警报，则自动扩展过程将 12 减 12*0.2 得到 9.6，然后将其向上取整为 10，因此不执行扩展操作。
- 冷却时间：您为竞价型实例集创建的扩展策略支持冷却时间。这是扩展活动完成后上一个与触发相关的扩展活动可影响将来扩展事件的秒数。对于横向扩展策略，虽然冷却时间有效，但启动冷却的上一个横向扩展事件所添加的容量将计算为下一次横向扩展所需容量的一部分。旨在持续（但不过度）横向扩展。对于横向缩减策略，冷却时间用于阻止后续横向缩减请求，直至到期。旨在谨慎地横向缩减以保护您的应用程序的可用性。但是，如果在横向缩减后，另一个警报在冷却时间内触发了横向扩展策略，自动扩展将立即横向扩展您的可扩展目标。
- 使用详细监控：建议将随实例指标扩展的频率设置为 1 分钟，这可确保更快地响应使用率变化。如果将随指标扩展的频率设置为 5 分钟，可能会导致响应时间变慢，并且可能导致系统依据陈旧的指标数据进行扩展。要每隔 1 分钟向 CloudWatch 发送一次实例的指标数据，您必须专门启用详细监控。有关更多信息，请参阅管理 EC2 实例的详细监控和使用已定义的参数创建竞价型实例集请求。
- Amazon CLI：如果要使用 Amazon CLI 来配置竞价型实例集的扩展，则应使用 application-autoscaling 命令。

竞价型实例集自动扩展所需的 IAM 权限

通过结合使用 Amazon EC2、Amazon CloudWatch 和 Application Auto Scaling API 可实现竞价型实例集的自动扩展。通过 Amazon EC2 可创建竞价型实例集请求，通过 CloudWatch 可创建警报，通过 Application Auto Scaling 可创建扩展策略。除使用竞价型实例集所需的 IAM 权限和 Amazon EC2 外，访问实例集扩展设置的用户必须具有使用支持自动扩缩的服务的适当权限。

要对竞价型实例集使用自动扩展，用户必须具有使用以下示例策略中所示操作的权限。

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ec2:DescribeSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "iam:CreateServiceLinkedRole",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

您还可以创建自己的 IAM 策略，从而使 Application Auto Scaling API 调用获得更精细的权限。有关更多信息，请参阅《Application Auto Scaling User Guide》中的 Identity and Access Management for Application Auto Scaling。

Application Auto Scaling 服务还需要描述竞价型实例集和 CloudWatch 警报的权限，以及代表您修改竞价型实例集目标容量的权限。如果您为竞价型实例集启用自动扩展功能，它将创建一个名为 `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest` 的服务相关角色。此服务相关角色授予 Application Auto Scaling 权限，以描述策略警报、监控实例集的当前容量以及修改实例集的容量。Application Auto Scaling 的原托管竞价型实例集角色为 `aws-ec2-spot-fleet-autoscale-role`，但今后已不再需要。此服务相关角色是 Application Auto Scaling 的默认角色。有关更多信息，请参阅《Application Auto Scaling 用户指南》中的 Application Auto Scaling 的服务相关角色。

目标跟踪扩展：通过确定特定指标的目标值来扩展竞价型实例集

凭借目标跟踪扩展，您可以通过选择指标并设置目标值来创建目标跟踪扩展策略。然后，竞价型实例集将创建和管理触发扩展策略的 CloudWatch 警报，并根据所选指标和目标值计算扩展调整。扩展策略通过根据需要添加或删除实例来调整容量，将指标保持在指定的目标值或接近指定的目标值。目标跟踪策略将指标保持在目标值附近，而且还可以根据由于负载模式波动而造成的指标波动进行调节，并最大限度地减少快速容量波动。

您可以为竞价型实例集创建多个目标跟踪扩展策略，但前提是每个策略分别使用不同的指标。实例集根据指定最大实例集容量的策略进行扩展。这样便可涵盖多种场景，以确保您的应用程序工作负载拥有足够的容量。

为了确保应用程序可用性，实例集针对指标尽快按比例向外扩展，但会逐渐向内扩展。

当竞价型实例集因目标容量下降而终止竞价型实例时，该实例将收到一条竞价型实例中断通知。

> **Note**
>
> 请勿编辑或删除竞价型实例集为目标跟踪扩展策略管理的 CloudWatch 警报。在删除目标跟踪扩展策略时，竞价型实例集将自动删除警报。

先决条件

- 竞价型实例集请求必须使用 `maintain` 作为请求类型。`request` 类型的请求不支持自动扩缩。
- 配置 竞价型实例集自动扩展所需的 IAM 权限。
- 查看注意事项。

配置目标跟踪策略

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择屏幕底部附近的自动扩缩选项卡。如果您选择了竞价型实例集的链接，则没有选项卡；而是向下滚动到自动扩缩部分。
5. 如果未配置自动扩展，请选择 Configure。

6. 使用扩展容量，使其介于设置实例集的最小和最大容量。实例集的自动扩展操作不会超出最小或最大容量范围。
7. 对于 Policy name（策略名称），输入此策略的名称。
8. 选择一个目标指标。
9. 为该指标键入一个目标值。
10. 对于冷却时间，指定新值（以秒为单位）或保留默认值。
11. （可选）要忽略根据当前配置创建横向缩减策略，请选择禁用缩减。您可以使用不同的配置创建一个横向缩减策略。
12. 选择 Save。

使用 Amazon CLI 配置目标跟踪策略

1. 使用 register-scalable-target 命令将竞价型实例集请求注册为可扩展目标。
2. 使用 put-scaling-policy 命令创建扩展策略。

步进扩展：使用步进扩展策略扩展竞价型实例集

在使用步进扩展策略时，您可以指定 CloudWatch 警报以触发扩展过程。例如，如果您希望在 CPU 利用率达到特定水平时横向扩展，可以使用 Amazon EC2 提供的 `CPUUtilization` 指标创建警报。

在创建步进扩展策略时，您必须指定以下扩展调整类型之一：

- 添加：按指定的容量单位数量或当前容量的指定百分比来增加实例集的目标容量。
- 移除：按指定的容量单位数量或当前容量的指定百分比来缩减实例集的目标容量。
- 设置为：将实例集的目标容量设为指定的容量单位数量。

当触发警报时，自动扩展过程使用执行容量和扩展策略计算新的目标容量，然后相应地更新目标容量。例如，假设目标容量和执行容量为 10，扩展策略加 1。触发警报时，自动扩展过程为 10 增加 1 得到 11，因此竞价型实例集将启动 1 个实例。

当竞价型实例集因目标容量下降而终止竞价型实例时，该实例将收到一条竞价型实例中断通知。

先决条件

- 竞价型实例集请求必须使用 `maintain` 作为请求类型。`request` 类型的请求不支持自动扩缩。

- 配置竞价型实例集自动扩展所需的 IAM 权限。
- 考虑哪些 CloudWatch 指标对您的应用程序比较重要。您可以根据 Amazon 提供的指标或您自己的自定义指标来创建 CloudWatch 告警。
- 如果您打算在扩展策略中使用 Amazon 指标，请为其启用 CloudWatch 指标集合（如果提供这些指标的服务默认未启用它的话）。
- 查看注意事项。

创建 CloudWatch 警报

1. 通过 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 在导航窗格中，展开警报，然后选择所有警报。
3. 选择 Create Alarm (创建警报)。
4. 在 Specify metric and conditions (指定指标和条件) 页面上，选择 Select metric (选择指标)。
5. 依次选择 EC2 竞价型实例、实例集请求指标，选择一个指标（例如，TargetCapacity），然后选择选择指标。

   这将显示 Specify metric and conditions (指定指标和条件) 页面，其中显示一个图表以及有关所选指标的其他信息。
6. 对于周期，选择警报的评估周期，例如 1 分钟。评估警报时，每个周期都聚合到一个数据点。

   > **Note**
   >
   > 周期越短，创建的警报越敏感。
7. 在 Conditions (条件) 下，通过定义阈值条件来定义警报。例如，您可以定义一个阈值，在指标值大于或等于 80% 时触发警报。
8. 在 Additional configuration (附加配置) 下，对于 Datapoints to alarm (触发警报的数据点数)，请指定必须有多少个数据点（评估期）处于 ALARM 状态才会触发警报，例如，1 个或 2 个（共 3 个）评估期。这将创建一个警报，如果多个连续周期超出阈值，该警报将进入 ALARM（警报）状态。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的评估警报。
9. 对于 Missing data treatment (缺失数据处理)，选择某个选项（或保留 Treat missing data as missing (将缺失的数据视为缺失) 的默认值）。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的配置 CloudWatch 警报处理缺少数据的方式。
10. 选择下一步。

11. （可选）要接收扩展事件的通知，在 Notification (通知) 中，您可以选择或创建要用于接收通知的 Amazon SNS 主题。当然，您也可以立即删除通知，之后按需添加通知。
12. 选择下一步。
13. 在添加名称和描述下，输入警报的名称和描述，然后选择下一步。
14. 选择创建警报。

为竞价型实例集配置分步扩展策略

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择屏幕底部附近的自动扩缩选项卡。如果您选择了竞价型实例集的链接，则没有选项卡；而是向下滚动到自动扩缩部分。
5. 如果未配置自动扩展，请选择 Configure。
6. 使用扩展容量，使其介于设置实例集的最小和最大容量。扩展策略不会将实例集扩展至超出最小或最大容量范围。
7. 在扩展策略下，对于策略类型，选择步进扩展策略。
8. 最初，扩展策略包含名为 ScaleUp 和 ScaleDown 的分步扩展策略。您可以完善这些策略，或选择 Remove policy 来删除它们。您也可以选择 Add policy (添加策略)。
9. 要定义策略，请执行以下操作：

    a. 对于 Policy name（策略名称），输入此策略的名称。

    b. 对于策略触发器，可以选择现有的警报，也可以选择创建警报打开 Amazon CloudWatch 控制台并创建警报。

    c. 对于修改容量，定义扩展量以及分步调整的下限和上限。您可以添加或删除特定数量的实例或现有实例集大小的百分比，也可将实例集设置为准确的大小。

    例如，要创建将实例集容量增加 30% 的步进扩展策略，请选择添加，在下一个字段中输入 30，然后选择百分比。默认情况下，添加策略的下限为警报阈值，上限为正（+）无穷。默认情况下，移除策略的上限为警报阈值，下限为负 (-) 无穷。

    d. （可选）要添加其他步骤，请选择添加步骤。

    e. 对于冷却时间，指定新值（以秒为单位）或保留默认值。

10. 选择保存。

使用 Amazon CLI 为竞价型实例集配置分步扩展策略

1. 使用 register-scalable-target 命令将竞价型实例集请求注册为可扩展目标。
2. 使用 put-scaling-policy 命令创建扩展策略。
3. 使用 put-metric-alarm 命令创建触发扩展策略的警报。

计划扩展：按计划扩展竞价型实例集

按计划扩展实例集使您可以按照可预测的需求变化来扩展应用程序。通过创建计划操作，您可以指示竞价型实例集在特定时间执行扩展活动。要创建计划的操作，您必须指定现有的竞价型实例集、必须发生扩展活动的时间以及所需的最小容量和最大容量。可以将计划操作配置为一次性或按定期计划进行扩展。如果需要更改，可以编辑或删除计划的操作。

先决条件

- 只能为现有竞价型实例集创建计划操作。您不能在创建竞价型实例集时创建计划操作。
- 竞价型实例集请求必须使用 `maintain` 作为请求类型。`request` 类型的请求不支持自动扩缩。
- 配置 竞价型实例集自动扩展所需的 IAM 权限。
- 查看注意事项。

创建一次性计划操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择屏幕底部附近的计划扩缩选项卡。如果您选择了竞价型实例集的链接，则没有选项卡；而是向下滚动到计划扩缩部分。
5. 选择创建计划操作。
6. 对于 Name (名称)，请指定计划操作的名称。
7. 对于 Minimum capacity (最小容量) 和/或 Maximum capacity (最大容量)，输入所需的值。
8. 对于 Recurrence (重复次数)，选择 Once (一次)。
9. （可选）对于 Start time (开始时间) 和/或 End time (结束时间)，选择所需的日期和时间。
10. 选择创建。

创建重复计划操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择屏幕底部附近的计划扩缩选项卡。如果您选择了竞价型实例集的链接，则没有选项卡；而是向下滚动到计划扩缩部分。
5. 对于 Name (名称)，请指定计划操作的名称。
6. 对于 Minimum capacity (最小容量) 和/或 Maximum capacity (最大容量)，输入所需的值。
7. 对于 Recurrence（重复次数），选择预定义计划之一（例如 Every day（每天）），或者选择 Custom（自定义）并输入 cron 表达式。有关计划扩展所支持的 cron 表达式的更多信息，请参阅《Amazon EventBridge User Guide》中的 Cron expressions。
8. （可选）对于 Start time (开始时间) 和/或 End time (结束时间)，选择所需的日期和时间。
9. 选择提交。

编辑计划操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择屏幕底部附近的计划扩缩选项卡。如果您选择了竞价型实例集的链接，则没有选项卡；而是向下滚动到计划扩缩部分。
5. 选择所需的计划操作，然后依次选择 Actions (操作) 和 Edit (编辑)。
6. 进行所需的更改，然后选择 Submit (提交)。

删除计划操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Spot Requests。
3. 选择竞价型实例集请求。
4. 选择屏幕底部附近的计划扩缩选项卡。如果您选择了竞价型实例集的链接，则没有选项卡；而是向下滚动到计划扩缩部分。

5. 选择所需的计划操作，然后依次选择 Actions (操作) 和 Delete (删除)。
6. 当系统提示进行确认时，选择 Delete (删除)。

使用 Amazon CLI 管理计划扩展

使用以下命令：

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

监控 EC2 实例集或竞价型实例集

有效监控您的 EC2 实例集或竞价型实例集对于保持最佳性能和确保可靠性至关重要。有各种工具可以帮助您实现这一目标，包括本主题中介绍的 Amazon CloudWatch 和 Amazon EventBridge。

使用 CloudWatch，您可以收集和跟踪指标，设置警报，并自动应对实例集的状态变化。

借助 EventBridge，您可以通过编程方式监控和响应实例集发出的事件。通过在 EventBridge 中定义规则，您可以自动响应特定的实例集事件，例如实例终止或实例集状态更改，从而提高运营效率。

主题

- 使用 CloudWatch 监控 EC2 实例集或竞价型实例集
- 使用 Amazon EventBridge 监控并以编程方式响应您的 EC2 实例集或竞价型实例集发出的事件

使用 CloudWatch 监控 EC2 实例集或竞价型实例集

您可以使用本节所述的 Amazon CloudWatch 指标监控 EC2 实例集或竞价型实例集。

> **Important**
>
> 为确保准确性，我们建议您在使用这些指标时启用详细监控。有关更多信息，请参阅 管理 EC2 实例的详细监控。

有关使用 CloudWatch 的更多信息，请参阅使用 CloudWatch 监控您的实例。

EC2 实例集和竞价型实例集指标

AWS/EC2Spot 命名空间包含以下实例集指标，以及实例集中竞价型实例的 CloudWatch 指标。有关更多信息，请参阅 实例指标。

指标	说明
AvailableInstancePoolsCount	实例集请求中指定的竞价型容量池。 单位：计数
BidsSubmittedForCapacity	Amazon EC2 已提交实例集请求的容量。 单位：计数
EligibleInstancePoolCount	在 Amazon EC2 可以完成请求的实例集请求中指定的竞价型容量池。在您愿意为竞价型实例支付的最高价格低于 Spot 价格或 Spot 价格高于按需型实例价格的池中，Amazon EC2 不会完成请求。 单位：计数
FulfilledCapacity	Amazon EC2 已执行的容量。 单位：计数
MaxPercentCapacityAllocation	实例集请求中指定的所有实例集池间的 PercentCapacityAllocation 最大值。 单位：百分比
PendingCapacity	TargetCapacity 与 FulfilledCapacity 之间的区别。 单位：计数
PercentCapacityAllocation	针对所指定维度的 Spot 容量池分配的容量。要获取所有 Spot 容量池中记录的最大值，请使用 MaxPercentCapacityAllocation 。 单位：百分比

指标	说明
TargetCapacity	实例集请求的目标容量。 单位：计数
TerminatingCapacity	因预置容量大于目标容量而终止的容量。 单位：计数

如果指标的度量单位是 Count，则最有用的统计信息是 Average。

EC2 实例集和竞价型实例集维度

要筛选实例集的数据，请使用以下维度。

Dimensions	描述
AvailabilityZone	按照可用区筛选数据。
FleetRequestId	按照实例集请求筛选数据。
InstanceType	按实例类型筛选数据。

查看 EC2 实例集或竞价型实例集的 CloudWatch 指标

您可以使用 Amazon CloudWatch 控制台查看实例集的 CloudWatch 指标。这些指标显示为监控图表。如果实例集处于活动状态，则这些图表会显示数据点。

指标首先按命名空间进行分组，然后按各命名空间内的各种维度组合进行分组。例如，您可以按实例集请求 ID、实例类型或可用区查看所有实例集指标或实例集指标组。

查看实例集指标

1. 通过 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 在导航窗格中，展开指标，然后选择所有指标。
3. 选择 EC2 Spot 命名空间。

> **Note**
>
> 如果未显示 EC2 Spot 命名空间，这种情况有两个原因。您从未在该区域中使用过 EC2 实例集或竞价型实例集 – 只有正在使用的 Amazon 服务才会将指标发送到 Amazon CloudWatch。或者，如果您在该区域中使用了 EC2 实例集或竞价型实例集，但过去两周没有使用，则不会显示命名空间。

4. 要按维度筛选指标，请选择下列选项之一：
   - 实例集请求指标 – 按实例集请求分组
   - 按可用区 – 按实例集请求和可用区分组
   - 按实例类型 – 按实例集请求和实例类型分组
   - 按可用区/实例类型 – 按实例集请求、可用区和实例类型分组
5. 要查看某指标的数据，请选中该指标旁边的复选框。

使用 Amazon EventBridge 监控并以编程方式响应您的 EC2 实例集或竞价型实例集发出的事件

当 EC2 实例集或竞价型实例集的状态发生变化时，其将发出通知。该通知作为发送到 Amazon EventBridge（以前称为 Amazon CloudWatch Events）的事件提供。尽最大努力发出事件。

您可以使用 Amazon EventBridge 创建触发编程操作以响应事件的规则。例如，您可以创建两个 EventBridge 规则：一个在实例集状态变化时触发，另一个在实例集中的实例终止时触发。在本例中，您可以配置第一条规则，这样如果实例集状态发生变化，该规则将调用 SNS 主题，向您发送电子邮件通知。您可以配置第二条规则，这样如果实例集中的实例被终止，该规则将调用 Lambda 函数以启动新实例。

> **Note**
>
> 只有类型 `maintain` 和 `request` 的队列才能触发事件。类型 `instant` 的队列不会触发事件，因为它们提交同步一次性请求，并且队列的状态立即可在响应中获知。要使用 Amazon EventBridge 监控实例集事件，请求类型必须为 `maintain` 或 `request`。

有关如何描述实例集事件历史记录的说明，请参阅描述 EC2 实例集的事件历史记录。

主题

- 创建 Amazon EventBridge 规则以监控 EC2 实例集或竞价型实例集事件
- EC2 队列 事件类型
- Spot 队列事件类型

创建 Amazon EventBridge 规则以监控 EC2 实例集或竞价型实例集事件

当 EC2 实例集或竞价型实例集发出状态变更通知时，该通知将作为事件以 JSON 文件的形式发送到 Amazon EventBridge。如果 EventBridge 检测到与规则中定义的模式匹配的事件模式，则 EventBridge 调用规则中指定的一个或多个目标。

您可以编写 EventBridge 规则，从而根据匹配的事件模式自动执行操作。

事件中的下列字段构成规则中定义的事件模式：

```
"source": "aws.ec2fleet"
```

识别来自 EC2 队列 的事件。

```
"detail-type": "EC2 Fleet State Change"
```

识别事件类型。

```
"detail": { "sub-type": "submitted" }
```

识别事件子类型。

有关 EC2 实例集和竞价型实例集事件和示例事件数据的列表，请参阅 EC2 队列 事件类型和Spot 队列事件类型。

示例

- 创建 EventBridge 规则以发送通知
- 创建 EventBridge 规则以触发 Lambda 函数

创建 EventBridge 规则以发送通知

下面的示例可创建 EventBridge 规则，以便在 Amazon EC2 每次发出 EC2 队列状态变化通知时发送电子邮件、短信或移动推送通知。本示例中的信号作为 `EC2 Fleet State Change` 事件发出，这将触发规则定义的操作。

先决条件

创建 EventBridge 规则之前，您必须为电子邮件、短信或移动推送通知创建 Amazon SNS 主题。

创建 EventBridge 规则以在 EC2 队列 状态变化时发送通知

1. 打开位于 https://console.aws.amazon.com/events/ 的 Amazon EventBridge 控制台。
2. 选择 Create rule（创建规则）。
3. 对于定义规则详细信息，请执行以下操作：

   a. 输入规则的 Name (名称) 和“Description (描述)”（可选）。

      规则不能与同一区域中的另一个规则和同一事件总线上的名称相同。

   b. 对于 Event bus（事件总线），选择 default（默认）。当您账户中的某个 Amazon 服务生成一个事件时，它始终会发送到您账户的默认事件总线。

   c. 对于 Rule type（规则类型），选择 Rule with an event pattern（具有事件模式的规则）。

   d. 选择下一步。

4. 对于 Build event pattern（构建事件模式），执行以下操作：

   a. 对于 Event source（事件源），选择 Amazon 事件或 EventBridge 合作伙伴事件。

   b. 对于 Event pattern（事件模式），在此示例中，您将指定以下事件模式以匹配 `EC2 Fleet Instance Change` 事件。

```
{
 "source": ["aws.ec2fleet"],
 "detail-type": ["EC2 Fleet Instance Change"]
}
```

      要添加事件模式，您可以通过选择 Event pattern form（事件模式表）使用模板，或者通过选择 Custom pattern (JSON editor)（自定义模式（JSON 编辑器））指定您自己的模式，如下所示：

      i. 要使用模板创建事件模式，请执行以下操作：

         A. 选择 Event pattern form（事件模式表）。

         B. 对于 Event source（事件源），选择 Amazon services（服务）。

         C. 对于 Amazon 服务，选择 EC2 实例集。

D. 对于 Event type（事件类型），选择 EC2 Fleet Instance Change（EC2 实例集实例更改）。

E. 要自定义模板，请选择 Edit pattern（编辑模式），然后进行更改以匹配示例事件模式。

ii. （可选）要指定自定义事件模式，请执行以下操作：

A. 选择 Custom pattern (JSON editor)（自定义模式（JSON 编辑器））。

B. 在 Event pattern（事件模式）框中，为此示例添加事件模式。

c. 选择下一步。

5. 对于 Select target(s)（选择目标），请执行以下操作：

a. 对于 Target types（目标类型），选择 Amazon service（服务）。

b. 对于 Select a target（选择目标），选择 SNS topic（SNS 主题）以在事件发生时发送电子邮件、短信或移动推送通知。

c. 对于 Topic（主题），选择现有主题。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。有关更多信息，请参阅Amazon Simple Notification Service 开发人员指南中的使用 Amazon SNS 进行应用程序对人 (A2P) 消息传送。

d. （可选）在 Additional settings（其他设置）下，您可以选择配置其他设置。有关更多信息，请参阅《Amazon EventBridge User Guide》中的 Creating Amazon EventBridge rules that react to events（步骤 16）。

e. 选择下一步。

6. （可选）对于 Tags（标签），您可以选择向规则分配一个或多个标签，然后选择 Next（下一步）。

7. 对于 Review and create（查看与创建），执行以下操作：

a. 查看规则的详细信息并根据需要对其进行修改。

b. 选择 Create rule（创建规则）。

有关更多信息，请参阅 Amazon EventBridge 用户指南中的 Amazon EventBridge 规则和 Amazon EventBridge 事件模式。

创建 EventBridge 规则以触发 Lambda 函数

下面的示例可创建 EventBridge 规则，以便在 Amazon EC2 每次在实例启动时发出 EC2 队列实例变更通知时触发 Lambda 函数。本示例中的信号作为 `EC2 Fleet Instance Change` 事件子类型 `launched` 发出，这将触发规则定义的操作。

在创建 EventBridge 规则之前，您必须创建 Lambda 函数。

创建要在 EventBridge 规则中使用的 Lambda 函数

1. 通过 https://console.aws.amazon.com/lambda/ 打开 Amazon Lambda 控制台。
2. 选择创建函数。
3. 输入函数的名称，配置代码，然后选择 Create function（创建函数）。

   有关更多信息，请参阅《Amazon Lambda 开发人员指南》中的创建第一个 Lambda 函数。

创建 EventBridge 规则以在 EC2 队列 中的实例改变状态时触发 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/events/ 的 Amazon EventBridge 控制台。
2. 选择 Create rule（创建规则）。
3. 对于定义规则详细信息，请执行以下操作：

   a. 输入规则的 Name (名称) 和“Description (描述)”（可选）。

      规则不能与同一区域中的另一个规则和同一事件总线上的名称相同。

   b. 对于 Event bus（事件总线），选择 default（默认）。当您账户中的某个 Amazon 服务生成一个事件时，它始终会发送到您账户的默认事件总线。

   c. 对于 Rule type（规则类型），选择 Rule with an event pattern（具有事件模式的规则）。

   d. 选择下一步。
4. 对于 Build event pattern（构建事件模式），执行以下操作：

   a. 对于 Event source（事件源），选择 Amazon 事件或 EventBridge 合作伙伴事件。

   b. 对于 Event pattern（事件模式），在此示例中，您将指定以下事件模式以匹配 `EC2 Fleet Instance Change` 事件和 `launched` 子类型。

```
{
 "source": ["aws.ec2fleet"],
 "detail-type": ["EC2 Fleet Instance Change"],
```

```
 "detail": {
   "sub-type": ["launched"]
}
```

要添加事件模式，您可以通过选择 Event pattern form（事件模式表）使用模板，或者通过选择 Custom pattern (JSON editor)（自定义模式（JSON 编辑器））指定您自己的模式，如下所示：

i. 要使用模板创建事件模式，请执行以下操作：

A. 选择 Event pattern form（事件模式表）。

B. 对于 Event source（事件源），选择 Amazon services（服务）。

C. 对于 Amazon 服务，选择 EC2 实例集。

D. 对于 Event type（事件类型），选择 EC2 Fleet Instance Change（EC2 实例集实例更改）。

E. 选择 Edit pattern（编辑模式），然后添加 `"detail": {"sub-type": ["launched"]` 以匹配示例事件模式。对于正确的 JSON 格式，在前面的方括号（，）之后插入一个逗号（]）。

ii. （可选）要指定自定义事件模式，请执行以下操作：

A. 选择 Custom pattern (JSON editor)（自定义模式（JSON 编辑器））。

B. 在 Event pattern（事件模式）框中，为此示例添加事件模式。

c. 选择下一步。

5. 对于 Select target(s)（选择目标），请执行以下操作：

a. 对于 Target types（目标类型），选择 Amazon service（服务）。

b. 对于 Select a target（选择目标），选择 SNS topic（SNS 主题）以在事件发生时发送电子邮件、短信或移动推送通知。

c. 对于 Topic（主题），选择 Lambda function（Lambda 函数），对于 Function（函数），选择您创建的用于在事件发生时响应的函数。

d. （可选）在 Additional settings（其他设置）下，您可以选择配置其他设置。有关更多信息，请参阅《Amazon EventBridge User Guide》中的 Creating Amazon EventBridge rules that react to events（步骤 16）。

e. 选择下一步。

6. （可选）对于 Tags（标签），您可以选择向规则分配一个或多个标签，然后选择 Next（下一步）。

7. 对于 Review and create（查看与创建），执行以下操作：

   a. 查看规则的详细信息并根据需要对其进行修改。

   b. 选择 Create rule（创建规则）。

有关如何创建 Lambda 函数和运行 Lambda 函数的 EventBridge 规则的教程，请参阅 Amazon Lambda 开发人员指南中的教程：使用 EventBridge 记录 Amazon EC2 实例的状态。

EC2 队列 事件类型

EC2 队列 事件类型有五种。对于每种事件类型，都有几个子类型。

事件类型

- EC2 队列状态变化
- EC2 实例集竞价型实例请求更改
- EC2 队列实例更改
- EC2 队列信息
- EC2 队列错误

EC2 队列状态变化

当 EC2 队列 状态变化时，EC2 队列 会将 `EC2 Fleet State Change` 事件发送到 Amazon EventBridge。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "715ed6b3-b8fc-27fe-fad6-528c7b8bf8a2",
    "detail-type": "EC2 Fleet State Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:20Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-
be4d-6b0809bfff0a"
    ],
    "detail": {
```

```
        "sub-type": "active"
    }
}
```

sub-type 可使用的值为：

active

EC2 队列 请求已验证，并且 Amazon EC2 正在尝试使正在运行的实例保持目标数量。

deleted

EC2 队列 请求已删除，没有正在运行的实例。EC2 队列 将在其实例终止两天后被删除。

deleted_running

EC2 队列 请求已删除，且不启动其他实例。现有实例将继续运行，直至被中断或终止。请求会保持此状态，直到所有实例都已中断或终止。

deleted_terminating

EC2 队列 请求已删除，正在终止其实例。请求会保持此状态，直到所有实例都已终止。

expired

EC2 队列 请求已过期。如果请求使用 TerminateInstancesWithExpiration 集创建，则随后的 terminated 事件指示实例已终止。

modify_in_progress

正在修改 EC2 队列 请求。请求将保持该状态，直到完全处理修改。

modify_succeeded

EC2 队列 请求已修改。

submitted

EC2 队列 正在接受评估，并且 Amazon EC2 正准备启动目标数量的实例。

progress

该 EC2 队列 请求正在执行过程中。

EC2 实例集竞价型实例请求更改

当队列中的 Spot 实例请求状态变化时，EC2 队列 会将 EC2 Fleet Spot Instance Request Change 事件发送到 Amazon EventBridge。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "19331f74-bf4b-a3dd-0f1b-ddb1422032b9",
    "detail-type": "EC2 Fleet Spot Instance Request Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/
fleet-83fd4e48-552a-40ef-9532-82a3acca5f10"
    ],
    "detail": {
        "spot-instance-request-id": "sir-rmqske6h",
        "description": "SpotInstanceRequestId sir-rmqske6h, PreviousState:
 cancelled_running",
        "sub-type": "cancelled"
    }
}
```

sub-type 可使用的值为：

active

竞价型实例请求已执行并有关联的竞价型实例。

cancelled

您取消了竞价型实例请求，或竞价型实例请求已过期。

disabled

您停止了竞价型实例。

submitted

已提交竞价型实例请求。

EC2 队列实例更改

当队列中的实例状态变化时，EC2 队列 会将 EC2 Fleet Instance Change 事件发送到 Amazon EventBridge。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "542ce428-c8f1-0608-c015-e8ed6522c5bc",
    "detail-type": "EC2 Fleet Instance Change",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T09:00:23Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-
be4d-6b0809bfff0a"
    ],
    "detail": {
        "instance-id": "i-0c594155dd5ff1829",
        "description": "{\"instanceType\":\"c5.large\",\"image\":\"ami-6057e21a\",
\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":\"us-east-1d\"}",
        "sub-type": "launched"
    }
}
```

sub-type 可使用的值为：

launched

启动了一个新实例。

terminated

实例已终止。

termination_notified

当 Amazon EC2 在缩减期间终止竞价型实例时，机群的目标容量被缩减，例如，从目标容量 4 减至目标容量 3 时，会发送实例终止通知。

EC2 队列信息

在履行过程中出错时，EC2 队列 会将 EC2 Fleet Information 事件发送到 Amazon EventBridge。信息事件不会阻止队列尝试实现其目标容量。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "76529817-d605-4571-7224-d36cc1b2c0c4",
    "detail-type": "EC2 Fleet Information",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-11-09T08:17:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-8becf5fe-
bb9e-415d-8f54-3fa5a8628b91"
    ],
    "detail": {
        "description": "c4.xlarge, ami-0947d2ba12ee1ff75, Linux/UNIX, us-east-1a,
 Spot price in either SpotFleetRequestConfigData or SpotFleetLaunchSpecification or
 LaunchTemplate or LaunchTemplateOverrides is less than Spot market price $0.0619",
        "sub-type": "launchSpecUnusable"
    }
}
```

sub-type 可使用的值为：

fleetProgressHalted

每个启动规范中的价格无效，因为它低于 Spot 价格（所有启动规范都已生成 launchSpecUnusable 事件）。如果 Spot 价格发生变化，启动规范可能会变为有效。

launchSpecTemporarilyBlacklisted

配置无效，多次启动实例的尝试都已失败。有关更多信息，请参阅事件的描述。

launchSpecUnusable

启动规范中的价格无效，因为它低于 Spot 价格。

registerWithLoadBalancersFailed

尝试向负载均衡器注册实例失败。有关更多信息，请参阅事件的描述。

EC2 队列错误

在履行过程中出错时，EC2 队列 会将 EC2 Fleet Error 事件发送到 Amazon EventBridge。错误事件阻止了队列尝试实现其目标容量。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "69849a22-6d0f-d4ce-602b-b47c1c98240e",
    "detail-type": "EC2 Fleet Error",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-10-07T01:44:24Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-9bb19bc6-60d3-4fd2-ae47-
d33e68eafa08"
    ],
    "detail": {
        "description": "m3.large, ami-00068cd7555f543d5, Linux/UNIX: IPv6 is not
 supported for the instance type 'm3.large'. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

sub-type 可使用的值为：

iamFleetRoleInvalid

EC2 机群没有启动或终止实例所需的权限。

allLaunchSpecsTemporarilyBlacklisted

所有配置均无效，多次启动实例的尝试已失败。有关更多信息，请参阅事件的描述。

spotInstanceCountLimitExceeded

您已达到可启动的竞价型实例的数量限制。

spotFleetRequestConfigurationInvalid

该配置无效。有关更多信息，请参阅事件的描述。

Spot 队列事件类型

> **Note**
>
> 竞价型实例集仍在使用旧版 API，且暂无相关投资计划。建议改用 EC2 实例集或自动扩缩组。有关更多信息，请参阅 哪种是实例集最佳使用方法？。

Spot 队列事件类型有五种。对于每种事件类型，都有几个子类型。

事件类型

- EC2 Spot 队列状态变化
- EC2竞价型实例集竞价型实例请求更改
- EC2 Spot 队列实例更改
- EC2 Spot 队列信息
- EC2 Spot 队列错误

EC2 Spot 队列状态变化

当 Spot 队列状态变化时，Spot 队列会将 `EC2 Spot Fleet State Change` 事件发送到 Amazon EventBridge。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "d1af1091-6cc3-2e24-203a-3b870e455d5b",
    "detail-type": "EC2 Spot Fleet State Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:57:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-4b6d274d-0cea-4b2c-
b3be-9dc627ad1f55"
    ],
    "detail": {
        "sub-type": "submitted"
    }
```

```
}
```

sub-type 可使用的值为：

active

已验证竞价型实例集请求，并且 Amazon EC2 正在尝试维护目标数量的正在运行的实例。

cancelled

竞价型实例集请求已取消，没有正在运行的实例。Spot 队列将在其实例终止两天后被删除。

cancelled_running

竞价型实例集请求已取消，且不启动其他实例。现有实例将继续运行，直至被中断或终止。请求会保持此状态，直到所有实例都已中断或终止。

cancelled_terminating

竞价型实例集请求已取消，正在终止其实例。请求会保持此状态，直到所有实例都已终止。

expired

Spot 队列请求已过期。如果请求使用 TerminateInstancesWithExpiration 集创建，则随后的 terminated 事件指示实例已终止。

modify_in_progress

Spot 队列请求正在修改中。请求将保持该状态，直到完全处理修改。

modify_succeeded

Spot 队列请求已修改。

submitted

正在评估 Spot 队列请求，并且 Amazon EC2 正准备启动目标数量的实例。

progress

Spot 队列请求正在执行过程中。

EC2竞价型实例集竞价型实例请求更改

当队列中的 Spot 实例请求状态变化时，Spot 队列会将 EC2 Spot Fleet Spot Instance Request Change 事件发送到 Amazon EventBridge。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "cd141ef0-14af-d670-a71d-fe46e9971bd2",
    "detail-type": "EC2 Spot Fleet Spot Instance Request Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:53:21Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-
a98d2133-941a-47dc-8b03-0f94c6852ad1"
    ],
    "detail": {
        "spot-instance-request-id": "sir-a2w9gc5h",
        "description": "SpotInstanceRequestId sir-a2w9gc5h, PreviousState:
 cancelled_running",
        "sub-type": "cancelled"
    }
}
```

sub-type 可使用的值为：

active

竞价型实例请求已执行并有关联的竞价型实例。

cancelled

您取消了竞价型实例请求，或竞价型实例请求已过期。

disabled

您停止了竞价型实例。

submitted

已提交竞价型实例请求。

EC2 Spot 队列实例更改

当队列中的实例状态变化时，Spot 队列会将 EC2 Spot Fleet Instance Change 事件发送到 Amazon EventBridge。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "11591686-5bd7-bbaa-eb40-d46529c2710f",
    "detail-type": "EC2 Spot Fleet Instance Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T07:25:02Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-c8a764a4-bedc-4b62-
af9c-0095e6e3ba61"
    ],
    "detail": {
        "instance-id": "i-08b90df1e09c30c9b",
        "description": "{\"instanceType\":\"r4.2xlarge\",\"image\":
\"ami-032930428bf1abbff\",\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":
\"us-east-1a\"}",
        "sub-type": "launched"
    }
}
```

sub-type 可使用的值为：

launched

启动了一个新实例。

terminated

实例已终止。

termination_notified

当 Amazon EC2 在缩减期间终止竞价型实例时，机群的目标容量被缩减，例如，从目标容量 4 减至目标容量 3 时，会发送实例终止通知。

EC2 Spot 队列信息

在履行过程中出错时，Spot 队列会将 EC2 Spot Fleet Information 事件发送到 Amazon EventBridge。信息事件不会阻止队列尝试实现其目标容量。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "73a60f70-3409-a66c-635c-7f66c5f5b669",
    "detail-type": "EC2 Spot Fleet Information",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-08T20:56:12Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-2531ea06-
af18-4647-8757-7d69c94971b1"
    ],
    "detail": {
        "description": "r3.8xlarge, ami-032930428bf1abbff, Linux/UNIX, us-east-1a, Spot
 bid price is less than Spot market price $0.5291",
        "sub-type": "launchSpecUnusable"
    }
}
```

sub-type 可使用的值为：

fleetProgressHalted

每个启动规范中的价格无效，因为它低于 Spot 价格（所有启动规范都已生成 launchSpecUnusable 事件）。如果 Spot 价格发生变化，启动规范可能会变为有效。

launchSpecTemporarilyBlacklisted

配置无效，多次启动实例的尝试都已失败。有关更多信息，请参阅事件的描述。

launchSpecUnusable

启动规范中的价格无效，因为它低于 Spot 价格。

registerWithLoadBalancersFailed

尝试向负载均衡器注册实例失败。有关更多信息，请参阅事件的描述。

EC2 Spot 队列错误

在履行过程中出错时，Spot 队列会将 EC2 Spot Fleet Error 事件发送到 Amazon EventBridge。错误事件阻止了队列尝试实现其目标容量。

以下是此事件的示例数据。

```
{
    "version": "0",
    "id": "10adc4e7-675c-643e-125c-5bfa1b1ba5d2",
    "detail-type": "EC2 Spot Fleet Error",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T06:56:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/
sfr-38725d30-25f1-4f30-83ce-2907c56dba17"
    ],
    "detail": {
        "description": "r4.2xlarge, ami-032930428bf1abbff, Linux/UNIX: The
 associatePublicIPAddress parameter can only be specified for the network interface
 with DeviceIndex 0. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

sub-type 可使用的值为：

iamFleetRoleInvalid

竞价型实例集不具有启动或终止实例所需的权限。

allLaunchSpecsTemporarilyBlacklisted

所有配置均无效，多次启动实例的尝试已失败。有关更多信息，请参阅事件的描述。

spotInstanceCountLimitExceeded

您已达到可启动的竞价型实例的数量限制。

spotFleetRequestConfigurationInvalid

该配置无效。有关更多信息，请参阅事件的描述。

EC2 实例集的教程

可通过不同的方法配置 EC2 实例集。您选择的配置取决于特定使用案例。

以下教程涵盖了一些可能的使用案例，并提供实现这些使用案例所需的任务。

使用案例	指向教程的链接
使用实例权重来管理 EC2 实例集的可用性和性能。 您可以借助实例权重为 EC2 实例集中的每种实例类型分配权重，来表示它们相对于彼此的计算容量和性能。实例集可以根据权重使用指定实例类型的任意组合，前提是可以满足所需的目标容量。	教程：将 EC2 实例集配置为使用实例权重
使用按需容量可确保高峰时段的可用性，但可从更低成本的额外竞价容量获益。 将您的 EC2 实例集配置为使用按需型实例作为主要容量，以确保高峰时段的可用容量。此外，为竞价型实例分配一些容量以享受折扣价，同时请记住，如果 Amazon EC2 需要恢复容量，则竞价型实例可能会中断。	教程：将 EC2 实例集配置为使用按需型实例作为主容量
使用容量预留为按需型实例预留计算容量。 将 EC2 实例集配置为在启动按需型实例时首先使用 `targeted` 容量预留。如果您有严格的容量要求，并且正在运行需要一定级别的长期或短期容量保证的业务关键型工作负载，我们建议您创建容量预留，以确保在需要时始终可以获得所需时长的 Amazon EC2 容量。	教程：使用目标容量预留配置 EC2 实例集以启动按需型实例
使用容量块为您的 ML 工作负载预留广受欢迎的 GPU 实例。 将 EC2 实例集配置为将实例启动到容量块中。	教程：将 EC2 实例集配置为将实例启动到容量块中
使用可中断容量预留，可以临时重新利用整个 Amazon 组织的闲置容量并节省成本。 配置 EC2 队列，使其在可中断容量预留中启动实例，这些预留代表您 Amazon 组织内的空闲	教程：将 EC2 实例集配置为将实例启动到可中断容量预留中

使用案例	指向教程的链接
容量。容量所有者可随时收回容量。回收后，EC2 会在 2 分钟通知后终止实例。	

教程：将 EC2 实例集配置为使用实例权重

该教程使用名为 Example Corp 的虚构公司说明使用实例权重请求 EC2 队列的过程。

目标

Example Corp 是一家医药公司，该公司想要使用 Amazon EC2 的计算功能来筛查可能用于对抗癌症的化学成分。

规划

Example Corp 首先查看Spot 最佳实践。然后，Example Corp 确定了他们的 EC2 队列的要求。

实例类型

Example Corp 有一个计算和内存密集型应用程序，该应用程序在至少 60 GB 内存和八个虚拟 CPU (vCPU) 的情况下性能最佳。他们希望以尽可能低的价格为该应用程序提供尽可能多的这些资源。Example Corp 认定以下任意 EC2 实例类型都能满足其需求：

实例类型	内存 (GiB)	vCPU
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

以单位数表示的目标容量

采用实例权重，目标容量可以等于几个实例 (默认) 或一些因素 (如内核 (vCPU)、内存 (GiB) 和存储 (GB)) 的组合。将其应用程序的基本要求 (60 GB RAM 和八个 vCPU) 作为一个单位，Example Corp

确定，此数量的 20 倍应可满足其需求。因此该公司将其 EC2 实例集请求的目标容量设置为 20 个单位。

实例权重

确定目标容量后，Example Corp 计算了实例权重。为了计算每个实例类型的实例权重，他们按如下所示确定每个实例类型需要多少单位才能达到目标容量：

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 个 20 单位
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 个 20 单位
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 个 20 单位

因此，Example Corp 在其EC2 队列请求中将实例权重 1、2 和 4 分配给相应的启动配置。

每单位小时价格

Example Corp 将每实例小时按需价格作为其价格的起点。他们也可以使用最近的 Spot 价格或两者的组合。为了计算每单位小时价格，他们将每实例小时起始价格除以权重。例如：

实例类型	按需价格	实例权重	每单位小时价格
r3.2xLarge	0.7 美元	1	0.7 美元
r3.4xLarge	1.4 美元	2	0.7 美元
r3.8xLarge	2.8 美元	4	0.7 美元

Example Corp 可能会使用每单位小时全局价格 0.7 美元，这对于所有三种实例类型来说是非常有竞争力的。他们可能还会使用每单位小时全局价格 0.7 USD，并在 `r3.8xlarge` 启动规范中使用特定的每单位小时价格 0.9 USD。

验证权限

在创建 EC2 队列请求之前，Example Corp 验证它是否拥有具备所需权限的 IAM 角色。有关更多信息，请参阅EC2 实例集先决条件。

创建启动模板

接下来，Example Corp 会创建启动模板。启动模板 ID 将在下一个步骤中使用。有关更多信息，请参阅创建 Amazon EC2 启动模板。

创建 EC2 队列

Example Corp 为其 EC2 队列创建一个具有以下配置的文件 (`config.json`)。在以下示例中，将资源标识符替换为您自己的资源标识符。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r3.2xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "r3.4xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 2
                },
                {
                    "InstanceType": "r3.8xlarge",
                    "MaxPrice": "0.90",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 4
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Example Corp 使用以下 create-fleet 命令创建 EC2 队列。

```
aws ec2 create-fleet --cli-input-json file://config.json
```

有关更多信息，请参阅 创建 EC2 实例集。

执行

分配策略确定竞价型实例所来自的 Spot 容量池。

使用 `lowest-price` 策略（这是默认策略）时，Spot 实例来自在执行时具有最低每单位价格的池。为了提供 20 个单位的容量，EC2 队列有三种做法：启动 20 个 `r3.2xlarge` 实例（20 除以 1）、10 个 `r3.4xlarge` 实例（20 除以 2）或 5 个 `r3.8xlarge` 实例（20 除以 4）。

如果 Example Corp 使用 `diversified` 策略，则 Spot 实例来自所有三个池。EC2 队列会启动 6 个 `r3.2xlarge` 实例（提供 6 个单位）、3 个 `r3.4xlarge` 实例（提供 6 个单位）和 2 个 `r3.8xlarge` 实例（提供 8 个单位），总共 20 个单位。

教程：将 EC2 实例集配置为使用按需型实例作为主容量

该教程使用名为 ABC Online 的虚构公司说明请求 EC2 队列并将按需作为主容量和 Spot 容量（如果可用）的过程。

目标

ABC Online 是一家餐饮送货公司，旨在跨 EC2 实例类型和购买选项预置 Amazon EC2 容量，以实现其预期的扩展、性能和成本。

规划

ABC Online 需要固定容量以应对高峰时段，但也想要以较低价格从额外容量获益。该公司确定其 EC2 实例集的以下要求：

- 按需型实例容量 – ABC Online 需要使用 15 个按需型实例，以确保它们可以处理高峰期的流量。
- 竞价型实例容量：为了以较低的价格提高性能，ABC Online 打算预置 5 个竞价型实例。

验证权限

在创建 EC2 队列之前，ABC Online 验证它是否拥有具备所需权限的 IAM 角色。有关更多信息，请参阅EC2 实例集先决条件。

创建启动模板

接下来，ABC Online 会创建启动模板。启动模板 ID 将在下一个步骤中使用。有关更多信息，请参阅创建 Amazon EC2 启动模板。

创建 EC2 队列

ABC Online 为其 EC2 队列创建一个具有以下配置的文件 (config.json)。在以下示例中，将资源标识符替换为您自己的资源标识符。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "2"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "OnDemandTargetCapacity":15,
        "DefaultTargetCapacityType": "spot"
    }
}
```

ABC Online 使用以下 create-fleet 命令创建 EC2 队列。

```
aws ec2 create-fleet --cli-input-json file://config.json
```

有关更多信息，请参阅 创建 EC2 实例集。

执行

分配策略确定按需容量始终得到满足，而目标容量的余额将在具有可用容量的情况下作为竞价型实例容量执行。

教程：使用目标容量预留配置 EC2 实例集以启动按需型实例

本教程将指导您完成必须执行的所有步骤，以便 EC2 队列将按需实例启动到 targeted 容量预留。

您将学习如何将队列配置为使用 targeted 启动按需实例时，首先使用按需容量预留。您还将了解如何配置队列，以便当按需目标总容量超过可用未使用容量预留的数量时，队列使用指定的分配策略来选择启动剩余目标容量的实例池。

EC2 队列配置

在本教程中，实例集配置如下：

- 目标容量：10 个按需实例
- 未使用总数 targeted 容量预留：6（少于队列的 10 个按需实例的按需目标容量）
- 容量预留池数量：2 (us-east-1a 和 us-east-1b)
- 每个池的容量预留数量：3
- 按需分配策略：lowest-price（如果未使用的容量预留数少于按需目标容量，队列将根据按需分配策略确定启动剩余的按需容量的池。）

  请注意，您也可以使用 prioritized 分配策略而不是 lowest-price 分配策略。

要将按需实例启动到 targeted 能力预留，您必须执行以下几个步骤：

- 步骤 1：创建容量预留
- 步骤 2：创建容量预留资源组
- 步骤 3：将容量预留添加到容量预留资源组
- （可选）步骤 4：查看资源组中的容量预留
- 步骤 5：创建启动模板，该模板指定容量预留的目标是特定资源组
- （可选）步骤 6：描述启动模板
- 步骤 7：创建 EC2 队列
- （可选）步骤 8：查看剩余未使用的容量预留数

步骤 1：创建容量预留

使用创建容量预留命令创建容量预留，三个 us-east-1a，另外三个 us-east-1b。除了可用区之外，容量预留的其他属性是相同的。

us-east-1a 中的 3 个容量预留

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1a \
```

```
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

生成的容量预留 ID 示例

```
cr-1234567890abcdef1
```

us-east-1b 中的 3 个容量预留

```
aws ec2 create-capacity-reservation \
    --availability-zone us-east-1b \
    --instance-type c5.xlarge \
    --instance-platform Linux/UNIX \
    --instance-count 3 \
    --instance-match-criteria targeted
```

生成的容量预留 ID 示例

```
cr-54321abcdef567890
```

步骤 2：创建容量预留资源组

使用 `resource-groups` 服务和 create-group 命令可创建容量预留资源组。在此示例中，资源组名为 `my-cr-group`。有关必须创建资源组原因的信息，请参阅 使用容量预留来预留 EC2 实例集中的按需型容量。

```
aws resource-groups create-group \
    --name my-cr-group \
    --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-
types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

步骤 3：将容量预留添加到容量预留资源组

使用 `resource-groups` 服务和 group-resources 命令将您在步骤 1 中创建的容量预留添加到容量预留资源组。请注意，您必须通过其 ARN 引用按需容量预留。

```
aws resource-groups group-resources \
    --group my-cr-group \
```

```
    --resource-arns \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1 \
      arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

输出示例

```
{
   "Failed": [],
   "Succeeded": [
   "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
   "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
   ]
}
```

（可选）步骤 4：查看资源组中的容量预留

使用 `resource-groups` 服务和 list-group-resources 命令可根据需要描述资源组以查看其容量预留。

```
aws resource-groups list-group-resources --group my-cr-group
```

输出示例

```
{
    "ResourceIdentifiers": [
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1"
        },
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/
cr-54321abcdef567890"
        }
    ]
}
```

步骤 5：创建启动模板，该模板指定容量预留的目标是特定资源组

使用 create-launch-template 命令可创建启动模板，在此启动模板中指定要使用的容量预留。在此示例中，队列将使用已添加到资源组的 `targeted` 容量预留。因此，启动模板数据指定容量预留以特定资源组为目标。在此示例中，启动模板名为 `my-launch-template`。

```
aws ec2 create-launch-template \
    --launch-template-name my-launch-template \
    --launch-template-data \
        '{"ImageId": "ami-0123456789example",
          "CapacityReservationSpecification":
            {"CapacityReservationTarget":
                { "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-1:123456789012:group/my-cr-group" }
            }
        }'
```

（可选）步骤 6：描述启动模板

使用 describe-launch-template-versions 命令（可选）描述启动模板以查看其配置。

```
aws ec2 describe-launch-template-versions --launch-template-name my-launch-template
```

输出示例

```
{
    "LaunchTemplateVersions": [
        {
            "LaunchTemplateId": "lt-01234567890example",
            "LaunchTemplateName": "my-launch-template",
            "VersionNumber": 1,
            "CreateTime": "2021-01-19T20:50:19.000Z",
            "CreatedBy": "arn:aws:iam::123456789012:user/Admin",
            "DefaultVersion": true,
            "LaunchTemplateData": {
                "ImageId": "ami-0947d2ba12ee1ff75",
                "CapacityReservationSpecification": {
                    "CapacityReservationTarget": {
                        "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:us-east-1:123456789012:group/my-cr-group"
                    }
                }
            }
        }
    ]
}
```

步骤 7：创建 EC2 队列

创建一个 EC2 队列，指定将启动的实例配置信息。以下 EC2 队列配置仅显示该示例的相关配置。启动模板 my-launch-template 是您在步骤 5 中创建的启动模板。有两个实例池，每个实例池具有相同的实例类型 (c5.xlarge)，但具有不同的可用区域（us-east-1a 和 us-east-1b)。实例池的价格是相同的，因为定价是针对区域定义的，而不是按可用区定义的。总目标容量为 10，而原定设置目标容量类型为 on-demand。按需分配策略为 lowest-price。容量预留的使用策略是 use-capacity-reservations-first。

> **Note**
>
> 队列类型必须为 instant。其他队列类型不支持 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
             "UsageStrategy": "use-capacity-reservations-first"
         }
```

```
    },
    "Type": "instant"
}
```

在使用上述配置创建 instant 队列后，将启动下面的 10 个实例来满足目标容量：

- 容量预留首先用于启动 6 个按需实例，如下所示：
  - 3 个按需实例启动到 us-east-1a 中的 3 个c5.xlarge targeted容量预留
  - 3 个按需实例启动到 us-east-1b 中的 3 个c5.xlarge targeted容量预留
- 为了满足目标容量，根据按需分配策略，将 4 个额外的按需实例启动到常规按需容量中（在本示例中为 lowest-price）。但是，由于池的价格相同（因为价格是每个区域而不是每个可用区），因此队列将在其中一个池中启动剩余的 4 个按需实例。

（可选）步骤 8：查看剩余未使用的容量预留数

在启动队列后，您可以运行 describe-capacity-reservations 来查看保留的未使用的容量预留的数目。在此示例中，您应看到以下响应，该响应指示所有池中的所有 容量预留 均已使用。

```
{ "CapacityReservationId":  "cr-111",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}

 { "CapacityReservationId":  "cr-222",
     "InstanceType":  "c5.xlarge",
     "AvailableInstanceCount":  0
}
```

教程：将 EC2 实例集配置为将实例启动到容量块中

本教程将指导您完成必须执行的步骤，以便 EC2 实例集在容量块中启动实例。

在大多数情况下，EC2 实例集请求的目标容量应小于或等于您所定位的容量块预留的可用容量。无法满足超过容量块预留限制的目标容量请求。如果目标容量请求超过容量块预留的限制，则对于超出容量块预留限制的容量，您将收到 Insufficient Capacity Exception。

> **Note**
>
> 对于容量块，EC2 实例集将不会回退到为剩余的所需目标容量启动按需型实例。

如果 EC2 实例集无法满足可用容量块预留中请求的目标容量，EC2 Fleet 将尽可能多地满足容量，并返回其能够启动的实例。您可以再次调用 EC2 实例集，直到所有实例都已预调配。

配置 EC2 实例集请求后，您必须等到容量块预留的开始日期。如果您请求 EC2 实例集启动到尚未启动的容量块中，您将收到 `Insufficient Capacity Error`。

在容量块预留变为活动状态后，您可以调用 EC2 实例集 API，并根据您选择的参数将实例预调配到您的容量块中。在容量块中运行的实例将继续运行，直到您手动停止或终止这些实例，或者直到 Amazon EC2 在容量块预留结束时终止这些实例。

有关容量块的更多信息，请参阅 ML 容量块。

注意事项

- 只有类型为 `instant` EC2 实例集请求才支持将实例启动到容量块中。有关更多信息，请参阅 配置 instant 类型的 EC2 实例集。
- 不支持在同一 EC2 实例集请求中有多个容量块。
- 不支持在使用 `OnDemandTargetCapacity` 或 `SpotTargetCapacity` 的同时将 `capacity-block` 设置为 `DefaultTargetCapacity`。
- 如果将 `DefaultTargetCapacityType` 设置为 `capacity-block`，则无法提供 `OnDemandOptions::CapacityReservationOptions`。将会出现异常。

配置 EC2 实例集以将实例启动到容量块中

1. 创建启动模板。

   在启动模板中，执行以下操作：

   - 对于 `InstanceMarketOptionsRequest`，将 `MarketType` 设置为 `capacity-block`。
   - 要确定容量块预留目标，对于 `CapacityReservationID`，指定容量块预留 ID。

   记下启动模板的名称和版本。您在下一步中将使用此信息。

   有关创建启动模板的更多信息，请参阅创建 Amazon EC2 启动模板。
2. 配置 EC2 实例集。

   为您的 EC2 实例集创建一个具有以下配置的文件，文件名为 `config.json`。在以下示例中，将资源标识符替换为您自己的资源标识符。

有关配置 EC2 实例集的更多信息，请参阅创建 EC2 实例集。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "CBR-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "p5.48xlarge",
                    "AvailabilityZone": "us-east-1a"
                },
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "capacity-block"
    },
    "Type": "instant"
}
```

3. 启动实例集。

   使用以下 create-fleet 命令。

   ```
   aws ec2 create-fleet --cli-input-json file://config.json
   ```

   有关更多信息，请参阅 创建 EC2 实例集。

教程：将 EC2 实例集配置为将实例启动到可中断容量预留中

本教程将指导您完成必须执行的步骤，以便 EC2 实例集在可中断容量预留中启动实例。

可中断容量预留是指 Amazon 组织内按需容量预留所有者借给您的备用容量。这些预留适用于可中断的工作负载，因为容量所有者可以随时收回容量，只需发出 2 分钟的中断通知，之后 EC2 就会终止实例。有关容量预留的更多信息，请参阅 可中断容量预留。

EC2 实例集请求的目标容量应小于或等于您所定位的可中断容量预留的可用容量。如果目标容量请求超过您的可中断容量预留的可用容量，则 EC2 实例集会启动尽可能多的实例，并在 API 响应中报告已启动的实例。

在可中断容量预留中运行的实例将继续运行，直到您手动停止或终止它们，或直到容量所有者收回容量。当容量所有者收回容量时，Amazon EC2 会在终止实例前 2 分钟发送一条 Amazon EventBridge 通知。

注意事项

- 只有类型为 `instant` EC2 实例集请求才支持将实例启动到可中断容量预留中。
- 不支持在使用 `OnDemandTargetCapacity` 或 `SpotTargetCapacity` 的同时将 `reserved-capacity` 设置为 `DefaultTargetCapacityType`。
- 当您指定多个启动模板（每个模板都针对不同的可中断容量预留）时，EC2 实例集会为所有匹配的预留配置实例。
- 对于可中断容量预留，EC2 实例集将不会回退到为剩余的所需目标容量启动按需型实例或竞价型实例。

验证权限

在创建 EC2 实例集之前，验证您是否拥有具备所需权限的 IAM 角色。有关更多信息，请参阅 EC2 实例集先决条件。

要将实例启动到可中断容量预留中，您必须执行以下步骤：

步骤 1：创建启动模板

使用 create-launch-template 命令可创建启动模板，指定目标可中断容量预留。在启动模板中，将 `MarketType` 设置为 `interruptible-capacity-reservation`，并指定可中断容量预留的 `CapacityReservationId`。

启动模板配置示例：

```
{
    "LaunchTemplateName": "interruptible-cr-launch-template",
    "LaunchTemplateData": {
        "InstanceType": "m5.large",
        "ImageId": "ami-0abcdef1234567890",
        "CapacityReservationSpecification": {
            "CapacityReservationTarget": {
```

```
                "CapacityReservationId": "cr-0123456789abcdef0"
            }
        },
        "InstanceMarketOptions": {
            "MarketType": "interruptible-capacity-reservation"
        }
    }
}
```

输出示例

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-0123456789example",
        "LaunchTemplateName": "interruptible-cr-launch-template",
        "CreateTime": "2026-03-12T10:00:00.000Z",
        "CreatedBy": "arn:aws:iam::123456789012:user/Admin",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

有关更多信息，请参阅 创建 Amazon EC2 启动模板。

步骤 2：配置 EC2 实例集。

为 EC2 实例集创建一个配置文档，指定启动模板和目标容量。以下配置使用步骤 1 中创建的启动模板 `interruptible-cr-launch-template`。

使用 `reserved-capacity` 作为 `DefaultTargetCapacityType` 时，必须指定 `ReservedCapacityOptions`，并将 `ReservationType` 设置为 `interruptible-capacity-reservation`。

使用以下内容创建名为 `config.json` 的文件：

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "interruptible-cr-launch-template",
                "Version": "1"
            },
```

```
            "Overrides": [
                {
                    "InstanceType": "m5.large",
                    "AvailabilityZone": "us-east-1a"
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "reserved-capacity"
    },
    "ReservedCapacityOptions": {
        "ReservationType": ["interruptible-capacity-reservation"]
    },
    "Type": "instant"
}
```

关键配置参数：

参数	说明
DefaultTargetCapacityType	设置为 reserved-capacity 表示应将实例启动到预留容量。
ReservedCapacityOptions	指定预留容量的类型。对于可中断容量预留，设置 ReservationType 为 ["interruptible-capacity-reservation"] 。
Type	必须设置为 instant。可中断容量预留仅支持实例集。

步骤 3：启动实例集并查看结果

使用 create-fleet 命令创建实例集。

```
aws ec2 create-fleet \
    --cli-input-json file://config.json
```

使用前面的配置创建 instant 实例集后，EC2 实例集会向可中断容量预留启动 10 个实例，以满足目标容量。

> **Note**
>
> 如果实例集无法满足全部目标容量，则响应将包括已启动的实例以及未实现容量的所有错误。

输出示例

```
{
    "FleetId": "fleet-12345678-1234-1234-1234-123456789012",
    "Instances": [
        {
            "LaunchTemplateAndOverrides": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-0123456789example",
                    "Version": "1"
                },
                "Overrides": {
                    "InstanceType": "m5.large",
                    "AvailabilityZone": "us-east-1a"
                }
            },
            "Lifecycle": "interruptible-capacity-reservation",
            "InstanceIds": [
                "i-0123456789example1",
                "i-0123456789example2",
                "i-0123456789example3",
                "i-0123456789example4",
                "i-0123456789example5",
                "i-0123456789example6",
                "i-0123456789example7",
                "i-0123456789example8",
                "i-0123456789example9",
                "i-0123456789example0"
            ],
            "InstanceType": "m5.large",
            "Platform": "Linux/UNIX"
        }
    ],
    "Errors": []
}
```

有关更多信息，请参阅 创建 EC2 实例集。

清理

要停止产生费用，请在不再需要时终止实例。请注意，当容量所有者收回容量时，EC2 还会自动终止在可中断容量预留中启动的实例。

相关资源

- 可中断容量预留
- 使用 EC2 按需容量预留来预留计算容量
- 使用 EC2 实例集
- 创建 EC2 实例集
- 在 Amazon EC2 启动模板中存储实例启动参数

EC2 实例集 CLI 配置示例

要创建实例集，请在 JSON 文件中定义 EC2 实例集配置，然后使用 create-fleet 命令引用该文件，如下所示：

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

以下示例说明了各种 EC2 实例集用例的启动配置。有关配置参数的更多信息，请参阅 create-fleet。

示例

- 示例 1：启动竞价型实例作为默认购买选项
- 示例 2：启动按需型实例作为默认购买选项
- 示例 3：启动按需型实例作为主容量
- 示例 4：使用多个容量保留按需启动实例
- 示例 5：当总目标容量大于未使用的容量预留数量时，使用容量预留启动按需实例
- 示例 6：使用容量预留启动按需实例
- 示例 7：配置容量再平衡以启动替换竞价型实例
- 示例 8：在容量优化的队列中启动竞价型实例
- 示例 9：在具有优先级的容量优化队列中启动竞价型实例
- 示例 10：在价格容量优化的实例集中启动竞价型实例

- 示例 11：配置基于属性的实例类型选择

有关 instant 类型实例集的更多 CLI 示例，请参阅配置 instant 类型的 EC2 实例集。

示例 1：启动竞价型实例作为默认购买选项

下面的示例指定了 EC2 队列中所需的最少参数：启动模板、目标容量和默认购买选项。启动模板由其启动模板 ID 和版本号标识。队列的目标容量为 2 个实例，默认购买选项为 spot，因此队列启动两个 Spot 实例。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

示例 2：启动按需型实例作为默认购买选项

下面的示例指定了 EC2 队列中所需的最少参数：启动模板、目标容量和默认购买选项。启动模板由其启动模板 ID 和版本号标识。队列的目标容量为 2 个实例，默认购买选项为 on-demand，因此队列启动两个 按需实例。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
```

```
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "on-demand"
    }
}
```

示例 3：启动按需型实例作为主容量

下面的示例为队列指定 2 个实例的总目标容量和 1 个按需实例的目标容量。默认购买选项为 spot。队列按照指定的方式启动 1 个按需型实例，但需要再启动一个实例以满足总目标容量要求。差值的购买选项是通过 TotalTargetCapacity – OnDemandTargetCapacity = DefaultTargetCapacityType 计算得出，因而实例集会启动 1 个竞价型实例。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

示例 4：使用多个容量保留按需启动实例

可以通过将 容量预留 的使用策略配置为 use-capacity-reservations-first 来将队列配置为在启动 按需实例 时首先使用 按需容量预留。此示例演示了当容量预留数量超过满足目标容量所需的容量预留数量时，队列如何选择要使用的容量预留。

队列配置如下所示：

- 目标容量：12 个按需实例
- 未使用的容量预留总数：15（超过了队列的 12 个按需实例的目标容量）

- 容量预留池数量：3 (m5.large、m4.xlarge 和 m4.2xlarge)
- 每个池的容量预留数量：5
- 按需分配策略：lowest-price (当多个实例池中存在多个未使用的容量预留时，队列将根据按需分配策略确定启动按需实例的池。)

  请注意，您也可以使用 prioritized 分配策略而不是 lowest-price 分配策略。

容量预留

账户在 3 个不同的池中有以下 15 个未使用的 容量预留。每个池中的 容量预留 数由 AvailableInstanceCount 指示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

队列配置

以下队列配置仅显示该示例的相关配置。总目标容量为 12，而默认目标容量类型为 on-demand。按需分配策略为 lowest-price。容量预留的使用策略是 use-capacity-reservations-first。

在此示例中，个按需型实例 价格为：

- m5.large – 每小时 0.096 美元
- m4.xlarge – 每小时 0.20 美元
- m4.2xlarge – 每小时 0.40 美元

> **Note**
>
> 队列类型必须为类型 instant。其他队列类型不支持 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-abc1234567example",
                "Version": "1"
            },
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]
```

```
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant"
}
```

在使用上述配置创建 instant 队列后，将启动下面的 12 个实例来满足目标容量：

- us-east-1a 中的 5 个 m5.large 按需实例–us-east-1a 中的 m5.large 是最低的价格,并且有 5 个可用的未使用 m5.large 容量预留
- us-east-1a 中的 5 个 m4.xlarge 按需实例 – us-east-1a 中的 m4.xlarge 是第二低的价格，并且有 5 个可用的未使用 m4.xlarge 容量预留
- us-east-1a 中的 2 个 m4.2xlarge 按需实例 – us-east-1a 中的 m4.2xlarge 是第三低的价格，并且有 5 个可用的未使用 m4.2xlarge 容量预留，只需其中的 2 个即可满足目标容量

在启动队列后，您可以运行 describe-capacity-reservations 来查看保留的未使用的 容量预留 的数目。在此示例中，您该看到以下响应，该响应指示已使用所有 m5.large 和 m4.xlarge 容量预留，有 3 个 m4.2xlarge 容量预留未使用。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}
```

```
{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 3
}
```

示例 5：当总目标容量大于未使用的容量预留数量时，使用容量预留启动按需实例

可以通过将 容量预留 的使用策略配置为 use-capacity-reservations-first 来将队列配置为在启动 按需实例 时首先使用 按需容量预留。此示例演示了当总目标容量超过可用未使用容量预留数时，队列如何选择要在其中启动按需实例的实例池。

在此示例中，队列配置如下所示：

- 目标容量：16 个按需实例
- 未使用的容量预留总数：15 (少于队列的 16 个按需实例的目标容量)
- 容量预留池数量：3 (m5.large、m4.xlarge 和 m4.2xlarge)
- 每个池的容量预留数量：5
- 按需分配策略：lowest-price (如果未使用的容量预留数少于按需目标容量，队列将根据按需分配策略确定启动剩余的按需容量的池。)

  请注意，您也可以使用 prioritized 分配策略而不是 lowest-price 分配策略。

容量预留

账户在 3 个不同的池中有以下 15 个未使用的 容量预留。每个池中的 容量预留 数由 AvailableInstanceCount 指示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

```
{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount":5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

队列配置

以下队列配置仅显示该示例的相关配置。总目标容量为 16，而默认目标容量类型为 on-demand。按需分配策略为 lowest-price。容量预留的使用策略是 use-capacity-reservations-first。

在此示例中，个按需型实例 价格为：

- m5.large – 每小时 0.096 美元
- m4.xlarge – 每小时 0.20 美元
- m4.2xlarge – 每小时 0.40 美元

> **Note**
>
> 队列类型必须为 instant。其他队列类型不支持 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
```

```
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            },
            "Overrides": [
                {
                  "InstanceType": "m5.large",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                },
                {
                  "InstanceType": "m4.2xlarge",
                  "AvailabilityZone": "us-east-1a",
                  "WeightedCapacity": 1
                }
              ]

        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

在使用上述配置创建 instant 队列后，将启动下面的 16 个实例来满足目标容量：

- us-east-1a 中的 6个 m5.large 按需实例–us-east-1a 中的 m5.large 是最低的价格，并且有 5 个可用的未使用 m5.large 容量预留。容量预留首先用于启动 5 个按需实例。使用了剩余 m4.xlarge 和 m4.2xlarge 容量保留后，为了满足目标容量，将额外的按需实例启动到常规按需容量中（在本示例中为 lowest-price）。

- us-east-1a 中的 5个 m4.xlarge 按需实例–us-east-1a 中的 m4.xlarge 是第二低的价格，并且有 5 个可用的未使用 m4.xlarge 容量预留。
- us-east-1a 中的 5个 m4.2xlarge 按需实例–us-east-1a 中的 m4.2xlarge 是第三低的价格，并且有 5 个可用的未使用 m4.2xlarge 容量预留。

在启动队列后，您可以运行 describe-capacity-reservations 来查看保留的未使用的 容量预留 的数目。在此示例中，您应看到以下响应，该响应指示所有池中的所有 容量预留 均已使用。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 0
}
```

示例 6：使用容量预留启动按需实例

通过将容量保留的使用策略设置为 use-capacity-reservations-first，您可以将震源组配置为在启动按需实例时首先使用 targeted 按需容量保留。此示例演示如何将按需实例启动到 targeted 容量预留，其中容量预留的属性相同，但其可用区域除外 (us-east-1a 和 us-east-1b)。它还演示了当总目标容量超过可用未使用容量预留数时，队列如何选择要在其中启动按需实例的实例池。

在此示例中，队列配置如下所示：

- 目标容量：10 个按需实例
- 未使用总数 targeted 容量预留：6（少于队列的 10 个按需实例的按需目标容量）
- 容量预留池数量：2 (us-east-1a 和 us-east-1b)
- 每个池的容量预留数量：3

- 按需分配策略：lowest-price（如果未使用的容量预留数少于按需目标容量，队列将根据按需分配策略确定启动剩余的按需容量的池。）

  请注意，您也可以使用 prioritized 分配策略而不是 lowest-price 分配策略。

有关完成此示例所必须执行的过程的演练，请参阅 教程：使用目标容量预留配置 EC2 实例集以启动按需型实例。

容量预留

账户在 2 个不同的池中有以下 6 个未使用的容量预留。在此示例中，池因其可用区而异。每个池中的容量预留 数由 AvailableInstanceCount 指示。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1b",
    "AvailableInstanceCount": 3,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

队列配置

以下队列配置仅显示该示例的相关配置。总目标容量为 10，而默认目标容量类型为 on-demand。按需分配策略为 lowest-price。容量预留的使用策略是 use-capacity-reservations-first。

在此示例中，us-east-1 中 c5.xlarge 的按需实例价格为：每小时 0.17 美元

> **Note**
>
> 队列类型必须为 instant。其他队列类型不支持 use-capacity-reservations-first。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
               {
                   "InstanceType": "c5.xlarge",
                   "AvailabilityZone": "us-east-1a"
               },
               {
                    "InstanceType": "c5.xlarge",
                    "AvailabilityZone": "us-east-1b"
               }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 10,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
             "UsageStrategy": "use-capacity-reservations-first"
         }
    },
    "Type": "instant"
}
```

在使用上述配置创建 instant 队列后，将启动下面的 10 个实例来满足目标容量：

- 容量预留首先用于启动 6 个按需实例，如下所示：
  - 3 个按需实例启动到 us-east-1a 中的 3 个c5.xlarge targeted容量预留

- 3 个按需实例启动到 us-east-1b 中的 3 个c5.xlarge targeted容量预留
- 为了满足目标容量，根据按需分配策略，将 4 个额外的按需实例启动到常规按需容量中（在本示例中为 lowest-price）。但是，由于池的价格相同（因为价格是每个区域而不是每个可用区），因此队列将在一个池中启动剩余的 4 个按需实例。

在启动队列后，您可以运行 describe-capacity-reservations 来查看保留的未使用的 容量预留 的数目。在此示例中，您应看到以下响应，该响应指示所有池中的所有 容量预留 均已使用。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c5.xlarge",
    "AvailableInstanceCount": 0
}
```

示例 7：配置容量再平衡以启动替换竞价型实例

下面的示例将 EC2 实例集配置为在 Amazon EC2 为队列中的竞价型实例发出再平衡建议时启动替换竞价型实例。要配置 Spot 实例 的自动替换，对于 ReplacementStrategy，请指定 launch-before-terminate。要配置从启动新替换竞价型实例到自动删除旧竞价型实例的时间延迟，请对于 termination-delay，请指定值（以秒为单位）。有关更多信息，请参阅 配置选项。

> **Note**
>
> 我们建议仅当您可以预测实例关闭过程完成所需的时长时才使用 launch-before-terminate，这样只有在这些过程完成后才会终止旧实例。在实例运行期间，您需要为它们付费。

容量再平衡策略的有效性取决于 EC2 队列 请求中指定的 Spot 容量池的数量。我们建议您使用一组多样化的实例类型和可用区配置队列，对于 AllocationStrategy，请指定 capacity-optimized。有关为容量再平衡配置 EC2 队列 时应考虑的内容的更多信息，请参阅在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例。

```
{
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "LaunchTemplate",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "c3.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c4.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c5.large",
                           "WeightedCapacity": 1,
                            "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       }
                ]
          }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
```

```
            }
        }
    }
}
```

示例 8：在容量优化的队列中启动竞价型实例

以下示例演示如何使用针对容量进行优化的 Spot 分配策略配置 EC2 队列。要优化容量，您必须将 `AllocationStrategy` 设置为 `capacity-optimized`。

在以下示例中，三个启动规范指定了三个 Spot 容量池。目标容量为 50 个竞价型实例。EC2 实例集尝试在 Spot 容量池中启动 50 个竞价型实例，并为启动的实例数量提供最佳容量。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "r4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2a"
                           },
                      },
                       {
                           "InstanceType": "m4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           },
                       },
                       {
                           "InstanceType": "c5.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           }
                       }
                  ]
```

```
            }
    ],
    "TargetCapacitySpecification": {
            "TotalTargetCapacity": 50,
            "DefaultTargetCapacityType": "spot"

    }
}
```

示例 9：在具有优先级的容量优化队列中启动竞价型实例

以下示例演示如何通过 Spot 分配策略配置 EC2 队列，该策略针对容量进行优化，同时尽最大努力使用优先级。

使用 capacity-optimized-prioritized 分配策略时，您可以使用 Priority 参数指定 Spot 容量池的优先级，数字越小优先级越高。如果您平等地支持多个 Spot 容量池，您还可以为它们设置相同的优先级。如果您没有为某个池设置优先级，则该池将在优先级方面被视为最后一个。

要设置 Spot 容量池的优先级，您必须将 AllocationStrategy 设置为 capacity-optimized-prioritized。EC2 队列首先会针对容量进行优化，但会尽最大努力遵循优先级。（例如，如果遵循优先级不会显著影响 EC2 队列预置最佳容量的能力）。对于必须最大限度地减少中断可能性，同时对某些实例类型的偏好也很重要的工作负载来说，这是一个不错的选择。

在以下示例中，三个启动规范指定了三个 Spot 容量池。对每个池进行优先级排序，数字越小优先级越高。目标容量为 50 个竞价型实例。EC2 实例集尝试以最高优先级在 Spot 容量池中启动 50 个竞价型实例，但首先针对容量进行优化。

```
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized-prioritized"
        },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                        {
                           "InstanceType": "r4.2xlarge",
                           "Priority": 1,
                           "Placement": {
```

```
                            "AvailabilityZone": "us-west-2a"
                        },
                    },
                     {
                        "InstanceType": "m4.2xlarge",
                        "Priority": 2,
                        "Placement": {
                            "AvailabilityZone": "us-west-2b"
                        },
                     },
                     {
                        "InstanceType": "c5.2xlarge",
                        "Priority": 3,
                        "Placement": {
                            "AvailabilityZone": "us-west-2b"
                        }
                     }
                ]
            }
    ],
    "TargetCapacitySpecification": {
            "TotalTargetCapacity": 50,
            "DefaultTargetCapacityType": "spot"
}
```

示例 10：在价格容量优化的实例集中启动竞价型实例

以下示例演示了如何通过 Spot 分配策略配置 EC2 实例集，该策略针对容量和最低价格进行了优化。若要在考虑价格的同时优化容量，必须将 Spot `AllocationStrategy` 设置为 `price-capacity-optimized`。

在以下示例中，三个启动规范指定了三个 Spot 容量池。目标容量为 50 个竞价型实例。EC2 实例集尝试在 Spot 容量池中启动 50 个竞价型实例，并为启动的实例数量提供最佳容量，同时选择最低价格的池。

```
{
    "SpotOptions": {
        "AllocationStrategy": "price-capacity-optimized",
        "MinTargetCapacity": 2,
        "SingleInstanceType": true
    },
    "OnDemandOptions": {
```

```
        "AllocationStrategy": "lowest-price"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
                 "Overrides": [
                       {
                           "InstanceType": "r4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2a"
                           },
                      },
                       {
                           "InstanceType": "m4.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           },
                       },
                       {
                           "InstanceType": "c5.2xlarge",
                           "Placement": {
                               "AvailabilityZone": "us-west-2b"
                           }
                       }
                ]
           }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 50,
        "OnDemandTargetCapacity":0,
        "SpotTargetCapacity":50,
        "DefaultTargetCapacityType": "spot"
    },
    "Type": "instant"
}
```

示例 11：配置基于属性的实例类型选择

以下示例演示了如何配置 EC2 实例集，以使用基于属性的实例类型选择来识别实例类型。若要指定所需的实例属性，请在 InstanceRequirements 结构中指定属性。

在以下示例中，指定了 2 个实例属性：

- VCpuCount – 至少指定了 2 个 vCPU。由于未指定最大值，因此没有最大限制。
- MemoryMiB – 至少指定了 4 MiB 的内存。由于未指定最大值，因此没有最大限制。

将识别具有 2 个或更多 vCPU 和 4 MiB 或更大内存的任何实例类型。然而，当 EC2 实例集预置实例集时，价格保护和分配策略可能会排除某些实例类型。

有关您可以指定的所有可能属性的列表和描述，请参阅 Amazon EC2 API Reference（《Amazon EC2 API 参考》）中的 InstanceRequirements。

```
{
 "SpotOptions": {
  "AllocationStrategy": "price-capacity-optimized"
 },
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }],
 "TargetCapacitySpecification": {
  "TotalTargetCapacity": 20,
  "DefaultTargetCapacityType": "spot"
 },
 "Type": "instant"
}
```

竞价型实例集 CLI 配置示例

要创建实例集，请在 JSON 文件中定义竞价型实例集配置，然后使用 request-spot-fleet Amazon CLI 命令引用该文件，如下所示：

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://file_name.json
```

以下示例说明了各种竞价型实例集用例的启动配置。有关配置参数的更多信息，请参阅 request-spot-fleet。有关创建竞价型实例集的更多信息，请参阅创建竞价型实例集。

> **Note**
>
> 对于竞价型实例集，您无法在启动模板或启动规范中指定网络接口 ID。请务必要忽略启动模板或启动规范中的 `NetworkInterfaceID` 参数。

示例

- 示例 1：使用区域中价格最低的可用区启动竞价型实例
- 示例 2：使用列表中价格最低的可用区启动竞价型实例
- 示例 3：使用列表中价格最低的实例类型启动竞价型实例
- 示例 4。覆盖请求的价格
- 示例 5：使用多样化分配策略启动 Spot 队列
- 示例 6：使用实例权重启动 Spot 队列
- 示例 7：使用按需容量启动竞价型实例集
- 示例 8：配置容量再平衡以启动替换竞价型实例
- 示例 9：在容量优化的队列中启动竞价型实例
- 示例 10：在具有优先级的容量优化队列中启动竞价型实例
- 示例 11：在 priceCapacityOptimized 实例集中启动竞价型实例
- 示例 12：配置基于属性的实例类型选择

示例 1：使用区域中价格最低的可用区启动竞价型实例

以下示例指定一个没有可用区或子网的启动规范。Spot 队列会在具有默认子网且价格最低的可用区中启动实例。您支付的价格不得超过按需价格。

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

示例 2：使用列表中价格最低的可用区启动竞价型实例

以下示例指定具有的可用区或子网不同但实例类型和 AMI 相同的两种启动规范。

可用区

Spot 队列会在价格最低的指定可用区的默认子网中启动实例。

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "Placement": {
              "AvailabilityZone": "us-west-2a, us-west-2b"
```

```
            },
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            }
        }
    ]
}
```

子网

您可以指定默认子网或非默认子网，并且非默认子网可来自默认 VPC 或非默认 VPC。Spot 服务会在位于价格最低的可用区的子网中启动实例。

您无法在 Spot 队列请求中指定来自相同可用区的不同子网。

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "KeyName": "my-key-pair",
          "SecurityGroups": [
              {
                  "GroupId": "sg-1a2b3c4d"
              }
          ],
          "InstanceType": "m3.medium",
          "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
          "IamInstanceProfile": {
              "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
          }
      }
  ]
}
```

如果在默认 VPC 中启动实例，则实例在默认情况下会收到一个公有 IPv4 地址。如果在非默认 VPC 中启动实例，则实例在默认情况下不会收到一个公有 IPv4 地址。在启动规范中使用网络接口来将一个公有 IPv4 地址分配给在非默认 VPC 中启动的实例。指定网络接口时，您必须包括使用网络接口的子网 ID 和安全组 ID。

```
  ...
```

```
    {
        "ImageId": "ami-1a2b3c4d",
        "KeyName": "my-key-pair",
        "InstanceType": "m3.medium",
        "NetworkInterfaces": [
            {
                "DeviceIndex": 0,
                "SubnetId": "subnet-1a2b3c4d",
                "Groups": [ "sg-1a2b3c4d" ],
                "AssociatePublicIpAddress": true
            }
        ],
        "IamInstanceProfile": {
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        }
    }
...
```

示例 3：使用列表中价格最低的实例类型启动竞价型实例

以下示例指定实例类型不同、但 AMI 和可用区或子网相同的两种启动配置。Spot 队列使用价格最低的指定实例类型启动实例。

可用区

```
{
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
    {
        "ImageId": "ami-1a2b3c4d",
        "SecurityGroups": [
            {
                "GroupId": "sg-1a2b3c4d"
            }
        ],
        "InstanceType": "c5.4xlarge",
        "Placement": {
          "AvailabilityZone": "us-west-2b"
        }
    },
    {
        "ImageId": "ami-1a2b3c4d",
```

```
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

子网

```
{
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "c5.4xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}
```

示例 4。覆盖请求的价格

我们建议您使用默认最高价 (这是按需价格)。如果愿意，您可以为队列请求以及各个启动规范指定最高价。

以下示例为队列请求以及两个启动规范 (共三个) 指定最高价。队列请求的最高价用于未指定最高价的任何启动规范。Spot 队列使用价格最低的实例类型启动实例。

可用区

```
{
  "SpotPrice": "1.00",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
          "SpotPrice": "0.10"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.4xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
          "SpotPrice": "0.20"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.8xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      }
    ]
}
```

子网

```
{
  "SpotPrice": "1.00",
  "TargetCapacity": 30,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "SpotPrice": "0.10"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.4xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "SpotPrice": "0.20"
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.8xlarge",
          "SubnetId": "subnet-1a2b3c4d"
      }
  ]
}
```

示例 5：使用多样化分配策略启动 Spot 队列

以下示例使用 `diversified` 分配策略。启动规范具有不同的实例类型，但具有相同的 AMI 和可用区或子网。Spot 队列在 3 个启动规范间分配 30 个实例，以便每种类型有 10 个实例。有关更多信息，请参阅 使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量。

可用区

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
```

```
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

子网

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
```

```
        }
    ]
}
```

在其中一个可用区中断的情况下，增加 EC2 容量可以满足 Spot 请求的几率的最佳实践是跨区域实现多样化。对于这种情况，请在启动规范中包含每个对您可用的可用区。并且，不是每次使用同一个子网，而是使用三个唯一的子网（每个子网映射到不同的区域）。

可用区

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c4.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2a"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "m3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          }
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2c"
          }
      }
  ]
}
```

子网

```
{
```

```
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-2a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-3a2b3c4d"
        }
    ]
}
```

示例 6：使用实例权重启动 Spot 队列

以下示例使用实例权重，这意味着价格是每单位小时价格，而不是每实例小时价格。每个启动配置列出不同的实例类型和不同的权重。Spot 队列选择每单位小时价格最低的实例类型。竞价型实例集通过将目标容量除以实例权重，计算出要启动的竞价型实例数。如果结果不是整数，则 Spot 队列会将其向上舍入到下一个整数，以便队列的大小不低于其目标容量。

如果 `r3.2xlarge` 请求成功，Spot 将预置其中的 4 个实例。将 20 除以 6 可得到总共 3.33 个实例，然后向上舍入为 4 个实例。

如果 `c3.xlarge` 请求成功，Spot 将预置其中的 7 个实例。将 20 除以 3 可得到总共 6.66 个实例，然后向上舍入为 7 个实例。

有关更多信息，请参阅 使用实例权重来管理 EC2 实例集或竞价型实例集的成本和性能。

可用区

```
{
  "SpotPrice": "0.70",
```

```
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
          "WeightedCapacity": 6
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.xlarge",
          "Placement": {
              "AvailabilityZone": "us-west-2b"
          },
          "WeightedCapacity": 3
      }
    ]
}
```

子网

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 20,
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "LaunchSpecifications": [
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "r3.2xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 6
      },
      {
          "ImageId": "ami-1a2b3c4d",
          "InstanceType": "c3.xlarge",
          "SubnetId": "subnet-1a2b3c4d",
          "WeightedCapacity": 3
      }
  ]
}
```

示例 7：使用按需容量启动竞价型实例集

为确保始终拥有实例容量，您可以在 Spot 队列请求中包含按需容量请求。如果具有容量，则将始终执行按需请求。目标容量的余量将在具有容量且可用的情况下作为 Spot 容量执行。

以下示例将所需的目标容量指定为 10，其中 5 个必须为按需容量。未指定 Spot 容量；它由目标容量减去按需容量的余量隐含指定。Amazon EC2 启动 5 个容量单位作为按需容量，并在有可用的 Amazon EC2 容量并且可用时，启动 5 个容量单位 (10-5=5) 作为 Spot 容量。

```
{
  "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
  "AllocationStrategy": "lowestPrice",
  "TargetCapacity": 10,
  "SpotPrice": null,
  "ValidFrom": "2018-04-04T15:58:13Z",
  "ValidUntil": "2019-04-04T15:58:13Z",
  "TerminateInstancesWithExpiration": true,
  "LaunchSpecifications": [],
  "Type": "maintain",
  "OnDemandTargetCapacity": 5,
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",
        "Version": "2"
      },
      "Overrides": [
        {
          "InstanceType": "t2.medium",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-d0dc51fb"
        }
      ]
    }
  ]
}
```

示例 8：配置容量再平衡以启动替换竞价型实例

下面的示例将竞价型实例集配置为在 Amazon EC2 为队列中的竞价型实例发出再平衡建议时启动替换竞价型实例。要配置 Spot 实例 的自动替换，对于 `ReplacementStrategy`，请指定 `launch-`

before-terminate。要配置从启动新替换竞价型实例到自动删除旧竞价型实例的时间延迟，请对于 termination-delay，请指定值（以秒为单位）。有关更多信息，请参阅 配置选项。

> **Note**
>
> 我们建议仅当您可以预测实例关闭过程完成所需的时长时才使用 launch-before-terminate。这将确保只有在关闭过程完成后才终止旧实例。在实例运行期间，您需要为它们付费。

容量再平衡策略的有效性取决于 Spot 队列请求中指定的 Spot 容量池的数量。我们建议您使用一组多样化的实例类型和可用区配置队列，对于 AllocationStrategy，请指定 capacityOptimized。有关为容量再平衡配置竞价型实例集时应考虑的内容的更多信息，请参阅 在 EC2 实例集和竞价型实例集中使用“容量再平衡”功能来替换存在风险的竞价型实例。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
        "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateName": "LaunchTemplate",
                    "Version": "1"
                },
                 "Overrides": [
                       {
                           "InstanceType": "c3.large",
                           "WeightedCapacity": 1,
                           "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
                           "InstanceType": "c4.large",
                           "WeightedCapacity": 1,
                           "Placement": {
                               "AvailabilityZone": "us-east-1a"
                           }
                       },
                       {
```

```
                            "InstanceType": "c5.large",
                            "WeightedCapacity": 1,
                            "Placement": {
                                "AvailabilityZone": "us-east-1a"
                            }
                        }
                ]
            }
    ],
        "TargetCapacity": 5,
        "SpotMaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch-before-terminate",
                "TerminationDelay": "720"
            }
        }
    }
}
```

示例 9：在容量优化的队列中启动竞价型实例

以下示例演示如何使用针对容量进行优化的 Spot 分配策略配置 Spot 队列。要优化容量，您必须将 `AllocationStrategy` 设置为 `capacityOptimized`。

在以下示例中，三个启动规范指定了三个 Spot 容量池。目标容量为 50 个竞价型实例。竞价型实例集尝试在 Spot 容量池中启动 50 个竞价型实例，并为启动的实例数量提供最佳容量。

```
{
    "TargetCapacity": "50",
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                    "AvailabilityZone": "us-west-2a"
                },
```

```
                {
                    "InstanceType": "m4.2xlarge",
                    "AvailabilityZone": "us-west-2b"
                },
                {
                    "InstanceType": "c5.2xlarge",
                    "AvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

示例 10：在具有优先级的容量优化队列中启动竞价型实例

以下示例演示如何通过 Spot 分配策略配置 Spot 队列，该策略针对容量进行优化，同时尽最大努力使用优先级。

使用 `capacityOptimizedPrioritized` 分配策略时，您可以使用 `Priority` 参数指定 Spot 容量池的优先级，数字越小优先级越高。如果您平等地支持多个 Spot 容量池，您还可以为它们设置相同的优先级。如果您没有为某个池设置优先级，则该池将在优先级方面被视为最后一个。

要设置 Spot 容量池的优先级，您必须将 `AllocationStrategy` 设置为 `capacityOptimizedPrioritized`。Spot 队列首先会针对容量进行优化，但会尽最大努力遵循优先级。（例如，如果遵循优先级不会显著影响 Spot 队列预置最佳容量的能力）。对于必须最大限度地减少中断可能性，同时对某些实例类型的偏好也很重要的工作负载来说，这是一个不错的选择。

在以下示例中，三个启动规范指定了三个 Spot 容量池。对每个池进行优先级排序，数字越小优先级越高。目标容量为 50 个竞价型实例。竞价型实例集尝试以最高优先级在 Spot 容量池中启动 50 个竞价型实例，但首先针对容量进行优化。

```
{
    "TargetCapacity": "50",
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimizedPrioritized"
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
```

```
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                   "Priority": 1,
                    "AvailabilityZone": "us-west-2a"
                },
                {
                           "InstanceType": "m4.2xlarge",
                           "Priority": 2,
                           "AvailabilityZone": "us-west-2b"
                },
                {
                           "InstanceType": "c5.2xlarge",
                           "Priority": 3,
                           "AvailabilityZone": "us-west-2b"
                }
            ]
        }
    ]
}
```

示例 11：在 priceCapacityOptimized 实例集中启动竞价型实例

以下示例演示了如何通过 Spot 分配策略配置竞价型实例集，该策略针对容量和最低价格进行了优化。若要在考虑价格的同时优化容量，必须将 Spot `AllocationStrategy` 设置为 `priceCapacityOptimized`。

在以下示例中，三个启动规范指定了三个 Spot 容量池。目标容量为 50 个竞价型实例。竞价型实例集尝试在 Spot 容量池中启动 50 个竞价型实例，并为启动的实例数量提供最佳容量，同时选择最低价格的池。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "priceCapacityOptimized",
        "OnDemandAllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111111111111:role/aws-ec2-spot-fleet-tagging-
role",
        "LaunchTemplateConfigs": [
            {
                "LaunchTemplateSpecification": {
```

```
                    "LaunchTemplateId": "lt-0123456789example",
                    "Version": "1"
                },
                "Overrides": [
                     {
                          "InstanceType": "r4.2xlarge",
                          "AvailabilityZone": "us-west-2a"
                     },
                     {
                          "InstanceType": "m4.2xlarge",
                          "AvailabilityZone": "us-west-2b"
                     },
                     {
                          "InstanceType": "c5.2xlarge",
                          "AvailabilityZone": "us-west-2b"
                     }
                ]
            }
        ],
        "TargetCapacity": 50,
        "Type": "request"
    }
}
```

示例 12：配置基于属性的实例类型选择

以下示例演示了如何配置竞价型实例集，以使用基于属性的实例类型选择来识别实例类型。若要指定所需的实例属性，请在 InstanceRequirements 结构中指定属性。

在以下示例中，指定了 2 个实例属性：

- VCpuCount – 至少指定了 2 个 vCPU。由于未指定最大值，因此没有最大限制。
- MemoryMiB – 至少指定了 4 MiB 的内存。由于未指定最大值，因此没有最大限制。

将识别具有 2 个或更多 vCPU 和 4 MiB 或更大内存的任何实例类型。然而，当竞价型实例集预置实例集时，价格保护和分配策略可能会排除某些实例类型。

有关您可以指定的所有可能属性的列表和描述，请参阅 Amazon EC2 API Reference（《Amazon EC2 API 参考》）中的 InstanceRequirements。

```
{
```

```
 "AllocationStrategy": "priceCapacityOptimized",
 "TargetCapacity": 20,
 "Type": "request",
 "LaunchTemplateConfigs": [{
  "LaunchTemplateSpecification": {
   "LaunchTemplateName": "my-launch-template",
   "Version": "1"
  },
  "Overrides": [{
   "InstanceRequirements": {
    "VCpuCount": {
     "Min": 2
    },
    "MemoryMiB": {
     "Min": 4
    }
   }
  }]
 }]
}
```

EC2 实例集和竞价型实例集的限额

对于每项 Amazon 服务，您的 Amazon Web Services 账户都具有默认配额（以前称为限制）。除非另有说明，否则，每个配额是区域特定的。

常用的 Amazon EC2 配额适用于由 EC2 队列或 Spot 队列启动的实例，例如 Spot 实例限制和卷限制。

此外，Amazon Web Services 账户 存在以下与 EC2 实例集和竞价型实例集相关的限额：

配额描述	配额
每个区域中状态为 `active`、`deleted_running` 和 `cancelled_running` 的 `maintain` 和 `request` 类型 EC2 实例集和竞价型实例集的数量	1,000 [1] [2] [3]
`instant` 类型 EC2 实例集的数量	无限制

配额描述	配额
`maintain` 和 `request` 类型 EC2 实例集和竞价型实例集的竞价型容量池数量（不重复的实例类型和子网组合）	300[1]
`instant` 类型 EC2 实例集的竞价型容量池数量（不重复的实例类型和子网组合）	无限制
启动规范中的用户数据大小	16 KB [2]
每个 EC2 实例集或竞价型实例集的目标容量	10000
区域中所有 EC2 实例集和竞价型实例集的目标容量	100,000 [1]
EC2 队列请求或 Spot 队列请求不能跨区域。	
EC2 队列请求或 Spot 队列请求不能跨同一可用区内的不同子网。	

[1] 这些限制同时适用于您的 EC2 实例集和竞价型实例集。

[2] 这些是硬配额。您不能请求提高这些限额。

[3] 删除 EC2 实例集或取消竞价型实例集请求之后，如果您指定在您删除或取消请求时该实例集不应终止其竞价型实例，实例集请求将输入 `deleted_running`（EC2 实例集）或 `cancelled_running`（竞价型实例集）状态，并且实例将继续运行直至它们中断或您手动将其终止。如果终止这些实例，实例集请求将输入 `deleted_terminating`（EC2 实例集）或 `cancelled_terminating`（竞价型实例集）状态，并且不计入此配额。有关更多信息，请参阅删除 EC2 实例集请求和实例集中的实例和取消（删除）竞价型实例集请求。

请求增加目标容量配额

如果您需要的容量超出目标容量的默认配额，您可以在此请求增加配额。

请求增加目标容量配额

1. 打开 Amazon Web Services 支持 Center 的 Create case（创建案例）表。
2. 选择提高服务限制。

3. 对于 Limit type（限制类型），选择 EC2 Fleet（EC2 实例集）。
4. 对于 Region（区域），选择请求增加配额的 Amazon 区域。
5. 对于 Limit（限制），选择 Target Fleet Capacity per Fleet (in units)（每个实例集的目标实例集容量（单位））或 Target Fleet Capacity per Region (in units)（每个区域的目标实例集容量（单位）），具体取决于您要增加的配额。
6. 对于 New limit value（新增限制值），输入新的配额值。
7. 要请求增加其他配额，选择 Add another request（添加另一个请求），然后重复步骤 4-6。
8. 对于 Use case description（使用案例描述），输入请求增加配额的原因。
9. 在 Contact options（联系选项）下，指定您首选的联系语言和联系方式。
10. 选择提交。

Amazon EC2 中的联网功能

通过 Amazon VPC，您可以在专供您的Amazon账户使用的虚拟网络（称为 Virtual Private Cloud (VPC)）中启动Amazon资源，如 Amazon EC2 实例。启动实例时，您可以从 VPC 中选择子网。该实例配置了主网络接口，即逻辑虚拟网卡。该实例从子网的 IPv4 地址接收主要私有 IP 地址，并将其分配到主网络接口。

您可以控制实例是否从 Amazon 的公有 IP 地址池接收公有 IP 地址。实例的公有 IP 地址会与您的实例关联，直至实例停止或终止。如果您需要持久性公有 IP 地址，可以为 Amazon 账户分配弹性 IP 地址，并将其与实例或网络接口关联。在您释放弹性 IP 地址之前，该地址仍然与您的 Amazon 账户关联，您可以根据需要将其从一个实例移动到另一个实例。您可以在 Amazon 账户中使用您自己的 IP 地址，它会显示为地址池，然后从地址池中分配弹性 IP 地址。

要提高网络性能并减少延迟，您可以在置放群组中启动实例。您可以使用增强型联网功能显著地提高每秒数据包数 (PPS) 性能。您可以使用 Elastic Fabric Adapter (EFA) 加速高性能计算和机器学习应用程序，这是一种可以附加到支持的实例类型的网络设备。

功能

- 区域和可用区
- Amazon EC2 实例 IP 寻址
- EC2 实例主机名和域
- 自带 IP 地址（BYOIP）到 Amazon EC2
- 弹性 IP 地址
- 弹性网络接口
- Amazon EC2 实例网络带宽
- Amazon EC2 实例上的增强联网功能
- 适用于 Amazon EC2 上 AI/ML 和 HPC 工作负载的 Elastic Fabric Adapter
- Amazon EC2 拓扑
- Amazon EC2 实例的置放群组
- EC2 实例的网络最大传输单位 (MTU)
- 适用于您的 EC2 实例的虚拟私有云
- 辅助网络

区域和可用区

Amazon EC2 托管在全球多个位置。这些位置由 Amazon Web Services 区域、可用区、Local Zones、Amazon Outposts 和 Wavelength Zones 组成。

- 区域是独立的地理区域。
- 可用区是每个区域内的多个相互隔离的位置。
- Local Zones 让您可以在多个距离最终用户较近的位置放置资源（如计算和存储）。
- Wavelength Zones 让您能够为 5G 设备和最终用户打造具有超低延迟的应用程序。Wavelength 可以将标准 Amazon 计算和存储服务部署到电信运营商的 5G 网络边缘。
- Amazon Outposts 让您能够几乎在任何数据中心、主机托管空间或本地设施使用原生的 Amazon 服务、基础设施和运营模式。

Amazon运行着具有高可用性的先进数据中心。数据中心有时会发生影响托管于同一位置的所有实例的可用性的故障，虽然这种故障极少发生。如果您将所有实例都托管在受故障影响的同一个位置，则您的所有实例都将不可用。

如需更多信息，请参阅 Amazon 全球基础设施。

内容

- Regions
- 可用区
- Local Zones
- Wavelength 区域
- Amazon Outposts

Regions

从设计而言，每个 区域都与其他 区域隔离。这可实现最大程度的容错能力和稳定性。

当您启动实例时，请选择能让您的实例更靠近特定客户的区域，或选择能够满足法律或其他要求的区域。您可以在多个区域中启动实例。

当您查看资源时，只会看到与您指定的 区域关联的资源。这是因为 区域间彼此隔离，而且我们不会自动跨 区域复制资源。

可用区

有关可用区域的列表，请参阅 Amazon 区域。

Amazon EC2 区域性端点

当您通过命令行界面或 API 操作使用实例时，必须指定其区域端点。有关 Amazon EC2 区域和端点的更多信息，请参阅《Amazon EC2 Developer Guide》中的 Amazon EC2 service endpoints。

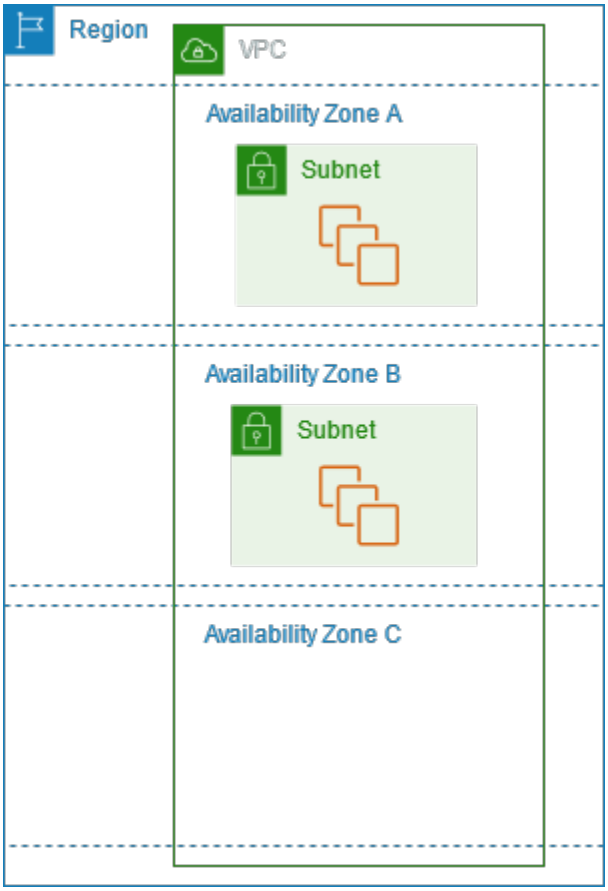
有关更多信息，请参阅《Amazon Web Services 区域 和可用区用户指南》 中的 Amazon Web Services 区域。

可用区

每个区域都有多个相互隔离的位置，称为可用区。可用区的代码由其区域代码后跟一个字母标识符组成。例如 `us-east-1a`。

通过在多个可用区启动 EC2 实例，可以确保应用程序不受区域中单一位置故障的影响。

下图说明 Amazon 区域中的多个可用区。可用区 A 和可用区 B 各有一个子网，每个子网均有 EC2 实例。可用区 C 没有子网，因此您无法在此可用区内启动实例。



有关更多信息，请参阅 适用于您的 EC2 实例的虚拟私有云。

各区域的可用区

有关各区域的可用区列表，请参阅 Amazon 可用区。

可用区中的实例

启动实例时，您需要选择区域和虚拟私有云（VPC）。然后，您可以选择其中一个可用区内的子网，也可以让我们为您选择子网。启动初始实例时，建议让我们根据系统运行状况和可用容量为您选择可用区。如果要启动其他实例，请仅在新实例必须靠近现有实例或必须与现有实例相隔离时指定可用区。

如果实例跨多个可用区分布且其中的某个实例发生故障，则应用程序的设计应确保可以使用另一可用区中的实例可代为处理相关请求。

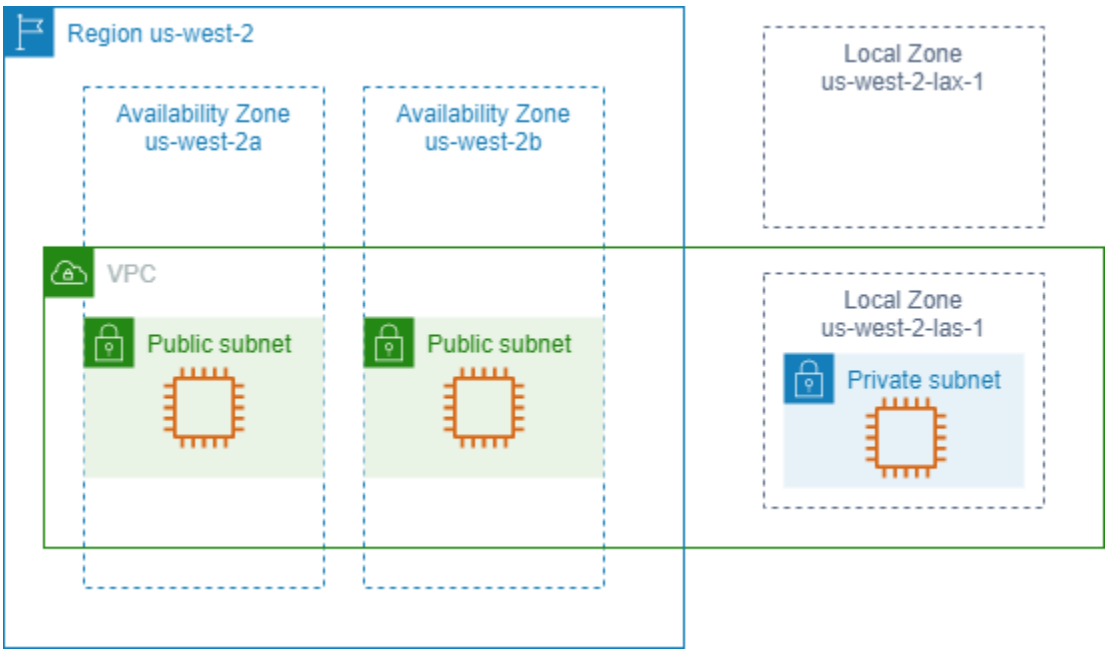
有关更多信息，请参阅《Amazon Web Services 区域 用户指南》中的 Amazon 可用区。

Local Zones

本地区域是在地理上靠近您的用户的 Amazon 区域的扩展。Local Zones 有自己的互联网连接并支持 Amazon Direct Connect，以便在本地区域中创建的资源可以通过低延迟通信为本地用户提供服务。有关更多信息，请参阅《Amazon Local Zones 用户指南》中的 What is Amazon Local Zones?。

本地区域的代码由其区域代码后跟一个指示其实际位置的标识符组成。例如，在洛杉矶的 `us-west-2-lax-1`。

下图说明了 Amazon 区域 `us-west-2`、其两个可用区及其两个本地区域。VPC 跨多个可用区和其中一个本地区域。VPC 中的每个区域都有一个子网，每个子网都有一个实例。



可用 Local Zones

有关可用本地区的列表，请参阅《Amazon Local Zones User Guide》中的 Available Local Zones。有关已公布的本地区列表，请参阅Amazon Local Zones 站点。

本地区中的实例

要使用 Local Zones，您必须先启用它。然后，在本地区中创建子网。您可以在启动实例时指定本地区子网，这会将其置于本地区的本地区子网中。

启动本地区中的实例时，您还可从网络边界组分配 IP 地址。网络边界组是一组唯一的可用区、Local Zones 或 Wavelength 区域（Amazon 可从中公告 IP 地址，例如 `us-west-2-lax-1a`）。您可以从网络边界组分配以下 IP 地址：

- Amazon 提供的弹性 IPv4 地址
- Amazon 提供的 IPv6 VPC 地址（仅在洛杉矶区域可用）

有关如何在 Local Zone 中启动实例的更多信息，请参阅《Amazon Local Zone 用户指南》中的 Amazon Local Zone 入门。

Wavelength 区域

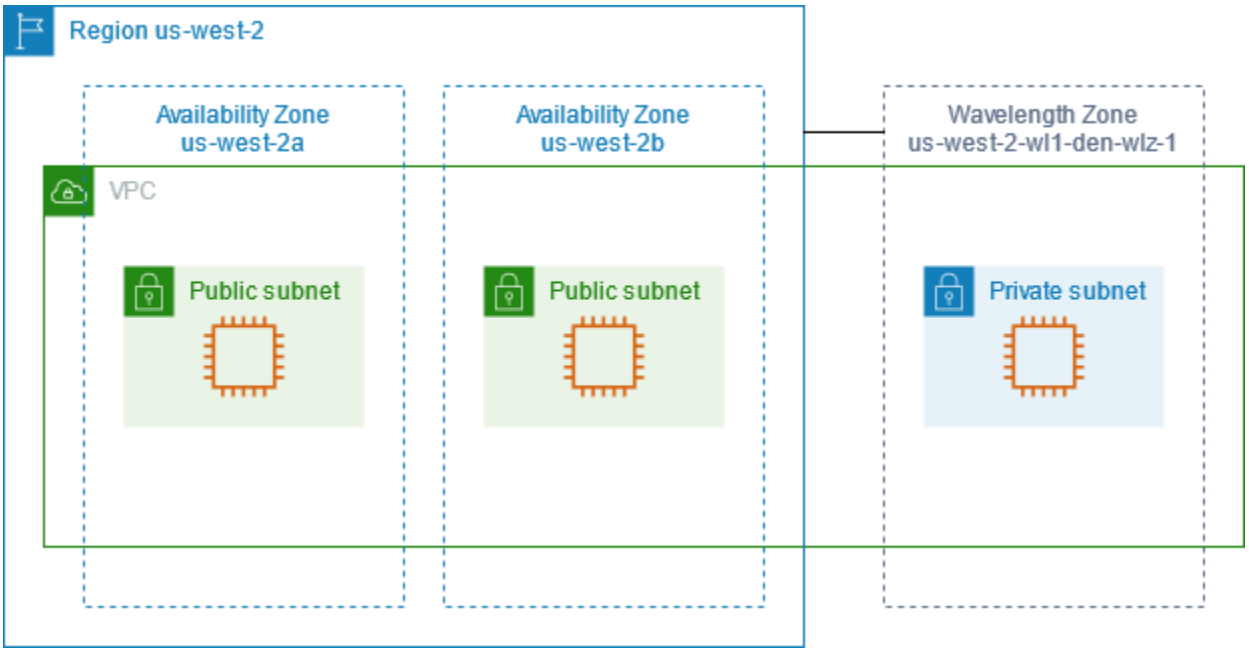
利用 Amazon Wavelength，开发人员可以为移动设备和最终用户打造具有超低延迟的应用程序。Wavelength 可以将标准 Amazon 计算和存储服务部署到电信运营商的 5G 网络边缘。开发人员可

以将 Virtual Private Cloud（VPC）扩展到一个或多个 Wavelength 区域，然后使用 Amazon EC2 实例等 Amazon 资源来运行需要超低延迟并连接到区域中的 Amazon 服务的应用程序。

Wavelength 区域是在其中部署 Wavelength 基础设施的运营商位置中的隔离区域。Wavelength 区域与一个区域相关联。Wavelength 区域是区域的逻辑扩展，由区域中的控制平面管理。

Wavelength 区域的代码由其区域代码后跟一个指示实际位置的标识符组成。例如，在波士顿的 `us-east-1-wl1-bos-wlz-1`。

下图说明了 Amazon 区域 `us-west-2`、其两个可用区及 Wavelength 区域。VPC 跨多个可用区和 Wavelength 区域。VPC 中的每个区域都有一个子网，每个子网都有一个实例。



Wavelength 区域并非在每个区域中都可用。有关支持 Wavelength 区域的区域的信息，请参阅 Amazon Wavelength 开发人员指南中的可用 Wavelength 区域。

可用的 Wavelength 区

有关可用 Wavelength 区的列表，请参阅《Amazon Wavelength Guide》中的 Available Wavelength Zones。

Wavelength 区中的实例

要使用 Wavelength 区域，您必须首先选择加入此区域。然后，在 Wavelength 区中创建子网。您可以在启动实例时指定 Wavelength 子网。您还可以从网络边界组分配运营商 IP 地址，该组是一组唯一的可用区、Local Zones 或 Wavelength 区域（Amazon 可从中公告 IP 地址，例如 `us-east-1-wl1-bos-wlz-1`）。

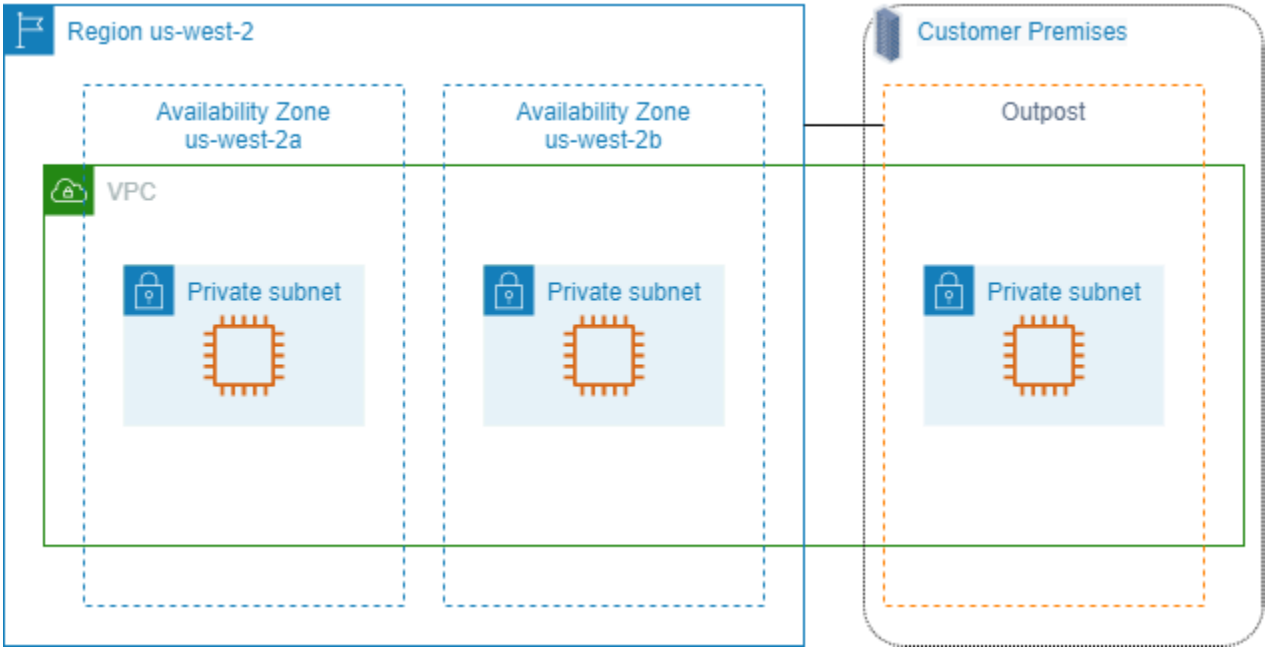
有关如何在 Wavelength 区中启动实例的信息，请参阅《Amazon Wavelength Guide》中的 Get started with Amazon Wavelength。

Amazon Outposts

Amazon Outposts 是一项完全托管的服务，可将 Amazon 基础设施、服务、API 和工具扩展到客户场所。通过提供对 Amazon 托管基础设施的本地访问，Amazon Outposts 使客户能够使用与 Amazon 区域中相同的编程接口在本地构建和运行应用程序，同时使用本地计算和存储资源来满足更低的延迟和本地数据处理需求。

Outpost 是部署在客户站点的 Amazon 计算和存储容量池。Amazon 作为 Amazon 区域的一部分运营、监控和管理此容量。您可以在 Outpost 上创建子网，并在创建 Amazon 资源时指定这些子网。Outpost 子网中的实例使用私有 IP 地址与 Amazon 区域中的其他实例通信，全部都在相同 VPC 内进行。

下图说明了 Amazon 区域 `us-west-2`、其两个可用区及 Outpost。VPC 跨多个可用区和 Outpost。Outpost 位于本地部署的客户数据中心。VPC 中的每个区域都有一个子网，每个子网都有一个实例。



Outpost 上的实例

要开始使用 Amazon Outposts，必须创建一个 Outpost 并订购 Outpost 容量。Amazon Outposts 提供两种外形规格：Outposts 机架和 Outposts 服务器。有关 Outposts 配置的更多信息，请参阅 Amazon Outposts 系列。安装 Outpost 设备后，当您在 Outpost 上启动 EC2 实例时，就可以使用计算和存储容量。

要启动 EC2 实例，您必须创建 Outpost 子网。安全组控制 Outpost 子网中实例的入站和出站流量，就像控制可用区子网中的实例一样。要使用 SSH 连接到 Outpost 子网中的实例，请像为可用区子网中的实例指定密钥对一样，在启动实例时指定密钥对。

有关更多信息，请参阅 Get started with Outposts racks 或 Get started with Outposts servers。

Outposts 机架上的卷

如果 Outposts 容量位于 Outposts 机架上，则可在创建的 Outpost 子网中创建 EBS 卷。创建卷时，请指定 Outpost 的 Amazon Resource Name (ARN)。

以下 create-volume 命令在指定的 Outpost 上创建一个空的 50 GB 卷。

```
aws ec2 create-volume --availability-zone us-east-2a --outpost-arn arn:aws:outposts:us-east-2:123456789012:outpost/op-03e6fecad652a6138 --size 50
```

您可以在不分离卷的情况下动态修改 Amazon EBS gp2 卷的大小。有关在不分离卷的情况下修改卷的更多信息，请参阅《Amazon EBS User Guide》中 Request modifications to your EBS volumes。

建议您将 Outpost 机架上实例的根卷大小限制在 30GiB 或以下。您可以在 AMI 或实例的块设备映射中指定数据卷以提供额外存储。要从引导卷中裁剪未使用的块，请参阅 Amazon Partner Network Blog 中的 How to Build Sparse EBS Volumes。

我们建议您增加根卷的 NVMe 超时。有关更多信息，请参阅《Amazon EBS User Guide》中的 I/O operation timeout。

Outposts 服务器上的卷

Outposts 服务器上的实例提供实例存储卷，但不支持 EBS 卷。选择只有单个 EBS 快照的 Amazon EBS-backed AMI。选择具有足够实例存储空间的实例大小来满足应用程序需求。有关更多信息，请参阅 实例存储卷限制。

Amazon EC2 实例 IP 寻址

Amazon EC2 和 Amazon VPC 支持 IPv4 和 IPv6 寻址协议。默认情况下，Amazon VPC 使用 IPv4 寻址协议；您无法禁用此行为。创建 VPC 时，您必须指定 IPv4 CIDR 块 (一系列私有 IPv4 地址)。您可以选择将 IPv6 CIDR 块分配给您的 VPC，并将来自该块的 IPv6 地址分配给您子网中的实例。

启动 EC2 实例时，您需要指定 VPC 和子网。实例会从子网的 CIDR 范围接收私有 IPv4 地址。您可以选择使用公有 IPv4 地址和 IPv6 地址配置实例。如果不同 VPC 中的 EC2 实例使用公有 IP 地址通信，则流量将保留在 Amazon 私有全球网络中，不会遍历公共互联网。

内容

- 私有 IPv4 地址
- 公有 IPv4 地址
- 公有 IPv4 地址优化
- IPv6 地址
- 多个 IP 地址
- EC2 实例主机名
- 链路本地地址
- 管理 EC2 实例的 IPv4 地址
- 管理 EC2 实例的 IPv6 地址
- EC2 实例的辅助 IP 地址
- 为 Windows 实例配置辅助私有 IPv4 地址

私有 IPv4 地址

私有 IPv4 地址是指无法通过 Internet 访问的 IP 地址。您可以使用私有 IPv4 地址在同一 VPC 中实现实例之间的通信。有关私有 IPv4 地址标准和规范的更多信息，请参阅 RFC 1918。我们会使用 DHCP 将私有 IPv4 地址分配到实例。

> **Note**
>
> 您可以创建一个具有公共可路由的 CIDR 块 (不在 RFC 1918 中指定的私有 IPv4 地址范围内) 的 VPC。但是，出于本文档的写作目的，我们的私有 IPv4 地址(或“私有 IP 地址”)指的是位于 VPC 的 IPv4 CIDR 范围内的 IP 地址。

VPC 子网可以是以下类型之一：

- 仅 IPv4 子网：您只能在这些子网中创建已分配 IPv4 地址的资源。
- 仅 IPv6 子网：您只能在这些子网中创建已分配 IPv6 地址的资源。
- IPv4 和 IPv6 子网：您可以在这些子网中创建已分配 IPv4 或 IPv6 地址的资源。

在仅 IPv4 或双协议堆栈（IPv4 和 IPv6）子网中启动 EC2 实例时，该实例将在子网的 IPv4 地址范围内收到一个主要私有 IP 地址。有关更多信息，请参阅《Amazon VPC 用户指南》中的 IP 寻址。如果

您在启动实例时未指定主要私有 IP 地址，我们会在子网的 IPv4 范围内为您选择一个可用的 IP 地址。每个实例都具有分配了主要私有 IPv4 地址的默认网络接口（索引 0）。您还可以指定其他私有 IPv4 地址，即辅助私有 IPv4 地址。与主要私有 IP 地址不同的是，辅助私有 IP 地址可以从一个实例重新分配到另一个实例。有关更多信息，请参阅 多个 IP 地址。

私有 IPv4 地址（无论是主地址还是辅助地址）会在实例停止和重新启动或休眠和启动时保持与网络接口的关联，并在实例终止时释放。

公有 IPv4 地址

公有 IP 地址是指可通过 Internet 访问的 IPv4 地址。您可以使用公用 地址在您的实例和 Internet 之间进行通信。

在默认 VPC 中启动实例时，默认情况下，我们会为实例分配公有 IP 地址。当您在非默认 VPC 中启动实例时，子网的一个属性会确定在该子网中启动的实例是否从公有 IPv4 地址池接收公有 IP 地址。默认情况下，我们不会将公有 IP 地址分配到非默认子网中启动的实例。

您可以按如下所示控制实例是否接收公有 IP 地址：

- 修改子网的公有 IP 寻址属性。有关更多信息，请参阅《Amazon VPC 用户指南》中的修改子网的公有 IPv4 寻址属性。
- 在启动时启用或禁用公有 IP 寻址功能。这将覆盖子网的公有 IP 寻址属性。有关更多信息，请参阅在启动时分配公有 IPv4 地址。
- 启动后取消分配实例的公有 IP 地址。有关更多信息，请参阅 the section called “管理 IP 地址”。

公有 IP 地址将从 Amazon 的公有 IPv4 地址池分配给实例，不与您的 Amazon 账户关联。在取消公有 IP 地址与实例的关联后，该地址即会释放回公有 IPv4 地址池中，并且您无法重新使用该地址。

在以下情况下，我们会从您的实例释放公有 IP 地址，并为其分配新地址：

- 当该实例停止、休眠或终止后，我们会释放公有 IP 地址。当您启动已停止或休眠的实例时，我们会分配一个新的公有 IP 地址。
- 当您将弹性 IP 地址与该实例关联时，我们会释放该公有 IP 地址。当您将弹性 IP 地址与实例取消关联时，我们会分配一个新的公有 IP 地址。
- 如果我们释放了实例的公有 IP 地址并且其有辅助网络接口，则我们不会分配新的公有 IP 地址。
- 如果我们释放了实例的公有 IP 地址并且其有与弹性 IP 地址关联的辅助私有 IP 地址，则我们不会分配新的公有 IP 地址。

如果您需要可根据需要关联到实例并从实例进行关联的永久公有 IP 地址，可改为使用弹性 IP 地址。

如果您使用动态 DNS 来将现有 DNS 名称映射到新实例的公有 IP 地址，可能需要 24 小时，以便 IP 地址可以传递到整个 Internet。其结果是，新的实例可能无法接收流量，而已终止实例继续接收请求。要解决此问题，请使用弹性 IP 地址。您可以分配自己的弹性 IP 地址，并将其与您的实例相关联。有关更多信息，请参阅 弹性 IP 地址。

如果使用的是 Amazon VPC IP 地址管理器（IPAM），则可以从 Amazon 中获取连续的公有 IPv4 地址块，再用其将弹性 IP 地址分配给 Amazon 资源。使用连续的 IPv4 地址块可以显著减少安全访问控制列表的管理开销，简化在 Amazon 上扩展的企业的 IP 地址分配和跟踪工作。有关更多信息，请参阅《Amazon VPC IPAM User Guide》中的 Allocate sequential Elastic IP addresses from an IPAM pool。

注意事项

- Amazon 将对所有公有 IPv4 地址收费，包括与运行的实例相关联的公有 IPv4 地址和弹性 IP 地址。有关更多信息，请参阅 Amazon VPC 定价页面中的公有 IPv4 地址定价选项卡。
- 通过公有 NAT IP 地址访问其他实例的实例需要支付区域或 Internet 数据传输费用，具体取决于这些实例是否在同一区域内。

公有 IPv4 地址优化

Amazon 将对所有公有 IPv4 地址收费，包括与运行的实例相关联的公有 IPv4 地址和弹性 IP 地址。有关更多信息，请参阅 Amazon VPC 定价页面中的公有 IPv4 地址定价选项卡。

以下列表包含相应的措施，您可以采取这些措施来优化所使用的公有 IPv4 地址数量：

- 使用弹性负载均衡器对 EC2 实例的流量进行负载均衡，以及在分配给这些实例的主 ENI 上禁用自动分配公有 IP。负载均衡器使用单个公有 IPv4 地址，因此可减少公有 IPv4 地址数量。您可能还需要整合现有的负载均衡器，以进一步减少公有 IPv4 地址数量。
- 如果使用 NAT 网关的唯一原因是将 SSH 连接到私有子网中的 EC2 实例以进行维护或应对紧急情况，请考虑改用 EC2 Instance Connect Endpoint。使用 EC2 Instance Connect Endpoint，您可从互联网连接到实例，而无需实例具有公有 IPv4 地址。
- 如果您的 EC2 实例位于公有子网中并已分配公有 IP 地址，请考虑将这些实例移至私有子网，移除公有 IP 地址，然后使用公有 NAT 网关来允许访问您的 EC2 实例和从实例中访问。使用 NAT 网关需要考虑成本。使用此计算方法来确定 NAT 网关是否具有成本效益。您可以通过创建 Amazon 账单成本和使用情况报告来获取此计算所需的`Number of public IPv4 addresses`。

```
NAT gateway per hour + NAT gateway public IPs + NAT gateway transfer / Existing
 public IP cost
```

其中：

- NAT gateway per hour = $0.045 * 730 hours in a month * Number of Availability Zones the NAT gateways are in
- NAT gateway public IPs = $0.005 * 730 hours in a month * Number of IPs associated with your NAT gateways
- NAT gateway transfer = $0.045 * Number of GBs that will go through the NAT gateway in a month
- Existing public IP cost = $0.005 * 730 hours in a month * Number of public IPv4 addresses

如果总计值小于 1，则 NAT 网关比公有 IPv4 地址便宜。

- 使用 Amazon PrivateLink 私密连接至 Amazon 服务或其他 Amazon 账户托管服务，而不是使用公有 IPv4 地址和互联网网关。
- 自带 IP 地址范围（BYOIP）至 Amazon 并对公有 IPv4 地址使用该范围，而不是使用 Amazon 拥有的公有 IPv4 地址。
- 关闭公有 IPv4 地址自动分配到在子网中启动的实例。创建子网时，VPC 通常默认禁用此选项，但您应检查现有子网以确保其处于禁用状态。
- 如果您的 EC2 实例不需要公有 IPv4 地址，请检查连接到您实例的网络接口是否已禁用自动分配公有 IP。
- 为私有子网中的 EC2 实例在 Amazon Global Accelerator 中配置加速器端点，以使互联网流量无需公有 IP 地址即可直接流向 VPC 中的端点。您也可以将自己的地址带入 Amazon Global Accelerator，并使用自己的 IPv4 地址作为加速器的静态 IP 地址。

IPv6 地址

IPv6 地址具有全局唯一性，可以配置为保持私有或通过互联网进行访问。Amazon 中提供公有和私有 IPv6 寻址：

- 私有 IPv6：Amazon 认为私有 IPv6 地址是那些未公开发布且不能从 Amazon 公开发布在互联网上的地址。
- 公共 IPv6：Amazon 认为公共 IPv6 地址是 Amazon 在互联网上已公开发布的地址。

有关公有和私有 IPv6 地址的更多信息，请参阅《Amazon VPC 用户指南》中的 IPv6 地址。

除了 C1、M1、M2、M3 和 T1 实例类型，其余实例类型都支持 IPv6 地址。

如果您的 VPC 和子网关联了 IPv6 CIDR 块，并且满足以下条件之一，则您的 EC2 实例会收到 IPv6 地址：

- 您的子网配置为在启动期间向实例自动分配 IPv6 地址。有关更多信息，请参阅修改子网的 IP 寻址属性。
- 您在启动期间为实例分配了 IPv6 地址。
- 您在启动后为实例的主网络接口分配了 IPv6 地址。
- 您向同一子网中的某个网络接口分配 IPv6 地址，并在启动后将此网络接口附加到您的实例。

当实例在启动期间收到 IPv6 地址时，此地址将与实例的主网络接口（索引 0）关联。您可以通过以下方式管理实例主网络接口的 IPv6 地址：

- 从网络接口分配和取消分配 IPv6 地址。可以分配给网络接口的 IPv6 地址数量以及可以附加到实例的网络接口数量因实例类型而异。有关更多信息，请参阅 每个网络接口的最大 IP 地址数。
- 启用主 IPv6 地址。主 IPv6 地址使您能够避免中断实例或 ENI 的流量。有关更多信息，请参阅 为 Amazon EC2 实例创建网络接口 或 管理网络接口的 IP 地址。

IPv6 地址会在您停止和启动或休眠和启动实例时保留下来，并在您终止实例时释放出来。您无法重新分配已分配给某个网络接口的 IPv6 地址；您必须先取消分配此 IPv6 地址。

您可以通过控制子网的路由或通过使用安全组和网络 ACL 规则来控制能否通过实例的 IPv6 地址对其进行访问。有关更多信息，请参阅《Amazon VPC 用户指南》中的互联网络流量隐私。

有关预留 IPv6 地址范围的更多信息，请参阅 IANA IPv6 特殊用途地址注册表和 RFC4291。

多个 IP 地址

您可以为实例指定多个私有 IPv4 和 IPv6 地址。您可为实例指定的网络接口和私有 IPv4 和 IPv6 地址的数量取决于该实例的类型。有关更多信息，请参阅 每个网络接口的最大 IP 地址数。

使用案例

- 在单个服务器上使用多个 SSL 证书，并为每个证书关联一个指定的 IP 地址，以在单个服务器上托管多个网站。

- 操作每个网络接口有多个 IP 地址的网络应用，如防火墙或负载均衡器。
- 当实例发生故障时，可将内部流量重定向到备用实例，方法是为备用实例重新分配辅助 IP 地址。

多个 IP 地址的工作方式

- 您可以为任何网络接口分配辅助私有 IPv4 地址。
- 您可以将多个 IPv6 地址分配给拥有关联 IPv6 CIDR 块的子网中的网络接口。
- 您必须从子网的 IPv4 CIDR 块范围内为网络接口选择辅助 IPv4 地址。
- 您必须从子网的 IPv6 CIDR 块范围内为网络接口选择辅助 IPv6。
- 将安全组与网络接口关联，而不是与各 IP 地址关联。因此，网络接口中指定的每个 IP 地址均受其网络接口的安全组约束。
- 可将多个 IP 地址分配给附加到正在运行或已停止实例的网络接口，也可以取消分配操作。
- 如果您明确允许，已分配给某个网络接口的辅助私有 IPv4 地址可重新分配给其他网络接口。
- 无法将 IPv6 地址重新分配给其他网络接口；您必须先取消分配给现有网络接口的 IPv6 地址。
- 当使用命令行工具或 API 将多个 IP 地址分配给某个网络接口时，如果其中有一个 IP 地址无法分配，整个操作都会失败。
- 当网络接口与实例分离或附加到实例时，主要私有 IPv4 地址、辅助私有 IPv4 地址、弹性 IP 地址以及 IPv6 地址将仍然属于此辅助网络接口。
- 尽管您无法从实例分离主要网络接口，但是您可以将主要网络接口的辅助私有 IPv4 地址重新分配给另一个网络接口。

有关更多信息，请参阅 the section called “辅助 IP 地址”。

EC2 实例主机名

当您创建 EC2 实例时，Amazon 为该实例创建主机名。有关主机名类型及其 Amazon 预置方式的更多信息，请参阅 EC2 实例主机名和域。Amazon 提供了 DNS 服务器，可将 Amazon 提供的主机名解析为 IPv4 和 IPv6 地址。Amazon DNS 服务器位于 VPC 网络范围起始地址 + 2 的位置。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。

链路本地地址

众所周知，链路本地地址是不可路由的 IP 地址。Amazon EC2 使用来自链路本地地址空间的地址来提供只能从 EC2 实例访问的服务。这些服务不在实例上运行，而是在底层主机上运行。当您通过链路本地地址访问这些服务时，您将与 Xen 虚拟机管理程序或 Nitro 控制器进行通信。

链路本地地址范围

- IPv4 – 169.254.0.0/16（169.254.0.0 到 169.254.255.255）
- IPv6 – fe80::/10

您通过链路本地地址访问的服务

- 实例元数据服务
- Amazon Route 53 Resolver（也称为 Amazon DNS 服务器）
- Amazon Time Sync Service
- Amazon KMS 服务器

管理 EC2 实例的 IPv4 地址

您可以在启动实例时为其分配公有 IPv4 地址。您可以在控制台中通过 Instances（实例）或 Network Interfaces（网络接口）页面查看实例的 IPv4 地址。

任务

- 在启动时分配公有 IPv4 地址
- 在启动时分配私有 IPv4 地址
- 查看主 IPv4 地址
- 使用实例元数据查看 IPv4 地址

在启动时分配公有 IPv4 地址

所有子网都有一个属性可确定在子网中启动的实例是否分配了公有 IP 地址。默认情况下，非默认子网的此属性设置为 false，默认子网的此属性设置为 true。启动实例时，您也可以通过公有 IPv4 寻址功能来控制是否为实例分配公有 IPv4 地址；您可以覆盖子网 IP 寻址属性的默认行为。公有 IPv4 地址从 Amazon 的公有 IPv4 地址池进行分配，并分配给设备索引为 0 的网络接口。此功能取决于启动实例时的特定条件。

注意事项

- 启动后，您可以通过管理与网络接口关联的 IP 地址来将实例的公有 IP 地址取消分配。有关公有 IPv4 地址的更多信息，请参阅公有 IPv4 地址。

- 如果指定多个网络接口，则不能自动分配公有 IP 地址。此外，如果指定设备索引为 0 的某个现有网络接口，则无法使用自动分配公有 IP 功能来覆盖子网设置。
- 无论您是否在启动时为实例分配了公有 IP 地址，您都可以在启动后将弹性 IP 地址与实例相关联。有关更多信息，请参阅 弹性 IP 地址。您还可以修改子网的公有 IPv4 寻址行为。有关更多信息，请参阅修改子网的公有 IPv4 寻址属性。

Console

在启动时分配公有 IPv4 地址

请按照步骤启动实例，并在配置网络设置时选择 Auto-assign Public IP（自动分配公有 IP）。

Amazon CLI

在启动时分配公有 IPv4 地址

使用 run-instances 命令和 `--associate-public-ip-address` 选项。

```
--associate-public-ip-address
```

PowerShell

在启动时分配公有 IPv4 地址

将 New-EC2Instance cmdlet 与 `-AssociatePublicIp` 参数结合使用。

```
-AssociatePublicIp $true
```

在启动时分配私有 IPv4 地址

您可以在子网 IPv4 地址范围中指定一个私有 IPv4 地址，也可以让 Amazon EC2 为您选择一个。该地址会分配给主网络接口。

要在启动后分配 IPv4 地址，请参阅the section called “为实例分配辅助 IP 地址”。

Console

在启动时分配私有 IPv4 地址

按照程序启动实例。配置网络设置时，展开高级网络配置，然后输入主 IP 的值。

Amazon CLI

在启动时分配私有 IPv4 地址

使用 run-instances 命令和 `--private-ip-address` 选项。

```
--private-ip-addresses 10.251.50.12
```

要让 Amazon EC2 选择 IP 地址，请省略此选项。

PowerShell

在启动时分配私有 IPv4 地址

将 New-EC2Instance cmdlet 与 `-PrivateIpAddress` 参数结合使用。

```
-PrivateIpAddress 10.251.50.12
```

要让 Amazon EC2 选择 IP 地址，请省略此参数。

查看主 IPv4 地址

公有 IPv4 地址在控制台中显示为网络接口的属性，但它通过 NAT 映射到主要私有 IPv4 地址。因此，如果您检查实例网络接口的属性 (例如，通过 `ifconfig` [Linux] 或 `ipconfig` [Windows])，则不会显示公有 IPv4 地址。

Console

查看实例的 IPv4 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 在联网选项卡上，找到公有 IPv4 地址和私有 IPv4 地址。
5. （可选）联网选项卡中还包含实例的网络接口和弹性 IP 地址。

Amazon CLI

查看实例的主 IPv4 地址

可以使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].PrivateIpAddress" \
    --output text
```

下面是示例输出。

```
10.251.50.12
```

PowerShell

查看实例的主 IPv4 地址

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.PrivateIpAddress
```

下面是示例输出。

```
10.251.50.12
```

使用实例元数据查看 IPv4 地址

可以通过检索实例元数据来获取实例的 IPv4 地址。有关更多信息，请参阅 使用实例元数据管理 EC2 实例。

使用实例元数据查看 IPv4 地址

1. 连接到您的实例。有关更多信息，请参阅 连接到您的 EC2 实例。
2. 运行以下命令之一。

   IMDSv2

   Linux

   在 Linux 实例上运行以下命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/local-ipv4
```

Windows

在 Windows 实例上运行以下命令。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

Linux

在 Linux 实例上运行以下命令。

```
curl http://169.254.169.254/latest/meta-data/local-ipv4
```

Windows

在 Windows 实例上运行以下命令。

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/local-ipv4
```

3. 使用以下命令之一访问公有 IP 地址。如果实例关联了弹性 IP 地址，则命令会返回该弹性 IP 地址。

IMDSv2

Linux

在 Linux 实例上运行以下命令。

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
 "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/
meta-data/public-ipv4
```

Windows

在 Windows 实例上运行以下命令。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-
seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/public-ipv4
```

IMDSv1

Linux

在 Linux 实例上运行以下命令。

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

Windows

在 Windows 实例上运行以下命令。

```
Invoke-RestMethod http://169.254.169.254/latest/meta-data/public-ipv4
```

管理 EC2 实例的 IPv6 地址

如果您的 VPC 和子网有与之关联的 IPv6 CIDR 块，则您可以在启动期间或之后向实例分配 IPv6 地址。您可以在控制台的实例页面或网络接口页面上访问实例的 IPv6 地址。以下任务为您的实例配置 IP 地址。要改为配置网络接口的 IP 地址，请参阅the section called “管理 IP 地址”。

任务

- 向实例分配 IPv6 地址
- 查看实例的 IPv6 地址
- 使用实例元数据查看 IPv6 地址
- 取消分配给实例的 IPv6 地址

向实例分配 IPv6 地址

您可以在子网 IPv6 地址范围中指定一个 IPv6 地址，也可以让 Amazon EC2 为您选择一个。该地址会分配给主网络接口。请注意，C1、M1、M2、M3 和 T1 实例类型不支持 IPv6 地址。

Console

在启动后分配 IPv6 地址

按照程序启动实例。在配置网络设置时，选择自动分配 IPv6 IP 选项。如果看不到此选项，则表示所选子网没有关联任何 IPv6 CIDR 块。

在启动实例后分配 IPv6 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您的实例，然后依次选择操作、联网和管理 IP 地址。
4. 展开网络接口。在 IPv6 地址下，选择分配新 IP 地址。
5. 输入子网范围内的一个 IPv6 地址，也可将该字段留空，以便让 Amazon EC2 为您选择 IPv6 地址。如果看不到此选项，则表示实例子网没有关联任何 IPv6 CIDR 块。
6. 选择保存。

Amazon CLI

在启动后分配 IPv6 地址

使用 run-instances 命令和 `--ipv6-addresses` 选项。以下示例会分配两个 IPv6 地址。

```
--ipv6-addresses Ipv6Address=2001:db8::1234:5678:1.2.3.4
 Ipv6Address=2001:db8::1234:5678:5.6.7.8
```

要让 Amazon EC2 选择 IPv6 地址，请改用 `--ipv6-address-count` 选项。以下示例会分配两个 IPv6 地址。

```
--ipv6-address-count 2
```

在启动实例后分配 IPv6 地址

使用 assign-ipv6-addresses 命令。以下示例会分配两个 IPv6 地址。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 2001:db8::1234:5678:5.6.7.8
```

要让 Amazon EC2 选择 IPv6 地址，请改用 `--ipv6-address-count` 选项。以下示例会分配两个 IPv6 地址。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-address-count 2
```

PowerShell

在启动后分配 IPv6 地址

将 New-EC2Instance cmdlet 与 `-Ipv6Address` 参数结合使用。以下示例会分配两个 IPv6 地址。

```
-Ipv6Address $ipv6addr1,$ipv6addr2
```

按如下所示定义 IPv6 地址。

```
$ipv6addr1 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
$ipv6addr2 = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr2.Ipv6Address = "2001:db8::1234:5678:5.6.7.8"
```

要让 Amazon EC2 选择 IPv6 地址，请改用 `-Ipv6AddressCount` 参数。以下示例会分配两个 IPv6 地址。

```
-Ipv6AddressCount 2
```

在启动实例后分配 IPv6 地址

使用 Register-EC2Ipv6AddressList cmdlet。以下示例会分配两个 IPv6 地址。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address "2001:db8::1234:5678:1.2.3.4","2001:db8::1234:5678:5.6.7.8"
```

要让 Amazon EC2 选择 IPv6 地址，请改用 `-Ipv6AddressCount` 参数。以下示例会分配两个 IPv6 地址。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6AddressCount 2
```

查看实例的 IPv6 地址

您可以查看实例的 IPv6 地址。

Console

查看实例的 IPv6 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 在联网选项卡上，找到 IPv6 地址。

Amazon CLI

查看实例的 IPv6 地址

可以使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].Ipv6Address" \
    --output text
```

下面是示例输出。

```
2001:db8::1234:5678:1.2.3.4
```

PowerShell

查看实例的 IPv6 地址

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.Ipv6Address
```

下面是示例输出。

```
2001:db8::1234:5678:1.2.3.4
```

使用实例元数据查看 IPv6 地址

连接到实例后，您可以使用实例元数据检索 IPv6 地址。首先，您必须从 `http://169.254.169.254/latest/meta-data/network/interfaces/macs/` 中获取实例的 MAC 地址。

IMDSv2

Linux

在 Linux 实例上运行以下命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Windows

在 Windows 实例上运行以下 cmdlet。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
```

```
    -Method GET -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/mac-address/ipv6s
```

IMDSv1

Linux

在 Linux 实例上运行以下命令。

```
curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/
ipv6s
```

Windows

在 Windows 实例上运行以下 cmdlet。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/network/interfaces/
macs/mac-address/ipv6s
```

取消分配给实例的 IPv6 地址

您可以随时取消分配给实例的 IPv6 地址。

Console

取消分配给实例的 IPv6 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您的实例，然后依次选择操作、联网和管理 IP 地址。
4. 展开网络接口。在 IPv6 地址下方，选择 IPv6 地址旁边的取消分配。
5. 选择 Save。

Amazon CLI

取消分配给实例的 IPv6 地址

使用 unassign-ipv6-addresses 命令。

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4
```

PowerShell

取消分配给实例的 IPv6 地址

使用 Unregister-EC2Ipv6AddressList cmdlet。

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8::1234:5678:1.2.3.4
```

EC2 实例的辅助 IP 地址

主 IP 地址是指分配给网络接口的第一个 IPv4 地址。辅助 IP 地址是指分配给网络接口的其他 IPv4 地址。有关更多信息，请参阅 the section called “多个 IP 地址”。

您还可以为实例分配多个 IPv6 地址。有关更多信息，请参阅 the section called “IPv6 地址”。

任务

- 为实例分配辅助 IP 地址
- 配置操作系统以使用辅助 IP 地址
- 取消分配给实例的辅助 IP 地址

为实例分配辅助 IP 地址

您可以在启动实例时或在实例运行后为实例的网络接口分配辅助 IP 地址。

Console

在启动时分配辅助 IP 地址

1. 按照程序启动实例。配置网络设置时，展开高级网络配置。
2. 对于辅助 IP，选择自动分配，然后输入要让 Amazon EC2 分配的 IP 地址数量。也可以选择手动分配，然后输入相应的 IPv4 地址。
3. 完成剩余步骤以启动实例。

在启动后分配辅助 IP 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您的实例，然后依次选择操作、联网和管理 IP 地址。
4. 展开网络接口。
5. 要添加 IPv4 地址，请在 IPv4 地址下选择分配新 IP 地址。输入子网范围内的一个 IPv4 地址，也可将该字段留空，以便让 Amazon EC2 为您选择一个。
6. 选择保存。

Amazon CLI

在启动时分配辅助 IP 地址

使用 run-instances 命令和 `--secondary-private-ip-addresses` 选项。

```
--secondary-private-ip-addresses 10.251.50.12
```

要让 Amazon EC2 选择 IP 地址，请改用 `--secondary-private-ip-address-count` 选项。以下示例会分配一个辅助 IP 地址。

```
--secondary-private-ip-address-count 1
```

您也可以创建网络接口。有关更多信息，请参阅 the section called “创建网络接口”。

在启动后分配辅助 IP 地址

使用 assign-private-ip-addresses 命令和 `--private-ip-addresses` 选项。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --private-ip-addresses 10.251.50.12
```

要让 Amazon EC2 选择 IPv4 地址，请改用 `--secondary-private-ip-address-count` 参数。以下示例会分配一个 IPv4 地址。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-ids eni-1234567890abcdef0 \
    --secondary-private-ip-address-count 1
```

PowerShell

在启动时分配辅助 IP 地址

您必须创建一个网络接口。有关更多信息，请参阅 the section called “创建网络接口”。

在启动后分配辅助 IP 地址

使用 Register-EC2PrivateIpAddress cmdlet 和 `-PrivateIpAddress` 参数。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

要让 Amazon EC2 选择 IPv4 地址，请改用 `-SecondaryPrivateIpAddressCount` 参数。以下示例会分配一个 IPv4 地址。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SecondaryPrivateIpAddressCount 1
```

配置操作系统以使用辅助 IP 地址

为实例分配辅助 IP 地址后，您必须配置实例上的操作系统，以便识别此额外的私有 IP 地址。

Linux 实例

- 如果您使用的是 Amazon Linux，ec2-net-utils 包可以在此步骤上为您提供帮助。它能在实例运行期间配置您附加的其他网络接口，在 DHCP 租约续订期间更新辅助 IPv4 地址，并更新相关的路由规则。根据您的系统，您可以使用以下命令之一立即刷新接口列表：`sudo systemctl restart systemd-networkd`（AL2023）或 `sudo service network restart`（Amazon Linux 2）。您可以使用以下命令查看最新列表：`ip addr li`。如果您需要手动控制网络配置，可以删除 ec2-net-utils 包。有关更多信息，请参阅 Configure your network interface using ec2-net-utils。
- 如果您正在使用其他 Linux 分配，请参阅有关 Linux 分配的文档。您可以搜索有关配置其他网络接口和辅助 IPv4 地址的信息。如果实例在同一子网中有两个或更多接口，请搜索有关利用路由规则解决非对称路由的信息。

Windows 实例

有关更多信息，请参阅 为 Windows 实例配置辅助私有 IPv4 地址。

取消分配给实例的辅助 IP 地址

如果不再需要某个辅助 IP 地址，您可以将其从实例或网络接口取消分配。当取消分配给网络接口的辅助私有 IPv4 地址后，弹性 IP 地址 (如果存在) 也会断开相关联系。

Console

取消分配给实例的辅助私有 IPv4 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择操作、联网和管理 IP 地址。
4. 展开网络接口。在 IPv4 地址中，为要取消分配的 IPv4 地址选择取消分配。
5. 选择保存。

Amazon CLI

取消分配辅助私有 IP 地址

使用 unassign-private-ip-addresses 命令。

```
aws ec2 unassign-private-ip-addresses \
    --network-interface eni-1234567890abcdef0\
    --private-ip-addresses 10.251.50.12
```

PowerShell

取消分配辅助私有 IP 地址

使用 Unregister-EC2PrivateIpAddress cmdlet。

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterface eni-1234567890abcdef0 `
    -PrivateIpAddress 10.251.50.12
```

为 Windows 实例配置辅助私有 IPv4 地址

您可以为实例指定多个私有 IPv4 地址。将辅助私有 IPv4 地址分配给实例后，您必须在实例上配置操作系统才能识别辅助私有 IPv4 地址。

> **Note**
>
> 以下说明是基于 Windows Server 2022 的。这些步骤的实施可能因 Windows 实例的操作系统而异。

任务

- 先决条件
- 步骤 1：在实例中配置静态 IP 寻址
- 步骤 2：为实例配置辅助私有 IP 地址
- 步骤 3：配置应用程序以使用辅助私有 IP 地址

先决条件

- 将辅助私有 IPv4 地址分配给实例的网络接口。您可以在启动实例时或在实例运行后分配辅助私有 IPv4 地址。有关更多信息，请参阅 为实例分配辅助 IP 地址。

步骤 1：在实例中配置静态 IP 寻址

要使 Windows 实例能够使用多个 IP 地址，必须配置实例，使其使用静态 IP 寻址，而不是 DHCP 服务器。

> **Important**
>
> 当在实例中配置静态 IP 寻址时，IP 地址必须与控制台、CLI 或 API 中显示的地址精确匹配。如果您输入的 IP 地址不正确，实例可能会不可连接。

在 Windows 实例上配置静态 IP 寻址

1. 连接到您的实例。
2. 通过执行以下步骤，查找实例的 IP 地址、子网掩码和默认网关地址：
   - 在 PowerShell 中运行下面的命令：

```
ipconfig /all
```

检查输出，记下网络接口的 IPv4 地址、子网掩码、默认网关和 DNS 服务器值。输出应与以下示例类似：

```
...

Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . : us-west-2.compute.internal
   Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
   Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, April 8, 2024 12:19:29 PM
   Lease Expires . . . . . . . . . . : Monday, April 8, 2024 4:49:30 PM
   Default Gateway . . . . . . . . . : 10.200.0.1
   DHCP Server . . . . . . . . . . . : 10.200.0.1
   DHCPv6 IAID . . . . . . . . . . . : 151166011
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-
E7
   DNS Servers . . . . . . . . . . . : 10.200.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

3. 在 PowerShell 中运行以下命令，打开网络和共享中心：

```
& $env:SystemRoot\system32\control.exe ncpa.cpl
```

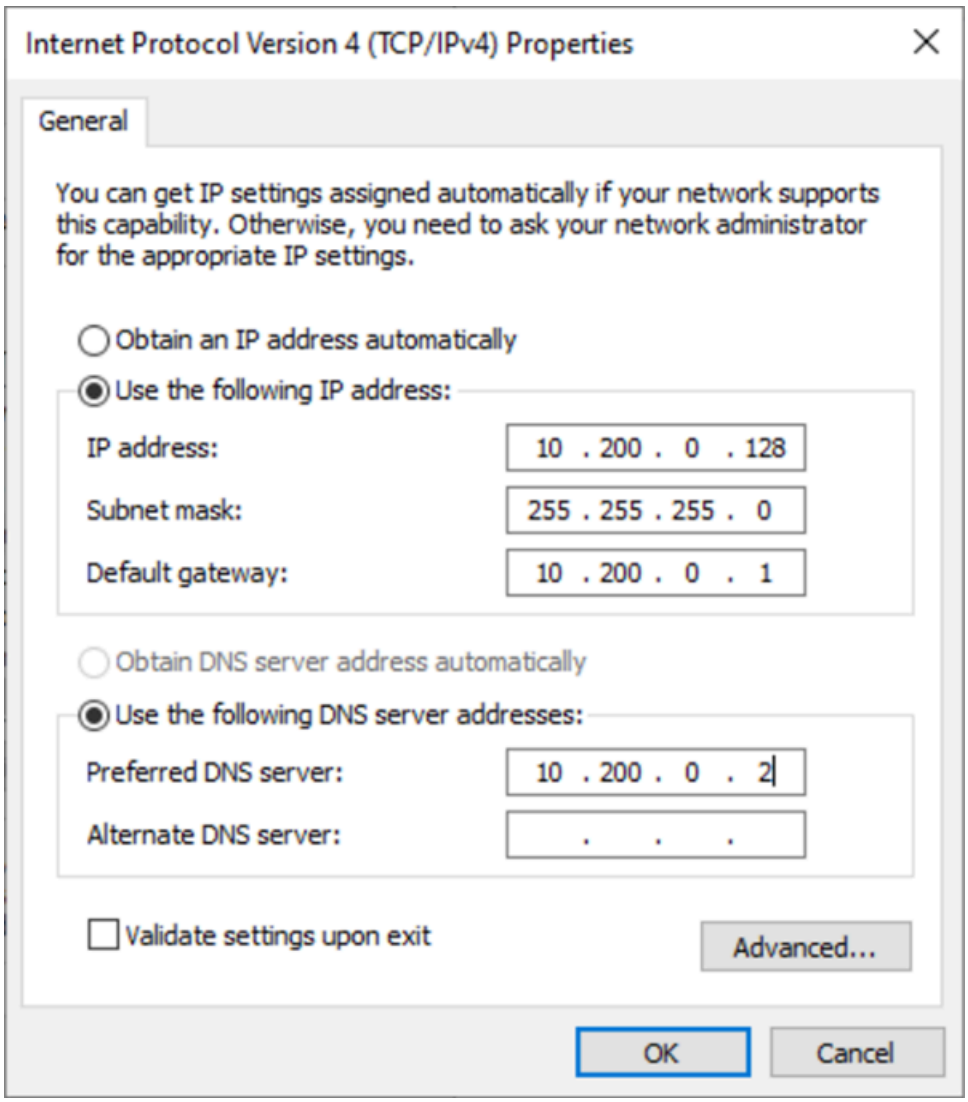
4. 打开网络接口（本地连接或以太网）的上下文（右键单击）菜单，选择属性。
5. 依次选择 Internet 协议版本 4 (TCP/IPv4)、属性。
6. 在 Internet 协议版本 4 (TCP/IPv4)属性对话框中，选择使用下面的 IP 地址，输入以下值，然后选择确定。

字段	值
IP 地址	上面的步骤 2 中包含的 IPv4 地址。
子网掩码	上面的步骤 2 中包含的子网掩码。

字段	值
默认网关	上面的步骤 2 中包含的默认网关地址。
首选 DNS 服务器	上面的步骤 2 中包含的 DNS 服务器。
备用 DNS 服务器	上面的步骤 2 中包含的备用 DNS 服务器。如果未列出备用 DNS 服务器，请将此字段留为空白。

Important

如果将 IP 地址设置为当前 IP 地址以外的任何值，则会丢失与实例的连接。



当实例从使用 DHCP 转为使用静态寻址时，会短时丢失与 Windows 实例的 RDP 连接。实例会像之前一样保留 IP 地址信息，但是现在，这些信息是静态的，且不受 DHCP 管理。

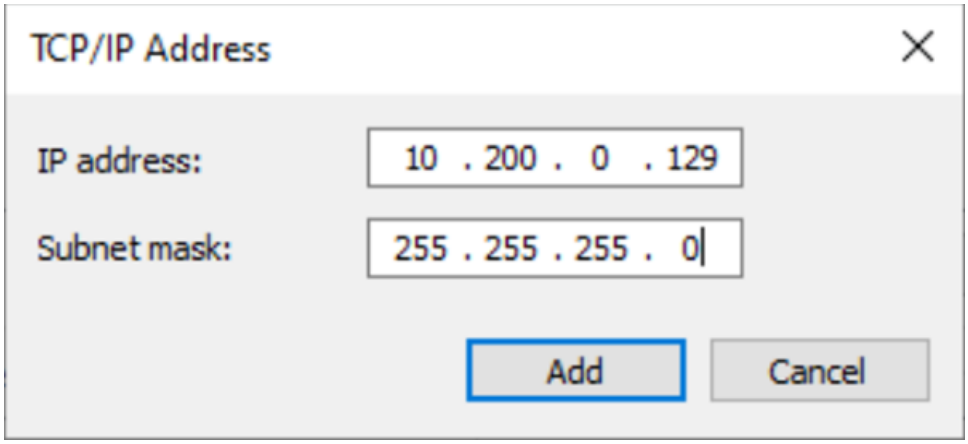
步骤 2：为实例配置辅助私有 IP 地址

在 Windows 实例上设置了静态 IP 寻址之后，您便可准备第二个私有 IP 地址。

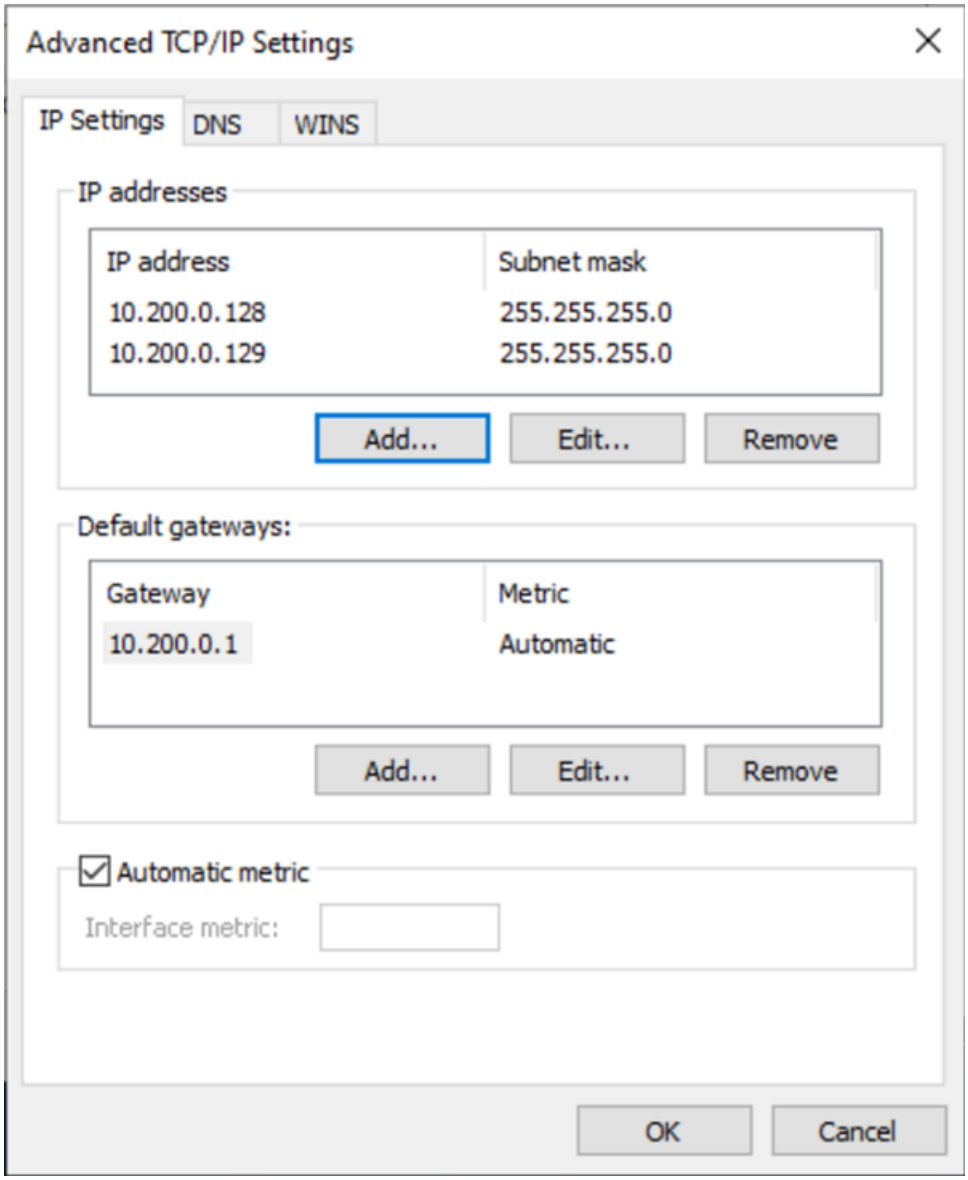
配置辅助 IP 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择您的实例。
3. 在 Networking (联网) 上，记下辅助 IP 地址。

4. 连接到您的实例。
5. 在您的 Windows 实例上，选择开始、控制面板。
6. 选择网络和 Internet、网络和共享中心。
7. 选择网络接口（本地连接或以太网），然后选择属性。
8. 在本地连接属性页面上，选择 Internet 协议版本 4 (TCP/IPv4)、属性、高级。
9. 选择添加。
10. 在 TCP/IP 地址对话框中，键入辅助私有 IP 地址作为 IP 地址。对于子网掩码，键入与步骤 1：在实例中配置静态 IP 寻址 中为主私有 IP 地址输入的子网掩码，然后选择添加。



11. 验证 IP 地址设置，然后选择确定。



12. 选择确定、关闭。
13. 要确认辅助 IP 地址已添加到操作系统，请在 PowerShell 中运行 `ipconfig /all` 命令。输出应与以下内容类似：

```
Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Amazon Elastic Network Adapter #2
   Physical Address. . . . . . . . . : 02-9C-3B-FC-8E-67
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f4d1:a773:5afa:cd1%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.200.0.128(Preferred)
```

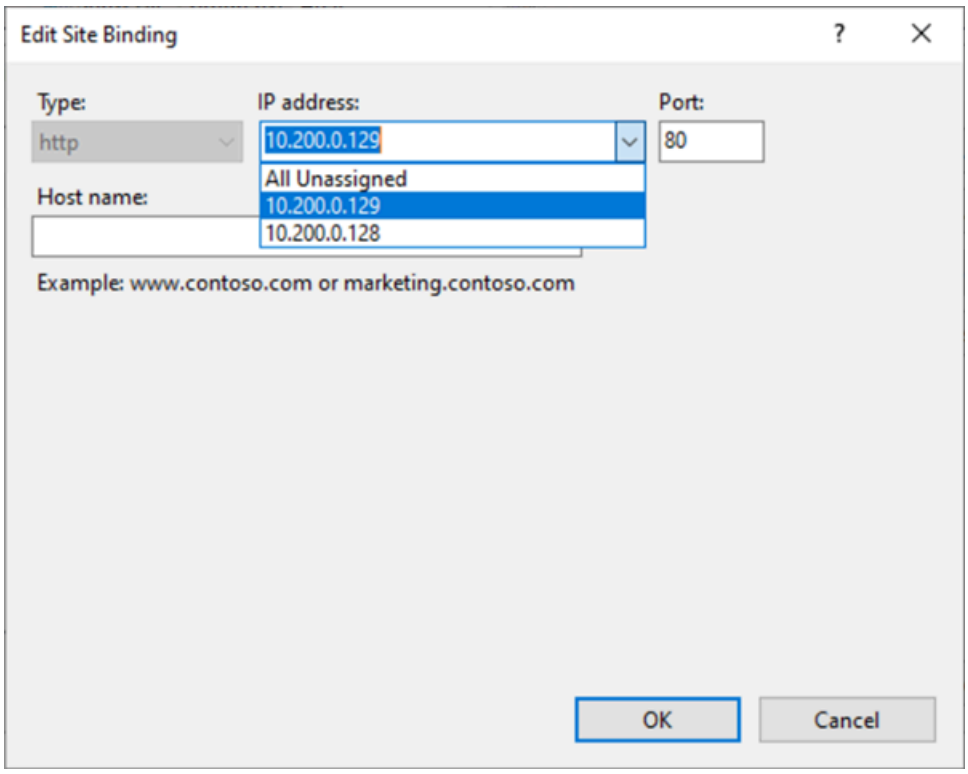
```
Subnet Mask . . . . . . . . . . . : 255.255.255.0
IPv4 Address. . . . . . . . . . . : 10.200.0.129(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 10.200.0.1
DHCPv6 IAID . . . . . . . . . . . : 151166011
DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-67-AC-FC-12-34-9A-BE-A5-E7
DNS Servers . . . . . . . . . . . : 10.200.0.2
NetBIOS over Tcpip. . . . . . . . : Enabled
```

步骤 3：配置应用程序以使用辅助私有 IP 地址

可配置任何应用程序来使用辅助私有 IP 地址。例如，如果您的实例在 IIS 上运行网站，则可以配置 IIS 使用辅助私有 IP 地址。

要配置 IIS 以使用辅助私有 IP 地址

1. 连接到您的实例。
2. 打开互联网信息服务 (IIS) 管理器。
3. 在“Connections”窗格，展开“Sites”。
4. 打开您网站的上下文（右键单击）菜单，选择编辑绑定。
5. 在网站绑定对话框中，对于类型，选择 http、编辑。
6. 在编辑网站绑定对话框中，对于 IP 地址，选择辅助私有 IP 地址。(默认情况下，各网站均可接受来自所有 IP 地址的 HTTP 请求。)



7. 选择确定、关闭。

EC2 实例主机名和域

了解 EC2 实例主机名和域对于有效管理和访问 Amazon EC2 实例非常重要。每个 EC2 实例都可以有不同类型的主机名（私有和公有），从而用于不同的用途并遵循特定的命名惯例。

本主题介绍了 EC2 实例主机名的结构、主机名的构造方式，以及可用的不同主机名类型。您将学习如何查看和修改主机名设置，了解何时使用各种类型的主机名，以及了解 Amazon 环境中主机名管理的最佳实践。

内容

- 了解 EC2 实例主机名和域
- 主机名类型

了解 EC2 实例主机名和域

EC2 实例地址由若干不同的部分组成。以下是一个使用实例私有 IPv4 地址的 EC2 实例地址示例：

```
IP address          Domain name
```

```
   #--------# #------------------------#
ip-10-24-34-0.us-west-2.compute.internal
#-----------#
  Hostname
#-------------------------------------#
    Fully qualified domain name (FQDN)
```

其中：

- IP 地址：与实例关联的主网络接口的主 IPv4 地址。
- 主机名：具体 EC2 实例的本地名称（由操作系统使用并用作本地网络标识）
- 域名：FQDN 中 Amazon 提供的部分
- 完全限定域名（FQDN）：包含主机名和域名的完整地址。这是用于通过网络访问实例的完整全球唯一标识符。

根据您为实例或该实例所挂载主网络接口选择的主机名类型，主机名和域名格式将与上面的示例不同。本节介绍主机名类型选项。

主机名类型

Amazon 提供的主机名分为私有和公用这两种类型。下表比较了私有主机名和公有主机名之间的主要区别，包括其解析方式、配置方式以及使用场景。

	私有主机名	公有主机名
DNS 解析	私有主机名用于无法从公共互联网访问的私有 FQDN。私有主机名仅允许将请求解析为 VPC 内的私有 IPv4 和 IPv6 GUA 地址。	公有主机名用于可从公共互联网访问的公有 FQDN。公有主机名允许将请求解析为 VPC 内的私有 IPv4 和 IPv6 GUA 以及来自互联网的公有 IP（水平分割 DNS）。
配置	私有主机名在实例级别配置。	公共主机名在网络接口级别配置。
何时使用	• 适用于仅在 VPC 内解析以用于内部通信的主机名 • 将内部命名方案保密	• 从 IPv4 和 IPv6 地址通过互联网访问实例 • 确保从 IPv4 环境顺利迁移到 IPv6 环境

	私有主机名	公有主机名
	• 简化内部资源管理，无需考虑外部 DNS 因素	

内容

- 公有主机名
- 私有主机名

公有主机名

您可以使用公有主机名来访问 EC2 实例，这些主机名将会解析为实例的公有 IPv4 或 IPv6 地址，以方便您过渡到 IPv6。

使用公有主机名时，您可以：

- 通过 IPv4 和 IPv6 访问 EC2 实例，从而提高资源连接方式的灵活性。
- 按照自己的节奏从 IPv4 环境迁移到 IPv6 环境。例如，您可以分开迁移数据库和应用程序，从而降低复杂性和风险。
- 使用多个主机名选项（仅 IPv4、仅 IPv6 和双栈），自动解析为相应的 IP 地址。
- 享受水平分割 DNS 提供的更好安全性。在您的 VPC 内进行查询时，系统会将主机名解析为私有 IP 地址。

内容

- 公有主机名的类型及使用场景
- 查看公有主机名
- 修改公有主机名类型

公有主机名的类型及使用场景

要使用公有主机名，必须修改现有的网络接口。本节介绍了三种公有主机名类型选项，并提供了有关使用决策的帮助：

- 双栈 – 基于 IP 的名称

- 如果您正在迁移或计划从 IPv4 迁移到 IPv6，那么此选项将非常适合。此选项同时支持通过 IPv4 和 IPv6 进行连接，为可能使用任一协议的客户端提供了灵活性，并使客户端能够在迁移到 IPv6 的整个过程中保持相同的主机名。
- 来自 VPC 内部的请求将会解析为网络接口的私有 IPv4 地址和 IPv6 全局单播地址（GUA）；而来自互联网的请求则会解析为网络接口的公有 IPv4 和 IPv6 GUA 地址。
- 示例
  - 选择此选项时，将为此网络接口生成一个双栈 FQDN。以下是将生成的 FQDN 示例：
    - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO.ap-southeast-2.ip.aws
  - 其中：
    - f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdpO 是主机名，即网络接口主公有 IPv6 地址（f5lnz-0khrm-nt2u3-gyqqt-nbdl5）的 base36 表示法加该网络接口主公有 IPv4 地址（q3cdpO）的 base36 表示法。
    - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 将由 Amazon DNS 解析器解析为 IPv6 GUA 地址 FFFF:1407:4:f000:81d:2689:1066:4489。这是分配给网络接口的第一个 IPv6 GUA。
    - q3cdpO 将会解析为 IPv4 地址 52.54.55.56。这是附加到主网络接口的公有 IPv4 地址。
    - ap-southeast-2 是网络接口所在子网所属区域。
    - ip.aws 是由 Amazon 提供的域。
- IPv6 – 基于 IP 的名称
  - 如果您已经迁移到 IPv6 并且只需要通过 IPv6 进行连接，那么此选项将非常适合。
  - VPC 内或来自互联网的请求都会解析为网络接口的 IPv6 GUA 地址。
  - 示例
    - 选择此选项时，将为此网络接口生成一个 FQDN。以下是将生成的 FQDN 示例：
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
    - 其中：
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 是主机名，即网络接口主公有 IPv6 地址的 base36 表示法。
      - f5lnz-0khrm-nt2u3-gyqqt-nbdl5 将由 Amazon DNS 解析器解析为 IPv6 GUA 地址 FFFF:1407:4:f000:81d:2689:1066:4489。这是分配给网络接口的第一个 IPv6 GUA。
      - ap-southeast-2 是网络接口所在子网所属区域。
      - ip.aws 是由 Amazon 提供的域。
- IPv4 – 基于 IP 的名称
  - 如果使用此网络接口的实例在过渡到 IPv6 期间需要继续使用 IPv4 访问，或者在该实例上运行的应用程序或系统仅支持 IPv4，则此选项将非常适合。如果只需要保持 IPv4 连接，并且工作负载不

需要 IPv6 支持，则此选项将非常适合。例如，假设要迁移到 IPv6，并且您决定某些应用程序保留 IPv4，而其他应用程序则迁移到 IPv6。

- 来自 VPC 内部的请求将会解析为网络接口的私有主 IPv4 地址；而来自互联网的请求则会解析为网络接口的公有 IPv4 地址。
- 示例
  - 选择此选项时，将为此网络接口生成一个支持 IPv4 的公有主机名。以下是将生成的 DNS 名称示例：
    - ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
  - 其中：
    - ec2-52-54-55-66 是主机名，即网络接口主公有 IPv4 地址的 base36 表示法。
    - ec2-52-54-55-66 将会解析为 IPv4 地址 52.54.55.56。这是附加到主网络接口的公有 IPv4 地址。
    - ap-southeast-2 是网络接口所在子网所属区域。
    - ip.aws 是由 Amazon 提供的域。

> **Important**
>
> 在上面的示例中，IP 地址被用于生成主机名。如果您更改分配给网络接口的主私有 IPv4 地址或第一个 IPv6 GUA，则主机名中转换为 IP 地址的部分也将改变，并且先前生成的公有主机名将不再有效。此外，更改主 IPv4 公有地址会强制下游刷新《Amazon EC2 用户指南》中所述的实例元数据服务（IMDS），因此 EC2 实例 IMDS 元数据也会自动更新。

查看公有主机名

如果网络接口所在的 VPC 未同时启用 EnableDnsHostnames 和 EnableDnsSupport，则不会定义或生成任何主机名类型。

Console

可查看实例或主网络接口的公有主机名。

查看实例的主机名类型和 DNS 名称

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。

3. 选中该实例的复选框。
4. 在网络选项卡上的主机名和 DNS下，查找以下内容：

   - 公共主机名类型
   - 公有 DNS
   - 仅 IPv4 - 基于 IP 的名称
   - 仅 IPv6 – 基于 IP 的名称
   - 双栈 – 基于 IP 的名称

查看网络接口的主机名类型和 DNS 名称

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 在搜索字段中，输入实例 ID。选择网络接口的 ID 以打开其详细信息页面。
4. 在主机名和 DNS下，查找以下内容：

   - 公共主机名类型
   - 公有 DNS 名称
   - 公有 IPv4 DNS 名称
   - 公有 IPv6 DNS 名称
   - 公有双栈 DNS 名称

Amazon CLI

查看网络接口的主机名类型和 DNS 名称

使用 describe-network-interfaces 命令。

```
aws ec2 describe-network-interfaces \
    --network-interface-id eni-1234567890abcdef0 \
    --query NetworkInterfaces[].PublicIpDnsNameOptions
```

下面是示例输出。由于主机名类型为 `public-dual-stack-dns-name`，因此 DNS 主机名与 `PublicDualStackDnsName` 相同。

```
[
```

```
    {
        "DnsHostnameType": "public-dual-stack-dns-name",
        "PublicIpv4DnsName": "ec2-52-54-55-66.ap-
southeast-2.compute.amazonaws.com",
        "PublicIpv6DnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-
southeast-2.ip.aws",
        "PublicDualStackDnsName": "f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-
southeast-2.ip.aws"
    }
]
```

PowerShell

查看网络接口的主机名类型和 DNS 名称

使用 Get-EC2NetworkInterface cmdlet。

```
(Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0).PublicIpDnsNameOptions
```

下面是示例输出。由于主机名类型为 public-dual-stack-dns-name，因此 DNS 主机名与 PublicDualStackDnsName 相同。

```
DnsHostnameType        : public-dual-stack-dns-name
PublicDualStackDnsName : f5lnz-0khrm-nt2u3-gyqqt-nbdl5-q3cdp0.ap-southeast-2.ip.aws
PublicIpv4DnsName      : ec2-52-54-55-66.ap-southeast-2.compute.amazonaws.com
PublicIpv6DnsName      : f5lnz-0khrm-nt2u3-gyqqt-nbdl5.ap-southeast-2.ip.aws
```

修改公有主机名类型

公有主机名类型选项取决于与网络接口关联的 IP 地址：

- 如果网络接口只有一个公有 IPv4 地址，则主机名类型必须为 IPv4 – 基于 IP 的名称。
- 如果网络接口只有一个 IPv6 地址，则主机名类型必须为 IPv6 – 基于 IP 的名称。
- 如果网络接口同时拥有一个公有 IPv4 地址和一个 IPv6 地址，则主机名类型可以为 双栈 – 基于 IP 的名称。

先决条件

- 网络接口必须具有关联的公有 IPv4 或 IPv6 地址。

- 网络接口所在的 VPC 必须已经启用了 EnableDnsHostnames 和 EnableDnsSupport 选项。请参阅《Amazon VPC 用户指南》中的查看和更新 VPC 的 DNS 属性。

Console

修改公共主机名类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 在搜索字段中，输入实例 ID。选中该网络接口的复选框。

   或者，在实例详细信息页面中选择网络选项卡，然后为设备索引 0 选择网络接口的 ID。
4. 选择操作、修改公有主机名类型。
5. 选择一个选项：

   - 双栈 – 基于 IP 的名称：网络接口的双栈公有主机名。来自 VPC 内部的请求将会解析为网络接口的私有 IPv4 地址和 IPv6 全局单播地址；而来自互联网的请求则会解析为网络接口的公有 IPv4 和 IPv6 GUA 地址。
   - IPv4 – 基于 IP 的名称：已启用 IPv4 的网络接口公有主机名。来自 VPC 内部的请求将会解析为网络接口的私有主 IPv4 地址；而来自互联网的请求则会解析为网络接口的公有 IPv4 地址。
   - IPv6 – 基于 IP 的名称：已启用 IPv6 的网络接口公有主机名。VPC 内或来自互联网的请求都会解析为网络接口的 IPv6 GUA 地址。
6. 选择 Modify(修改)。

Amazon CLI

修改公共主机名类型

使用 modify-public-ip-dns-name-options 命令。

```
aws ec2 modify-public-ip-dns-name-options \
    --network-interface-id eni-1234567890abcdef0 \
    --hostname-type public-dual-stack-dns-name
```

下面是示例输出。

```
{
```

```
    "Successful": true
}
```

PowerShell

修改公共主机名类型

使用 edit-EC2PublicIpDnsNameOption cmdlet。

```
Edit-EC2PublicIpDnsNameOption `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -HostNameType public-dual-stack-dns-name
```

私有主机名

本节介绍在 VPC 子网中启动实例时可用的 Amazon EC2 实例私有主机名。

私有主机名用来区分您网络中的 EC2 实例。例如，假设您想通过运行脚本来与网络上的部分或全部实例进行通信，则可以使用实例的私有主机名。

内容

- 私有主机名类型
- 在哪里可以找到资源名称和 IP 名称
- 在资源名称和 IP 名称之间进行选择
- 更改 Amazon EC2 基于资源的命名选项

私有主机名类型

在 VPC 中启动 EC2 实例时，来宾操作系统主机名有两种私有主机名类型：

- IP name（IP 名称）：传统命名方案，当您启动实例时，实例的私有 IPv4 地址包含在实例的主机名中。IP 名称在 EC2 实例的生命周期内存在。当用作私有 DNS 主机名时，它只会返回私有 IPv4 地址（A 记录）。
- Resource name（资源名称）：当您启动实例时，EC2 instance ID（EC2 实例 ID）包含在实例的主机名中。资源的名称在 EC2 实例的生命周期内存在。当用作私有 DNS 主机名时，它可以返回私有 IPv4 地址（A 记录）和/或 IPv6 全局单播地址（AAAA 记录）。

EC2 实例来宾操作系统主机名类型取决于子网设置：

- 如果在仅 IPv4 子网中启动实例，则可以选择 IP 名称或资源名称。
- 如果在双堆栈（IPv4+IPv6）子网中启动实例，则可以选择 IP 名称或资源名称。
- 如果在仅 IPv6 子网中启动实例，则会自动使用资源名称。

内容

- IP 名称
- 资源名称
- IP 名称和资源名称之间的区别

IP 名称

当您启动具有 IP name（IP 名称）的 Hostname type（主机名类型）的 EC2 实例时，来宾操作系统主机名将配置为使用私有 IPv4 地址。

- us-east-1 中的实例的格式：*private-ipv4-address*.ec2.internal
- 示例：*ip-10-24-34-0*.ec2.internal
- 任何其它Amazon区域中的实例的格式：*private-ipv4-address.region*.compute.internal
- 示例：*ip-10-24-34-0.us-west-2*.compute.internal

资源名称

当您在仅 IPv6 子网中启动 EC2 实例时，Resource name（资源名称）的 Hostname type（主机名类型）默认处于选中状态。当您在仅 IPv4 子网或双堆栈（IPv4+IPv6）子网中启动实例时，Resource name（资源名称）是您可以选择的选项。启动实例后，您可以管理主机名配置。有关更多信息，请参阅 更改 Amazon EC2 基于资源的命名选项。

当您启动具有 Resource name（资源名称）的Hostname type（主机名类型）的 EC2 实例时，来宾操作系统主机名将配置为使用 EC2 实例 ID。

- us-east-1 中的实例的格式：*ec2-instance-id*.ec2.internal
- 示例：*i-0123456789abcdef*.ec2.internal
- 任何其它Amazon区域中的实例的格式：*ec2-instance-id.region*.compute.internal
- 示例：*i-0123456789abcdef.us-west-2*.compute.internal

IP 名称和资源名称之间的区别

IP 名称和资源名称的 DNS 查询共存，以确保向后兼容并允许您从基于 IP 的主机名命名迁移到基于资源的命名。对于基于 IP 名称的私有 DNS 主机名，您无法配置是否响应实例的 DNS A 记录查询。无论客户操作系统主机名设置如何，DNS A 记录查询始终得到响应。相反，对于基于资源名称的私有 DNS 主机名，您可以配置是否响应实例的 DNS A 和/或 DNS AAAA 查询。当您启动实例或修改子网时，您可以配置响应行为。有关更多信息，请参阅 更改 Amazon EC2 基于资源的命名选项。

在哪里可以找到资源名称和 IP 名称

您可以在 Amazon EC2 控制台中查看主机名类型、资源名称和 IP 名称。

内容

- 创建 EC2 实例时
- 查看现有 EC2 实例的详细信息时

创建 EC2 实例时

当您创建 EC2 实例时，根据您选择的子网类型，Resource name（资源名称）的 Hostname type（主机名类型）可能是可用的，也可能已选中且无法修改。本节介绍您会在什么场景中看到主机名类型、资源名称和 IP 名称。

方案 1

您在向导中创建 EC2 实例（参阅 使用控制台中的启动实例向导来启动 EC2 实例），然后配置详细信息时，您选择配置为仅 IPv6 的子网。

在这种情况下，Resource name（资源名称）的 Hostname type（主机名类型）已自动选中且无法修改。将自动取消选择 Enable IP name IPv4 (A record) DNS requests（启用 IP 名称 IPv4（A 记录）DNS 请求）和 Enable resource-based IPv4 (A record) DNS requests（启用基于资源的 IPv4（A 记录）DNS 请求）的 DNS Hostname（DNS 主机名）选项且不可修改。Enable resource-based IPv6 (AAAA record) DNS requests（启用基于资源的 IPv6（AAAA 记录）DNS 请求）默认处于选中状态，但是可修改。如果选择该选项，对资源名称的 DNS 请求将解析为此 EC2 实例的 IPv6 地址（AAAA 记录）。

方案 2

您在向导中创建 EC2 实例（见 使用控制台中的启动实例向导来启动 EC2 实例），然后配置详细信息时，您选择配置为 IPv4 CIDR 块或同时配置 IPv4 和 IPv6 CIDR 块（“双堆栈”）的子网。

在此情况下，Enable IP name IPV4 (A record) DNS requests（启用 IP 名称 IPv4（A 记录）DNS 请求）已自动选中且无法更改。这意味着对 IP 名称的请求将解析为此 EC2 实例的 IPv4 地址（A 记录）。

选项默认为子网的配置，但是您可以根据子网设置修改此实例的选项：

- Hostname type（主机名类型）：确定您希望 EC2 实例的来宾操作系统主机名是资源名称还是 IP 名称。默认值是 IP name（IP 名称）。
- Enable resource-based IPv4 (A record) DNS requests（启用基于资源的 IPV4（A 记录）DNS 请求）：确定对您的资源名称的请求是否解析为此 EC2 实例的私有 IPv4 地址（A 记录）。此选项非默认选定。
- Enable resource-based IPV6 (AAAA record) DNS requests（启用基于资源的 IPV6（AAAA 记录）DNS 请求）：确定对您的资源名称的请求是否解析为此 EC2 实例的 IPv6 GUA 地址（AAAA 记录）。此选项非默认选定。

查看现有 EC2 实例的详细信息时

您可以在 EC2 实例的 Details（详细信息）选项卡中查看现有 EC2 实例的主机名值：

- Hostname type（主机名类型）：IP 名称或资源名称格式的主机名。
- Private IP DNS name (IPv4 only)（私有 IP DNS 名称（仅 IPv4））：将始终解析为实例的私有 IPv4 地址的 IP 名称。
- Private resource DNS name（私有资源 DNS 名称）：解析为为此实例选择的 DNS 记录的资源名称。
- Answer private resource DNS name（回答私有资源 DNS 名称）：资源名称解析为 IPv4（A）、IPv6（AAAA）或 IPv4 和 IPv6（A 和 AAAA）DNS 记录。

此外，如果您直接通过 SSH 连接到 EC2 实例，然后输入 `hostname` 命令，您将看到 IP 名称或资源名称格式的主机名。

在资源名称和 IP 名称之间进行选择

当您启动 EC2 实例时（请参阅使用控制台中的启动实例向导来启动 EC2 实例），如果选择 Resource name（资源名称）的 Hostname type（主机名类型），EC2 实例将使用资源名称格式的主机名启动。在这种情况下，此 EC2 实例的 DNS 记录也可以指向资源名称。这使您可以灵活地选择该主机名是解析为实例的 IPv4 地址、IPv6 地址，还是同时解析实例的 IPv4 和 IPv6 地址。如果您计划在未来使用 IPv6，或者如果您现在使用双堆栈子网，最好使用 Resource name（资源名称）的 Hostname

type（主机名类型），以便在不对 DNS 记录本身进行任何更改的情况下更改实例主机名的 DNS 解析。资源名称允许您在 EC2 实例上添加和删除 IPv4 和 IPv6 DNS 解析。

如果选择了 IP name（IP 名称）的 Hostname type（主机名类型），并将其用作 DNS 主机名，则只能解析为实例的 IPv4 地址。即使实例同时具有与之关联 IPv4 地址和 IPv6 地址，它也不会解析为实例的 IPv6 地址。

更改 Amazon EC2 基于资源的命名选项

您可以更改子网的主机名类型和 DNS 主机名配置，这会影响该主题中的所有后续实例启动，也可以在启动 EC2 实例后更改实例的这些内容。

基于资源的命名选项

- 主机名类型：确定在子网中启动的 EC2 实例的访客操作系统主机名的默认设置。这可以是资源名称，也可以是 IP 名称。
- 启用 DNS 主机名 IPv4（A 记录）请求：确定是否将对资源名称的 DNS 请求/查询解析为此 EC2 实例的私有 IPv4 地址（A 记录）。
- 启用 DNS 主机名 IPv6（AAAA 记录）请求：确定是否将对资源名称的 DNS 请求/查询解析为此 EC2 实例的 IPv6 地址（AAAA 记录）。

子网

更改子网设置不会更改子网中已启动的 EC2 实例的配置。

Console

修改子网的选项

打开 Amazon VPC 控制台并选择子网。依次选择操作、编辑子网设置。根据需要修改设置并保存更改。

Amazon CLI

修改子网的选项

使用 modify-subnet-attribute 命令。

```
aws ec2 modify-subnet-attribute \
    --subnet-id subnet-0abcdef1234567890 \
```

```
    --private-dns-hostname-type-on-launch resource-name \
    --enable-resource-name-dns-a-record-on-launch \
    --enable-resource-name-dns-aaaa-record-on-launch
```

PowerShell

修改子网的选项

使用 Edit-EC2SubnetAttribute cmdlet。

```
Edit-EC2SubnetAttribute `
    -SubnetId subnet-0abcdef1234567890 `
    -PrivateDnsHostnameTypeOnLaunch ResourceName `
    -EnableResourceNameDnsAAAARecordOnLaunch $true `
    -EnableResourceNameDnsARecordOnLaunch $true
```

EC2 实例

注意事项

- 要更改主机名类型，必须先停止实例。要更改其他两个选项，则无需停止实例。
- 因为无法停止具有实例存储根卷的实例，所以您只能在实例启动时配置主机名类型和 DNS 主机名选项。只有以下实例类型支持实例存储根卷：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

Console

修改实例的主机名类型和 DNS 主机名选项

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 如果您要更改 Use resource based naming as guest OS hostname（将基于资源的命名用作来宾操作系统主机名）设置，首先停止该 EC2 实例。否则，请跳过此步骤。

   要停止实例，请选择该实例，然后依次选择 Instance state（实例状态）、Stop instance（停止实例）。
3. 选择实例，然后依次选择 Actions（操作）、Instance settings（实例设置）、Change resource based naming options（更改基于资源的命名选项）。

   - Use resource based naming as guest OS hostname（将基于资源的命名用作来宾操作系统主机名）：确定您希望 EC2 实例的来宾操作系统主机名是资源名称还是 IP 名称。

- Answer DNS hostname IPv4 (A record) requests [回答 DNS 主机名 IPv4（A 记录）请求]：确定是否将对资源名称的 DNS 请求/查询解析为此 EC2 实例的私有 IPv4 地址。
- Answer DNS hostname IPv6 (AAAA record) requests（回答 DNS 主机名 IPv6（AAAA 记录）请求）：确定对您的资源名称的 DNS 请求/查询是否解析为此 EC2 实例的 IPv6 地址（AAAA 记录）。

4. 选择保存。
5. 如果您已经停止实例，可以重新启动它。

Amazon CLI

修改实例的主机名类型和 DNS 主机名选项

使用 modify-private-dns-name-options 命令。

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --private-dns-hostname-type resource-name \
    --enable-resource-name-dns-a-record \
    --enable-resource-name-dns-aaaa-record
```

PowerShell

修改实例的主机名类型和 DNS 主机名选项

使用 Edit-EC2PrivateDnsNameOption cmdlet。

```
Edit-EC2PrivateDnsNameOption `
    -InstanceId i-1234567890abcdef0 `
    -PrivateDnsHostnameType ResourceName `
    -EnableResourceNameDnsAAAARecord $true`
    -EnableResourceNameDnsARecord $true
```

自带 IP 地址（BYOIP）到 Amazon EC2

您可将自己的部分或全部可公开路由的 IPv4 或 IPv6 地址从本地网络引入到 Amazon Web Services 账户中。您可以继续控制地址范围，并且可以通过 Amazon 在互联网上公告地址范围。在将地址范围引入 Amazon EC2 中之后，它会在您的 Amazon Web Services 账户中显示为地址池。

> **Note**
>
> 本文档介绍如何将您自己的 IP 地址范围仅用于 Amazon EC2。要在 Amazon Global Accelerator 中使用自己的 IP 地址范围，请参阅《Amazon Global Accelerator 开发人员指南》中的 自带 IP 地址（BYOIP）。要将您自己的 IP 地址范围与 Amazon VPC IP Address Manager 结合使用，请参阅《Amazon VPC IPAM 用户指南》中的 教程：自带 IP 地址到 IPAM。

如果将 IP 地址范围带入 Amazon，Amazon 会验证您是否控制了 IP 地址范围。有两种方法可用于表明您控制了范围：

- 如果 IP 地址范围是在支持 RDAP 的互联网注册机构（例如 ARIN、RIPE 和 APNIC）注册，您可以通过此页面上的流程，使用 X.509 证书验证对域的控制。证书必须仅在预调配过程中有效。预调配完成后，可以从 RIR 的记录中删除证书。
- 无论互联网注册机构是否支持 RDAP，您都可以使用 Amazon VPC IPAM，通过 DNS TXT 记录验证对域的控制。该流程记录在《Amazon VPC IPAM 用户指南》中的教程：将 IP 地址带入 IPAM。

内容

- BYOIP 定义
- 要求和配额
- Amazon EC2 中 BYOIP 的先决条件
- 载入您在 Amazon EC2 中使用的地址范围
- 在 Amazon EC2 中使用您的 BYOIP 地址范围

BYOIP 定义

- X.509 自签名证书：最常用于加密和验证网络内数据的证书标准。这是 Amazon 用来验证从 RDAP 记录控制 IP 空间的证书。有关 X.509 证书的更多信息，请参阅 RFC 3280。
- 自治系统号（ASN）是一种全局唯一标识符，它定义了一组 IP 前缀，这些前缀由一个或多个维护单一、明确定义的路由策略的网络运营商运行。
- 区域互联网注册机构（RIR）是管理世界某个区域内 IP 地址和 ASN 的分配和注册的组织。

- 注册数据访问协议（RDAP）：用于查询 RIR 中当前注册数据的只读协议。查询的 RIR 数据库中的条目称为“RDAP 记录”。某些记录类型需要客户通过 RIR 提供的机制进行更新。这些记录将由 Amazon 查询以验证对 RIR 中地址空间的控制。
- 路由来源授权 (ROA)：由 RIR 创建的对象，供客户对特定自治系统中的 IP 公告进行身份验证。如需相关概览，请参阅 ARIN 网站上的路由来源授权 (ROA)。
- 本地互联网注册机构（LIR）是网络服务提供商等企业，从 RIR 为其客户分配 IP 地址数据块。

要求和配额

- 地址范围必须在您所在的区域互联网注册机构（RIR）注册。有关地理区域的任何策略，请咨询您的 RIR。BYOIP 目前支持在美国互联网号码注册机构（ARIN）、欧洲 IP 资源网络协调中心（RIPE）或亚太网络信息中心（APNIC）注册。它必须由企业或机构实体注册，而不能由个人注册。
- 您可以引入的最具体 IPv4 地址范围是 /24。
- 对于公开发布的 CIDR，可以引入的最具体 IPv6 地址范围是 /48；对于不公开发布的 CIDR，可以引入的最具体 IPv6 地址范围是 /60。
- 不公开发布的 CIDR 范围不需要 ROA，但 RDAP 记录仍需更新。
- 您可以将每个地址范围一次添加到一个 Amazon 区域中。
- 对于每个 Amazon 区域，您可以将总共 5 个 BYOIP IPv4 和 IPv6 地址范围引入到您的 Amazon 账户中。您无法使用服务限额控制台调整 BYOIP CIDR 的限额，但可以按照 Amazon Web Services 一般参考 中的 Amazon 服务限额所述，通过联系 Amazon 支持中心来请求提高限额。
- 您无法使用 Amazon RAM 与其它账户共享您的 IP 地址范围，除非您使用 Amazon VPC IP 地址管理器 (IPAM) 并将 IPAM 与 Amazon Organizations 集成。有关更多信息，请参阅 Amazon VPC IPAM 用户指南中的将 IPAM 与 Amazon Organizations 集成。
- IP 地址范围中的地址必须具有干净的历史记录。我们可能会调查 IP 地址范围的声誉，并保留权利以拒绝包含的 IP 地址具有不良声誉或与恶意行为关联的 IP 地址范围。
- 遗留地址空间是指在区域互联网注册机构（RIR）系统成立之前由互联网号码分配机构（IANA）的中央注册管理机构分配的 IPv4 地址空间，仍然需要相应的 ROA 对象。
- 如果为 LIR，常见做法是通过手动过程更新记录。这可能需要几天的时间来部署，具体取决于 LIR。
- 大型 CIDR 块需要单个 ROA 对象和 RDAP 记录。您可以使用单个对象和记录，将多个较小的 CIDR 块从该范围添加到 Amazon，甚至可以跨多个 Amazon 区域添加。
- Wavelength 区域或 Amazon Outposts 上不支持 BYOIP。
- 不要在 RADb 对 BYOIP 或任何其他 IRR 进行任何手动更改。BYOIP 将自动更新 RADb。任何包含 BYOIP ASN 的手动更改都将导致 BYOIP 预置操作失败。

- 将 IPv4 地址范围设置为 Amazon 后，您可以使用该范围内的所有 IP 地址，包括第一个地址（网络地址）和最后一个地址（广播地址）。

Amazon EC2 中 BYOIP 的先决条件

BYOIP 的载入过程分为两个阶段，您必须为此执行三个步骤。这些步骤与下图中描述的步骤相对应。我们在本文档中包含了手动操作步骤，但您的 RIR 可能会提供托管服务来帮助您完成这些步骤。

> **Tip**
>
> 本节中的任务需要 Linux 终端，可以使用 Linux、Amazon CloudShell 或 适用于 Linux 的 Windows 子系统来执行。

内容

- 概述
- 创建私有密钥并生成 X.509 证书
- 将 X.509 证书上传到 RIR 中的 RDAP 记录
- 在 RIR 中创建 ROA 对象

概述

准备阶段

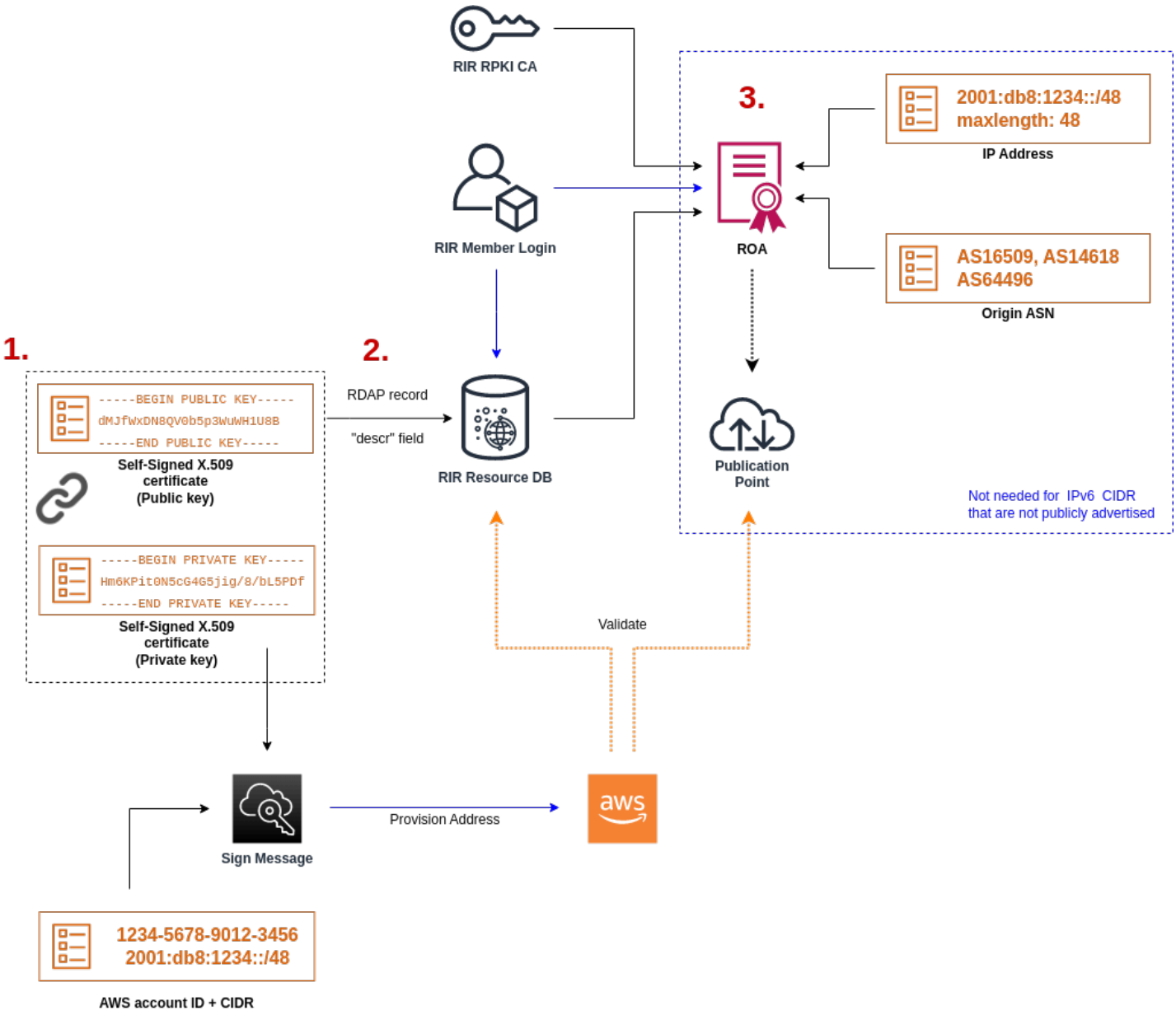
[1] 为进行身份验证，请创建私有密钥并用其生成自签名 X.509 证书。此证书仅用于预调配阶段。预调配完成后，可以从 RIR 的记录中删除证书

RIR 配置阶段

[2] 将自签名证书上传到 RDAP 记录注释。

[3] 在 RIR 中创建 ROA 对象。ROA 定义了所需的地址范围、允许公告地址范围的自治系统编号 (ASN) 以及向您 RIR 的资源公有密钥基础设施 (RPKI) 进行注册的到期日期。

> **Note**
>
> 不公开发布的 IPv6 地址空间不需要 ROA。



要引入多个不连续地址范围，您必须对每个地址范围重复此过程。不过，如果在几个不同 Amazon 区域之间拆分连续数据块，则无需执行重复准备和 RIR 配置步骤。

引入地址范围不会影响您之前引入的任何地址范围。

创建私有密钥并生成 X.509 证书

使用以下过程创建自签名 X.509 证书，并将其添加到您的 RIR 的 RDAP 记录。此密钥对用于向 RIR 对地址范围进行身份验证。openssl 命令需要 OpenSSL 版本 1.0.2 或更高版本。

复制以下命令并仅替换占位符值（以彩色斜体文本显示）。

此过程遵循加密私有 RSA 密钥并要求使用通行短语来访问该密钥的最佳实践。

1. 生成 RSA 2048 位私有密钥，如下所示。

```
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out
 private-key.pem
```

-aes256 参数指定用于加密私有密钥的算法。该命令返回以下输出，包括设置通行短语的提示：

```
......+++
.+++
Enter PEM pass phrase: xxxxxxx
Verifying - Enter PEM pass phrase: xxxxxxx
```

您可以使用以下命令检查密钥：

```
$ openssl pkey -in private-key.pem -text
```

这会返回通行短语提示和密钥内容，应类似于以下内容：

```
Enter pass phrase for private-key.pem: xxxxxxx
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDFBXHRI4HVKAhh
3seiciooizCRTbJe1+YsxNTja4XyKypVGIFWDGhZs44FCHlPOOSVJ+NqP74w96oM
7DPS3xo9kaQyZBFn2YEp2EBq5vf307KHNRmZZUmkn0zHOSEpNmY2fMxISBxewlxR
FAniwmSd/8TDvHJMY9FvAIvWuTsv5l0tJKk+a91K4+tO3UdDR7Sno5WXExfsBrW3
g1ydo3TBsx8i5/YiVOcNApy7ge2/FiwY3aCXJB6r6nuF6H8mRgI4r4vkMRsOlAhJ
DnZPNeweboo+K3Q3lwbgbmOKD/z9svk8N/+hUTBtIX0fRtbG+PLIw3xWRHGrMSn2
BzsPVuDLAgMBAAECggEACiJUj2hfJkKv47Dc3es3Zex67A5uDVjXmxfox2Xhdupn
fAcNqAptV6fXt0SPUNbhUxbBKNbshoJGufFwXPli1SXnpzvkdU4Hyco4zgbhXFsE
RNYjYfOGzTPwdBLpNMB6k3Tp4RHse6dNrlH0jDhpioL8cQEBdBJyVF5X0wymEbmV
mC0jgH/MxsBAPWW6ZKicg9ULMlWiAZ3MRAZPjHHgpYkAAsUWKAbCBwVQcVjGO59W
jfZjzTX5pQtVVH68ruciH88DTZCwjCkjBhxg+OIkJBLE5wkh82jIHSivZ63flwLw
z+E0+HhELSZJrn2MY6Jxmik3qNNUOF/Z+3msdj2luQKBgQDjwlC/3jxp8zJy6P8o
JQKv7TdvMwUj4VSWOHZBHLv4evJaaia0uQjIo1UDa8AYitqhX1NmCCehGH8yuXj/
v6V3CzMKDkmRr1NrONnSz5QsndQ04Z6ihAQlPmJ96g4wKtgoC7AYpyP0g1a+4/sj
b1+o3YQI4pD/F71c+qaztH7PRwKBgQDdc23yNmT3+Jyptf0fKjEvONK+xwUKzi9c
L/OzBq5yOIC1Pz2T85gOe1i8kwZws+xlpG6uBT6lmIJELd0k59FyupNu4dPvX5SD
6GGqdx4jk9KvI74usGeOBohmF0phTHkrWKBxXiyT0oS8zjnJlEn8ysIpGgO28jjr
LpaHNZ/MXQKBgQDfLNcnS0LzpsS2aK0tzyZU8SMyqVHOGMxj7quhneBq2T6FbiLD
T9TVlYaGNZ0j71vQaLI19qOubWymbautH0Op5KV8owdf4+bf1/NJaPIOzhDUSIjD
```

```
Qo01WW31Z9XDSRhKFTnWzmCjBdeIcajyzf10YKsycaAW9lItu8aBrMndnQKBgQDb
nNp/JyRwqjOrNljk7DHEs+SD39kHQzzCfqd+dnTPv2sc06+cpym3yulQcbokULpy
fmRo3bin/pvJQ3aZX/Bdh9woTXqhXDdrrSwWInVYMQPyPk8f/D9mIOJp5FUWMwHD
U+whIZSxsEeE+jtixlWtheKRYkQmzQZXbWdIhYyI3QKBgD+F/6wcZ85QW8nAUykA
3WrSIx/3cwDGdm4NRGct8ZOZjTHjiy9ojMOD1L7iMhRQ/3k3hUsin5LDMp/ryWGG
x4uIaLat40kiC7T4I66DM7P59euqdz3w0PD+VU+h7GSivvsFDdySUt7bNK0AUVLh
dMJfWxDN8QV0b5p3WuWH1U8B
-----END PRIVATE KEY-----
Private-Key: (2048 bit)
modulus:
    00:c5:05:71:d1:23:81:d5:28:08:61:de:c7:a2:72:
    2a:28:8b:30:91:4d:b2:5e:d7:e6:2c:c4:d4:e3:6b:
    85:f2:2b:2a:55:18:81:56:0c:68:59:b3:8e:05:08:
    79:4f:38:e4:95:27:e3:6a:3f:be:30:f7:aa:0c:ec:
    33:d2:df:1a:3d:91:a4:32:64:11:67:d9:81:29:d8:
    40:6a:e6:f7:f7:d3:b2:87:35:19:99:65:49:a4:9f:
    4c:c7:39:21:29:36:66:36:7c:cc:48:48:1c:5e:c2:
    5c:51:14:09:e2:c2:64:9d:ff:c4:c3:bc:72:4c:63:
    d1:6f:00:8b:d6:b9:3b:2f:e6:5d:2d:24:a9:3e:6b:
    dd:4a:e3:eb:4e:dd:47:43:47:b4:a7:a3:95:97:13:
    17:ec:06:b5:b7:83:5c:9d:a3:74:c1:b3:1f:22:e7:
    f6:22:54:e7:0d:02:9c:bb:81:ed:bf:16:2c:18:dd:
    a0:97:24:1e:ab:ea:7b:85:e8:7f:26:46:02:38:af:
    8b:e4:31:1b:0e:94:08:49:0e:76:4f:35:ec:1e:6e:
    8a:3e:2b:74:37:97:06:e0:6e:63:8a:0f:fc:fd:b2:
    f9:3c:37:ff:a1:51:30:6d:21:7d:1f:46:d6:c6:f8:
    f2:c8:c3:7c:56:44:71:ab:31:29:f6:07:3b:0f:56:
    e0:cb
publicExponent: 65537 (0x10001)
privateExponent:
    0a:22:54:8f:68:5f:26:42:af:e3:b0:dc:dd:eb:37:
    65:ec:7a:ec:0e:6e:0d:58:d7:9b:17:e8:c7:65:e1:
    76:ea:67:7c:07:0d:a8:0a:6d:57:a7:d7:b7:44:8f:
    50:d6:e1:53:16:c1:28:d6:ec:86:82:46:b9:f1:70:
    5c:f9:62:d5:25:e7:a7:3b:e4:75:4e:07:c9:ca:38:
    ce:06:e1:5c:5b:04:44:d6:23:61:f3:86:cd:33:f0:
    74:12:e9:34:c0:7a:93:74:e9:e1:11:ec:7b:a7:4d:
    ae:51:f4:8c:38:69:8a:82:fc:71:01:01:74:12:72:
    54:5e:57:d3:0c:a6:11:b9:95:98:2d:23:80:7f:cc:
    c6:c0:40:3d:65:ba:64:a8:9c:83:d5:0b:32:55:a2:
    01:9d:cc:44:06:4f:8c:71:e0:a5:89:00:02:c5:16:
    28:06:c2:07:05:50:71:58:c6:3b:9f:56:8d:f6:63:
    cd:35:f9:a5:0b:55:54:7e:bc:ae:e7:22:1f:cf:03:
    4d:90:b0:8c:29:23:06:1c:60:f8:e2:24:24:12:c4:
```

```
    e7:09:21:f3:68:c8:1d:28:af:67:ad:df:97:02:f0:
    cf:e1:34:f8:78:44:2d:26:49:ae:7d:8c:63:a2:71:
    9a:29:37:a8:d3:54:38:5f:d9:fb:79:ac:76:3d:a5:
    b9
prime1:
    00:e3:c2:50:bf:de:3c:69:f3:32:72:e8:ff:28:25:
    02:af:ed:37:6f:33:05:23:e1:54:96:38:76:41:1c:
    bb:f8:7a:f2:5a:6a:26:b4:b9:08:c8:a3:55:03:6b:
    c0:18:8a:da:a1:5f:53:66:08:27:a1:18:7f:32:b9:
    78:ff:bf:a5:77:0b:33:0a:0e:49:91:af:53:6b:38:
    d9:d2:cf:94:2c:9d:d4:34:e1:9e:a2:84:04:25:3e:
    62:7d:ea:0e:30:2a:d8:28:0b:b0:18:a7:23:f4:83:
    56:be:e3:fb:23:6f:5f:a8:dd:84:08:e2:90:ff:17:
    bd:5c:fa:a6:b3:b4:7e:cf:47
prime2:
    00:dd:73:6d:f2:36:64:f7:f8:9c:a9:b5:fd:1f:2a:
    31:2f:38:d2:be:c7:05:0a:ce:2f:5c:2f:f3:b3:06:
    ae:72:38:80:b5:3f:3d:93:f3:98:0e:7b:58:bc:93:
    06:70:b3:ec:65:a4:6e:ae:05:3e:a5:98:82:44:2d:
    dd:24:e7:d1:72:ba:93:6e:e1:d3:ef:5f:94:83:e8:
    61:aa:77:1e:23:93:d2:af:23:be:2e:b0:67:8e:06:
    88:66:17:4a:61:4c:79:2b:58:a0:71:5e:2c:93:d2:
    84:bc:ce:39:c9:94:49:fc:ca:c2:29:1a:03:b6:f2:
    38:eb:2e:96:87:35:9f:cc:5d
exponent1:
    00:df:2c:d7:27:4b:42:f3:a6:c4:b6:68:ad:2d:cf:
    26:54:f1:23:32:a9:51:ce:18:cc:63:ee:ab:a1:9d:
    e0:6a:d9:3e:85:6e:22:c3:4f:d4:d5:95:86:86:35:
    9d:23:ef:5b:d0:68:b2:35:f6:a3:ae:6d:6c:a6:6d:
    ab:ad:1f:43:a9:e4:a5:7c:a3:07:5f:e3:e6:df:d7:
    f3:49:68:f2:0e:ce:10:d4:48:88:c3:42:8d:35:59:
    6d:f5:67:d5:c3:49:18:4a:15:39:d6:ce:60:a3:05:
    d7:88:71:a8:f2:cd:fd:74:60:ab:32:71:a0:16:f6:
    52:2d:bb:c6:81:ac:c9:dd:9d
exponent2:
    00:db:9c:da:7f:27:24:70:aa:33:ab:36:58:e4:ec:
    31:c4:b3:e4:83:df:d9:07:43:3c:c2:7e:a7:7e:76:
    74:cf:bf:6b:1c:d3:af:9c:a7:29:b7:ca:e9:50:71:
    ba:24:50:ba:72:7e:64:68:dd:b8:a7:fe:9b:c9:43:
    76:99:5f:f0:5d:87:dc:28:4d:7a:a1:5c:37:6b:ad:
    2c:16:22:75:58:31:03:f2:3e:4f:1f:fc:3f:66:20:
    e2:69:e4:55:16:33:01:c3:53:ec:21:21:94:b1:b0:
    47:84:fa:3b:62:c6:55:ad:85:e2:91:62:44:26:cd:
    06:57:6d:67:48:85:8c:88:dd
```

```
coefficient:
    3f:85:ff:ac:1c:67:ce:50:5b:c9:c0:53:29:00:dd:
    6a:d2:23:1f:f7:73:00:c6:76:6e:0d:44:67:2d:f1:
    93:99:8d:31:e3:8b:2f:68:8c:c3:83:d4:be:e2:32:
    14:50:ff:79:37:85:4b:22:9f:92:c3:32:9f:eb:c9:
    61:86:c7:8b:88:68:b6:ad:e3:49:22:0b:b4:f8:23:
    ae:83:33:b3:f9:f5:eb:aa:77:3d:f0:d0:f0:fe:55:
    4f:a1:ec:64:a2:be:fb:05:0d:dc:92:52:de:db:34:
    ad:00:51:52:e1:74:c2:5f:5b:10:cd:f1:05:74:6f:
    9a:77:5a:e5:87:d5:4f:01
```

当您的私有密钥不用时，请将其保存在安全位置。

2. 使用上一步中创建的私有密钥生成 X.509 证书。在此示例中，该证书在 365 天后过期，在此日期后它将不能是受信任的。请务必相应地设置到期时间。证书必须仅在预调配过程中有效。预调配完成后，可以从 RIR 的记录中删除证书。`tr -d "\n"` 命令从输出中删除换行符。您需要在出现提示时提供公用名称，但其他字段可以留空。

```
$ openssl req -new -x509 -key private-key.pem -days 365 | tr -d "\n" >
 certificate.pem
```

这会产生类似于下述信息的输出：

```
Enter pass phrase for private-key.pem: xxxxxxx
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:example.com
Email Address []:
```

> **Note**
>
> Amazon 预置不需要公用名。其可以是任何内部或公有域名。

您可以使用以下命令检查证书：

```
$ cat certificate.pem
```

输出应该是一个没有换行符的 PEM 编码的长字符串，前面是 `-----BEGIN CERTIFICATE-----`，后面是 `-----END CERTIFICATE-----`。

将 X.509 证书上传到 RIR 中的 RDAP 记录

将之前创建的证书添加到 RIR 的 RDAP 记录。请务必在编码部分前后包含 `-----BEGIN CERTIFICATE-----` 和 `-----END CERTIFICATE-----` 字符串。所有这些内容必须都在单个长线上。更新 RDAP 的过程取决于您的 RIR：

- 对于 ARIN，使用客户经理门户，在代表您地址范围的“网络信息”对象的“公共注释”部分中添加证书。请勿将其添加到您组织的注释部分。
- 对于 RIPE，将证书作为新的“descr”字段添加到代表您地址范围的“inetnum”或“inet6num”对象中。通常可在 RIPE 数据库门户的“我的资源”部分中了解到相关信息。请勿将其添加到您所在组织的注释部分或上述对象的“备注”字段中。
- 对于 APNIC，通过编辑“inetnum”或“inet6num”记录上的备注来添加证书。

完成下方预调配阶段后，可从 RIR 的记录中删除证书。

在 RIR 中创建 ROA 对象

创建 ROA 对象以授权 Amazon ASN 16509 和 14618 来公告您的地址范围，以及当前授权的 ASN 来发布该地址范围。对于 Amazon GovCloud (US) Regions，请授权 ASN 8987 而不是 16509 和 14618。您必须将最大长度设置为您要引入的 CIDR 的大小。您可以引入的最具体 IPv4 前缀是 /24。对于公开发布的 CIDR，可以引入的最具体 IPv6 地址范围是 /48；对于不公开发布的 CIDR，可以引入的最具体 IPv6 地址范围是 /60。

> **Important**
>
> 如果您正在为 Amazon VPC IP Address Manager (IPAM) 创建 ROA 对象，则在创建 ROA 时，对于 IPv4 CIDR，必须将 IP 地址前缀的最大长度设置为 /24。对于 IPv6 CIDR，如果要将它们添加到可传播池中，IP 地址前缀的最大长度必须为 /48。这可以确保您有充分的灵活性来跨 Amazon 区域划分您的公有 IP 地址。IPAM 强制执行您设置的最大长度。有关通向 IPAM 的 BYOIP 地址的更多信息，请参阅《Amazon VPC IPAM 用户指南》中的教程：通向 IPAM 的 BYOIP 地址 CIDR。

若要 ROA 可用于 Amazon，可能需要多达 24 小时的时间。有关更多信息，请参阅 RIR：

- ARIN — ROA 请求
- RIPE — 管理 ROA
- APNIC - 路由管理

将公告从本地工作负载迁移到 Amazon 时，在为 Amazon 的 ASN 创建 ROA 之前，您必须为现有的 ASN 创建 ROA。否则，您可能会看到对现有路由和公告的影响。

> **Important**
>
> 为了让 Amazon 能够公告并持续公告您的 IP 地址范围，您使用 Amazon ASN 的 ROA 必须符合上述指南。如果 ROA 无效或不符合上述指南，则 Amazon 保留停止公告您的 IP 地址范围的权利。

> **Note**
>
> 不公开发布的 IPv6 地址空间不需要完成此步骤。

载入您在 Amazon EC2 中使用的地址范围

根据您的需求，BYOIP 的载入过程包括以下任务。

任务

- 在 Amazon 中预置公开发布的地址范围

- 预置不公开发布的 IPv6 地址范围
- 通过 Amazon 公告地址范围
- 取消预配置地址范围
- 验证 BYOIP

在 Amazon 中预置公开发布的地址范围

预置地址范围与 Amazon 配合使用时，您需要确认您控制该地址范围，并授权 Amazon 公告该地址范围。我们还会通过签名授权消息验证您控制该地址范围。该消息是使用在通过 X.509 证书更新 RDAP 记录时使用的自签名 X.509 密钥对签名的。Amazon 需要提供给 RIR 的加密签名授权消息。RIR 根据您添加到 RDAP 的证书验证签名，并根据 ROA 检查授权详细信息。

要预置地址范围

1. 撰写消息

   撰写明文授权消息。消息的格式如下所示，其中，日期是消息的到期日期：

   ```
   1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
   ```

   将账号、地址范围和到期日期替换为您自己的值，以创建类似于以下内容的消息：

   ```
   text_message="1|aws|0123456789AB|198.51.100.0/24|20211231|SHA256|RSAPSS"
   ```

   这不能与外观类似的 ROA 消息混淆。

2. 对消息进行签名

   使用之前创建的私有密钥对明文消息进行签名。此命令返回的签名是一个长字符串，您需要在下一步中使用。

   > **Important**
   >
   > 我们建议您复制并粘贴此命令。除消息内容之外，不要修改或替换任何值。

```
signed_message=$( echo -n $text_message | openssl dgst -sha256 -sigopt
 rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private-key.pem -keyform PEM
 | openssl base64 | tr -- '+=/' '-_~' | tr -d "\n")
```

3. 预置地址

   使用 Amazon CLI provision-byoip-cidr 命令预置地址范围。`--cidr-authorization-context` 选项使用之前创建的消息和签名字符串。

   > **Important**
   >
   > 如果 BYOIP 范围与 Configure the Amazon CLI `Default region name` 不同，则您必须指定应在其中预置 BYOIP 范围的 Amazon 区域。

   ```
   aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
    Message="$text_message",Signature="$signed_message" --region us-east-1
   ```

   预配置地址范围是一项异步操作，因此该调用会立即返回，但地址范围未准备就绪，直到其状态从 `pending-provision` 更改为 `provisioned` 才可供使用。

4. 监控进度

   虽然大多数预配置将在两小时内完成，但对于公开发布的范围，完成预配置过程可能需要一周时间。使用 describe-byoip-cidrs 命令来监视进度，如下例所示：

   ```
   aws ec2 describe-byoip-cidrs --max-results 5 --region us-east-1
   ```

   如果在预置过程中出现问题，并且状态变为 `failed-provision`，则您必须在解决问题后再次运行 `provision-byoip-cidr` 命令。

预置不公开发布的 IPv6 地址范围

默认情况下，将预置一个地址范围以公开发布到互联网上。您可以预置不公开发布的 IPv6 地址范围。对于不可公开通告的路由，预置过程通常在几分钟内完成。如果将非公有地址范围的 IPv6 CIDR 块与 VPC 关联，则只能通过支持 IPv6 的混合连接选项（Amazon Direct Connect、Amazon Site-to-Site VPN 或 Amazon VPC Transit Gateway）访问 IPv6 CIDR。

不需要使用 ROA 以预调配非公有地址范围。

> **Important**
>
> - 只能指定在预置期间是否公开发布地址范围。您以后无法更改可发布状态。
> - Amazon VPC 不支持唯一本地地址（ULA）CIDR。所有 VPC 都必须具有唯一的 IPv6 CIDR。两个 VPC 不能具有相同的 IPv6 网域范围。

要预置不公开发布的 IPv6 地址范围，请使用以下 provision-byoip-cidr 命令。

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context
 Message="$text_message",Signature="$signed_message" --no-publicly-advertisable --
region us-east-1
```

通过 Amazon 公告地址范围

预配置地址范围后，即可对其进行公告。您必须公告预配置的确切地址范围。您不能只公告预配置的地址范围的一部分。

如果您预置不公告的 IPv6 地址范围，则无需完成该步骤。

在通过 Amazon 公告地址范围之前，建议您停止从其他位置公告该地址范围或该范围的任何部分。如果您一直从其他位置公告 IP 地址范围或该范围的任何部分，我们将无法为其提供可靠支持或解决问题。具体来说，我们无法保证到地址范围或该范围的一部分的流量进入我们的网络。

为最大限度地减少停机时间，您可以在公告之前将 Amazon 资源配置为使用地址池中的某一地址，然后停止从当前位置公告该地址并同时开始通过 Amazon 公告该地址。有关从地址池分配弹性 IP 地址的更多信息，请参阅分配弹性 IP 地址。

限制

- 您最多只能每 10 秒运行一次 advertise-byoip-cidr 命令，即使每次指定不同的地址范围也是如此。
- 您最多只能每 10 秒运行一次 withdraw-byoip-cidr 命令，即使每次指定不同的地址范围也是如此。

要公告地址范围，请使用以下 advertise-byoip-cidr 命令。

```
aws ec2 advertise-byoip-cidr --cidr address-range --region us-east-1
```

要停止公告地址范围，请使用以下 withdraw-byoip-cidr 命令。

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

> **Note**
>
> 在 eu-central-1-ist-1 本地区域中，公告和撤回是异步操作。操作完成之前，地址范围进入 pending-advertising 或 pending-withdrawal 状态。有关更多信息，请参阅本地区域可用性。

取消预配置地址范围

要停止在 Amazon 上使用您的地址范围，请先释放任何弹性 IP 地址，并取消关联仍从地址池中分配的任何 IPv6 CIDR 块。接下来，停止公告该地址范围，并最终取消预置该地址范围。

您无法取消预置地址范围的一部分。如果要在 Amazon 上使用更具体的地址范围，请取消预置整个地址范围，并预置一个更具体的地址范围。

(IPv4) 要释放每个弹性 IP 地址，请使用以下 release-address 命令。

```
aws ec2 release-address --allocation-id eipalloc-12345678abcabcabc --region us-east-1
```

(IPv6) 要取消关联 IPv6 CIDR 块，请使用以下 disassociate-vpc-cidr-block 命令。

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-12345abcd1234abc1
 --region us-east-1
```

要停止公告地址范围，请使用以下 withdraw-byoip-cidr 命令。

```
aws ec2 withdraw-byoip-cidr --cidr address-range --region us-east-1
```

要取消预置地址范围，请使用以下 deprovision-byoip-cidr 命令。

```
aws ec2 deprovision-byoip-cidr --cidr address-range --region us-east-1
```

取消预置地址范围最多可能需要一天的时间。

验证 BYOIP

1. 验证自签名 x.509 密钥对

   通过 whois 命令验证证书是否已上传且有效。

   对于 ARIN，使用 `whois -h whois.arin.net r + 2001:0DB8:6172::/48` 查找地址范围的 RDAP 记录。检查命令输出中 NetRange（网络范围）的 Public Comments 部分。应将证书添加到地址范围的 Public Comments 部分。

   您可以使用以下命令检查包含证书的 Public Comments：

   ```
   whois -h whois.arin.net r + 2001:0DB8:6172::/48 | grep Comments | grep BEGIN
   ```

   这会返回包含密钥内容的输出，应类似于以下内容：

   ```
   Public Comments:
   -----BEGIN CERTIFICATE-----
   MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
   LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
   hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
   wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
   NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
   rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
   VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
   W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
   R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
   prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
   mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
   Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
   xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
   3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
   AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
   QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
   08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
   Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
   UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
   ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
   MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
   -----END CERTIFICATE-----
   ```

对于 RIPE，使用 `whois -r -h whois.ripe.net` *2001:0DB8:7269::/48* 查找地址范围的 RDAP 记录。检查命令输出中 `inetnum` 对象（网络范围）的 `descr` 部分。应将证书添加为地址范围的新 `descr` 字段。

您可以使用以下命令检查包含证书的 `descr`：

```
whois -r -h whois.ripe.net 2001:0DB8:7269::/48 | grep descr | grep BEGIN
```

这会返回包含密钥内容的输出，应类似于以下内容：

```
descr:
-----BEGIN CERTIFICATE-----MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8
RDAHSP+I1TowDQYJKoZIhvcNAQELBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAg
MCEF1Y2tsYW5kMREwDwYDVQQHDAhBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIF
dlYiBTZXJ2aWNlczETMBEGA1UECwwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPS
VAgRGVtbzAeFw0yMTEyMDcyMDI0NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNV
BAYTAk5aMREwDwYDVQQIDAhBdWNrbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDA
aBgNVBAoME0FtYXpvbiBXZWIgU2VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW
8xEzARBgNVBAMMCkJZT0lQIERlbW8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQCfmacvDp0wZ0ceiXXcR/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanA
EskgAseyFypwEEQr4CJijI/5hp9prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3
stsI5QesHVRwOaXUdprAnndaTugmDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq
35wU/x+wXlAqBXg4MZK2KoUu27kYt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp
1ZnVIc7NqnhdeIW48QaYjhMlUEfxdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2r
GlHWkJsbhr0VEUyAGu1bwkgcdww3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBS
tFyujN6SYBr2glHpGt0XGF7GbGTAfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0
XGF7GbGTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6Y
Lhz521lfyVfxY0t6o3410bQAeAF08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyL
xngwMYN0XY5tVhDQqk4/gmDNEKSZy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9
wySL507XQz76Uk5cFypBOzbnk35UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8
mBGqVpPpey+dXpzzzv1iBKN/VY4ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGC
vRDl/qdO/GIDJi77dmZWkh/ic90MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoN
PyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

对于 APNIC，使用 `whois -h whois.apnic.net` *2001:0DB8:6170::/48* 查找 BYOIP 地址范围的 RDAP 记录。检查命令输出中 `inetnum` 对象（网络范围）的 `remarks` 部分。应将证书添加为地址范围的新 `remarks` 字段。

您可以使用以下命令检查包含证书的 `remarks`：

```
whois -h whois.apnic.net 2001:0DB8:6170::/48 | grep remarks | grep BEGIN
```

这会返回包含密钥内容的输出，应类似于以下内容：

```
remarks:
-----BEGIN CERTIFICATE-----
MIID1zCCAr+gAwIBAgIUBkRPNSLrPqbRAFP8RDAHSP+I1TowDQYJKoZIhvcNAQE
LBQAwezELMAkGA1UEBhMCTloxETAPBgNVBAgMCEF1Y2tsYW5kMREwDwYDVQQHDA
hBdWNrbGFuZDEcMBoGA1UECgwTQW1hem9uIFdlYiBTZXJ2aWNlczETMBEGA1UEC
wwKQllPSVAgRGVtbzETMBEGA1UEAwwKQllPSVAgRGVtbzAeFw0yMTEyMDcyMDI0
NTRaFw0yMjEyMDcyMDI0NTRaMHsxCzAJBgNVBAYTAk5aMREwDwYDVQQIDAhBdWN
rbGFuZDERMA8GA1UEBwwIQXVja2xhbmQxHDAaBgNVBAoME0FtYXpvbiBXZWIgU2
VydmljZXMxEzARBgNVBAsMCkJZT0lQIERlbW8xEzARBgNVBAMMCkJZT0lQIERlb
W8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCfmacvDp0wZ0ceiXXc
R/q27mHI/U5HKt7SST4X2eAqufR9wXkfNanAEskgAseyFypwEEQr4CJijI/5hp9
prh+jsWHWwkFRoBRR9FBtwcU/45XDXLga7D3stsI5QesHVRwOaXUdprAnndaTug
mDPkD0vrl475JWDSIm+PUxGWLy+60aBqiaZq35wU/x+wXlAqBXg4MZK2KoUu27k
Yt2zhmy0S7Ky+oRfRJ9QbAiSu/RwhQbh5Mkp1ZnVIc7NqnhdeIW48QaYjhMlUEf
xdaqYUinzz8KpjfADZ4Hvqj9jWZ/eXo/9b2rGlHWkJsbhr0VEUyAGu1bwkgcdww
3A7NjOxQbAgMBAAGjUzBRMB0GA1UdDgQWBBStFyujN6SYBr2glHpGt0XGF7GbGT
AfBgNVHSMEGDAWgBStFyujN6SYBr2glHpGt0XGF7GbGTAPBgNVHRMBAf8EBTADA
QH/MA0GCSqGSIb3DQEBCwUAA4IBAQBX6nn6YLhz521lfyVfxY0t6o3410bQAeAF
08ud+ICtmQ4IO4A4B7zV3zIVYr0clrOOaFyLxngwMYN0XY5tVhDQqk4/gmDNEKS
Zy2QkX4Eg0YUWVzOyt6fPzjOvJLcsqc1hcF9wySL507XQz76Uk5cFypBOzbnk35
UkWrzA9KK97cXckfIESgK/k1N4ecwxwG6VQ8mBGqVpPpey+dXpzzzv1iBKN/VY4
ydjgH/LBfdTsVarmmy2vtWBxwrqkFvpdhSGCvRDl/qdO/GIDJi77dmZWkh/ic90
MNk1f38gs1jrCj8lThoar17Uo9y/Q5qJIsoNPyQrJRzqFU9F3FBjiPJF
-----END CERTIFICATE-----
```

2. 验证 ROA 对象创建

使用 RIPEstat Data API 验证 ROA 对象是否成功创建。请务必根据 Amazon ASN 16509 和 14618 以及当前授权公告地址范围的 ASN 来测试地址范围。

您可以使用以下命令，通过地址范围检查不同 Amazon ASN 的 ROA 对象：

```
curl --location --request GET "https://stat.ripe.net/data/rpki-validation/data.json?
resource=ASN&prefix=CIDR
```

在此示例输出中，Amazon ASN 16509 的响应结果为 `"status": "valid"`。这表示该地址范围的 ROA 对象已成功创建：

```
{
    "messages": [],
    "see_also": [],
    "version": "0.3",
    "data_call_name": "rpki-validation",
    "data_call_status": "supported",
    "cached": false,
    "data": {
        "validating_roas": [
            {
                "origin": "16509",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "valid"
            },
            {
                "origin": "14618",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            },
            {
                "origin": "64496",
                "prefix": "2001:0DB8::/32",
                "max_length": 48,
                "validity": "invalid_asn"
            }
        ],
        "status": "valid",
        "validator": "routinator",
        "resource": "16509",
        "prefix": "2001:0DB8::/32"
    },
    "query_id": "20230224152430-81e6384e-21ba-4a86-852a-31850787105f",
    "process_time": 58,
    "server_id": "app116",
    "build_version": "live.2023.2.1.142",
    "status": "ok",
    "status_code": 200,
    "time": "2023-02-24T15:24:30.773654"
}
```

“unknown” 状态表示该地址范围的 ROA 对象尚未创建。“invalid_asn” 状态表示该地址范围的 ROA 对象尚未成功创建。

在 Amazon EC2 中使用您的 BYOIP 地址范围

您可以查看和使用在您的账户中预置的 IPv4 和 IPv6 地址范围。有关更多信息，请参阅 the section called “载入您的地址范围”。

IPv4 地址范围

您可以从 IPv4 地址池中创建弹性 IP 地址，并将其用于您的 Amazon 资源，如 EC2 实例、NAT 网关和 Network Load Balancer。

要查看有关在您的账户中预置的 IPv4 地址池的信息，请使用以下 describe-public-ipv4-pools 命令。

```
aws ec2 describe-public-ipv4-pools --region us-east-1
```

要从 IPv4 地址池中创建弹性 IP 地址，请使用 allocate-address 命令。您可以使用 `--public-ipv4-pool` 选项指定 `describe-byoip-cidrs` 返回的地址池的 ID。或者，您可以使用 `--address` 选项从您预置的地址范围中指定一个地址。

IPv6 地址范围

要查看有关在您的账户中预置的 IPv6 地址池的信息，请使用以下 describe-ipv6-pools 命令。

```
aws ec2 describe-ipv6-pools --region us-east-1
```

要创建 VPC 并从 IPv6 地址池中指定 IPv6 CIDR，请使用以下 create-vpc 命令。要让 Amazon 从 IPv6 地址池中选择 IPv6 CIDR，请省略 `--ipv6-cidr-block` 选项。

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --ipv6-cidr-block ipv6-cidr --ipv6-
pool pool-id --region us-east-1
```

要将 IPv6 地址池中的 IPv6 CIDR 块与 VPC 关联，请使用以下 associate-vpc-cidr-block 命令。要让 Amazon 从 IPv6 地址池中选择 IPv6 CIDR，请省略 `--ipv6-cidr-block` 选项。

```
aws ec2 associate-vpc-cidr-block --vpc-id vpc-123456789abc123ab --ipv6-cidr-block ipv6-
cidr --ipv6-pool pool-id --region us-east-1
```

要查看 VPC 和关联的 IPv6 地址池信息，请使用 describe-vpcs 命令。要查看有关特定 IPv6 地址池中的关联 IPv6 CIDR 块的信息，请使用以下 get-associated-ipv6-pool-cidrs 命令。

```
aws ec2 get-associated-ipv6-pool-cidrs --pool-id pool-id --region us-east-1
```

如果将 IPv6 CIDR 块与 VPC 取消关联，则会将其释放回 IPv6 地址池中。

弹性 IP 地址

弹性 IP 地址 是专为动态云计算设计的静态 IPv4 地址。弹性 IP 地址会分配给您的 Amazon 账户，并且在您释放它之前一直属于您。使用弹性 IP 地址，您可以快速将地址重新映射到您的账户中的另一个实例，从而屏蔽实例故障。或者，您可以在域的 DNS 记录中指定弹性 IP 地址，以使域指向您的实例。有关更多信息，请参阅您的域注册商的文档。

弹性 IP 地址是公有 IPv4 地址，可通过 Internet 访问。如果需要连接到没有公有 IPv4 地址的实例，则可以将弹性 IP 地址与实例关联以实现与互联网的通信。

内容

- 弹性 IP 地址定价
- 弹性 IP 地址基本信息
- 弹性 IP 地址配额
- 将弹性 IP 地址与实例关联
- 在 Amazon Web Services 账户之间转移弹性 IP 地址
- 释放弹性 IP 地址
- 在 Amazon EC2 上为电子邮件创建反向 DNS 记录

弹性 IP 地址定价

所有正在使用（已分配给 EC2 实例等资源）和闲置（已在您的账户中创建但未分配）的弹性 IP 地址都会产生费用。

Amazon 将对所有公有 IPv4 地址收费，包括与运行的实例相关联的公有 IPv4 地址和弹性 IP 地址。有关更多信息，请参阅 Amazon VPC 定价页面中的公有 IPv4 地址定价选项卡。

弹性 IP 地址基本信息

下面是弹性 IP 地址的基本特征：

- 弹性 IP 地址是静态地址，不会随着时间的推移而改变。
- 一个弹性 IP 地址只能在一个特定区域中使用，不能移动到其他区域。
- 弹性 IP 地址来自 Amazon 的 IPv4 地址池，或来自已引入到您的 Amazon Web Services 账户的自定义 IPv4 地址池。我们不支持对 IPv6 使用弹性 IP 地址。
- 要使用弹性 IP 地址，您应首先向您的账户分配这样一个地址，然后将其与您的实例或网络接口关联。
- 当您将弹性 IP 地址与实例关联时，该地址也会与实例的主网络接口相关联。当您将弹性 IP 地址与连接到实例的网络接口关联时，它也与该实例关联。
- 当您将弹性 IP 地址与实例或其主网络接口关联时，如果该实例已具有与之关联的公有 IPv4 地址，则该公有 IPv4 地址将被释放回 Amazon 的公有 IPv4 地址池，并且弹性 IP 地址将与该实例关联。您不能重用以前与实例关联的公有 IPv4 地址，也不能将该公有 IPv4 地址转换为弹性 IP 地址。有关更多信息，请参阅 公有 IPv4 地址。
- 您可以取消弹性 IP 地址与资源的关联，然后重新将此地址与其他资源关联。为避免意外行为，请确保在进行更改之前关闭到现有关联中指定资源的所有活动连接。将弹性 IP 地址关联到其他资源后，可以重新打开与新关联的资源的连接。
- 取消关联的弹性 IP 地址保持分配到您的账户，直至您明确释放它。您需要为账户中的所有弹性 IP 地址付费，无论这些地址是否与实例关联。有关更多信息，请参阅 Amazon VPC 定价页面中的公有 IPv4 地址定价选项卡。
- 在将弹性 IP 地址与之前具有公有 IPv4 地址的实例关联时，该实例的公有 DNS 主机名将发生更改以匹配弹性 IP 地址。
- 我们会将公有 DNS 主机名解析为实例所在网络外部的该实例的公有 IPv4 地址或弹性 IP 地址，以及实例所在网络内部的该实例的私有 IPv4 地址。
- 从已引入到您 Amazon 账户的 IP 地址池分配弹性 IP 地址时，该地址不会计入弹性 IP 地址限制。有关更多信息，请参阅弹性 IP 地址配额。
- 分配弹性 IP 地址时，您可以将弹性 IP 地址与网络边界组关联。这是我们公告 CIDR 块的位置。设置网络边界组会将 CIDR 块限制到此组。如果您未指定网络边界组，我们将设置包含区域中所有可用区的边界组（例如，`us-west-2`）。
- 弹性 IP 地址仅用于特定网络边界组。

弹性 IP 地址配额

默认情况下，所有 Amazon Web Services 账户在每个区域的限额为五（5）个弹性 IP 地址，因为公有（IPv4）互联网地址是稀缺的公有资源。我们强烈建议您使用弹性 IP 地址，主要是为了在实例出现故障的情况下能够将该地址映射到另一实例，同时能够将 DNS 主机名用于所有其他节点间通信。

如果您认为您的架构需要额外的弹性 IP 地址，则可直接从 Service Quotas 控制台请求提高配额。要请求增加限额，请选择请求增加账户限额。有关更多信息，请参阅 Amazon EC2 Service Quotas。

将弹性 IP 地址与实例关联

分配弹性 IP 地址后，您可以将其与 EC2 实例、NAT 网关或网络负载均衡器等 Amazon 资源关联。要在稍后将弹性 IP 地址与其他 Amazon 资源关联，您可以将其与当前资源取消关联，然后将其与新资源关联。

任务

- 分配弹性 IP 地址
- 关联弹性 IP 地址
- 解除弹性 IP 地址的关联

分配弹性 IP 地址

您可以分配弹性 IP 地址以在区域中使用。所有正在使用（已与 EC2 实例等资源关联）和闲置（已在您的账户中创建但未关联）的弹性 IP 地址都会产生费用。

Console

分配弹性 IP 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Network & Security（网络与安全）下的Elastic IPs（弹性 IP）。
3. 选择 Allocate Elastic IP address (分配弹性 IP 地址)。
4. （可选）分配弹性 IP 地址（EIP）时，您可以选择要在其中分配 EIP 的网络边界组。网络边界组是一组可用区（AZ）、Local Zones 或 Wavelength Zones（Amazon 可从中公告公有 IP 地址）。Local Zones 和 Wavelength Zones 的网络边界组可能与区域中的可用区不同，以确保 Amazon 网络与访问这些区域中资源的客户之间保持最低延迟或最短物理距离。

> **Important**
>
> 您必须在将与该 EIP 关联的 Amazon 资源所在的网络边界组中分配一个 EIP。一个网络边界组中的 EIP 只能在该网络边界组中的区域内公告，而不能在由其他网络边界组代表的任何其他区域内公告。

如果您启用 Local Zones 或 Wavelength Zones（有关更多信息，请参阅启用 Local Zone 或启用 Wavelength Zones），则可以为可用区、Local Zones 或 Wavelength Zones 选择网络边界组。请谨慎选择网络边界组，因为 EIP 及其关联的 Amazon 资源必须位于同一个网络边界组中。您可以使用 EC2 控制台查看可用区、本地区或 Wavelength 区所在的网络边界组。通常，一个区域中的所有可用区都属于同一个网络边界组，而 Local Zones 或 Wavelength Zones 则属于各自独立的网络边界组。

如果您未启用 Local Zones 或 Wavelength Zones，则在分配 EIP 时，系统会为您预定义代表该区域所有可用区（例如 `us-west-2`）的网络边界组，并且您无法对其进行更改。这意味着您分配给该网络边界组的 EIP 将在您所属区域的所有可用区中公告。

5. 对于Public IPv4 address pool（公有 IPv4 地址池），选择以下选项之一：

   - Amazon 的 IPv4 地址池—如果要从 Amazon 的 IPv4 地址池中分配 IPv4 地址。
   - 引入 Amazon 账户的公有 IPv4 地址 – 如果您想从已引入到 Amazon 账户的 IP 地址池中分配一个不连续（非顺序）的 IPv4 地址。如果您没有任何 IP 地址池，则此选项将被禁用。有关将您自己的 IP 地址范围引入到您 Amazon 账户的更多信息，请参阅 自带 IP 地址（BYOIP）到 Amazon EC2。
   - Customer owned pool of IPv4 addresses (客户拥有的 IPv4 地址池) – 如果要从在本地网络创建的池中分配 IPv4 地址以供 Amazon Outpost 使用。如果您没有 Amazon Outpost，则此选项将被禁用。
   - 使用 IPAM IPv4 池进行分配：如果您想从 IPAM 池中连续的公有 IPv4 块中分配连续的弹性 IP 地址。分配连续的弹性 IP 地址可以显著减少安全访问控制列表的管理开销，简化在 Amazon 上扩展的企业的 IP 地址分配和跟踪工作。有关更多信息，请参阅《Amazon VPC IPAM User Guide》中的 Allocate sequential Elastic IP addresses from an IPAM pool。

6. （可选）要添加标签，请选择添加新标签，然后输入标签键和标签值。

Amazon CLI

分配弹性 IP 地址

使用 allocate-address Amazon CLI 命令。

在以下示例中，Amazon EC2 从 Amazon 的地址池中选择地址。

```
aws ec2 allocate-address
```

在以下示例中，Amazon EC2 从您通过 BYOIP 引入 Amazon 的指定地址池中选择地址。

```
aws ec2 allocate-address \
    --public-ipv4-pool ipv4pool-ec2-012345abcdef67890
```

以下示例指定来自指定 IPv4 IPAM 地址池的地址。

```
aws ec2 allocate-address \
    --ipam-pool-id ipam-pool-1234567890abcdef0 \
    --address 192.0.2.0
```

PowerShell

分配弹性 IP 地址

使用 New-EC2Address cmdlet。

在以下示例中，Amazon EC2 从 Amazon 的地址池中选择地址。

```
New-EC2Address
```

在以下示例中，Amazon EC2 从您通过 BYOIP 引入 Amazon 的指定地址池中选择地址。

```
New-EC2Address `
    -PublicIpv4Pool ipv4pool-ec2-012345abcdef67890
```

以下示例指定来自指定 IPv4 IPAM 地址池的地址。

```
New-EC2Address `
    -IpamPoolId ipam-pool-1234567890abcdef0 `
    -Address 192.0.2.0
```

关联弹性 IP 地址

如果要将弹性 IP 地址与您的实例关联以启用与 Internet 的通信，您还必须确保您的实例在公有子网中。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用互联网网关启用 VPC 互联网访问。

Console

要关联弹性 IP 地址和实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择弹性 IP。
3. 选择要关联的弹性 IP 地址，然后依次选择 Actions (操作)、Associate Elastic IP address (关联弹性 IP 地址)。
4. 对于 Resource type (资源类型)，选择 Instance (实例)。
5. 例如，选择要将弹性 IP 地址关联到的实例。您还可以输入文本以搜索特定实例。
6. （可选）对于 Private IP address (私有 IP 地址)，请指定要将弹性 IP 地址关联到的私有 IP 地址。
7. 选择 Associate。

将弹性 IP 地址与网络接口相关联

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择弹性 IP。
3. 选择要关联的弹性 IP 地址，然后依次选择 Actions (操作)、Associate Elastic IP address (关联弹性 IP 地址)。
4. 对于 Resource type (资源类型)，选择 Network interface (网络接口)。
5. 对于 Network interface (网络接口)，请选择要将弹性 IP 地址关联到的网络接口。您还可以输入文本来搜索特定网络接口。
6. （可选）对于 Private IP address (私有 IP 地址)，请指定要将弹性 IP 地址关联到的私有 IP 地址。
7. 选择 Associate。

Amazon CLI

关联弹性 IP 地址

使用 associate-address Amazon CLI 命令。

```
aws ec2 associate-address \
    --instance-id i-0b263919b6498b123 \
```

```
    --allocation-id eipalloc-64d5890a
```

PowerShell

关联弹性 IP 地址

使用 Register-EC2Address cmdlet。

```
Register-EC2Address `
    -InstanceId i-0b263919b6498b123 `
    -AllocationId eipalloc-64d5890a
```

解除弹性 IP 地址的关联

您可以随时取消弹性 IP 地址与实例或网络接口的关联。取消关联弹性 IP 地址后，您可以将其与其他资源关联。

Console

取消关联和重新关联弹性 IP 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择弹性 IP。
3. 选择要取消关联的弹性 IP 地址，然后依次选择 Actions (操作)、Disassociate Elastic IP address (取消关联弹性 IP 地址)。
4. 选择取消关联。

Amazon CLI

撤销弹性 IP 地址的关联

使用 disassociate-address Amazon CLI 命令。

```
aws ec2 disassociate-address --association-id eipassoc-12345678
```

PowerShell

撤销弹性 IP 地址的关联

使用 Unregister-EC2Address cmdlet。

```
Unregister-EC2Address -AssociationId eipassoc-12345678
```

在 Amazon Web Services 账户之间转移弹性 IP 地址

您可以将弹性 IP 地址从一个 Amazon Web Services 账户转移到另一个。这可能在以下情况下有帮助：

- 灾难恢复：在紧急事件期间为面向公众的互联网工作负载快速重新映射 IP 地址。
- 组织调整：将工作负载从一个 Amazon Web Services 账户快速移动到另一个。地址转移无需等待安全组和网络 ACL 允许即可使用新弹性 IP 地址。
- 集中式安全管理 – 使用集中式 Amazon 安全账户跟踪和转移已通过安全合规性审查的弹性 IP 地址。

定价

转移弹性 IP 地址不收取任何费用。

任务

- 启用弹性 IP 地址转移
- 接受转移的弹性 IP 地址
- 禁用弹性 IP 地址转移

启用弹性 IP 地址转移

本节介绍如何接受转移的弹性 IP 地址。请注意以下与启用弹性 IP 地址转移相关的限制：

- 您可以将弹性 IP 地址从任何 Amazon Web Services 账户（源账户）转移到同一 Amazon Web Services 区域中的任何其他 Amazon Web Services 账户（转入账户）。无法将弹性 IP 地址转移到其他区域。
- 在转移弹性 IP 地址时，Amazon Web Services 账户 账户之间有两步握手。当源账户开始转移时，转移账户有七天的时间来接受这个弹性 IP 地址转移。在那七天之内，从源账户可以看得见有待转移的那个弹性 IP 地址，比如：在 Amazon 控制台中或使用 describe-address-transfers 命令即可看到。七天之后，转移将过期，而且弹性 IP 地址的所有权将被返回给其原始账户。
- 在接受转移后的 14 天内，源账户可以查看已接受的转移，例如：在 Amazon 控制台中或使用 describe-address-transfers 命令即可查看。

- Amazon 不会向转移账户通知待处理的弹性 IP 地址转移请求。源账户的所有者必须通知转移账户的所有者：他们必须接受弹性 IP 地址转移请求。
- 转移完成后，与正在转移的弹性 IP 地址关联的所有标签都将被重置。
- 您无法将从自带的公有 IPv4 地址池 [通常称为自带 IP（BYOIP）地址池] 分配的弹性 IP 地址转移到您的 Amazon Web Services 账户。
- 从 Amazon 提供的连续公有 IPv4 Amazon VPC IP 地址管理器（IPAM）池中分配的弹性 IP 地址不支持转移。不过，通过将 IPAM 与 Amazon Organizations 集成并使用 Amazon RAM，IPAM 允许跨 Amazon 账户共享 IPAM 池。有关更多信息，请参阅《Amazon VPC IPAM User Guide》中的 Allocate sequential Elastic IP addresses from an IPAM pool。
- 如果您尝试转移与反向 DNS 记录相关联的弹性 IP 地址，则可以开始转移过程，但是在删除关联的 DNS 记录之前，转移账户将无法接受转移。
- 如果已启用并配置了 Amazon Outposts，则您可能已从客户拥有的 IP 地址池（CoIP）中分配了弹性 IP 地址。您无法转移从 CoIP 分配的弹性 IP 地址。但是，您可以使用 Amazon RAM 与其他账户共享 CoIP。有关更多信息，请参阅《Amazon Outposts 用户指南》中的客户拥有的 IP 地址。
- 您可以使用 Amazon VPC IPAM 跟踪向 Amazon Organizations 组织中的账户转移弹性 IP 地址的情况。有关更多信息，请参阅查看 IP 地址历史记录。但是，如果将弹性 IP 地址转移到组织外部的 Amazon Web Services 账户，则弹性 IP 地址的 IPAM 审核历史记录将丢失。

这些步骤必须由源账户完成。

Console

启用弹性 IP 地址转移

1. 确保您使用的是源 Amazon 账户。
2. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
3. 在导航窗格中，选择弹性 IP。
4. 选择一个或多个要启用转移的弹性 IP 地址，然后选择 Actions（操作）、Enable transfer（启用转移）。
5. 如果要转移多个弹性 IP 地址，则会看到 Transfer type（转移类型）选项。请选择以下选项之一：
   - 如果要将弹性 IP 地址转移到单个 Amazon 账户，请选择 Single account（单个账户）。
   - 如果要将弹性 IP 地址转移到多个 Amazon 账户，请选择 Multiple accounts（多个账户）。

6. 在 Transfer account ID（转移账户 ID）下，输入要向其转移弹性 IP 地址的 Amazon 账户的 ID。
7. 在文本框中输入 **enable** 以确认转移。
8. 选择提交。
9. 若要接受转移，请参阅 接受转移的弹性 IP 地址。若要禁用转移，请参阅 禁用弹性 IP 地址转移。

Amazon CLI

启用弹性 IP 地址转移

使用 enable-address-transfer 命令。

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

PowerShell

启用弹性 IP 地址转移

使用 Enable-EC2AddressTransfer cmdlet。

```
Enable-EC2AddressTransfer `
    -AllocationId eipalloc-09ad461b0d03f6aaf `
    -TransferAccountId 123456789012
```

接受转移的弹性 IP 地址

本节介绍如何接受转移的弹性 IP 地址。

在转移弹性 IP 地址时，Amazon Web Services 账户 账户之间有两步握手。当源账户开始转移时，转移账户有七天的时间来接受这个弹性 IP 地址转移。在那七天之内，从源账户可以看得见有待转移的那个弹性 IP 地址，比如：在 Amazon 控制台中或使用 describe-address-transfers 命令即可看到。七天之后，转移将过期，而且弹性 IP 地址的所有权将被返回给其原始账户。

接受转移时，请注意可能发生的以下异常情况以及如何解决这些异常：

- AddressLimitExceeded：如果您的转移账户相应的弹性 IP 地址配额已超出，可以通过源账户来启动弹性 IP 地址转移，但此举将会导致系统出现异常，因为只要是转移账户在尝试接受这样的转移，此异常情况就会发生。默认情况下，每个 Amazon 账户的限额为每个区域仅能使用最多 5 个弹性 IP 地址。欲提高此限制，请见此说明:弹性 IP 地址配额。
- InvalidTransfer.AddressCustomPtrSet：如果您或组织中的某人已将您正在尝试转移的弹性 IP 地址配置为使用反向 DNS 查找，则源账户可以启用弹性 IP 地址转移，但是当转移账户尝试接受转移时，会发生此异常。若要解决此问题，源账户必须删除弹性 IP 地址的 DNS 记录。有关更多信息，请参阅 在 Amazon EC2 上为电子邮件创建反向 DNS 记录。
- InvalidTransfer.AddressAssociated：如果弹性 IP 地址与 ENI 或 EC2 实例相关联，则源帐户可以启用弹性 IP 地址转移，但是当转移帐户尝试接受转移时，会发生此异常。若要解决此问题，源帐户必须解除弹性 IP 地址的关联。有关更多信息，请参阅 解除弹性 IP 地址的关联。

对于任何其他异常，请联系 Amazon Web Services 支持。

这些步骤必须由转移账户完成。

Console

接受弹性 IP 地址转移

1. 确保您使用的是转移账户。
2. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
3. 在导航窗格中，选择弹性 IP。
4. 选择 Actions（操作）、Accept transfer（接受转移）。
5. 当您接受转移时，与要转移的弹性 IP 地址关联的标签不会随弹性 IP 地址一起转移。如果要为所接受的弹性 IP 地址定义 Name（名称）标签，请选择 Create a tag with a key of 'Name' and a value that you specify（创建具有“Name”键以及您指定的值的标签）。
6. 输入要转移的弹性 IP 地址。
7. 如果您接受多个转移的弹性 IP 地址，请选择 Add address（添加地址）以输入附加弹性 IP 地址。
8. 选择提交。

Amazon CLI

接受弹性 IP 地址转移

使用 accept-address-transfer 命令。

```
aws ec2 accept-address-transfer --address 100.21.184.216
```

PowerShell

接受弹性 IP 地址转移

使用 Approve-EC2AddressTransfer cmdlet。

```
Approve-EC2AddressTransfer -Address 100.21.184.216
```

禁用弹性 IP 地址转移

本节介绍如何在启用转移后禁用弹性 IP 转移。

这些步骤必须由启用转移的源账户完成。

Console

禁用弹性 IP 地址转移

1. 确保您使用的是源 Amazon Web Services 账户。
2. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
3. 在导航窗格中，选择弹性 IP。
4. 在弹性 IP 的资源列表中，确保启用了显示 Transfer status（转移状态）列的属性。
5. 选择一个或多个 Transfer status（转移状态）为 Pending（待处理）的弹性 IP 地址，然后选择 Actions（操作）、Disable transfer（禁用转移）。
6. 在文本框中输入 **disable** 以确认。
7. 选择提交。

Amazon CLI

禁用弹性 IP 地址转移

使用 disable-address-transfer 命令。

```
aws ec2 disable-address-transfer --allocation-id eipalloc-09ad461b0d03f6aaf
```

PowerShell

禁用弹性 IP 地址转移

使用 Disable-EC2AddressTransfer cmdlet。

```
Disable-EC2AddressTransfer -AllocationId eipalloc-09ad461b0d03f6aaf
```

释放弹性 IP 地址

如果您不再需要某个弹性 IP 地址，建议您释放该地址。要发布的弹性 IP 地址当前不得与 Amazon 资源关联。

Console

释放弹性 IP 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择弹性 IP。
3. 选择要释放的弹性 IP 地址，然后依次选择 Actions (操作)、Release Elastic IP addresses (释放弹性 IP 地址)。
4. 选择 Release (释放)。

Amazon CLI

释放弹性 IP 地址

使用 release-address Amazon CLI 命令。

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

PowerShell

释放弹性 IP 地址

使用 Remove-EC2Address cmdlet。

```
Remove-EC2Address -AllocationId eipalloc-64d5890a
```

释放弹性 IP 地址后，您可能能够恢复它。以下规则适用：

- 如果弹性 IP 地址已分配至其他 Amazon 账户，或者该地址将导致您超出弹性 IP 地址限制，则您无法恢复该地址。
- 您不能恢复与弹性 IP 地址关联的标签。

Amazon CLI

恢复弹性 IP 地址

使用 allocate-address 命令。

```
aws ec2 allocate-address \
    --domain vpc \
    --address 203.0.113.3
```

PowerShell

恢复弹性 IP 地址

使用 New-EC2Address cmdlet。

```
New-EC2Address `
    -Address 203.0.113.3 `
    -Domain vpc `
    -Region us-east-1
```

在 Amazon EC2 上为电子邮件创建反向 DNS 记录

如果您打算从 EC2 实例向第三方发送电子邮件，我们建议您预置一个或多个弹性 IP 地址，并向您用于发送电子邮件的弹性 IP 地址分配静态反向 DNS 记录。这样有助于避免电子邮件被一些反垃圾电子邮件组织标记为垃圾电子邮件。Amazon 与 ISP 以及 Internet 反垃圾电子邮件组织合作，减少从这些地址发送的电子邮件被标记为垃圾电子邮件的几率。

注意事项

- 在创建反向 DNS 记录之前，您必须设置指向您的弹性 IP 地址的对应的正向 DNS 记录（记录类型 A）。

- 如果反向 DNS 记录与弹性 IP 地址关联，则该弹性 IP 地址将锁定到您的账户中且无法从您的账户中释放，直至删除记录。
- 如果您联系了 Amazon Web Services 支持 为弹性 IP 地址设置反向 DNS，您可以删除该反向 DNS，但不能释放弹性 IP 地址，因为它已被 Amazon Web Services 支持 锁定。要解锁弹性 IP 地址，请联系 Amazon Web Services 支持。解锁弹性 IP 地址后，您就可以释放该地址。
- [Amazon GovCloud (US) Region] 您不能创建反向 DNS 记录。Amazon 必须为您分配静态反向 DNS 记录。建立支持案例，取消反向 DNS 和电子邮件发送限制。您必须提供弹性 IP 地址和反向 DNS 记录。

创建反向 DNS 记录

您可以按如下方式为弹性 IP 地址创建反向 DNS 记录。

Console

创建反向 DNS 记录

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择弹性 IP。
3. 选择弹性 IP 地址，然后依次选择 Actions（操作）和 Update reverse DNS（更新反向 DNS）。
4. 对于 Reverse DNS domain name（反向 DNS 域名），输入域名。
5. 输入 **update** 以确认。
6. 选择更新。

Amazon CLI

创建反向 DNS 记录

使用 modify-address-attribute 命令。

```
aws ec2 modify-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --domain-name example.com
```

下面是示例输出。

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net.",
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
            }
        }
    ]
}
```

PowerShell

创建反向 DNS 记录

使用 Edit-EC2AddressAttribute cmdlet。

```
Edit-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -DomainName 'example.com' |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

下面是示例输出。

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.net.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.com.
```

删除反向 DNS 记录

您可以按以下方式从弹性 IP 地址中删除反向 DNS 记录。

如果收到以下错误，您可以向 Amazon Web Services 支持 团队提交删除电子邮件发送限制的请求来获得帮助。

```
The address cannot be released because it is locked to your account.
```

Console

移除反向 DNS 记录

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择弹性 IP。
3. 选择弹性 IP 地址，然后依次选择 Actions（操作）和 Update reverse DNS（更新反向 DNS）。
4. 对于 Reverse DNS domain name（反向 DNS 域名），清除域名。
5. 输入 **update** 以确认。
6. 选择更新。

Amazon CLI

移除反向 DNS 记录

使用 reset-address-attribute 命令。

```
aws ec2 reset-address-attribute \
    --allocation-id eipalloc-abcdef01234567890 \
    --attribute domain-name
```

下面是示例输出。

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com.",
            "PtrRecordUpdate": {
                "Value": "example.net.",
                "Status": "PENDING"
```

```
            }
        }
    ]
}
```

PowerShell

移除反向 DNS 记录

使用 Reset-EC2AddressAttribute cmdlet。

```
Reset-EC2AddressAttribute `
    -AllocationId 'eipalloc-abcdef01234567890' `
    -Attribute domain-name |
Format-List `
    AllocationId, PtrRecord, PublicIp,
    @{Name='PtrRecordUpdate';Expression={$_.PtrRecordUpdate | Format-List | Out-
String}}
```

下面是示例输出。

```
AllocationId    : eipalloc-abcdef01234567890
PtrRecord       : example.com.
PublicIp        : 192.0.2.0
PtrRecordUpdate :
                  Reason :
                  Status : PENDING
                  Value  : example.net.
```

弹性网络接口

弹性网络接口 是 VPC 中表示虚拟网卡的逻辑网络组件。您可以创建并配置网络接口，并将其连接到同一可用区中启动的实例。将网络接口附加到一个实例或者从一个实例分离并重新附加到另一实例时，网络接口的属性不会变化。当您将网络接口从一个实例移动到另一个实例时，网络流量也会从原始实例重定向到新实例。

请注意，该 Amazon 资源在 Amazon Web Services 管理控制台 和 Amazon EC2 API 中称为网络接口。因此，我们在本文档中使用“网络接口”，而不是“弹性网络接口”。本文档中的“网络接口”术语始终表示“弹性网络接口”。

网络接口属性

网络接口可以包含以下属性：

- 您子网的 IPv4 地址范围内的一个主要私有 IPv4 地址
- 您子网的 IPv6 地址范围内的一个主 IPv6 地址
- 子网 IPv4 地址范围内的辅助私有 IPv4 地址
- 每个私有 IPv4 地址一个弹性 IP 地址（IPv4）
- 一个公有 IPv4 地址
- 辅助 IPv6 地址
- 安全组
- 一个 MAC 地址
- 一个源/目标检查标记
- 一个描述

监控流量

您可以在网络接口上启用 VPC 流日志以捕获有关出入该网络接口的流量的信息。创建流日志后，您可以在 Amazon CloudWatch Logs 中查看和检索其数据。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 流日志。

内容

- 网络接口概念
- 网卡
- 每个网络接口的最大 IP 地址数
- 为 Amazon EC2 实例创建网络接口
- EC2 实例的网络接口连接
- 管理网络接口的 IP 地址
- 修改网络接口属性
- 您的 Amazon EC2 实例的多个网络接口
- 请求者托管的网络接口
- Amazon EC2 网络接口的前缀委派

- 删除网络接口

网络接口概念

以下是开始使用网络接口时需要理解的重要概念。

主网络接口

每个实例都有一个默认网络接口，称为主网络接口。您无法将主网络接口与实例分离。

辅助网络接口

您可以创建辅助网络接口并将其附加到您的实例。最大网络接口数量因实例类型而不同。有关更多信息，请参阅 每个网络接口的最大 IP 地址数。

网络接口的 IPv4 地址

在仅 IPv4 或双堆栈子网中启动 EC2 实例时，该实例将在子网的 IPv4 地址范围内收到一个主要私有 IP 地址。您还可以指定其他私有 IPv4 地址，即辅助私有 IPv4 地址。与主要私有 IP 地址不同的是，辅助私有 IP 地址可以从一个实例重新分配到另一个实例。

网络接口的公有 IPv4 地址

所有子网都有一个可以修改的属性，该属性可以确定在子网中创建的网络接口（以及因此在该子网中启动的实例）是否会分配一个公有 IPv4 地址。有关更多信息，请参阅《Amazon VPC 用户指南》中的子网设置。在启动实例时，将为主网络接口分配 IP 地址。如果启动实例时指定现有的网络接口作为主网络接口，则公有 IPv4 地址由该网络接口确定。

当您创建一个网络接口时，它会继承子网的公有 IPv4 寻址属性。如果您日后修改了子网的公有 IPv4 寻址属性，网络接口仍会继续使用在其创建时生效的设置。

当该实例停止、休眠或终止后，我们会释放公有 IP 地址。当您启动已停止或休眠状态的实例时，我们会分配新的公有 IP 地址，除非其具有辅助网络接口或与弹性 IP 地址关联的辅助私有 IPv4 地址。

网络接口的 IPv6 地址

如果您将 IPv6 CIDR 块与您的 VPC 和子网关联，则可以将子网范围内的 IPv6 地址分配给一个网络接口。每个 IPv6 地址可以分配给一个网络接口。

所有子网都有一个可以修改的属性，该属性可以确定在子网中创建的网络接口 (以及在该子网中启动的实例) 是否会自动分配到一个处于子网范围内的 IPv6 地址。在启动实例时，将为主网络接口分配 IPv6 地址。

网络接口的弹性 IP 地址

您可以将弹性 IP 地址与网络接口的其中一个私有 IPv4 地址关联。您可以为每个私有 IPv4 地址关联一个弹性 IP 地址。如果解除弹性 IP 地址与网络接口的关联，则可以释放该弹性 IP 地址或将其与其他实例关联。

终止操作

您可以设置附加到实例的网络接口的终止操作。您可以指定在终止网络接口附加到的实例时是否自动删除该接口。

源/目标检查

您可以启用或禁用源/目标检查，以确保实例是其接收的任何流量的源或目标。默认情况下会启用源/目标检查。如果实例运行网络地址转换、路由或防火墙等服务，您必须禁用源/目标检查。

请求者托管的网络接口

这些网络接口由 Amazon Web Services 服务创建和管理，以使您能够使用一些资源和服务。您自己无法管理这些网络接口。有关更多信息，请参阅 请求者托管的网络接口。

前缀委派

前缀是预留的私有 IPv4 或 IPv6 CIDR 范围，用于自动或手动分配给与实例关联的网络接口。通过使用委派前缀，您可以通过将一系列 IP 地址分配为单个前缀来更快启动服务。

托管网络接口

托管网络接口是指由服务提供商（例如 Amazon EKS 自动模式）管理的实例。您不能直接修改托管网络接口的设置。托管网络接口由托管字段中的 true 进行标识。有关更多信息，请参阅 Amazon EC2 托管式实例。

网卡

大多数实例类型都支持一个网卡。支持多个网卡的实例类型可提供更高的网络性能，包括 100 Gbps 以上的带宽能力和更高的数据包速率性能。在将网络接口附加到支持多个网卡的实例时，可以为该网络接口选择网卡。必须将主网络接口分配给网卡索引 0。

EFA 和仅限 EFA 的网络接口算作网络接口。每个网卡只能分配一个 EFA 或仅限 EFA 的网络接口。主网络接口不能是仅限 EFA 的网络接口。

以下实例类型支持多个网卡。有关实例类型支持的网络接口数量的信息，请参阅每个网络接口的最大 IP 地址数。

实例类型	网卡数
c6in.32xlarge	2
c6in.metal	2
c8gb.48xlarge	2
c8gb.metal-48xl	2
c8gn.48xlarge	2
c8gn.metal-48xl	2
c8in.96xlarge	2
c8in.metal-96xl	2
c8ib.96xlarge	2
c8ib.metal-96xl	2
dl1.24xlarge	4
g6e.24xlarge	2
g6e.48xlarge	4
g7e.24xlarge	2
g7e.48xlarge	4
hpc6id.32xlarge	2
hpc7a.12xlarge	2
hpc7a.24xlarge	2
hpc7a.48xlarge	2
hpc7a.96xlarge	2

实例类型	网卡数
hpc8a.96xlarge	2
m6idn.32xlarge	2
m6idn.metal	2
m6in.32xlarge	2
m6in.metal	2
m8gb.48xlarge	2
m8gb.metal-48xl	2
m8gn.48xlarge	2
m8gn.metal-48xl	2
m8in.96xlarge	2
m8in.metal-96xl	2
m8idn.96xlarge	2
m8idn.metal-96xl	2
m8ib.96xlarge	2
m8ib.metal-96xl	2
m8idb.96xlarge	2
m8idb.metal-96xl	2
p4d.24xlarge	4
p4de.24xlarge	4
p5.48xlarge	32

实例类型	网卡数
p5e.48xlarge	32
p5en.48xlarge	16
p6-b200.48xlarge	8
p6-b300.48xlarge	17
p6e-gb200.36xlarge	17
r8gb.48xlarge	2
r8gb.metal-48xl	2
r8gn.48xlarge	2
r8gn.metal-48xl	2
r8in.96xlarge	2
r8in.metal-96xl	2
r8idn.96xlarge	2
r8idn.metal-96xl	2
r8ib.96xlarge	2
r8ib.metal-96xl	2
r8idb.96xlarge	2
r8idb.metal-96xl	2
r6idn.32xlarge	2
r6idn.metal	2
r6in.32xlarge	2

实例类型	网卡数
r6in.metal	2
trn1.32xlarge	8
trn1n.32xlarge	16
trn2.48xlarge	16
trn2u.48xlarge	16
u7in-16tb.224xlarge	2
u7in-24tb.224xlarge	2
u7in-32tb.224xlarge	2
u7inh-32tb.480xlarge	2

每个网络接口的最大 IP 地址数

每个实例类型支持的每个网络接口的最大数量、每个网络接口支持的私有 IPv4 地址的最大数量，以及每个网络接口支持的 IPv6 地址的最大数量。每个网络接口的 IPv6 地址与私有 IPv4 地址有不同的限制并且分别列出。请注意，除了 C1、M1、M2、M3 和 T1 实例类型，其余实例类型都支持 IPv6 寻址。

可用的网络接口

《Amazon EC2 实例类型指南》提供有关每种实例类型可以使用的网络接口的信息。有关更多信息，请参阅下列内容：

- 网络规格 – 通用型
- 网络规格 – 计算优化型
- 网络规格 – 内存优化型
- 网络规格 – 存储优化型
- 网络规格 – 加速计算型
- 网络规格 – 高性能计算型
- 网络规格 – 上一代

Console

检索最大网络接口数

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instance Types (实例类型)。
3. 添加筛选条件来指定实例类型（实例类型=c5.12xlarge）或实例系列（实例系列=c5）。
4. （可选）单击首选项图标，然后打开最大网络接口数。此列会指示每种实例类型的最大网络接口数量。
5. （可选）选择实例类型。在联网选项卡上，找到最大网络接口数。

Amazon CLI

检索最大网络接口数

您可以使用 describe-instance-types 命令来显示有关实例类型的信息，例如其支持的网络接口和每个接口的 IP 地址。以下示例显示了所有 C8i 实例的此类信息。

```
{ echo -e "InstanceType\tMaximumNetworkInterfaces\tIpv4AddressesPerInterface"; \
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c8i.*" \
    --query 'InstanceTypes[*].[InstanceType, NetworkInfo.MaximumNetworkInterfaces,
 NetworkInfo.Ipv4AddressesPerInterface]' \
    --output text | sort -k2 -n; } | column -t
```

下面是示例输出。

```
InstanceType    MaximumNetworkInterfaces  Ipv4AddressesPerInterface
c8i.large       3                         20
c8i.2xlarge     4                         30
c8i.xlarge      4                         30
c8i.4xlarge     8                         50
c8i.8xlarge     10                        50
c8i.12xlarge    12                        50
c8i.16xlarge    16                        64
c8i.24xlarge    16                        64
c8i.32xlarge    24                        64
c8i.48xlarge    24                        64
c8i.96xlarge    24                        64
c8i.metal-48xl  24                        64
```

```
c8i.metal-96xl  24                          64
```

PowerShell

检索最大网络接口数

您可以使用 Get-EC2InstanceType PowerShell 命令来显示有关实例类型的信息，例如其支持的网络接口和每个接口的 IP 地址。以下示例显示了所有 C8i 实例的此类信息。

```
Get-EC2InstanceType -Filter @{Name="instance-type"; Values="c8i.*"} |
Select-Object `
    InstanceType,
    @{Name='MaximumNetworkInterfaces';
 Expression={$_.NetworkInfo.MaximumNetworkInterfaces}},
    @{Name='Ipv4AddressesPerInterface';
 Expression={$_.NetworkInfo.Ipv4AddressesPerInterface}} |
Sort-Object MaximumNetworkInterfaces |
Format-Table -AutoSize
```

下面是示例输出。

```
InstanceType   MaximumNetworkInterfaces Ipv4AddressesPerInterface
------------   ------------------------ -------------------------
c8i.large                             3                        20
c8i.xlarge                            4                        30
c8i.2xlarge                           4                        30
c8i.4xlarge                           8                        50
c8i.8xlarge                          10                        50
c8i.12xlarge                         12                        50
c8i.24xlarge                         16                        64
c8i.16xlarge                         16                        64
c8i.96xlarge                         24                        64
c8i.48xlarge                         24                        64
c8i.metal-96xl                       24                        64
c8i.32xlarge                         24                        64
c8i.metal-48xl                       24                        64
```

为 Amazon EC2 实例创建网络接口

您可以创建一个网络接口供您的 EC2 实例使用。创建网络接口时，需要指定创建该接口的子网。您不能在创建网络接口之后将其移动至另一子网。您必须将网络接口连接到同一可用区中的实例。您可以

从实例中分离辅助网络接口，然后将其附加到同一可用区中的其他实例。您无法将主网络接口与实例分离。有关更多信息，请参阅 the section called “网络接口连接”。

Console

创建网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选择创建网络接口。
4. （可选）对于描述，请输入一个描述性的名称。
5. 对于 Subnet (子网)，选择一个子网。后续步骤中可用的选项根据您选择的子网类型（仅 IPv4、仅 IPv6 或双堆栈（IPv4 和 IPv6））而有所变化。
6. 对于集成类型，请选择下列选项之一：
   - ENA：一种高性能网络接口，旨在满足高吞吐量和每秒数据包速率的 TCP/IP 协议处理需求，同时尽可能降低 CPU 使用率。这是默认值。有关更多信息，请参阅弹性网络适配器。
   - 带 ENA 的 EFA：一种同时支持 ENA 和 EFA 设备的网络接口，用于传统基于 TCP/IP 的传输以及基于 SRD 的传输。如果选择带 ENA 的 EFA，则挂载该接口的实例必须支持 EFA。有关 EFA 的更多信息，请参阅 Elastic Fabric Adapter。
   - 仅 EFA：一种高性能网络接口，旨在满足基于 SRD 的传输对高吞吐量、低延迟节点间通信的处理需求，同时避开操作系统堆栈。如果选择此选项，则挂载该接口的实例必须支持 EFA。仅 EFA 网络接口不支持 IP 地址。有关 EFA 的更多信息，请参阅 Elastic Fabric Adapter。
7. 对于私有 IPv4 地址，执行以下操作之一：
   - 选择自动分配允许 Amazon EC2 从子网中选择 IPv4 地址。
   - 选择自定义然后输入从子网中选择的 IPv4 地址。
8. （仅具有 IPv6 地址的子网）对于 IPv6 地址，请执行以下操作之一：
   - 如果您不想为网络接口分配 IPv6 地址，选择没有。
   - 选择自动分配允许 Amazon EC2 从子网中选择 IPv6 地址。
   - 选择自定义然后输入从子网中选择的 IPv6 地址。
9. （可选）如果您在双栈或仅使用 IPv6 的子网中创建网络接口，则可以选择分配主要 IPv6 IP。这将为网络接口分配一个主 IPv6 全局单播地址（GUA）。分配主要 IPv6 地址使您能够避免中断实例或 ENI 的流量。如果此 ENI 要连接至的实例依赖于其 IPv6 地址不变，则选择启用。Amazon 会自动将与您的实例连接的 ENI 相关联 IPv6 地址分配为主要 IPv6 地址。一旦将

IPv6 GUA 地址启用为主要 IPv6，您将无法将其禁用。当您将 IPv6 GUA 地址启用为主要 IPv6 时，第一个 IPv6 GUA 将成为主要 IPv6 地址，直到实例终止或网络接口断开为止。如果您有多个 IPv6 地址与连接到实例的 ENI 相关联，并且启用了主要 IPv6 地址，则与该 ENI 关联的第一个 IPv6 GUA 地址将成为主要 IPv6 地址。

10. （可选）要创建 Elastic Fabric Adapter，请选择 Elastic Fabric Adapter，启用。
11. （可选）在高级设置下，您可以选择设置 IP 前缀委托。有关更多信息，请参阅 前缀委派。
    - 自动分配：Amazon 从子网的 IPv4 或 IPv6 CIDR 块中选择前缀，并将其分配给该网络接口。
    - 手动分配：您可以从子网的 IPv4 或 IPv6 CIDR 块中指定前缀，Amazon 会在将该前缀分配给网络接口之前，验证其是否尚未分配给其他资源。
12. （可选）在高级设置下，对于空闲连接跟踪超时，请修改默认的空闲连接超时。有关更多信息，请参阅 空闲连接跟踪超时。
    - TCP 建立超时：处于已建立状态的空闲 TCP 连接的超时时间（以秒为单位）。
      - 最小值：60 秒
      - 最大值：432000 秒
      - 默认值：Nitro v6 实例类型（不包括 p6e-GB200）为 350 秒；所有其他实例类型（包括 p6e-GB200）为 432000 秒。
      - 建议值：小于 432000 秒
    - UDP 超时：空闲 UDP 流的超时时间（以秒为单位），这些流仅在单个方向或单个请求-响应事务上看到流量。
      - 最小值：30 秒
      - 最大值：60 秒
      - 原定设置：30 秒
    - UDP 流超时：空闲 UDP 流的超时（以秒为单位），这些流被归类为已看到多个请求-响应事务的流。
      - 最小值：60 秒
      - 最大值：180 秒
      - 原定设置：180 秒
13. 对于 Security groups，选择一个或多个安全组。
14. （可选）对于每个标签，请选择添加新标签，然后输入标签键和可选的标签值。
15. 选择创建网络接口。

Amazon CLI

示例 1：使用 Amazon EC2 选择的 IP 地址创建网络接口

使用以下 create-network-interface 命令。此示例将使用 Amazon EC2 选择的一个公有 IPv4 地址和一个 IPv6 地址创建一个网络接口。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --ipv6-address-count 1 \
    --groups sg-1234567890abcdef0
```

示例 2：使用特定的 IP 地址创建网络接口

使用以下 create-network-interface 命令。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my dual-stack network interface" \
    --private-ip-address 10.251.50.12 \
    --ipv6-addresses 2001:db8::1234:5678:1.2.3.4 \
    --groups sg-1234567890abcdef0
```

示例 3：使用一组辅助 IP 地址创建网络接口

使用以下 create-network-interface 命令。在此示例中，Amazon EC2 选择了主 IP 地址和若干辅助 IP 地址。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --secondary-private-ip-address-count 2 \
    --groups sg-1234567890abcdef0
```

示例 4：使用特定的辅助 IP 地址创建网络接口

使用以下 create-network-interface 命令。此示例会指定一个主 IP 地址和一个辅助 IP 地址。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --description "my network interface" \
    --private-ip-addresses PrivateIpAddress=10.0.1.30,Primary=true \
```

```
                              PrivateIpAddress=10.0.1.31,Primary=false
    --groups sg-1234567890abcdef0
```

PowerShell

示例 1：使用 Amazon EC2 选择的 IP 地址创建网络接口

使用 New-EC2NetworkInterface cmdlet。此示例将使用 Amazon EC2 选择的一个公有 IPv4 地址和一个 IPv6 地址创建一个网络接口。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -Ipv6AddresCount 1 `
    -Group sg-1234567890abcdef0
```

示例 2：使用特定的 IP 地址创建网络接口

使用 New-EC2NetworkInterface cmdlet。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my dual-stack network interface" `
    -PrivateIpAddress 10.251.50.12 `
    -Ipv6Address $ipv6addr `
    -Group sg-1234567890abcdef0
```

按如下所示定义 IPv6 地址。

```
$ipv6addr = New-Object Amazon.EC2.Model.InstanceIpv6Address
$ipv6addr1.Ipv6Address = "2001:db8::1234:5678:1.2.3.4"
```

示例 3：使用一组辅助 IP 地址创建网络接口

使用 New-EC2NetworkInterface cmdlet。在此示例中，Amazon EC2 选择了主 IP 地址和若干辅助 IP 地址。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -SecondaryPrivateIpAddressCount 2 `
    -Group sg-1234567890abcdef0
```

示例 4：使用特定的辅助 IP 地址创建网络接口

使用 New-EC2NetworkInterface cmdlet。此示例会指定一个主 IP 地址和一个辅助 IP 地址。

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -Description "my network interface" `
    -PrivateIpAddresses @($primary, $secondary) `
    -Group sg-1234567890abcdef0
```

按如下所示定义辅助地址。

```
$primary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$primary.PrivateIpAddress = "10.0.1.30"
$primary.Primary = $true
$secondary = New-Object Amazon.EC2.Model.PrivateIpAddressSpecification
$secondary.PrivateIpAddress = "10.0.1.31"
$secondary.Primary = $false
```

EC2 实例的网络接口连接

可以创建网络接口，供 EC2 实例用作主要或辅助网络接口。如果某 EC2 实例与某网络接口位于同一可用区中，则必须将该网络接口附加到该 EC2 实例。实例类型决定了可以附加到该实例的网络接口数量。有关更多信息，请参阅 the section called “每个网络接口的 IP 地址数”。

注意事项

- 您可以在实例运行时 (热附加)、实例停止时 (暖附加) 或实例启动时 (冷附加) 将网络接口连接到实例。
- 您可以在实例运行时或停止时分离次要网络接口。不过，您无法分离主网络接口。
- 您可以从一个实例分离辅助网络接口并将其附加到另一个实例。
- 在使用 CLI、API 或开发工具包启动实例时，您可以指定主网络接口和额外的网络接口。请注意，如果您在启动期间添加了辅助网络接口，则无法启用公有 IPv4 地址的自动分配。
- 启动具有多个网络接口的 Amazon Linux 或 Windows Server 实例会自动在该实例的操作系统上配置接口、私有 IPv4 地址和路由表。
- 如果要通过暖附加或热附加方式连接一个额外的网络接口，您可能需要手动添加第二个接口、配置私有 IPv4 地址并相应修改路由表。运行 Amazon Linux 或 Windows Server 的实例会自动识别暖附加或热附加，并自行进行配置。

- 您无法通过将另一个网络接口附加到实例（例如一种网卡绑定配置）以增加或加倍双主机实例的网络带宽。
- 如果将来自同一子网的多个网络接口连接到一个实例，则可能会遇到非对称路由等联网问题。请尽可能在主网络接口上添加辅助私有 IPv4 地址。
- 对于仅 IPv6 子网中的 EC2 实例，如果您附加辅助网络接口，则辅助网络接口的私有 DNS 主机名将解析为主网络接口的主 IPv6 地址。
- [Windows 实例] – 如果向一个实例添加多个网络接口，则必须将网络接口配置为使用静态路由。

附加网络接口

您可以使用 Amazon EC2 控制台的 Instances (实例) 或 Network Interfaces (网络接口) 页面，将网络接口连接至相同可用区中的任何实例。或者，您可以在启动实例时指定现有的网络接口。

如果您的实例上的公有 IPv4 地址已释放，并且有多个网络接口附加到实例，那么该实例不会收到新地址。有关公有 IPv4 地址行为的更多信息，请参阅公有 IPv4 地址。

Console

使用“实例”页面挂载网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选中该实例的复选框。
4. 依次选择操作、联网、附加网络接口。
5. 选择 VPC。该网络接口可以位于您的实例所在的同一 VPC 中，也可以位于您所拥有的其他 VPC 中，只要网络接口与实例位于同一可用区中即可。这使您能够在具有不同网络和安全配置的 VPC 之间创建多宿主实例。
6. 选择网络接口。如果实例支持多个网卡，则可以选择网卡。
7. 选择 Attach。

使用“网络接口”挂载网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选中该网络接口的复选框。
4. 依次选择操作、附加。

5. 选择一个实例。如果实例支持多个网卡，则可以选择网卡。
6. 选择 Attach。

Amazon CLI

挂载网络接口

使用以下 attach-network-interface 命令。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

PowerShell

挂载网络接口

使用 Add-EC2NetworkInterface cmdlet。

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1
```

分离网络接口

您可以随时使用 Amazon EC2 控制台的 Instances (实例) 或 Network Interfaces (网络接口) 页面，分离附加到 EC2 实例的辅助网络接口。

如果试图将连接至资源的网络接口从另一个服务（如 Elastic Load Balancing 负载均衡器、Lambda 函数、WorkSpace 或 NAT 网关）中分离，您将收到错误消息，提示您无权访问资源。要查找哪个服务创建了连接到网络接口的资源，请检查网络接口的说明。如果删除资源，则其网络接口将被删除。

Console

使用“实例”页面分离网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。

3. 选中该实例的复选框。检查联网选项卡的网络接口部分，以验证网络接口是否作为辅助网络接口附加到实例。
4. 依次选择操作、联网、分离网络接口。
5. 选择网络接口，然后选择分离。

使用“网络接口”页面分离网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选中该网络接口的复选框。检查详细信息选项卡的实例详细信息部分，以验证网络接口是否作为辅助网络接口附加到实例。
4. 依次选择操作、分离。
5. 当系统提示进行确认时，选择分离。
6. 如果网络接口未能与实例分离，请选择强制分离、启用，然后重试。我们建议仅将强制分离用作最后手段。强制分离可以防止您在重新启动实例之前在同一索引上附加不同的网络接口。它还可以防止实例元数据反映网络接口在您重新启动实例已分离。

Amazon CLI

分离网络接口

使用以下 detach-network-interface 命令。

```
aws ec2 detach-network-interface --attachment-id eni-attach-016c93267131892c9
```

PowerShell

分离网络接口

使用 Dismount-EC2NetworkInterface cmdlet。

```
Dismount-EC2NetworkInterface -AttachmentId eni-attach-016c93267131892c9
```

管理网络接口的 IP 地址

您可以管理网络接口的以下 IP 地址：

- 弹性 IP 地址（每个私有 IPv4 地址一个）
- IPv4 地址
- IPv6 地址

Console

管理网络接口的 IP 地址

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选中该网络接口的复选框。
4. 要管理 IPv4 和 IPv6 地址，请执行以下操作：
   a. 依次选择操作、管理 IP 地址。
   b. 展开网络接口。
   c. 对于 IPv4 地址，根据需要编辑 IP 地址。要分配其他 IPv4 地址，请选择分配新 IP 地址，然后从子网范围内指定一个 IPv4 地址，或者让 Amazon 为您选择一个。
   d. （双栈或仅 IPv6）对于 IPv6 地址，请根据需要编辑 IP 地址。要分配其他 IPv6 地址，请选择分配新 IP 地址，然后从子网范围内指定一个 IPv6 地址，或者让 Amazon 为您选择一个。
   e. 要为网络接口分配或取消分配公有 IPv4 地址，请选择自动分配公有 IP。您可以为任何网络接口启用或禁用此选项，但只会影响主网络接口。
   f. （双栈或仅 IPv6）对于 分配主 IPv6 地址，请选择启用来分配一个主 IPv6 地址。将与该网络接口关联的第一个 GUA 选择为主 IPv6 地址。分配主 IPv6 地址后，您无法对其进行更改。该地址将为主 IPv6 地址，直到实例终止或网络接口分离。
   g. 选择保存。
5. 要关联弹性 IP 地址，请执行以下操作：
   a. 依次选择操作、关联地址。
   b. 对于弹性 IP 地址，选择弹性 IP 地址。
   c. 对于私有 IPv4 地址，选择要与弹性 IP 地址关联的私有 IPv4 地址。
   d. （可选）如果网络接口当前已与其他实例或网络接口关联，选择允许重新关联弹性 IP 地址。
   e. 选择关联 。

6. 要取消关联弹性 IP 地址，请执行以下操作：

   a. 选择 Actions、Disassociate address。

   b. 对于公有 IP 地址，选择弹性 IP 地址。

   c. 选择取消关联。

Amazon CLI

管理 IPv4 地址

使用以下 assign-private-ip-addresses 命令分配 IPv4 地址。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

使用以下 unassign-private-ip-addresses 命令取消分配 IPv4 地址。

```
aws ec2 unassign-private-ip-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --private-ip-addresses 10.0.0.82
```

管理 IPv6 地址

使用以下 assign-ipv6-addresses 命令分配 IPv6 地址。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

使用以下 unassign-ipv6-addresses 命令取消分配 IPv6 地址。

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-1234567890abcdef0 \
    --ipv6-addresses 2001:db8:1234:1a00:9691:9503:25ad:1761
```

管理主私有 IPv4 地址的弹性 IP 地址

使用以下 associate-address 命令，将弹性 IP 地址关联到主私有 IPv4 地址。

```
aws ec2 associate-address \
```

```
    --allocation-id eipalloc-0b263919b6EXAMPLE \
    --network-interface-id eni-1234567890abcdef0
```

使用以下 disassociate-address 命令，将弹性 IP 地址与主私有 IPv4 地址取消关联。

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745a1EXAMPLE
```

PowerShell

管理 IPv4 地址

使用 Register-EC2PrivateIpAddress cmdlet 分配 IPv4 地址。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

使用 Unregister-EC2PrivateIpAddress cmdlet 取消分配 IPv4 地址。

```
Unregister-EC2PrivateIpAddress `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -PrivateIpAddress 10.0.0.82
```

管理 IPv6 地址

使用 Register-EC2Ipv6AddressList cmdlet 分配 IPv6 地址。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

使用 Unregister-EC2Ipv6AddressList cmdlet 取消分配 IPv6 地址。

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Ipv6Address 2001:db8:1234:1a00:9691:9503:25ad:1761
```

管理主私有 IPv4 地址的弹性 IP 地址

使用 Register-EC2Address cmdlet，将弹性 IP 地址与主私有 IPv4 地址相关联。

```
Register-EC2Address `
```

```
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -AllocationId eipalloc-0b263919b6EXAMPLE
```

使用 Unregister-EC2Address cmdlet，将弹性 IP 地址与主私有 IPv4 地址取消关联。

```
Unregister-EC2Address -AssociationId eipassoc-2bebb745a1EXAMPLE
```

修改网络接口属性

您可以更改以下网络接口属性：

- 描述
- 安全组
- 终止时删除
- 源/目标检查
- 空闲连接跟踪超时

注意事项

不能更改请求者托管式网络接口的属性。

Console

修改网络接口属性

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选中该网络接口的复选框。
4. 要更改描述，请执行以下操作：
   a. 依次选择操作、更改描述。
   b. 对于说明，输入说明。
   c. 选择保存。
5. 要更改安全组，请执行以下操作：
   a. 依次选择操作、更改安全组。

b. 对于关联安全组，根据需要添加和移除安全组。安全组和网络接口必须针对相同 VPC 创建。

c. 选择保存。

6. 要更改终止行为，请执行以下操作：

a. 依次选择操作、更改终止行为。

b. 选择或清除终止时删除、启用。

c. 选择保存。

7. 要更改源/目标检查，请执行以下操作：

a. 依次选择操作、更改源/目标检查。

b. 选择或清除源/目标、启用。

c. 选择保存。

8. 要更改空闲连接跟踪超时，请执行以下操作：

a. 选择操作、修改空闲连接跟踪超时。

b. 根据需要修改超时值。有关更多信息，请参阅 空闲连接跟踪超时。

- TCP 建立超时：处于已建立状态的空闲 TCP 连接的超时时间（以秒为单位）。
  - 最小值：60 秒
  - 最大值：432000 秒
  - 默认值：Nitro v6 实例类型（不包括 p6e-GB200）为 350 秒；所有其他实例类型（包括 p6e-GB200）为 432000 秒。
  - 建议值：小于 432000 秒
- UDP 超时：空闲 UDP 流的超时时间（以秒为单位），这些流仅在单个方向或单个请求-响应事务上看到流量。
  - 最小值：30 秒
  - 最大值：60 秒
  - 原定设置：30 秒
- UDP 流超时：空闲 UDP 流的超时（以秒为单位），这些流被归类为已看到多个请求-响应事务的流。
  - 最小值：60 秒
  - 最大值：180 秒

- 原定设置：180 秒

c. 选择保存。

Amazon CLI

Example示例：修改描述

使用以下 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --description "my updated description"
```

Example示例：修改安全组

使用以下 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

Example示例：修改终止行为

使用以下 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --attachment AttachmentId=eni-attach-43348162abEXAMPLE,DeleteOnTermination=false
```

Example示例：启用源/目标检查

使用以下 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --source-dest-check
```

Example示例：修改空闲连接跟踪超时

使用以下 modify-network-interface-attribute 命令。有关更多信息，请参阅 空闲连接跟踪超时。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --connection-tracking-specification
 TcpEstablishedTimeout=172800,UdpStreamTimeout=90,UdpTimeout=60
```

PowerShell

Example示例：修改描述

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Description "my updated description"
```

Example示例：修改安全组

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

Example示例：修改终止行为

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -Attachment_AttachmentId eni-attach-43348162abEXAMPLE `
    -Attachment_DeleteOnTermination $false
```

Example示例：启用源/目标检查

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -SourceDestCheck $true
```

Example示例：修改空闲连接跟踪超时

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。有关更多信息，请参阅 空闲连接跟踪超时。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -ConnectionTrackingSpecification_TcpEstablishedTimeout 172800 `
    -ConnectionTrackingSpecification_UdpStreamTimeout 90 `
    -ConnectionTrackingSpecification_UdpTimeout 60
```

您的 Amazon EC2 实例的多个网络接口

当您需要以下功能时，将多个网络接口附加到一个实例很有帮助：

- 一个管理网络。
- 网络和安全设备。
- 工作负载位于不同子网或 VPC 中的双主机实例。
- 一个低预算、高可用性解决方案。

管理网络

以下概述描述了使用多个网络接口创建的管理网络。

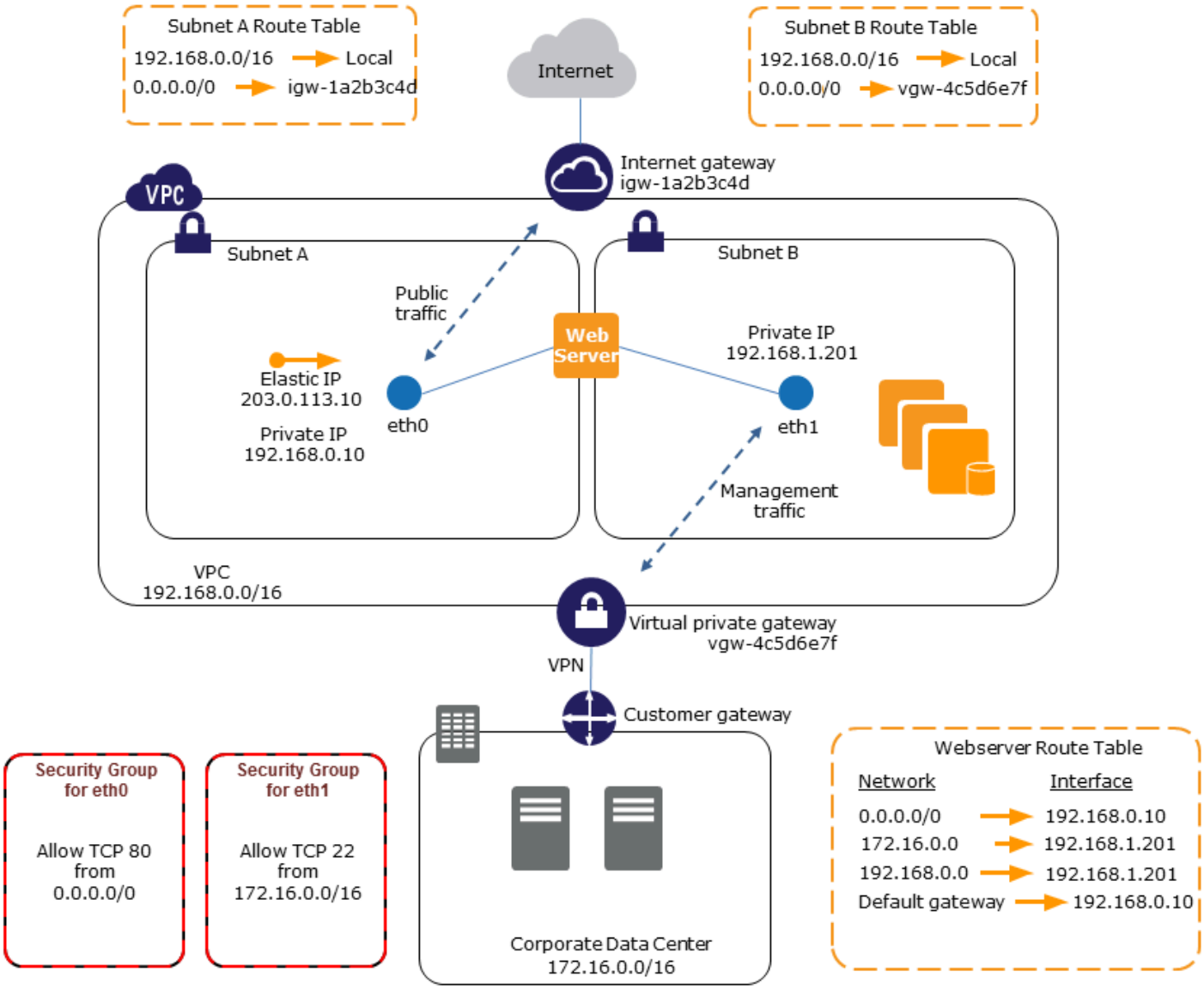
标准

- 实例上的主网络接口（例如 eth0）处理公有流量。
- 实例上的辅助网络接口（例如 eth1）处理后端管理流量。它连接到具有更严格的访问控制的单独子网，并作为主网络接口位于可用区内。

设置

- 主网络接口（可能配置或未配置负载均衡器）具有允许从 Internet 访问服务器的关联的安全组。例如，允许来自 0.0.0.0/0 或来自负载均衡器的 TCP 端口 80 和 443。
- 辅助网络接口具有一个关联的安全组，该安全组仅允许 SSH 访问，从以下位置之一启动：
  - 允许的 IP 地址范围，可以是 VPC 之内，也可以是来自互联网。
  - 与主网络接口位于同一可用区内的私有子网。
  - 虚拟私有网关。

> **Note**
>
> 为确保故障转移功能，可以考虑针对网络接口上的传入流量使用辅助私有 IPv4。在某个实例失效时，您可以将接口和/或辅助私有 IPv4 地址移动到备用实例中。



网络和安全设备

负载均衡器、网络地址转换 (NAT) 服务器和代理服务器等网络和安全设备更偏向于配置多个网络接口。您可以创建并附加辅助网络接口至正在运行这些类型的应用程序的实例中，并用实例自己的公用和私有 IP 地址、安全组和源/目标检查设置其他接口。

工作负载位于不同子网中的双主机实例

您可以将网络接口放置到每一个与承载应用程序服务器的中间层网络相连接的 Web 服务器。应用程序服务器也可以用双主机连接至承载数据库服务器的后端网络（子网）。每一个双主机实例都在前端接收和处理请求、启动与后端的连接，然后将请求发送至后端网络上的服务器，而不是通过双主机实例路由网络数据包。

工作负载位于同一账户下不同 VPC 中的双主机实例

您可以在一个 VPC 中启动 EC2 实例，并从其他 VPC 附加辅助 ENI，只要网络接口与实例位于同一可用区中即可。这使您能够在具有不同网络和安全配置的 VPC 之间创建多宿主实例。您不能在不同 Amazon 账户中的 VPC 之间创建多宿主实例。

在以下使用案例中，您可以跨 VPC 使用双宿主实例：

- 克服两个无法对等互连的 VPC 之间的 CIDR 重叠：您可以利用 VPC 中的辅助 CIDR 并允许实例跨两个不重叠的 IP 范围进行通信。
- 在单个账户中连接多个 VPC：启用通常由 VPC 边界分隔的各个资源之间的通信。

低预算、高可用性解决方案

如果您的一个提供特定功能的实例失效，则其网络接口可附加到一个针对同一种角色预配置的替代或热备用实例，以快速恢复服务。例如，您可以将一个网络接口用作连接数据库实例或 NAT 实例等关键服务的主要或辅助网络接口。如果实例失效，您 (或更有可能是代表您运行的代码) 可以将网络接口附加到热备用实例。由于接口保持其私有 IP 地址、弹性 IP 地址和 MAC 地址，因此只要您将网络接口附加到替代实例，网络流量就会立即开始流向备用实例。在实例失效之后、网络接口附加到备用实例之前，用户会暂时失去连接，但不需要更改路由表或您的 DNS 服务器。

请求者托管的网络接口

请求者托管式网络接口是 Amazon Web Services 服务 代表您在 VPC 中创建的网络接口。网络接口与其他服务的资源相关联，例如 Amazon RDS 的数据库实例、NAT 网关或 Amazon PrivateLink 的接口 VPC 端点。

注意事项

- 您可以在账户中查看请求者托管式网络接口，并可以添加或删除标签。但是，您无法对请求者管理的网络接口执行以下操作：

- 修改网络接口属性
- 附加或分离网络接口
- 关联或撤销关联弹性 IP 地址
- 分配或取消分配私有 IP 地址
- 在启动实例时指定网络接口

虽然无法修改网络接口属性，但您仍然可以进行重置

- 如果您删除与请求者托管式网络接口关联的资源，Amazon Web Services 服务 会断开网络接口并将其删除。如果服务断开了网络接口但没有将其删除，您可以删除断开的网络接口。

Console

查看请求者托管式网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，依次选择 Network & Security（网络与安全）、Network Interfaces（网络接口）。
3. 选择网络接口的 ID 以打开其详细信息页面。
4. 以下是可用于确定网络接口用途的关键字段：
   - Description（描述）：由创建接口的 Amazon 服务提供的描述。例如，“VPC Endpoint Interface vpce 089f2123488812123（VPC 端点接口 vpce 089f2123488812123）”。
   - Requester-managed（请求者托管式）：指明网络接口是否由 Amazon 托管。
   - Requester ID（请求者 ID）：创建网络接口的主体或服务的别名或 Amazon 账户 ID。如果您创建了网络接口，这便是您的 Amazon Web Services 账户 ID。如果没有，则由其他主体或服务创建。

Amazon CLI

查看请求者托管式网络接口

按如下方式使用 describe-network-interfaces 命令。

```
aws ec2 describe-network-interfaces \
    --filters "Name=requester-managed,Values=true" \
    --query "NetworkInterfaces[*].[Description, InterfaceType]" \
```

```
    --output table
```

以下示例输出显示了可用于确定网络接口用途的关键字段 Description 和 InterfaceType。

```
-------------------------------------------------------------------------
|                          DescribeNetworkInterfaces                    |
+---------------------------------------------------+-------------------+
|  VPC Endpoint Interface: vpce-0f00567fa8477a1e6   |  interface        |
|  VPC Endpoint Interface vpce-0d8ddce4be80e4474    |  interface        |
|  VPC Endpoint Interface vpce-078221a1e27d1ea5b    |  vpc_endpoint     |
|  Resource Gateway Interface rgw-0bba03f3d56060135 |  interface        |
|  VPC Endpoint Interface: vpce-0cc199f605eaeace7   |  interface        |
|  VPC Endpoint Interface vpce-019b90d6f16d4f958    |  interface        |
+---------------------------------------------------+-------------------+
```

PowerShell

查看请求者托管式网络接口

按如下所示使用 Get-EC2NetworkInterface cmdlet。

```
Get-EC2NetworkInterface -Filter @{Name="requester-managed"; Values="true"} | Select
 Description, InterfaceType
```

以下示例输出显示了可用于确定网络接口用途的关键字段 Description 和 InterfaceType。

```
Description                                       InterfaceType
-----------                                       -------------
VPC Endpoint Interface: vpce-0f00567fa8477a1e6    interface
VPC Endpoint Interface vpce-0d8ddce4be80e4474     interface
VPC Endpoint Interface vpce-078221a1e27d1ea5b     vpc_endpoint
Resource Gateway Interface rgw-0bba03f3d56060135  interface
VPC Endpoint Interface: vpce-0cc199f605eaeace7    interface
VPC Endpoint Interface vpce-019b90d6f16d4f958     interface
```

Amazon EC2 网络接口的前缀委派

您可以自动或手动为网络接口分配私有 IPv4 或 IPv6 CIDR 范围。通过分配前缀，您可以扩展和简化应用程序的管理，包括一个实例上需要多个 IP 地址的容器和网络应用程序。有关 IPv4 和 IPv6 地址的更多信息，请参阅 Amazon EC2 实例 IP 寻址。

可用的分配选项如下：

- 自动分配 - Amazon选择前缀并将其分配给您的网络接口。如果网络接口的子网的子网 CIDR 预留类型为`prefix`，我们会从子网 CIDR 预留中选择前缀。否则，我们会从子网 CIDR 范围中选择它们。
- 手动分配 – 您指定前缀后，Amazon 会在将前缀分配给您的网络接口之前，验证该前缀是否尚未分配给其他资源。

分配前缀具有以下优势：

- 增加了网络接口上的 IP 地址 – 使用前缀时，您可以分配一个 IP 地址块，而不是单个 IP 地址。这会增加网络接口的 IP 地址数量。
- 简化了容器的 VPC 管理 – 在容器应用程序中，每个容器都需要一个唯一的 IP 地址。为您的实例分配前缀可简化 VPC 的管理，因为您可以启动和终止容器，而无需为单个 IP 分配调用 Amazon EC2 API。

目录

- 基本功能
- 注意事项
- 管理网络接口的前缀
  - 在网络接口创建过程中分配前缀
  - 分配前缀至现有网络接口
  - 从网络接口中移除前缀

基本功能

- 您可以为新的或现有网络接口分配前缀。
- 要使用前缀，请为您的网络接口分配前缀，将该网络接口连接至您的实例，然后配置您的操作系统。
- 选择指定前缀的选项时，前缀必须满足以下要求：
  - 您可以指定的 IPv4 前缀为 `/28`。
  - 您可以指定的 IPv6 前缀为 `/80`。
  - 前缀位于网络接口的子网 CIDR 中，并且不会与分配给子网中现有资源的其他前缀或 IP 地址重叠。
- 您可以为主网络接口或辅助网络接口分配前缀。

- 您可以将弹性 IP 地址分配给有前缀的网络接口。
- 您也可以将弹性 IP 地址分配给已分配前缀的 IP 地址。
- 我们将实例的私有 DNS 主机名解析为主私有 IPv4 地址。
- 我们使用以下格式为网络接口分配每个私有 IPv4 地址，包括前缀中的地址：
  - us-east-1 区域

    ```
    ip-private-ipv4-address.ec2.internal
    ```

  - 所有其他区域

    ```
    ip-private-ipv4-address.region.compute.internal
    ```

注意事项

使用前缀时请注意以下事项：

- 基于 Nitro 的实例支持带有前缀的网络接口。
- 网络接口的前缀必须使用 IPv6 地址和私有 IPv4 地址。
- 您可以分配给网络接口的 IP 地址的最大数量取决于实例类型。您分配给网络接口的每个前缀都算作一个 IP 地址。例如，c5.large 实例具有每个网络接口 10 个 IPv4 地址的限制。此实例的每个网络接口都有一个主 IPv4 地址。如果网络接口没有辅助 IPv4 地址，则您最多可以为网络接口分配 9 个前缀。您每为网络接口多分配一个 IPv4 地址，则您就可以为网络接口少分配一个前缀。有关更多信息，请参阅 每个网络接口的最大 IP 地址数。
- 前缀包含在源/目标检查中。
- 您必须配置操作系统，才能使用带有前缀的网络接口。请注意以下几点：
  - 一些 Amazon Linux AMI 包含由 Amazon 安装的其他脚本，它们称为 ec2-net-utils。这些脚本可以选择性地自动配置您的网络接口。它们仅在 Amazon Linux 上使用。
  - 对于容器，您可以将容器网络接口（CNI）用于 Kubernetes 插件或 dockerd（如果您使用 Docker 管理容器）。

管理网络接口的前缀

在为网络接口分配前缀时，您可以选择是否让我们自动分配前缀，您也可以指定自定义前缀。如果您让我们自动分配前缀，并且网络接口的子网的子网 CIDR 预留类型为prefix，我们会从子网 CIDR 预留中选择前缀。否则，我们会从子网 CIDR 范围中选择它们。

任务

- 在网络接口创建过程中分配前缀
- 分配前缀至现有网络接口
- 从网络接口中移除前缀

在网络接口创建过程中分配前缀

您可以在创建网络接口时分配自动或自定义前缀。

Console

在网络接口创建过程中自动分配前缀

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选择创建网络接口。
4. 输入网络接口的描述，选择要在其中创建网络接口的子网，并配置私有 IPv4 和 IPv6 地址。
5. 展开 Advanced settings（高级设置）。
6. 对于 IPv4 前缀委派，请执行下列操作之一：
   - 要自动分配 IPv4 前缀，请选择自动分配。对于 IPv4 前缀的数量，输入要分配的前缀数。
   - 要分配特定 IPv4 前缀，请选择自定义。选择添加新的前缀，然后输入该前缀。
7. 对于 IPv6 前缀委派，请执行下列操作之一：
   - 要自动分配 IPv6 前缀，请选择自动分配。对于 IPv6 前缀的数量，输入要分配的前缀数。
   - 要分配特定 IPv6 前缀，请选择自定义。选择添加新的前缀，然后输入该前缀。

   > **Note**
   >
   > IPv6 prefix delegation（IPv6 前缀委派）仅当为 IPv6 启用了选定的子网时才会显示。

8. 选择要与网络接口关联的安全组，并根据需要分配资源标签。
9. 选择创建网络接口。

Amazon CLI

在网络接口创建过程中自动分配 IPv4 前缀

使用 create-network-interface 命令并将 --ipv4-prefix-count 设置为要让 Amazon 分配的 IPv4 前缀数。在下面的示例中，Amazon 分配一个 IPv4 前缀。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv4 automatic example" \
    --ipv4-prefix-count 1
```

在网络接口创建过程中分配特定 IPv4 前缀

使用 create-network-interface 命令并将 --ipv4-prefixes 设置为该前缀范围。Amazon 会从此范围中选择 IPv4 地址。在下面的示例中，前缀 CIDR 为 10.0.0.208/28。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv4 manual example" \
    --ipv4-prefixes Ipv4Prefix=10.0.0.208/28
```

在网络接口创建过程中自动分配 IPv6 前缀

使用 create-network-interface 命令并将 --ipv6-prefix-count 设置为要让 Amazon 分配的 IPv6 前缀数。在下面的示例中，Amazon 分配一个 IPv6 前缀。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv6 automatic example" \
    --ipv6-prefix-count 1
```

在网络接口创建过程中分配特定 IPv6 前缀

使用 create-network-interface 命令并将 --ipv6-prefixes 设置为该前缀范围。Amazon 会从此范围中选择 IPv6 地址。在下面的示例中，前缀 CIDR 为 2600:1f13:fc2:a700:1768::/80。

```
aws ec2 create-network-interface \
    --subnet-id subnet-047cfed18eEXAMPLE \
    --description "IPv6 manual example" \
    --ipv6-prefixes Ipv6Prefix=2600:1f13:fc2:a700:1768::/80
```

PowerShell

在网络接口创建过程中自动分配 IPv4 前缀

使用 New-EC2NetworkInterface cmdlet 并将 `Ipv4PrefixCount` 设置为要让 Amazon 分配的 IPv4 前缀数。在下面的示例中，Amazon 分配一个 IPv4 前缀。

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 automatic example' `
    -Ipv4PrefixCount 1
```

在网络接口创建过程中分配特定 IPv4 前缀

使用 New-EC2NetworkInterface cmdlet 并将 `Ipv4Prefix` 设置为该前缀范围。Amazon 会从此范围中选择 IPv4 地址。在下面的示例中，前缀 CIDR 为 10.0.0.208/28。

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv4 manual example' `
    -Ipv4Prefix (New-Object `
        -TypeName Amazon.EC2.Model.Ipv4PrefixSpecificationRequest `
        -Property @{Ipv4Prefix = '10.0.0.208/28'})
```

在网络接口创建过程中自动分配 IPv6 前缀

使用 New-EC2NetworkInterface cmdlet 并将 `Ipv6PrefixCount` 设置为要让 Amazon 分配的 IPv6 前缀数。在下面的示例中，Amazon 分配一个 IPv6 前缀。

```
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 automatic example' `
    -Ipv6PrefixCount 1
```

在网络接口创建过程中分配特定 IPv6 前缀

使用 New-EC2NetworkInterface cmdlet 并将 `Ipv6Prefixes` 设置为该前缀范围。Amazon 会从此范围中选择 IPv6 地址。在下面的示例中，前缀 CIDR 为 2600:1f13:fc2:a700:1768::/80。

```
Import-Module AWS.Tools.EC2
New-EC2NetworkInterface `
    -SubnetId 'subnet-047cfed18eEXAMPLE' `
    -Description 'IPv6 manual example' `
    -Ipv6Prefix (New-Object `
```

```
    -TypeName Amazon.EC2.Model.Ipv6PrefixSpecificationRequest `
    -Property @{Ipv6Prefix = '2600:1f13:fc2:a700:1768::/80'})
```

分配前缀至现有网络接口

您可以为现有网络接口分配自动或自定义前缀。

Console

自动分配前缀至现有网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选择要为其分配前缀的网络接口，然后选择 Actions（操作）、Manage prefixes（管理前缀）。
4. 对于 IPv4 前缀委派，请执行下列操作之一：
   - 要自动分配 IPv4 前缀，请选择自动分配。对于 IPv4 前缀的数量，输入要分配的前缀数。
   - 要分配特定 IPv4 前缀，请选择自定义。选择添加新的前缀，然后输入该前缀。
5. 对于 IPv6 前缀委派，请执行下列操作之一：
   - 要自动分配 IPv6 前缀，请选择自动分配。对于 IPv6 前缀的数量，输入要分配的前缀数。
   - 要分配特定 IPv6 前缀，请选择自定义。选择添加新的前缀，然后输入该前缀。

   > **Note**
   >
   > IPv6 prefix delegation（IPv6 前缀委派）仅当为 IPv6 启用了选定的子网时才会显示。

6. 选择保存。

Amazon CLI

使用 assign-ipv6-addresses 命令为现有网络接口分配 IPv6 前缀，并使用 assign-private-ip-addresses 命令为现有网络接口分配 IPv4 前缀。

自动分配 IPv4 前缀至现有网络接口

使用 assign-private-ip-addresses 命令并将 `--ipv4-prefix-count` 设置为要让 Amazon 分配的 IPv4 前缀数。在下面的示例中，Amazon 分配一个 IPv4 前缀。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefix-count 1
```

分配特定 IPv4 前缀至现有网络接口

使用 assign-private-ip-addresses 命令并设置 `--ipv4-prefixes` 至前缀。Amazon 会从此范围中选择 IPv4 地址。在下面的示例中，前缀 CIDR 为 10.0.0.208/28。

```
aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefixes 10.0.0.208/28
```

自动分配 IPv6 前缀至现有网络接口

使用 assign-ipv6-addresses 命令并将 `--ipv6-prefix-count` 设置为要让 Amazon 分配的 IPv6 前缀数。在下面的示例中，Amazon 分配一个 IPv6 前缀。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefix-count 1
```

分配特定 IPv6 前缀至现有网络接口

使用 assign-ipv6-addresses 命令并设置 `--ipv6-prefixes` 至前缀。Amazon 会从此范围中选择 IPv6 地址。在下面的示例中，前缀 CIDR 为 2600:1f13:fc2:a700:18bb::/80。

```
aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

自动分配 IPv4 前缀至现有网络接口

使用 Register-EC2PrivateIpAddress cmdlet 并将 `Ipv4PrefixCount` 设置为要让 Amazon 分配的 IPv4 前缀数。在下面的示例中，Amazon 分配一个 IPv4 前缀。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
```

```
    -Ipv4PrefixCount 1
```

分配特定 IPv4 前缀至现有网络接口

使用 Register-EC2PrivateIpAddress cmdlet 并将 `Ipv4Prefix` 设置为该前缀。Amazon 会从此范围中选择 IPv4 地址。在下面的示例中，前缀 CIDR 为 10.0.0.208/28。

```
Register-EC2PrivateIpAddress `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

自动分配 IPv6 前缀至现有网络接口

使用 Register-EC2Ipv6AddressList cmdlet 并将 `Ipv6PrefixCount` 设置为要让 Amazon 分配的 IPv4 前缀数。在下面的示例中，Amazon 分配一个 IPv6 前缀。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6PrefixCount 1
```

分配特定 IPv6 前缀至现有网络接口

使用 Register-EC2Ipv6AddressList cmdlet 并将 `Ipv6Prefix` 设置为该前缀。Amazon 会从此范围中选择 IPv6 地址。在下面的示例中，前缀 CIDR 为 2600:1f13:fc2:a700:18bb::/80。

```
Register-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

从网络接口中移除前缀

您可以从现有网络接口中删除前缀。

Console

从网络接口中移除前缀

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选择网络接口。

4. 选择操作，管理前缀。
5. 对于 IPv4 前缀委派，要删除特定的前缀，请选择要删除的前缀旁边的取消分配。要移除所有前缀，请选择不分配。
6. 对于 IPv6 前缀委派，要删除特定的前缀，请选择要删除的前缀旁边的取消分配。要移除所有前缀，请选择不分配。

> **Note**
>
> IPv6 prefix delegation（IPv6 前缀委派）仅当为 IPv6 启用了选定的子网时才会显示。

7. 选择保存。

Amazon CLI

您可以使用 unassign-ipv6-addresses 命令来移除 IPv6 前缀，使用 unassign-private-ip-addresses 命令将 IPv4 前缀从现有网络接口中移除。

从网络接口中移除 IPv4 前缀

使用 unassign-private-ip-addresses 命令并将 `--ipv4-prefix` 设置为要移除的前缀 CIDR。

```
aws ec2 unassign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefixes 10.0.0.176/28
```

从网络接口中移除 IPv6 前缀

使用 unassign-ipv6-addresses 命令并将 `--ipv6-prefix` 设置为要移除的前缀 CIDR。

```
aws ec2 unassign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefix 2600:1f13:fc2:a700:18bb::/80
```

PowerShell

从网络接口中移除 IPv4 前缀

使用 Unregister-EC2PrivateIpAddress cmdlet 并将 `Ipv4Prefix` 设置为要移除的前缀 CIDR。

```
Unregister-EC2PrivateIpAddress `
```

```
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv4Prefix '10.0.0.208/28'
```

从网络接口中移除 IPv6 前缀

使用 Unregister-EC2Ipv6AddressList cmdlet 并将 `Ipv6Prefix` 设置为要移除的前缀 CIDR。

```
Unregister-EC2Ipv6AddressList `
    -NetworkInterfaceId 'eni-00d577338cEXAMPLE' `
    -Ipv6Prefix '2600:1f13:fc2:a700:18bb::/80'
```

删除网络接口

删除网络接口之后，所有与该接口关联的属性都会被释放，而且所有私有 IP 地址或弹性 IP 地址也都会被释放以供另一个实例使用。

您无法删除正在使用的网络接口。首先，您必须分离网络接口。

Console

删除网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择网络接口。
3. 选中网络接口的复选框，然后依次选择操作、删除。
4. 当系统提示进行确认时，选择 Delete（删除）。

Amazon CLI

删除网络接口

使用以下 delete-network-interface 命令。

```
aws ec2 delete-network-interface --network-interface-id eni-1234567890abcdef0
```

PowerShell

删除网络接口

使用 Remove-EC2NetworkInterface cmdlet。

```
Remove-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0
```

Amazon EC2 实例网络带宽

实例带宽规格适用于实例的入站和出站流量。例如，假设实例指定了高达 10 Gbps 的带宽，这意味着该实例将同时具有高达 10 Gbps 的入站流量带宽和高达 10 Gbps 的出站流量带宽。EC2 实例可用的网络带宽取决于多个因素，如下所示。

多流流量

当流量通过互联网网关或本地网关时，多流流量的可用带宽会取决于实例类型。

实例类型	可用带宽
vCPU 少于 32 个的实例类型	限制为 5Gbps
vCPU 超过 32 个的实例类型	限制为实例类型可用带宽的 50%
C8in、C8ine、M8in、M8ine、M8idn、R8in、R8idn 实例	限制为实例类型的基准带宽

单流流量

如果实例不在同一集群置放群组中，单流流量的带宽限制为 5Gbps。要减少延迟并增加单流带宽，请尝试以下任一操作：

- 使用集群置放群组，使同一置放群组内的实例具有高达 10 Gbps 的带宽。
- 在任意两个端点之间设置多条路径，以通过多路径 TCP（MPTCP）实现更高带宽。
- 为同一可用区内符合条件的实例配置 ENA Express，以便在这些实例之间实现高达 25 Gbps 的带宽。

> **Note**
>
> 单流被视作唯一的 5 元组 TCP 或 UDP 流。对于位于 IP 标头之后的其他协议（例如 GRE 或 IPsec），使用源 IP、目标 IP 和下一个协议的 3 元组来定义流。

可用实例带宽

实例的可用网络带宽取决于其拥有的 vCPU 数量。例如，m5.8xlarge 实例有 32 个 vCPU 和 10 Gbps 的网络带宽，m5.16xlarge 实例有 64 个 vCPU 和 20 Gbps 的网络带宽。但是，如果实例超过实例级别的网络限额（例如每秒数据包或跟踪的连接数），则实例可能无法实现此带宽。流量可以利用多少可用带宽取决于 vCPU 的数量和目的地。例如，m5.16xlarge 实例具有 64 个 vCPU，因此到该区域中另一个实例的流量可以利用完整的可用带宽 (20 Gbps)。但是，通过互联网网关或本地网关的流量只能利用 50% 的可用带宽 (10 Gbps)。

通常，有 16 个或更少 vCPU 的实例（大小为 4xlarge 或更小）被记录为具有“高达”的指定带宽；例如，“高达 10 Gbps”。这些实例具备基准带宽。为满足其他需求，可以使用网络 I/O 积分机制，以突增超出其基准带宽。实例可以在有限时间内使用突增带宽，通常为 5 到 60 分钟，具体取决于实例的大小。

实例在启动时接收最大数量的网络 I/O 积分。如果实例耗尽了其网络 I/O 积分，则返回至其基准带宽。正在运行的实例只要使用少于其基准带宽的网络带宽，就会获得网络 I/O 积分。已停止的实例不会获得网络 I/O 积分。即使实例有可用积分，实例突增也已尽了最大努力，因为突增带宽是一个共享资源。

入站和出站流量都有单独的网络 I/O 积分桶。

基本和爆发网络性能

《Amazon EC2 实例类型指南》介绍了每种实例类型的网络性能，以及可以使用突增带宽的实例可用的基准网络带宽。有关更多信息，请参阅下列内容：

- 网络规格 – 通用型
- 网络规格 – 计算优化型
- 网络规格 – 内存优化型
- 网络规格 – 存储优化型
- 网络规格 – 加速计算型
- 网络规格 – 高性能计算型
- 网络规格 – 上一代

或者，您可以使用命令行工具来获取此信息。Amazon EC2 控制台不显示实例类型的基准网络带宽。

Amazon CLI

您可以使用 describe-instance-types 命令来显示有关实例类型的信息。以下示例显示了所有 C5 实例的网络性能信息。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=c5.*" \
    --query "InstanceTypes[].[InstanceType, NetworkInfo.NetworkPerformance,
 NetworkInfo.NetworkCards[0].BaselineBandwidthInGbps] | sort_by(@,&[2])" \
    --output table
```

下面是示例输出。如果您的输出未达到基准带宽，请更新到最新版本的 Amazon CLI。

```
-----------------------------------------------
|           DescribeInstanceTypes             |
+--------------+--------------------+---------+
|  c5.large    |  Up to 10 Gigabit  |  0.75 |
|  c5.xlarge   |  Up to 10 Gigabit  |  1.25 |
|  c5.2xlarge  |  Up to 10 Gigabit  |  2.5  |
|  c5.4xlarge  |  Up to 10 Gigabit  |  5.0  |
|  c5.9xlarge  |  12 Gigabit        |  12.0 |
|  c5.12xlarge |  12 Gigabit        |  12.0 |
|  c5.18xlarge |  25 Gigabit        |  25.0 |
|  c5.24xlarge |  25 Gigabit        |  25.0 |
|  c5.metal    |  25 Gigabit        |  25.0 |
+--------------+--------------------+---------+
```

PowerShell

您可以使用 Get-EC2InstanceType PowerShell 命令来显示有关实例类型的信息。以下示例显示了所有 C5 实例的网络性能信息。

```
Get-EC2InstanceType -Filter @{Name = "instance-type"; Values = "c5.*" } | `
    Select-Object `
    InstanceType,
    @{Name = 'NetworkPerformance'; Expression =
 {($_.Networkinfo.NetworkCards.NetworkPerformance)}},
    @{Name = 'BaselineBandwidthInGbps'; Expression =
 {($_.Networkinfo.NetworkCards.BaselineBandwidthInGbps)}} | `
Format-Table -AutoSize
```

下面是示例输出。

```
InstanceType NetworkPerformance BaselineBandwidthInGbps
------------ ------------------ ----------------------
c5.4xlarge   Up to 10 Gigabit                     5.00
```

```
c5.xlarge    Up to 10 Gigabit                    1.25
c5.12xlarge  12 Gigabit                         12.00
c5.9xlarge   12 Gigabit                         12.00
c5.24xlarge  25 Gigabit                         25.00
c5.metal     25 Gigabit                         25.00
c5.2xlarge   Up to 10 Gigabit                    2.50
c5.large     Up to 10 Gigabit                    0.75
c5.18xlarge  25 Gigabit                         25.00
```

EC2 实例带宽权重配置

某些实例类型支持可配置的带宽权重，即允许选择有利于网络处理或 EBS 操作的基准带宽权重。基准带宽的默认设置由实例类型决定。您可以在启动期间配置带宽权重，也可以使用以下权重首选项修改实例设置：

- 默认值：此选项会对实例类型使用标准带宽配置。
- vpc-1：此选项会增加可用于网络的基准带宽，减少用于 EBS 操作的基准带宽。
- ebs-1：此选项会增加可用于 EBS 操作的基准带宽，减少用于网络的基准带宽。

带宽权重注意事项

以下是一些可能会影响带宽权重策略的注意事项。

- 设置带宽权重首选项只会对带宽规格产生影响。每秒网络数据包数（PPS）和 EBS 每秒进行读写操作的次数（IOPS）规格不会发生改变。
- 网络和 EBS 之间的总带宽规格并无变化。在选择带宽权重配置时，选定选项的可用基准带宽会增加，其余选项的基准带宽会减少相同的绝对量。对于除 Flex 实例之外的所有实例，选定选项的可用突增带宽保持不变，其余选项的可用突增带宽会相应减少。对于高达 4xlarge 的 Flex 实例，突发带宽保持不变。对于 8xlarge 及更大的 Flex 实例，突发带宽增加的量与基准带宽相同。
- 了解带宽分配的变化如何影响 EBS 的 I/O 性能非常重要。如果 EC2 实例具有 `vpc-1` 配置（增加网络带宽），若在达到 IOPS 限制之前达到了 EBS 带宽限制，可能会降低 EBS 卷的 IOPS。这种情况在 I/O 大小较大时更为明显。

  例如，对于通常支持 24 万 IOPS、I/O 大小为 16 KiB 的实例类型，如果选择 `vpc-1` 权重，则会因调整了 EBS 基准带宽限制而降低可实现的 IOPS。

在规划工作负载时，要考虑到 I/O 大小和模式。较小的 I/O 大小受带宽限制影响的可能性较小，较大的 I/O 大小或连续工作负载受带宽变化影响的可能性更大。务必测试特定工作负载，确保所选配置具有最佳性能。

- 通过互联网网关或本地网关的流量的网络多流带宽规格，调整为配置选项基准带宽的 50% 或 5 Gbps（若适用）。有关更多信息，请参阅 Amazon EC2 实例网络带宽。

  以下示例以默认基准带宽为 40 Gbps、默认边界带宽为 20 Gbps 的实例类型为基础。如果为此实例选择 `vpc-1` 带宽权重，则加权基准带宽会更改为 50 Gbps，边界带宽会更改为 25 Gbps。
- 此功能在所有商业区域均可用，与 EC2 实例的可用性和支持保持一致。
- 此功能不会给 EC2 实例增加额外费用。

带宽权重支持的实例类型

以下实例系列中的实例类型支持可配置的带宽权重。

- 通用型：M8a、M8g、M8gd、M8i、M8id、M8i-flex、M9g、M9gd
- 计算优化型：C8a、C8g、C8gd、C8i、C8id、C8i-flex
- 内存优化型：R8a、R8g、R8gd、R8i、R8id、R8i-flex、X8g、X8aedz、X8i

查看当前带宽设置

要查看实例的当前带宽设置，请选择下方某个选项卡获取相应说明。

Console

获取实例的带宽设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 从列表中选择要查看的实例，然后导航到网络选项卡。当前设置显示在已配置带宽字段中。如果带宽未设置为特定值，Amazon EC2 会对实例类型使用默认设置。

Amazon CLI

获取实例的带宽设置

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].NetworkPerformanceOptions.BandwidthWeighting
 \
    --output text
```

下面是示例输出。

```
default
```

此示例会列出通过将带宽权重首选项设置为 vpc-1 来获得更高网络带宽的所有实例。

```
aws ec2 describe-instances \
    --filters "Name=network-performance-options.bandwidth-weighting,Values=vpc-1" \
    --query Reservations[].Instances[].InstanceId \
    --output text
```

PowerShell

获取实例的带宽设置

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.NetworkPerformanceOptions.BandwidthWeighting.Value
```

下面是示例输出。

```
default
```

此示例会列出通过将带宽权重首选项设置为 vpc-1 来获得更高网络带宽的所有实例。

```
(Get-EC2Instance `
    -Filter @{Name="network-performance-options.bandwidth-
weighting";Values="vpc-1"}).Instances.InstanceId
```

配置实例带宽权重

您可以在启动时配置带宽权重，也可以使用 EC2 控制台、API/SDK 或 CLI 修改现有实例来配置带宽权重。

启动实例时配置带宽权重

要在启动实例时配置带宽设置，请选择下方某个选项卡获取相应说明。

您还可以在启动模板中指定带宽权重。要创建启动模板，请参阅创建 Amazon EC2 启动模板。要设置的参数与通过控制台直接启动的实例位于同一位置。展开高级详细信息部分，设置实例带宽配置。

要使用启动模板启动实例，请参阅使用启动模板来启动 EC2 实例。

Console

启动可配置带宽权重的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择 Launch instances。此操作会打开启动实例对话框。您也可以根据偏好，选择其他几种方式进入启动对话框。例如，您可以直接通过 AMI 或 Amazon EC2 控制面板本身启动实例。
4. 从中启动实例的亚马逊机器映像（AMI）必须基于 `Arm` 架构。许多快速入门映像同时支持 `x86` 和 `Arm` 架构。在为实例选择操作系统后，请从架构列表中选择相应的 `Arm` 选项。
5. 实例类型必须是该功能的一个 支持的实例类型。
6. 展开高级详细信息部分后，可以向下滚动找到实例带宽配置设置。为实例选择带宽配置选项。
7. 照常配置实例的其余设置，然后选择启动实例。

Amazon CLI

启动可配置带宽权重的实例

使用 run-instances 命令和以下选项，启动配置了更高网络带宽权重的实例。

```
--network-performance-options BandwidthWeighting=vpc-1
```

使用 run-instances 命令和以下选项，启动为配置了更高 EBS 带宽权重的实例。

```
--network-performance-options BandwidthWeighting=ebs-1
```

PowerShell

启动可配置带宽权重的实例

使用 New-EC2Instance cmdlet 和以下参数，启动配置了更高网络带宽权重的实例。

```
-NetworkPerformanceOptions_BandwidthWeighting vpc-1
```

使用 New-EC2Instance cmdlet 和以下参数，启动配置了更高 EBS 带宽权重的实例。

```
-NetworkPerformanceOptions_BandwidthWeighting ebs-1
```

更新现有实例的带宽权重

要更新现有实例的带宽权重，实例必须处于 Stopped 状态。

Console

更新带宽权重

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 从列表中选择要更新的实例。
4. 在更改带宽配置之前，实例必须处于 Stopped 状态。如果实例正在运行，请从实例状态菜单中选择停止实例。
5. 从操作 > 网络菜单中选择管理带宽。此操作会打开管理带宽对话框。

   > **Note**
   >
   > 如果实例类型不支配置带宽权重，则会禁用该菜单项。

6. 选择要用于更新实例的选项，再选择更改保存设置。

Amazon CLI

更新带宽权重

使用 modify-instance-network-performance-options 命令，为指定实例配置更高的网络带宽权重。

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=vpc-1
```

以下示例为指定实例配置了更高的 EBS 带宽权重。

```
aws ec2 modify-instance-network-performance-options \
    --instance-id i-1234567890abcdef0 \
    --bandwidth-weighting=ebs-1
```

PowerShell

更新带宽权重

使用 Edit-EC2InstanceNetworkPerformanceOption cmdlet，为指定实例配置更高的网络带宽权重。

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting vpc-1
```

以下示例为指定实例配置了更高的 EBS 带宽权重。

```
Edit-EC2InstanceNetworkPerformanceOption `
    -InstanceId i-1234567890abcdef0 `
    -BandwidthWeighting ebs-1
```

联网性能

下表显示了使用 `default`、`vpc-1` 和 `ebs-1` 配置可以实现的网络性能（以 Gbps 为单位）。

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8a.medium	0.52/12.5	0.65/12.5	0.438 / 10.0
c8a.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8a.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8a.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8a.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8a.8xlarge	15.0	18.75	12.5
c8a.12xlarge	22.5	28.125	18.75
c8a.16xlarge	30.0	37.5	25.0
c8a.24xlarge	40.0	50.0	32.5
c8a.48xlarge	75.0	93.75	60.0
c8a.metal-24xl	40.0	50.0	32.5
c8a.metal-48xl	75.0	93.75	60.0
c8g.medium	0.52/12.5	0.65/12.5	0.441 / 10.0
c8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
c8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8g.8xlarge	15.0	18.75	12.5
c8g.12xlarge	22.5	28.125	18.75
c8g.16xlarge	30.0	37.5	25.0
c8g.24xlarge	40.0	50.0	32.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8g.48xlarge	50.0	62.5	40.0
c8g.metal-24xl	40.0	50.0	32.5
c8g.metal-48xl	50.0	62.5	40.0
c8gd.medium	0.52/12.5	0.65/12.5	0.441 / 10.0
c8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
c8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8gd.8xlarge	15.0	18.75	12.5
c8gd.12xlarge	22.5	28.125	18.75
c8gd.16xlarge	30.0	37.5	25.0
c8gd.24xlarge	40.0	50.0	32.5
c8gd.48xlarge	50.0	62.5	40.0
c8gd.metal-24xl	40.0	50.0	32.5
c8gd.metal-48xl	50.0	62.5	40.0
c8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
c8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8i.8xlarge	15.0	18.75	12.5
c8i.12xlarge	22.5	28.125	18.75
c8i.16xlarge	30.0	37.5	25.0
c8i.24xlarge	40.0	50.0	32.5
c8i.32xlarge	50.0	62.5	40.0
c8i.48xlarge	75.0	93.75	60.0
c8i.96xlarge	100.0	125.0	80.0
c8i.metal-48xl	75.0	93.75	60.0
c8i.metal-96xl	100.0	125.0	80.0
c8id.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
c8id.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
c8id.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8id.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
c8id.8xlarge	15.0	18.75	12.5
c8id.12xlarge	22.5	28.125	18.75
c8id.16xlarge	30.0	37.5	25.0
c8id.24xlarge	40.0	50.0	32.5
c8id.32xlarge	50.0	62.5	40.0
c8id.48xlarge	75.0	93.75	60.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8id.96xlarge	100.0	125.0	80.0
c8id.metal-48xl	75.0	93.75	60.0
c8id.metal-96xl	100.0	125.0	80.0
c8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
c8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
c8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
c8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
c8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
c8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
c8i-flex.16xlarge	15.0/ 30.0	18.75/ 37.5	12.5 / 25.0
m8a.medium	0.52/12.5	0.65/12.5	0.438 / 10.0
m8a.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8a.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8a.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8a.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8a.8xlarge	15.0	18.75	12.5
m8a.12xlarge	22.5	28.125	18.75
m8a.16xlarge	30.0	37.5	25.0
m8a.24xlarge	40.0	50.0	32.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8a.48xlarge	75.0	93.75	60.0
m8a.metal-24xl	40.0	50.0	32.5
m8a.metal-48xl	75.0	93.75	60.0
m8g.medium	0.52/12.5	0.65/12.5	0.441 / 10.0
m8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
m8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8g.8xlarge	15.0	18.75	12.5
m8g.12xlarge	22.5	28.125	18.75
m8g.16xlarge	30.0	37.5	25.0
m8g.24xlarge	40.0	50.0	32.5
m8g.48xlarge	50.0	62.5	40.0
m8g.metal-24xl	40.0	50.0	32.5
m8g.metal-48xl	50.0	62.5	40.0
m8gd.medium	0.52/12.5	0.65/12.5	0.441 / 10.0
m8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
m8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8gd.8xlarge	15.0	18.75	12.5
m8gd.12xlarge	22.5	28.125	18.75
m8gd.16xlarge	30.0	37.5	25.0
m8gd.24xlarge	40.0	50.0	32.5
m8gd.48xlarge	50.0	62.5	40.0
m8gd.metal-24xl	40.0	50.0	32.5
m8gd.metal-48xl	50.0	62.5	40.0
m8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8i.8xlarge	15.0	18.75	12.5
m8i.12xlarge	22.5	28.125	18.75
m8i.16xlarge	30.0	37.5	25.0
m8i.24xlarge	40.0	50.0	32.5
m8i.32xlarge	50.0	62.5	40.0
m8i.48xlarge	75.0	93.75	60.0
m8i.96xlarge	100.0	125.0	80.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8i.metal-48xl	75.0	93.75	60.0
m8i.metal-96xl	100.0	125.0	80.0
m8id.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
m8id.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
m8id.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
m8id.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
m8id.8xlarge	15.0	18.75	12.5
m8id.12xlarge	22.5	28.125	18.75
m8id.16xlarge	30.0	37.5	25.0
m8id.24xlarge	40.0	50.0	32.5
m8id.32xlarge	50.0	62.5	40.0
m8id.48xlarge	75.0	93.75	60.0
m8id.96xlarge	100.0	125.0	80.0
m8id.metal-48xl	75.0	93.75	60.0
m8id.metal-96xl	100.0	125.0	80.0
m8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
m8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
m8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
m8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
m8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
m8i-flex.16xlarge	15.0/ 30.0	18.75/ 37.5	12.5 / 25.0
m9g.medium	0.55 / 15.0	0.688 / 15.0	0.455/12.0
m9g.large	1.0 / 15.0	1.25 / 15.0	0.81/12.0
m9g.xlarge	2.1 / 15.0	2.625 / 15.0	1.725/12.0
m9g.2xlarge	4.25/17.0	5.313/17.0	3.5/14.0
m9g.4xlarge	8.5/17.0	10.625/17.0	7.0/14.0
m9g.8xlarge	17.0	21.25	14.0
m9g.12xlarge	25.0	31.25	20.5
m9g.16xlarge	34.0	42.5	28.0
m9g.24xlarge	50.0	62.5	41.0
m9g.48xlarge	100.0	125.0	82.0
m9g.metal-48xl	100.0	125.0	82.0
m9gd.medium	0.55 / 15.0	0.688 / 15.0	0.455/12.0
m9gd.large	1.0 / 15.0	1.25 / 15.0	0.81/12.0
m9gd.xlarge	2.1 / 15.0	2.625 / 15.0	1.725/12.0
m9gd.2xlarge	4.25/17.0	5.313/17.0	3.5/14.0
m9gd.4xlarge	8.5/17.0	10.625/17.0	7.0/14.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m9gd.8xlarge	17.0	21.25	14.0
m9gd.12xlarge	25.0	31.25	20.5
m9gd.16xlarge	34.0	42.5	28.0
m9gd.24xlarge	50.0	62.5	41.0
m9gd.48xlarge	100.0	125.0	82.0
m9gd.metal-48xl	100.0	125.0	82.0
r8a.medium	0.52/12.5	0.65/12.5	0.438 / 10.0
r8a.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8a.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8a.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8a.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8a.8xlarge	15.0	18.75	12.5
r8a.12xlarge	22.5	28.125	18.75
r8a.16xlarge	30.0	37.5	25.0
r8a.24xlarge	40.0	50.0	32.5
r8a.48xlarge	75.0	93.75	60.0
r8a.metal-24xl	40.0	50.0	32.5
r8a.metal-48xl	75.0	93.75	60.0
r8g.medium	0.52/12.5	0.65/12.5	0.441 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
r8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8g.8xlarge	15.0	18.75	12.5
r8g.12xlarge	22.5	28.125	18.75
r8g.16xlarge	30.0	37.5	25.0
r8g.24xlarge	40.0	50.0	32.5
r8g.48xlarge	50.0	62.5	40.0
r8g.metal-24xl	40.0	50.0	32.5
r8g.metal-48xl	50.0	62.5	40.0
r8gd.medium	0.52/12.5	0.65/12.5	0.441 / 10.0
r8gd.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
r8gd.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8gd.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8gd.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8gd.8xlarge	15.0	18.75	12.5
r8gd.12xlarge	22.5	28.125	18.75
r8gd.16xlarge	30.0	37.5	25.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8gd.24xlarge	40.0	50.0	32.5
r8gd.48xlarge	50.0	62.5	40.0
r8gd.metal-24xl	40.0	50.0	32.5
r8gd.metal-48xl	50.0	62.5	40.0
r8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
r8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8i.8xlarge	15.0	18.75	12.5
r8i.12xlarge	22.5	28.125	18.75
r8i.16xlarge	30.0	37.5	25.0
r8i.24xlarge	40.0	50.0	32.5
r8i.32xlarge	50.0	62.5	40.0
r8i.48xlarge	75.0	93.75	60.0
r8i.96xlarge	100.0	125.0	80.0
r8i.metal-48xl	75.0	93.75	60.0
r8i.metal-96xl	100.0	125.0	80.0
r8id.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
r8id.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8id.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8id.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
r8id.8xlarge	15.0	18.75	12.5
r8id.12xlarge	22.5	28.125	18.75
r8id.16xlarge	30.0	37.5	25.0
r8id.24xlarge	40.0	50.0	32.5
r8id.32xlarge	50.0	62.5	40.0
r8id.48xlarge	75.0	93.75	60.0
r8id.96xlarge	100.0	125.0	80.0
r8id.metal-48xl	75.0	93.75	60.0
r8id.metal-96xl	100.0	125.0	80.0
r8i-flex.large	0.468 / 12.5	0.585 / 12.5	0.389 / 10.0
r8i-flex.xlarge	0.937 / 12.5	1.172 / 12.5	0.779 / 10.0
r8i-flex.2xlarge	1.875 / 15.0	2.344 / 15.0	1.562 / 12.5
r8i-flex.4xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
r8i-flex.8xlarge	7.5 / 15.0	9.375 / 18.75	6.25 / 12.5
r8i-flex.12xlarge	11.25 / 22.5	14.063 / 28.125	9.375 / 18.75
r8i-flex.16xlarge	15.0/ 30.0	18.75/ 37.5	12.5 / 25.0
x8g.medium	0.52/12.5	0.65/12.5	0.441 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
x8g.large	0.937 / 12.5	1.171 / 12.5	0.779 / 10.0
x8g.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
x8g.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
x8g.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
x8g.8xlarge	15.0	18.75	12.5
x8g.12xlarge	22.5	28.125	18.75
x8g.16xlarge	30.0	37.5	25.0
x8g.24xlarge	40.0	50.0	32.5
x8g.48xlarge	50.0	62.5	40.0
x8g.metal-24xl	40.0	50.0	32.5
x8g.metal-48xl	50.0	62.5	40.0
x8aedz.large	1.562 / 18.75	1.953 / 18.75	1.249 / 15.0
x8aedz.xlarge	3.125 / 18.75	3.907 / 18.75	2.5 / 15.0
x8aedz.3xlarge	9.375 / 18.75	11.719 / 18.75	7.5 / 15.0
x8aedz.6xlarge	18.75	23.438	15.0
x8aedz.12xlarge	37.5	46.875	30.0
x8aedz.24xlarge	75.0	93.75	60.0
x8aedz.metal-12xl	37.5	46.875	30.0
x8aedz.metal-24xl	75.0	93.75	60.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
x8i.large	0.937 / 12.5	1.172 / 12.5	0.774 / 10.0
x8i.xlarge	1.875 / 12.5	2.344 / 12.5	1.562 / 10.0
x8i.2xlarge	3.75 / 15.0	4.688 / 15.0	3.125 / 12.5
x8i.4xlarge	7.5 / 15.0	9.375 / 15.0	6.25 / 12.5
x8i.8xlarge	15.0	18.75	12.5
x8i.12xlarge	22.5	28.125	18.75
x8i.16xlarge	30.0	37.5	25.0
x8i.24xlarge	40.0	50.0	32.5
x8i.32xlarge	50.0	62.5	40.0
x8i.48xlarge	75.0	93.75	60.0
x8i.64xlarge	80.0	100.0	62.5
x8i.96xlarge	100.0	125.0	80.0
x8i.metal-48xl	75.0	93.75	60.0
x8i.metal-96xl	100.0	125.0	80.0

Amazon EBS 性能

下表显示了使用 `default`、`vpc-1` 和 `ebs-1` 配置可以实现的 Amazon EBS 性能（以 Gbps 为单位）。

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8a.medium	0.325 / 10.0	0.195 / 6.25	0.407 / 10.0
c8a.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
c8a.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8a.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8a.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8a.8xlarge	10.0	6.25	12.5
c8a.12xlarge	15.0	9.375	18.75
c8a.16xlarge	20.0	12.5	25.0
c8a.24xlarge	30.0	20.0	37.5
c8a.48xlarge	60.0	41.25	75.0
c8a.metal-24xl	30.0	20.0	37.5
c8a.metal-48xl	60.0	41.25	75.0
c8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
c8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
c8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8g.8xlarge	10.0	6.25	12.5
c8g.12xlarge	15.0	9.375	18.75

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8g.16xlarge	20.0	12.5	25.0
c8g.24xlarge	30.0	20.0	37.5
c8g.48xlarge	40.0	27.5	50.0
c8g.metal-24xl	30.0	20.0	37.5
c8g.metal-48xl	40.0	27.5	50.0
c8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
c8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
c8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8gd.8xlarge	10.0	6.25	12.5
c8gd.12xlarge	15.0	9.375	18.75
c8gd.16xlarge	20.0	12.5	25.0
c8gd.24xlarge	30.0	20.0	37.5
c8gd.48xlarge	40.0	27.5	50.0
c8gd.metal-24xl	30.0	20.0	37.5
c8gd.metal-48xl	40.0	27.5	50.0
c8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
c8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8i.8xlarge	10.0	6.25	12.5
c8i.12xlarge	15.0	9.375	18.75
c8i.16xlarge	20.0	12.5	25.0
c8i.24xlarge	30.0	20.0	37.5
c8i.32xlarge	40.0	27.5	50.0
c8i.48xlarge	60.0	41.25	75.0
c8i.96xlarge	80.0	55.0	100.0
c8i.metal-48xl	60.0	41.25	75.0
c8i.metal-96xl	80.0	55.0	100.0
c8id.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
c8id.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8id.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8id.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
c8id.8xlarge	10.0	6.25	12.5
c8id.12xlarge	15.0	9.375	18.75
c8id.16xlarge	20.0	12.5	25.0
c8id.24xlarge	30.0	20.0	37.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
c8id.32xlarge	40.0	27.5	50.0
c8id.48xlarge	60.0	41.25	75.0
c8id.96xlarge	80.0	55.0	100.0
c8id.metal-48xl	60.0	41.25	75.0
c8id.metal-96xl	80.0	55.0	100.0
c8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
c8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
c8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
c8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
c8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
c8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75
c8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
m8a.medium	0.325 / 10.0	0.195 / 6.25	0.407 / 10.0
m8a.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8a.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8a.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8a.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8a.8xlarge	10.0	6.25	12.5
m8a.12xlarge	15.0	9.375	18.75

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8a.16xlarge	20.0	12.5	25.0
m8a.24xlarge	30.0	20.0	37.5
m8a.48xlarge	60.0	41.25	75.0
m8a.metal-24xl	30.0	20.0	37.5
m8a.metal-48xl	60.0	41.25	75.0
m8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
m8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
m8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8g.8xlarge	10.0	6.25	12.5
m8g.12xlarge	15.0	9.375	18.75
m8g.16xlarge	20.0	12.5	25.0
m8g.24xlarge	30.0	20.0	37.5
m8g.48xlarge	40.0	27.5	50.0
m8g.metal-24xl	30.0	20.0	37.5
m8g.metal-48xl	40.0	27.5	50.0
m8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
m8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8gd.8xlarge	10.0	6.25	12.5
m8gd.12xlarge	15.0	9.375	18.75
m8gd.16xlarge	20.0	12.5	25.0
m8gd.24xlarge	30.0	20.0	37.5
m8gd.48xlarge	40.0	27.5	50.0
m8gd.metal-24xl	30.0	20.0	37.5
m8gd.metal-48xl	40.0	27.5	50.0
m8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8i.8xlarge	10.0	6.25	12.5
m8i.12xlarge	15.0	9.375	18.75
m8i.16xlarge	20.0	12.5	25.0
m8i.24xlarge	30.0	20.0	37.5
m8i.32xlarge	40.0	27.5	50.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8i.48xlarge	60.0	41.25	75.0
m8i.96xlarge	80.0	55.0	100.0
m8i.metal-48xl	60.0	41.25	75.0
m8i.metal-96xl	80.0	55.0	100.0
m8id.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
m8id.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8id.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8id.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
m8id.8xlarge	10.0	6.25	12.5
m8id.12xlarge	15.0	9.375	18.75
m8id.16xlarge	20.0	12.5	25.0
m8id.24xlarge	30.0	20.0	37.5
m8id.32xlarge	40.0	27.5	50.0
m8id.48xlarge	60.0	41.25	75.0
m8id.96xlarge	80.0	55.0	100.0
m8id.metal-48xl	60.0	41.25	75.0
m8id.metal-96xl	80.0	55.0	100.0
m8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
m8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
m8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
m8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
m8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75
m8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
m9g.medium	0.38/12.0	0.242/7.75	0.475/12.0
m9g.large	0.76/12.0	0.51/7.75	0.95/12.0
m9g.xlarge	1.5/12.0	0.975/7.75	1.875 / 12.0
m9g.2xlarge	3.0/12.0	1.937/7.75	3.75 / 12.0
m9g.4xlarge	6.0/12.0	3.875/7.75	7.5 / 12.0
m9g.8xlarge	12.0	7.75	15.0
m9g.12xlarge	18.0	11.75	22.5
m9g.16xlarge	24.0	15.5	30.0
m9g.24xlarge	36.0	23.5	45.0
m9g.48xlarge	72.0	47.0	90.0
m9g.metal-48xl	72.0	47.0	90.0
m9gd.medium	0.38/12.0	0.242/7.75	0.475/12.0
m9gd.large	0.76/12.0	0.51/7.75	0.95/12.0
m9gd.xlarge	1.5/12.0	0.975/7.75	1.875 / 12.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
m9gd.2xlarge	3.0/12.0	1.937/7.75	3.75 / 12.0
m9gd.4xlarge	6.0/12.0	3.875/7.75	7.5 / 12.0
m9gd.8xlarge	12.0	7.75	15.0
m9gd.12xlarge	18.0	11.75	22.5
m9gd.16xlarge	24.0	15.5	30.0
m9gd.24xlarge	36.0	23.5	45.0
m9gd.48xlarge	72.0	47.0	90.0
m9gd.metal-48xl	72.0	47.0	90.0
r8a.medium	0.325 / 10.0	0.195 / 6.25	0.407 / 10.0
r8a.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8a.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8a.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8a.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8a.8xlarge	10.0	6.25	12.5
r8a.12xlarge	15.0	9.375	18.75
r8a.16xlarge	20.0	12.5	25.0
r8a.24xlarge	30.0	20.0	37.5
r8a.48xlarge	60.0	41.25	75.0
r8a.metal-24xl	30.0	20.0	37.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8a.metal-48xl	60.0	41.25	75.0
r8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
r8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
r8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8g.8xlarge	10.0	6.25	12.5
r8g.12xlarge	15.0	9.375	18.75
r8g.16xlarge	20.0	12.5	25.0
r8g.24xlarge	30.0	20.0	37.5
r8g.48xlarge	40.0	27.5	50.0
r8g.metal-24xl	30.0	20.0	37.5
r8g.metal-48xl	40.0	27.5	50.0
r8gd.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
r8gd.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
r8gd.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8gd.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8gd.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8gd.8xlarge	10.0	6.25	12.5

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8gd.12xlarge	15.0	9.375	18.75
r8gd.16xlarge	20.0	12.5	25.0
r8gd.24xlarge	30.0	20.0	37.5
r8gd.48xlarge	40.0	27.5	50.0
r8gd.metal-24xl	30.0	20.0	37.5
r8gd.metal-48xl	40.0	27.5	50.0
r8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8i.8xlarge	10.0	6.25	12.5
r8i.12xlarge	15.0	9.375	18.75
r8i.16xlarge	20.0	12.5	25.0
r8i.24xlarge	30.0	20.0	37.5
r8i.32xlarge	40.0	27.5	50.0
r8i.48xlarge	60.0	41.25	75.0
r8i.96xlarge	80.0	55.0	100.0
r8i.metal-48xl	60.0	41.25	75.0
r8i.metal-96xl	80.0	55.0	100.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8id.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
r8id.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8id.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8id.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
r8id.8xlarge	10.0	6.25	12.5
r8id.12xlarge	15.0	9.375	18.75
r8id.16xlarge	20.0	12.5	25.0
r8id.24xlarge	30.0	20.0	37.5
r8id.32xlarge	40.0	27.5	50.0
r8id.48xlarge	60.0	41.25	75.0
r8id.96xlarge	80.0	55.0	100.0
r8id.metal-48xl	60.0	41.25	75.0
r8id.metal-96xl	80.0	55.0	100.0
r8i-flex.large	0.315 / 10.0	0.198 / 6.25	0.394 / 10.0
r8i-flex.xlarge	0.63 / 10.0	0.395 / 6.25	0.788 / 10.0
r8i-flex.2xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
r8i-flex.4xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
r8i-flex.8xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 12.5
r8i-flex.12xlarge	7.5 / 15.0	4.687 / 9.375	9.375 / 18.75

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
r8i-flex.16xlarge	10.0 / 20.0	6.25 / 12.5	12.5 / 25.0
x8g.medium	0.315 / 10.0	0.185 / 6.25	0.394 / 10.0
x8g.large	0.63 / 10.0	0.396 / 6.25	0.788 / 10.0
x8g.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
x8g.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
x8g.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
x8g.8xlarge	10.0	6.25	12.5
x8g.12xlarge	15.0	9.375	18.75
x8g.16xlarge	20.0	12.5	25.0
x8g.24xlarge	30.0	20.0	37.5
x8g.48xlarge	40.0	27.5	50.0
x8g.metal-24xl	30.0	20.0	37.5
x8g.metal-48xl	40.0	27.5	50.0
x8aedz.large	1.25 / 15.0	0.859/10.312	1.563 / 15.0
x8aedz.xlarge	2.5 / 15.0	1.718/10.312	3.125 / 15.0
x8aedz.3xlarge	7.5 / 15.0	5.156/10.312	9.375 / 15.0
x8aedz.6xlarge	15.0	10.312	18.75
x8aedz.12xlarge	30.0	20.625	37.5
x8aedz.24xlarge	60.0	41.25	75.0

实例类型	**default** （基准/突增）	**vpc-1** （基准/突增）	**ebs-1** （基准/突增）
x8aedz.metal-12xl	30.0	20.625	37.5
x8aedz.metal-24xl	60.0	41.25	75.0
x8i.large	0.65 / 10.0	0.415 / 6.25	0.813 / 10.0
x8i.xlarge	1.25 / 10.0	0.781 / 6.25	1.563 / 10.0
x8i.2xlarge	2.5 / 10.0	1.562 / 6.25	3.125 / 10.0
x8i.4xlarge	5.0 / 10.0	3.125 / 6.25	6.25 / 10.0
x8i.8xlarge	10.0	6.25	12.5
x8i.12xlarge	15.0	9.375	18.75
x8i.16xlarge	20.0	12.5	25.0
x8i.24xlarge	30.0	20.0	37.5
x8i.32xlarge	40.0	27.5	50.0
x8i.48xlarge	60.0	41.25	75.0
x8i.64xlarge	70.0	50.0	87.5
x8i.96xlarge	80.0	55.0	100.0
x8i.metal-48xl	60.0	41.25	75.0
x8i.metal-96xl	80.0	55.0	100.0

监控实例带宽

您可以使用 CloudWatch 指标监控实例网络带宽以及发送和接收的数据包。您可以使用弹性网络适配器（ENA）驱动程序提供的网络性能指标来监控流量何时超过 Amazon EC2 在实例级别定义的网络限额。

您可以配置 Amazon EC2 是使用 1 分钟还是 5 分钟时段将实例的指标数据发送到 CloudWatch。网络性能指标可能会显示超出限额并丢弃数据包，而 CloudWatch 实例指标不会显示。当实例对网络资源的需求出现短暂峰值（称为微爆发），但 CloudWatch 指标的精细程度不足以反映这些微秒峰值时，就会发生这种情况。

了解详情

- 实例指标
- 监控网络性能

Amazon EC2 实例上的增强联网功能

增强联网使用单个根 I/O 虚拟化 (SR-IOV) 在支持的实例类型上提供高性能的联网功能。SR-IOV 是一种设备虚拟化方法，与传统虚拟化网络接口相比，它不仅能提高 I/O 性能，还能降低 CPU 使用率。增强联网可以提高带宽，提高每秒数据包数（PPS）性能，并不断降低实例间的延迟。使用增强联网不收取任何额外费用。

有关每个实例类型支持的网络速度的信息，请参阅 Amazon EC2 实例类型。

您可以使用以下某种机制启用增强联网：

弹性网络适配器（ENA）

对于支持的实例类型，弹性网络适配器（ENA）最多支持 100 Gbps 的网络速度。

所有基于 Nitro 的实例都使用 ENA 来增强联网性能。此外，以下基于 Xen 的实例使用 ENA：H1、I3、G3、`m4.16xlarge`、P3、P3dn 和 R4。

有关更多信息，请参阅 在 EC2 实例上使用 ENA 启用增强联网功能。

Intel 82599 虚拟功能 (VF) 接口

对于受支持的实例类型，Intel 82599 虚拟功能接口最多支持 10 Gbps 的网络速度。

以下实例类型使用 Intel 82599 VF 接口来实现增强联网：C3、C4、D2、I2、M4（不包括 m4.16xlarge）和 R3。

有关更多信息，请参阅 通过 Intel 82599 VF 接口实现增强联网。

内容

- 在 EC2 实例上使用 ENA 启用增强联网功能
- 使用 ENA Express 提高 EC2 实例之间的网络性能
- 通过 Intel 82599 VF 接口实现增强联网
- 监控 EC2 实例上 ENA 设置的网络性能
- 改善基于 Linux 的 EC2 实例的网络延迟
- Nitro 系统性能调整注意事项
- 优化 EC2 Windows 实例上的网络性能

在 EC2 实例上使用 ENA 启用增强联网功能

Amazon EC2 通过弹性网络适配器（ENA）提供增强联网功能。要启用增强联网功能，必须使用包含所需 ENA 驱动程序的 AMI，或者手动安装该驱动程序。然后，您可以在实例上启用 ENA 支持。

要查看 ENA 驱动程序的发行说明或安装说明，请参阅与实例操作系统平台匹配的选项卡。

Linux

您可以在 GitHub 上查看以下文档：

- 在 GitHub 上查看 ENA Linux kernel driver release notes。
- 有关包含安装说明的 ENA Linux 内核驱动程序的概述，请参阅 GitHub 上的 Linux kernel driver for Elastic Network Adapter (ENA) family。

Windows

您可以在本指南的管理设备驱动程序部分查看以下文档：

- 跟踪 ENA Windows 驱动程序发行版.
- 在 EC2 Windows 实例上安装 ENA 驱动程序.

对于基于 Nitro 的实例，增强联网功能因实例类型实现的 Nitro 版本而异。

要查看实例的网络规范，请选择适合实例类型的实例系列链接。如果不确定适用哪个实例系列，请参阅《Amazon EC2 Instance Types Guide》中的 Naming conventions。

- 加速型计算实例的网络规格
- 计算优化型实例的网络规格
- 通用实例的网络规格
- 高性能计算型实例的网络规格
- 内存优化型实例的网络规格
- 存储优化型实例的网络规格

内容

- 使用 ENA 启用增强联网功能的先决条件
- 测试是否启用了增强联网功能
- 在实例上启用增强联网
- ENA 队列数
- 排查 Linux 上 ENA 内核驱动程序的问题
- 对弹性网络适配器 Windows 驱动程序进行问题排查

使用 ENA 启用增强联网功能的先决条件

要使用 ENA 准备增强联网，请按如下方式设置您的实例：

- 启动基于 Nitro 的实例。
- 确保实例具有 Internet 连接。
- 如果您的实例上有重要的数据需要保留，则应立即从您的实例创建 AMI，来备份这些数据。更新 ENA 内核驱动程序以及启用 `enaSupport` 属性可能会导致实例不兼容或操作系统无法访问。如果您有最新备份，则发生此情况时仍将保留数据。
- Linux 实例：使用支持的 Linux 内核版本及支持的发行版启动实例，以便为实例自动启用 ENA 增强联网。有关更多信息，请参阅 ENA Linux 内核驱动程序发行说明。
- Windows 实例：如果实例运行的是 Windows Server 2008 R2 SP1，则请确保该实例具有 SHA-2 代码签名支持更新。

- 使用 Amazon Web Services 管理控制台的 Amazon CloudShell，或者将 Amazon CLI 或 Amazon Tools for Windows PowerShell 安装到您选择的任意电脑上（最好是本地台式机或笔记本电脑）并进行配置。有关的更多信息，请参阅 访问 Amazon EC2 或 Amazon CloudShell 用户指南。不能从 Amazon EC2 控制台管理增强联网。

测试是否启用了增强联网功能

您可以测试实例或 AMI 中是否启用了增强联网。

实例属性

检查 enaSupport 实例属性的值。

Amazon CLI

使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[].Instances[].EnaSupport"
```

如果启用了增强联网功能，则输出会如下所示。

```
[
    true
]
```

PowerShell

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

如果启用了增强联网功能，则输出会如下所示。

```
True
```

映像属性

检查 enaSupport 映像属性的值。

Amazon CLI

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
    --query "Images[].EnaSupport"
```

如果启用了增强联网功能，则输出会如下所示。

```
[
    true
]
```

PowerShell

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).EnaSupport
```

如果启用了增强联网功能，则输出会如下所示。

```
True
```

Linux 网络接口驱动程序

使用以下命令验证是否在特定接口上使用了 ena 内核驱动程序（替代要检查的接口名称）。如果您使用单个接口（默认设置），则为 eth0。如果您的 Linux 发行版支持可预测的网络名称，则这可以是类似于 ens5 的名称。有关更多信息，请展开在实例上启用增强联网中的 RHEL、SUSE 和 CentOS 部分。

在以下示例中，ena 内核驱动程序未加载，因为列出的驱动程序是 vif。

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
```

```
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

在本示例中，ena 内核驱动程序已加载并具有推荐的最低版本。此实例正确配置了增强联网。

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 1.5.0g
firmware-version:
expansion-rom-version:
bus-info: 0000:00:05.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

在实例上启用增强联网

您使用的过程取决于实例的操作系统。

Amazon Linux

适用于 Amazon Linux 的 AMI 包含安装了 ENA 的增强联网所需的内核驱动程序，并启用了 ENA 支持。因此，如果使用 HVM 版本的 Amazon Linux 在支持的实例类型上启动实例，则已为您的实例启用增强联网。有关更多信息，请参阅 测试是否启用了增强联网功能。

Ubuntu

最新的 Ubuntu HVM AMI 包含安装了 ENA 的增强联网所需的内核驱动程序，并启用了 ENA 支持。因此，如果使用最新的 Ubuntu HVM AMI 在支持的实例类型上启动实例，则已为您的实例启用增强联网。有关更多信息，请参阅测试是否启用了增强联网功能。

如果您使用较旧的 AMI 启动了实例，并且实例尚未启用增强联网，则可以安装 `linux-aws` 内核程序包以获取最新的增强联网驱动程序并更新所需的属性。

安装 **`linux-aws`** 内核程序包（Ubuntu 16.04 或更高版本）

Ubuntu 16.04 和 18.04 随附 Ubuntu 自定义内核（`linux-aws` 内核程序包）。要使用其他内核，请联系 Amazon Web Services 支持。

安装 **linux-aws** 内核程序包（Ubuntu Trusty 14.04）

1. 连接到您的 实例。
2. 更新包缓存和包。

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

> **Important**
>
> 如果在更新过程中系统提示您安装 grub，请使用 /dev/xvda 安装 grub，然后选择保留当前版本的 /boot/grub/menu.lst。

3. [由 EBS 支持的实例] 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机停止实例：stop-instances（Amazon CLI）或 Stop-EC2Instance（Amazon Tools for Windows PowerShell）。

   [实例存储支持的实例] 您无法停止实例来修改属性。请执行此过程：在 Ubuntu 上启用增强联网（由实例存储支持的实例）。
4. 使用以下任一命令从本地电脑启用增强联网属性：

   - modify-instance-attribute (Amazon CLI)

     ```
     aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
     ```

   - Edit-EC2InstanceAttribute（适用于 Windows PowerShell 的工具）

     ```
     Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
     ```
5. (可选) 从实例创建 AMI，如 中所述创建 Amazon EBS-backed AMI 该 AMI 从实例继承增强联网 enaSupport 属性。因此，您可以使用此 AMI 启动默认情况下启用了增强联网功能的其他实例。
6. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机启动实例：start-instances（Amazon CLI）或 Start-EC2Instance（Amazon Tools for Windows PowerShell）。

在 Ubuntu 上启用增强联网（由实例存储支持的实例）

按照上述过程操作，直到您停止实例的步骤。按照创建 Amazon S3 支持的 AMI中所述创建新 AMI，确保在注册 AMI 时启用增强联网属性。

- register-image (Amazon CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (Amazon Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport $true ...
```

RHEL、SUSE、CentOS

Red Hat Enterprise Linux、SUSE Linux Enterprise Server 和 CentOS 的最新 AMI 包含使用了 ENA 的增强联网所需的内核驱动程序，并启用了 ENA 支持。因此，如果使用最新的 AMI 在支持的实例类型上启动实例，则已为您的实例启用增强联网。有关更多信息，请参阅测试是否启用了增强联网功能。

下面的过程提供了在 Amazon Linux AMI 或 Ubuntu 之外的 Linux 发行版上启用增强联网的一般步骤。有关更多信息（如命令的详细语法、文件位置或包和工具支持），请参阅您的 Linux 发行版的文档。

对 Linux 启用增强联网

1. 连接到您的实例。
2. 从 GitHub 的以下网址克隆实例上的 ena 内核驱动程序源代码：https://github.com/amzn/amzn-drivers。（默认情况下，SUSE Linux Enterprise Server 12 SP2 及更高版本包含 ENA 2.02，因此不需要下载和编译 ENA 驱动程序。对于 SUSE Linux Enterprise Server 12 SP2 及更高版本，您应该提交一个请求，将您想要的驱动程序版本添加到存储内核中）。
3. 在实例上编译并安装 ena 内核驱动程序。这些步骤取决于 Linux 发行版。有关在 Red Hat Enterprise Linux 上编译内核驱动程序的更多信息，请参阅如何在运行 RHEL 的 Amazon EC2 实例上安装并激活最新的 ENA 驱动程序，以便获得增强型网络支持？
4. 运行 sudo depmod 命令来更新内核驱动程序依赖项。
5. 在实例上更新 initramfs，确保在启动时加载新内核驱动程序。例如，如果您的发行版支持 dracut，则可使用以下命令。

   ```
   dracut -f -v
   ```

6. 确定您的系统是否默认使用可预测的网络接口名称。使用 systemd 或 udev 版本 197 或更高版本的系统可以重命名以太网设备，它们不保证单个网络接口将命名为 eth0。此行为可能导致连接到实例时出现问题。要获取更多信息并查看其他配置选项，请参阅 freedesktop.org 网站上的可预测的网络接口名称。

a. 您可以使用以下命令在基于 RPM 的系统上检查 systemd 或 udev 版本。

```
rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
systemd-208-11.el7_0.2.x86_64
```

在以上 Red Hat Enterprise Linux 7 示例中，systemd 版本是 208，因此必须禁用可预测的网络接口名称。

b. 通过将 `net.ifnames=0` 选项添加到 `GRUB_CMDLINE_LINUX` 中的 `/etc/default/grub` 行，可禁用可预测的网络接口名称。

```
sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /etc/default/
grub
```

c. 重新构建 grub 配置文件。

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [由 EBS 支持的实例] 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机停止实例：stop-instances（Amazon CLI）、Stop-EC2Instance（Amazon Tools for Windows PowerShell）。

   [实例存储支持的实例] 您无法停止实例来修改属性。请执行此过程：在 Linux 上启用增强联网（由实例存储支持的实例）。

8. 使用以下任一命令从本地计算机启用增强联网 `enaSupport` 属性：

   - modify-instance-attribute (Amazon CLI)

     ```
     aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
     ```

   - Edit-EC2InstanceAttribute（适用于 Windows PowerShell 的工具）

     ```
     Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
     ```

9. (可选) 从实例创建 AMI，如 中所述创建 Amazon EBS-backed AMI 该 AMI 从实例继承增强联网 `enaSupport` 属性。因此，您可以使用此 AMI 启动默认情况下启用了增强联网功能的其他实例。

   如果您的实例操作系统包含 `/etc/udev/rules.d/70-persistent-net.rules` 文件，则必须在创建 AMI 之前将其删除。此文件包含原始实例的以太网适配器 MAC 地址。如果其他实例使

用此文件启动，操作系统将找不到设备，eth0 会失败，从而导致启动问题。此文件将在下次启动过程中重新生成，从 AMI 启动的任意实例都会创建这个文件的自有版本。

10. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机启动实例：start-instances（Amazon CLI）或 Start-EC2Instance（Amazon Tools for Windows PowerShell）。
11. （可选）连接到实例并验证是否安装了内核驱动程序。

    如果您在启用增强联网之后无法连接到实例，请参阅排查 Linux 上 ENA 内核驱动程序的问题。

在 Linux 上启用增强联网（由实例存储支持的实例）

按照上述过程操作，直到您停止实例的步骤。按照创建 Amazon S3 支持的 AMI中所述创建新 AMI，确保在注册 AMI 时启用增强联网属性。

- register-image (Amazon CLI)

```
aws ec2 register-image --ena-support ...
```

- Register-EC2Image (Amazon Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport ...
```

带有 DKMS 的 Ubuntu

此方法仅用于测试和反馈目的。它不供生产部署使用。有关生产部署，请参阅Ubuntu。

> **Important**
> 使用 DKMS 可避免您的订阅的支持协议。不得将其用于生产部署。

在 Ubuntu 上启用 ENA 增强联网 (EBS 支持的实例)

1. 按照Ubuntu中的步骤 1 和 2 操作。
2. 安装 build-essential 程序包来编译内核驱动程序和 dkms 程序包，这样每次更新内核时都会重建 ena 内核驱动程序。

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

3. 从 GitHub 的以下网址克隆实例上的 ena 内核驱动程序源：https://github.com/amzn/amzn-drivers。

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

4. 将 amzn-drivers 包移动到 /usr/src/ 目录，以便 DKMS 可以在每次内核更新中找到并构建该模块。将源代码的版本号 (您可在发行说明中找到当前版本号) 附加到目录名称。例如，版本 1.0.0 显示在以下示例中。

```
ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
```

5. 使用以下值创建 DKMS 配置文件（替代您的 ena 版本）。

   创建 文件。

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

   编辑文件并添加以下值。

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
BUILT_MODULE_NAME[0]="ena"
BUILT_MODULE_LOCATION="kernel/linux/ena"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ena"
AUTOINSTALL="yes"
```

6. 使用 DKMS 在实例上添加、生成和安装 ena 内核驱动程序。

   将内核驱动程序添加到 DKMS。

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

   使用 dkms 命令生成内核驱动程序。

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

使用 dkms 安装内核驱动程序。

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

7. 重新生成 `initramfs`，以便在启动时加载正确的内核驱动程序。

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. 使用测试是否启用了增强联网功能中的 modinfo ena 命令验证是否安装了 ena 内核驱动器。

```
ubuntu:~$ modinfo ena
filename:    /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko
version:  1.0.0
license:  GPL
description: Elastic Network Adapter (ENA)
author:   Amazon.com, Inc. or its affiliates
srcversion:  9693C876C54CA64AE48F0CA
alias:    pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:    pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
vermagic:    3.13.0-74-generic SMP mod_unload modversions
parm:     debug:Debug level (0=none,...,16=all) (int)
parm:     push_mode:Descriptor / header push mode (0=automatic,1=disable,3=enable)
    0 - Automatically choose according to device capability (default)
    1 - Don't push anything to device memory
    3 - Push descriptors and header buffer to device memory (int)
parm:     enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1) (int)
parm:     enable_missing_tx_detection:Enable missing Tx completions. (default=1)
 (int)
parm:     numa_node_override_array:Numa node override map
(array of int)
parm:     numa_node_override:Enable/Disable numa node override (0=disable)
(int)
```

9. 继续执行Ubuntu中的步骤 3。

在 Windows 上启用增强联网

如果您启动了您的实例且该实例未启用增强联网，则必须下载所需的网络适配器驱动程序并将其安装到您的实例上，然后设置 enaSupport 实例属性以激活增强联网。

启用增强联网

1. 连接到您的实例并以本地管理员身份登录。
2. [仅限 Windows Server 2016 和 2019] 运行以下 EC2Launch PowerShell 脚本，以便在安装此驱动程序后配置实例。

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

3. 从实例安装驱动程序，如下所示：
   a. 将最新驱动程序下载到实例。
   b. 提取 zip 存档。
   c. 通过运行 install.ps1 PowerShell 脚本安装驱动程序。

> **Note**
>
> 如果您收到了执行策略错误，请将策略设置为 Unrestricted（默认情况下设置为 Restricted 或 RemoteSigned）。在命令行中，运行 Set-ExecutionPolicy -ExecutionPolicy Unrestricted，然后再次运行 install.ps1 PowerShell 脚本。

4. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机停止实例：stop-instances（Amazon CLI）或 Stop-EC2Instance）（Amazon Tools for Windows PowerShell）。
5. 在您的实例上启用 ENA 支持，如下所示：
   a. 在您的本地计算机上，通过运行以下命令之一来检查实例的 EC2 实例 ENA 支持属性。如果未启用属性，则输出将为“[]”或为空。默认情况下，EnaSupport 设置为 false。
      - describe-instances (Amazon CLI)

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
 "Reservations[].Instances[].EnaSupport"
```

      - Get-EC2Instance（适用于 Windows PowerShell 的工具）

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.EnaSupport
```

b. 要启用 ENA 支持，请运行以下命令之一：

- modify-instance-attribute (Amazon CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
```

- Edit-EC2InstanceAttribute (Amazon Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $true
```

如果在重新启动实例时遇到问题，也可以使用以下命令之一禁用 ENA 支持：

- modify-instance-attribute (Amazon CLI)

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --no-ena-support
```

- Edit-EC2InstanceAttribute (Amazon Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -EnaSupport $false
```

c. 如前所示，使用 `true` 或 describe-instances，验证属性是否已设置为 Get-EC2Instance。现在，您应看到以下输出：

```
[
 true
]
```

6. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机启动实例：start-instances（Amazon CLI）或 Start-EC2Instance（Amazon Tools for Windows PowerShell）。
7. 在实例上，验证 ENA 驱动程序是否已如下所示安装并且启用：

a. 右键单击网络图标，然后选择 Open Network and Sharing Center。

b. 选择以太网适配器 (例如 Ethernet 2)。

c. 选择 Details。对于网络连接详细信息，检查描述是否为 Amazon 弹性网络适配器。

8. (可选) 从实例创建 AMI。该 AMI 继承实例的 `enaSupport` 属性。因此，您可以使用该 AMI 来启动另一个默认启用 ENA 的实例。

ENA 队列数

ENA 队列分配给网络接口，其默认静态限制取决于实例类型和大小。在受支持的实例类型上，您可以跨弹性网络接口（ENI）动态分配这些队列。虽然每个实例的队列总数取决于其类型和大小，但可以使用 ENA 队列配置多个 ENI，直到满足 ENI 和实例的最大队列数为止。

灵活分配 ENA 队列有助于优化资源分配，从而尽可能提高 vCPU 利用率。需要高网络性能的工作负载通常需要使用多个 ENA 队列。您可以根据特定的工作负载调整微调队列数，从而微调网络性能和每秒数据包数（PPS）。例如，网络密集型应用程序需要使用的队列数可能会多于 CPU 密集型应用程序。

主题

- 支持的实例
- 修改队列数

支持的实例

以下实例支持动态分配多个 ENA 队列。

通用型

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
	M6i		
m6i.large	2	2	6
m6i.xlarge	4	4	16
m6i.2xlarge	8	8	32
m6i.4xlarge	8	16	64
m6i.8xlarge	8	32	64
m6i.12xlarge	8	32	64

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m6i.16xlarge	8	32	120
m6i.24xlarge	8	32	120
m6i.32xlarge	8	32	120
M6id			
m6id.large	2	2	6
m6id.xlarge	4	4	16
m6id.2xlarge	8	8	32
m6id.4xlarge	8	16	64
m6id.8xlarge	8	32	64
m6id.12xlarge	8	32	64
m6id.16xlarge	8	32	120
m6id.24xlarge	8	32	120
m6id.32xlarge	8	32	120
M6idn			
m6idn.large	2	2	6
m6idn.xlarge	4	4	16
m6idn.2xlarge	8	8	32
m6idn.4xlarge	8	16	64
m6idn.8xlarge	16	32	128
m6idn.12xlarge	16	32	128

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m6idn.16xlarge	16	32	240
m6idn.24xlarge	32	32	480
m6idn.32xlarge	32	32	512
M6in			
m6in.large	2	2	6
m6in.xlarge	4	4	16
m6in.2xlarge	8	8	32
m6in.4xlarge	8	16	64
m6in.8xlarge	16	32	128
m6in.12xlarge	16	32	128
m6in.16xlarge	16	32	240
m6in.24xlarge	32	32	480
m6in.32xlarge	32	32	512
M8a			
m8a.medium	1	1	3
m8a.large	2	2	6
m8a.xlarge	4	4	16
m8a.2xlarge	8	8	32
m8a.4xlarge	8	16	64
m8a.8xlarge	8	32	128

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8a.12xlarge	16	64	192
m8a.16xlarge	16	64	256
m8a.24xlarge	16	128	384
m8a.48xlarge	32	128	768
m8a.metal-24xl	16	128	384
m8a.metal-48xl	32	128	768
M8azn			
m8azn.medium	1	1	3
m8azn.large	2	2	8
m8azn.xlarge	4	4	16
m8azn.3xlarge	4	16	48
m8azn.6xlarge	8	32	96
m8azn.12xlarge	8	64	192
m8azn.24xlarge	16	128	384
m8azn.metal-12xl	8	64	192
m8azn.metal-24xl	16	128	384
M8gb			
m8gb.medium	1	1	2
m8gb.large	2	2	6
m8gb.xlarge	4	4	16

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8gb.2xlarge	8	8	32
m8gb.4xlarge	8	16	64
m8gb.8xlarge	8	32	128
m8gb.12xlarge	16	64	192
m8gb.16xlarge	16	64	256
m8gb.24xlarge	16	128	384
m8gb.48xlarge	32	128	768
m8gb.metal-24xl	32	128	768
m8gb.metal-48xl	32	128	768
M8gn			
m8gn.medium	1	1	2
m8gn.large	2	2	6
m8gn.xlarge	4	4	16
m8gn.2xlarge	8	8	32
m8gn.4xlarge	8	16	64
m8gn.8xlarge	8	32	128
m8gn.12xlarge	16	64	192
m8gn.16xlarge	16	64	256
m8gn.24xlarge	16	128	384
m8gn.48xlarge	32	128	768

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8gn.metal-24xl	32	128	768
m8gn.metal-48xl	32	128	768
M8i			
m8i.large	2	2	6
m8i.xlarge	4	4	16
m8i.2xlarge	8	8	32
m8i.4xlarge	8	16	64
m8i.8xlarge	8	32	128
m8i.12xlarge	16	64	192
m8i.16xlarge	16	64	256
m8i.24xlarge	16	128	384
m8i.32xlarge	16	128	512
m8i.48xlarge	32	128	768
m8i.96xlarge	32	128	1536
m8i.metal-48xl	32	128	768
m8i.metal-96xl	32	128	1536
M8id			
m8id.large	2	2	6
m8id.xlarge	4	4	16
m8id.2xlarge	8	8	32

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8id.4xlarge	8	16	64
m8id.8xlarge	8	32	128
m8id.12xlarge	16	64	192
m8id.16xlarge	16	64	256
m8id.24xlarge	16	128	384
m8id.32xlarge	16	128	512
m8id.48xlarge	32	128	768
m8id.96xlarge	32	128	1536
m8id.metal-48xl	32	128	768
m8id.metal-96xl	32	128	1536
M8i-flex			
m8i-flex.large	1	1	3
m8i-flex.xlarge	2	2	8
m8i-flex.2xlarge	4	4	16
m8i-flex.4xlarge	4	8	32
m8i-flex.8xlarge	4	16	64
m8i-flex.12xlarge	8	32	96
m8i-flex.16xlarge	8	32	128
M8in			
m8in.large	2	2	8

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8in.xlarge	4	4	16
m8in.2xlarge	8	8	32
m8in.4xlarge	8	16	64
m8in.8xlarge	16	32	128
m8in.12xlarge	16	64	192
m8in.16xlarge	16	64	256
m8in.24xlarge	16	128	256
m8in.32xlarge	32	128	512
m8in.48xlarge	32	128	768
m8in.96xlarge	32	128	1536 *
m8in.metal-48xl	32	128	768
m8in.metal-96xl	32	128	1536 *
	M8idn		
m8idn.large	2	2	8
m8idn.xlarge	4	4	16
m8idn.2xlarge	8	8	32
m8idn.4xlarge	8	16	64
m8idn.8xlarge	16	32	128
m8idn.12xlarge	16	64	192
m8idn.16xlarge	16	64	256

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8idn.24xlarge	16	128	256
m8idn.32xlarge	32	128	512
m8idn.48xlarge	32	128	768
m8idn.96xlarge	32	128	1536 *
m8idn.metal-48xl	32	128	768
m8idn.metal-96xl	32	128	1536 *
M8ine			
m8ine.large	2	2	8
m8ine.xlarge	4	4	16
m8ine.2xlarge	8	8	32
m8ine.4xlarge	16	16	128
m8ine.8xlarge	32	32	256
m8ine.12xlarge	32	64	384
m8iB			
m8ib.large	2	2	8
m8ib.xlarge	4	4	16
m8ib.2xlarge	8	8	32
m8ib.4xlarge	8	16	64
m8ib.8xlarge	16	32	128
m8ib.12xlarge	16	64	192

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8ib.16xlarge	16	64	256
m8ib.24xlarge	16	128	256
m8ib.32xlarge	32	128	512
m8ib.48xlarge	32	128	768
m8ib.96xlarge	32	128	1536 *
m8ib.metal-48xl	32	128	768
m8ib.metal-96xl	32	128	1536 *
M8idb			
m8idb.large	2	2	8
m8idb.xlarge	4	4	16
m8idb.2xlarge	8	8	32
m8idb.4xlarge	8	16	64
m8idb.8xlarge	16	32	128
m8idb.12xlarge	16	64	192
m8idb.16xlarge	16	64	256
m8idb.24xlarge	16	128	256
m8idb.32xlarge	32	128	512
m8idb.48xlarge	32	128	768
m8idb.96xlarge	32	128	1536 *
m8idb.metal-48xl	32	128	768

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m8idb.metal-96xl	32	128	1536 *
M9g			
m9g.medium	1	1	2
m9g.large	2	2	6
m9g.xlarge	4	4	16
m9g.2xlarge	8	8	32
m9g.4xlarge	8	16	64
m9g.8xlarge	8	32	128
m9g.12xlarge	16	64	192
m9g.16xlarge	16	64	256
m9g.24xlarge	16	128	384
m9g.48xlarge	32	128	768
m9g.metal-48xl	32	128	768
M9gd			
m9gd.medium	1	1	2
m9gd.large	2	2	6
m9gd.xlarge	4	4	16
m9gd.2xlarge	8	8	32
m9gd.4xlarge	8	16	64
m9gd.8xlarge	8	32	128

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
m9gd.12xlarge	16	64	192
m9gd.16xlarge	16	64	256
m9gd.24xlarge	16	128	384
m9gd.48xlarge	32	128	768
m9gd.metal-48xl	32	128	768

> **Note**
>
> * 这些实例类型配备多个网卡。其他实例类型配备单个网卡。有关更多信息，请参阅 网卡。

计算优化型

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
	C6i		
c6i.large	2	2	6
c6i.xlarge	4	4	16
c6i.2xlarge	8	8	32
c6i.4xlarge	8	16	64
c6i.8xlarge	8	32	64
c6i.12xlarge	8	32	64
c6i.16xlarge	8	32	120

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c6i.24xlarge	8	32	120
c6i.32xlarge	8	32	120
C6id			
c6id.large	2	2	6
c6id.xlarge	4	4	16
c6id.2xlarge	8	8	32
c6id.4xlarge	8	16	64
c6id.8xlarge	8	32	64
c6id.12xlarge	8	32	64
c6id.16xlarge	8	32	120
c6id.24xlarge	8	32	120
c6id.32xlarge	8	32	120
C6in			
c6in.large	2	2	6
c6in.xlarge	4	4	16
c6in.2xlarge	8	8	32
c6in.4xlarge	8	16	64
c6in.8xlarge	16	32	128
c6in.12xlarge	16	32	128
c6in.16xlarge	16	32	240

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c6in.24xlarge	32	32	480
c6in.32xlarge	32	32	512
C8a			
c8a.medium	1	1	3
c8a.large	2	2	6
c8a.xlarge	4	4	16
c8a.2xlarge	8	8	32
c8a.4xlarge	8	16	64
c8a.8xlarge	8	32	128
c8a.12xlarge	16	64	192
c8a.16xlarge	16	64	256
c8a.24xlarge	16	128	384
c8a.48xlarge	32	128	768
c8a.metal-24xl	16	128	384
c8a.metal-48xl	32	128	768
C8gb			
c8gb.medium	1	1	2
c8gb.large	2	2	6
c8gb.xlarge	4	4	16
c8gb.2xlarge	8	8	32

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c8gb.4xlarge	8	16	64
c8gb.8xlarge	8	32	128
c8gb.12xlarge	16	64	192
c8gb.16xlarge	16	64	256
c8gb.24xlarge	16	128	384
c8gb.48xlarge	32	128	768
c8gb.metal-24xl	32	128	768
c8gb.metal-48xl	32	128	768
C8gn			
c8gn.medium	1	1	2
c8gn.large	2	2	6
c8gn.xlarge	4	4	16
c8gn.2xlarge	8	8	32
c8gn.4xlarge	8	16	64
c8gn.8xlarge	8	32	128
c8gn.12xlarge	16	64	192
c8gn.16xlarge	16	64	256
c8gn.24xlarge	16	128	384
c8gn.48xlarge	32	128	768
c8gn.metal-24xl	32	128	768

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c8gn.metal-48xl	32	128	768
c8i			
c8i.large	2	2	6
c8i.xlarge	4	4	16
c8i.2xlarge	8	8	32
c8i.4xlarge	8	16	64
c8i.8xlarge	8	32	128
c8i.12xlarge	16	64	192
c8i.16xlarge	16	64	256
c8i.24xlarge	16	128	384
c8i.32xlarge	16	128	512
c8i.48xlarge	32	128	768
c8i.96xlarge	32	128	1536
c8i.metal-48xl	32	128	768
c8i.metal-96xl	32	128	1536
C8id			
c8id.large	2	2	6
c8id.xlarge	4	4	16
c8id.2xlarge	8	8	32
c8id.4xlarge	8	16	64

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c8id.8xlarge	8	32	128
c8id.12xlarge	16	64	192
c8id.16xlarge	16	64	256
c8id.24xlarge	16	128	384
c8id.32xlarge	16	128	512
c8id.48xlarge	32	128	768
c8id.96xlarge	32	128	1536
c8id.metal-48xl	32	128	768
c8id.metal-96xl	32	128	1536
C8i-flex			
c8i-flex.large	1	1	3
c8i-flex.xlarge	2	2	8
c8i-flex.2xlarge	4	4	16
c8i-flex.4xlarge	4	8	32
c8i-flex.8xlarge	4	16	64
c8i-flex.12xlarge	8	32	96
c8i-flex.16xlarge	8	32	128
C8in			
c8in.large	2	2	8
c8in.xlarge	4	4	16

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c8in.2xlarge	8	8	32
c8in.4xlarge	8	16	64
c8in.8xlarge	16	32	128
c8in.12xlarge	16	64	192
c8in.16xlarge	16	64	256
c8in.24xlarge	16	128	256
c8in.32xlarge	32	128	512
c8in.48xlarge	32	128	768
c8in.96xlarge	32	128	1536 *
c8in.metal-48xl	32	128	768
c8in.metal-96xl	32	128	1536 *
C8ine			
c8ine.large	2	2	8
c8ine.xlarge	4	4	16
c8ine.2xlarge	8	8	32
c8ine.4xlarge	16	16	128
c8ine.8xlarge	32	32	256
c8ine.12xlarge	32	64	384
C8ib			
c8ib.large	2	2	8

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
c8ib.xlarge	4	4	16
c8ib.2xlarge	8	8	32
c8ib.4xlarge	8	16	64
c8ib.8xlarge	16	32	128
c8ib.12xlarge	16	64	192
c8ib.16xlarge	16	64	256
c8ib.24xlarge	16	128	256
c8ib.32xlarge	32	128	512
c8ib.48xlarge	32	128	768
c8ib.96xlarge	32	128	1536 *
c8ib.metal-48xl	32	128	768
c8ib.metal-96xl	32	128	1536 *

Note

* 这些实例类型配备多个网卡。其他实例类型配备单个网卡。有关更多信息，请参阅 网卡。

内存优化型

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
R6i			

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r6i.large	2	2	6
r6i.xlarge	4	4	16
r6i.2xlarge	8	8	32
r6i.4xlarge	8	16	64
r6i.8xlarge	8	32	64
r6i.12xlarge	8	32	64
r6i.16xlarge	8	32	120
r6i.24xlarge	8	32	120
r6i.32xlarge	8	32	120
R6id			
r6id.large	2	2	6
r6id.xlarge	4	4	16
r6id.2xlarge	8	8	32
r6id.4xlarge	8	16	64
r6id.8xlarge	8	32	64
r6id.12xlarge	8	32	64
r6id.16xlarge	8	32	120
r6id.24xlarge	8	32	120
r6id.32xlarge	8	32	120
R6idn			

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r6idn.large	2	2	6
r6idn.xlarge	4	4	16
r6idn.2xlarge	8	8	32
r6idn.4xlarge	8	16	64
r6idn.8xlarge	16	32	128
r6idn.12xlarge	16	32	128
r6idn.16xlarge	16	32	240
r6idn.24xlarge	32	32	480
r6idn.32xlarge	32	32	512
R6in			
r6in.large	2	2	6
r6in.xlarge	4	4	16
r6in.2xlarge	8	8	32
r6in.4xlarge	8	16	64
r6in.8xlarge	16	32	128
r6in.12xlarge	16	32	128
r6in.16xlarge	16	32	240
r6in.24xlarge	32	32	480
r6in.32xlarge	32	32	512
R8a			

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8a.medium	1	1	3
r8a.large	2	2	6
r8a.xlarge	4	4	16
r8a.2xlarge	8	8	32
r8a.4xlarge	8	16	64
r8a.8xlarge	8	32	128
r8a.12xlarge	16	64	192
r8a.16xlarge	16	64	256
r8a.24xlarge	16	128	384
r8a.48xlarge	32	128	768
r8a.metal-24xl	16	128	384
r8a.metal-48xl	32	128	768
R8gb			
r8gb.medium	1	1	2
r8gb.large	2	2	6
r8gb.xlarge	4	4	16
r8gb.2xlarge	8	8	32
r8gb.4xlarge	8	16	64
r8gb.8xlarge	8	32	128
r8gb.12xlarge	16	64	192

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8gb.16xlarge	16	64	256
r8gb.24xlarge	16	128	384
r8gb.48xlarge	32	128	768
r8gb.metal-24xl	32	128	768
r8gb.metal-48xl	32	128	768
R8gn			
r8gn.medium	1	1	2
r8gn.large	2	2	6
r8gn.xlarge	4	4	16
r8gn.2xlarge	8	8	32
r8gn.4xlarge	8	16	64
r8gn.8xlarge	8	32	128
r8gn.12xlarge	16	64	192
r8gn.16xlarge	16	64	256
r8gn.24xlarge	16	128	384
r8gn.48xlarge	32	128	768
r8gn.metal-24xl	32	128	768
r8gn.metal-48xl	32	128	768
R8i			
r8i.large	2	2	6

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8i.xlarge	4	4	16
r8i.2xlarge	8	8	32
r8i.4xlarge	8	16	64
r8i.8xlarge	8	32	128
r8i.12xlarge	16	64	192
r8i.16xlarge	16	64	256
r8i.24xlarge	16	128	384
r8i.32xlarge	16	128	512
r8i.48xlarge	32	128	768
r8i.96xlarge	32	128	1536
r8i.metal-48xl	32	128	768
r8i.metal-96xl	32	128	1536
	R8id		
r8id.large	2	2	6
r8id.xlarge	4	4	16
r8id.2xlarge	8	8	32
r8id.4xlarge	8	16	64
r8id.8xlarge	8	32	128
r8id.12xlarge	16	64	192
r8id.16xlarge	16	64	256

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8id.24xlarge	16	128	384
r8id.32xlarge	16	128	512
r8id.48xlarge	32	128	768
r8id.96xlarge	32	128	1536
r8id.metal-48xl	32	128	768
r8id.metal-96xl	32	128	1536
r8i-flex			
r8i-flex.large	1	1	3
r8i-flex.xlarge	2	2	8
r8i-flex.2xlarge	4	4	16
r8i-flex.4xlarge	4	8	32
r8i-flex.8xlarge	4	16	64
r8i-flex.12xlarge	8	32	96
r8i-flex.16xlarge	8	32	128
R8in			
r8in.large	2	2	8
r8in.xlarge	4	4	16
r8in.2xlarge	8	8	32
r8in.4xlarge	8	16	64
r8in.8xlarge	16	32	128

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8in.12xlarge	16	64	192
r8in.16xlarge	16	64	256
r8in.24xlarge	16	128	256
r8in.32xlarge	32	128	512
r8in.48xlarge	32	128	768
r8in.96xlarge	32	128	1536 *
r8in.metal-48xl	32	128	768
r8in.metal-96xl	32	128	1536 *
R8idn			
r8idn.large	2	2	8
r8idn.xlarge	4	4	16
r8idn.2xlarge	8	8	32
r8idn.4xlarge	8	16	64
r8idn.8xlarge	16	32	128
r8idn.12xlarge	16	64	192
r8idn.16xlarge	16	64	256
r8idn.24xlarge	16	128	256
r8idn.32xlarge	32	128	512
r8idn.48xlarge	32	128	768
r8idn.96xlarge	32	128	1536 *

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8idn.metal-48xl	32	128	768
r8idn.metal-96xl	32	128	1536 *
R8ib			
r8ib.large	2	2	8
r8ib.xlarge	4	4	16
r8ib.2xlarge	8	8	32
r8ib.4xlarge	8	16	64
r8ib.8xlarge	16	32	128
r8ib.12xlarge	16	64	192
r8ib.16xlarge	16	64	256
r8ib.24xlarge	16	128	256
r8ib.32xlarge	32	128	512
r8ib.48xlarge	32	128	768
r8ib.96xlarge	32	128	1536 *
r8ib.metal-48xl	32	128	768
r8ib.metal-96xl	32	128	1536 *
R8idb			
r8idb.large	2	2	8
r8idb.xlarge	4	4	16
r8idb.2xlarge	8	8	32

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
r8idb.4xlarge	8	16	64
r8idb.8xlarge	16	32	128
r8idb.12xlarge	16	64	192
r8idb.16xlarge	16	64	256
r8idb.24xlarge	16	128	256
r8idb.32xlarge	32	128	512
r8idb.48xlarge	32	128	768
r8idb.96xlarge	32	128	1536 *
r8idb.metal-48xl	32	128	768
r8idb.metal-96xl	32	128	1536 *
X8aedz			
x8aedz.large	2	2	8
x8aedz.xlarge	4	4	16
x8aedz.3xlarge	4	16	48
x8aedz.6xlarge	8	32	96
x8aedz.12xlarge	8	64	192
x8aedz.24xlarge	16	128	384
x8aedz.metal-12xl	8	64	192
x8aedz.metal-24xl	16	128	384
X8i			

实例类型	每个接口的默认 ENA 队列	每个接口的最大 ENA 队列数	每个实例的最大 ENA 队列数
x8i.large	2	2	6
x8i.xlarge	4	4	16
x8i.2xlarge	8	8	32
x8i.4xlarge	8	16	64
x8i.8xlarge	8	32	128
x8i.12xlarge	16	64	192
x8i.16xlarge	16	64	256
x8i.24xlarge	16	128	384
x8i.32xlarge	16	128	512
x8i.48xlarge	32	128	768
x8i.64xlarge	32	128	1024
x8i.96xlarge	32	128	1536
x8i.metal-48xl	32	128	768
x8i.metal-96xl	32	128	1536

> **Note**
>
> * 这些实例类型配备多个网卡。其他实例类型配备单个网卡。有关更多信息，请参阅 网卡。

修改队列数

您可以使用 Amazon Web Services 管理控制台、Amazon CLI 或 PowerShell 修改 ENA 队列数。在 Amazon Web Services 管理控制台中，ENA 队列配置位于每个网络接口的设置下。

> **Note**
>
> - 必须停止实例后才能修改 ENA 队列数。
> - ENA 队列的值必须是 2 的乘方，例如 1、2、4、8、16、32 等。
> - 分配给任何单个 ENI 的队列数不能超过实例上可用的 vCPU 数。

在修改队列数之前，请使用以下命令检查您当前的队列数。

Amazon CLI

```
aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

PowerShell

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances.NetworkInterfaces |
  Select-Object NetworkInterfaceId,
    @{N='DeviceIndex';E={$_.Attachment.DeviceIndex}},
    @{N='AttachmentId';E={$_.Attachment.AttachmentId}},
    @{N='EnaQueueCount';E={$_.Attachment.EnaQueueCount}}
```

将网络接口连接至 ENA 队列

以下示例在一个 ENI 上配置了 16 个 ENA 队列。

Amazon CLI

要将网络接口连接至 ENA 队列

使用 attach-network-interface 命令。

```
aws ec2 attach-network-interface \
  --network-interface-id eni-abcdef01234567890 \
  --instance-id i-1234567890abcdef0 \
  --device-index 1 \
  --ena-queue-count 16
```

PowerShell

要将网络接口连接至 ENA 队列

使用 Add-EC2NetworkInterface cmdlet。

```
Add-EC2NetworkInterface `
  -NetworkInterfaceId eni-abcdef01234567890 `
  -InstanceId i-1234567890abcdef0 `
  -DeviceIndex 1 `
  -EnaQueueCount 16
```

启动带 ENA 队列的实例

以下示例有 3 个 ENI，每个均配置了 16 个 ENA 队列。

Amazon CLI

要启动带 ENA 队列的实例

可以使用 run-instances 命令。

```
aws ec2 run-instances \
  --image-id ami-1234567890abcdef0 \
  --instance-type c8i.4xlarge \
  --network-interfaces \
    "[{\"DeviceIndex\":0,\"SubnetId\":\"subnet-abcdef01234567890\",\"EnaQueueCount
\":16},
      {\"DeviceIndex\":1,\"SubnetId\":\"subnet-abcdef01234567890\",\"EnaQueueCount
\":16},
      {\"DeviceIndex\":2,\"SubnetId\":\"subnet-abcdef01234567890\",\"EnaQueueCount
\":16}]"
```

PowerShell

要启动带 ENA 队列的实例

使用 New-EC2Instance cmdlet。

```
New-EC2Instance `
  -ImageId ami-1234567890abcdef0 `
  -InstanceType c8i.4xlarge `
  -NetworkInterface @(
    @{DeviceIndex=0; SubnetId="subnet-abcdef01234567890"; EnaQueueCount=16},
    @{DeviceIndex=1; SubnetId="subnet-abcdef01234567890"; EnaQueueCount=16},
    @{DeviceIndex=2; SubnetId="subnet-abcdef01234567890"; EnaQueueCount=16}
```

```
)
```

修改现有网络接口上的 ENA 队列

以下示例在一个 ENI 上配置了 16 个 ENA 队列。

Amazon CLI

要修改网络接口上的 ENA 队列

使用 modify-network-interface-attribute 命令。

```
aws ec2 modify-network-interface-attribute \
  --network-interface-id eni-1234567890abcdef0 \
  --attachment AttachmentId=eni-attach-1234567890abcdef0,EnaQueueCount=16
```

PowerShell

要修改网络接口上的 ENA 队列

使用 Edit-EC2NetworkInterfaceAttribute cmdlet。

```
Edit-EC2NetworkInterfaceAttribute `
  -NetworkInterfaceId eni-1234567890abcdef0 `
  -Attachment_AttachmentId eni-attach-1234567890abcdef0 `
  -Attachment_EnaQueueCount 16
```

以下示例将 ENA 数重置为默认值。

Amazon CLI

```
aws ec2 modify-network-interface-attribute \
  --network-interface-id eni-1234567890abcdef0 \
  --attachment AttachmentId=eni-attach-1234567890abcdef0,DefaultEnaQueueCount=true
```

PowerShell

```
Edit-EC2NetworkInterfaceAttribute `
  -NetworkInterfaceId eni-1234567890abcdef0 `
  -Attachment_AttachmentId eni-attach-1234567890abcdef0 `
  -Attachment_DefaultEnaQueueCount $true
```

排查 Linux 上 ENA 内核驱动程序的问题

Elastic Network Adapter (ENA) 旨在改进操作系统运行状况和降低因意外硬件行为或故障而导致长期中断的几率。ENA 架构保持设备或驱动程序故障对系统尽可能透明。本主题提供了关于 ENA 的故障排除信息。

如果您不能连接到实例，请先从对连接问题进行故障排除部分开始。

如果您在迁移到第六代实例类型后遇到性能下降，请参阅文章在将 EC2 实例迁移到第六代实例之前，我需要做些什么才能确保获得最大的网络性能？

如果您能连接到实例，则可以使用本主题后面部分中涵盖的故障检测和恢复机制收集诊断信息。

目录

- 对连接问题进行故障排除
- 保持活动机制
- 注册表读取超时
- 统计信息
- syslog 中的驱动程序错误日志
- 次优配置通知

对连接问题进行故障排除

如果您在启用增强联网时丢失连接，则 ena 模块可能与您的实例当前运行的内核不兼容。如果您为特定内核版本安装该模块（不使用 dkms，或使用配置错误的 dkms.conf 文件），然后更新您的实例内核，则会发生这种情况。如果在启动时加载的实例内核未正确安装 ena 模块，则您的实例将无法识别网络适配器，并且您的实例将变得无法访问。

如果您为 PV 实例或 AMI 启用增强联网，这也会使您的实例无法访问。

如果在启用 ENA 增强联网后您的实例变得无法访问，可以为您的实例禁用 `enaSupport` 属性，这将回退到库存网络适配器。

禁用 ENA 增强联网 (EBS 支持的实例)

1. 在本地计算机上，使用 Amazon EC2 控制台、stop-instances 命令（Amazon CLI）或 Stop-EC2Instance cmdlet（Amazon Tools for PowerShell）停止实例。

2. 在本地计算机上，使用 modify-instance-attribute 命令与 `--no-ena-support` 选项或使用 Edit-EC2InstanceAttribute cmdlet 与 `-EnaSupport $false` 参数来禁用增强型联网属性。
3. 在本地计算机上，使用 Amazon EC2 控制台、 start-instances 命令或 Start-EC2Instance cmdlet 启动实例。
4. (可选) 连接到您的实例，并按照ena中的步骤尝试重新安装具有当前内核版本的 在 EC2 实例上使用 ENA 启用增强联网功能 模块。

禁用 ENA 增强联网 (实例存储支持的实例)

1. 按创建 Amazon S3 支持的 AMI 中的说明创建一个新 AMI。
2. 注册 AMI 时，请务必在 stop-instances 命令（ Amazon CLI ）中包含 `--no-ena-support` 选项或在 Register-EC2Image cmdlet 中包含 `-EnaSupport $false` 参数。

保持活动机制

ENA 设备按固定速度 (通常每秒一次) 发布保持活动事件。ENA 驱动程序实施一种监视机制，用于检查是否存在这些保持活动消息。如果存在一条或多条消息，则重新启动监视，否则此驱动程序将认为设备出现故障，然后执行以下操作：

- 将当前统计数据转储到 syslog
- 重置 ENA 设备
- 重置 ENA 驱动程序状态

上述重置过程可能会在短时间内导致一些流量丢失 (TCP 连接应该能恢复)，但应该不会影响到用户。

例如，如果 ENA 设备在加载无法恢复的配置后进入未知状态，ENA 设备也可能会间接请求设备重置过程，而不发送保持活动通知。

下面是重置过程的示例：

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The watchdog
 process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the current
 statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
```

```
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.......
........
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx 0x7 // At the
 end of the down process, the driver discards incomplete packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version: 0.10 //The
 driver begins its up process
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version: 0.0.1
 implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8] Build date
 [Wed Apr  6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't supported
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The reset
 process is complete
```

注册表读取超时

ENA 架构建议使用有限的内存映射的 I/O (MMIO) 读取操作。ENA 设备驱动程序仅在其初始化过程中访问 MMIO 注册表。

如果驱动程序日志（在 dmesg 输出中可用）指示读取操作失败，这可能是由驱动程序不兼容或编译错误、硬件设备繁忙或硬件故障所导致的。

指示读取操作失败的间歇性日志条目不应视为问题；在这种情况下，驱动程序将重试读取操作。但是，一系列包含读取失败的日志条目则指示驱动程序或硬件问题。

下面是指示读取操作因超时而失败的驱动程序日志条目示例：

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout.
 expected: req id[1] offset[88] actual: req id[57006] offset[0]
[ 47.333715] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout.
 expected: req id[2] offset[8] actual: req id[57007] offset[0]
[ 47.346221] [ENA_COM: ena_com_dev_reset] Reg read32 timeout occurred
```

统计信息

如果您遇到网络性能差或延迟问题，您应该检索设备统计数据并检查这些数据。可以使用 ethtool 获取这些统计数据，如下所示：

```
[ec2-user ~]$ ethtool -S ethN
NIC statistics:
 tx_timeout: 0
 suspend: 0
 resume: 0
 wd_expired: 0
 interface_up: 1
 interface_down: 0
 admin_q_pause: 0
 bw_in_allowance_exceeded: 0
 bw_out_allowance_exceeded: 0
 pps_allowance_exceeded: 0
 conntrack_allowance_available: 450878
 conntrack_allowance_exceeded: 0
 linklocal_allowance_exceeded: 0
 queue_0_tx_cnt: 4329
 queue_0_tx_bytes: 1075749
 queue_0_tx_queue_stop: 0
 ...
```

命令输出参数如下所述：

`tx_timeout`: *N*

Netdev 监视的激活次数。

`suspend`: *N*

驱动程序执行挂起操作的次数。

`resume`: *N*

驱动程序执行恢复操作的次数。

`wd_expired`: *N*

驱动程序在过去的 3 秒内未收到保持活动事件的次数。

`interface_up`: *N*

ENA 接口启动的次数。

`interface_down`: *N*

ENA 接口关闭的次数。

`admin_q_pause`: *N*

在运行状态下找不到管理员队列的次数。

`bw_in_allowance_exceeded`: *N*

因入站聚合带宽超过实例的最大值而排队或丢弃的数据包的数量。

`bw_out_allowance_exceeded`: *N*

因出站聚合带宽超过实例的最大值而排队或丢弃的数据包的数量。

`pps_allowance_exceeded`: *N*

因双向 PPS 超过实例的最大值而排队或丢弃的数据包的数量。*

`conntrack_allowance_available`: *N*

在达到该实例类型的跟踪连接限额之前，实例可以建立的跟踪连接数。仅适用于基于 Nitro 的实例。FreeBSD 实例或 DPDK 环境不支持。

`conntrack_allowance_exceeded`: *N*

由于连接跟踪超过实例的最大值且无法建立新连接而丢弃的数据包的数量。这可能会导致进出实例的流量丢失数据包。

`linklocal_allowance_exceeded`: *N*

由于到本地代理服务的流量的 PPS 超出网络接口的最大值而丢弃的数据包数量。这会影响流向 Amazon DNS 服务、实例元数据服务和 Amazon Time Sync Service 的流量，但不会影响流向自定义 DNS 解析器的流量。

queue_*N*_tx_cnt: *N*

为此队列传输的数据包数。

queue_*N*_tx_bytes: *N*

为此队列传输的字节数。

queue_*N*_tx_queue_stop: *N*

队列 *N* 已满并停止的次数。

queue_*N*_tx_queue_wakeup: *N*

队列 *N* 在停止后恢复的次数。

queue_*N*_tx_dma_mapping_err: *N*

直接内存访问错误计数。如果此值不为 0，则表示系统资源不足。

queue_*N*_tx_linearize: *N*

此对队列尝试 SKB 线性化处理的次数。

queue_*N*_tx_linearize_failed: *N*

此队列的 SKB 线性化处理失败的次数。

queue_*N*_tx_napi_comp: *N*

napi 处理程序为此队列调用 napi_complete 的次数。

queue_*N*_tx_tx_poll: *N*

为此队列 计划 napi 处理程序的次数。

queue_*N*_tx_doorbells: *N*

此队列的传输门铃数。

queue_*N*_tx_prepare_ctx_err: *N*

此队列的 ena_com_prepare_tx 失败的次数。

queue_*N*_tx_bad_req_id: *N*

此队列的 req_id 无效。有效的 req_id = 0 - queue_size - 1。

queue_*N*_tx_llq_buffer_copy: *N*

此队列的报头大小大于 llq 条目的数据包的数量。

queue_*N*_tx_missed_tx: *N*

此队列剩下未完成的数据包数。

queue_*N*_tx_unmask_interrupt: *N*

此队列的 tx 中断被揭开的次数。

queue_*N*_rx_cnt: *N*

为此队列接收的数据包数。

queue_*N*_rx_bytes: *N*

为此队列接收的字节数。

queue_*N*_rx_rx_copybreak_pkt: *N*

rx 队列收到小于此队列的 rx_copybreak 数据包大小的数据包的次数。

queue_*N*_rx_csum_good: *N*

rx 队列收到数据包的次数，其中检查校验和并且对此队列是正确的。

queue_*N*_rx_refil_partial: *N*

驱动程序未成功使用此队列的缓冲区重填空的 rx 队列部分的次数。如果此值不为零，则表示内存资源不足。

queue_*N*_rx_bad_csum: *N*

rx 队列具有此队列的错误校验和的次数（仅当支持 rx 校验和卸载时）。

queue_*N*_rx_page_alloc_fail: *N*

此队列的页分配失败的次数。如果此值不为零，则表示内存资源不足。

queue_*N*_rx_skb_alloc_fail: *N*

此队列的 SKB 分配失败的次数。如果此值不为零，则表示系统资源不足。

queue_*N*_rx_dma_mapping_err: *N*

直接内存访问错误计数。如果此值不为 0，则表示系统资源不足。

queue_*N*_rx_bad_desc_num: *N*

每个数据包使用的缓冲区太多。如果此值不为 0，则表示使用的缓冲区非常小。

queue_*N*_rx_bad_req_id: *N*

此队列的 req_id 无效。有效的 req_id 来自 [0, queue_size-1]。

queue_*N*_rx_empty_rx_ring: *N*

此队列的 rx 队列为空的次数。

queue_*N*_rx_csum_unchecked: *N*

rx 队列收到未检查此队列校验和的数据包的次数。

queue_*N*_rx_xdp_aborted: *N*

XDP 数据包被归类为 XDP_ABORT 的次数。

queue_*N*_rx_xdp_drop: *N*

XDP 数据包被归类为 XDP_DROP 的次数。

queue_*N*_rx_xdp_pass: *N*

XDP 数据包被归类为 XDP_PASS 的次数。

queue_*N*_rx_xdp_tx: *N*

XDP 数据包被归类为 XDP_TX 的次数。

queue_*N*_rx_xdp_invalid: *N*

数据包的 XDP 返回代码无效的次数。

queue_*N*_rx_xdp_redirect: *N*

XDP 数据包被归类为 XDP_REDIRECT 的次数。

queue_*N*_xdp_tx_cnt: *N*

为此队列传输的数据包数。

queue_*N*_xdp_tx_bytes: *N*

为此队列传输的字节数。

queue_*N*_xdp_tx_queue_stop: *N*

此队列已满并停止的次数。

queue_*N*_xdp_tx_queue_wakeup: *N*

此队列在停止后恢复的次数。

queue_*N*_xdp_tx_dma_mapping_err: *N*

直接内存访问错误计数。如果此值不为 0，则表示系统资源不足。

queue_*N*_xdp_tx_linearize: *N*

此队列尝试 XDP 缓冲区线性化的次数。

queue_*N*_xdp_tx_linearize_failed: *N*

此队列的 XDP 缓冲区线性化失败的次数。

queue_*N*_xdp_tx_napi_comp: *N*

为此队列调用 napi_complete 的 napi 处理程序的次数。

queue_*N*_xdp_tx_tx_poll: *N*

为此队列安排了 napi 处理程序的次数。

queue_*N*_xdp_tx_doorbells: *N*

此队列的传输门铃数。

queue_*N*_xdp_tx_prepare_ctx_err: *N*

此队列 ena_com_prepare_tx 失败的次数。此值应始终为零；否则，请查看驱动程序日志。

queue_*N*_xdp_tx_bad_req_id: *N*

此队列的 req_id 无效。有效的 req_id 来自 [0, queue_size-1]。

queue_*N*_xdp_tx_llq_buffer_copy: *N*

使用此队列的 llq 缓冲区副本复制了其标头的数据包的数量。

queue_*N*_xdp_tx_missed_tx: *N*

tx 队列条目错过此队列的完成超时的次数。

queue_*N*_xdp_tx_unmask_interrupt: *N*

此队列的 tx 中断被揭开的次数。

ena_admin_q_aborted_cmd: *N*

已中止的管理命令数。这通常发生在自动恢复过程中。

ena_admin_q_submitted_cmd: *N*

管理队列门铃数。

ena_admin_q_completed_cmd: *N*

管理队列完成数。

ena_admin_q_out_of_space: *N*

驱动程序尝试提交新管理命令但队列已满的次数。

ena_admin_q_no_completion: *N*

驱动程序未获得命令的管理完成的次数。

syslog 中的驱动程序错误日志

ENA 驱动程序会在系统启动期间将日志消息写入到 syslog 中。如果您遇到问题，则可以查看这些日志以检查错误。下面是 ENA 驱动程序在系统启动期间记录在 syslog 中的信息示例以及一些选择消息注释。

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.416939] [ENA_COM:
 ena_com_validate_version] ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.420915] [ENA_COM:
 ena_com_validate_version] ena controller version: 0.0.1 implementation version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.256831] ena 0000:00:03.0: Device
 watchdog is Enabled
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.672947] ena 0000:00:03.0: creating 8 io
 queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.680885] [ENA_COM:
 ena_com_init_interrupt_moderation] Feature 20 isn't supported  // Interrupt moderation
 is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.691609] [ENA_COM:
 ena_com_get_feature_ex] Feature 10 isn't supported // RSS HASH function configuration
 is not supported by the
 device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.694583] [ENA_COM:
 ena_com_get_feature_ex] Feature 18 isn't supported //RSS HASH input source
 configuration is not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.697433] [ENA_COM:
 ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.701064] ena 0000:00:03.0 (unnamed
 net_device) (uninitialized): Cannot set host attributes
```

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.704917] ena 0000:00:03.0: Elastic
 Network Adapter (ENA) found at mem f3000000, mac addr 02:8a:3c:1e:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  480.805037] EXT4-fs (xvda1): re-mounted.
 Opts: (null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  481.025842] NET: Registered protocol family
 10
```

可以忽略哪些错误？

可以忽略以下可能出现在系统错误日志中的关于 Elastic Network Adapter 的警告：

Set host attribute isn't supported

此设备不支持主机属性。

failed to alloc buffer for rx queue

这是可恢复的错误，引发此错误时，表示可能存在内存压力问题。

Feature *X* isn't supported

Elastic Network Adapter 不支持引用的功能。*X* 的可能值包括：

- 10：此设备不支持 RSS 哈希函数配置。
- 12：此设备不支持 RSS 间接表配置。
- 18：此设备不支持 RSS 哈希输入配置。
- 20：此设备不支持中断裁决。
- 27：Elastic Network Adapter 驱动程序不支持从 snmpd 轮询以太网功能。

Failed to config AENQ

Elastic Network Adapter 不支持 AENQ 配置。

Trying to set unsupported AENQ events

此错误表示尝试设置 Elastic Network Adapter 不支持的 AENQ 事件组。

次优配置通知

ENA 设备检测驱动程序中您可以更改的次优配置设置。设备将通知 ENA 驱动程序并将警告记录到控制台。以下示例显示警告消息的格式。

```
Sub-optimal configuration notification code: 1. Refer to Amazon ENA documentation for
 additional details and mitigation options.
```

以下列表显示通知代码详细信息以及对次优配置调查发现的建议操作。

- 代码 1：不建议使用宽 LLQ 配置的 ENA Express

  ENA Express ENI 配置为宽 LLQ。此配置为次优配置，可能会影响 ENA Express 的性能。建议您在使用 ENA Express ENI 时禁用宽 LLQ 设置，如下所示。

  ```
  sudo rmmod ena && sudo modprobe ena force_large_llq_header=0
  ```

  有关 ENA Express 最佳配置的更多信息，请参阅 使用 ENA Express 提高 EC2 实例之间的网络性能。

- 代码 2：不建议使用具有次优 Tx 队列深度的 ENA Express ENI

  ENA Express ENI 配置为次优的 Tx 队列深度。此配置为可能会影响 ENA Express 的性能。我们建议您在使用 ENA Express ENI 时将所有 Tx 队列放大到网络接口的最大值，如下所示。

  您可以运行以下 ethtool 命令来调整 LLQ 大小。要了解有关如何控制、查询和启用 Wide-LLQ 的更多信息，请参阅适用于 ENA 的 Linux 内核驱动程序文档中的 Large Low-Latency Queue (Large LLQ) 主题（详见《Amazon Drivers GitHub 存储库》。

  ```
  ethtool -g interface
  ```

  将 Tx 队列设置为最大深度：

  ```
  ethtool -G interface tx depth
  ```

  有关 ENA Express 最佳配置的更多信息，请参阅 使用 ENA Express 提高 EC2 实例之间的网络性能。

- 代码 3：具有常规 LLQ 大小和 Tx 数据包流量的 ENA 超过了支持的最大标头大小

  默认情况下，ENA LLQ 支持的 Tx 数据包标头大小最大为 96 字节。如果数据包标头大小超过 96 字节，则会丢弃该数据包。要缓解此问题，我们建议您启用 Wide-LLQ，这会将支持的 Tx 数据包标头大小增加到最大 224 字节。

  但在启用 Wide-LLQ 时，最大 Tx 环大小将从 1000 个条目减少到 512 个条目。默认情况下，所有 Nitro v4 及更高版本的实例类型都启用了 Wide-LLQ。

  - Nitro v4 实例类型的默认最大 Wide-LLQ Tx 环大小为 512 个条目，且无法更改。
  - Nitro v5 实例类型的默认 Wide-LLQ Tx 环大小为 512 个条目，但可以增加至 1000 个条目。

您可以运行以下 ethtool 命令来调整 LLQ 大小。要了解有关如何控制、查询和启用 Wide-LLQ 的更多信息，请参阅适用于 ENA 的 Linux 内核驱动程序文档中的 Large Low-Latency Queue (Large LLQ) 主题（详见《Amazon Drivers GitHub 存储库》。

查找 Tx 队列的最大深度：

```
ethtool -g interface
```

将 Tx 队列设置为最大深度：

```
ethtool -G interface tx depth
```

对弹性网络适配器 Windows 驱动程序进行问题排查

Elastic Network Adapter (ENA) 旨在改进操作系统运行状况和降低可能会中断 Windows 实例运行的意外硬件行为或故障。ENA 架构保持设备或驱动程序故障对操作系统尽可能透明。

收集有关实例的诊断信息

打开 Windows 操作系统 (OS) 工具的步骤会有所不同，具体取决于实例上安装的操作系统版本。在以下各节中，我们使用 Run（运行）对话框以打开工具，该工具在所有操作系统版本中都是一样的。但是，您可以使用任何您喜欢的方法访问这些工具。

访问 Run（运行）对话框

- 使用 Windows 徽标键组合：Windows + R
- 使用搜索栏：
  - 在搜索栏中输入 run。
  - 从搜索结果中选择 Run（运行）应用程序。

有些步骤需要上下文菜单才能访问属性或上下文相关操作。根据操作系统版本和硬件，有多种方法可执行此操作。

访问上下文菜单

- 使用鼠标：右键单击某个项目以弹出其上下文菜单。
- 使用键盘：

- 根据您的操作系统版本，使用Shift + F10 或 Ctrl + Shift + F10。
- 如果键盘上有上下文键（一个框中有三条水平线），请选择所需的项目，然后按上下文键。

如果您可以连接到实例，请使用以下方法收集诊断信息以进行故障排除。

检查 ENA 设备状态

要使用 Windows 设备管理器检查 ENA Windows 驱动程序的状态，请按照以下步骤操作：

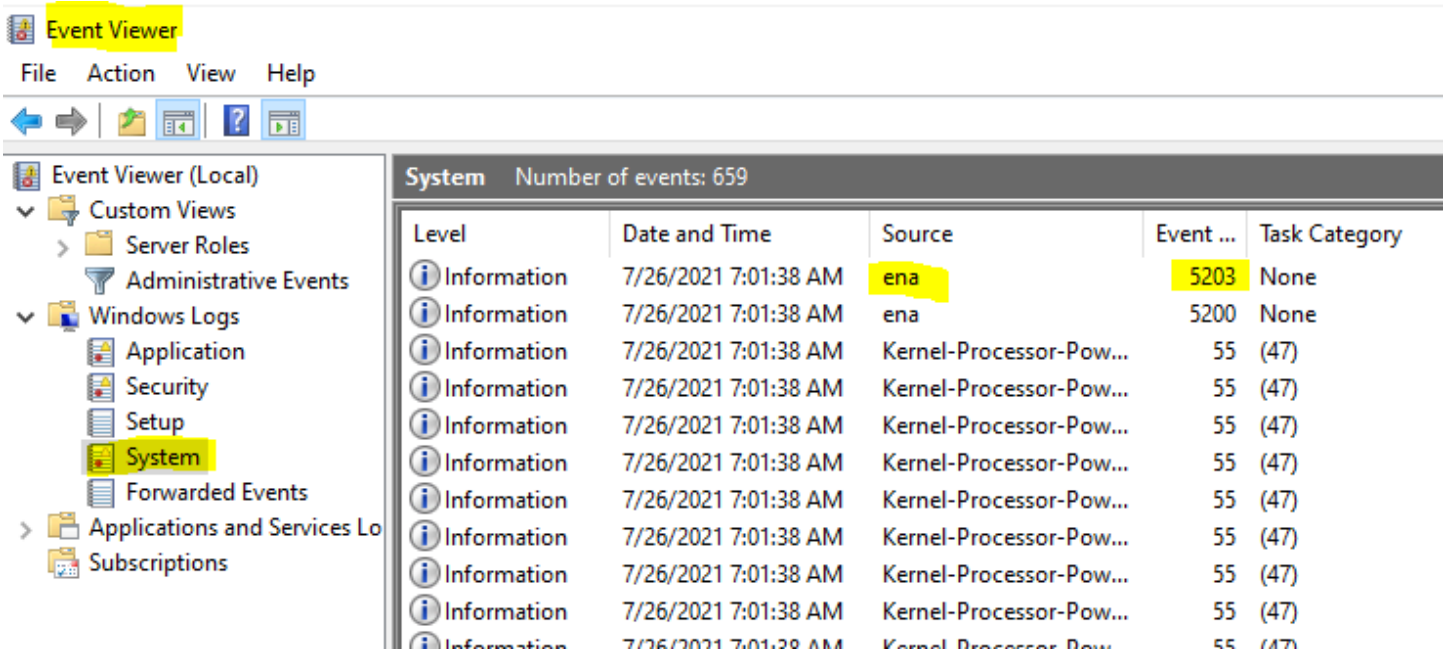
1. 使用上一个部分介绍的方法之一打开 Run（运行）对话框。
2. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 devmgmt.msc。
3. 选择确定。这将打开 Device Manager（设备管理器）窗口。
4. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。
5. 选择名称或打开 Amazon Elastic Network Adapter 的上下文菜单，然后选择 Properties（属性）。这将打开 Amazon 弹性网络适配器属性对话框。
6. 验证常规选项卡中的消息是否显示“此设备工作正常”。

调查驱动程序事件消息

要使用 Windows 事件查看器查看 ENA Windows 驱动程序事件日志，请执行以下步骤：

1. 使用上一个部分介绍的方法之一打开 Run（运行）对话框。
2. 要打开 Windows 事件查看器，请在 Run（运行）框中输入 eventvwr.msc。
3. 选择确定。此操作将打开 Event Viewer（事件查看器）窗口。
4. 展开 Windows Logs（Windows 日志）菜单，然后选择 System（系统）。
5. 在 Actions（操作）下的右上角面板中，选择 Filter Current Log（筛选当前日志）。此时将显示筛选对话框。
6. 在 Event sources（事件来源）框中，输入 ena。这会将结果限制为由 ENA Windows 驱动程序生成的事件。
7. 选择确定。这将在窗口的详细信息部分中显示已筛选的事件日志结果。
8. 要深入了解详细信息，请从列表中选择一条事件消息。

以下示例显示了 Windows 事件查看器系统事件列表中的 ENA 驱动程序事件：



事件消息摘要

下表显示了 ENA Windows 驱动程序生成的事件消息。

Input

事件 ID	ENA 驱动程序事件描述	Type
5001	硬件资源不足	错误
5002	适配器检测到硬件错误	错误
5005	适配器的 NDIS 操作超时，未及时完成	错误
5032	适配器无法重置设备	错误
5200	适配器已初始化	信息性
5201	适配器已停止	信息性
5202	适配器已暂停	信息性
5203	适配器已重新启动	信息性
5204	适配器已关闭	信息性

事件 ID	ENA 驱动程序事件描述	Type
5205	适配器已重置	错误
5206	适配器被意外移除	错误
5208	适配器初始化例程失败	错误
5210	适配器遇到并成功恢复了内部问题	错误

查看性能指标

ENA Windows 驱动程序从启用指标的实例发布网络性能指标。您可以使用本机性能监视器应用程序查看和启用实例上的指标。有关 ENA Windows 驱动程序生成的指标的更多信息，请参阅 监控 EC2 实例上 ENA 设置的网络性能。

在启用了 ENA 指标并安装了 Amazon CloudWatch 代理的实例上，CloudWatch 将收集与 Windows 性能监控器中的计数器关联的指标以及 ENA 的一些高级指标。除了在 EC2 实例上默认启用的指标以外，还会收集这些指标。有关这些指标的更多信息，请参阅 Amazon CloudWatch 用户指南中的 CloudWatch 代理收集的指标。

> **Note**
>
> 性能指标适用于 ENA 驱动程序版本 2.4.0 及更高版本（也适用于版本 2.2.3）。由于第六代 EC2 实例的潜在性能下降，ENA 驱动程序版本 2.2.4 已回滚。我们建议您将驱动程序升级到当前版本，以确保您拥有最新更新。

使用性能指标的一些方法包括：

- 排查实例性能问题。
- 为工作负载选择合适的实例大小。
- 主动计划扩缩活动。
- 设置应用程序基准，以确定它们是否最大限度地提高实例的可用性能。

刷新率

默认情况下，驱动程序使用 1 秒的间隔刷新指标。但是，检索指标的应用程序可能会使用不同的时间间隔进行轮询。您可以使用驱动程序的高级属性在设备管理器中更改刷新间隔。

要更改 ENA Windows 驱动程序的指标刷新间隔，请执行以下步骤：

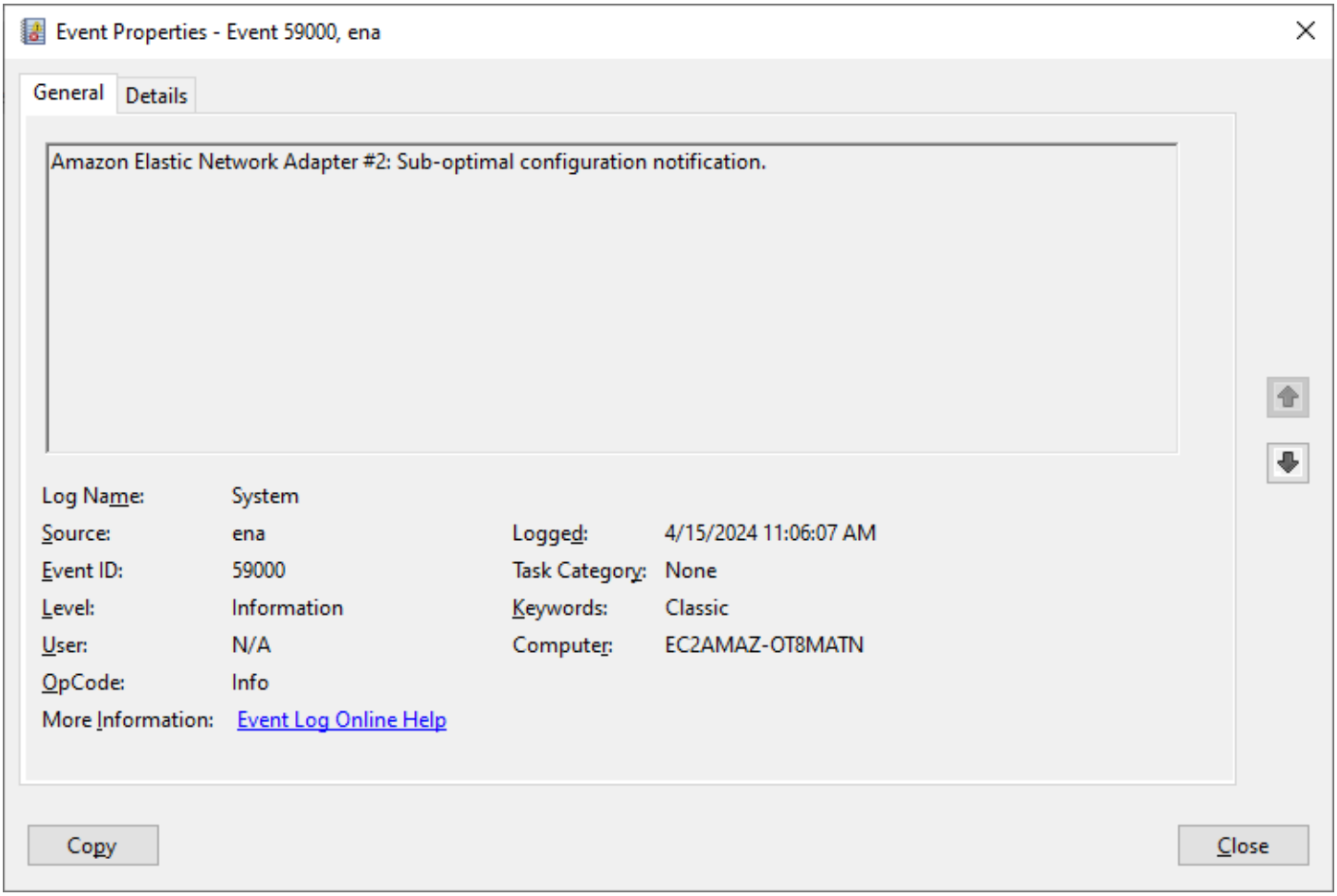
1. 使用上一个部分介绍的方法之一打开 Run（运行）对话框。
2. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 `devmgmt.msc`。
3. 选择确定。这将打开 Device Manager（设备管理器）窗口。
4. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。
5. 选择名称或打开 Amazon Elastic Network Adapter 的上下文菜单，然后选择 Properties（属性）。这将打开 Amazon 弹性网络适配器属性对话框。
6. 打开弹出窗口中的 Advanced（高级）选项卡。
7. 从 Property（属性）列表中选择 Metrics Refresh Interval（指标刷新间隔）以更改值。
8. 完成后，选择 OK（确定）。

调查次优配置通知

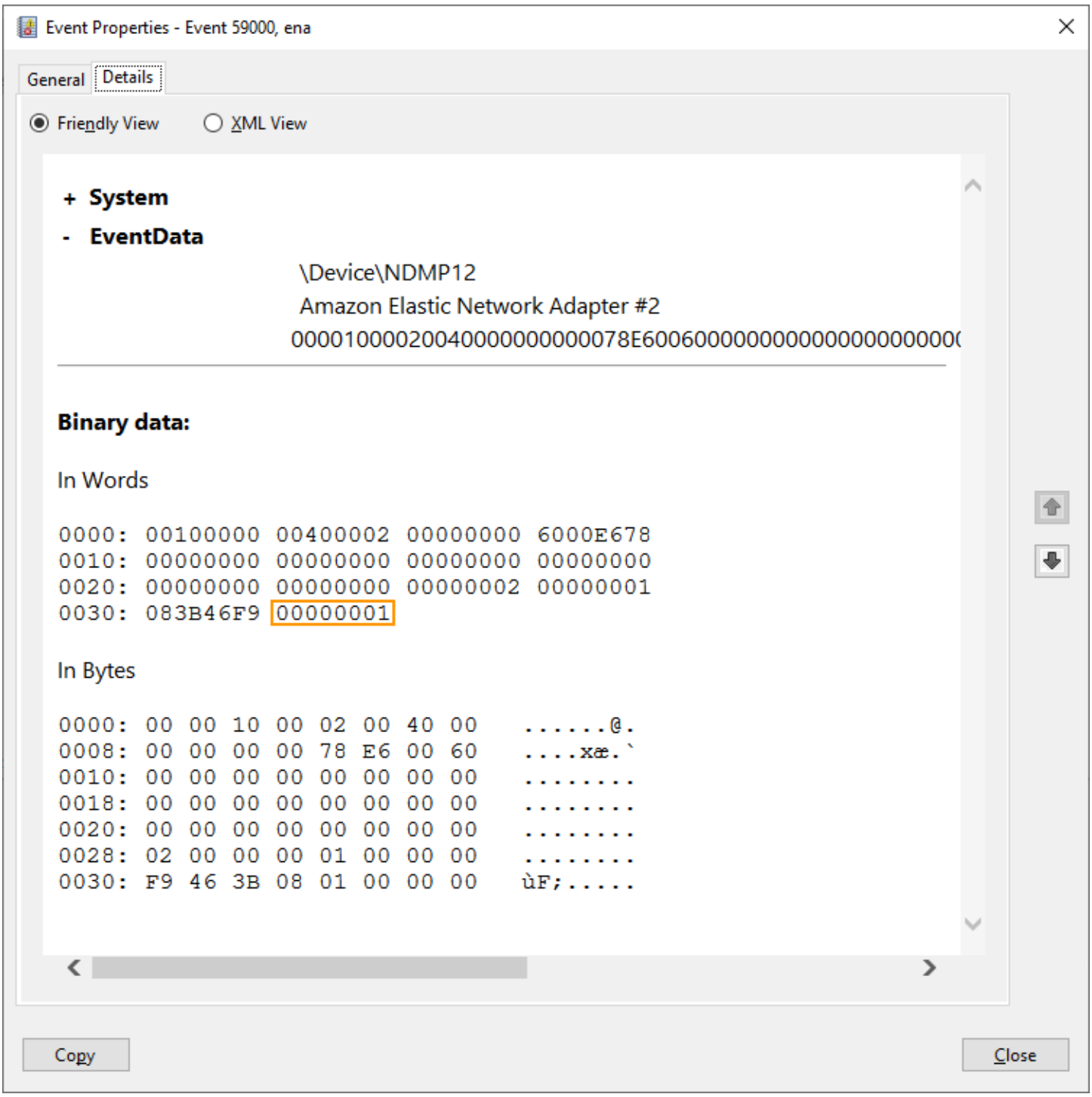
ENA 设备检测驱动程序中您可以更改的次优配置设置。设备将通知 ENA 驱动程序并记录事件通知。要在 Windows 事件查看器中查看次优事件

1. 使用上一个部分介绍的方法之一打开 Run（运行）对话框。
2. 要打开 Windows 事件查看器，请在 Run（运行）框中输入 `eventvwr.msc`。
3. 选择确定。此操作将打开 Event Viewer（事件查看器）窗口。
4. 展开 Windows Logs（Windows 日志）菜单，然后选择 System（系统）。
5. 在 Actions（操作）下的右上角面板中，选择 Filter Current Log（筛选当前日志）。此时将显示筛选对话框。
6. 在 Event sources（事件来源）框中，输入 `ena`。这会将结果限制为由 ENA Windows 驱动程序生成的事件。
7. 选择确定。这将在窗口的详细信息部分中显示已筛选的事件日志结果。

ID 为 `59000` 的事件会通知您次优配置调查发现。右键单击事件并选择事件属性以打开详细视图，或者从视图菜单中选择预览窗格以查看相同的详细信息。



打开详细信息选项卡以查看事件代码。在二进制数据：单词部分中，最后一个单词是代码。



以下列表显示通知代码详细信息以及对次优配置调查发现的建议操作。

- 代码**1**：不建议使用宽 LLQ 配置的 ENA Express

  ENA Express ENI 配置为宽 LLQ。此配置为次优配置，可能会影响 ENA Express 的性能。建议您在使用 ENA Express ENI 时禁用宽 LLQ 设置，如下所示。

1. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 `devmgmt.msc`。
2. 选择确定。这将打开 Device Manager（设备管理器）窗口。
3. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。
4. 打开 `Amazon Elastic Network Adapter` 的设备属性。
5. 从这里请打开高级选项卡进行更改。
6. 选择 LLQ 标头大小策略属性，并将其值设置为 `Normal (128 Bytes)`。
7. 选择确定以保存您的更改。

- 代码 **2**：不建议使用具有次优 Tx 队列深度的 ENA Express ENI

  ENA Express ENI 配置为次优的 Tx 队列深度。此配置为可能会影响 ENA Express 的性能。我们建议您在使用 ENA Express ENI 时将所有 Tx 队列放大到网络接口的最大值，如下所示。

  按照以下步骤将 Tx 队列放大到最大深度：

  1. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 `devmgmt.msc`。
  2. 选择确定。这将打开 Device Manager（设备管理器）窗口。
  3. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。
  4. 打开 `Amazon Elastic Network Adapter` 的设备属性。
  5. 从这里请打开高级选项卡进行更改。
  6. 选择传输缓冲区属性，并将其值设置为支持的最大值。
  7. 选择确定以保存您的更改。

ENA 适配器重置

当 ENA Windows 驱动程序检测到适配器上的错误并将适配器标记为运行状况不佳时，将启动重置过程。驱动程序无法自行重置，因此依赖操作系统来检查适配器的运行状况，并调用 ENA Windows 驱动程序的重置句柄。重置过程可能会在短时间内造成流量丢失。但是，TCP 连接应该能够恢复。

ENA 适配器也可能会间接请求设备重置过程，因为无法发送保持活动状态的通知。例如，如果 ENA 适配器在加载无法恢复的配置后达到未知状态，它可能会停止发送保持活动通知。

ENA 适配器重置的常见原因

- 保持活动消息丢失

ENA 适配器按固定速度（通常为每秒一次）发布保持活动事件。ENA Windows 驱动程序实施一种监视机制，用于定期检查是否存在这些保持活动消息。如果它自上次检查以来检测到一条或多条新消息，则会记录成功的结果。否则，驱动程序得出结论认为设备出现故障，并且会启动重置序列。

- 数据包卡在传输队列中

  ENA 适配器验证数据包是否按预期流过传输队列。ENA Windows 驱动程序检测数据包是否卡住，如果卡住，则启动重置序列。

- 内存映射输入/输出 (MMIO) 寄存器的读取超时

  为了限制内存映射输入/输出 (MMIO) 读取操作，ENA Windows 驱动程序仅在初始化和重置过程中访问 MMIO 寄存器。如果驱动程序检测到超时，则需要执行以下操作之一，具体取决于正在运行的进程：

  - 如果在初始化过程中检测到超时，流动将失败，从而导致驱动程序在 Windows 设备管理器中显示 ENA 适配器的黄色感叹号。
  - 如果在重置期间检测到超时，流动将失败。然后，操作系统会开始突然删除 ENA 适配器，然后通过停止和启动已移除的适配器来恢复它。有关突然移除网络接口卡 (NIC) 的更多信息，请参阅 Microsoft Windows 硬件开发人员文档中的处理 NIC 的意外移除。

诊断场景

以下场景可帮助您解决 ENA Windows 驱动程序可能遇到的问题。如果没有最新版本，我们建议您先升级 ENA 驱动程序。要查找适用于 Windows 操作系统版本的最新驱动程序，请参阅 跟踪 ENA Windows 驱动程序发行版。

安装了意外的 ENA 驱动程序

说明

完成安装特定版本 ENA 驱动程序的步骤后，Windows 设备管理器显示 Windows 安装了其他版本的 ENA 驱动程序。

原因

运行安装驱动程序包时，Windows 会首先为本地驱动程序存储中所有给定设备有效的驱动程序包进行排名，然后才会开始安装。系统会将排名最低的程序视为最佳匹配项。这可能与您计划安装的程序包版本不同。有关设备驱动程序包选择过程的更多信息，请参阅《Microsoft 文档网站》上的 Windows 如何为设备选择驱动程序包。

解决方案

为确保 Windows 安装的是您选择的驱动程序包版本，您可以使用 PnPUtil 命令行工具从驱动程序存储中移除排名较低的驱动程序包。

按照以下步骤安装 ENA 驱动程序：

1. 连接到您的实例并以本地管理员身份登录。
2. 打开设备管理器属性窗口，如 检查 ENA 设备状态 部分所述。这将打开 Amazon 弹性网络适配器属性窗口的常规选项卡。
3. 打开 Driver（驱动程序）选项卡。
4. 选择 Update Driver (更新驱动程序)。这将打开更新驱动程序软件 – Amazon 弹性网络适配器对话框。
   a. 在您希望如何搜索驱动程序软件？部分中，选择浏览计算机以查找驱动程序软件。
   b. 在浏览计算机上的驱动程序软件页面上，选择让我从计算机上的设备驱动程序列表中选择（位于搜索栏下方）。
   c. 在选择要为此硬件安装的设备驱动程序页面上，选择硬盘驱动器...。
   d. 在从磁盘安装窗口中，从下拉列表中选择文件位置旁的浏览...。
   e. 导航至您下载了目标 ENA 驱动程序包的位置。选择名为 `ena.inf` 的文件，然后选择打开。
   f. 要开始安装，请选择确定，然后选择下一步。
5. 如果安装程序没有自动重启实例，请运行 Restart-Computer PowerShell cmdlet。

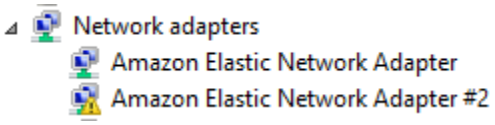
```
PS C:\> Restart-Computer
```

ENA 驱动程序的设备警告

说明

设备管理器网络适配器部分中的 ENA 适配器图标显示一个警告标志（里面带有感叹号的黄色三角形）。

以下示例显示了 Windows 设备管理器中带有警告图标的 ENA 适配器：



原因

此设备警告通常是由环境问题引起的，环境问题可能需要更多的研究，而且通常需要一个消除过程来确定潜在原因。有关设备错误的完整列表，请参阅 Microsoft 文档中的设备管理器错误消息。

解决方案

此设备警告的解决方案取决于根本原因。此处描述的消除过程包括一些基本步骤，以帮助确定和解决可能有一个简单的解决方案的最常见问题。如果这些步骤不能解决问题，则需要进行额外的根本原因分析。

请按照以下步骤帮助确定和解决常见问题：

1. 停止和启动设备

   打开设备管理器属性窗口，如 检查 ENA 设备状态 部分所述。这样一来，将打开 Amazon Elastic Network Adapter Properties（Amazon Elastic Network Adapter 属性）窗口的 General（常规）选项卡，Device status（设备状态）在其中显示错误代码和简短消息。

   a. 打开 Driver（驱动程序）选项卡。
   b. 选择 Disable Device（禁用设备），然后回应 Yes（是）转到显示的警告消息。
   c. 选择 Enable Device（启用设备）。

2. 停止和启动 EC2 实例

   如果适配器仍然在设备管理器中显示警告图标，则下一步是停止并启动 EC2 实例。在大多数情况下，这会在不同的硬件上重新启动实例。

3. 调查可能的实例资源问题

   如果您已停止并启动 EC2 实例，但问题仍然存在，则可能表明您的实例存在资源问题，例如内存不足。

适配器重置后的连接超时（错误代码 5007、5205）

说明

Windows 事件查看器显示 ENA 适配器的组合发生的适配器超时和重置事件。消息与以下示例类似：

- 事件 ID 5007：Amazon Elastic Network Adapter：在操作过程中超时。
- 事件 ID 5205：Amazon Elastic Network Adapter：适配器重置已开始。

适配器重置会导致最小的流量中断。即使进行了多次重置，对他们来说，造成任何严重的网络中断都是不寻常的。

原因

这一系列事件表示 ENA Windows 驱动程序为无响应的 ENA 适配器启动了重置。但是，设备驱动程序用来检测此问题的机制受 CPU 0 匮乏造成的误报的影响。

解决方案

如果这种错误组合频繁发生，请检查您的资源分配以了解哪些调整可能会有帮助。

1. 使用上一个部分介绍的方法之一打开 Run（运行）对话框。
2. 要打开 Windows 资源监视器，请在 Run（运行）框中输入 `resmon`。
3. 选择确定。这将打开资源监视器窗口。
4. 打开 CPU 选项卡。按 CPU 的使用率图表显示在资源监视器窗口的右侧。
5. 检查 CPU 0 的使用水平以查看它们是否太高。

我们建议您将 RSS 配置为在较大的实例类型（超过 16 个 vCPU）上排除 ENA 适配器的 CPU 0。对于较小的实例类型，配置 RSS 可能会改善体验，但是由于可用内核数量减少，必须进行测试，以确保限制 CPU 内核不会对性能产生负面影响。

使用 Set-NetAdapterRss 命令为 ENA 适配器配置 RSS，如以下示例所示。

```
Set-NetAdapterRss -name (Get-NetAdapter | Where-Object {$_.InterfaceDescription -like
 "*Elastic*"}).Name -Baseprocessorgroup 0 -BaseProcessorNumber 1
```

迁移到第六代实例基础设施会影响性能或附加

说明

如果迁移到第六代 EC2 实例，若尚未更新 ENA Windows 驱动程序版本，则可能会遇到性能降低或 ENA 附加失败的情况。

原因

根据实例操作系统（OS），第六代 EC2 实例类型需要以下 ENA Windows 驱动程序的最低版本。

最低版本

Windows Server 版本	ENA 驱动程序版本
Windows Server 2008 R2	2.2.3 或 2.4.0
Windows Server 2012 及更高版本	2.2.3 及更高版本
Windows 工作站	2.2.3 及更高版本

解决方案

在升级到第六代 EC2 实例之前，请确保从中启动的 AMI 具有基于实例操作系统的兼容驱动程序，如上表所示。有关更多信息，请参阅 Amazon Web Services re:Post 知识中心中的在将 EC2 实例迁移到第六代实例之前，我需要做些什么才能确保获得最大的网络性能？。

弹性网络接口的性能不佳

说明

ENA 接口没有按预期运行。

原因

性能问题的根本原因分析是一个消除过程。涉及的变量太多，无法指出共同原因。

解决方案

根本原因分析的第一步是查看未按预期运行的实例的诊断信息，以确定是否存在可能导致问题的错误。想要了解更多信息，请参阅收集有关实例的诊断信息部分。

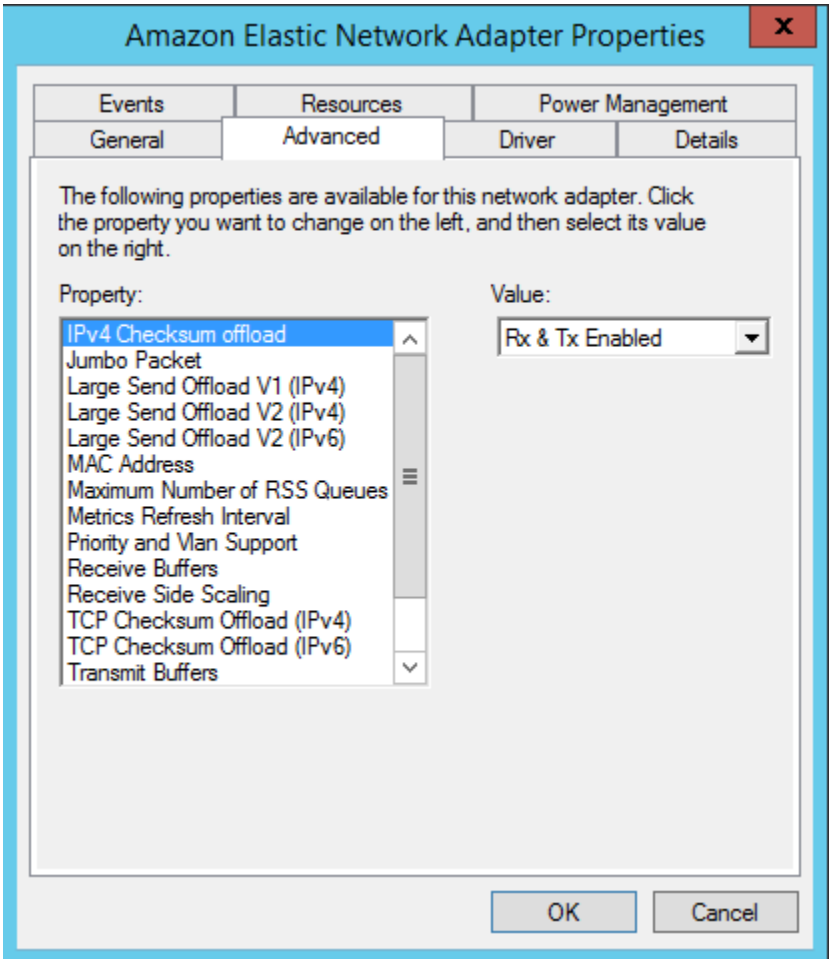
若要在具有增强联网的实例上实现最高网络性能，您可能需要修改原定设置操作系统配置。官方 Windows AMI 已默认配置了一些优化（例如打开校验和卸载和启用 RSS）。有关可以应用于 ENA 适配器的其他优化，请参阅 ENA 适配器性能调整 中显示的性能调整。

我们建议您谨慎行事，并将设备属性调整限制为本节中列出的调整，或者仅限于 Amazon 支持团队建议的特定更改。

要更改 ENA 适配器属性，请执行以下步骤：

1. 使用上一个部分介绍的方法之一打开 Run（运行）对话框。
2. 要打开 Windows 设备管理器，请在 Run（运行）框中输入 `devmgmt.msc`。
3. 选择确定。这将打开 Device Manager（设备管理器）窗口。
4. 选择 Network adapters（网络适配器）左侧的箭头以展开列表。
5. 选择名称或打开 Amazon Elastic Network Adapter 的上下文菜单，然后选择 Properties（属性）。这将打开 Amazon 弹性网络适配器属性对话框。
6. 要进行更改，请打开高级选项卡。
7. 完成后，选择确定以保存更改。

以下示例显示了 Windows 设备管理器的 ENA 适配器属性：



ENA 适配器性能调整

下表包括可以调整以提高 ENA 接口性能的属性。

Input

属性	说明	默认 值	调整
接收缓冲区	控制软件接收队列中的条目数。	1024	可以最高增加到 8192。
接收端缩放 (RSS)	支持在多处理器系统中的多个 CPU 之间高效分配网络接收处理。	已启用	您可以将负载分散到多个处理器之间。要了解更多信息，请参阅优化 EC2 Windows 实例上的网络性能。
RSS 队列的最大数量	设置在启用 RSS 时允许的最大 RSS 队列数。	32	RSS 队列的数量是在驱动程序初始化过程中确定的，其中包括以下限制（除其他外）： • 此属性设置的 RSS 队列限制 • 实例限制（vCPU 计数） • 硬件生成限制（ENAv1 中最多 8 个 RSS 队列，ENAv2 中最多 32 个 RSS 队列） 根据实例和硬件生成限制，您可以将值设置为 1 到 32。要了解

属性	说明	默认 值	调整
			更多信息，请参阅优化 EC2 Windows 实例上的网络性能。
巨型数据包	允许使用巨型以太网帧（超过 1500 字节的有效负载）。	已禁用（这会将有效负载限制为 1500 字节或以下）	可以将值最高设置为 9015，这将转换为 9001 字节的有效负载。这是巨型以太网帧的最大有效负载。请参阅使用巨型以太网帧的注意事项。

使用巨型以太网帧的注意事项

巨型帧通过增加每个数据包的有效负载大小，从而增加数据包中不属于数据包开销的百分比来支持 1500 个字节以上的数据。发送等量的可用数据所需要的数据包更少。但是，在以下情况下，流量限制为最大 MTU 为 1500：

- EC2 Classic 给定Amazon区域外的流量
- 单个 VPC 外的流量。
- 区域间 VPC 对等连接中的流量。
- VPN 连接的流量。
- 互联网网关流量。

> **Note**
>
> 超过 1500 字节的数据包被分成片段。如果您在 IP 标头中设置了 `Don't Fragment` 标记，这些数据包将被丢弃。
>
> 对于面向 Internet 的流量或离开 VPC 的任何流量，应谨慎使用巨型帧。中间系统会对数据包进行分段，从而减缓此流量。要在 VPC 内使用巨型帧而不影响离开 VPC 的出站流量，请尝试以下选项之一：
>
> - 按路由配置 MTU 大小。

- 使用具有不同 MTU 大小和不同路由的多个网络接口。

巨型帧的推荐使用案例

巨型帧对于 VPC 内部和之间的流量非常有用。我们建议在以下使用案例中使用巨型帧：

- 对于在集群置放群组中并置的实例，巨型帧有助于实现可能的最大网络吞吐量。有关更多信息，请参阅 Amazon EC2 实例的置放群组。
- 您可以通过 Amazon Direct Connect 使用巨型帧在 VPC 与本地网络之间进行通信。有关使用 Amazon Direct Connect 以及验证巨型帧功能的更多信息，请参阅《Amazon Direct Connect User Guide》中的 MTUs for private virtual interfaces or transit virtual interfaces。
- 有关中转网关支持的 MTU 大小的更多信息，请参阅 Amazon VPC 中转网关中的您的中转网关的配额。

使用 ENA Express 提高 EC2 实例之间的网络性能

ENA Express 由 Amazon 可扩展的可靠数据报（SRD）技术提供支持。SRD 是一种高性能网络传输协议，使用动态路由来提高吞吐量并最大限度地减少尾部延迟。使用 ENA Express，您可在同一可用区中的两个 EC2 实例之间通信，或在同一区域不同可用区中的两个 EC2 实例之间通信。

ENA Express 的优势

- 将同一区域内的单个流可用最大带宽从 5 Gbps 提升到高达 25 Gbps，最大不超过实例的总计限制。
- 减少同一可用区内 EC2 实例之间网络流量的尾部延迟，尤其是在网络高负载期间。
- 检测并避开拥塞的网络路径。
- 直接在网络层处理某些任务，比如接收端的数据包重新排序以及所需的大部分重传。这样可以腾出应用层用于其他工作。

Note

- 如果您的应用程序对每秒网络数据包数的要求很高，并且需要针对非拥塞时期的延迟进行优化，那么增强联网可能更合适。
- ENA Express 流量无法在本地区域中发送。

- ENA Express 对跨可用区流量的支持在南美洲（圣保罗）、中东（巴林）和中东（阿联酋）区域不可用。

为实例上的网络接口挂载启用 ENA Express 后，发送实例会启动与接收实例的通信，SRD 会检测 ENA Express 是否同时在发送实例和接收实例上运行。如果 ENA Express 正在运行，则通信可以使用 SRD 传输。如果 ENA Express 未在运行，则通信将退回到标准 ENA 传输。

在网络流量较少的时段内，当数据包使用 ENA Express 时，您可能会注意到数据包中位数延迟略有增加（数十微秒）。在此期间，优先考虑特定网络性能特征的应用程序可以从 ENA Express 中受益，如下所示：

- 同一个区域内的最大单个流带宽从 5 Gbps 提升到高达 25 Gbps（最大不超过实例的总计限制）时，进程可以从中受益。例如，如果一种特定的实例类型支持最大 12.5Gbps 的贷款，单个流的带宽上限也将是 12.5Gbps。
- 在网络拥塞时段，同一可用区中运行时间较长的进程会减少尾部延迟。
- 网络响应时间分配更流畅、更标准时，进程可以从中受益。

主题

- ENA Express 的工作原理
- ENA Express 支持的实例类型
- 调整 Linux 实例上 ENA Express 设置的性能
- 查看 EC2 实例的 ENA Express 设置
- 为 EC2 实例配置 ENA Express 设置

ENA Express 的工作原理

ENA Express 由 Amazon 可扩展的可靠数据报（SRD）技术提供支持。该技术通过不同的 Amazon 网络路径为每个网络流分配数据包，并在检测到拥塞迹象时动态调整分配。它还能管理接收端的数据包重新排序。

为确保 ENA Express 能够按预期方式管理网络流量，实例的发送和接收以及实例之间的通信必须满足以下所有要求：

- 同时支持发送和接收实例类型。有关更多信息，请参阅 ENA Express 支持的实例类型 表。

- 发送和接收实例都必须配置 ENA Express。如配置存在差异，则可能会遇到流量默认为标准 ENA 传输的情况。以下方案显示了可能发生的情况。

  方案：配置差异

实例	已启用 ENA Express	UDP 使用 ENA Express
实例 1	支持	是
实例 2	是	否

  在这种情况下，两个实例之间的 TCP 流量可以使用 ENA Express，因为两个实例均已启用该功能。但是，由于其中一个实例不将 ENA Express 用于 UDP 流量，因此这两个实例之间通过 UDP 进行的通信则使用标准 ENA 传输。
- 发送和接收实例必须在同一区域中运行。
- 实例之间的网络路径不得包含中间件。ENA Express 目前不支持中间件。
- （仅限 Linux 实例）要充分利用带宽潜力，请使用驱动程序版本 2.2.9 或更高版本。
- （仅限 Linux 实例）要生成指标，请使用驱动程序版本 2.8 或更高版本。

如果未满足任何要求，实例将使用标准 TCP/UDP 协议，但不使用 SRD 进行通信。

为确保您的实例网络驱动程序配置为最佳性能，请查看 ENA 驱动程序的最佳实践推荐。这些最佳实践同样适用于 ENA Express。有关更多信息，请参阅 GitHub 网站上的 ENA Linux Driver Best Practices and Performance Optimization Guide（ENA Linux 驱动程序最佳实践和性能优化指南）。

> **Note**
>
> Amazon EC2 是指实例与作为附件附加到该实例的网络接口之间的关系。ENA Express 设置适用于该附件。如果网络接口从实例中分离，则该附件将不再存在，应用于它的 ENA Express 设置将不再生效。终止实例时也是如此，即使网络接口仍然存在。

在为发送和接收实例上的网络接口附件启用 ENA Express 后，您可以使用 ENA Express 指标来帮助确保实例充分利用 SRD 技术所提供的性能改进。有关 ENA Express 指标的更多信息，请参阅 ENA Express 指标。

ENA Express 支持的实例类型

以下实例类型支持 ENA Express。

General purpose

实例类型	架构
m6a.12xlarge	x86_64
m6a.16xlarge	x86_64
m6a.24xlarge	x86_64
m6a.32xlarge	x86_64
m6a.48xlarge	x86_64
m6a.metal	x86_64
m6i.8xlarge	x86_64
m6i.12xlarge	x86_64
m6i.16xlarge	x86_64
m6i.24xlarge	x86_64
m6i.32xlarge	x86_64
m6i.metal	x86_64
m6id.8xlarge	x86_64
m6id.12xlarge	x86_64
m6id.16xlarge	x86_64
m6id.24xlarge	x86_64
m6id.32xlarge	x86_64

实例类型	架构
m6id.metal	x86_64
m6idn.8xlarge	x86_64
m6idn.12xlarge	x86_64
m6idn.16xlarge	x86_64
m6idn.24xlarge	x86_64
m6idn.32xlarge	x86_64
m6idn.metal	x86_64
m6in.8xlarge	x86_64
m6in.12xlarge	x86_64
m6in.16xlarge	x86_64
m6in.24xlarge	x86_64
m6in.32xlarge	x86_64
m6in.metal	x86_64
m7a.12xlarge	x86_64
m7a.16xlarge	x86_64
m7a.24xlarge	x86_64
m7a.32xlarge	x86_64
m7a.48xlarge	x86_64
m7a.metal-48xl	x86_64
m7g.12xlarge	arm64

实例类型	架构
m7g.16xlarge	arm64
m7g.metal	arm64
m7gd.12xlarge	arm64
m7gd.16xlarge	arm64
m7gd.metal	arm64
m7i.12xlarge	x86_64
m7i.16xlarge	x86_64
m7i.24xlarge	x86_64
m7i.48xlarge	x86_64
m7i.metal-24xl	x86_64
m7i.metal-48xl	x86_64
m8a.16xlarge	x86_64
m8a.24xlarge	x86_64
m8a.48xlarge	x86_64
m8a.metal-24xl	x86_64
m8a.metal-48xl	x86_64
m8azn.12xlarge	x86_64
m8azn.24xlarge	x86_64
m8azn.metal-12xl	x86_64
m8azn.metal-24xl	x86_64

实例类型	架构
m8g.8xlarge	arm64
m8g.12xlarge	arm64
m8g.16xlarge	arm64
m8g.24xlarge	arm64
m8g.48xlarge	arm64
m8g.metal-24xl	arm64
m8g.metal-48xl	arm64
m8gb.4xlarge	arm64
m8gb.8xlarge	arm64
m8gb.12xlarge	arm64
m8gb.16xlarge	arm64
m8gb.24xlarge	arm64
m8gb.48xlarge	arm64
m8gb.metal-24xl	arm64
m8gb.metal-48xl	arm64
m8gd.8xlarge	arm64
m8gd.12xlarge	arm64
m8gd.16xlarge	arm64
m8gd.24xlarge	arm64
m8gd.48xlarge	arm64

实例类型	架构
m8gd.metal-24xl	arm64
m8gd.metal-48xl	arm64
m8gn.4xlarge	arm64
m8gn.8xlarge	arm64
m8gn.12xlarge	arm64
m8gn.16xlarge	arm64
m8gn.24xlarge	arm64
m8gn.48xlarge	arm64
m8gn.metal-24xl	arm64
m8gn.metal-48xl	arm64
m8i.8xlarge	x86_64
m8i.12xlarge	x86_64
m8i.16xlarge	x86_64
m8i.24xlarge	x86_64
m8i.32xlarge	x86_64
m8i.48xlarge	x86_64
m8i.96xlarge	x86_64
m8i.metal-48xl	x86_64
m8i.metal-96xl	x86_64
m8id.8xlarge	x86_64

实例类型	架构
m8id.12xlarge	x86_64
m8id.16xlarge	x86_64
m8id.24xlarge	x86_64
m8id.32xlarge	x86_64
m8id.48xlarge	x86_64
m8id.96xlarge	x86_64
m8id.metal-48xl	x86_64
m8id.metal-96xl	x86_64
m8in.8xlarge	x86_64
m8in.12xlarge	x86_64
m8in.16xlarge	x86_64
m8in.24xlarge	x86_64
m8in.32xlarge	x86_64
m8in.48xlarge	x86_64
m8in.96xlarge	x86_64
m8in.metal-48xl	x86_64
m8in.metal-96xl	x86_64
m8idn.8xlarge	x86_64
m8idn.12xlarge	x86_64
m8idn.16xlarge	x86_64

实例类型	架构
m8idn.24xlarge	x86_64
m8idn.32xlarge	x86_64
m8idn.48xlarge	x86_64
m8idn.96xlarge	x86_64
m8idn.metal-48xl	x86_64
m8idn.metal-96xl	x86_64
m8ib.8xlarge	x86_64
m8ib.12xlarge	x86_64
m8ib.16xlarge	x86_64
m8ib.24xlarge	x86_64
m8ib.32xlarge	x86_64
m8ib.48xlarge	x86_64
m8ib.96xlarge	x86_64
m8ib.metal-48xl	x86_64
m8ib.metal-96xl	x86_64
m8idb.8xlarge	x86_64
m8idb.12xlarge	x86_64
m8idb.16xlarge	x86_64
m8idb.24xlarge	x86_64
m8idb.32xlarge	x86_64

实例类型	架构
m8idb.48xlarge	x86_64
m8idb.96xlarge	x86_64
m8idb.metal-48xl	x86_64
m8idb.metal-96xl	x86_64
m9g.8xlarge	arm64
m9g.12xlarge	arm64
m9g.16xlarge	arm64
m9g.24xlarge	arm64
m9g.48xlarge	arm64
m9g.metal-48xl	arm64
m9gd.8xlarge	arm64
m9gd.12xlarge	arm64
m9gd.16xlarge	arm64
m9gd.24xlarge	arm64
m9gd.48xlarge	arm64
m9gd.metal-48xl	arm64

Compute optimized

实例类型	架构
c6a.12xlarge	x86_64

实例类型	架构
c6a.16xlarge	x86_64
c6a.24xlarge	x86_64
c6a.32xlarge	x86_64
c6a.48xlarge	x86_64
c6a.metal	x86_64
c6gn.4xlarge	arm64
c6gn.8xlarge	arm64
c6gn.12xlarge	arm64
c6gn.16xlarge	arm64
c6i.8xlarge	x86_64
c6i.12xlarge	x86_64
c6i.16xlarge	x86_64
c6i.24xlarge	x86_64
c6i.32xlarge	x86_64
c6i.metal	x86_64
c6id.8xlarge	x86_64
c6id.12xlarge	x86_64
c6id.16xlarge	x86_64
c6id.24xlarge	x86_64
c6id.32xlarge	x86_64

实例类型	架构
c6id.metal	x86_64
c6in.8xlarge	x86_64
c6in.12xlarge	x86_64
c6in.16xlarge	x86_64
c6in.24xlarge	x86_64
c6in.32xlarge	x86_64
c6in.metal	x86_64
c7a.12xlarge	x86_64
c7a.16xlarge	x86_64
c7a.24xlarge	x86_64
c7a.32xlarge	x86_64
c7a.48xlarge	x86_64
c7a.metal-48xl	x86_64
c7g.12xlarge	arm64
c7g.16xlarge	arm64
c7g.metal	arm64
c7gd.12xlarge	arm64
c7gd.16xlarge	arm64
c7gd.metal	arm64
c7gn.4xlarge	arm64

实例类型	架构
c7gn.8xlarge	arm64
c7gn.12xlarge	arm64
c7gn.16xlarge	arm64
c7gn.metal	arm64
c7i.12xlarge	x86_64
c7i.16xlarge	x86_64
c7i.24xlarge	x86_64
c7i.48xlarge	x86_64
c7i.metal-24xl	x86_64
c7i.metal-48xl	x86_64
c8a.16xlarge	x86_64
c8a.24xlarge	x86_64
c8a.48xlarge	x86_64
c8a.metal-24xl	x86_64
c8a.metal-48xl	x86_64
c8g.8xlarge	arm64
c8g.12xlarge	arm64
c8g.16xlarge	arm64
c8g.24xlarge	arm64
c8g.48xlarge	arm64

实例类型	架构
c8g.metal-24xl	arm64
c8g.metal-48xl	arm64
c8gb.4xlarge	arm64
c8gb.8xlarge	arm64
c8gb.12xlarge	arm64
c8gb.16xlarge	arm64
c8gb.24xlarge	arm64
c8gb.48xlarge	arm64
c8gb.metal-24xl	arm64
c8gb.metal-48xl	arm64
c8gd.8xlarge	arm64
c8gd.12xlarge	arm64
c8gd.16xlarge	arm64
c8gd.24xlarge	arm64
c8gd.48xlarge	arm64
c8gd.metal-24xl	arm64
c8gd.metal-48xl	arm64
c8gn.4xlarge	arm64
c8gn.8xlarge	arm64
c8gn.12xlarge	arm64

实例类型	架构
c8gn.16xlarge	arm64
c8gn.24xlarge	arm64
c8gn.48xlarge	arm64
c8gn.metal-24xl	arm64
c8gn.metal-48xl	arm64
c8i.8xlarge	x86_64
c8i.12xlarge	x86_64
c8i.16xlarge	x86_64
c8i.24xlarge	x86_64
c8i.32xlarge	x86_64
c8i.48xlarge	x86_64
c8i.96xlarge	x86_64
c8i.metal-48xl	x86_64
c8i.metal-96xl	x86_64
c8id.8xlarge	x86_64
c8id.12xlarge	x86_64
c8id.16xlarge	x86_64
c8id.24xlarge	x86_64
c8id.32xlarge	x86_64
c8id.48xlarge	x86_64

实例类型	架构
c8id.96xlarge	x86_64
c8id.metal-48xl	x86_64
c8id.metal-96xl	x86_64
c8in.8xlarge	x86_64
c8in.12xlarge	x86_64
c8in.16xlarge	x86_64
c8in.24xlarge	x86_64
c8in.32xlarge	x86_64
c8in.48xlarge	x86_64
c8in.96xlarge	x86_64
c8in.metal-48xl	x86_64
c8in.metal-96xl	x86_64
c8ib.8xlarge	x86_64
c8ib.12xlarge	x86_64
c8ib.16xlarge	x86_64
c8ib.24xlarge	x86_64
c8ib.32xlarge	x86_64
c8ib.48xlarge	x86_64
c8ib.96xlarge	x86_64
c8ib.metal-48xl	x86_64

实例类型	架构
c8ib.metal-96xl	x86_64

Memory optimized

实例类型	架构
r6a.12xlarge	x86_64
r6a.16xlarge	x86_64
r6a.24xlarge	x86_64
r6a.32xlarge	x86_64
r6a.48xlarge	x86_64
r6a.metal	x86_64
r6i.8xlarge	x86_64
r6i.12xlarge	x86_64
r6i.16xlarge	x86_64
r6i.24xlarge	x86_64
r6i.32xlarge	x86_64
r6i.metal	x86_64
r6id.8xlarge	x86_64
r6id.12xlarge	x86_64
r6id.16xlarge	x86_64
r6id.24xlarge	x86_64

实例类型	架构
r6id.32xlarge	x86_64
r6id.metal	x86_64
r6idn.8xlarge	x86_64
r6idn.12xlarge	x86_64
r6idn.16xlarge	x86_64
r6idn.24xlarge	x86_64
r6idn.32xlarge	x86_64
r6idn.metal	x86_64
r6in.8xlarge	x86_64
r6in.12xlarge	x86_64
r6in.16xlarge	x86_64
r6in.24xlarge	x86_64
r6in.32xlarge	x86_64
r6in.metal	x86_64
r7a.12xlarge	x86_64
r7a.16xlarge	x86_64
r7a.24xlarge	x86_64
r7a.32xlarge	x86_64
r7a.48xlarge	x86_64
r7a.metal-48xl	x86_64

实例类型	架构
r7g.12xlarge	arm64
r7g.16xlarge	arm64
r7g.metal	arm64
r7gd.12xlarge	arm64
r7gd.16xlarge	arm64
r7gd.metal	arm64
r7i.12xlarge	x86_64
r7i.16xlarge	x86_64
r7i.24xlarge	x86_64
r7i.48xlarge	x86_64
r7i.metal-24xl	x86_64
r7i.metal-48xl	x86_64
r7iz.8xlarge	x86_64
r7iz.12xlarge	x86_64
r7iz.16xlarge	x86_64
r7iz.32xlarge	x86_64
r7iz.metal-16xl	x86_64
r7iz.metal-32xl	x86_64
r8a.16xlarge	x86_64
r8a.24xlarge	x86_64

实例类型	架构
r8a.48xlarge	x86_64
r8a.metal-24xl	x86_64
r8a.metal-48xl	x86_64
r8g.8xlarge	arm64
r8g.12xlarge	arm64
r8g.16xlarge	arm64
r8g.24xlarge	arm64
r8g.48xlarge	arm64
r8g.metal-24xl	arm64
r8g.metal-48xl	arm64
r8gb.4xlarge	arm64
r8gb.8xlarge	arm64
r8gb.12xlarge	arm64
r8gb.16xlarge	arm64
r8gb.24xlarge	arm64
r8gb.48xlarge	arm64
r8gb.metal-24xl	arm64
r8gb.metal-48xl	arm64
r8gd.8xlarge	arm64
r8gd.12xlarge	arm64

实例类型	架构
r8gd.16xlarge	arm64
r8gd.24xlarge	arm64
r8gd.48xlarge	arm64
r8gd.metal-24xl	arm64
r8gd.metal-48xl	arm64
r8gn.4xlarge	arm64
r8gn.8xlarge	arm64
r8gn.12xlarge	arm64
r8gn.16xlarge	arm64
r8gn.24xlarge	arm64
r8gn.48xlarge	arm64
r8gn.metal-24xl	arm64
r8gn.metal-48xl	arm64
r8i.8xlarge	x86_64
r8i.12xlarge	x86_64
r8i.16xlarge	x86_64
r8i.24xlarge	x86_64
r8i.32xlarge	x86_64
r8i.48xlarge	x86_64
r8i.96xlarge	x86_64

实例类型	架构
r8i.metal-48xl	x86_64
r8i.metal-96xl	x86_64
r8id.8xlarge	x86_64
r8id.12xlarge	x86_64
r8id.16xlarge	x86_64
r8id.24xlarge	x86_64
r8id.32xlarge	x86_64
r8id.48xlarge	x86_64
r8id.96xlarge	x86_64
r8id.metal-48xl	x86_64
r8id.metal-96xl	x86_64
r8in.8xlarge	x86_64
r8in.12xlarge	x86_64
r8in.16xlarge	x86_64
r8in.24xlarge	x86_64
r8in.32xlarge	x86_64
r8in.48xlarge	x86_64
r8in.96xlarge	x86_64
r8in.metal-48xl	x86_64
r8in.metal-96xl	x86_64

实例类型	架构
r8idn.8xlarge	x86_64
r8idn.12xlarge	x86_64
r8idn.16xlarge	x86_64
r8idn.24xlarge	x86_64
r8idn.32xlarge	x86_64
r8idn.48xlarge	x86_64
r8idn.96xlarge	x86_64
r8idn.metal-48xl	x86_64
r8idn.metal-96xl	x86_64
r8ib.8xlarge	x86_64
r8ib.12xlarge	x86_64
r8ib.16xlarge	x86_64
r8ib.24xlarge	x86_64
r8ib.32xlarge	x86_64
r8ib.48xlarge	x86_64
r8ib.96xlarge	x86_64
r8ib.metal-48xl	x86_64
r8ib.metal-96xl	x86_64
r8idb.8xlarge	x86_64
r8idb.12xlarge	x86_64

实例类型	架构
r8idb.16xlarge	x86_64
r8idb.24xlarge	x86_64
r8idb.32xlarge	x86_64
r8idb.48xlarge	x86_64
r8idb.96xlarge	x86_64
r8idb.metal-48xl	x86_64
r8idb.metal-96xl	x86_64
u7i-6tb.112xlarge	x86_64
u7i-8tb.112xlarge	x86_64
u7i-12tb.224xlarge	x86_64
u7in-16tb.224xlarge	x86_64
u7in-24tb.224xlarge	x86_64
u7in-32tb.224xlarge	x86_64
u7inh-32tb.480xlarge	x86_64
x2idn.16xlarge	x86_64
x2idn.24xlarge	x86_64
x2idn.32xlarge	x86_64
x2idn.metal	x86_64
x2iedn.8xlarge	x86_64
x2iedn.16xlarge	x86_64

实例类型	架构
x2iedn.24xlarge	x86_64
x2iedn.32xlarge	x86_64
x2iedn.metal	x86_64
x8g.8xlarge	arm64
x8g.12xlarge	arm64
x8g.16xlarge	arm64
x8g.24xlarge	arm64
x8g.48xlarge	arm64
x8g.metal-24xl	arm64
x8g.metal-48xl	arm64
x8aedz.24xlarge	x86_64
x8aedz.metal-24xl	x86_64
x8i.8xlarge	x86_64
x8i.12xlarge	x86_64
x8i.16xlarge	x86_64
x8i.24xlarge	x86_64
x8i.32xlarge	x86_64
x8i.48xlarge	x86_64
x8i.64xlarge	x86_64
x8i.96xlarge	x86_64

实例类型	架构
x8i.metal-48xl	x86_64
x8i.metal-96xl	x86_64

Accelerated computing

实例类型	架构
g6.48xlarge	x86_64
g6e.12xlarge	x86_64
g6e.24xlarge	x86_64
g6e.48xlarge	x86_64
g7e.12xlarge	x86_64
g7e.24xlarge	x86_64
g7e.48xlarge	x86_64
p5.4xlarge	x86_64
p5.48xlarge	x86_64
p5e.48xlarge	x86_64
p5en.48xlarge	x86_64
p6-b200.48xlarge	x86_64
p6-b300.48xlarge	x86_64

Storage optimized

实例类型	架构
i4g.4xlarge	arm64
i4g.8xlarge	arm64
i4g.16xlarge	arm64
i4i.8xlarge	x86_64
i4i.12xlarge	x86_64
i4i.16xlarge	x86_64
i4i.24xlarge	x86_64
i4i.32xlarge	x86_64
i4i.metal	x86_64
i7i.12xlarge	x86_64
i7i.16xlarge	x86_64
i7i.24xlarge	x86_64
i7i.48xlarge	x86_64
i7i.metal-24xl	x86_64
i7i.metal-48xl	x86_64
i7ie.6xlarge	x86_64
i7ie.12xlarge	x86_64
i7ie.18xlarge	x86_64
i7ie.24xlarge	x86_64

实例类型	架构
i7ie.48xlarge	x86_64
i7ie.metal-24xl	x86_64
i7ie.metal-48xl	x86_64
i8g.8xlarge	arm64
i8g.12xlarge	arm64
i8g.16xlarge	arm64
i8g.24xlarge	arm64
i8g.48xlarge	arm64
i8g.metal-24xl	arm64
i8g.metal-48xl	arm64
i8ge.6xlarge	arm64
i8ge.12xlarge	arm64
i8ge.18xlarge	arm64
i8ge.24xlarge	arm64
i8ge.48xlarge	arm64
i8ge.metal-24xl	arm64
i8ge.metal-48xl	arm64
im4gn.4xlarge	arm64
im4gn.8xlarge	arm64
im4gn.16xlarge	arm64

调整 Linux 实例上 ENA Express 设置的性能

为确保 ENA Express 能够有效运行，您的 Linux 实例必须满足多项网络配置要求。

您可以从 Amazon GitHub 存储库下载并运行 ENA Express 设置检查脚本，不必手动配置各项设置。该脚本可根据 ENA Express 的必要设置和建议设置来验证您的实例，并输出确切的命令以修复脚本发现的任何问题。

https://github.com/amzn/amzn-ec2-ena-utilities/blob/main/ena-express/check-ena-express-settings.sh

该脚本可检查以下设置和配置：

- MTU 大小：ENA Express 需要 MTU 低于默认值才能容纳额外的 Amazon SRD 标头。新建立的 TCP 连接会自动限制 MSS 以缓解此问题，但 UDP 流量仍需要较低的 MTU。
- TCP 输出队列大小限制：检查每个套接字的传输中字节限制是否足以维持高吞吐量。网络延迟较高的环境需要更高的限制。
- 字节队列限制：确认是否在网络接口上禁用了字节队列限制（BQL）。BQL 会限制排队等候设备级传输的数据量，从而限制 ENA Express 的性能。

> **Note**
>
> 默认情况下，Amazon Linux 发行版的 ENA 驱动程序会禁用字节队列限制。

- TCP 自动阻塞：检查 TCP 自动阻塞是否已禁用。禁用自动阻塞可以减少某些 ENA Express TCP 流量模式（例如请求-响应工作负载）的延迟。这可能会导致数据包处理开销增加幅度很小。
- TX 队列大小和大型低延迟队列：验证网络接口的发送队列大小是否足以实现最佳性能。该脚本还会检查 ENA 模块参数是否明确禁用大型低延迟队列（Large LLQ）功能，因为该功能会减少可用的 TX 队列深度。有关大型低延迟队列及其对 TX 队列大小的影响的更多信息，请参阅 GitHub 上的 Large Low Latency Queue (Large LLQ)。
- RX 队列大小：检查网络接口的接收环形缓冲区大小是否足以高效处理传入流量并避免在负载下丢包。
- TCP 和网络套接字缓冲区大小：验证 TCP 接收和发送缓冲区的最大大小，以及核心网络套接字缓冲区的默认值和最大值是否足以维持高吞吐量。这些设置在网络延迟较高的环境中非常重要，在这种环境中，需要更大的缓冲区才能充分利用连接。
- TCP 拥塞控制：验证 TCP 拥塞控制配置是否经过优化，可在网络延迟较高的环境中与 ENA Express 配合使用。

该脚本还会报告其他诊断信息，包括 ENA 驱动程序版本、ENA SRD 统计信息、中断裁决设置、队列配置和套接字缓冲区大小。此信息对于 ENA Express 性能的问题排查非常有用。

为确保您的实例网络驱动程序配置为最佳性能，另请参阅 GitHub 上的 ENA Linux Driver Best Practices and Performance Optimization Guide。

查看 EC2 实例的 ENA Express 设置

可以通过实例或网络接口验证 ENA Express 设置。要更新 ENA Express 设置，请参阅the section called “配置实例设置”。

Console

查看网络接口的 ENA Express 设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Network interfaces（网络接口）。
3. 选择一个网络接口以查看该实例的详细信息。您可以选择 Network interface ID（网络接口 ID）链接以打开详情页面，也可以选择列表左侧的复选框，在页面底部的详情窗格中查看详细信息。
4. 在详细信息选项卡或详情页面的网络接口附件部分中，查看 ENA Express 和 ENA Express UDP 的设置。

查看实例的 ENA Express 设置

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择实例以查看该实例的详细信息。您可以选择 Instance ID（实例类型）链接以打开详情页面，也可以选择列表左侧的复选框，在页面底部的详情窗格中查看详细信息。
4. 在 Networking（网络）选项卡的 Network interfaces（网络接口）部分中，向右滚动查看 ENA Express 和 ENA Express UDP 的设置。

Amazon CLI

获取实例的 ENA Express 设置

使用 describe-instances 命令。此命令示例将为 `--instance-ids` 参数所指定每个正在运行的实例连接的网络接口返回 ENA Express 配置列表。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7 \
    --query 'Reservations[*].Instances[*].[InstanceId,
 NetworkInterfaces[*].Attachment.EnaSrdSpecification]'
```

下面是示例输出。

```
[
    [
        [
            "i-1234567890abcdef0",
            [
                {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ],
    [
        [
            "i-0598c7d356eba48d7",
            [
            {
                    "EnaSrdEnabled": true,
                    "EnaSrdUdpSpecification": {
                        "EnaSrdUdpEnabled": false
                    }
                }
            ]
        ]
    ]
]
```

获取网络接口的 ENA Express 设置

使用 describe-network-interfaces 命令。

```
aws ec2 describe-network-interfaces \
    --network-interface-ids eni-1234567890abcdef0 \
```

```
    --query NetworkInterfaces[].[NetworkInterfaceId,Attachment.EnaSrdSpecification]
```

下面是示例输出。

```
[
    [
        "eni-1234567890abcdef0",
        {
            "EnaSrdEnabled": true,
            "EnaSrdUdpSpecification": {
                "EnaSrdUdpEnabled": false
            }
        }
    ]
]
```

PowerShell

获取网络接口的 ENA Express 设置

使用 Get-EC2NetworkInterface cmdlet。

```
Get-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    Association,
    NetworkInterfaceId,
    OwnerId,
    @{Name = 'AttachTime'; Expression = { $_.Attachment.AttachTime } },
    @{Name = 'AttachmentId'; Expression = { $_.Attachment.AttachmentId } },
    @{Name = 'DeleteOnTermination'; Expression =
 { $_.Attachment.DeleteOnTermination } },
    @{Name = 'NetworkCardIndex'; Expression = { $_.Attachment.NetworkCardIndex } },
    @{Name = 'InstanceId'; Expression = { $_.Attachment.InstanceId } },
    @{Name = 'InstanceOwnerId'; Expression = { $_.Attachment.InstanceOwnerId } },
    @{Name = 'Status'; Expression = { $_.Attachment.Status } },
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled } },
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled } }
```

下面是示例输出。

```
Association         :
NetworkInterfaceId  : eni-0d1234e5f6a78901b
OwnerId             : 111122223333
AttachTime          : 6/11/2022 1:13:11 AM
AttachmentId        : eni-attach-0d1234e5f6a78901b
DeleteOnTermination : True
NetworkCardIndex    : 0
InstanceId          : i-1234567890abcdef0
InstanceOwnerId     : 111122223333
Status              : attached
EnaSrdEnabled       : True
EnaSrdUdpEnabled    : False
```

为 EC2 实例配置 ENA Express 设置

您可为支持的 EC2 实例类型配置 ENA Express，而无需安装任何其他软件。有关更多信息，请参阅 the section called “ENA Express 支持的实例类型”。

Console

管理网络接口的 ENA Express

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Network interfaces（网络接口）。
3. 选择附加到实例的网络接口。您可以选择 Network interface ID（网络接口 ID）链接以打开详情页面，也可以选择列表左侧的复选框。
4. 从页面右上角的 Action（操作）菜单中选择 Manage ENA Express（管理 ENA Express）。这样将打开 Manage ENA Express（管理 ENA Express）对话框，显示选定的网络接口 ID 和当前设置。

   如果您选择的网络接口未附加到实例，则菜单中不会出现此操作。
5. 要使用 ENA Express，请选择启用复选框。
6. 启用 ENA Express 后，您可以配置 UDP 设置。要使用 ENA Express UDP，请选择启用复选框。
7. 选择保存以保存您的设置。

管理实例的 ENA Express

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择想要管理的实例。您可以选择 Instance ID（实例 ID）以打开详情页面，也可以选择列表左侧的复选框。
4. 选择要为您的实例配置的 Network interface（网络接口）。
5. 从页面右上角的 Action（操作）菜单中选择 Manage ENA Express（管理 ENA Express）。
6. 要为附加到实例的网络接口配置 ENA Express，请从 Network interface（网络接口）列表中进行选择。
7. 要对选定的网络接口附件使用 ENA Express，请选择启用复选框。
8. 启用 ENA Express 后，您可以配置 UDP 设置。要使用 ENA Express UDP，请选择启用复选框。
9. 选择保存以保存您的设置。

挂载网络接口时配置 ENA Express

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Network interfaces（网络接口）。
3. 选择未附加到实例的网络接口 [Status（状态）为可用]。您可以选择 Network interface ID（网络接口 ID）链接以打开详情页面，也可以选择列表左侧的复选框。
4. 选择要附加到的 Instance（实例）。
5. 将网络接口附加到实例后要使用 ENA Express，请选择启用复选框。
6. 启用 ENA Express 后，您可以配置 UDP 设置。要使用 ENA Express UDP，请选择启用复选框。
7. 要将网络接口附加到实例并保存 ENA Express 设置，请选择 Attach（附加）。

Amazon CLI

挂载网络接口时配置 ENA Express

使用 attach-network-interface 命令，如以下示例所示。

示例 1：将 ENA Express 用于 TCP 流量，而非 UDP 流量

此示例将 EnaSrdEnabled 配置为 true，且允许 EnaSrdUdpEnabled 默认为 false。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

示例 2：将 ENA Express 用于 TCP 流量和 UDP 流量

此示例将 EnaSrdEnabled 和 EnaSrdUdpEnabled 均配置为 true。

```
aws ec2 attach-network-interface \
    --network-interface-id eni-1234567890abcdef0 \
    --instance-id i-1234567890abcdef0 \
    --device-index 1 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

更新网络接口挂载的 ENA Express 设置

使用 modify-network-interface-attribute 命令，如以下示例所示。

示例 1：将 ENA Express 用于 TCP 流量，而非 UDP 流量

此示例将 EnaSrdEnabled 配置为 true，并且如果之前从未设置过，则允许 EnaSrdUdpEnabled 默认为 false。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdEnabled=true'
```

示例 2：将 ENA Express 用于 TCP 流量和 UDP 流量

此示例将 EnaSrdEnabled 和 EnaSrdUdpEnabled 均配置为 true。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification
 'EnaSrdEnabled=true,EnaSrdUdpSpecification={EnaSrdUdpEnabled=true}'
```

示例 3：停止将 ENA Express 用于 UDP 流量

此示例将 EnaSrdUdpEnabled 配置为 false。

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-1234567890abcdef0 \
    --ena-srd-specification 'EnaSrdUdpSpecification={EnaSrdUdpEnabled=false}'
```

PowerShell

挂载网络接口时配置 ENA Express

使用 Add-EC2NetworkInterface cmdlet，如以下示例中所示。

示例 1：将 ENA Express 用于 TCP 流量，而非 UDP 流量

此示例将 EnaSrdEnabled 配置为 true，且允许 EnaSrdUdpEnabled 默认为 false。

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true
```

示例 2：将 ENA Express 用于 TCP 流量和 UDP 流量

此示例将 EnaSrdEnabled 和 EnaSrdUdpEnabled 均配置为 true。

```
Add-EC2NetworkInterface `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -DeviceIndex 1 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdUdpSpecification_EnaSrdUdpEnabled $true
```

配置网络接口挂载的 ENA Express 设置

使用 Edit-EC2NetworkInterfaceAttribute cmdlet，如以下示例中所示。

示例 1：将 ENA Express 用于 TCP 流量，而非 UDP 流量

此示例将 EnaSrdEnabled 配置为 true，并且如果之前从未设置过，则允许 EnaSrdUdpEnabled 默认为 false。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

示例 2：将 ENA Express 用于 TCP 流量和 UDP 流量

此示例将 EnaSrdEnabled 和 EnaSrdUdpEnabled 均配置为 true。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-1234567890abcdef0 `
    -EnaSrdSpecification_EnaSrdEnabled $true `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $true ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-1234567890abcdef0 | `
Select-Object `
    NetworkInterfaceId,
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

示例 3：停止将 ENA Express 用于 UDP 流量

此示例将 EnaSrdUdpEnabled 配置为 false。

```
Edit-EC2NetworkInterfaceAttribute `
    -NetworkInterfaceId eni-0123f4567890a1b23 `
    -EnaSrdSpecification_EnaSrdUdpSpecification_EnaSrdUdpEnabled $false ;
Get-EC2NetworkInterface -NetworkInterfaceId eni-0123f4567890a1b23 | `
Select-Object `
    NetworkInterfaceId,
```

```
    @{Name = 'EnaSrdEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdEnabled }},
    @{Name = 'EnaSrdUdpEnabled'; Expression =
 { $_.Attachment.EnaSrdSpecification.EnaSrdUdpSpecification.EnaSrdUdpEnabled }} | `
Format-List
```

启动时配置 ENA Express

启动实例时，您可以使用以下方法之一直接配置 ENA Express。指定的链接会提供关于这些方法的 Amazon Web Services 管理控制台 说明。

- 启动实例向导：使用启动实例向导时，可以在启动时配置 ENA Express。有关详细信息，请参阅启动实例向导网络设置中的高级网络配置。
- 启动模板：使用启动模板时，可以在启动时配置 ENA Express。有关更多信息，请参阅创建 Amazon EC2 启动模板页面，然后展开网络设置部分，从中查看高级网络配置。

通过 Intel 82599 VF 接口实现增强联网

对于基于 Xen 的实例，Intel 82599 虚拟功能（VF）接口会提供增强的联网功能。该接口使用 Intel `ixgbevf` 驱动程序。

下列选项卡展示了如何验证为实例操作系统安装的网络适配器驱动程序。

Linux

Linux 网络接口驱动程序

使用以下命令验证是否在特定接口上使用了该模块（替代要检查的接口名称）。如果您使用单个接口（默认设置），则为 `eth0`。如果操作系统支持可预测的网络名称，这可以是 `ens5` 等名称。

在以下示例中，`ixgbevf` 模块未加载，因为列出的驱动程序是 `vif`。

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
```

```
supports-register-dump: no
supports-priv-flags: no
```

在此示例中，已加载 ixgbevf 模块。此实例正确配置了增强联网。

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 4.0.3
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

Windows

Windows 网络适配器

要验证是否已安装驱动程序，请连接到您的实例并打开设备管理器。您应该会在网络适配器下看到列出的 Intel(R) 82599 Virtual Function。

内容

- 为增强联网准备好实例
- 测试是否启用了增强联网功能
- 在实例上启用增强联网
- 对连接问题进行故障排除

为增强联网准备好实例

要使用 Intel 82599 VF 接口准备增强联网，请按如下方式设置您的实例：

- 验证实例类型是否为以下类型之一：C3、C4、D2、I2、M4（不包括 m4.16xlarge）和 R3。
- 确保实例具有 Internet 连接。
- 如果您的实例上有重要的数据需要保留，则应立即从您的实例创建 AMI，来备份这些数据。更新内核和内核模块以及启用 sriovNetSupport 属性可能会导致实例不兼容或无法访问操作系统。如果您有最新备份，则发生此情况时仍将保留数据。

- Linux 实例：从使用 Linux 内核版本 2.6.32 或更高版本的 HVM AMI 启动实例。最新的 Amazon Linux HVM AMI 安装有增强联网所需的模块，并已设置所需的属性。因此，如果使用最新 Amazon Linux HVM AMI 启动由 Amazon EBS 提供支持且支持增强联网的实例，则已为您的实例启用增强联网。

  > **Warning**
  >
  > 仅 HVM 实例支持增强联网。使用半虚拟化实例启用增强联网会让该实例无法访问。在模块或模块版本不正确的情况下设置此属性还可能导致您的实例不可访问。

- Windows 实例：从 64 位 HVM AMI 启动实例。您不能对 Windows Server 2008 启用增强联网。已针对 Windows Server 2012 R2 和 Windows Server 2016 及更高版本的 AMI 启用了增强联网功能。Windows Server 2012 R2 包含 Intel 驱动程序 1.0.15.3，我们建议您使用 Pnputil.exe 实用工具将该驱动程序升级到最新版本。
- 使用 Amazon Web Services 管理控制台的 Amazon CloudShell，或者将 Amazon CLI 或 Amazon Tools for Windows PowerShell 安装到您选择的任意电脑上（最好是本地台式机或笔记本电脑）并进行配置。有关的更多信息，请参阅 访问 Amazon EC2 或 Amazon CloudShell 用户指南。不能从 Amazon EC2 控制台管理增强联网。

测试是否启用了增强联网功能

确认已在实例或映像上设置了 `sriovNetSupport` 属性。

Amazon CLI

检查实例属性（sriovNetSupport）

使用以下 describe-instance-attribute 命令。如果设置了该属性，则值为 `simple`。

```
aws ec2 describe-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --attribute sriovNetSupport
```

检查映像属性（sriovNetSupport）

使用以下 describe-images 命令。如果设置了该属性，则值为 `simple`。

```
aws ec2 describe-images \
    --image-id ami-0abcdef1234567890 \
```

```
    --query "Images[].SriovNetSupport"
```

PowerShell

检查实例属性（sriovNetSupport）

使用 Get-EC2InstanceAttribute cmdlet。如果设置了该属性，则值为 `simple`。

```
Get-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Attribute sriovNetSupport
```

检查映像属性（sriovNetSupport）

使用以下 describe-images 命令。如果设置了该属性，则值为 `simple`。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).SriovNetSupport
```

在实例上启用增强联网

您使用的过程取决于实例的操作系统。

> **Warning**
> 增强联网属性启用之后将无法禁用。

Amazon Linux

最新的 Amazon Linux HVM AMI 安装有增强联网所需的 `ixgbevf` 模块，并已设置所需的 `sriovNetSupport` 属性。因此，如果使用最新的 Amazon Linux HVM AMI 启动实例类型，则已为您的实例启用增强联网。有关更多信息，请参阅测试是否启用了增强联网功能。

如果您使用较旧的 Amazon Linux AMI 启动了实例，并且实例尚未启用增强联网，请通过以下步骤启用增强联网。

启用增强联网

1. 连接到您的实例。
2. 从实例运行以下命令以使用最新内核和内核模块 (包括 `ixgbevf`) 更新实例：

```
[ec2-user ~]$ sudo yum update
```

3. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机重启实例：reboot-instances（Amazon CLI）或 Restart-EC2Instance（Amazon Tools for Windows PowerShell）。
4. 再次连接到您的实例，并使用ixgbevf中的 modinfo ixgbevf 命令验证 测试是否启用了增强联网功能 模块是否已安装并具有推荐的最低版本。
5. [由 EBS 支持的实例] 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机停止实例：stop-instances（Amazon CLI）或 Stop-EC2Instance（Amazon Tools for Windows PowerShell）。

   [实例存储支持的实例] 您无法停止实例来修改属性。请跳到下一步。
6. 使用以下任一命令从本地电脑启用增强联网属性：

   Amazon CLI

   按照如下所示使用 modify-instance-attribute 命令。

   ```
   aws ec2 modify-instance-attribute \
       --instance-id i-1234567890abcdef0 \
       --sriov-net-support simple
   ```

   PowerShell

   按如下方式使用 Edit-EC2InstanceAttribute cmdlet。

   ```
   Edit-EC2InstanceAttribute `
       -InstanceId i-1234567890abcdef0 `
       -SriovNetSupport "simple"
   ```
7. (可选) 从实例创建 AMI，如 中所述创建 Amazon EBS-backed AMI 该 AMI 从实例继承增强联网属性。因此，您可以使用此 AMI 启动默认情况下启用了增强联网功能的其他实例。
8. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机启动实例：start-instances（Amazon CLI）或 Start-EC2Instance（Amazon Tools for Windows PowerShell）。
9. 连接到您的实例，并使用ixgbevf中的 ethtool -i eth***n*** 命令验证是否在网络接口上安装并加载了 测试是否启用了增强联网功能 模块。

启用增强联网（由实例存储支持的实例）

按照上述过程操作，直到您停止实例的步骤。按照创建 Amazon S3 支持的 AMI中所述创建新 AMI，确保在注册 AMI 时启用增强联网属性。

Amazon CLI

按照如下所示使用 register-image 命令。

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

按如下方式使用 Register-EC2Image。

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Ubuntu

在开始之前，在您的实例上检查是否已启用增强联网。

Quick Start Ubuntu HVM AMI 包含实现增强联网所需的驱动程序。如果您的 ixgbevf 版本早于 2.16.4，则可以安装 linux-aws 内核程序包以获取最新增强的联网功能驱动程序。

以下过程提供了在 Ubuntu 实例上编译 ixgbevf 模块的一般步骤。

安装 **linux-aws** 内核程序包

1. 连接到您的 实例。
2. 更新包缓存和包。

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

> **Important**
>
> 如果在更新过程中系统提示您安装 grub，请使用 /dev/xvda 安装 grub，然后选择保留当前版本的 /boot/grub/menu.lst。

其他 Linux 分配

在开始之前，在您的实例上检查是否已启用增强联网。最新 Quick Start HVM AMI 包含实现增强联网所需的驱动程序，因此您无需执行其他步骤。

下面的过程提供了在 Amazon Linux 或 Ubuntu 之外的 Linux 发行版上启用 Intel 82599 VF 接口增强联网所需的一般步骤。有关更多信息 (如命令的详细语法、文件位置或包和工具支持)，请参阅您的 Linux 发行版的特定文档。

对 Linux 启用增强联网

1. 连接到您的 实例。
2. 从 Sourceforge 上的 ixgbevfhttps://sourceforge.net/projects/e1000/files/%20stable/ 位置在实例上下载 ixgbevf 模块的源代码。

   早于 2.16.4 的 ixgbevf 版本 (包括版本 2.14.2) 在某些 Linux 发行版 (包括某些 Ubuntu 版本) 上构建不正确。
3. 在实例上编译并安装 ixgbevf 模块。

   > **Warning**
   >
   > 如果您为当前内核编译 ixgbevf 模块，然后升级内核而不为新内核重新构建驱动程序，则系统会在下次重新启动时恢复为特定于发行版的 ixgbevf 模块。这可能会在特定于发行版的版本与增强联网不兼容时使您的系统无法访问。
4. 运行 sudo depmod 命令以更新模块依赖项。
5. 在实例上更新 initramfs 以确保在启动时加载新模块。
6. 确定您的系统是否默认使用可预测的网络接口名称。使用 systemd 或 udev 版本 197 或更高版本的系统可以重命名以太网设备，它们不保证单个网络接口将命名为 eth0。此行为可能导致连接到实例时出现问题。要获取更多信息并查看其他配置选项，请参阅 freedesktop.org 网站上的可预测的网络接口名称。

   a. 您可以使用以下命令在基于 RPM 的系统上检查 systemd 或 udev 版本：

   ```
   [ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'
       systemd-208-11.el7_0.2.x86_64
   ```

   在以上 Red Hat Enterprise Linux 7 示例中，systemd 版本是 208，因此必须禁用可预测的网络接口名称。

b. 通过将 net.ifnames=0 选项添加到 GRUB_CMDLINE_LINUX 中的 /etc/default/grub 行，可禁用可预测的网络接口名称。

```
[ec2-user ~]$ sudo sed -i '/^GRUB\_CMDLINE\_LINUX/s/\"$/\ net\.ifnames\=0\"/' /
etc/default/grub
```

c. 重新构建 grub 配置文件。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [由 EBS 支持的实例] 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机停止实例：stop-instances（Amazon CLI）或 Stop-EC2Instance（Amazon Tools for Windows PowerShell）。

   [实例存储支持的实例] 您无法停止实例来修改属性。请跳到下一步。

8. 使用以下任一命令从本地电脑启用增强联网属性：

   Amazon CLI

   按照如下所示使用 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 -\
    -sriov-net-support simple
```

   PowerShell

   按如下方式使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

9. (可选) 从实例创建 AMI，如 中所述创建 Amazon EBS-backed AMI 该 AMI 从实例继承增强联网属性。因此，您可以使用此 AMI 启动默认情况下启用了增强联网功能的其他实例。

   如果您的实例操作系统包含 /etc/udev/rules.d/70-persistent-net.rules 文件，则必须在创建 AMI 之前将其删除。此文件包含原始实例的以太网适配器 MAC 地址。如果其他实例使用此文件启动，操作系统将找不到设备，eth0 会失败，从而导致启动问题。此文件将在下次启动过程中重新生成，从 AMI 启动的任意实例都会创建这个文件的自有版本。

10. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机启动实例：start-instances（Amazon CLI）或 Start-EC2Instance（Amazon Tools for Windows PowerShell）。
11. (可选) 连接到实例并确认已安装模块。

启用增强联网（由实例存储支持的实例）

按照上述过程操作，直到您停止实例的步骤。按照创建 Amazon S3 支持的 AMI中所述创建新 AMI，确保在注册 AMI 时启用增强联网属性。

Amazon CLI

按照如下所示使用 register-image 命令。

```
aws ec2 register-image --sriov-net-support simple ...
```

PowerShell

按如下方式使用 Register-EC2Image。

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Windows

如果您启动了您的实例且该实例未启用增强联网，则必须下载所需的网络适配器驱动程序并将其安装到您的实例上，然后设置 `sriovNetSupport` 实例属性以激活增强联网。您只能对支持的实例类型启用此属性。有关更多信息，请参阅 Amazon EC2 实例上的增强联网功能。

启用增强联网

1. 连接到您的实例并以本地管理员身份登录。
2. [Windows Server 2016 及更高版本] 运行以下 EC2 Launch PowerShell 脚本，以便在安装此驱动程序后配置实例。

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -
Schedule
```

> **Important**
>
> 当您启用初始化实例 EC2 Launch 脚本时，管理员密码将重置。您可以修改配置文件，通过在初始化任务的设置中指定禁用管理员密码重置来禁用它。

3. 从实例中，为操作系统下载 Intel 网络适配器驱动程序：

   - Windows Server 2022

     访问下载页面并下载 `Wired_driver_version_x64.zip`。
   - Windows Server 2019（包括服务器版本 1809 及更高版本*）

     访问下载页面并下载 `Wired_driver_version_x64.zip`。
   - Windows Server 2016（包括服务器版本 1803 及更低版本*）

     访问下载页面并下载 `Wired_driver_version_x64.zip`。
   - Windows Server 2012 R2

     访问下载页面并下载 `Wired_driver_version_x64.zip`。
   - Windows Server 2012

     访问下载页面并下载 `Wired_driver_version_x64.zip`。
   - Windows Server 2008 R2

     访问下载页面并下载 `PROWinx64Legacy.exe`。

   * “Intel 驱动程序和软件”页面上未明确说明服务器版本 1803 及更低版本和 1809 及更高版本。

4. 为操作系统安装 Intel 网络适配器驱动程序。

   - Windows Server 2008 R2
     1. 在 Downloads (下载) 文件夹中，找到 `PROWinx64Legacy.exe` 文件，并将其重命名为 `PROWinx64Legacy.zip`。
     2. 提取 `PROWinx64Legacy.zip` 文件内容。
     3. 打开命令行，转到提取的文件夹，然后运行以下命令，以使用 `pnputil` 实用工具在驱动程序存储中添加和安装 INF 文件。

```
C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
```

- Windows Server 2022、Windows Server 2019、Windows Server 2016、Windows Server 2012 R2 和 Windows Server 2012

  1. 在 Downloads (下载) 文件夹中，提取 `Wired_driver_version_x64.zip` 文件内容。
  2. 打开命令行，转到提取的文件夹，然后运行以下命令之一，以使用 `pnputil` 实用工具在驱动程序存储中添加和安装 INF 文件。

     - Windows Server 2022

       ```
       pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
       ```

     - Windows Server 2019

       ```
       pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
       ```

     - Windows Server 2016

       ```
       pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
       ```

     - Windows Server 2012 R2

       ```
       pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
       ```

     - Windows Server 2012

       ```
       pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
       ```

5. 使用以下任一命令从本地电脑启用增强联网属性：

   Amazon CLI

   按照如下所示使用 modify-instance-attribute 命令。

   ```
   aws ec2 modify-instance-attribute \
       --instance-id i-1234567890abcdef0 \
       --sriov-net-support simple
   ```

PowerShell

按如下方式使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -SriovNetSupport "simple"
```

6. (可选) 从实例创建 AMI，如 中所述创建 Amazon EBS-backed AMI 该 AMI 从实例继承增强联网属性。因此，您可以使用此 AMI 启动默认情况下启用了增强联网功能的其他实例。
7. 使用 Amazon EC2 控制台或以下任一命令从您的本地计算机启动实例：start-instances（Amazon CLI）或 Start-EC2Instance（Amazon Tools for Windows PowerShell）。

对连接问题进行故障排除

如果您在启用增强联网期间丢失连接，则 `ixgbevf` 模块可能与内核不兼容。请尝试安装用于您的实例的 Linux 发行版所附带的 `ixgbevf` 模块版本。

如果您为半虚拟化实例或 AMI 启用增强网络，则这可能会使您的实例无法访问。

有关更多信息，请参阅如何在 EC2 实例上启用和配置增强联网？

监控 EC2 实例上 ENA 设置的网络性能

弹性网络适配器（ENA）驱动程序从启用这些指标的实例发布网络性能指标。您可以使用这些指标来排查实例性能问题、为工作负载选择合适的实例大小、主动计划扩展活动以及对应用程序进行基准测试，以确定它们是否最大限度地提高了实例的可用性能。

Amazon EC2 定义实例级别的网络最大值，以确保高质量的联网体验，包括不同大小实例间一致的网络性能。Amazon 为每个实例提供以下最大值：

- 带宽功能 – 根据实例类型和大小，每个 EC2 实例都具有聚合入站和出站流量的最大带宽。一些实例使用一种网络 I/O 积分机制，根据平均带宽使用率分配网络带宽。Amazon EC2 还拥有最大的带宽，可用于流向 Amazon Direct Connect 和互联网的流量。有关更多信息，请参阅 Amazon EC2 实例网络带宽。
- 每秒数据包 (PPS) 性能 – 根据实例类型和大小，每个 EC2 实例都具有最大的 PPS 性能。
- 跟踪连接 – 安全组会跟踪建立的每个连接，以确保返回数据包按预期交付。每个实例都有可以跟踪的最大连接数量。有关更多信息，请参阅 Amazon EC2 安全组连接跟踪。

- 链接本地服务访问 – 对于流向 Amazon DNS 服务、实例元数据服务和 Amazon Time Sync Service 等本地代理服务的流量，Amazon EC2 规定了每个网络接口的最大 PPS。

当实例的网络流量超过最大值时，Amazon 将通过排队然后丢弃网络数据包来调整超过最大值的流量。您可以使用网络性能指标监控流量何时超过最大值。这些指标可以实时告知您对网络流量的影响以及可能的网络性能问题。

内容

- 要求
- ENA 驱动程序的指标
- 查看实例的网络性能指标
- ENA Express 指标
- ENA 的 DPDK 驱动程序的网络性能指标
- 运行 FreeBSD 的实例的指标

要求

Linux 实例

- 安装 ENA 驱动程序版本 2.2.10 或更高版本。要验证安装的版本，请使用 ethtool 命令。在以下示例中，版本符合最低要求。

```
[ec2-user ~]$ ethtool -i eth0 | grep version
version: 2.2.10
```

  要升级 ENA 驱动程序，请参阅增强联网。
- 要将这些指标导入到 Amazon CloudWatch，请安装 CloudWatch 代理。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的收集网络性能指标。
- 要支持 `conntrack_allowance_available` 指标，请安装 ENA 驱动程序 2.8.1 或更高版本。
- 要覆盖 1024 的出口片段 PPS 限制，请安装 ENA 驱动程序 2.13.3 或更高版本。

Windows 实例

- 安装 ENA 驱动程序版本 2.2.2 或更高版本。要验证安装的版本，请按如下方式使用设备管理器。
  1. 通过运行 `devmgmt.msc` 打开设备管理器。

2. 展开 Network Adapters（网络适配器）。
3. 选择 Amazon Elastic Network Adapter、Properties（属性）。
4. 在 Driver（驱动程序）选项卡上，查找 Driver Version（驱动程序版本）。

要升级 ENA 驱动程序，请参阅增强联网。

- 要将这些指标导入到 Amazon CloudWatch，请安装 CloudWatch 代理。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的收集高级网络指标。

ENA 驱动程序的指标

ENA 驱动程序实时向实例传送以下指标。它们提供了自上次驱动程序重置以来在每个网络接口上排队或丢弃的累计数据包数。

指标	说明	在以下实例上受支持：
bw_in_allowance_exceeded	因入站聚合带宽超过实例的最大值而排队或丢弃的数据包的数量。	所有实例类型
bw_out_allowance_exceeded	因出站聚合带宽超过实例的最大值而排队或丢弃的数据包的数量。	所有实例类型
conntrack_allowance_exceeded	由于连接跟踪超过实例的最大值且无法建立新连接而丢弃的数据包的数量。这可能会导致进出实例的流量丢失数据包。	所有实例类型
conntrack_allowance_available	在达到该实例类型的跟踪连接限额之前，实例可以建立的跟踪连接数。	仅限基于 Nitro 的实例
linklocal_allowance_exceeded	由于到本地代理服务的流量的 PPS 超出网络接口的最大值而丢弃的数据包数量。这会影响流向 Amazon DNS 服务、实例元数据服务和 Amazon Time Sync	所有实例类型

指标	说明	在以下实例上受支持：
	Service 的流量，但不会影响流向自定义 DNS 解析器的流量。	
pps_allowance_exceeded	因双向 PPS 超过实例的最大值而排队或丢弃的数据包的数量。*	所有实例类型

*根据 ENA Linux 驱动程序 v2.13.3 或更高版本的片段代理模式设置，此限制还包括网络接口超过 1024 PPS 的出口片段丢弃。如果为 Linux 驱动程序启用了片段代理模式，则出口片段丢弃会绕过通常适用的 1024 PPS 限制，并计入标准 PPS 限额。片段代理模式默认处于禁用状态。

查看实例的网络性能指标

您使用的过程取决于实例的操作系统。

Linux 实例

您可以将指标发布到自己的收藏夹工具以可视化指标数据。例如，您可以使用 CloudWatch 代理将指标发布到 Amazon CloudWatch。代理使您能够选择单个指标并控制发布。

您还可以使用 ethtool 按如下方式检索每个网络接口的指标，例如 eth0。

```
[ec2-user ~]$ ethtool -S eth0
     bw_in_allowance_exceeded: 0
     bw_out_allowance_exceeded: 0
     pps_allowance_exceeded: 0
     conntrack_allowance_exceeded: 0
     linklocal_allowance_exceeded: 0
     conntrack_allowance_available: 136812
```

Windows 实例

您可以使用 Windows 性能计数器的任何使用者查看指标。数据可以根据 EnaPerfCounters 清单进行解析。这是一个定义性能计数器提供程序及其计数器集的 XML 文件。

安装清单

如果您已使用包含 ENA 驱动程序 2.2.2 或更高版本的 AMI 启动实例，或已使用 ENA 驱动程序 2.2.2 的驱动程序包中的安装脚本，则清单已安装。要手动安装清单，请使用以下步骤：

1. 使用以下命令删除现有清单：

```
unlodctr /m:EnaPerfCounters.man
```

2. 将清单文件 EnaPerfCounters.man 从驱动程序安装包复制到 %SystemRoot%\System32\drivers。
3. 使用以下命令安装新清单：

```
lodctr /m:EnaPerfCounters.man
```

使用性能监视器查看指标

1. 打开性能监视器。
2. 按下 Ctrl+N 添加新的计数器。
3. 从列表中选择 ENA Packets Shaping（ENA 数据包塑造）。
4. 选择要监控的实例，然后选择 Add（添加）。
5. 选择 OK。

ENA Express 指标

ENA Express 由 Amazon 可扩展的可靠数据报（SRD）技术提供支持。SRD 是一种高性能网络传输协议，使用动态路由来提高吞吐量并最大限度地减少尾部延迟。如果已为发送和接收实例上的网络接口附件启用 ENA Express，您可以使用 ENA Express 指标来帮助确保实例充分利用 SRD 技术所提供的性能改进。例如：

- 评估您的资源，确保其有足够的容量来建立更多的 SRD 连接。
- 确定哪些位置存在阻止符合条件的传出数据包使用 SRD 的潜在问题。
- 计算实例使用 SRD 的传出流量的百分比。
- 计算实例使用 SRD 的传入流量的百分比。

> **Note**
>
> 要生成指标，请使用驱动程序版本 2.8 或更高版本。

要查看针对 ENA Express 筛选的 Linux 实例指标列表，请对您的网络接口（此处显示为 eth0）运行以下 ethtool 命令。记下 ena_srd_mode 指标的值。

```
[ec2-user ~]$ ethtool -S eth0 | grep ena_srd
NIC statistics:
 ena_srd_mode: 1
 ena_srd_tx_pkts: 0
 ena_srd_eligible_tx_pkts: 0
 ena_srd_rx_pkts: 0
 ena_srd_resource_utilization: 0
```

以下指标适用于所有已启用 ENA Express 的实例。

ena_srd_mode

描述启用了哪些 ENA Express 功能。值如下所示：

- 0 = ENA Express 关闭，UDP 关闭
- 1 = ENA Express 开启，UDP 关闭
- 2 = ENA Express 关闭，UDP 开启

> **Note**
>
> 仅在最初启用 ENA Express 并将 UDP 配置为使用该功能时，才会发生这种情况。UDP 流量将保留先前值。

- 3 = ENA Express 开启，UDP 开启

ena_srd_eligible_tx_pkts

网络编号如下：

- 同时支持发送和接收实例类型。有关更多信息，请参阅 ENA Express 支持的实例类型 表。
- 发送和接收实例都必须配置 ENA Express。
- 发送和接收实例必须在同一区域中运行。
- 实例之间的网络路径不得包含中间件。ENA Express 目前不支持中间件。

> **Note**
>
> ENA Express 资格指标包括源和目标要求以及两个端点之间的网络。符合条件的数据包在已计数后仍可取消资格。例如，如果某个符合条件的数据包超过了最大传输单元（MTU）的限制，则尽管该数据包在计数器中仍显示为符合条件，但其仍会退回到标准 ENA 传输。

ena_srd_tx_pkts

指定时间段内传输的 SRD 数据包数量。

ena_srd_rx_pkts

指定时间段内接收的 SRD 数据包数量。

ena_srd_resource_utilization

实例已消耗的 SRD 并发连接的最大允许内存利用率的百分比。

要确认数据包传输是否使用 SRD，您可以将符合条件的数据包数量（`ena_srd_eligible_tx_pkts` 指标）与指定时间段内传输的 SRD 数据包数量（`ena_srd_tx_pkts` 指标）进行比较。

出口流量（传出数据包）

为确保您的出口流量按预期使用 SRD，请将符合条件的 SRD 数据包数量（`ena_srd_eligible_tx_pkts`）与指定时间段内发送的 SRD 数据包数量（`ena_srd_tx_pkts`）进行比较。

符合条件的数据包数量和发送的 SRD 数据包数量之间的显著差异通常是由资源利用率问题引起的。当附加到实例的网卡已用尽其最大资源时，或如果数据包超过 MTU 限制，则符合条件的数据包将无法通过 SRD 传输，必须退回到标准 ENA 传输。在实时迁移或实时服务器更新期间，数据包也可能出现这种问题。要确定根本原因，还需进行额外的故障排除。

> **Note**
>
> 您可以忽略符合条件的数据包数量与 SRD 数据包数量之间偶尔的细微差异。例如，当您的实例为 SRD 流量与另一个实例建立连接时，就会发生这种情况。

要找出在给定时间段内使用 SRD 的总出口流量百分比，请将该时间段内发送的 SRD 数据包数（`ena_srd_tx_pkts`）与针对该实例发送的数据包总数（`NetworkPacketOut`）进行比较。

入口流量（传入数据包）

要找出入口流量使用 SRD 的百分比，请将在给定时间段内接收的 SRD 数据包数（`ena_srd_rx_pkts`）与该时间段内针对该实例接收的数据包总数（`NetworkPacketIn`）进行比较。

资源利用率

资源利用率基于单个实例在指定时间可以容纳的 SRD 并发连接的数量。资源利用率指标（`ena_srd_resource_utilization`）会持续跟踪您当前对实例的利用率。当利用率接近 100% 时，可能会出现性能问题。ENA Express 从 SRD 退回到标准 ENA 传输，丢弃数据包的可能性也随之增加。高资源利用率表明是时候向外扩展实例以提高网络性能了。

> **Note**
>
> 当实例的网络流量超过最大值时，Amazon 将通过排队然后丢弃网络数据包来调整超过最大值的流量。

Persistence

为实例启用 ENA Express 时，会累积出口和入口指标。如果停用 ENA Express，指标就会停止累积，但只要实例仍在运行，指标就会一直存在。如果实例重启或终止，或网络接口与实例分离，则指标将重置。

ENA 的 DPDK 驱动程序的网络性能指标

ENA 驱动程序版本 2.2.0 及更高版本支持网络指标报告。DPDK 20.11 包括 ENA 驱动程序 2.2.0，它是第一个支持此功能的 DPDK 版本。

DPDK 驱动程序 v25.03 或更高版本支持片段代理模式。如果为 DPDK 驱动程序启用了片段代理模式，则出口片段丢弃会绕过通常适用的 1024 PPS 限制，并计入标准 PPS 限额。片段代理模式默认处于禁用状态。

您可以使用示例应用程序查看 DPDK 统计信息。要启动示例应用程序的交互式版本，请运行以下命令。

```
./app/dpdk-testpmd -- -i
```

在此交互式会话中，您可以输入命令来检索端口的扩展统计信息。以下示例命令将检索端口 0 的统计信息。

```
show port xstats 0
```

以下是与 DPDK 示例应用程序进行交互式会话的示例。

```
[root@ip-192.0.2.0 build]# ./app/dpdk-testpmd -- -i
        EAL: Detected 4 lcore(s)
        EAL: Detected 1 NUMA nodes
        EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
        EAL: Selected IOVA mode 'PA'
        EAL: Probing VFIO support...
        EAL:   Invalid NUMA socket, default to 0
        EAL:   Invalid NUMA socket, default to 0
        EAL: Probe PCI driver: net_ena (1d0f:ec20) device: 0000:00:06.0
(socket 0)
        EAL: No legacy callbacks, legacy socket not created
        Interactive-mode selected

        Port 0: link state change event
        testpmd: create a new mbuf pool <mb_pool_0>: n=171456,
size=2176, socket=0
        testpmd: preferred mempool ops selected: ring_mp_mc

        Warning! port-topology=paired and odd forward ports number, the
last port will pair with itself.

        Configuring Port 0 (socket 0)
        Port 0: 02:C7:17:A2:60:B1
        Checking link statuses...
        Done
        Error during enabling promiscuous mode for port 0: Operation
not supported - ignore
        testpmd> show port xstats 0
        ###### NIC extended statistics for port 0
        rx_good_packets: 0
        tx_good_packets: 0
        rx_good_bytes: 0
        tx_good_bytes: 0
        rx_missed_errors: 0
        rx_errors: 0
        tx_errors: 0
        rx_mbuf_allocation_errors: 0
        rx_q0_packets: 0
        rx_q0_bytes: 0
```

```
        rx_q0_errors: 0
        tx_q0_packets: 0
        tx_q0_bytes: 0
        wd_expired: 0
        dev_start: 1
        dev_stop: 0
        tx_drops: 0
        bw_in_allowance_exceeded: 0
        bw_out_allowance_exceeded: 0
        pps_allowance_exceeded: 0
        conntrack_allowance_exceeded: 0
        linklocal_allowance_exceeded: 0
        rx_q0_cnt: 0
        rx_q0_bytes: 0
        rx_q0_refill_partial: 0
        rx_q0_bad_csum: 0
        rx_q0_mbuf_alloc_fail: 0
        rx_q0_bad_desc_num: 0
        rx_q0_bad_req_id: 0
        tx_q0_cnt: 0
        tx_q0_bytes: 0
        tx_q0_prepare_ctx_err: 0
        tx_q0_linearize: 0
        tx_q0_linearize_failed: 0
        tx_q0_tx_poll: 0
        tx_q0_doorbells: 0
        tx_q0_bad_req_id: 0
        tx_q0_available_desc: 1023
        testpmd>
```

有关示例应用程序以及使用它检索扩展统计信息的更多信息。请参阅 DPDK 文档中的 Testpmd 应用程序用户指南。

运行 FreeBSD 的实例的指标

从 2.3.0 版开始，ENA FreeBSD 驱动程序支持在运行 FreeBSD 的实例上收集网络性能指标。要启用 FreeBSD 指标的收集，请输入以下命令并将 *interval* 设置为 1 到 3600 之间的值。这将指定收集 FreeBSD 指标的频率（以秒为单位）。

```
sysctl dev.ena.network_interface.eni_metrics.sample_interval=interval
```

例如，以下命令将驱动程序设置为每 10 秒在网络接口 1 上收集 FreeBSD 指标：

```
sysctl dev.ena.1.eni_metrics.sample_interval=10
```

要关闭 FreeBSD 指标收集，您可以运行上述命令并将 *interval* 指定为 0。

启用 FreeBSD 指标收集后，您可以通过运行以下命令来检索最新收集的指标集。

```
sysctl dev.ena.network_interface.eni_metrics
```

改善基于 Linux 的 EC2 实例的网络延迟

网络延迟是指数据包从源传输到目的地所花费的时间。通过网络发送数据的应用程序依赖于及时的响应来提供积极的用户体验。高网络延迟可能会导致各种问题，例如：

- 网页加载时间慢
- 视频流滞后
- 难以访问在线资源

本节概述了您可以采取哪些步骤来改善 Linux 上运行的 Amazon EC2 实例的网络延迟。要实现最佳延迟，请按照以下步骤配置您的实例、内核和 ENA 驱动程序设置。有关其他配置指南，请参阅 GitHub 上的 ENA Linux 驱动程序最佳实践和性能优化指南。

> **Note**
>
> 步骤和设置可能略有不同，具体取决于您的特定网络硬件、启动实例的 AMI 以及您的应用程序用例。在进行任何更改之前，请全面测试和监控您的网络性能，以确保获得预期的结果。

减少数据包的网络跃点数

数据包在路由器之间移动时经过的每个跃点都会增加网络延迟。通常，流量必须经过多个跃点才能到达目的地。有两种方法可以减少 Amazon EC2 实例的网络跃点数，具体如下：

- 集群置放群组：当您指定集群置放群组时，Amazon EC2 会启动彼此靠近、物理上位于同一可用区（AZ）且封装更紧密的实例。群组中实例的物理接近允许它们利用高速连接，从而实现低延迟和高单流吞吐量。
- 专属主机：专属主机是一种专供您使用的物理服务器。通过专属主机，您可以启动您的实例，使其在同一台物理服务器上运行。在同一专属主机上运行的实例无需任何额外的网络跃点即可在彼此之间进行通信。

Linux 内核配置如何影响延迟

Linux 内核配置可以增加或减少网络延迟。要实现延迟优化目标，必须根据工作负载的具体要求对 Linux 内核配置进行微调。

Linux 内核有许多配置选项可能有助于减少网络延迟。最具影响力的选项如下。

- 启用忙轮询模式 – 忙轮询模式可减少网络接收路径上的延迟。启用忙轮询模式时，套接字层代码可以直接轮询网络设备的接收队列。繁忙轮询的缺点是，由于在紧密循环中轮询新数据，主机中的 CPU 使用率更高。有两种全局设置可以控制等待所有接口数据包的微秒数。

  `busy_read`

  套接字读取的低延迟繁忙轮询超时。这可以控制等待套接字层读取设备队列上的数据包的微秒数。要使用 sysctl 命令全局启用该功能，Linux 内核组织推荐的值为 50 微秒。有关更多信息，请参阅 Linux 内核用户和管理员指南中的 busy_read。

  ```
  [ec2-user ~]$ sudo sysctl -w net.core.busy_read=50
  ```

  `busy_poll`

  轮询和选择的低延迟繁忙轮询超时。这可以控制等待事件的微秒数。推荐值介于 50-100 微秒之间，具体取决于您要轮询的套接字数。您添加的套接字越多，数字就应该越大。

  ```
  [ec2-user ~]$ sudo sysctl -w net.core.busy_poll=50
  ```

- 配置 CPU 电源状态（C 状态）– C 状态控制内核在非活动状态下可能进入的睡眠级别。您可能需要控制 C 状态来调整系统的延迟与性能。在更深层的 C 状态中，CPU 本质上处于睡眠状态，在唤醒并转回活动状态之前无法响应请求。将核心置于睡眠状态需要时间，尽管睡眠中的核心可为其他核心提供更多空间以加速至更高频率，但该睡眠中的核心也需要时间来重新唤醒并执行工作。

  例如，如果某个负责处理网络数据包中断的核心处于睡眠状态，那么在处理此类中断时可能会出现延迟。您可以配置系统，使其不使用更深层的 C 状态。虽然这种配置降低了处理器反应延迟，但也减少了 Turbo Boost 其他内核的可用空间。

  为了降低处理器的反应延迟，可以限制深层 C 状态。有关更多信息，请参阅《Amazon Linux 2 User Guide》中的 High performance and low latency by limiting deeper C-states。

中断裁决

ENA 网络驱动程序支持实例和网络之间的通信。该驱动程序可处理网络数据包，并将其传递到网络堆栈或 Nitro 卡。当网络数据包进入时，Nitro 卡会产生一个中断，以便 CPU 通知软件某个事件。

中断

中断是设备或应用程序发送到处理器的信号。中断通知处理器发生某个事件或达到条件，需要立即关注。中断可以处理时间敏感型任务，例如从网络接口接收数据、处理硬件事件或处理来自其他设备的请求。

中断裁决

中断裁决是一种技术，通过聚合或延迟中断来减少设备产生的中断数量。中断裁决的目的是通过减少与处理大量中断相关的开销来提高系统性能。中断过多会增加 CPU 使用率，对吞吐量产生不利影响，而中断过少会增加延迟。

动态中断裁决

动态中断裁决是一种增强的中断裁决形式，可根据当前系统负载和流量模式动态调整中断速率。它的目标是在减少中断开销和每秒数据包或带宽之间取得平衡。

> **Note**
>
> 某些 AMI 默认启用动态中断裁决（但在所有 AMI 中都可以启用或禁用）。

为了最大限度减少网络延迟，可能需要禁用中断裁决。但是，这也会增加中断处理的开销。在减少延迟和最小化开销之间找到正确的平衡非常重要。`ethtool` 命令可以帮助您配置中断裁决。默认情况下，`rx-usecs` 设置为 `20`，`tx-usecs` 设置为 `64`。

要获取当前的中断裁决配置，请使用以下命令：

```
[ec2-user ~]$ ethtool -c interface | egrep "rx-usecs:|tx-usecs:|Adaptive RX"
Adaptive RX: on  TX: off
rx-usecs: 20
tx-usecs: 64
```

要禁用中断裁决和动态中断裁决，请使用以下命令。

```
[ec2-user ~]$ sudo ethtool -C interface adaptive-rx off rx-usecs 0 tx-usecs 0
```

Nitro 系统性能调整注意事项

Nitro 系统是由 Amazon 打造的硬件和软件组件集合，可实现高性能、高可用性和高安全性。Nitro System 提供类似裸机的功能，从而消除了所有虚拟化开销并支持需要完全访问主机硬件的工作负载。有关更多详细信息，请参阅 Amazon Nitro System。

所有最新一代 EC2 实例类型都在 EC2 Nitro 卡上执行网络数据包处理。本主题涵盖 Nitro 卡上的高级数据包处理、影响数据包处理性能的网络架构和配置的常见方面，以及您可以采取哪些措施来实现基于 Nitro 的实例的峰值性能。

Nitro 卡可处理所有输入和输出（I/O）接口，例如虚拟私有云（VPC）所需的接口。对于通过网络发送或接收信息的所有组件，Nitro 卡可充当 I/O 流量的独立计算设备，在物理上与运行客户工作负载的系统主板分开。

Nitro 卡上的网络数据包流

基于 Nitro System 构建的 EC2 实例具有硬件加速功能，可以更快地处理数据包，以每秒数据包数（PPS）吞吐率衡量。当 Nitro 卡对新流执行初始评估时，其会保存流中所有数据包相同的信息，例如安全组、访问控制列表和路由表条目。当处理同一个流的额外数据包时，其可以使用保存的信息来减少这些数据包的开销。

您的连接速率通过每秒连接数（CPS）指标来衡量。每个新连接都需要额外的处理开销，必须将其纳入工作负载能力估计中考虑。在设计工作负载时，请务必同时考虑 CPS 和 PPS 指标。

如何建立连接

在基于 Nitro 的实例与另一个端点之间建立连接时，Nitro 卡会评估两个端点之间发送或接收的第一个数据包的完整流。对于同一个流的后续数据包，通常不需要完全重新评估。但有一些例外情况。有关异常的更多信息，请参阅 不使用硬件加速的数据包。

以下属性定义了两个端点及其之间的数据包流。这五个属性一起统称为 5 元组流。

- 源 IP
- 源端口
- 目的地 IP
- 目的地端口
- 通信协议

数据包流的方向称为入口（入站）和出口（出站）。以下高级描述总结了端到端网络数据包流。

- 入口 – 当 Nitro 卡处理入站网络数据包时，它会根据状态防火墙规则和访问控制列表评估数据包。它跟踪连接、计量连接并执行其他适用的操作。然后，它将数据包转发到主机 CPU 上的目标。
- 出口 – 当 Nitro 卡处理出站网络数据包时，它会查找远程接口目标，评估各种 VPC 功能，应用速率限制，并执行其他适用的操作。然后，它会将数据包转发到网络上的下一跃点目标。

实现最佳性能的网络设计

要利用 Nitro 系统的性能，您必须了解网络处理需求以及这些需求如何影响 Nitro 资源的工作负载。然后，您可以针对网络环境进行设计以获得最佳性能。您的基础架构设置以及应用程序工作负载设计和配置可能会影响数据包处理和连接速率。例如，如果您的应用程序建立了高速率连接（例如 DNS 服务、防火墙或虚拟路由器），则其利用硬件加速的机会就会减少，此类加速仅会发生在建立连接后。

您可以配置应用程序和基础架构设置，以简化工作负载并提高网络性能。但是，并非所有数据包都符合加速条件。Nitro 系统使用完整的网络流处理新连接和不符合加速条件的数据包。

本节的其余部分将重点介绍应用程序和基础架构设计注意事项，以帮助确保数据包尽可能在加速路径内流动。

Nitro 系统的网络设计注意事项

在为实例配置网络流量时，需要考虑许多可能影响 PPS 性能的方面。建立流量后，大多数定期传入或传出的数据包都符合加速条件。但是，为了确保基础架构设计和数据包流继续符合协议标准，也存在例外情况。

要从 Nitro 卡中获得最佳性能，您应仔细考虑以下基础架构和应用程序配置详细信息的优缺点。

基础架构注意事项

您的基础架构配置可能会影响数据包流和处理效率。以下列表包括一些重要注意事项。

具有非对称性的网络接口配置

安全组使用连接跟踪来跟踪有关流入和流出实例的流量的信息。非对称路由（即流量通过一个网络接口进入实例，然后通过另一个网络接口离开）可能会降低实例在跟踪流量时所能达到的峰值性能。有关安全组连接跟踪、未跟踪的连接和自动跟踪的连接的更多信息，请参阅Amazon EC2 安全组连接跟踪。

网络驱动程序

网络驱动程序会定期更新和发布。如果您的驱动程序已过时，则可能会严重影响性能。让驱动程序保持最新，以确保您拥有最新的补丁并可以利用性能改进，例如仅适用于最新一代驱动程序的加速路径功能。早期版本的驱动程序不支持加速路径功能。

要利用加速路径功能，建议您在实例上安装最新的 ENA 驱动程序。

Linux 实例：ENA Linux 驱动程序 2.2.9 或更高版本。要从 Amazon Drivers GitHub 存储库安装或更新 ENA Linux 驱动程序，请参阅自述文件中的驱动程序编译部分。

Windows 实例：ENA Windows 驱动程序 2.0.0 或更高版本。要安装或更新 ENA Windows 驱动程序，请参阅 在 EC2 Windows 实例上安装 ENA 驱动程序。

端点之间的距离

由于应用程序层的 TCP 窗口化，同一可用区域中两个实例之间的连接每秒可以处理的数据包比跨区域连接更多，这决定了在任何给定时间可以传输的数据量。实例之间的距离较长会增加延迟，并减少端点可以处理的数据包数量。

字节队列限制（BQL）

BQL 是一项功能，用于限制传递给 Nitro 卡的字节数以减少排队。在 ENA 驱动程序、Amazon Linux 操作系统以及大多数 Linux 发行版中，BQL 默认处于禁用状态。如果同时启用 BQL 和片段代理覆盖，则会限制处理所有片段之前传递给 Nitro 的字节数，从而导致性能限制。

应用程序设计注意事项

应用程序设计和配置的某些方面可能会影响您的处理效率。以下列表包括一些重要注意事项。

数据包大小

较大的数据包大小可以增加实例可在网络上发送和接收的数据的吞吐量。Amazon EC2 支持 9001 字节的巨型帧，但其他服务可能会实施不同的限制。较小的数据包大小可以提高数据包处理速率，但当数据包数量超过 PPS 限额时，这可能会降低实现的最大带宽。

如果数据包的大小超过网络跃点的最大传输单元（MTU），则路径上的路由器可能会将其分段。生成的数据包片段被视为例外，且通常以标准速率（未加速）进行处理。这可能会导致您的性能发生变化。但您可以使用片段代理模式设置覆盖出站片段数据包的标准行为。有关更多信息，请参阅 最大限度地提高 Nitro 系统的网络性能。我们建议您在配置 MTU 时评估拓扑。

协议权衡

TCP 等可靠协议比 UDP 等不可靠协议具有更多开销。UDP 传输协议较低的开销和简化的网络处理可以带来更高的 PPS 速率，但代价是牺牲可靠的数据包传输。如果可靠的数据包传输对您的应用程序并不重要，则 UDP 可能是一个不错的选择。

微爆

当流量在短时间内超过限额而不是均匀分布时，就会发生微爆。这通常发生在微秒级。

例如，假设您有一个实例可以发送高达 10Gbps 的速率，而您的应用程序在半秒钟内发送了完整的 10Gb。这种微爆在前半秒超出了允许值，而在后半秒的剩余时间里没有留下任何内容。尽管您在 1 秒的时间范围内发送了 10Gb，但前半秒的限额可能会导致数据包排队或被丢弃。

您可以使用诸如 Linux Traffic Control 之类的网络调度程序来帮助调整吞吐量，避免因微爆导致数据包排队或被丢弃。

流数

单个流限制在 5Gbps 以内，除非它位于支持高达 10Gbps 的集群置放群组内，或者它使用支持高达 25Gbps 的 ENA Express。

同样，Nitro 卡可以跨多个流处理更多数据包，而不是使用单个流。为了实现每个实例的峰值数据包处理速率，我们建议在聚合带宽为 100 Gbps 或更高的实例上至少有 100 个流。随着聚合带宽容量的增加，实现峰值处理速率所需的流数也随之增加。基准测试将帮助您确定在网络上实现峰值速率所需的配置。

弹性网络适配器（ENA）队列

ENA（弹性网络适配器）使用多个接收（Rx）和传输（Tx）队列（统称为 ENA 队列），来提高 EC2 实例上的网络性能和可扩展性。这些队列通过跨可用队列进行收发数据的负载均衡，从而有效地管理网络流量。

有关更多信息，请参阅 the section called “ENA 队列数”。

功能处理开销

流量镜像和 ENA Express 等功能可能会增加更多的处理开销，从而降低绝对数据包处理性能。您可以限制功能使用或禁用功能以提高数据包处理速率。

连接跟踪以保持状态

您的安全组使用连接跟踪来存储有关进出实例的流量的信息。连接跟踪对网络流量每个单独的流应用规则，以确定是允许还是拒绝该流量。Nitro 卡使用流跟踪来维护流状态。随着安全组规则的应用越来越多，评估流程需要做更多工作。

> **Note**
>
> 并非所有网络流量都会被跟踪。如果安全组规则配置了 未跟踪的连接，则除了在有多个有效回复路径时自动跟踪连接以确保对称路由之外，无需进行其他工作。

不使用硬件加速的数据包

并非所有数据包都能利用硬件加速。处理这些例外情况会涉及一些处理开销，这是确保网络流正常运行所必需的。网络流必须可靠地满足协议标准，符合 VPC 设计的变化，并且仅将数据包路由到允许的目标。但是，开销会降低您的性能。

数据包片段

如应用程序注意事项下所述，由超过网络 MTU 的数据包生成的数据包片段通常作为例外情况处理，无法利用硬件加速。但是，您可以使用片段代理模式绕过出口片段限制，具体取决于您的驱动程序版本。有关更多信息，请参阅 最大限度地提高 Nitro 系统的网络性能 部分中所列的具体操作。

空闲连接

当连接在一段时间内没有活动时，即使该连接尚未达到其超时限制，系统也可以取消其优先级。然后，如果数据是在取消连接优先级后进入的，则系统需要将其作为例外情况处理才能重新连接。

要管理连接，您可以使用连接跟踪超时来关闭空闲连接。您也可以使用 TCP keepalive 使空闲连接保持打开状态。有关更多信息，请参阅 空闲连接跟踪超时。

VPC 突变

安全组、路由表和访问控制列表的更新都需要在处理路径中进行重新评估，以确保路由条目和安全组规则仍然按预期适用。

ICMP 流

互联网控制信息协议 (ICMP) 是网络设备用来诊断网络通信问题的网络层协议。这些数据包始终使用完整流。

非对称 L2 流量

对于同一子网中的两个 ENI，如果其中一个 ENI 使用默认网关路由器，而另一个不使用默认网关路由器，则 Nitrov3 和更早版本的平台不会为两者之间的流量使用硬件加速，而 Nitrov4 及更高版本的平台会为此情况使用硬件加速。为确保在 Nitrov3 或更早版本的平台上获得更好的性能，请确保这两个 ENI 之间使用的默认网关路由器一致，或者确保这两个 ENI 位于不同的子网中。

最大限度地提高 Nitro 系统的网络性能

您可以通过调整网络设置来提高 Nitro 系统上的网络性能。

主题

- 注意事项

- 调整 PPS 性能
- 配置 ENA 队列分配
- 监控 Linux 实例的性能

注意事项

在您做出任何设计决策或调整实例上的任何网络设置之前，建议您采取以下步骤来帮助确保获得最佳结果：

1. 通过回顾 Nitro 系统的网络设计注意事项，了解为提高性能可以采取的措施的利弊。

   有关 Linux 上实例配置的更多注意事项和最佳实践，请参阅 GitHub 上的 ENA Linux Driver Best Practices and Performance Optimization Guide。
2. 使用峰值活跃流计数对您的工作负载进行基准测试，以确定应用程序性能的基准。使用性能基准，您可以测试设置或应用程序设计中的变体，以了解哪些注意事项的影响最大，尤其是在您计划纵向扩展或横向扩展时。

调整 PPS 性能

以下列表包含您可以根据系统需求调整 PPS 性能的措施。

- 缩短两个实例之间的物理距离。当发送和接收实例位于同一个可用区或使用集群置放群组时，您可以减少数据包从一个端点传输到另一个端点所需的跃点数。
- 使用 未跟踪的连接。
- 对网络流量使用 UDP 协议。
- 对于聚合带宽为 100Gbps 或更高的 EC2 实例，请将工作负载分配到 100 个或更多的单个流中，以将工作均匀分布到 Nitro 卡上。
- 要消除 EC2 实例的出口片段 PPS 限制，您可以启用片段代理模式（取决于驱动程序版本）。此设置允许在处理路径中评估分段的数据包，从而消除 1024 的出口 PPS 限制。加载驱动程序时，请运行以下命令之一以启用或禁用片段代理模式：

  启用片段代理模式

  ```
  sudo insmod ena.ko enable_frag_bypass=1
  ```

  禁用片段代理模式

```
sudo insmod ena.ko enable_frag_bypass=0
```

配置 ENA 队列分配

在受支持的实例类型上，您可以跨弹性网络接口（ENI）动态分配这些队列。灵活分配 ENA 队列有助于优化资源分配，从而尽可能提高 vCPU 利用率。需要高网络性能的工作负载通常需要使用多个 ENA 队列。有关更多信息，请参阅 ENA 队列数。

监控 Linux 实例的性能

您可以在 Linux 实例上使用 Ethtool 指标，监控带宽、数据包速率和连接跟踪等实例网络性能指标。有关更多信息，请参阅 监控 EC2 实例上 ENA 设置的网络性能。

优化 EC2 Windows 实例上的网络性能

要在具有增强联网的 Windows 实例上实现最高网络性能，您可能需要修改默认操作系统配置。我们建议对于需要高网络性能的应用程序进行以下配置更改。官方 Windows AMI 上已经配置了其他优化（例如打开校验和卸载和启用 RSS）。

> **Note**
>
> 大多数使用情况下应禁用 TCP Chimney 分载，自 Windows Server 2016 开始已弃用该功能。

除了这些操作系统优化之外，您还应考虑网络流量的最大传输单位 (MTU)，并根据您的工作负载和网络架构相应调整。有关更多信息，请参阅EC2 实例的网络最大传输单位 (MTU)。

Amazon 定期测量在集群置放群组中启动的实例之间的平均往返延迟，在 99.9% 的情况下该值为 50us，尾延迟为 200us。如果您的应用程序需要稳定的低延迟，建议在基于 Nitro 系统的固定性能实例上，使用最新版本的 ENA 驱动程序。

配置接收端缩放 CPU 关联

使用接收端缩放 (RSS) 在多个处理器之间分布网络流量 CPU 负载。默认情况下，Amazon 官方 Windows AMI 配置为启用 RSS。ENA 弹性网络接口最多提供 8 个 RSS 队列。通过为 RSS 队列以及其他系统进程定义 CPU 关联，可以在多核系统上分布 CPU 负载，允许处理更多的网络流量。如果实例类型上的 vCPU 超过 16 个，建议使用 `Set-NetAdapterRSS` PowerShell cmdlt，因为这会手动从所有弹性网络接口的 RSS 配置中排除启动处理器（启用超线程时为逻辑处理器 0 和 1），以防与各种系统组件争用。

Windows 具有超线程感知功能，可确保单个网络接口卡（NIC）的 RSS 队列始终位于不同的物理内核上。因此，除非禁用超线程，否则为了完全防止与其他 NIC 的争用，请将每个 NIC 的 RSS 配置分布在 16 个逻辑处理器的范围内。`Set-NetAdapterRss` cmdlet 允许您通过定义 BaseProcessorGroup、BaseProcessorNumber、MaxProcessingGroup、MaxProcessorNumber 和 NumaNode（可选）的值来定义有效逻辑处理器的各 NIC 范围。如果没有足够的物理内核来完全消除 NIC 间的争用，则尽量缩小重叠范围或减少弹性网络接口范围内的逻辑处理器数量，具体取决于接口的预期工作负载（换句话说，低容量管理网络接口可能不需要分配尽可能多的 RSS 队列）。此外，如前所述，各种组件必须在 CPU 0 上运行，因此我们建议在有足够的 vCPU 时将其从所有 RSS 配置中排除。

例如，当具有 72 个 vCPU 的实例上有 3 个弹性网络接口，其中 2 个 NUMA 节点启用了超线程时，以下命令会在两个 CPU 之间分布网络负载而不会重叠，并完全阻止内核 0 的使用。

```
Set-NetAdapterRss -Name NIC1 -BaseProcessorGroup 0 -BaseProcessorNumber 2 -
MaxProcessorNumber 16
Set-NetAdapterRss -Name NIC2 -BaseProcessorGroup 1 -BaseProcessorNumber 0 -
MaxProcessorNumber 14
Set-NetAdapterRss -Name NIC3 -BaseProcessorGroup 1 -BaseProcessorNumber 16 -
MaxProcessorNumber 30
```

请注意，这些设置对每个网络适配器都是持久的。如果将实例调整为具有不同数量的 vCPU 的实例，则应重新评估启用的每个弹性网络接口的 RSS 配置。cmdlet 的完整 Microsoft 文档可在以下位置找到：Set-NetAdapterRss。

SQL 工作负载特别注意事项：建议检查输入/输出线程关联设置以及弹性网络接口 RSS 配置，以便最大限度地减少相同 CPU 的输入/输出和网络争用。请参阅 Server configuration: affinity mask。

适用于 Amazon EC2 上 AI/ML 和 HPC 工作负载的 Elastic Fabric Adapter

Elastic Fabric Adapter（EFA）是一种网络设备，可以将其附加到 Amazon EC2 实例以加速人工智能（AI）、机器学习（ML）和高性能计算（HPC）应用程序。EFA 使您能够利用 Amazon 云提供的可扩展性、灵活性和弹性，实现本地 AI/ML 或 HPC 集群的应用程序性能。

与以前在基于云的 HPC 系统中使用的 TCP 传输相比，EFA 提供更低且更一致的延迟和更高的吞吐量。它提高了实例间通信的性能，这对于扩展 AI/ML 和 HPC 应用程序至关重要。它经过优化以在现有的 Amazon 网络基础设施上使用，并且可以根据应用程序要求进行扩展。

EFA 与 Libfabric 集成在一起，并支持适用于 AI 和 ML 应用程序的 Nvidia Collective Communications Library (NCCL) 和 NVIDIA Inference Xfer Library (NIXL)，以及适用于 HPC 应用程序的 Open MPI 4.1 及更高版本和 Intel MPI 2019 Update 5 及更高版本。NCCL 和 MPI 与 Libfabric 1.7.0 及更高版本集成。NIXL 与 Libfabric 1.21.0 及更高版本集成。

EFA 支持在大多数具有 Nitro 版本 4 及更高版本的受支持实例类型上进行 RDMA（远程直接内存访问）写入。所有具有 Nitro 版本 4 及更高版本的实例均支持 RDMA 读取。有关更多信息，请参阅 支持的实例类型。

内容

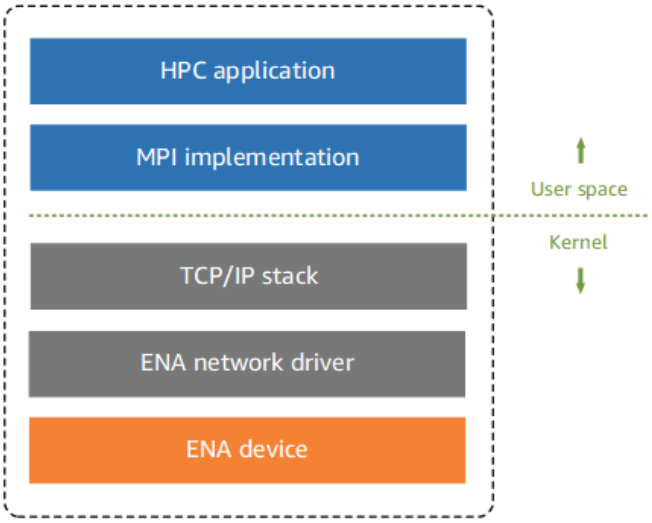
- EFA 基础知识
- 支持的接口和库
- 支持的实例类型
- 支持的操作系统
- EFA 限制
- EFA 定价
- 开始使用 EFA 和 MPI 处理 Amazon EC2 上的 HPC 工作负载
- 开始使用 EFA 和 NCCL 处理 Amazon EC2 上的 ML 工作负载
- 开始使用 EFA 和 NIXL 处理 Amazon EC2 上的推理工作负载
- 使用多网卡最大化 Amazon EC2 实例上的网络带宽
- 创建 Elastic Fabric Adapter 并将其附加到 Amazon EC2 实例
- 将 EFA 与 Amazon EC2 实例分离并删除 EFA
- 监控 Amazon EC2 上的 Elastic Fabric Adapter
- 使用校验和验证 EFA 安装程序
- Elastic Fabric Adapter 发布说明

EFA 基础知识

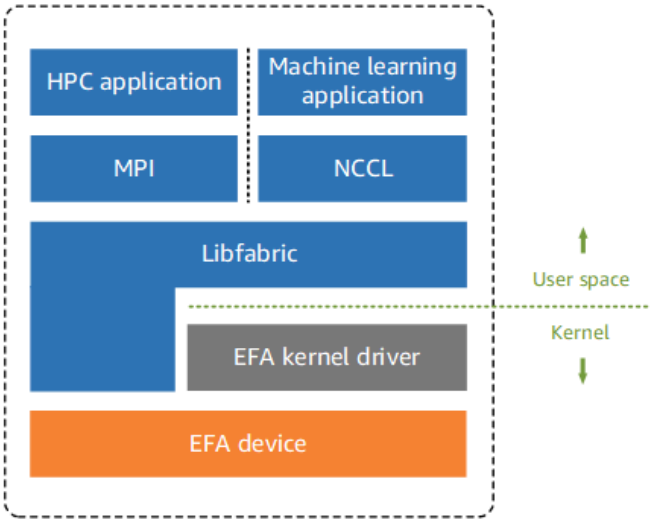
EFA 设备可以通过两种方式附加到 EC2 实例：

1. 使用传统 EFA 接口（也称为带 ENA 的 EFA），它可以创建 EFA 设备和 ENA 设备。
2. 使用仅限 EFA 的接口，该接口仅创建 EFA 设备。

EFA 设备通过可扩展的可靠数据报（SRD）协议提供内置操作系统绕过和拥塞控制等功能。EFA 设备功能支持低延迟、可靠的传输功能，可让 EFA 接口为 Amazon EC2 上的 HPC 和 ML 应用程序提供更理想的应用程序性能。ENA 设备则提供传统的 IP 联网。有关根据使用案例使用哪种实例配置的指导，请参阅 最大化网络带宽。



Traditional HPC software stack in EC2



HPC software stack in EC2 with EFA

传统上，AI/ML 应用程序使用 NCCL 和 NIXL（用于分解推理）。HPC 应用程序使用消息传递接口 (MPI) 与系统的网络传输进行交互。在 Amazon 云中，这意味着应用程序与 NCCL、NIXL 或 MPI 进行交互，然后 MPI 使用操作系统的 TCP/IP 堆栈和 ENA 设备驱动程序以启用实例之间的网络通信。

对于传统 EFA（带 ENA 的 EFA）或仅限 EFA 的接口，AI/ML 应用程序使用 NCCL 和 NIXL（带 ENA 的 EFA）或仅限 EFA 的接口（用于分解推理）。HPC 应用程序使用 MPI 直接与 Libfabric API 进行交互。Libfabric API 绕过操作系统内核，并直接与 EFA 设备通信以将数据包放在网络上。这减少了开销，并且可以更有效地运行 AL/ML 和 HPC 应用程序。

> **Note**
>
> libfabric 是 OpenFabrics 接口 (OFI) 框架的核心组件，它定义并导出 OFI 的 user-space API。有关更多信息，请参阅 libfabric OpenFabrics 网站。

ENA、EFA 和仅限 EFA 的网络接口之间的区别

Amazon EC2 提供两种类型的网络接口：

- ENA 接口提供支持 VPC 的 IP 联网所需的所有传统 IP 联网和路由功能。有关更多信息，请参阅 在 EC2 实例上使用 ENA 启用增强联网功能。

- EFA（带 ENA 的 EFA）接口既提供用于 IP 联网的 ENA 设备，又提供用于低延迟、高吞吐量通信的 EFA 设备。
- 仅限 EFA 的接口仅支持 EFA 设备功能，不支持用于传统 IP 联网的 ENA 设备。

下表比较 ENA、EFA 和仅限 EFA 的网络接口。

	ENA	EFA（带 ENA 的 EFA）	仅限 EFA
支持 IP 联网功能	支持	是	否
可以分配 IPv4 或 IPv6 地址	支持	是	否
例如，可以用作主网络接口	支持	是	否
计入实例的 ENI 附件限制	支持	是	是
实例类型支持	在所有基于 Nitro 的实例类型上支持	支持的实例类型	支持的实例类型
EC2 API 中的参数命名	`interface`	`efa`	`efa-only`
EC2 控制台中的字段命名	未选择	带 ENA 的 EFA	仅限 EFA

支持的接口和库

EFA 支持以下接口和库：

- Open MPI 4.1 及更高版本
- Intel MPI 2019 Update 5 及更高版本
- NVIDIA Collective Communications Library (NCCL) 2.4.2 及更高版本

- NVIDIA Inference Xfer Library (NIXL) 1.0.0 及更高版本
- Amazon Neuron SDK 版本 2.3 及更高版本

支持的实例类型

以下所有实例类型均支持 EFA。此外，这些表还指示对实例类型的 RDMA 读取和 RDMA 写入支持。

Nitro v6 (EFA v4)

实例类型	RDMA 读取支持	RDMA 写入支持
通用		
m8a.48xlarge	Yes	Yes
m8a.metal-48xl	Yes	Yes
m8azn.24xlarge	Yes	Yes
m8azn.metal-24xl	Yes	Yes
m8gb.16xlarge	Yes	Yes
m8gb.24xlarge	Yes	Yes
m8gb.48xlarge	Yes	Yes
m8gb.metal-24xl	Yes	Yes
m8gb.metal-48xl	Yes	Yes
m8gn.16xlarge	Yes	Yes
m8gn.24xlarge	Yes	Yes
m8gn.48xlarge	Yes	Yes
m8gn.metal-24xl	Yes	Yes
m8gn.metal-48xl	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
m8i.48xlarge	Yes	Yes
m8i.96xlarge	Yes	Yes
m8i.metal-48xl	Yes	Yes
m8i.metal-96xl	Yes	Yes
m8id.48xlarge	Yes	Yes
m8id.96xlarge	Yes	Yes
m8id.metal-48xl	Yes	Yes
m8id.metal-96xl	Yes	Yes
m8in.48xlarge	Yes	Yes
m8in.96xlarge	Yes	Yes
m8in.metal-48xl	Yes	Yes
m8in.metal-96xl	Yes	Yes
m8idn.48xlarge	Yes	Yes
m8idn.96xlarge	Yes	Yes
m8idn.metal-48xl	Yes	Yes
m8idn.metal-96xl	Yes	Yes
m8ib.48xlarge	Yes	Yes
m8ib.96xlarge	Yes	Yes
m8ib.metal-48xl	Yes	Yes
m8ib.metal-96xl	Yes	Yes
m8idb.48xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
m8idb.96xlarge	Yes	Yes
m8idb.metal-48xl	Yes	Yes
m8idb.metal-96xl	Yes	Yes
m9g.48xlarge	Yes	Yes
m9g.metal-48xl	Yes	Yes
m9gd.48xlarge	Yes	Yes
m9gd.metal-48xl	Yes	Yes
计算优化		
c8a.48xlarge	Yes	Yes
c8a.metal-48xl	Yes	Yes
c8gb.16xlarge	Yes	Yes
c8gb.24xlarge	Yes	Yes
c8gb.48xlarge	Yes	Yes
c8gb.metal-24xl	Yes	Yes
c8gb.metal-48xl	Yes	Yes
c8gn.16xlarge	Yes	Yes
c8gn.24xlarge	Yes	Yes
c8gn.48xlarge	Yes	Yes
c8gn.metal-24xl	Yes	Yes
c8gn.metal-48xl	Yes	Yes
c8i.48xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
c8i.96xlarge	Yes	Yes
c8i.metal-48xl	Yes	Yes
c8i.metal-96xl	Yes	Yes
c8id.48xlarge	Yes	Yes
c8id.96xlarge	Yes	Yes
c8id.metal-48xl	Yes	Yes
c8id.metal-96xl	Yes	Yes
c8in.48xlarge	Yes	Yes
c8in.96xlarge	Yes	Yes
c8in.metal-48xl	Yes	Yes
c8in.metal-96xl	Yes	Yes
c8ib.48xlarge	Yes	Yes
c8ib.96xlarge	Yes	Yes
c8ib.metal-48xl	Yes	Yes
c8ib.metal-96xl	Yes	Yes
内存优化		
r8a.48xlarge	Yes	Yes
r8a.metal-48xl	Yes	Yes
r8gb.16xlarge	Yes	Yes
r8gb.24xlarge	Yes	Yes
r8gb.48xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
r8gb.metal-24xl	Yes	Yes
r8gb.metal-48xl	Yes	Yes
r8gn.16xlarge	Yes	Yes
r8gn.24xlarge	Yes	Yes
r8gn.48xlarge	Yes	Yes
r8gn.metal-24xl	Yes	Yes
r8gn.metal-48xl	Yes	Yes
r8i.48xlarge	Yes	Yes
r8i.96xlarge	Yes	Yes
r8i.metal-48xl	Yes	Yes
r8i.metal-96xl	Yes	Yes
r8id.48xlarge	Yes	Yes
r8id.96xlarge	Yes	Yes
r8id.metal-48xl	Yes	Yes
r8id.metal-96xl	Yes	Yes
r8in.48xlarge	Yes	Yes
r8in.96xlarge	Yes	Yes
r8in.metal-48xl	Yes	Yes
r8in.metal-96xl	Yes	Yes
r8idn.48xlarge	Yes	Yes
r8idn.96xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
r8idn.metal-48xl	Yes	Yes
r8idn.metal-96xl	Yes	Yes
r8ib.48xlarge	Yes	Yes
r8ib.96xlarge	Yes	Yes
r8ib.metal-48xl	Yes	Yes
r8ib.metal-96xl	Yes	Yes
r8idb.48xlarge	Yes	Yes
r8idb.96xlarge	Yes	Yes
r8idb.metal-48xl	Yes	Yes
r8idb.metal-96xl	Yes	Yes
x8aedz.24xlarge	Yes	Yes
x8aedz.metal-24xl	Yes	Yes
x8i.48xlarge	Yes	Yes
x8i.64xlarge	Yes	Yes
x8i.96xlarge	Yes	Yes
x8i.metal-48xl	Yes	Yes
x8i.metal-96xl	Yes	Yes
存储优化		
i8ge.48xlarge	Yes	Yes
i8ge.metal-48xl	Yes	Yes
加速计算		

实例类型	RDMA 读取支持	RDMA 写入支持
g7e.8xlarge	Yes	Yes
g7e.12xlarge	Yes	Yes
g7e.24xlarge	Yes	Yes
g7e.48xlarge	Yes	Yes
p6-b200.48xlarge	Yes	Yes
p6-b300.48xlarge	Yes	Yes
高性能计算		
hpc8a.96xlarge	Yes	Yes

Nitro v5 (EFA v3)

实例类型	RDMA 读取支持	RDMA 写入支持
通用		
m8g.24xlarge	Yes	Yes
m8g.48xlarge	Yes	Yes
m8g.metal-24xl	Yes	Yes
m8g.metal-48xl	Yes	Yes
m8gd.24xlarge	Yes	Yes
m8gd.48xlarge	Yes	Yes
m8gd.metal-24xl	Yes	Yes
m8gd.metal-48xl	Yes	Yes
计算优化		

实例类型	RDMA 读取支持	RDMA 写入支持
c7gn.16xlarge	Yes	No
c7gn.metal	Yes	No
c8g.24xlarge	Yes	Yes
c8g.48xlarge	Yes	Yes
c8g.metal-24xl	Yes	Yes
c8g.metal-48xl	Yes	Yes
c8gd.24xlarge	Yes	Yes
c8gd.48xlarge	Yes	Yes
c8gd.metal-24xl	Yes	Yes
c8gd.metal-48xl	Yes	Yes
内存优化		
r8g.24xlarge	Yes	Yes
r8g.48xlarge	Yes	Yes
r8g.metal-24xl	Yes	Yes
r8g.metal-48xl	Yes	Yes
r8gd.24xlarge	Yes	Yes
r8gd.48xlarge	Yes	Yes
r8gd.metal-24xl	Yes	Yes
r8gd.metal-48xl	Yes	Yes
x8g.24xlarge	Yes	Yes
x8g.48xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
x8g.metal-24xl	Yes	Yes
x8g.metal-48xl	Yes	Yes
存储优化		
i7ie.48xlarge	Yes	Yes
i7ie.metal-48xl	Yes	Yes
i8g.48xlarge	Yes	Yes
i8g.metal-48xl	Yes	Yes
加速计算		
p5en.48xlarge	Yes	Yes
p6e-gb200.36xlarge	Yes	Yes
trn2.3xlarge	Yes	Yes
trn2.48xlarge	Yes	Yes
trn2u.48xlarge	Yes	Yes
高性能计算		
hpc7g.4xlarge	Yes	No
hpc7g.8xlarge	Yes	No
hpc7g.16xlarge	Yes	No

Nitro v4 (EFA v2)

实例类型	RDMA 读取支持	RDMA 写入支持
通用		

实例类型	RDMA 读取支持	RDMA 写入支持
m6a.48xlarge	Yes	Yes
m6a.metal	Yes	Yes
m6i.32xlarge	Yes	Yes
m6i.metal	Yes	Yes
m6id.32xlarge	Yes	Yes
m6id.metal	Yes	Yes
m6idn.32xlarge	Yes	Yes
m6idn.metal	Yes	Yes
m6in.32xlarge	Yes	Yes
m6in.metal	Yes	Yes
m7a.48xlarge	Yes	Yes
m7a.metal-48xl	Yes	Yes
m7g.16xlarge	Yes	Yes
m7g.metal	Yes	Yes
m7gd.16xlarge	Yes	Yes
m7gd.metal	Yes	Yes
m7i.48xlarge	Yes	Yes
m7i.metal-48xl	Yes	Yes
计算优化		
c6a.48xlarge	Yes	Yes
c6a.metal	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
c6gn.16xlarge	Yes	Yes
c6i.32xlarge	Yes	Yes
c6i.metal	Yes	Yes
c6id.32xlarge	Yes	Yes
c6id.metal	Yes	Yes
c6in.32xlarge	Yes	Yes
c6in.metal	Yes	Yes
c7a.48xlarge	Yes	Yes
c7a.metal-48xl	Yes	Yes
c7g.16xlarge	Yes	Yes
c7g.metal	Yes	Yes
c7gd.16xlarge	Yes	Yes
c7gd.metal	Yes	Yes
c7i.48xlarge	Yes	Yes
c7i.metal-48xl	Yes	Yes
内存优化		
r6a.48xlarge	Yes	Yes
r6a.metal	Yes	Yes
r6i.32xlarge	Yes	Yes
r6i.metal	Yes	Yes
r6id.32xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
r6id.metal	Yes	Yes
r6idn.32xlarge	Yes	Yes
r6idn.metal	Yes	Yes
r6in.32xlarge	Yes	Yes
r6in.metal	Yes	Yes
r7a.48xlarge	Yes	Yes
r7a.metal-48xl	Yes	Yes
r7g.16xlarge	Yes	Yes
r7g.metal	Yes	Yes
r7gd.16xlarge	Yes	Yes
r7gd.metal	Yes	Yes
r7i.48xlarge	Yes	Yes
r7i.metal-48xl	Yes	Yes
r7iz.32xlarge	Yes	Yes
r7iz.metal-32xl	Yes	Yes
u7i-6tb.112xlarge	Yes	Yes
u7i-8tb.112xlarge	Yes	Yes
u7i-12tb.224xlarge	Yes	Yes
u7in-16tb.224xlarge	Yes	Yes
u7in-24tb.224xlarge	Yes	Yes
u7in-32tb.224xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
u7inh-32tb.480xlarge	Yes	Yes
x2idn.32xlarge	Yes	Yes
x2idn.metal	Yes	Yes
x2iedn.32xlarge	Yes	Yes
x2iedn.metal	Yes	Yes
存储优化		
i4g.16xlarge	Yes	Yes
i4i.32xlarge	Yes	Yes
i4i.metal	Yes	Yes
i7i.24xlarge	Yes	Yes
i7i.48xlarge	Yes	Yes
i7i.metal-48xl	Yes	Yes
im4gn.16xlarge	Yes	Yes
加速计算		
f2.48xlarge	Yes	Yes
g6.8xlarge	Yes	Yes
g6.12xlarge	Yes	Yes
g6.16xlarge	Yes	Yes
g6.24xlarge	Yes	Yes
g6.48xlarge	Yes	Yes
g6e.8xlarge	Yes	Yes

实例类型	RDMA 读取支持	RDMA 写入支持
g6e.12xlarge	Yes	Yes
g6e.16xlarge	Yes	Yes
g6e.24xlarge	Yes	Yes
g6e.48xlarge	Yes	Yes
gr6.8xlarge	Yes	Yes
p5.4xlarge	Yes	Yes
p5.48xlarge	Yes	Yes
p5e.48xlarge	Yes	Yes
trn1.32xlarge	Yes	Yes
trn1n.32xlarge	Yes	Yes
高性能计算		
hpc6a.48xlarge	Yes	Yes
hpc6id.32xlarge	Yes	Yes
hpc7a.12xlarge	Yes	Yes
hpc7a.24xlarge	Yes	Yes
hpc7a.48xlarge	Yes	Yes
hpc7a.96xlarge	Yes	Yes

Nitro v3 (EFA v1)

实例类型	RDMA 读取支持	RDMA 写入支持
通用		

实例类型	RDMA 读取支持	RDMA 写入支持
m5dn.24xlarge	No	No
m5dn.metal	No	No
m5n.24xlarge	No	No
m5n.metal	No	No
m5zn.12xlarge	No	No
m5zn.metal	No	No
计算优化		
c5n.9xlarge	No	No
c5n.18xlarge	No	No
c5n.metal	No	No
内存优化		
r5dn.24xlarge	No	No
r5dn.metal	No	No
r5n.24xlarge	No	No
r5n.metal	No	No
x2iezn.12xlarge	No	No
x2iezn.metal	No	No
存储优化		
i3en.12xlarge	No	No
i3en.24xlarge	No	No
i3en.metal	No	No

实例类型	RDMA 读取支持	RDMA 写入支持
加速计算		
dl2q.24xlarge	No	No
g4dn.8xlarge	No	No
g4dn.12xlarge	No	No
g4dn.16xlarge	No	No
g4dn.metal	No	No
g5.8xlarge	No	No
g5.12xlarge	No	No
g5.16xlarge	No	No
g5.24xlarge	No	No
g5.48xlarge	No	No
inf1.24xlarge	No	No
p3dn.24xlarge	No	No
p4d.24xlarge	Yes	No
p4de.24xlarge	Yes	No
vt1.24xlarge	No	No
上一代		
p3dn.24xlarge	No	No

查看特定区域中支持 EFA 的可用实例类型

可用的实例类型因区域而异。要查看某个区域中支持 EFA 的可用实例类型，请使用带 `--region` 参数的 describe-instance-types 命令。包括 `--filters` 参数以将结果范围限定为支持 EFA 的实例类型，并包括 `--query` 参数以将输出范围限定为 `InstanceType` 的值。

```
aws ec2 describe-instance-types \
    --region us-east-1  \
    --filters Name=network-info.efa-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]"  \
    --output text | sort
```

支持的操作系统

操作系统支持因处理器类型而异。下表显示支持的操作系统。

操作系统	Intel/AMD（**x86_64**）实例类型	Amazon Graviton（**arm64**）实例类型
Amazon Linux 2023	✓	✓
Amazon Linux 2	✓	✓
RHEL 8、9 和 10	✓	✓
Debian 11、12 和 13	✓	✓
Rocky Linux 8 和 9	✓	✓
Ubuntu 22.04 和 24.04	✓	✓
SUSE Linux Enterprise 15 SP2 和更高版本	✓	✓

> **Note**
>
> Intel MPI 可能不支持部分列出的操作系统。如果您使用的是 Intel MPI，请参阅 Intel MPI 文档验证是否您的操作系统是否受支持。

EFA 限制

EFA 具有以下限制：

- 并非所有实例类型都支持 RDMA 写入。有关更多信息，请参阅 支持的实例类型。
- 目前不支持 P4d/P4de/DL1 实例与其他实例类型之间的 EFA 流量[1]。
- 支持多个网卡的实例类型可以为每个网卡配置一个 EFA。所有其他支持的实例类型每个实例仅支持一个 EFA。
- `c7g.16xlarge`、`m7g.16xlarge` 和 `r7g.16xlarge`，当附加 EFA 时，不支持专用实例和专属主机。
- EFA 流量[1]无法跨越可用区或 VPC。这不适用于来自 EFA 接口的 ENA 设备的正常 IP 流量。
- 无法路由 EFA 流量[1]。仍然可以路由来自 EFA 接口的 ENA 设备的普通 IP 流量。
- Amazon Outposts 上不支持 EFA。
- 仅在基于 Amazon Cloud Digital Interface 软件开发工具包（Amazon CDI SDK）的应用程序的 Windows 实例上支持 EFA（带 ENA 的 EFA）接口的 EFA 设备。如果您将 EFA（带 ENA 的 EFA）接口附加到基于非 CDI SDK 的应用程序的 Windows 实例，则该接口将充当 ENA 接口，而没有任何附加 EFA 设备功能。Windows 或 Linux 上基于 Amazon CDI 的应用程序不支持仅限 EFA 的接口。有关更多信息，请参阅 Amazon Cloud Digital Interface 软件开发工具包（Amazon CDI SDK）用户指南。

[1]EFA 流量是指通过 EFA（带 ENA 的 EFA）或仅限 EFA 接口的 EFA 设备传输的流量。

EFA 定价

EFA 作为一项可选的 Amazon EC2 联网功能提供，您可以在任何支持的实例上启用该功能，无需支付额外费用。

开始使用 EFA 和 MPI 处理 Amazon EC2 上的 HPC 工作负载

本教程可帮助您为 HPC 工作负载启动启用了 EFA 和 MPI 的实例集群。

> **Note**
>
> u7i-12tb.224xlarge、u7in-16tb.224xlarge、u7in-24tb.224xlarge、和 u7in-32tb.224xlarge 实例使用 Open MPI 最多可以运行 128 个并行 MPI 进程，使用 Intel MPI 最多可以运行 256 个并行 MPI 进程。

任务

- 步骤 1：准备启用 EFA 的安全组
- 步骤 2：启动临时实例
- 步骤 3：安装 EFA 软件
- 步骤 4：（可选）启用 Open MPI 5
- 步骤 5：（可选）安装 Intel MPI
- 步骤 6：禁用 Ptrace 保护
- 第 7 步。确认安装
- 步骤 8：安装 HPC 应用程序
- 步骤 9：创建启用 EFA 的 AMI
- 步骤 10：在集群置放群组中启动启用 EFA 的实例
- 步骤 11：终止临时实例
- 步骤 12：启用无密码 SSH

步骤 1：准备启用 EFA 的安全组

EFA 需要使用一个安全组，以允许进出安全组本身的所有入站和出站流量。以下过程创建了一个安全组，该安全组允许所有进出其本身的入站和出站流量，并允许来自任何 IPv4 地址的入站 SSH 流量进行 SSH 连接。

> **Important**
>
> 此安全组仅用于测试目的。对于您的生产环境，建议您创建入站 SSH 规则，该规则仅允许来自您连接的 IP 地址的流量，例如计算机的 IP 地址或本地网络中的一系列 IP 地址。

有关其他场景，请参阅 针对不同使用案例的安全组规则。

创建启用 EFA 的安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Security Groups (安全组)，然后选择 Create Security Group (创建安全组)。
3. 在 Create security group（创建安全组）窗口中，执行以下操作：
   a. 对于安全组名称，请输入一个描述性的安全组名称，例如 `EFA-enabled security group`。
   b. （可选）对于描述，请输入安全组的简要描述。
   c. 对于 VPC，请选择要在其中启动启用了 EFA 的实例的 VPC。
   d. 选择创建安全组。
4. 选择您创建的安全组，然后在 Details（详细信息）选项卡上复制 Security group ID（安全组 ID）。
5. 在安全组仍然选中的情况下，依次选择 Actions（操作）、Edit inbound rules（编辑入站规则），然后执行以下操作：
   a. 选择添加规则。
   b. 对于 Type (类型)，请选择 All traffic (所有流量)。
   c. 对于 Source type（源类型），请选择 Custom（自定义）并将您复制的安全组 ID 粘贴到该字段中。
   d. 选择添加规则。
   e. 对于 Type，选择 SSH。
   f. 对于 Source type（源类型），请选择 Anywhere-IPv4。
   g. 选择保存规则。
6. 在安全组仍然选中的情况下，依次选择 Actions（操作）、Edit outbound rules（编辑出站规则），然后执行以下操作：
   a. 选择添加规则。
   b. 对于 Type (类型)，请选择 All traffic (所有流量)。
   c. 对于 Destination type（目标类型），请选择 Custom（自定义）并将您复制的安全组 ID 粘贴到该字段中。
   d. 选择保存规则。

步骤 2：启动临时实例

启动一个临时实例，可用于安装和配置 EFA 软件组件。您使用该实例创建一个启用了 EFA 的 AMI，您可以从中启动启用了 EFA 的实例。

启动临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在 Name and tags（名称和标签）部分中，提供实例的名称，例如 `EFA-instance`。名称作为资源标签（Name=*`EFA-instance`*）分配给实例。
4. 在 Application and OS Images（应用程序和操作系统映像）部分中，为其中一个支持的操作系统选择 AMI。
5. 在 Instance type（实例类型）部分中，选择 supported instance type（支持的实例类型）。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 Network settings（网络设置）部分中，请选择 Edit（编辑），然后执行以下操作：

   a. 对于子网，选择要在其中启动实例的子网。

   > **Important**
   >
   > 必须选择子网。如果您未选择子网，则不能启用 EFA 的实例。

   b. 对于 Firewall（security groups）（防火墙（安全组）），请选择 Select existing security group（选择现有安全组），然后选择您在上一步中创建的安全组。

   c. 展开高级网络配置部分。

   对于网络接口 1，选择网卡索引 = 0、设备索引 = 0、接口类型 = 带 ENA 的 EFA。

   （可选）如果您使用的是多卡实例类型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，则对于每个额外的网络接口，选择添加网络接口；对于网卡索引，选择下一个未使用的索引，然后选择设备索引 = 1 和接口类型 = 带 ENA 的 EFA 或仅限 EFA。
8. 在 Storage（存储）部分中，根据需要配置卷。
9. 在右侧的 Summary（摘要）面板中，选择 Launch instance（启动实例）。

> **Note**
>
> 考虑要求临时实例以及您将在步骤 9 中创建的 AMI 使用 IMDSv2，除非您已将 IMDSv2 设置为账户的默认值。有关 IMDSv2 配置的更多信息，请参阅 为新实例配置实例元数据选项。

步骤 3：安装 EFA 软件

在临时实例上安装支持 EFA 所需的启用 EFA 的内核、EFA 驱动程序、Libfabric 和 Open MPI 堆栈。

根据您打算将 EFA 与 Open MPI、与 Intel MPI 还是同时与这两者结合使用，这些步骤有所不同。

> **Note**
>
> Intel MPI 可能不支持部分操作系统。如果您使用的是 Intel MPI，请参阅 Intel MPI 文档验证是否您的操作系统是否受支持。

安装 EFA 软件

1. 连接到您启动的实例。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
2. 为确保您的所有软件包都处于最新状态，请对您的实例执行快速软件更新。此过程可能需要几分钟时间。

   - Amazon Linux 2023、Amazon Linux 2、RHEL 8/9、Rocky Linux 8/9

     ```
     $ sudo yum update -y
     ```

   - Ubuntu 和 Debian

     ```
     $ sudo apt-get update && sudo apt-get upgrade -y
     ```

   - SUSE Linux Enterprise

     ```
     $ sudo zypper update -y
     ```

3. 重启实例并重新连接到它。
4. 下载 EFA 软件安装文件。软件安装文件将打包为压缩的 tarball（`.tar.gz`）文件。要下载最新的稳定 版本，请使用以下命令。

您也可以通过将上面命令中的版本号替换为 `latest` 来获取最新版本。

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.48.0.tar.gz
```

5. （可选）验证 EFA tarball（`.tar.gz`）文件的真实性和完整性。

建议您执行此操作以验证软件发布者的身份，并检查该文件自发布以来是否已被更改或损坏。如果您不想验证 tarball 文件，请跳过此步骤。

> **Note**
>
> 或者，如果您希望使用 MD5 或 SHA256 校验和验证 tarball 文件，请参阅 使用校验和验证 EFA 安装程序。

a. 下载公有 GPG 密钥并将其导入到您的密钥环中。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
import aws-efa-installer.key
```

该命令应返回一个密钥值。请记下密钥值，因为需要在下一步中使用该值。

b. 验证 GPG 密钥的指纹。运行以下命令并指定上一步中的密钥值。

```
$ gpg --fingerprint key_value
```

该命令应返回一个与 `4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC` 相同的指纹。如果指纹不匹配，请不要运行 EFA 安装脚本，并联系 Amazon Web Services 支持。

c. 下载签名文件并验证 EFA tarball 文件的签名。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.48.0.tar.gz.sig
 && gpg --verify ./aws-efa-installer-1.48.0.tar.gz.sig
```

下面显示了示例输出。

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
```

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

如果结果包含 Good signature，并且指纹与上一步中返回的指纹相匹配，请继续下一步。如果没有，请不要运行 EFA 安装脚本，并联系 Amazon Web Services 支持。

6. 从压缩的 .tar.gz 文件中提取文件，并导航到提取的目录。

```
$ tar -xf aws-efa-installer-1.48.0.tar.gz && cd aws-efa-installer
```

7. （可选）在安装过程中验证各个软件包的签名。

从 EFA 安装程序 1.48.0 开始，安装程序包括 GPG 签名的各个 RPM 和 DEB 软件包。要在安装过程中验证每个软件包的真实性和完整性，请使用 --check-signatures 标志。启用此标志后，安装程序会首先验证所有软件包的签名，只有在每个软件包都通过验证后才会继续安装。如果有任何软件包未通过验证，安装程序将立即退出，而不安装任何程序。

a. 下载 GPG 公有密钥。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
```

b. 导出密钥路径。然后，在下一步中，将 --check-signatures 附加到安装命令并使用 sudo -E 来保留环境变量，而不是 sudo。

```
$ export EFA_INSTALLER_KEY=$(pwd)/aws-efa-installer.key
```

在基于 RPM 的系统（Amazon Linux、RHEL、Rocky Linux 和 SUSE）上，安装程序使用 rpm --checksig 来验证每个 RPM。在基于 DEB 的系统（Ubuntu、Debian）上，安装程序使用 GPG 签名验证来验证每个 DEB。

如果任何软件包验证失败，安装将立即中止，从而保护您的系统免受损坏或恶意软件包的侵害。

> **Note**
>
> --check-signatures 标记是可选的。若没有此项验证，安装程序不会执行个人签名验证。

8. 安装 EFA 软件。根据您的使用案例执行以下某种操作。

> **Note**
>
> 如果您完成了前一个可选步骤来设置软件包签名验证，请将 `--check-signatures` 附加到安装命令并使用 `sudo -E`，而不是 `sudo`。例如：`sudo -E ./efa_installer.sh -y --check-signatures`。

> **Note**
>
> 在 SUSE Linux 操作系统下，EFA 不支持 NVIDIA GPUDirect。如果使用的是 SUSE Linux 操作系统，则必须另外指定 `--skip-kmod` 选项来防止 kmod 安装。预设情况下，SUSE Linux 不允许树外的内核模块。

Open MPI and Intel MPI

如果打算将 EFA 同时与 Open MPI 和 Intel MPI 结合使用，则必须安装 EFA 软件及 Libfabric 和 Open MPI，并且必须完成步骤 5：安装 Intel MPI。

若要安装 EFA 软件以及 Libfabric 和 Open MPI，请运行以下命令。

> **Note**
>
> 从 EFA 1.30.0 开始，默认情况下会同时安装 Open MPI 4.1 和 Open MPI 5。您也可以指定要安装的 Open MPI 版本。要仅安装 Open MPI 4.1，请包含 `--mpi=openmpi4`。要仅安装 Open MPI 5，请指定 `--mpi=openmpi5`。要同时安装两者，请忽略 `--mpi` 选项。

```
$ sudo ./efa_installer.sh -y
```

Libfabric 已安装到 `/opt/amazon/efa`。Open MPI 4.1 会安装到 `/opt/amazon/openmpi`。Open MPI 5 已安装到 `/opt/amazon/openmpi5`。

Open MPI only

如果打算将 EFA 与 Open MPI 结合使用，则必须安装 EFA 软件及 Libfabric 和 Open MPI，并且可以跳过步骤 5：安装 Intel MPI。若要安装 EFA 软件以及 Libfabric 和 Open MPI，请运行以下命令。

> **Note**
>
> 从 EFA 1.30.0 开始，默认情况下会同时安装 Open MPI 4.1 和 Open MPI 5。您也可以指定要安装的 Open MPI 版本。要仅安装 Open MPI 4.1，请包含 `--mpi=openmpi4`。要仅安装 Open MPI 5，请指定 `--mpi=openmpi5`。要同时安装两者，请忽略 `--mpi` 选项。

```
$ sudo ./efa_installer.sh -y
```

Libfabric 已安装到 `/opt/amazon/efa`。Open MPI 4.1 会安装到 `/opt/amazon/openmpi`。Open MPI 5 已安装到 `/opt/amazon/openmpi5`。

Intel MPI only

如果您仅打算将 EFA 与 Intel MPI 结合使用，则可以安装 EFA 而不带 Libfabric 和 Open MPI。在这种情况下，Intel MPI 使用其嵌入式 Libfabric。如果选择执行此操作，则必须完成步骤 5：安装 Intel MPI。

若要在没有 Libfabric 和 Open MPI 的情况下安装 EFA 软件，请运行以下命令。

```
$ sudo ./efa_installer.sh -y --minimal
```

9. 如果 EFA 安装程序提示您重启实例，请执行此操作，然后重新连接到实例。否则，请注销实例，然后重新登录以完成安装。
10. 删除解压缩后的压缩包和压缩包本身。否则，这些内容将包含在您创建的启用 EFA 的 AMI 中，从而增加其大小。

步骤 4：（可选）启用 Open MPI 5

> **Note**
>
> 仅在打算使用 Open MPI 5 时执行此步骤。

从 EFA 1.30.0 开始，默认情况下会同时安装 Open MPI 4.1 和 Open MPI 5。您也可以选择仅安装 Open MPI 4.1 或 Open MPI 5。

如果在步骤 3：安装 EFA 软件中选择安装 Open MPI 5，并且打算使用该软件，则必须执行以下步骤才能将其启用。

启用 Open MPI 5

1. 将 Open MPI 5 添加到 PATH 环境变量中。

   ```
   $ module load openmpi5
   ```

2. 确认已启用 Open MPI 5 以供使用。

   ```
   $ which mpicc
   ```

   命令应返回 Open MPI 5 安装目录：`/opt/amazon/openmpi5`。

3. （可选）要确保每次启动实例时都将 Open MPI 5 添加到 PATH 环境变量中，请执行以下操作：

   bash shell

   将 `module load openmpi5` 添加到 `/home/`*`username`*`/.bashrc` 和 `/home/`*`username`*`/.bash_profile`。

   csh and tcsh shells

   将 `module load openmpi5` 添加到 `/home/`*`username`*`/.cshrc`。

如果需要从 PATH 环境变量中删除 Open MPI 5，请运行以下命令，并从 Shell 启动脚本中删除该命令。

```
$ module unload openmpi5
```

步骤 5：（可选）安装 Intel MPI

> **Important**
>
> 仅在您打算使用 Intel MPI 时执行此步骤。如果您仅打算使用 Open MPI，请跳过此步骤。

Intel MPI 需要使用额外的安装和环境变量配置。

先决条件

确保执行以下步骤的用户具有 sudo 权限。

安装 Intel MPI

1. 要下载 Intel MPI 安装脚本，请执行以下操作

   a. 访问 Intel 网站。

   b. 在网页的 Intel MPI Library 部分，选择 Intel MPI Library for Linux Offline 安装程序的链接。

2. 运行您在上一个步骤中下载的安装脚本。

   ```
   $ sudo bash installation_script_name.sh
   ```

3. 在安装程序中，选择 Accept & install（接受并安装）。

4. 阅读 Intel 改进计划，选择适当的选项，然后选择 Begin Installation（开始安装）。

5. 安装完成后，选择关闭。

6. 默认情况下，Intel MPI 使用其嵌入式（内部）Libfabric。您可以将 Intel MPI 配置为使用 EFA 安装程序附带的 Libfabric。通常，EFA 安装程序附带的 Libfabric 版本高于 Intel MPI。在某些情况下，EFA 安装程序附带的 Libfabric 比 Intel MPI 性能更强。要将 Intel MPI 配置为使用 EFA 安装程序附带的 Libfabric，请根据您的 Shell 执行以下操作之一。

   bash shells

   将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   export I_MPI_OFI_LIBRARY_INTERNAL=0
   ```

csh and tcsh shells

将以下语句添加到 /home/*username*/.cshrc。

```
setenv I_MPI_OFI_LIBRARY_INTERNAL 0
```

7. 将以下 source 命令添加到 Shell 脚本中，以从安装目录获取 vars.sh 脚本，以便在每次启动实例时设置编译器环境。根据您的 Shell 执行下列操作之一。

   bash shells

   将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   source /opt/intel/oneapi/mpi/latest/env/vars.sh
   ```

   csh and tcsh shells

   将以下语句添加到 /home/*username*/.cshrc。

   ```
   source /opt/intel/oneapi/mpi/latest/env/vars.csh
   ```

8. 默认情况下，如果 EFA 因配置错误而不可用，Intel MPI 将默认为 TCP/IP 网络堆栈，这可能会导致应用程序性能下降。您可以通过将 I_MPI_OFI_PROVIDER 设置为 efa 来防止这种情况。如果 EFA 不可用，这将导致 Intel MPI 出现以下错误：

   ```
   Abort (XXXXXX) on node 0 (rank 0 in comm 0): Fatal error in PMPI_Init: OtherMPI
    error,
   MPIR_Init_thread (XXX).......:
   MPID_Init (XXXX).............:
   MPIDI_OFI_mpi_init_hook (XXXX):
   open_fabric (XXXX)...........:
   find_provider (XXXX).........:
   OFI fi_getinfo() failed (ofi_init.c:2684:find_provider:
   ```

   根据您的 Shell 执行下列操作之一。

   bash shells

   将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export I_MPI_OFI_PROVIDER=efa
```

csh and tcsh shells

将以下语句添加到 /home/*username*/.cshrc。

```
setenv I_MPI_OFI_PROVIDER efa
```

9. 默认情况下，Intel MPI 不会输出调试信息。您可以指定不同的详细程度来控制调试信息。可能的值（按提供的详细信息量排列）包括：0（默认值）、1、2、3、4、5。1 级及以上将输出 `libfabric version` 和 `libfabric provider`。使用 `libfabric version` 查看 Intel MPI 使用的是内部 Libfabric，还是 EFA 安装程序附带的 Libfabric。如果使用的是内部 Libfabric，则该版本的后缀为 `impi`。使用 `libfabric provider` 查看 Intel MPI 使用的是 EFA，还是 TCP/IP 网络。如果使用的是 EFA，则值为 `efa`。如果使用的是 TCP/IP，则值为 `tcp;ofi_rxm`。

   要启用调试信息，请根据您的 Shell 执行以下操作之一。

   bash shells

   将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   export I_MPI_DEBUG=value
   ```

   csh and tcsh shells

   将以下语句添加到 /home/*username*/.cshrc。

   ```
   setenv I_MPI_DEBUG value
   ```

10. 默认情况下，Intel MPI 使用操作系统的共享内存 (shm) 进行节点内通信，且仅使用 Libfabric (ofi) 进行节点间通信。通常，此配置可提供最佳性能。但是，在某些情况下，Intel MPI shm 结构可能会导致某些应用程序无限期挂起。

    要解决此问题，您可以强制 Intel MPI 使用 Libfabric 进行节点内和节点间通信。为此，请根据您的 Shell 执行下列操作之一。

    bash shells

    将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export I_MPI_FABRICS=ofi
```

csh and tcsh shells

将以下语句添加到 /home/*username*/.cshrc。

```
setenv I_MPI_FABRICS ofi
```

> **Note**
>
> EFA Libfabric 提供程序使用操作系统的共享内存进行节点内通信。这意味着将 I_MPI_FABRICS 设置为 ofi 可产生与默认 shm:ofi 配置接近的性能。

11. 注销实例，然后重新登录。

如果您不再需要使用 Intel MPI，请从 Shell 启动脚本中删除环境变量。

步骤 6：禁用 Ptrace 保护

为了提高 HPC 应用程序的性能，当进程在同一实例上运行时，Libfabric 使用实例的本地内存进行进程间通信。

共享内存功能使用交叉内存附加 (CMA) 功能，但 ptrace 保护不支持该功能。如果您正在使用默认情况下启用 ptrace 保护的 Linux 发行版（例如 Ubuntu），则必须禁用它。如果您的 Linux 发行版默认未启用 ptrace 保护，请跳过此步骤。

禁用 ptrace 保护

请执行下列操作之一：

- 要临时禁用 ptrace 保护以进行测试，请运行以下命令。

```
$ sudo sysctl -w kernel.yama.ptrace_scope=0
```

- 要永久禁用 ptrace 保护，请将 kernel.yama.ptrace_scope = 0 添加到 /etc/sysctl.d/10-ptrace.conf 并重新启动实例。

第 7 步。确认安装

要确认安装成功

1. 要确认 MPI 已成功安装，请运行以下命令：

   ```
   $ which mpicc
   ```

   - 对于 Open MPI，返回的路径应包括 `/opt/amazon/`
   - 对于 Intel MPI，返回的路径应包括 `/opt/intel/`。如果您没有得到预期的输出，请确保您已获取 Intel MPI `vars.sh` 脚本。
2. 要确认已成功安装 EFA 软件组件和 Libfabric，请运行以下命令。

   ```
   $ fi_info -p efa -t FI_EP_RDM
   ```

   该命令应返回有关 Libfabric EFA 接口的信息。以下示例显示了命令输出。

   ```
   provider: efa
       fabric: EFA-fe80::94:3dff:fe89:1b70
       domain: efa_0-rdm
       version: 2.0
       type: FI_EP_RDM
       protocol: FI_PROTO_EFA
   ```

步骤 8：安装 HPC 应用程序

在临时实例上安装 HPC 应用程序。安装过程因特定的 HPC 应用程序而异。有关更多信息，请参阅《Amazon Linux 2 User Guide》中的 Manage software on your AL2 instance。

> **Note**
>
> 请参阅 HPC 应用程序文档以了解安装说明。

步骤 9：创建启用 EFA 的 AMI

在安装所需的软件组件后，您可以创建一个 AMI，然后可以将其重复使用以启动启用了 EFA 的实例。

从临时实例创建 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您创建的临时实例，然后依次选择 Actions (操作)、Image (映像)、Create image (创建映像)。
4. 对于 Create image (创建映像)，请执行以下操作：
   a. 对于 Image name (映像名称)，为 AMI 输入一个描述性名称。
   b. （可选）对于 Image description (映像描述)，输入 AMI 用途的简要描述。
   c. 选择创建映像。
5. 在导航窗格中，选择 AMIs。
6. 在列表中找到您创建的 AMI。等待状态从 `pending` 更改为 `available`，然后继续下一步。

步骤 10：在集群置放群组中启动启用 EFA 的实例

使用在步骤 7 中创建的启用了 EFA 的 AMI 以及在步骤 1 中创建的启用了 EFA 的安全组，在集群置放群组中启动启用了 EFA 的实例。

> **Note**
>
> - 在集群置放群组中启动启用了 EFA 的实例并不是一个绝对要求。不过，我们建议在集群置放群组中运行启用了 EFA 的实例，因为它在单个可用区的低延迟组中启动实例。
> - 为了确保在扩展集群实例时容量可用，您可以为集群置放群组创建容量预留。有关更多信息，请参阅 使用集群置放群组的容量预留。

启动实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在 Name and tags（名称和标签）部分中，提供实例的名称，例如 `EFA-instance`。名称作为资源标签（Name=*`EFA-instance`*）分配给实例。

4. 在 Application and OS Images（应用程序和操作系统映像）部分中，请选择 My AMIs（我的 AMI），然后选择您在上一步骤创建的 AMI。
5. 在 Instance type（实例类型）部分中，选择 supported instance type（支持的实例类型）。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 Network settings（网络设置）部分中，请选择 Edit（编辑），然后执行以下操作：

   a. 对于子网，选择要在其中启动实例的子网。

   > **Important**
   >
   > 必须选择子网。如果您未选择子网，则不能启用 EFA 的实例。

   b. 对于 Firewall（security groups）（防火墙（安全组）），请选择 Select existing security group（选择现有安全组），然后选择您在上一步中创建的安全组。

   c. 展开高级网络配置部分。

      对于网络接口 1，选择网卡索引 = 0、设备索引 = 0、接口类型 = 带 ENA 的 EFA。

      （可选）如果您使用的是多卡实例类型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，则对于每个额外的网络接口，选择添加网络接口；对于网卡索引，选择下一个未使用的索引，然后选择设备索引 1 和接口类型 = 带 ENA 的 EFA 或仅限 EFA。
8. （可选）在 Storage（存储）部分中，根据需要配置卷。
9. 在 Advanced details（高级详细信息）部分中，对于置放群组名称，选择要在其中启动实例的集群置放群组。如果您需要创建新的集群置放群组，请选择 Create new placement group（创建新置放群组）。
10. 在右侧的 Summary（摘要）面板中，为 Number of instances（实例数量）输入您要启动的启用了 EAA 的实例数量，然后选择 Launch instance（启动实例）。

步骤 11：终止临时实例

此时，您已不再需要在步骤 2 中启动的实例。您可以终止实例以停止产生费用。

终止临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您创建的临时实例，然后依次选择操作、实例状态、终止（删除）实例。

4. 当系统提示您确认时，选择终止（删除）。

步骤 12：启用无密码 SSH

要使应用程序能够在集群中的所有实例上运行，您必须启用从领导节点到成员节点的无密码 SSH 访问。领导节点是从中运行应用程序的实例。集群中的其余实例是成员节点。

在集群中的实例之间启用无密码 SSH

1. 在集群中选择一个实例作为领导节点，然后连接到该实例。
2. 在领导节点上禁用 `strictHostKeyChecking` 并启用 `ForwardAgent`。使用首选文本编辑器打开 `~/.ssh/config`，并添加以下内容。

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. 生成 RSA 密钥对。

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

密钥对在 `$HOME/.ssh/` 目录中创建。

4. 更改领导节点上私有密钥的权限。

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. 使用首选文本编辑器打开 `~/.ssh/id_rsa.pub` 并复制密钥。
6. 对于集群中的每个成员节点，请执行以下操作：
   a. 连接到实例。
   b. 使用首选文本编辑器打开 `~/.ssh/authorized_keys`，并添加之前复制的公有密钥。
7. 要测试无密码 SSH 是否按预期运行，请连接到领导节点并运行以下命令。

```
$ ssh member_node_private_ip
```

您应该连接到成员节点，而不会收到输入密钥或密码的提示。

开始使用 EFA 和 NCCL 处理 Amazon EC2 上的 ML 工作负载

NVIDIA Collective Communications Library (NCCL) 是一个标准集体通信例程库，它适用于跨单个节点或多个节点的多个 GPU。可将 NCCL 与 EFA、libfabric 和 MPI 结合使用来支持各种机器学习工作负载。有关更多信息，请参阅 NCCL 网站。

要求

- 仅支持加速型计算 P 系列实例类型。有关更多信息，请参阅 Amazon EC2 加速型计算实例。
- 仅支持 Amazon Linux 2023、Amazon Linux 2、Ubuntu 24.04 和 Ubuntu 22.04 基础 AMI。
- 仅支持将 NCCL 2.4.2 及更高版本与 EFA 结合使用。

有关使用 Amazon Deep Learning AMIs 运行包含 EFA 和 NCCL 的机器学习工作负载的更多信息，请参阅《Amazon Deep Learning AMIs 开发者指南》中的在 DLAMI 上使用 EFA。

Steps

- 步骤 1：准备启用 EFA 的安全组
- 步骤 2：启动临时实例
- 步骤 3：安装 Nvidia GPU 驱动程序、Nvidia CUDA 工具包和 cuDNN
- 步骤 4：安装 GDRCopy
- 步骤 5：安装 EFA 软件
- 步骤 6：安装 NCCL
- 步骤 7：安装 NCCL 测试
- 步骤 8：测试 EFA 和 NCCL 配置
- 步骤 9：安装机器学习应用程序
- 步骤 10：创建启用了 EFA 和 NCCL 的 AMI
- 步骤 11：终止临时实例
- 步骤 12：在集群置放群组中启动启用了 EFA 和 NCCL 的实例
- 步骤 13：启用无密码 SSH

步骤 1：准备启用 EFA 的安全组

EFA 需要使用一个安全组，以允许进出安全组本身的所有入站和出站流量。以下过程创建了一个安全组，该安全组允许所有进出其本身的入站和出站流量，并允许来自任何 IPv4 地址的入站 SSH 流量进行 SSH 连接。

> **Important**
>
> 此安全组仅用于测试目的。对于您的生产环境，建议您创建入站 SSH 规则，该规则仅允许来自您连接的 IP 地址的流量，例如计算机的 IP 地址或本地网络中的一系列 IP 地址。

有关其他场景，请参阅 针对不同使用案例的安全组规则。

创建启用 EFA 的安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Security Groups (安全组)，然后选择 Create Security Group (创建安全组)。
3. 在 Create security group（创建安全组）窗口中，执行以下操作：
   a. 对于安全组名称，请输入一个描述性的安全组名称，例如 `EFA-enabled security group`。
   b. （可选）对于描述，请输入安全组的简要描述。
   c. 对于 VPC，请选择要在其中启动启用了 EFA 的实例的 VPC。
   d. 选择创建安全组。
4. 选择您创建的安全组，然后在 Details（详细信息）选项卡上复制 Security group ID（安全组 ID）。
5. 在安全组仍然选中的情况下，依次选择 Actions（操作）、Edit inbound rules（编辑入站规则），然后执行以下操作：
   a. 选择添加规则。
   b. 对于 Type (类型)，请选择 All traffic (所有流量)。
   c. 对于 Source type（源类型），请选择 Custom（自定义）并将您复制的安全组 ID 粘贴到该字段中。
   d. 选择添加规则。
   e. 对于 Type，选择 SSH。

f. 对于 Source type（源类型），请选择 Anywhere-IPv4。

g. 选择保存规则。

6. 在安全组仍然选中的情况下，依次选择 Actions（操作）、Edit outbound rules（编辑出站规则），然后执行以下操作：

a. 选择添加规则。

b. 对于 Type (类型)，请选择 All traffic (所有流量)。

c. 对于 Destination type（目标类型），请选择 Custom（自定义）并将您复制的安全组 ID 粘贴到该字段中。

d. 选择保存规则。

步骤 2：启动临时实例

启动一个临时实例，可用于安装和配置 EFA 软件组件。您使用该实例创建一个启用了 EFA 的 AMI，您可以从中启动启用了 EFA 的实例。

启动临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在 Name and tags（名称和标签）部分中，提供实例的名称，例如 `EFA-instance`。名称作为资源标签（`Name=`*`EFA-instance`*）分配给实例。
4. 在应用程序和操作系统映像部分中，为其中一个支持的操作系统选择 AMI。
5. 在实例类型部分中，选择支持的实例类型。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 Network settings（网络设置）部分中，请选择 Edit（编辑），然后执行以下操作：

a. 对于子网，选择要在其中启动实例的子网。

> **Important**
>
> 必须选择子网。如果您未选择子网，则不能启用 EFA 的实例。

b. 对于 Firewall（security groups）（防火墙（安全组）），请选择 Select existing security group（选择现有安全组），然后选择您在上一步中创建的安全组。

c. 展开高级网络配置部分。

对于网络接口 1，选择网卡索引 = 0、设备索引 = 0、接口类型 = 带 ENA 的 EFA。

（可选）如果您使用的是多卡实例类型，例如 p4d.24xlarge 或 p5.48xlarge，则对于每个额外的网络接口，选择添加网络接口；对于网卡索引，选择下一个未使用的索引，然后选择设备索引 1 和接口类型 = 带 ENA 的 EFA 或仅限 EFA。

8. 在 Storage（存储）部分中，根据需要配置卷。

> **Note**
>
> 您必须为 Nvidia CUDA 工具包额外预置 10 到 20 GiB 的存储空间。如果您没有预置足够的存储空间，您将在尝试安装 Nvidia 驱动程序和 CUDA 工具包时收到 `insufficient disk space` 错误。

9. 在右侧的 Summary（摘要）面板中，选择 Launch instance（启动实例）。

步骤 3：安装 Nvidia GPU 驱动程序、Nvidia CUDA 工具包和 cuDNN

Amazon Linux 2023 and Amazon Linux 2

安装 Nvidia GPU 驱动程序、Nvidia CUDA 工具包和 cuDNN

1. 为确保您的所有软件包都处于最新状态，请对您的实例执行快速软件更新。

```
$ sudo yum upgrade -y && sudo reboot
```

实例重启后，重新连接到实例。

2. 安装在安装 Nvidia GPU 驱动程序和 Nvidia CUDA 工具包时需要的实用程序。

```
$ sudo yum groupinstall 'Development Tools' -y
```

3. 禁用 nouveau 开源驱动程序。

a. 为您当前运行的内核版本安装所需的实用程序和内核标头软件包。

```
$ sudo yum install -y wget kernel-devel-$(uname -r) kernel-headers-$(uname -r)
```

b. 将 nouveau 添加到/etc/modprobe.d/blacklist.conf 拒绝列表文件。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

c. 将 GRUB_CMDLINE_LINUX="rdblacklist=nouveau" 附加到 grub 文件并重新构建 Grub 配置。

```
$ echo 'GRUB_CMDLINE_LINUX="rdblacklist=nouveau"' | sudo tee -a /etc/
default/grub \
&& sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. 重启实例并重新连接到它。
5. 准备所需的存储库

a. 启用 EPEL 存储库，将发行版设置为 rhel7。

```
$ sudo amazon-linux-extras install epel \
&& distribution='rhel7'
```

b. 设置 CUDA 网络存储库并更新存储库缓存。

```
$ ARCH=$( /bin/arch ) \
&& sudo yum-config-manager --add-repo http://developer.download.nvidia.com/
compute/cuda/repos/$distribution/${ARCH}/cuda-$distribution.repo \
&& sudo yum clean expire-cache
```

c. （仅限 5.10 内核版本）请仅在您使用基于 5.10 内核版本的 Amazon Linux 2 时执行这些步骤。如果您使用基于 4.12 内核版本的 Amazon Linux 2，请跳过以下步骤。要检查内核版本，请运行 uname -r。

i. 创建名为 /etc/dkms/nvidia.conf 的 Nvidia 驱动程序配置文件。

```
$ sudo mkdir -p /etc/dkms \
```

```
&& echo "MAKE[0]=\"'make' -j2 module SYSSRC=\${kernel_source_dir}
 IGNORE_XEN_PRESENCE=1 IGNORE_PREEMPT_RT_PRESENCE=1 IGNORE_CC_MISMATCH=1
 CC=/usr/bin/gcc10-gcc\"" | sudo tee /etc/dkms/nvidia.conf
```

ii. （仅限 p4d.24xlarge 和 p5.48xlarge）复制 Nvidia 驱动程序配置文件。

```
$ sudo cp /etc/dkms/nvidia.conf /etc/dkms/nvidia-open.conf
```

6. 安装 Nvidia GPU 驱动程序、NVIDIA CUDA 工具包和 cuDNN。

```
$ sudo yum clean all \
&& sudo yum -y install nvidia-driver-latest-dkms \
&& sudo yum -y install cuda-drivers-fabricmanager cuda libcudnn8-devel
```

7. 重启实例并重新连接到它。

8. （仅限 p4d.24xlarge 和 p5.48xlarge）启动 Nvidia Fabric Manager 服务，并确保其在实例启动时自动启动。NV 交换管理需要 Nvidia Fabric 管理器。

```
$ sudo systemctl enable nvidia-fabricmanager && sudo systemctl start nvidia-
fabricmanager
```

9. 确保每次启动实例时均设置 CUDA 路径。

   - 对于 bash shell，请将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   export PATH=/usr/local/cuda/bin:$PATH
   export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
   lib64:$LD_LIBRARY_PATH
   ```

   - 对于 tcsh shell，请将以下语句添加到 /home/*username*/.cshrc。

   ```
   setenv PATH=/usr/local/cuda/bin:$PATH
   setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
   lib64:$LD_LIBRARY_PATH
   ```

10. 要验证 Nvidia GPU 驱动程序是否正常运行，请运行以下命令。

```
$ nvidia-smi -q | head
```

此命令应返回有关 Nvidia GPU、Nvidia GPU 驱动程序和 Nvidia CUDA 工具包的信息。

Ubuntu 24.04 and Ubuntu 22.04

安装 Nvidia GPU 驱动程序、Nvidia CUDA 工具包和 cuDNN

1. 为确保您的所有软件包都处于最新状态，请对您的实例执行快速软件更新。

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

2. 安装在安装 Nvidia GPU 驱动程序和 Nvidia CUDA 工具包时需要的实用程序。

```
$ sudo apt-get update && sudo apt-get install build-essential -y
```

3. 要使用 Nvidia GPU 驱动程序，您必须先禁用 nouveau 开源驱动程序。

   a. 为您当前运行的内核版本安装所需的实用程序和内核标头软件包。

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

   b. 将 nouveau 添加到/etc/modprobe.d/blacklist.conf 拒绝列表文件。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

   c. 使用首选文本编辑器打开 /etc/default/grub，并添加以下内容。

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

   d. 重新生成 Grub 配置。

```
$ sudo update-grub
```

4. 重启实例并重新连接到它。
5. 添加 CUDA 存储库并安装 Nvidia GPU 驱动程序、NVIDIA CUDA 工具包和 cuDNN。

   - p3dn.24xlarge

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-dkms-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535
 cuda-toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

- p4d.24xlarge 和 p5.48xlarge

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin \
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-kernel-open-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535
 cuda-toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

6. 重启实例并重新连接到它。
7. （仅限 p4d.24xlarge 和 p5.48xlarge）安装 Nvidia Fabric Manager。
   a. 您必须安装与上一步中安装的 Nvidia 内核模块版本匹配的 Nvidia Fabric Manager 版本。

运行以下命令以确定 Nvidia 内核模块的版本。

```
$ cat /proc/driver/nvidia/version | grep "Kernel Module"
```

下面是示例输出。

```
NVRM version: NVIDIA UNIX x86_64 Kernel Module  450.42.01  Tue Jun 15
 21:26:37 UTC 2021
```

上述示例中安装了内核模块的主要版本 450。这意味着您需要安装 Nvidia Fabric Manager 版本 450。

b. 安装 Nvidia Fabric Manager。运行以下命令并指定上一步中确定的主要版本。

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-major_version_number
```

例如，如果已安装内核模块的主要版本 450，请使用以下命令安装与之匹配的 Nvidia Fabric Manager 版本。

```
$ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
fabricmanager-450
```

c. 启动服务，并确保它在实例启动时自动启动。NV 交换管理需要 Nvidia Fabric 管理器。

```
$ sudo systemctl start nvidia-fabricmanager && sudo systemctl enable nvidia-
fabricmanager
```

8. 确保每次启动实例时均设置 CUDA 路径。

- 对于 bash shell，请将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

- 对于 tcsh shell，请将以下语句添加到 /home/*username*/.cshrc。

```
setenv PATH=/usr/local/cuda/bin:$PATH
```

```
setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/
lib64:$LD_LIBRARY_PATH
```

9. 要验证 Nvidia GPU 驱动程序是否正常运行，请运行以下命令。

```
$ nvidia-smi -q | head
```

此命令应返回有关 Nvidia GPU、Nvidia GPU 驱动程序和 Nvidia CUDA 工具包的信息。

步骤 4：安装 GDRCopy

安装 GDRCopy 以提高 Libfabric 的性能。有关 GDRCopy 的更多信息，请参阅 GDRCopy 存储库。

Amazon Linux 2023 and Amazon Linux 2

安装 GDRCopy

1. 安装所需的依赖项。

```
$ sudo yum -y install dkms rpm-build make check check-devel subunit subunit-
devel
```

2. 下载并解压缩 GDRCopy 程序包。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz ; cd gdrcopy-2.4/packages
```

3. 构建 GDRCopy RPM 程序包。

```
$ CUDA=/usr/local/cuda ./build-rpm-packages.sh
```

4. 安装 GDRCopy RPM 程序包。

```
$ sudo rpm -Uvh gdrcopy-kmod-2.4-1dkms.noarch*.rpm \
&& sudo rpm -Uvh gdrcopy-2.4-1.x86_64*.rpm \
&& sudo rpm -Uvh gdrcopy-devel-2.4-1.noarch*.rpm
```

Ubuntu 24.04 and Ubuntu 22.04

安装 GDRCopy

1. 安装所需的依赖项。

```
$ sudo apt -y install build-essential devscripts debhelper check libsubunit-dev
 fakeroot pkg-config dkms
```

2. 下载并解压缩 GDRCopy 程序包。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz \
&& cd gdrcopy-2.4/packages
```

3. 构建 GDRCopy RPM 程序包。

```
$ CUDA=/usr/local/cuda ./build-deb-packages.sh
```

4. 安装 GDRCopy RPM 程序包。

```
$ sudo dpkg -i gdrdrv-dkms_2.4-1_amd64.*.deb \
&& sudo dpkg -i libgdrapi_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy-tests_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy_2.4-1_amd64.*.deb
```

步骤 5：安装 EFA 软件

在实例上安装支持 EFA 所需的启用 EFA 的内核、EFA 驱动程序、Libfabric、aws-ofi-nccl 插件和 Open MPI 堆栈。

安装 EFA 软件

1. 连接到您启动的实例。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
2. 下载 EFA 软件安装文件。软件安装文件将打包为压缩的 tarball（`.tar.gz`）文件。要下载最新的稳定 版本，请使用以下命令。

   您也可以通过将上面命令中的版本号替换为 `latest` 来获取最新版本。

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.48.0.tar.gz
```

3. （可选）验证 EFA tarball（`.tar.gz`）文件的真实性和完整性。

   建议您执行此操作以验证软件发布者的身份，并检查该文件自发布以来是否已被更改或损坏。如果您不想验证 tarball 文件，请跳过此步骤。

   > **Note**
   >
   > 或者，如果您希望使用 MD5 或 SHA256 校验和验证 tarball 文件，请参阅 使用校验和验证 EFA 安装程序。

   a. 下载公有 GPG 密钥并将其导入到您的密钥环中。

   ```
   $ wget https://efa-installer.amazonaws.com/aws-efa-installer.key && gpg --
   import aws-efa-installer.key
   ```

   该命令应返回一个密钥值。请记下密钥值，因为需要在下一步中使用该值。

   b. 验证 GPG 密钥的指纹。运行以下命令并指定上一步中的密钥值。

   ```
   $ gpg --fingerprint key_value
   ```

   该命令应返回一个与 `4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC` 相同的指纹。如果指纹不匹配，请不要运行 EFA 安装脚本，并联系 Amazon Web Services 支持。

   c. 下载签名文件并验证 EFA tarball 文件的签名。

   ```
   $ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.48.0.tar.gz.sig
    && gpg --verify ./aws-efa-installer-1.48.0.tar.gz.sig
   ```

   下面显示了示例输出。

   ```
   gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
   gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
   gpg: WARNING: This key is not certified with a trusted signature!
   gpg:          There is no indication that the signature belongs to the owner.
   Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
   ```

如果结果包含 Good signature，并且指纹与上一步中返回的指纹相匹配，请继续下一步。如果没有，请不要运行 EFA 安装脚本，并联系 Amazon Web Services 支持。

4. 从压缩的 .tar.gz 文件中提取文件，并导航到提取的目录。

```
$ tar -xf aws-efa-installer-1.48.0.tar.gz && cd aws-efa-installer
```

5. （可选）在安装过程中验证各个软件包的签名。

从 EFA 安装程序 1.48.0 开始，安装程序包括 GPG 签名的各个 RPM 和 DEB 软件包。要在安装过程中验证每个软件包的真实性和完整性，请使用 --check-signatures 标志。启用此标志后，安装程序会首先验证所有软件包的签名，只有在每个软件包都通过验证后才会继续安装。如果有任何软件包未通过验证，安装程序将立即退出，而不安装任何程序。

a. 下载 GPG 公有密钥。

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
```

b. 导出密钥路径。然后，在下一步中，将 --check-signatures 附加到安装命令并使用 sudo -E 来保留环境变量，而不是 sudo。

```
$ export EFA_INSTALLER_KEY=$(pwd)/aws-efa-installer.key
```

在基于 RPM 的系统（Amazon Linux、RHEL、Rocky Linux 和 SUSE）上，安装程序使用 rpm --checksig 来验证每个 RPM。在基于 DEB 的系统（Ubuntu、Debian）上，安装程序使用 GPG 签名验证来验证每个 DEB。

如果任何软件包验证失败，安装将立即中止，从而保护您的系统免受损坏或恶意软件包的侵害。

> **Note**
>
> --check-signatures 标记是可选的。若没有此项验证，安装程序不会执行个人签名验证。

6. 运行 EFA 软件安装脚本。

> **Note**
>
> 如果您完成了前一个可选步骤来设置软件包签名验证，请将 `--check-signatures` 附加到安装命令并使用 `sudo -E`，而不是 `sudo`。例如：`sudo -E ./efa_installer.sh -y --mpi=openmpi4 --check-signatures`。

> **Note**
>
> 从 EFA 1.30.0 开始，默认情况下会同时安装 Open MPI 4.1 和 Open MPI 5。除非需要 Open MPI 5，否则我们建议只安装 Open MPI 4.1。以下命令仅会安装 Open MPI 4.1。如果需要同时安装 Open MPI 4.1 和 Open MPI 5，请移除 `--mpi=openmpi4`。

```
$ sudo ./efa_installer.sh -y --mpi=openmpi4
```

Libfabric 安装在了 `/opt/amazon/efa` 目录中。aws-ofi-nccl 插件安装在了 `/opt/amazon/ofi-nccl` 目录中。Open MPI 安装在了 `/opt/amazon/openmpi` 目录中。

7. 如果 EFA 安装程序提示您重启实例，请执行此操作，然后重新连接到实例。否则，请注销实例，然后重新登录以完成安装。
8. 确认已成功安装 EFA 软件组件。

```
$ fi_info -p efa -t FI_EP_RDM
```

该命令应返回有关 Libfabric EFA 接口的信息。以下示例显示了命令输出。

- `p3dn.24xlarge`（具有单个网络接口）

```
provider: efa
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-rdm
version: 2.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

- `p4d.24xlarge` 和 `p5.48xlarge` 具有多个网络接口

```
provider: efa
fabric: EFA-fe80::c6e:8fff:fef6:e7ff
domain: efa_0-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c34:3eff:feb2:3c35
domain: efa_1-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c0f:7bff:fe68:a775
domain: efa_2-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::ca7:b0ff:fea6:5e99
domain: efa_3-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

步骤 6：安装 NCCL

安装 NCCL。有关 NCCL 的更多信息，请参阅 NCCL 存储库。

安装 NCCL

1. 导航到 /opt 目录。

```
$ cd /opt
```

2. 将官方 NCCL 存储库克隆到实例，然后导航到本地克隆的存储库。

```
$ sudo git clone https://github.com/NVIDIA/nccl.git -b v2.23.4-1 && cd nccl
```

3. 生成并安装 NCCL，然后指定 CUDA 安装目录。

```
$ sudo make -j src.build CUDA_HOME=/usr/local/cuda
```

步骤 7：安装 NCCL 测试

安装 NCCL 测试。NCCL 测试使您能够确认是否已正确安装 NCCL 以及它是否正在按预期运行。有关 NCCL 测试的更多信息，请参阅 nccl-tests 存储库。

安装 NCCL 测试

1. 导航到您的主目录。

```
$ cd $HOME
```

2. 将官方 nccl-tests 存储库克隆到实例，然后导航到本地克隆的存储库。

```
$ git clone https://github.com/NVIDIA/nccl-tests.git && cd nccl-tests
```

3. 将 Libfabric 目录添加到 LD_LIBRARY_PATH 变量。

   - Amazon Linux 2023 和 Amazon Linux 2

   ```
   $ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
   ```

   - Ubuntu 24.04 和 Ubuntu 22.04

   ```
   $ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
   ```

4. 安装 NCCL 测试并指定 MPI、NCCL 和 CUDA 安装目录。

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=/opt/nccl/build CUDA_HOME=/usr/
local/cuda
```

步骤 8：测试 EFA 和 NCCL 配置

运行测试以确保为 EFA 和 NCCL 正确配置临时实例。

测试 EFA 和 NCCL 配置

1. 创建一个主机文件来指定要在其上运行测试的主机。以下命令创建一个名为 my-hosts 的主机文件，该文件包含对实例本身的引用。

   IMDSv2

   ```
   [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H
    "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
   && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/
   meta-data/local-ipv4 >> my-hosts
   ```

   IMDSv1

   ```
   [ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-
   hosts
   ```

2. 运行测试并指定主机文件 (--hostfile) 和要使用的 GPU 数 (-n)。以下命令在实例本身的 8 个 GPU 上运行 all_reduce_perf 测试，并指定以下环境变量。

   - FI_EFA_USE_DEVICE_RDMA=1 – (仅 p4d.24xlarge) 使用设备的 RDMA 功能进行单侧和双侧传输。
   - NCCL_DEBUG=INFO—启用详细的调试输出。您也可以指定 VERSION 以在测试开始时仅输出 NCCL 版本，或指定 WARN 以仅接收错误消息。

   有关 NCCL 测试参数的更多信息，请参阅官方 nccl-tests 存储库中的 NCCL 测试自述文件。

   - p3dn.24xlarge

     ```
     $ /opt/amazon/openmpi/bin/mpirun \
     -x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/
     lib:/opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
     -x NCCL_DEBUG=INFO \
     --hostfile my-hosts -n 8 -N 8 \
     --mca pml ^cm --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to
      none \
     $HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
     ```

   - p4d.24xlarge 和 p5.48xlarge

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_EFA_USE_DEVICE_RDMA=1 \
-x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/
lib:/opt/amazon/openmpi/lib:/opt/amazon/ofi-nccl/lib:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
--hostfile my-hosts -n 8 -N 8 \
--mca pml ^cm --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to
 none \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

3. 打印 NCCL_DEBUG 日志时，您可以确认 EFA 是否作为 NCCL 的底层提供程序处于活动状态。

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Selected Provider is efa*
```

使用 p4d.24xlarge 实例时会显示以下附加信息。

```
ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Running on P4d platform, Setting
 NCCL_TOPO_FILE environment variable to /home/ec2-user/install/plugin/share/aws-
ofi-nccl/xml/p4d-24xl-topo.xml
```

步骤 9：安装机器学习应用程序

在临时实例上安装机器学习应用程序。安装过程因特定的机器学习应用程序而异。有关在 Linux 实例上安装软件的更多信息，请参阅 Manage software on your Amazon Linux 2 instance。

> **Note**
>
> 请参阅机器学习应用程序文档以了解安装说明。

步骤 10：创建启用了 EFA 和 NCCL 的 AMI

在安装所需的软件组件后，您可以创建一个 AMI，然后可以将其重复使用以启动启用了 EFA 的实例。

从临时实例创建 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。

3. 选择您创建的临时实例，然后依次选择 Actions (操作)、Image (映像)、Create image (创建映像)。
4. 对于 Create image (创建映像)，请执行以下操作：
   a. 对于 Image name (映像名称)，为 AMI 输入一个描述性名称。
   b. （可选）对于 Image description (映像描述)，输入 AMI 用途的简要描述。
   c. 选择创建映像。
5. 在导航窗格中，选择 AMIs。
6. 在列表中找到您创建的 AMI。等待状态从 `pending` 更改为 `available`，然后继续下一步。

步骤 11：终止临时实例

现在，已不再需要您启动的临时实例。您可以终止实例以停止产生费用。

终止临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您创建的临时实例，然后依次选择 Actions (操作)、Instance state (实例状态)、Terminate instance (终止实例)。
4. 当系统提示您确认时，选择终止。

步骤 12：在集群置放群组中启动启用了 EFA 和 NCCL 的实例

使用启用了 EFA 的 AMI 以及之前创建的启用了 EFA 的安全组，在集群置放群组中启动启用了 EFA 和 NCCL 的实例。

> **Note**
>
> - 在集群置放群组中启动启用了 EFA 的实例并不是一个绝对要求。不过，我们建议在集群置放群组中运行启用了 EFA 的实例，因为它在单个可用区的低延迟组中启动实例。
> - 为了确保在扩展集群实例时容量可用，您可以为集群置放群组创建容量预留。有关更多信息，请参阅 使用集群置放群组的容量预留。

New console

启动临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在 Name and tags（名称和标签）部分中，提供实例的名称，例如 `EFA-instance`。名称作为资源标签（`Name=`*`EFA-instance`*）分配给实例。
4. 在 Application and OS Images（应用程序和操作系统映像）部分中，请选择 My AMIs（我的 AMI），然后选择您在上一步骤创建的 AMI。
5. 在 Instance type（实例类型）部分中，选择 `p3dn.24xlarge` 或 `p4d.24xlarge`。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 Network settings（网络设置）部分中，请选择 Edit（编辑），然后执行以下操作：
   a. 对于子网，选择要在其中启动实例的子网。如果您未选择子网，则不能启用 EFA 的实例。
   b. 对于 Firewall（security groups）（防火墙（安全组）），请选择 Select existing security group（选择现有安全组），然后选择您在上一步中创建的安全组。
   c. 展开高级网络配置部分。

      对于网络接口 1，选择网卡索引 = 0、设备索引 = 0、接口类型 = 带 ENA 的 EFA。

      （可选）如果您使用的是多卡实例类型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，则对于每个额外的网络接口，选择添加网络接口；对于网卡索引，选择下一个未使用的索引，然后选择设备索引 = 1 和接口类型 = 带 ENA 的 EFA 或仅限 EFA。
8. （可选）在 Storage（存储）部分中，根据需要配置卷。
9. 在 Advanced details（高级详细信息）部分中，对于置放群组名称，选择要在其中启动实例的集群置放群组。如果您需要创建新的集群置放群组，请选择 Create new placement group（创建新置放群组）。
10. 在右侧的 Summary（摘要）面板中，为 Number of instances（实例数量）输入您要启动的启用了 EAA 的实例数量，然后选择 Launch instance（启动实例）。

Old console

在集群置放群组中启动启用了 EFA 和 NCCL 的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择 Launch Instance（启动实例）。
3. 在 Choose an AMI (选择 AMI) 页面上，选择 My AMIs (我的 AMI)，找到之前创建的 AMI，然后选择 Select (选择)。
4. 在选择实例类型页面上，选择 p3dn.24xlarge，然后选择下一步：配置实例详细信息。
5. 在 Configure Instance Details (配置实例详细信息) 页面中，执行以下操作：
   a. 对于 Number of instances (实例的数量)，请输入要启动的启用了 EFA 和 NCCL 的实例数量。
   b. 对于 Network (网络) 和 Subnet (子网)，请选择要在其中启动实例的 VPC 和子网。
   c. 对于 Placement group (置放群组)，请选择 Add instance to placement group (将实例添加到置放群组)。
   d. 对于 Placement group name (置放群组名称)，请选择 Add to a new placement group (添加到新的置放群组)，然后输入一个描述性的置放群组名称。然后，对于 Placement group strategy (置放群组策略)，请选择 cluster (集群)。
   e. 对于 EFA，请选择 Enable (启用)。
   f. 在 Network Interfaces (网络接口) 部分中，为设备 eth0 选择 New network interface (新网络接口)。您可以选择指定主 IPv4 地址以及一个或多个辅助 IPv4 地址。如果在具有关联的 IPv6 CIDR 数据块的子网中启动实例，您可以选择指定主 IPv6 地址以及一个或多个辅助 IPv6 地址。
   g. 选择下一步：添加存储。
6. 在 Add Storage (添加存储) 页面上，除了 AMI 指定的卷（如根设备卷）以外，还要指定要附加到实例的卷。然后，选择 Next: Add Tags (下一步: 添加标签)。
7. 在 Add Tags (添加标签) 页面上，为实例指定标签（例如，便于用户识别的名称），然后选择 Next: Configure Security Group (下一步：配置安全组)。
8. 在 Configure Security Group (配置安全组) 页面上，为 Assign a security group (分配安全组) 选择 Select an existing security group (选择一个现有的安全组)，然后选择之前创建的安全组。
9. 选择审核并启动。

10. 在 Review Instance Launch (核查实例启动) 页面上，检查这些设置，然后选择 Launch (启动) 以选择一个密钥对并启动您的实例。

步骤 13：启用无密码 SSH

要使应用程序能够在集群中的所有实例上运行，您必须启用从领导节点到成员节点的无密码 SSH 访问。领导节点是从中运行应用程序的实例。集群中的其余实例是成员节点。

在集群中的实例之间启用无密码 SSH

1. 在集群中选择一个实例作为领导节点，然后连接到该实例。
2. 在领导节点上禁用 `strictHostKeyChecking` 并启用 `ForwardAgent`。使用首选文本编辑器打开 `~/.ssh/config`，并添加以下内容。

   ```
   Host *
       ForwardAgent yes
   Host *
       StrictHostKeyChecking no
   ```

3. 生成 RSA 密钥对。

   ```
   $ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
   ```

   密钥对在 `$HOME/.ssh/` 目录中创建。
4. 更改领导节点上私有密钥的权限。

   ```
   $ chmod 600 ~/.ssh/id_rsa
   chmod 600 ~/.ssh/config
   ```

5. 使用首选文本编辑器打开 `~/.ssh/id_rsa.pub` 并复制密钥。
6. 对于集群中的每个成员节点，请执行以下操作：

   a. 连接到实例。

   b. 使用首选文本编辑器打开 `~/.ssh/authorized_keys`，并添加之前复制的公有密钥。
7. 要测试无密码 SSH 是否按预期运行，请连接到领导节点并运行以下命令。

   ```
   $ ssh member_node_private_ip
   ```

您应该连接到成员节点，而不会收到输入密钥或密码的提示。

开始使用 EFA 和 NIXL 处理 Amazon EC2 上的推理工作负载

NVIDIA Inference Xfer Library (NIXL) 是一个高吞吐量、低延迟的通信库，专为分解推理工作负载而设计。NIXL 可以与 EFA 和 Libfabric 一起使用，以支持预填充和解码节点之间的 KV 缓存传输，并且它可以在不同存储层之间高效地移动 KV 缓存。有关更多信息，请参阅 NIXL 网站。

要求

- 仅支持 Ubuntu 24.04 和 Ubuntu 22.04 基本 AMI。
- EFA 仅支持 NIXL 1.0.0 及更高版本。

步骤 1：准备启用 EFA 的安全组

EFA 需要使用一个安全组，以允许进出安全组本身的所有入站和出站流量。以下过程创建了一个安全组，该安全组允许所有进出其本身的入站和出站流量，并允许来自任何 IPv4 地址的入站 SSH 流量进行 SSH 连接。

> **Important**
>
> 此安全组仅用于测试目的。对于您的生产环境，建议您创建入站 SSH 规则，该规则仅允许来自您连接的 IP 地址的流量，例如计算机的 IP 地址或本地网络中的一系列 IP 地址。

有关其他场景，请参阅 针对不同使用案例的安全组规则。

创建启用 EFA 的安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Security Groups (安全组)，然后选择 Create Security Group (创建安全组)。
3. 在 Create security group（创建安全组）窗口中，执行以下操作：
   a. 对于安全组名称，请输入一个描述性的安全组名称，例如 `EFA-enabled security group`。
   b. （可选）对于描述，请输入安全组的简要描述。
   c. 对于 VPC，请选择要在其中启动启用了 EFA 的实例的 VPC。

d. 选择创建安全组。

4. 选择您创建的安全组，然后在 Details（详细信息）选项卡上复制 Security group ID（安全组 ID）。
5. 在安全组仍然选中的情况下，依次选择 Actions（操作）、Edit inbound rules（编辑入站规则），然后执行以下操作：

   a. 选择添加规则。
   b. 对于 Type (类型)，请选择 All traffic (所有流量)。
   c. 对于 Source type（源类型），请选择 Custom（自定义）并将您复制的安全组 ID 粘贴到该字段中。
   d. 选择添加规则。
   e. 对于 Type，选择 SSH。
   f. 对于 Source type（源类型），请选择 Anywhere-IPv4。
   g. 选择保存规则。

6. 在安全组仍然选中的情况下，依次选择 Actions（操作）、Edit outbound rules（编辑出站规则），然后执行以下操作：

   a. 选择添加规则。
   b. 对于 Type (类型)，请选择 All traffic (所有流量)。
   c. 对于 Destination type（目标类型），请选择 Custom（自定义）并将您复制的安全组 ID 粘贴到该字段中。
   d. 选择保存规则。

步骤 2：启动临时实例

启动一个临时实例，可用于安装和配置 EFA 软件组件。您使用该实例创建一个启用了 EFA 的 AMI，您可以从中启动启用了 EFA 的实例。

启动临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在 Name and tags（名称和标签）部分中，提供实例的名称，例如 `EFA-instance`。名称作为资源标签（`Name=`*`EFA-instance`*）分配给实例。

4. 在应用程序和操作系统映像部分中，为其中一个支持的操作系统选择 AMI。您也可以在 DLAMI 发行说明页面上选择受支持的 DLAMI。
5. 在实例类型部分中，选择支持的实例类型。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 Network settings（网络设置）部分中，请选择 Edit（编辑），然后执行以下操作：

   a. 对于子网，选择要在其中启动实例的子网。如果您未选择子网，则不能启用 EFA 的实例。

   b. 对于 Firewall（security groups）（防火墙（安全组）），请选择 Select existing security group（选择现有安全组），然后选择您在上一步中创建的安全组。

   c. 展开高级网络配置部分。

      对于网络接口 1，选择网卡索引 = 0、设备索引 = 0、接口类型 = 带 ENA 的 EFA。

      （可选）如果您使用的是多卡实例类型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，则对于每个额外的网络接口，选择添加网络接口；对于网卡索引，选择下一个未使用的索引，然后选择设备索引 1 和接口类型 = 带 ENA 的 EFA 或仅限 EFA。

8. 在 Storage（存储）部分中，根据需要配置卷。

   > **Note**
   >
   > 您必须为 Nvidia CUDA 工具包额外预置 10 到 20 GiB 的存储空间。如果您没有预置足够的存储空间，您将在尝试安装 Nvidia 驱动程序和 CUDA 工具包时收到 `insufficient disk space` 错误。

9. 在右侧的 Summary（摘要）面板中，选择 Launch instance（启动实例）。

> **Important**
>
> 如果您的 AMI 已经包含 Nvidia GPU 驱动程序、CUDA 工具包和 cuDNN，或者您使用的是非 GPU 实例，请跳过步骤 3。

步骤 3：安装 Nvidia GPU 驱动程序、Nvidia CUDA 工具包和 cuDNN

安装 Nvidia GPU 驱动程序、Nvidia CUDA 工具包和 cuDNN

1. 为确保您的所有软件包都处于最新状态，请对您的实例执行快速软件更新。

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

2. 安装在安装 Nvidia GPU 驱动程序和 Nvidia CUDA 工具包时需要的实用程序。

```
$ sudo apt-get install build-essential -y
```

3. 要使用 Nvidia GPU 驱动程序，您必须先禁用 nouveau 开源驱动程序。

   a. 为您当前运行的内核版本安装所需的实用程序和内核标头软件包。

   ```
   $ sudo apt-get install -y gcc make linux-headers-$(uname -r)
   ```

   b. 将 nouveau 添加到/etc/modprobe.d/blacklist.conf 拒绝列表文件。

   ```
   $ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
   blacklist vga16fb
   blacklist nouveau
   blacklist rivafb
   blacklist nvidiafb
   blacklist rivatv
   EOF
   ```

   c. 使用首选文本编辑器打开 /etc/default/grub，并添加以下内容。

   ```
   GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
   ```

   d. 重新生成 Grub 配置。

   ```
   $ sudo update-grub
   ```

4. 重启实例并重新连接到它。

5. 添加 CUDA 存储库并安装 Nvidia GPU 驱动程序、NVIDIA CUDA 工具包和 cuDNN。

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/
machine-learning/repos/ubuntu2004/x86_64/7fa2af80.pub \
&& wget -O /tmp/deeplearning.deb http://developer.download.nvidia.com/
compute/machine-learning/repos/ubuntu2004/x86_64/nvidia-machine-learning-repo-
ubuntu2004_1.0.0-1_amd64.deb \
&& sudo dpkg -i /tmp/deeplearning.deb \
&& wget -O /tmp/cuda.pin https://developer.download.nvidia.com/compute/cuda/repos/
ubuntu2004/x86_64/cuda-ubuntu2004.pin \
```

```
&& sudo mv /tmp/cuda.pin /etc/apt/preferences.d/cuda-repository-pin-600 \
&& sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/
cuda/repos/ubuntu2004/x86_64/3bf863cc.pub \
&& sudo add-apt-repository 'deb http://developer.download.nvidia.com/compute/cuda/
repos/ubuntu2004/x86_64/ /' \
&& sudo apt update \
&& sudo apt install nvidia-dkms-535 \
&& sudo apt install -o Dpkg::Options::='--force-overwrite' cuda-drivers-535 cuda-
toolkit-12-3 libcudnn8 libcudnn8-dev -y
```

6. 重启实例并重新连接到它。
7. （仅限 p4d.24xlarge 和 p5.48xlarge）安装 Nvidia Fabric Manager。

   a. 您必须安装与上一步中安装的 Nvidia 内核模块版本匹配的 Nvidia Fabric Manager 版本。

      运行以下命令以确定 Nvidia 内核模块的版本。

      ```
      $ cat /proc/driver/nvidia/version | grep "Kernel Module"
      ```

      下面是示例输出。

      ```
      NVRM version: NVIDIA UNIX x86_64 Kernel Module  450.42.01  Tue Jun 15 21:26:37
       UTC 2021
      ```

      上述示例中安装了内核模块的主要版本 450。这意味着您需要安装 Nvidia Fabric Manager 版本 450。

   b. 安装 Nvidia Fabric Manager。运行以下命令并指定上一步中确定的主要版本。

      ```
      $ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
      fabricmanager-major_version_number
      ```

      例如，如果已安装内核模块的主要版本 450，请使用以下命令安装与之匹配的 Nvidia Fabric Manager 版本。

      ```
      $ sudo apt install -o Dpkg::Options::='--force-overwrite' nvidia-
      fabricmanager-450
      ```

   c. 启动服务，并确保它在实例启动时自动启动。NV 交换管理需要 Nvidia Fabric 管理器。

```
$ sudo systemctl start nvidia-fabricmanager && sudo systemctl enable nvidia-
fabricmanager
```

8. 确保每次启动实例时均设置 CUDA 路径。

   - 对于 bash shell，请将以下语句添加到 /home/*username*/.bashrc 和 /home/*username*/.bash_profile。

   ```
   export PATH=/usr/local/cuda/bin:$PATH
   export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:
   $LD_LIBRARY_PATH
   ```

   - 对于 tcsh shell，请将以下语句添加到 /home/*username*/.cshrc。

   ```
   setenv PATH=/usr/local/cuda/bin:$PATH
   setenv LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:
   $LD_LIBRARY_PATH
   ```

9. 要验证 Nvidia GPU 驱动程序是否正常运行，请运行以下命令。

   ```
   $ nvidia-smi -q | head
   ```

   此命令应返回有关 Nvidia GPU、Nvidia GPU 驱动程序和 Nvidia CUDA 工具包的信息。

> **Important**
>
> 如果您的 AMI 已经包含 GDRCopy，或者您使用的是非 GPU 实例，请跳过步骤 4。

步骤 4：安装 GDRCopy

安装 GDRCopy，以提高 Libfabric 在基于 GPU 的平台上的性能。有关 GDRCopy 的更多信息，请参阅 GDRCopy 存储库。

安装 GDRCopy

1. 安装所需的依赖项。

```
$ sudo apt -y install build-essential devscripts debhelper check libsubunit-dev
 fakeroot pkg-config dkms
```

2. 下载并解压缩 GDRCopy 程序包。

```
$ wget https://github.com/NVIDIA/gdrcopy/archive/refs/tags/v2.4.tar.gz \
&& tar xf v2.4.tar.gz \
&& cd gdrcopy-2.4/packages
```

3. 构建 GDRCopy DEB 程序包。

```
$ CUDA=/usr/local/cuda ./build-deb-packages.sh
```

4. 安装 GDRCopy DEB 程序包。

```
$ sudo dpkg -i gdrdrv-dkms_2.4-1_amd64.*.deb \
&& sudo dpkg -i libgdrapi_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy-tests_2.4-1_amd64.*.deb \
&& sudo dpkg -i gdrcopy_2.4-1_amd64.*.deb
```

> **Important**
>
> 如果您的 AMI 已经包含最新的 EFA 安装程序，请跳过步骤 5。

步骤 5：安装 EFA 软件

在实例上安装支持 EFA 所需的启用 EFA 的内核、EFA 驱动程序 和 Libfabric 堆栈。

安装 EFA 软件

1. 连接到您启动的实例。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
2. 下载 EFA 软件安装文件。软件安装文件将打包为压缩的 tarball（`.tar.gz`）文件。要下载最新的稳定 版本，请使用以下命令。

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.48.0.tar.gz
```

3. 从压缩的 `.tar.gz` 文件中提取文件，删除压缩包，并导航到提取的目录。

```
$ tar -xf aws-efa-installer-1.48.0.tar.gz && rm -rf aws-efa-installer-1.48.0.tar.gz
 && cd aws-efa-installer
```

4. （可选）在安装过程中验证各个软件包的签名。

   从 EFA 安装程序 1.48.0 开始，安装程序包括 GPG 签名的各个 RPM 和 DEB 软件包。要在安装过程中验证每个软件包的真实性和完整性，请使用 `--check-signatures` 标志。启用此标志后，安装程序会首先验证所有软件包的签名，只有在每个软件包都通过验证后才会继续安装。如果有任何软件包未通过验证，安装程序将立即退出，而不安装任何程序。

   a. 下载 GPG 公有密钥。

      ```
      $ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
      ```

   b. 导出密钥路径。然后，在下一步中，将 `--check-signatures` 附加到安装命令并使用 `sudo -E` 来保留环境变量，而不是 `sudo`。

      ```
      $ export EFA_INSTALLER_KEY=$(pwd)/aws-efa-installer.key
      ```

   在基于 RPM 的系统（Amazon Linux、RHEL、Rocky Linux 和 SUSE）上，安装程序使用 `rpm --checksig` 来验证每个 RPM。在基于 DEB 的系统（Ubuntu、Debian）上，安装程序使用 GPG 签名验证来验证每个 DEB。

   如果任何软件包验证失败，安装将立即中止，从而保护您的系统免受损坏或恶意软件包的侵害。

   > **Note**
   >
   > `--check-signatures` 标记是可选的。若没有此项验证，安装程序不会执行个人签名验证。

5. 运行 EFA 软件安装脚本。

   > **Note**
   >
   > 如果您完成了前一个可选步骤来设置软件包签名验证，请将 `--check-signatures` 附加到安装命令并使用 `sudo -E`，而不是 `sudo`。例如：`sudo -E ./efa_installer.sh -y --check-signatures`。

```
$ sudo ./efa_installer.sh -y
```

Libfabric 安装在了 /opt/amazon/efa 目录中。

6. 如果 EFA 安装程序提示您重启实例，请执行此操作，然后重新连接到实例。否则，请注销实例，然后重新登录以完成安装。
7. 确认已成功安装 EFA 软件组件。

```
$ fi_info -p efa -t FI_EP_RDM
```

该命令应返回有关 Libfabric EFA 接口的信息。以下示例显示了命令输出。

- p3dn.24xlarge（具有单个网络接口）

```
provider: efa
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-rdm
version: 2.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

- p4d.24xlarge 和 p5.48xlarge 具有多个网络接口

```
provider: efa
fabric: EFA-fe80::c6e:8fff:fef6:e7ff
domain: efa_0-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c34:3eff:feb2:3c35
domain: efa_1-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::c0f:7bff:fe68:a775
domain: efa_2-rdm
version: 111.0
type: FI_EP_RDM
```

```
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::ca7:b0ff:fea6:5e99
domain: efa_3-rdm
version: 111.0
type: FI_EP_RDM
protocol: FI_PROTO_EFA
```

步骤 6：安装 NIXL

安装 NIXL。有关 NIXL 的更多信息，请参阅 NIXL 存储库。

Pre-built distributions

使用 PyPI 安装 NIXL

1. 安装所需的依赖项。

```
$ sudo apt install pip
```

2. 安装 NIXL。

```
$ pip install nixl
```

Build from source

从源代码构建和安装 NIXL

1. 安装所需的依赖项。

```
$ sudo apt install cmake pkg-config meson pybind11-dev libaio-dev nvidia-cuda-
toolkit pip libhwloc-dev \
&& pip install meson ninja pybind11
```

2. 导航到您的主目录。

```
$ cd $HOME
```

3. 将官方 NIXL 存储库克隆到实例，然后导航到本地克隆的存储库。

```
$ sudo git clone https://github.com/ai-dynamo/nixl.git && cd nixl
```

4. 构建并安装 NIXL，并指定 Libfabric 安装目录的路径。

```
$ sudo meson setup . nixl --prefix=/usr/local/nixl -Dlibfabric_path=/opt/amazon/
efa
$ cd nixl && sudo ninja && sudo ninja install
```

步骤 7：安装 NIXL Benchmark 并测试 EFA 和 NIXL 配置

安装 NIXL Benchmark 并运行测试，以确保临时实例已为 EFA 和 NIXL 正确配置。通过 NIXL Benchmark，您可以确认 NIXL 已正确安装并按预期运行。有关更多信息，请参阅 nixlbench 存储库。

NIXL Benchmark (nixlbench) 需要 ETCD 来协调客户端和服务器。要将 ETCD 与 NIXL 一起使用，需要 ETCD 服务器和客户端，以及 ETCD CPP API。

Build from Docker

使用 Docker 安装和测试 NIXL Benchmark

1. 将官方 NIXL 存储库克隆到实例，然后导航到 nixlbench 构建目录。

```
$ git clone https://github.com/ai-dynamo/nixl.git
$ cd nixl/benchmark/nixlbench/contrib
```

2. 构建容器

```
$ ./build.sh
```

有关 Docker 构建选项的更多信息，请参阅 nixlbench 存储库。

3. 安装 Docker。

```
$ sudo apt install docker.io -y
```

4. 启动 ETCD 服务器进行协调。

```
$ docker run -d --name etcd-server \
    -p 2379:2379 -p 2380:2380 \
    quay.io/coreos/etcd:v3.5.18 \
```

```
    /usr/local/bin/etcd \
    --data-dir=/etcd-data \
    --listen-client-urls=http://0.0.0.0:2379 \
    --advertise-client-urls=http://0.0.0.0:2379 \
    --listen-peer-urls=http://0.0.0.0:2380 \
    --initial-advertise-peer-urls=http://0.0.0.0:2380 \
    --initial-cluster=default=http://0.0.0.0:2380
```

5. 验证 ETCD 服务器是否正在运行。

```
$ curl -L http://localhost:2379/health
```

预期输出：

```
{"health":"true"}
```

6. 为该实例打开两个终端。在两个终端，运行以下命令来验证安装。该命令在同一个实例上使用 ETCD 服务器，使用 Libfabric 作为后端，并使用 GPU 内存进行操作。

```
$ docker run -it --gpus all --network host nixlbench:latest \
    nixlbench --etcd_endpoints http://localhost:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM \
    --target_seg_type VRAM
```

> **Note**
>
> 对于非 GPU 实例，使用值 DRAM 代替 VRAM。

Build from source

> **Important**
>
> 只有在步骤 6 中选择 “从源代码构建” 时，才遵循此选项卡。

安装 NIXL Benchmark

1. 安装所需的系统依赖项。

```
$ sudo apt install libgflags-dev
```

2. 安装 ETCD 服务器和客户端。

```
$ sudo apt install -y etcd-server etcd-client
```

3. 安装 ETCD CPP API。

   a. 安装 ETCD CPP API 所需的依赖项。

```
$ sudo apt install libboost-all-dev libssl-dev libgrpc-dev libgrpc++-dev
 libprotobuf-dev protobuf-compiler-grpc libcpprest-dev
```

   b. 克隆并安装 ETCD CPP API。

```
$ cd $HOME
$ git clone https://github.com/etcd-cpp-apiv3/etcd-cpp-apiv3.git
$ cd etcd-cpp-apiv3
$ mkdir build && cd build
$ cmake ..
$ sudo make -j$(nproc) && sudo make install
```

4. 构建和安装 nxlbench。

```
$ sudo meson setup . $HOME/nixl/benchmark/nixlbench -Dnixl_path=/usr/local/nixl/
$ sudo ninja && sudo ninja install
```

测试 EFA 和 NIXL 配置

1. 在实例上启动 ETCD 服务器。

```
$ etcd --listen-client-urls "http://0.0.0.0:2379" \
    --advertise-client-urls "http://localhost:2379" &
```

2. 验证 ETCD 服务器是否正在运行。

```
$ curl -L http://localhost:2379/health
```

预期输出：

```
{"health":"true"}
```

3. 为该实例打开两个终端。在两个终端上，完成以下步骤，运行 nixlbench。

   a. 导航到安装 nxlbench 的目录。

   ```
   $ cd /usr/local/nixlbench/bin/
   ```

   b. 运行测试并指定后端、ETCD 服务器的地址和启动器分段类型。以下命令在同一实例上使用 ETCD 服务器，使用 Libfabric 作为后端，并使用 GPU 内存进行操作。环境变量配置如下：

   - NIXL_LOG_LEVEL=INFO — 启用详细的调试输出。您也可以指定 WARN 仅接收错误消息。
   - LD_LIBRARY_PATH — 设置 NIXL 库的路径。

   有关 NIXL Benchmark 参数的更多信息，请参阅官方 nxlbench 存储库中的 NIXLbench 自述文件。

   ```
   $ export NIXL_LOG_LEVEL=INFO
   $ export LD_LIBRARY_PATH=/usr/local/nixl/lib/$(gcc -dumpmachine):
   $LD_LIBRARY_PATH

   $ nixlbench --etcd-endpoints 'http://localhost:2379' \
       --backend 'LIBFABRIC' \
       --initiator_seg_type 'VRAM' \
       --target_seg_type 'VRAM'
   ```

   > **Note**
   >
   > 对于非 GPU 实例，使用值 DRAM 代替 VRAM。

步骤 8：安装机器学习应用程序

在临时实例上安装机器学习应用程序。安装过程因特定的机器学习应用程序而异。

> **Note**
>
> 请参阅机器学习应用程序文档以了解安装说明。

步骤 9：创建启用了 EFA 和 NIXL 的 AMI

在安装所需的软件组件后，您可以创建一个 AMI，然后可以将其重复使用以启动启用了 EFA 的实例。

从临时实例创建 AMI

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您创建的临时实例，然后依次选择 Actions (操作)、Image (映像)、Create image (创建映像)。
4. 对于 Create image (创建映像)，请执行以下操作：
   a. 对于 Image name (映像名称)，为 AMI 输入一个描述性名称。
   b. （可选）对于 Image description (映像描述)，输入 AMI 用途的简要描述。
   c. 选择创建映像。
5. 在导航窗格中，选择 AMIs。
6. 在列表中找到您创建的 AMI。等待状态从 `pending` 更改为 `available`，然后继续下一步。

步骤 10：终止临时实例

现在，已不再需要您启动的临时实例。您可以终止实例以停止产生费用。

终止临时实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您创建的临时实例，然后依次选择 Actions (操作)、Instance state (实例状态)、Terminate instance (终止实例)。
4. 当系统提示您确认时，选择终止。

步骤 11：启动启用 EFA 和 Nixl 的实例

使用在步骤 9 中创建的启用了 EFA 的 AMI 以及在步骤 1 中创建的启用了 EFA 的安全组，启动启用了 EFA 和 NIXL 的实例。

启动启用了 EFA 和 NIXL 的实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch Instances（启动实例）以打开新的启动实例向导。
3. （可选）在 Name and tags（名称和标签）部分中，提供实例的名称，例如 `EFA-instance`。名称作为资源标签（`Name=`*`EFA-instance`*）分配给实例。
4. 在 Application and OS Images（应用程序和操作系统映像）部分中，请选择 My AMIs（我的 AMI），然后选择您在上一步骤创建的 AMI。
5. 在实例类型部分中，选择支持的实例类型。
6. 在 Key pair（密钥对）部分中，选择要用于实例的密钥对。
7. 在 Network settings（网络设置）部分中，请选择 Edit（编辑），然后执行以下操作：
   a. 对于子网，选择要在其中启动实例的子网。如果您未选择子网，则不能启用 EFA 的实例。
   b. 对于防火墙（安全组），选择选择现有安全组，然后选择您在步骤 1 中创建的安全组。
   c. 展开高级网络配置部分。

      对于网络接口 1，选择网卡索引 = 0、设备索引 = 0、接口类型 = 带 ENA 的 EFA。

      （可选）如果您使用的是多卡实例类型，例如 `p4d.24xlarge` 或 `p5.48xlarge`，则对于每个额外的网络接口，选择添加网络接口；对于网卡索引，选择下一个未使用的索引，然后选择设备索引 1 和接口类型 = 带 ENA 的 EFA 或仅限 EFA。
8. （可选）在 Storage（存储）部分中，根据需要配置卷。
9. 在右侧的 Summary（摘要）面板中，为 Number of instances（实例数量）输入您要启动的启用了 EAA 的实例数量，然后选择 Launch instance（启动实例）。

步骤 12：启用无密码 SSH

要使应用程序能够在集群中的所有实例上运行，您必须启用从领导节点到成员节点的无密码 SSH 访问。领导节点是从中运行应用程序的实例。集群中的其余实例是成员节点。

在集群中的实例之间启用无密码 SSH

1. 在集群中选择一个实例作为领导节点，然后连接到该实例。
2. 在领导节点上禁用 strictHostKeyChecking 并启用 ForwardAgent。使用首选文本编辑器打开 ~/.ssh/config，并添加以下内容。

```
Host *
    ForwardAgent yes
Host *
    StrictHostKeyChecking no
```

3. 生成 RSA 密钥对。

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

密钥对在 $HOME/.ssh/ 目录中创建。

4. 更改领导节点上私有密钥的权限。

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. 使用首选文本编辑器打开 ~/.ssh/id_rsa.pub 并复制密钥。
6. 对于集群中的每个成员节点，请执行以下操作：
   a. 连接到实例。
   b. 使用首选文本编辑器打开 ~/.ssh/authorized_keys，并添加之前复制的公有密钥。
7. 要测试无密码 SSH 是否按预期运行，请连接到领导节点并运行以下命令。

```
$ ssh member_node_private_ip
```

您应该连接到成员节点，而不会收到输入密钥或密码的提示。

> **Important**
> 只有在执行步骤 7 时才执行步骤 13。

步骤 13：跨实例测试 EFA 和 NIXL 配置

运行测试，确保实例已为 EFA 和 NIXL 正确配置。

Build from Docker

使用 Docker 跨实例测试 EFA 和 NIXL 配置

1. 选择两台主机来运行 nixlbench 基准测试。使用第一台主机的 IP 地址作为 ETCD 服务器 IP 进行元数据交换。
2. 在主机 1 上启动 ETCD 服务器。

```
$ docker run -d --name etcd-server \
    -p 2379:2379 -p 2380:2380 \
    quay.io/coreos/etcd:v3.5.18 \
    /usr/local/bin/etcd \
    --data-dir=/etcd-data \
    --listen-client-urls=http://0.0.0.0:2379 \
    --advertise-client-urls=http://0.0.0.0:2379 \
    --listen-peer-urls=http://0.0.0.0:2380 \
    --initial-advertise-peer-urls=http://0.0.0.0:2380 \
    --initial-cluster=default=http://0.0.0.0:2380
```

3. 验证 ETCD 服务器是否正在运行。

```
$ curl -L http://localhost:2379/health
```

```
{"health":"true"}
```

4. 在主机 1 上运行 nixlbench 基准测试。

```
$ docker run -it --gpus all --network host nixlbench:latest \
    nixlbench --etcd_endpoints http://localhost:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

5. 在主机 2 上运行 nixlbench 基准测试。

```
$ docker run -it --gpus all --network host nixlbench:latest \
    nixlbench --etcd_endpoints http://ETCD_SERVER_IP:2379 \
    --backend LIBFABRIC \
```

```
    --initiator_seg_type VRAM
```

Build from source

> **Important**
>
> 只有在步骤 6 中选择 “从源代码构建” 时，才遵循此选项卡。

跨实例测试 EFA 和 NIXL 配置

1. 选择两台主机来运行 nixlbench 基准测试。使用第一台主机的 IP 地址作为 ETCD 服务器 IP 进行元数据交换。
2. 在主机 1 上启动 ETCD 服务器。

```
$ etcd --listen-client-urls "http://0.0.0.0:2379" \
    --advertise-client-urls "http://localhost:2379" &
```

3. 验证 ETCD 服务器是否正在运行。

```
$ curl -L http://localhost:2379/health
```

```
{"health":"true"}
```

4. 在主机 1 上运行 nixlbench 基准测试。

```
$ export NIXL_LOG_LEVEL=INFO
$ export LD_LIBRARY_PATH=$HOME/nixl/lib/x86_64-linux-gnu:$LD_LIBRARY_PATH

$ nixlbench \
    --etcd-endpoints http://localhost:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

5. 在主机 2 上运行 nixlbench 基准测试。

```
$ export NIXL_LOG_LEVEL=INFO
$ export LD_LIBRARY_PATH=$HOME/nixl/lib/x86_64-linux-gnu:$LD_LIBRARY_PATH

$ nixlbench \
```

```
    --etcd-endpoints http://ETCD_SERVER_IP:2379 \
    --backend LIBFABRIC \
    --initiator_seg_type VRAM
```

步骤 14：测试在 vlLM 上提供的分解推理（可选）

安装 NIXL 后，您可以通过 LLM 推理和服务框架（例如 vLLM、SGLang 和 TensorRT-LLM）来使用 NIXL。

使用 vLLM 为推理工作负载提供服务

1. 安装 vLLM。

```
$ pip install vllm
```

2. 使用 NIXL 启动 vLLM 服务器。以下示例命令为 NIXL 握手连接、KV 连接器、KV 角色和传输后端创建一个预填充（生产者）和一个解码（使用者）实例。有关详细的示例和脚本，请参阅 NIXLConnector 使用指南。

   要将 NIXL 与 EFA 配合使用，请根据您的设置和使用案例设置环境变量。

   - 生产者（预填器）配置

```
$ vllm serve your-application \
    --port 8200 \
    --enforce-eager \
    --kv-transfer-config
 '{"kv_connector":"NixlConnector","kv_role":"kv_both","kv_buffer_device":"cuda","kv_conne
{"backends":["LIBFABRIC"]}}'
```

   - 使用者（解码器）配置

```
$ vllm serve your-application \
    --port 8200 \
    --enforce-eager \
    --kv-transfer-config
 '{"kv_connector":"NixlConnector","kv_role":"kv_both","kv_buffer_device":"cuda","kv_conne
{"backends":["LIBFABRIC"]}}'
```

   前面的示例配置设置了以下内容：

- `kv_role` 到 `kv_both`，这就实现了对称功能，连接器既可以充当生产者，也可以充当使用者。这为没有预先确定角色区分的实验设置和方案提供了灵活性。
- `kv_buffer_device` 到 `cuda`，这允许使用 GPU 内存。
- NIXL 后端到 `LIBFABRIC`，这使得 NIXL 流量能够通过 EFA 传输。

使用多网卡最大化 Amazon EC2 实例上的网络带宽

许多支持 EFA 的实例类型也有多个网卡。有关更多信息，请参阅 网卡。如果您计划将 EFA 用于其中一种实例类型，建议使用以下基本配置：

- 对于主网络接口（网卡索引 0、设备索引 0），请创建一个 ENA 接口。您不能使用仅限 EFA 的网络接口作为主网络接口。
- 如果网卡索引 0 支持 EFA，则为网卡索引 0 设备索引 1 创建一个仅限 EFA 的网络接口。
- 对于其他的每个网络接口，请将下一个未使用的网卡索引、设备索引 0 用于仅限 EFA 的网络接口，以及/或者将设备索引 1 用于 ENA 网络接口，具体取决于您的使用案例，例如 ENA 带宽要求或 IP 地址空间。有关示例使用案例，请参阅 P5 和 P5e 实例的 EFA 配置。

> **Note**
>
> P5 实例需要以特定方式配置网络接口，以实现最大化网络带宽。有关更多信息，请参阅 P5 和 P5e 实例的 EFA 配置。

以下示例显示如何根据这些建议启动实例。

Instance launch

要在实例启动期间使用启动实例向导指定 EFA

1. 在网络设置部分中，选择编辑。
2. 展开高级网络配置。
3. 对于主网络接口（网卡索引 0、设备索引 0），请创建一个 ENA 接口。您不能使用仅限 EFA 的网络接口作为主网络接口。
4. 如果网卡索引 0 支持 EFA，则为网卡索引 0 设备索引 1 创建一个仅限 EFA 的网络接口。

5. 对于其他的每个网络接口，请将下一个未使用的网卡索引、设备索引 0 用于仅限 EFA 的网络接口，以及/或者将设备索引 1 用于 ENA 网络接口，具体取决于您的使用案例，例如 ENA 带宽要求或 IP 地址空间。有关示例使用案例，请参阅 P5 和 P5e 实例的 EFA 配置。

要在实例启动期间使用 run-instances 命令指定 EFA

对于 --network-interfaces，请指定所需的网络接口数量。对于主网络接口，指定 NetworkCardIndex=0、DeviceIndex=0 和 InterfaceType=interface。如果网卡索引 0 支持 EFA，请指定 NetworkCardIndex=0、DeviceIndex=1 和 InterfaceType=efa-only。对于任何其他网络接口，请将 NetworkCardIndex 指定为下一个未使用的索引，将 InterfaceType=efa-only 指定为 DeviceIndex=0，以及/或者将 InterfaceType=interface 指定为 DeviceIndex=1。

以下示例命令片段显示具有 32 个 EFA 设备和一个 ENA 设备的请求。

```
$ aws ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami-0abcdef1234567890 \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
 \
"NetworkCardIndex=0,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
```

```
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=17,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=18,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=19,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=20,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=21,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=22,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=23,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=24,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=25,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=26,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=27,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=28,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=29,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
"NetworkCardIndex=30,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only" \
```

```
"NetworkCardIndex=31,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType
only"
...
```

Launch templates

要使用 Amazon EC2 控制台将 EFA 添加到启动模板

1. 在网络设置中，展开高级网络配置。
2. 对于主网络接口（网卡索引 0、设备索引 0），请创建一个 ENA 接口。您不能使用仅限 EFA 的网络接口作为主网络接口。
3. 如果网卡索引 0 支持 EFA，则为网卡索引 0 设备索引 1 创建一个仅限 EFA 的网络接口。
4. 对于其他的每个网络接口，请将下一个未使用的网卡索引、设备索引 0 用于仅限 EFA 的网络接口，以及/或者将设备索引 1 用于 ENA 网络接口，具体取决于您的使用案例，例如 ENA 带宽要求或 IP 地址空间。有关示例使用案例，请参阅 P5 和 P5e 实例的 EFA 配置。

要使用 create-launch-template 命令将 EFA 添加到启动模板

对于 NetworkInterfaces，请指定所需的网络接口数量。对于主网络接口，指定 NetworkCardIndex=0、DeviceIndex=0 和 InterfaceType=interface。如果网卡索引 0 支持 EFA，请指定 NetworkCardIndex=0、DeviceIndex=1 和 InterfaceType=efa-only。对于任何其他网络接口，请将 NetworkCardIndex 指定为下一个未使用的索引，将 InterfaceType=efa-only 指定为 DeviceIndex=0，以及/或者将 InterfaceType=interface 指定为 DeviceIndex=1。

以下代码段显示可能的 32 个网络接口中 3 个网络接口的示例。

```
"NetworkInterfaces":[
{
  "NetworkCardIndex":0,
  "DeviceIndex":0,
  "InterfaceType": "interface",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 0,
```

```
  "DeviceIndex": 1,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 1,
  "DeviceIndex": 0,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 2,
  "DeviceIndex": 0,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
},
{
  "NetworkCardIndex": 3,
  "DeviceIndex": 0,
  "InterfaceType": "efa-only",
  "AssociatePublicIpAddress":false,
  "Groups":[
    "security_group_id"
  ],
  "DeleteOnTermination":true
}
...
```

P5 和 P5e 实例的 EFA 配置

p5.48xlarge 和 p5e.48xlarge 实例支持 32 个网卡，拥有 3200 Gbps 的总网络带宽容量，其中高达 800 Gbps 可用于 IP 网络流量。由于 EFA 和 IP 网络流量共享相同的底层资源，因此一方使用的带宽会减少另一方可用的带宽。这意味着，只要总带宽不超过 3,200 Gbps 且 IP 带宽不超过 800 Gbps，就可以按任意组合在 EFA 流量和 IP 流量之间分配网络带宽。例如，如果您使用 400 Gbps 的 IP 带宽，则可以同时获得高达 2800 Gbps 的 EFA 带宽。

使用案例 1：保存 IP 地址并避免潜在的 Linux IP 问题

此配置通过一个私有 IP 地址提供高达 3,200 Gbps 的 EFA 网络带宽和高达 100 Gbps 的 IP 联网带宽。此配置还有助于避免潜在的 Linux IP 问题，例如不允许自动分配公有 IP 地址和 IP 路由挑战（主机名到 IP 地址的映射问题和源 IP 地址不匹配），如果实例有多个网络接口，则可能会出现这些问题。

- 对于主网络接口（网卡索引 0、设备索引 0），请使用一个 ENA 接口。
- 对于网卡索引 0、设备索引 1，请创建一个仅限 EFA 的网络接口。
- 对于其余的网络接口（网卡索引 1-31、设备索引 0），请使用仅限 EFA 的网络接口。

使用案例 2：最大 EFA 和 IP 网络带宽

此配置通过 8 个私有 IP 地址提供高达 3,200 Gbps 的 EFA 网络带宽和高达 800 Gbps 的 IP 联网带宽。您无法使用此配置自动分配公有 IP 地址。但是，您可以在启动后将弹性 IP 地址连接到主网络接口（网卡索引 0、设备索引 0）以实现互联网连接。

- 对于主网络接口（网卡索引 0、设备索引 0），使用 EFA 网络接口。
- 对于其余接口，执行以下操作：
  - 在网卡索引 0、设备索引 1 上指定仅限 EFA 的网络接口，对于网卡接口 1、2 和 3，请使用设备索引 0。
  - 在以下每个网卡索引子集中，指定一个 ENA 网络接口和四个仅限 EFA 的网络接口；对仅限 EFA 的网络接口，请使用设备索引 0：
    - [4、5、6、7]
    - [8、9、10、11]
    - [12、13、14、15]
    - [16、17、18、19]
    - [20、21、22、23]
    - [24、25、26、27]

- [28、29、30、31]

下面的示例对此配置进行说明：

```
$ aws --region $REGION ec2 run-instances \
 --instance-type p5.48xlarge \
 --count 1 \
 --key-name key_pair_name \
 --image-id ami_id \
 --network-interfaces
 "NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=0,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
```

```
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=17,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=18,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=19,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=20,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=20,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=21,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=22,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=23,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=24,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=24,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=25,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=26,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=27,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=28,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=28,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=29,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
```

```
"NetworkCardIndex=30,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=31,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only"
...
```

P6-B200 实例的 EFA 配置

P6-B200 实例的总网络带宽容量为 3200 Gbps，其中高达 1600 Gbps 可用于 ENA。这些实例有 8 个 GPU 和 8 个网卡，其中每个网卡支持高达 400 Gbps 的 EFA 带宽和 200 Gbps 的 ENA 带宽。由于 EFA 和 ENA 流量共享相同的底层资源，因此一方使用的带宽会减少另一方可用的带宽。

使用案例 1：保存 IP 地址

此配置每个实例至少消耗一个私有 IP 地址并支持高达 3200 Gbps 的 EFA 带宽和高达 200 Gbps 的 ENA 带宽。

- 对于主网络接口（网卡索引 0、设备索引 0），请使用一个 ENA 接口。
- 对于网卡索引 0、设备索引 1，请创建一个仅限 EFA 的网络接口。
- 对于其余 7 个网卡（网卡索引 1-7，设备索引 0），使用仅限 EFA 的网络接口。

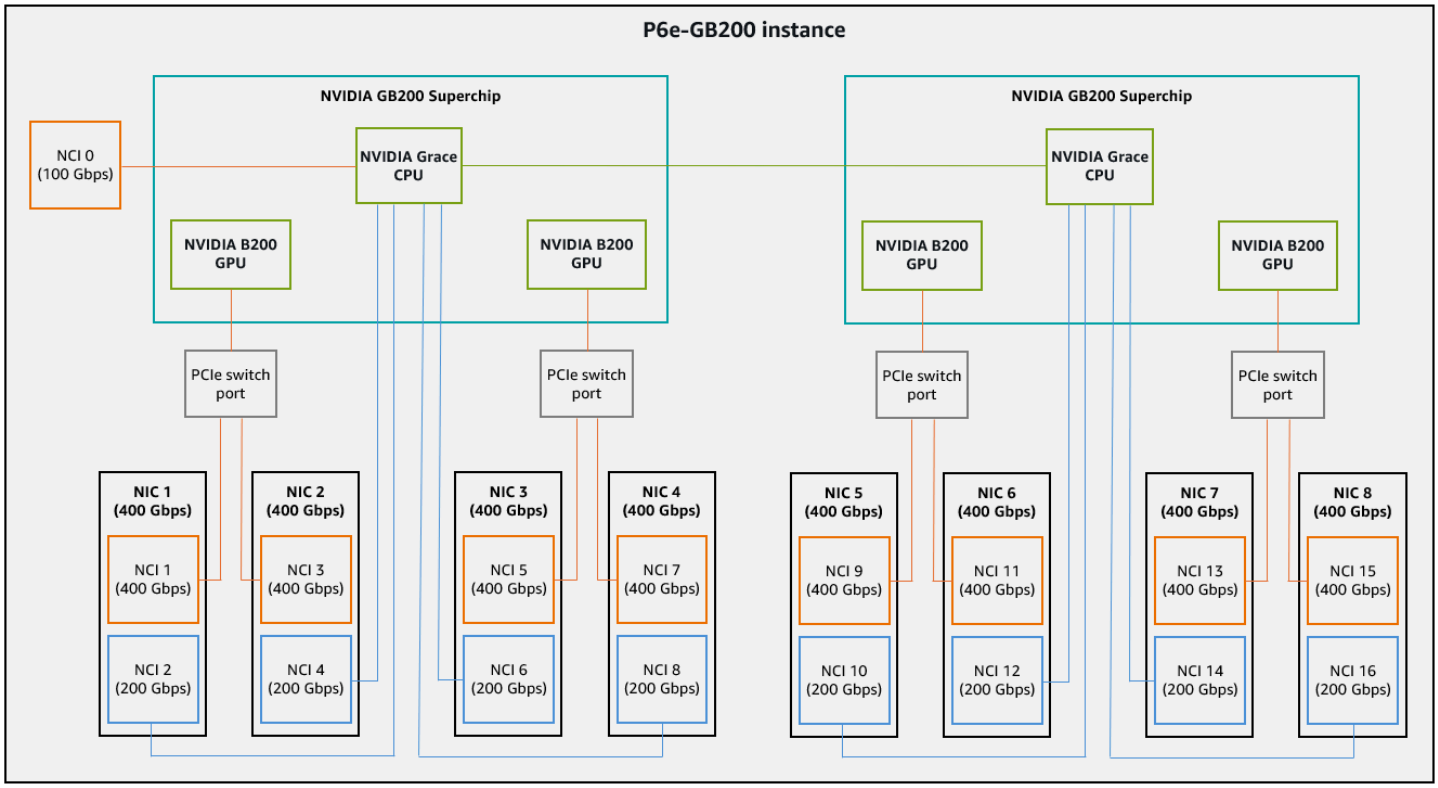
使用案例 2：最大 EFA 和 ENA 带宽

此配置每个实例至少消耗 8 个私有 IP 地址并支持高达 3200 Gbps 的 EFA 带宽和高达 1600 Gbps 的 ENA 带宽。

- 对于主网络接口（网卡索引 0、设备索引 0），请使用一个 ENA 接口。
- 对于网卡索引 0、设备索引 1，请创建一个仅限 EFA 的网络接口。
- 对于其余 7 个网卡（网卡索引 1-7），请在设备索引 0 上创建一个仅限 EFA 的网络接口，并在设备索引 1 上创建一个 ENA 网络接口。

P6e-GB200 实例的 EFA 配置

P6e-GB200 实例最多可以配置 17 个网卡。下图显示 P6e-GB200 实例的物理网络接口卡（NIC）布局以及网卡索引（NCI）的映射。



主 NCI（索引 0）支持高达 100 Gbps 的 ENA 带宽。具有以下索引的 NCI 支持仅限 EFA 的网络接口和 400 Gbps 的 EFA 带宽：[1、3、5、7、9、11、13、15]。具有以下索引的 NCI 支持高达 200 Gbps 的 ENA 或 EFA 带宽：[2、4、6、8、10、12、14、16]。

以下组中的 NCI 共享主机上的底层物理 NIC：

- [1 和 2]
- [3 和 4]
- [5 和 6]
- [7 和 8]
- [9 和 10]
- [11 和 12]
- [13 和 14]
- [15 和 16]

每个物理 NIC 支持高达 400 Gbps 的带宽。由于这些组中的 NCI 共享相同的底层物理 NIC，因此一个 NCI 使用的带宽会减少另一个 NCI 可用的带宽。例如，如果 NCI 2 使用 200 Gbps 的 ENA 带宽，则 NCI 1 可以同时使用高达 200 Gbps 的 EFA 带宽。

主机上的每个底层 GPU 都可以通过以下几对 NCI 直接发送流量：

- [1 和 3]
- [5 和 7]
- [9 和 11]
- [13 和 15]

每个 GPU 支持高达 400 Gbps 的 EFA 带宽。由于这些组中的网卡共享相同的 GPU，因此一个网卡使用的带宽会减少另一个网卡可用的带宽。例如，如果 NCI 1 使用 200 Gbps 的 EFA 带宽，则 NCI 3 可以同时使用高达 200 Gbps 的 EFA 带宽。因此，为了实现最高 EFA 性能，我们建议您执行以下操作之一，以实现总计 1600 Gbps 的 EFA 带宽：

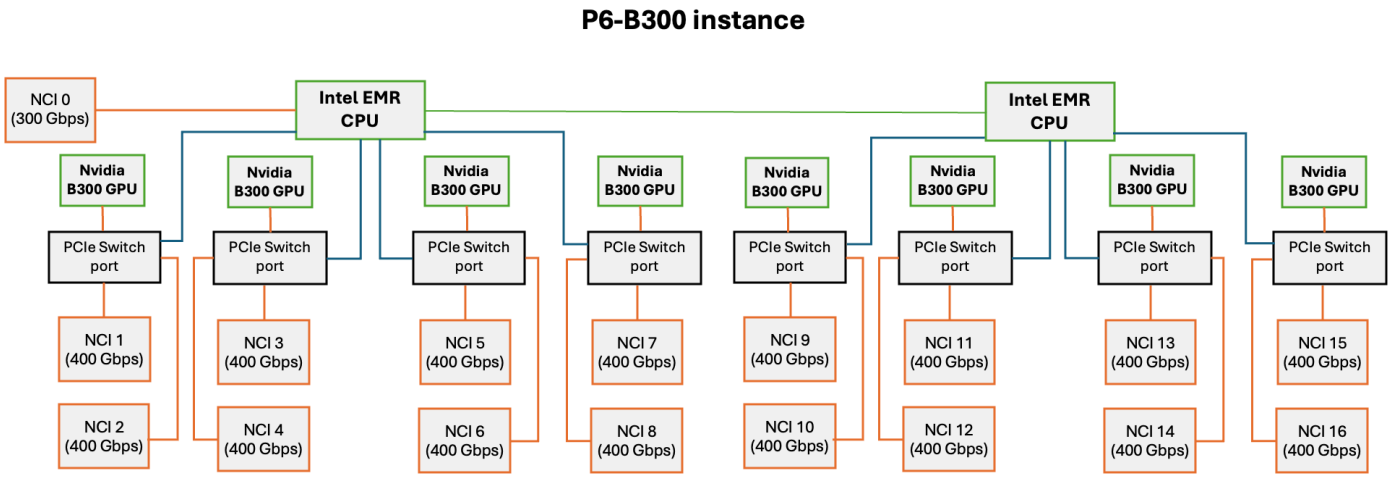
- 在每个组中仅向一个 NCI 添加仅限 EFA 的网络接口，以实现每个网络接口 400 Gbps（4 个 EFA 网络接口 x 400 Gbps）。
- 向每个组中的每个 NCI 添加仅限 EFA 的网络接口，以实现每个网络接口 200 Gbps（8 个 EFA 网络接口 x 200 Gbps）。

例如，以下配置使用每个 NCI 组中一个仅限 EFA 的网络接口提供高达 1600 Gbps 的 EFA 带宽，并仅使用主 NCI（索引 0）提供高达 100 Gbps 的 ENA 网络带宽。

- 对于主 NCI（网卡索引 0、设备索引 0），使用 EFA 网络接口。
- 将仅限 EFA 的网络接口添加到以下位置：
  - NCI 1，设备索引 0
  - NCI 5，设备索引 0
  - NCI 9，设备索引 0
  - NCI 13，设备索引 0

P6-B300 实例的 EFA 配置

对于 EFA 流量，P6-B300 实例的总网络带宽容量高达 6400 Gbps，对于 ENA 流量高达 3870 Gbps。它们有 8 个 GPU 和 17 个网卡，其中主网卡仅支持带宽高达 350 Gbps 的 ENA 网络接口。辅助网卡支持高达 400 Gbps 的 EFA 带宽和高达 220 Gbps 的 ENA 带宽。由于 EFA 和 ENA 流量共享相同的底层资源，因此一方使用的带宽会减少另一方可用的带宽。



使用案例 1：保存 IP 地址

此配置每个实例至少消耗一个私有 IP 地址并支持高达 6400 Gbps 的 EFA 带宽和高达 350 Gbps 的 ENA 带宽。

- 对于主网络接口（网卡索引 0、设备索引 0），使用 EFA 网络接口。
- 对于其余网卡（网卡索引 1-16，设备索引 0），请使用仅限 EFA 的网络接口。

```
--network-interfaces \
"NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
```

```
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only"
```

使用案例 2：最大 EFA 和 ENA 带宽

此配置每个实例至少消耗 17 个私有 IP 地址并支持高达 6400 Gbps 的 EFA 带宽和高达 3870 Gbps 的 ENA 带宽。

- 对于主网络接口（网卡索引 0、设备索引 0），使用 EFA 网络接口。
- 对于其余的网卡，请创建一个仅限 EFA 的接口（网卡索引 1-16，设备索引 0）和一个 ENA 接口（网卡索引 1-16，设备索引 1）。

```
--network-interfaces \
"NetworkCardIndex=0,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=1,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=2,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=3,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=4,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=5,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=6,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
```

```
"NetworkCardIndex=7,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=8,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=9,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=efa
only" \
"NetworkCardIndex=10,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=11,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=12,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=13,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=14,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=15,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=16,DeviceIndex=0,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=ef
only" \
"NetworkCardIndex=1,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=2,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=3,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=4,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=5,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=6,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=7,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=8,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=9,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=int
 \
"NetworkCardIndex=10,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=11,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=12,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
```

```
"NetworkCardIndex=13,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=14,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=15,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
 \
"NetworkCardIndex=16,DeviceIndex=1,Groups=security_group_id,SubnetId=subnet_id,InterfaceType=in
```

创建 Elastic Fabric Adapter 并将其附加到 Amazon EC2 实例

您可以创建 EFA 并将其附加到 Amazon EC2 实例，就像 Amazon EC2 中的任何其他弹性网络接口一样。不过，与弹性网络接口不同，无法将 EFA 附加到处于 `running` 状态的实例，也无法将其从处于该状态的实例中分离。

注意事项

- 您可以更改与 EFA 关联的安全组。要启用操作系统绕过功能，EFA 必须是一个安全组的成员，以允许进出安全组本身的所有入站和出站流量。有关更多信息，请参阅 步骤 1：准备启用 EFA 的安全组。

  您可以使用更改与弹性网络接口关联的安全组的相同方式更改与 EFA 关联的安全组。有关更多信息，请参阅 the section called “修改网络接口属性”。
- 您可以使用将 IP 地址分配给弹性网络接口的相同方式将弹性 IP (IPv4) 和 IPv6 地址分配给 EFA（带 ENA 的 EFA）。有关更多信息，请参阅管理 IP 地址。

  您不能为仅限 EFA 的网络接口分配 IP 地址。

任务

- 创建 EFA
- 将 EFA 附加到停止的实例
- 在启动实例时附加 EFA
- 将 EFA 添加到启动模板

创建 EFA

您可以在 VPC 上的子网中创建 EFA。在创建 EFA 后，您无法将其移动到另一个子网，并且只能将其附加到同一可用区中的已停止实例。

Console

创建 EFA（带 ENA 的 EFA 或仅限 ENA）网络接口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Network Interfaces（网络接口），然后选择 Create network interface（创建网络接口）。
3. 对于 Description (描述)，请输入一个描述性的 EFA 名称。
4. 对于 Subnet (子网)，请选择要在其中创建 EFA 的子网。
5. 接口类型，选择以下选项之一：
   - 带 ENA 的 EFA：创建同时支持 ENA 和 EFA 设备的网络接口。
   - 仅限 EFA：创建仅支持 EFA 设备的网络接口。
6. （仅适用于带 ENA 的 EFA）为网络接口配置 IP 地址和前缀分配。您可以分配的 IP 地址和前缀类型取决于所选的子网。对于仅限 IPv4 的子网，您只能分配 IPv4 IP 地址和前缀。对于仅限 IPv6 的子网，您只能分配 IPv6 IP 地址和前缀。对于双堆栈子网，可同时分配 IPv4 和 IPv6 IP 地址和前缀。

   > **Note**
   >
   > 您不能为仅限 EFA 的网络接口分配 IP 地址。

   a. 对于私有 IPv4 地址和/或 IPv6 地址，选择自动分配以使 Amazon EC2 从选定子网中自动分配 IP 地址，或选择自定义以手动指定要分配的 IP 地址。

   b. 如果您分配 IPv6 地址，则可以选择启用分配主 IPv6 IP。此操作将为网络接口分配一个主 IPv6 全局单播地址（GUA）。分配主要 IPv6 地址使您能够避免中断实例或 ENI 的流量。有关更多信息，请参阅 IPv6 地址。

   c. 对于 IPv4 前缀委派和/或 IPv6 前缀委派，选择自动分配以使 Amazon EC2 自动从子网的 CIDR 数据块中分配前缀，或选择自定义以从子网的 CIDR 数据块中手动指定前缀。如果您指定前缀，则 Amazon 会验证其是否已分配给其他资源。有关更多信息，请参阅 Amazon EC2 网络接口的前缀委派。

   d. （可选）配置空闲连接跟踪超时设置。有关更多信息，请参阅 空闲连接跟踪超时。
      - TCP 建立超时：处于已建立状态的空闲 TCP 连接的超时时间（秒）。最小值：60 秒。最大值：432000 秒（5 天）。默认值：432000 秒。建议值：小于 432000 秒。

- UDP 超时：空闲 UDP 流的超时时间（秒），这些流仅在单个方向或单个请求-响应事务中看到流量。最小值：30 秒。最大值：60 秒。默认值：30 秒。
- UDP 流超时：空闲 UDP 流的超时时间（秒），这些流被归类为已看到多个请求-响应事务的流。最小值：60 秒。最大值：180 秒（3 分钟）。默认值：180 秒。

7. 对于 Security groups，选择一个或多个安全组。
8. 选择创建网络接口。

Amazon CLI

创建 EFA

使用 create-network-interface 命令。对于 `--interface-type`，针对 EFA 网络接口指定 `efa` 或针对仅限 EFA 的网络接口指定 `efa-only`。

```
aws ec2 create-network-interface \
    --subnet-id subnet-0abcdef1234567890 \
    --interface-type efa \
    --description "my efa"
```

PowerShell

创建 EFA

使用 New-EC2NetworkInterface cmdlet。对于 `-InterfaceType`，针对 EFA 网络接口指定 `efa` 或针对仅限 EFA 的网络接口指定 `efa-only`

```
New-EC2NetworkInterface `
    -SubnetId subnet-0abcdef1234567890 `
    -InterfaceType efa `
    -Description "my efa"
```

将 EFA 附加到停止的实例

您可以将 EFA 附加到处于 `stopped` 状态的任何支持的实例。您无法将 EFA 附加到处于 `running` 状态的实例。有关支持的实例类型的更多信息，请参阅支持的实例类型。

您可以使用将网络接口连接到实例的相同方式将 EFA 连接到实例。有关更多信息，请参阅附加网络接口。

在启动实例时附加 EFA

Amazon CLI

在启动实例时附加现有的 EFA

使用 run-instances 命令和 --network-interfaces 选项。对于主网络接口，指定 EFA 网络接口以及 NetworkCardIndex=0、DeviceIndex=0。要附加多个 EFA 网络接口，请参阅最大化网络带宽。

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    NetworkInterfaceId=eni-1234567890abcdef0, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

在启动实例时附加新的 EFA

使用 run-instances 命令和 --network-interfaces 选项。对于主网络接口，使用 NetworkCardIndex=0、DeviceIndex=0 和 InterfaceType=efa。如果您要附加多个 EFA 网络接口，请参阅 最大化网络带宽。

```
--network-interfaces "NetworkCardIndex=0, \
    DeviceIndex=0, \
    InterfaceType=efa, \
    Groups=sg-1234567890abcdef0, \
    SubnetId=subnet-0abcdef1234567890"
```

PowerShell

在启动实例时附加现有的 EFA

将 New-EC2Instance cmdlet 与 -NetworkInterfaces 参数结合使用。

```
-NetworkInterface $networkInterface
```

按如下方式定义网络接口。

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.NetworkInterfaceId = "eni-1234567890abcdef0"
```

```
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

在启动实例时附加新的 EFA

将 New-EC2Instance cmdlet 与 `-NetworkInterfaces` 参数结合使用。

```
-NetworkInterface $networkInterface
```

按如下方式定义网络接口。

```
$networkInterface = New-Object
 Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$networkInterface.DeviceIndex = 0
$networkInterface.InterfaceType = "efa"
$networkInterface.Groups = @("sg-1234567890abcdef0")
$networkInterface.SubnetId = "subnet-0abcdef1234567890"
```

将 EFA 添加到启动模板

您可以创建一个启动模板，其中包含启动启用了 EFA 的实例所需的配置信息。您可以在启动模板中同时指定 EFA 和仅限 EFA 的网络接口。要创建启用了 EFA 的启动模板，请创建新的启动模板并指定支持的实例类型、启用了 EFA 的 AMI 以及启用了 EFA 的安全组。对于 `NetworkInterfaces`，请指定要附加的 EFA 网络接口。对于主网络接口，使用 `NetworkCardIndex=0`、`DeviceIndex=0` 和 `InterfaceType=efa`。如果您要附加多个 EFA 网络接口，请参阅 使用多网卡最大化 Amazon EC2 实例上的网络带宽。

您可以利用启动模板通过其他 Amazon 服务（如 Amazon Batch 或 Amazon ParallelCluster）启动启用了 EFA 的实例。

有关创建启动模板的更多信息，请参阅创建 Amazon EC2 启动模板。

将 EFA 与 Amazon EC2 实例分离并删除 EFA

您可以将 EFA 与 Amazon EC2 实例分离，然后将其删除，就像在 Amazon EC2 中删除任何其他弹性网络接口一样。

分离 EFA

要从实例中分离 EFA，您必须先停止该实例。您无法从处于运行状态的实例中分离 EFA。

您可以使用从实例中分离弹性网络接口的相同方式从实例中分离 EFA。有关更多信息，请参阅 分离网络接口。

删除 EFA

要删除 EFA，您必须先将其从实例中分离。在附加到实例时，您无法删除 EFA。

您可以使用删除弹性网络接口的相同方式删除 EFAs。有关更多信息，请参阅 删除网络接口。

监控 Amazon EC2 上的 Elastic Fabric Adapter

您可以使用以下功能监控 Elastic Fabric Adapters 的性能。

主题

- Amazon EC2 实例的 EFA 驱动程序指标
- Amazon VPC 流日志
- Amazon CloudWatch

Amazon EC2 实例的 EFA 驱动程序指标

Elastic Fabric Adapter（EFA）驱动程序实时发布来自已挂载 EFA 接口的实例的多个指标。您可以使用这些指标来排查应用程序性能和网络问题、为工作负载选择合适的集群大小、主动计划扩缩活动以及对应用程序进行基准测试，以确定是否最大限度利用了实例上可用的 EFA 性能。

主题

- 可用的 EFA 驱动程序指标
- 检索实例的 EFA 驱动程序指标

可用的 EFA 驱动程序指标

ENA 驱动程序会实时向实例发布以下指标。这些指标提供了自实例启动或上次驱动程序重置以来的累计错误数、连接事件数以及所挂载 EFA 设备发送、接收、重新传输或丢弃的数据包数。

指标	说明	支持的实例类型
`tx_bytes`	传输的字节数。 单位：字节	支持 EFA 的所有实例类型

指标	说明	支持的实例类型
rx_bytes	收到的字节数。 单位：字节	支持 EFA 的所有实例类型
tx_pkts	传输的数据包数。 单位：计数	支持 EFA 的所有实例类型
rx_pkts	收到的数据包数。 单位：计数	支持 EFA 的所有实例类型
rx_drops	收到但随后丢弃的数据包数。 单位：计数	支持 EFA 的所有实例类型
send_bytes	使用发送操作发送的字节数。 单位：字节	支持 EFA 的所有实例类型
recv_bytes	通过发送操作收到的字节数。 单位：字节	支持 EFA 的所有实例类型
send_wrs	使用发送操作发送的数据包数。 单位：计数	支持 EFA 的所有实例类型
recv_wrs	通过发送操作收到的数据包数。 单位：计数	支持 EFA 的所有实例类型

指标	说明	支持的实例类型
rdma_write_wrs	已完成的 rdma 写入操作数。 单位：计数	支持 EFA 的所有实例类型
rdma_read_wrs	已完成的 rdma 读取操作数。 单位：计数	支持 EFA 的所有实例类型
rdma_write_bytes	其他实例通过 rdma 写入操作写入的字节数。 单位：字节	支持 EFA 的所有实例类型
rdma_read_bytes	通过 rdma 读取操作收到的字节数。 单位：字节	支持 EFA 的所有实例类型
rdma_write_wr_err	出现本地或远程错误的 rdma 写入操作数。 单位：计数	支持 EFA 的所有实例类型
rdma_read_wr_err	出现本地或远程错误的 rdma 读取操作数。 单位：计数	支持 EFA 的所有实例类型
rdma_read_resp_bytes	为响应 rdma 读取操作而发送的字节数。 单位：字节	支持 EFA 的所有实例类型
rdma_write_recv_bytes	通过 rdma 写入操作收到的字节数。 单位：字节	支持 EFA 的所有实例类型

指标	说明	支持的实例类型
retrans_bytes	重新传输的 EFA SRD 字节数。 单位：计数	支持 EFA 的 Nitro v4 及更高版本的实例类型
retrans_pkts	重新传输的 EFA SRD 数据包数。 单位：字节	支持 EFA 的 Nitro v4 及更高版本的实例类型
retrans_timeout_events	EFA SRD 流量超时并导致网络路径更改的次数。 单位：计数	支持 EFA 的 Nitro v4 及更高版本的实例类型
impaired_remote_conn_events	EFA SRD 连接进入受损状态，从而导致吞吐速率限制降低的次数。 单位：计数	支持 EFA 的 Nitro v4 及更高版本的实例类型
unresponsive_remote_events	EFA SRD 远程连接无响应的次数。 单位：计数	支持 EFA 的 Nitro v4 及更高版本的实例类型

有关支持 EFA 的实例类型的更多信息，请参阅支持的实例类型。

检索实例的 EFA 驱动程序指标

您可以使用 rdma-tool 命令行工具来检索挂载到实例的所有 EFA 接口的指标，如下所示：

```
$ rdma -p statistic show
link rdmap0s31/1
    tx_bytes 0
    tx_pkts 0
    rx_bytes 0
    rx_pkts 0
    rx_drops 0
    send_bytes 0
```

```
    send_wrs 0
    recv_bytes 0
    recv_wrs 0
    rdma_read_wrs 0
    rdma_read_bytes 0
    rdma_read_wr_err 0
    rdma_read_resp_bytes 0
    rdma_write_wrs 0
    rdma_write_bytes 0
    rdma_write_wr_err 0
    retrans_bytes 0
    retrans_pkts 0
    retrans_timeout_events 0
    unresponsive_remote_events 0
    impaired_remote_conn_events 0
```

您也可以使用以下命令从 sys 文件中检索挂载到实例的每个 EFA 接口的指标。

```
$ more /sys/class/infiniband/device_number/ports/port_number/hw_counters/* | cat
```

例如

```
$ more /sys/class/infiniband/rdmap0s31/ports/1/hw_counters/* | cat
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/lifespan
::::::::::::::
12
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_resp_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_read_wrs
::::::::::::::
0
```

```
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wr_err
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rdma_write_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/recv_wrs
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_drops
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/rx_pkts
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/send_wrs
::::::::::::::
0
```

```
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_bytes
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/tx_pkts
::::::::::::::
0
::::::::::::::
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/retrans_bytes
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/retrans_pkts
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/retrans_timeout_events
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/unresponsive_remote_events
::::::::::::::
0
/sys/class/infiniband/rdmap0s31/ports/1/hw_counters/impaired_remote_conn_events
::::::::::::::
0
```

Amazon VPC 流日志

您可以创建 Amazon VPC 流日志以捕获有关进出 EFA 的流量的信息。流日志数据可以发布到 Amazon CloudWatch Logs 和 Amazon S3。在创建流日志后，您可以在所选的目标中检索和查看其数据。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 流日志。

您可以使用为弹性网络接口创建流日志的相同方式为 EFA 创建流日志。有关更多信息，请参阅《Amazon VPC 用户指南》中的创建流日志。

在流日志条目中，EFA 流量由 `srcAddress` 和 `destAddress` 标识，二者都格式化为 MAC 地址，如以下示例中所示。

```
version accountId  eniId        srcAddress        destAddress       sourcePort destPort
 protocol packets bytes start      end        action log-status
2       3794735123 eni-10000001 01:23:45:67:89:ab 05:23:45:67:89:ab -          -
 -        9       5689  1521232534 1524512343 ACCEPT OK
```

Amazon CloudWatch

如果在 Amazon EKS 集群中使用 EFA，则可以使用 CloudWatch Container Insights 监控您的 EFA。Amazon CloudWatch Container Insights 支持所有 EFA 驱动指标，以下指标除外：`retrans_bytes`、`retrans_pkts`、`retrans_timeout_events`、`unresponsive_remote_event` 和 `impaired_remote_conn_events`。

有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的 Amazon EKS 和 Kubernetes Container Insights 指标。

使用校验和验证 EFA 安装程序

您可以选择使用 MD5 或 SHA256 校验和来验证 EFA tarball（`.tar.gz` 文件）。建议您执行此操作以验证软件发布者的身份，并检查该应用程序自发布以来是否已被更改或损坏。

验证 tarball

使用 md5sum 实用程序生成 MD5 校验和，或使用 sha256sum 实用程序生成 SHA256 校验和，并指定 tarball 文件名。必须从保存 tarball 文件的目录运行命令。

- MD5

```
$  md5sum tarball_filename.tar.gz
```

- SHA256

```
$  sha256sum tarball_filename.tar.gz
```

这些命令应返回以下格式的校验和值。

```
checksum_value tarball_filename.tar.gz
```

将命令返回的校验和值与下表中提供的校验和值进行比较。如果校验和匹配，则可以放心地运行安装脚本。如果校验和不匹配，请不要运行安装脚本，并联系 Amazon Web Services 支持。

例如，以下命令使用 SHA256 校验和来验证 EFA 1.9.4 tarball。

```
$  sha256sum aws-efa-installer-1.9.4.tar.gz
```

输出示例：

```
1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14 aws-efa-
installer-1.9.4.tar.gz
```

下表列出了最近几个版本的 EFA 的校验和。

版本	校验和
EFA 1.48.0	MD5：d30960f1876fb16ac8238193a26c6b29 SHA256：f3a8aadc2b94288694a9b9375bcad84a82f0f843e0eefafda7b158b2200549cc
EFA 1.47.0	MD5：c81d4caf24dabc04a6e4818590620f5f SHA256：2df4201e046833c7dc8160907bee7f52b76ff80ed147376a2d0ed8a0dd66b2db
EFA 1.46.0	MD5：a88bbd9b71624d7ca401b54bc2fc0c19 SHA256：8302bd7849afb95c903a875d7dcb6f85b3d7629e9a8b67d020031cfc6f4d0ee1
EFA 1.45.1	MD5：91c3c87e16bbcaca1513252c38b771bb SHA256：9aeb20c645135b6039cc08986d8f14e63280f7839e882a74df5e83627ffeaa17
EFA 1.45.0	MD5：800aeddfa9d9c5f139a7b8f7c4fec627 SHA256：25ba26a0877fe3317390dc126aad2f23e27fc461cf0b940004f032cb342fa539
EFA 1.44.0	MD5：d024f6bebe080db42745103b84ca7c43

版本	校验和
	SHA256：f129a5b44a49d593d2 47e55a59eb9bcb57121566e1c2e 42b832a4e794fa83d8a
EFA 1.43.3	MD5：4dbc6eeecc516760253c10cbedb 6319d SHA256：6c470ebce254c71653 47b5048895ac2996c8856727164 2297f4c597738300652
EFA 1.43.2	MD5：7287b25a07c9747c0d4001e8fc5 f59b2 SHA256：de15c5bdbc83b952af bde876110830c604ad0796680e5 157c05f7c1979a41069
EFA 1.43.1	MD5：7cfafc8debaea51dd4966fa0b2b ba673 SHA256：54211eda0c193138ee8ed09b5fb 41c41fc76fe0a77935fa4ec8d989466342740
EFA 1.43.0	MD5：f2b3dd7dc8670b541f7c23fd58e 5e503 SHA256：786df3458c499237be 33bb8e50ffd4da7c18c20e25438 0ffc80fb90833d8cc73
EFA 1.42.0	MD5：94b2b1db09da1dde08ec049db1f 24370 SHA256：4114fe612905ee0508 3ae5cb391a00a012510f3abfecc 642d86c9a5ae4be9008

版本	校验和
EFA 1.41.0	MD5：086181c3ee3f8da512fc6e1c795e8936 SHA256：3506354cdfbe31ff552fe75f5d0d9bb7efd29cf79bd99457347d29c751c38f9f
EFA 1.40.0	MD5：f3ec6f73fbeaccba082327507581157c SHA256：30491b0fe7c3470d4439594538855c981b05fa69862d74f8c05eb9b97912368a
EFA 1.39.0	MD5：c223d5954a85a7fbcd248c942b866e43 SHA256：2cbc028c03064633bb990782b47c36156637769e2f48704417a9c700a7a32101
EFA 1.38.1	MD5：f112569e828ab65187777f794bab542c SHA256：83923374afd388b1cfcf4b3a21a2b1ba7cf46a01a587f7b519b8386cb95e4f81
EFA 1.38.0	MD5：43a2a446b33a2506f40853d55059f1ea SHA256：4f436954f35ad53754b4d005fd8d0be63de3b4184de41a695b504bdce0fecb22

版本	校验和
EFA 1.37.0	MD5：6328070192bae920eca45797ad4c1db1 SHA256：2584fc3c8bb99f29b3285e275747ff09d67c18e162c2a652e36c976b72154bfb
EFA 1.36.0	MD5：1bec83180fbffb23452ab6469ca21dfa SHA256：de183f333cfb58aeb7908a67bf9106985ba3ccb7f8638b851d2a0d8dbfacaec4
EFA 1.35.0	MD5：252f03c978dca5f8e8d9f34e488b256e SHA256：432b6ad4368ba0cd8b902729d14a908a97be7a3dcc5239422ea994a47f35a5e1
EFA 1.34.0	MD5：5cd4b28d27a31677c16139b54c9acb45 SHA256：bd68839e741b0afd3ec2e37d50603803cfa7a279c120f0a736cc57c2ff2d7fdc
EFA 1.33.0	MD5：e2f61fccbcaa11e2ccfddd3660522276 SHA256：0372877b87c6a7337bb7791d255e1053b907d030489fb2c3732ba70069185fce

版本	校验和
EFA 1.32.0	MD5：db8d65cc028d8d08b5a9f2d88881c1b1 SHA256：5f7233760be57f6fee6de8c09acbfbf59238de848e06048dc54d156ef578fc66
EFA 1.31.0	MD5：856352f12bef2ccbadcd75e35aa52aaf SHA256：943325bd37902a4300ac9e5715163537d56ecb4e7b87b37827c3e547aa1897bf
EFA 1.30.0	MD5：31f48e1a47fe93ede8ebd273fb747358 SHA256：876ab9403e07a0c3c91a1a34685a52eced890ae052df94857f6081c5f6c78a0a
EFA 1.29.1	MD5：e1872ca815d752c1d7c2b5c175e52a16 SHA256：178b263b8c25845b63dc93b25bcdff5870df5204ec509af26f43e8d283488744
EFA 1.29.0	MD5：39d06a002154d94cd982ed348133f385 SHA256：836655f87015547e733e7d9f7c760e4e24697f8bbc261bb5f3560abd4206bc36

版本	校验和
EFA 1.28.0	MD5：9dc13b7446665822605e66febe074035 SHA256：2e625d2d6d3e073b5178e8e861891273d896b66d03cb1a32244fd56789f1c435
EFA 1.27.0	MD5：98bfb515ea3e8d93f554020f3837fa15 SHA256：1d49a97b0bf8d964d91652a79ac851f2550e33a5bf9d0cf86ec9357ff6579aa3
EFA 1.26.1	MD5：884e74671fdef4725501f7cd2d451d0c SHA256：c616994c924f54ebfabfab32b7fe8ac56947fae00a0ff453d975e298d174fc96
EFA 1.26.0	MD5：f8839f12ff2e3b9ba09ae8a82b30e663 SHA256：bc1abc1f76e97d204d3755d2a9ca307fc423e51c63141f798c2f15be3715aa11
EFA 1.25.1	MD5：6d876b894547847a45bb8854d4431f18 SHA256：d2abc553d22b89a4ce92882052c1fa6de450d3a801fe005da718b7d4b9602b06

版本	校验和
EFA 1.25.0	MD5：1993836ca749596051da04694ea0d00c SHA256：98b7b26ce031a2d6a93de2297cc71b03af647194866369ca53b60d82d45ad342
EFA 1.24.1	MD5：211b249f39d53086f3cb0c07665f4e6f SHA256：120cfeec233af0955623ac7133b674143329f9561a9a8193e473060f596aec62
EFA 1.24.0	MD5：7afe0187951e2dd2c9cc4b572e62f924 SHA256：878623f819a0d9099d76ecd41cf4f569d4c3aac0c9bb7ba9536347c50b6bf88e
EFA 1.23.1	MD5：22491e114b6ee7160a8290145dca0c28 SHA256：5ca848d8e0ff4d1571cd443c36f8d27c8cdf2a0c97e9068ebf000c303fc40797
EFA 1.23.0	MD5：38a6d7c1861f5038dba4e441ca7683ca SHA256：555d497a60f22e3857fdeb3dfc53aa86d05926023c68c916d15d2dc3df6525bd

版本	校验和
EFA 1.22.1	MD5：600c0ad7cdbc06e8e846cb763f92901b SHA256：f90f3d5f59c031b9a964466b5401e86fd0429272408f6c207c3f9048254e9665
EFA 1.22.0	MD5：8f100c93dc8ab519c2aeb5dab89e98f8 SHA256：f329e7d54a86a03ea51da6ea9a5b68fb354fbae4a57a02f9592e21fce431dc3a
EFA 1.21.0	MD5：959ccc3a4347461909ec02ed3ba7c372 SHA256：c64e6ca34ccfc3ebe8e82d08899ae8442b3ef552541cf5429c43d11a04333050
EFA 1.20.0	MD5：7ebfbb8e85f1b94709df4ab3db47913b SHA256：aeefd2681ffd5c4c631d1502867db5b831621d6eb85b61fe3ec80df983d1dcf0
EFA 1.19.0	MD5：2fd45324953347ec5518da7e3fefa0ec SHA256：99b77821b9e72c8dea015cc92c96193e8db307deee05b91a58094cc331f16709

版本	校验和
EFA 1.18.0	MD5：fc2571a72f5d3c7b7b576ce2de38d91e SHA256：acb18a0808aedb9a5e485f1469225b9ac97f21db9af78e4cd6939700debe1cb6
EFA 1.17.3	MD5：0517df4a190356ab559235147174cafd SHA256：5130998b0d2883bbae189b21ab215ecbc1b01ae0231659a9b4a17b0a33ebc6ca
EFA 1.17.2	MD5：a329dedab53c4832df218a24449f4c9a SHA256：bca1fdde8b32b00346e175e597ffab32a09a08ee9ab136875fb38283cc4cd099
EFA 1.17.1	MD5：733ae2cfc9d14b52017eaf0a2ab6b0ff SHA256：f29322640a88ae9279805993cb836276ea240623820848463ca686c8ce02136f
EFA 1.17.0	MD5：d430fc841563c11c3805c5f82a4746b1 SHA256：75ab0cee4fb6bd38889dce313183f5d3a83bd233e0a6ef6205d8352821ea901d

版本	校验和
EFA 1.16.0	MD5：399548d3b0d2e812d74dd67937b696b4 SHA256：cecec36495a1bc6fdc82f97761a541e4fb6c9a3cbf3cfcb145acf25ea5dbd45b
EFA 1.15.2	MD5：955fea580d5170b05823d51acde7ca21 SHA256：84df4fbc1b3741b6c073176287789a601a589313accc8e6653434e8d4c20bd49
EFA 1.15.1	MD5：c4610267039f72bbe4e35d7bf53519bc SHA256：be871781a1b9a15fca342a9d169219260069942a8bda7a8ad06d4baeb5e2efd7
EFA 1.15.0	MD5：9861694e1cc00d884fadac07d22898be SHA256：b329862dd5729d2d098d0507fb486bf859d7c70ce18b61c302982234a3a5c88f
EFA 1.14.1	MD5：50ba56397d359e57872fde1f74d4168a SHA256：c7b1b48e86fe4b3eaa4299d3600930919c4fe6d88cc6e2c7e4a408a3f16452c7

版本	校验和
EFA 1.14.0	MD5：40805e7fd842c36ececb9fd7f921b1ae SHA256：662d62c12de85116df33780d40e0533ef7dad92709f4f613907475a7a1b60a97
EFA 1.13.0	MD5：c91d16556f4fd53becadbb345828221e SHA256：ad6705eb23a3fce44af3afc0f7643091595653a723ad0374084f4f2b715192e1
EFA 1.12.3	MD5：818aee81f097918cfaebd724eddea678 SHA256：2c225321824788b8ca3fbc118207b944cdb096b847e1e0d1d853ef2f0d727172
EFA 1.12.2	MD5：956bb1fc5ae0d6f0f87d2e481d49fccf SHA256：083a868a2c212a5a4fcf3e4d732b685ce39cceb3ca7e5d50d0b74e7788d06259
EFA 1.12.1	MD5：f5bfe52779df435188b0a2874d0633ea SHA256：5665795c2b4f09d5f3f767506d4d4c429695b36d4a17e5758b27f033aee58900

版本	校验和
EFA 1.12.0	MD5：d6c6b49fafb39b770297e1cc44fe68a6 SHA256：28256c57e9ecc0b0778b41c1f777a9982b4e8eae782343dfe1246079933dca59
EFA 1.11.2	MD5：2376cf18d1353a4551e35c33d269c404 SHA256：a25786f98a3628f7f54f7f74ee2b39bc6734ea9374720507d37d3e8bf8ee1371
EFA 1.11.1	MD5：026b0d9a0a48780cc7406bd51997b1c0 SHA256：6cb04baf5ffc58ddf319e956b5461289199c8dd805fe216f8f9ab8d102f6d02a
EFA 1.11.0	MD5：7d9058e010ad65bf2e14259214a36949 SHA256：7891f6d45ae33e822189511c4ea1d14c9d54d000f6696f97be54e915ce2c9dfa
EFA 1.10.1	MD5：78521d3d668be22976f46c6fecc7b730 SHA256：61564582de7320b21de319f532c3a677d26cc46785378eb3b95c636506b9bcb4

版本	校验和
EFA 1.10.0	MD5：46f73f5a7afe41b4bb918c81888fefa9 SHA256：136612f96f2a085a7d98296da0afb6fa807b38142e2fc0c548fa986c41186282
EFA 1.9.5	MD5：95edb8a209c18ba8d250409846eb6ef4 SHA256：a4343308d7ea4dc943ccc21bcebed913e8868e59bfb2ac93599c61a7c87d7d25
EFA 1.9.4	MD5：f26dd5c350422c1a985e35947fa5aa28 SHA256：1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14
EFA 1.9.3	MD5：95755765a097802d3e6d5018d1a5d3d6 SHA256：46ce732d6f3fcc9edf6a6e9f9df0ad136054328e24675567f7029edab90c68f1
EFA 1.8.4	MD5：85d594c41e831afc6c9305263140457e SHA256：0d974655a09b213d7859e658965e56dc4f23a0eee2dc44bb41b6d039cc5bab45

Elastic Fabric Adapter 发布说明

下表介绍了 Elastic Fabric Adapter 软件的版本历史记录和更改日志。

版本	更改	发行日期
1.48.0	• 升级至 libfabric 2.4.0amzn3.0 • 优化 RDM 操作中的 shm 地址检索 • 修复 DC 数据包完成、CQ 耗尽、AH 单元测试和 MR 日志记录中的 use-after-free 错误 • 修复 efa_rdm_ep_get_peer 和 QP 锁定中的争用条件 • 修复因未初始化描述符数组导致 FI_EFA_ENABLE_SHM_TRANSFER=0 时出现段错误的问题 • 修复 fi_info、err_buf、EFA 设备销毁和单元测试中的内存泄漏问题 • 在 CQ 轮询中为 base_ep 添加空值检查，以处理已销毁的 QP • 在 efa_qp_create 中设置 qp->base_ep，并在 efa_mr_reg_ibv_mr 返回 NULL 时返回错误 • 将 QP 表从域移至设备，并增加设备级锁定 • 为直接数据路径请求 ID 添加 QP 生成，并在 CQE 上验证 QP 编号 • 停止为直接数据路径设置 qp->ibv_qp_ex->wr_id • 收到接收完成后，从 longcts_send 列表中正确移除 txe • 重构并清理 efa_cq_handle_error	2026 年 4 月 14 日

版本	更改	发行日期
	• 将 MR 注册计数器转换为原子操作 • 迁移 shm MR 注册代码，并引入内部 mr regv 函数 • 在 ASAN 下默认禁用 MR 缓存 • 仅在 p2p 不可用但请求了 FI_HMEM 时禁用 zcpy_rx • 非调试版本中不跟踪 tx pkt 池 • 改进 wait_send 过程和模拟函数 • 调整 efa-direct CQ err_data 和 efa_show_help 的日志级别 • 定义 sanitizer 宏并在 ASAN 下禁用 memhooks • 添加针对 ep 启用错误路径和错误完成处理的单元测试 • 升级至 libnccl-ofi 1.19.0 • 修复了处理由 ROCm/HSA 创建的 dma-buf 句柄时出现的错误，并添加了在 ROCm 平台上使用 HSA 运行时检测 dma-buf 支持的功能 • 修复了潜在的内存泄漏问题 • 修复了 Docker 容器中为 GB200 生成 NCCL 拓扑的问题，其中 NUMA 节点与 Package 节点断开连接会导致拓扑生成错误 • 减少了初始化期间 NIC 0 上的 QP 使用率 • 改进了针对 P6-B200 和 P6-B300 实例类型的调谐器算法选择 • 支持在 NGC 容器上安装 libnccl-ofi • 添加对单个 RPM/DEB 软件包签名验证的支持	

版本	更改	发行日期
	• 添加对 RHEL 10 的支持 • 在 Debian 12 和 RHEL 10 上添加 OFI NCCL 插件	

版本	更改	发行日期

1.47.0	• 升级至 libfabric 2.4.0amzn1.0
	  • 重构对等管理：将端点→对等 hashmap 移至 AV 级别
	  • 添加数据包条目生成计数器以防止 ABA 问题
	  • 使用等待对象实现阻止 CQ 读取支持 (fi_cq_sread)
	  • 修复 eager/mulreq/longread RTM 数据包处理中的双重释放问题
	  • 基于设备 RDMA 功能修复 MR 注册 IBV 访问标志
	  • 禁用 FI_OPT_EFA_SENDRECV_IN_ORDER_ALIGNED_128_BYTES（返回 -FI_EOPNOTSUPP）
	  • 优化数据包条目以适应两行 x86 缓存（128 字节）
	  • 发生 RTM/RTA 错误时始终成功执行 efa_rdm_srx_start 和滑动 recv 窗口
	  • 在端点清理期间释放之前，先取消链接 RX 数据包条目
	• 升级至 libnccl-ofi 1.18.0
	  • P6-B300 支持：为 P6-B300 添加了自定义调谐器决策
	  • 通过减少较小消息大小的通道计数，提高了 P6-B200 上 PAT 的性能
	  • 更改了协议默认值，Trn1 默认使用 RDMA 协议，g7e.8x large 默认使用 SENDRECV 协议。
	  • 动态平台选择：添加了根据 Amazon 网卡的存在情况在运行时启用 Amazon 优化的功能。这允许单个插件二进制文件
		2026 年 1 月 29 日

版本	更改	发行日期
	用于 Amazon 和非 Amazon 平台。仍然可以在编译时禁用 Amazon 优化。 • 修复了 RDMA 协议中对非 FI_MR_VIRT_ADDR 提供程序的支持 • 改进了向 NCCL 报告 NIC PCIe 链路速度和宽度 • 重新设计了线程模型，以支持多线程应用程序，而无需为每个线程使用单独的 Libfabric 域。 • 通过按正确顺序清理资源，修复了对 FI_MR_ENDPOINT 提供程序的支持（仅支持 SENDRECV 协议） • 升级至 rdma-core 61.0 • 如果提供了线程域，则使用单线程 CQ • 验证 CQE 进程上的 QP 编号 • 升级至 efa driver 3.0.0 • 改进管理员错误处理 • 完成轮询后检查 QP 编号是否正确 • 移除最佳页面大小查找的无条件逆向移植 • 打印错误指针的 errno 字符串 • 简化 CQ 创建流中的代码 • 使用 umem 创建流逆向移植 CQ 中的上游更改 • 调整新 DKMS 版本的 DKMS 配置 • 升级至 Open MPI 5.0.9amzn1 • 错误修复：基于 MPI 线程支持设置域线程级别	

版本	更改	发行日期
	• 错误修复：请求 FI_COMPLETION 标志，以确保为所有数据传输事件生成完成 • 在 BTL 和 MTL 之间共享域，以减少为修复核心计数较高的系统上的资源耗尽而创建的域总数	

1.46.0	• 增加对 Debian 13 的支持
	• 升级至 efa-nv-peermem 1.2.3
	  • 使用新内核和 DKMS 修复编译错误和警告
	• 升级至 libfabric 2.3.1amzn4.0
	  • 为 efa-rdm 协议路径启用直接数据路径
	  • 启用 ROCr HMEM 支持
	  • 新的 FI_OPT_EFA_USE_UNSOLICITED_WRITE_RECV 选项用于禁用未经请求的写入接收
	  • 优化连接建立时间
	• 在 RHEL 9、Rocky Linux 9 和 Debian 13 上添加 libnccl-ofi
		2025 年 12 月 12 日

1.45.1	• 升级至 libfabric 2.3.1amzn3.0 • 修复了以下错误：cq_data 截断为 2 字节，而提供商声明支持 4 字节	2025 年 11 月 26 日

版本	更改	发行日期

1.45.0	• 升级至 rdma-core 60.amzn0
	  • 扩展 DV 查询 CQ 以返回门铃
	• 升级至 libfabric 2.3.1amzn2.0
	  • 对 efa-direct 支持 FI_RX_CQ_DATA 模式
	  • 优化数据路径直接路径中的 WQE 发布
	  • 修复 cntr 进度中的争用情况
	  • 改进警告消息的日志记录
	  • 正确处理已移除对等方的 RMA 完成
	  • 当达到 AH 限制时，从隐式 AV 中移除 AH 条目
	• 升级至 libnccl-ofi 1.17.2
	  • 修复了 Libfabric 初始化失败时 NCCL v2.28.x 的崩溃问题
	  • 增加了对 g7e 实例系列的支持
	  • 修复了以下问题：NCCL 可能会在支持 DMA-BUF 的平台上错误地尝试使用 GPUDirect RDMA 路径
	  • 修复了需要按端点内存注册的 NIC 上的关闭顺序问题（Cray Slingshot）
		2025 年 11 月 17 日

版本	更改	发行日期

1.44.0	• 升级至 rdma-core 59.amzn0
	  • 增加了对将 QP 和 CQ 绑定到线程域的支持
	  • 增加了对分配父域的支持
	  • 增加了对分配线程域的支持
	  • 增加了直接动词查询 QP 和 CQ
	  • 增加了使用外部内存`创建 CQ 的选项
	  • 修复了 QP 销毁后的 CQ 轮询
	• 升级至 libfabric 2.3.1amzn1.0
	• 升级至 Open MPI 5.0.8amzn1
	  • 功能增强
	  • 错误修复
	• 升级至 libnccl-ofi 1.17.1
	  • 支持 NCCL v2.28.3-1，同时保持与 NCCL v2.17.1 及更高版本的向后兼容性。
	  • 使用 platform-aws 进行构建需要 Libfabric v1.22.0amzn4.0 或更高版本。已使用最高至 Libfabric v2.3.1amzn1.0 的版本进行测试。
	  • 进行了修复，可支持跨 CUDA12 和 CUDA13 的兼容性。
		2025 年 10 月 29 日

版本	更改	发行日期

1.43.3	• 升级至 efa driver 2.17.3
	  • 使用 NVIDIA 580 驱动程序支持 P2P
	  • 调整 6.17 和内核上的 MR 注册
		2025 年 10 月 1 日

1.43.2	• 升级至 libnccl-ofi 1.16.3
	  • 支持 NCCL v2.27.7-1，同时保持与 NCCL v2.17.1 及更高版本的向后兼容性。
	  • 使用 platform-aws 进行构建需要 Libfabric v1.22.0amzn4.0 或更高版本。已使用最高至 Libfabric 2.1.0amzn5.0 的版本进行测试。
	  • 对所有 Amazon EC2 实例类型默认启用每线程域以提高 NCCL 创建多个代理线程的某些应用程序的性能
		2025 年 8 月 15 日

1.43.1	• 升级至 libnccl-ofi 1.16.2
	  • 支持 NCCL v2.27.6-1，同时保持与 NCCL v2.17.1 及更高版本的向后兼容性。
	  • 使用 platform-aws 进行构建需要 Libfabric v1.22.0amzn4.0 或更高版本。已使用最高至 Libfabric 2.1.0amzn4.0 的版本进行测试。
	  • 添加了新的平台配置，以支持在 Amazon EC2 p5.4xlarge 实例类型上使用 OFI NCCL 插件
	• 升级至 libfabric 2.1.0amzn5.0
		2025 年 7 月 31 日

版本	更改	发行日期

1.43.0	• 升级至 libnccl-ofi 1.16.1
	  • 支持 NCCL v2.27.6-1，同时保持与 NCCL v2.17.1 及更高版本的向后兼容性。
	  • 使用 platform-aws 进行构建需要 Libfabric v1.22.0amzn4.0 或更高版本。已使用最高至 Libfabric 2.1.0amzn3 的版本进行测试。
	  • 更新拓扑文件中报告的 PCI 链路速度格式以匹配内核 5.7+
	  • 添加了 SKIP_NICS_WITHOUT_ACCEL_AT_SAME_PCI_LEVEL 以跳过没有相同 pci 级别加速器的 libfabric nic
	• 升级至 efa driver 2.17.2
	• 升级至 efa-nv-peermem 1.2.2
	• 升级至 libfabric 2.1.0amzn4.0
	• 升级至 rdma-core 58.amzn0
	  • 修复工作请求索引重复使用问题
	  • 将 WQE 长度添加到 post_send 跟踪点
	• 启用 Graviton 平台优化
	• 弃用对 Ubuntu 20.04 的支持
		2025 年 7 月 25 日

版本	更改	发行日期

1.42.0	• 升级至 efa driver 2.15.3
	• 升级至 efa-nv-peermem 1.2.1
	• 升级至 rdma-core 57.amzn1
	  • 修复工作请求索引重复使用问题
	• 升级至 libfabric 2.1.0amzn3.0
	• 升级至 libnccl-ofi 1.15.0
	  • 支持 NCCL 2.26.6-1，同时保持与 NCCL v2.17.1 及更高版本的向后兼容性。
	  • 使用 platform-aws 进行构建需要 Libfabric v1.22.0amzn4.0 或更高版本。已使用最高至 Libfabric 2.1.0amzn3 的版本进行测试。
	  • 构建系统和平台支持
	  • 添加了 Amazon EC2 P6-B200 平台支持
	  • 将默认插件库名称更改为 libnccl-net-ofi.so，并默认创建从 libnccl-net-ofi.so 到 libnccl-net.so 的符号链接以保持向后兼容性。这允许用户将 NCCL_NET_PLUGIN=ofi 设置为强制 NCCL 以使用 OFI 插件进行通信。在配置时指定 --disable-nccl-net-symlink 将跳过符号链接，从而允许在同一个容器中安装多个插件。
	  • 微调和性能改进
	  • 在 P6-B200 上为 0x0 和 0x7 位掩码的 AllReduce、AllGather 和 ReduceScatter 区域添加了调优器支持
	  • 根据实证结果和分析，更新了 P5eN 和 P6-B200 平台的默认延迟
	  • 更新以使用支持 trafficClass 参数的 NCCL v10 API 确定未来流量的优先级
	  • 将插件代码库从 C 迁移到 C++
		2025 年 6 月 6 日

版本	更改	发行日期
	• 添加了对每个 GPU 的 NIC 数量因系统而异的作业的支持。有关更多信息，请参阅 OFI_NCCL_FORCE_NUM_RAILS 运行时环境变量文档。	
1.41.0	• 升级至 rdma-core 57.amzn0	2025 年 5 月 16 日
1.40.0	• libfabric 升级至 2.1.0amzn2.0 • 升级至 rdma-core 56.0 • 升级至 efa-config 1.18。 • 停止支持 Debian 10 • 升级至 libnccl-ofi 1.14.2 • 在 ARM 上的 Libfabric 中实现了适用于 Ubuntu 和 Amazon Linux 2023 的 CUDA 支持 • 在 ARM 上增加了适用于 Ubuntu 和 Amazon Linux 2023 的 libnccl-ofi	2025 年 5 月 2 日
1.39.0	• libfabric 升级至 2.1.0 • 升级至 Open MPI 5.0.6 • 升级至 libnccl-ofi 1.14.1 • 升级至 efa 驱动程序 2.15.0	2025 年 4 月 16 日
1.38.1	• 更新至 `Libfabric 1.22.0amzn5.0`	2025 年 3 月 3 日
1.38.0	• 添加 `libnccl-ofi 1.13.2-1`	2025 年 1 月 8 日
1.37.0	• 升级到 `libfabric 1.22.0amzn4.0` • 升级到 `rdma-core 54.amzn0`	2024 年 11 月 18 日

版本	更改	发行日期
1.36.0	• 增加支持 Debian 12 • 将 `efa_test.sh` 作为实用程序脚本分发到 `/opt/amazon/efa/bin` • 升级至 efa 驱动程序 2.13.0 • 升级到 `libfabric 1.22.0amzn3.0` • 升级到 `rdma-core 54.0` • 升级至 Open MPI 4.1.7	2024 年 11 月 7 日
1.35.0	• 升级至 Open MPI 5.0.5 • 升级至 PRRTE 3.0.6 并将 RPM 程序包重命名为 prrte-aws • 将 OpenPMIx RPM 程序包重命名为 pmix-aws • 配置了 dpkg 和 RPM 构建版本的构建选项 • 升级到 `Libfabric 1.22.0amzn2.0` • 升级至 efa 驱动程序 2.12.1	2024 年 10 月 14 日
1.34.0	• 停止支持 CentOS 7 和 RHEL 7 • 引入 `Libfabric 1.22.0amzn1.0` • efa-config 程序包更新至 1.17	2024 年 8 月 6 日
1.33.0	• 升级至 efa 驱动程序 2.10.0 • 升级到 `rdma-core 52.0` • 升级至 PRRTE 3.0.5 • 升级至 Open MPI 5.0.3 • 增加支持 Ubuntu 24.04 LTS • 移除了 OpenPMIx、PRRTE 和 Open MPI HTML 文档。用户应查阅官方网站。 • 取消支持 openSUSE 15.4 或更低版本	2024 年 6 月 20 日

版本	更改	发行日期
1.32.0	• 升级至 efa 驱动程序 2.8.0 • 升级至 efa-nv-peermem 驱动程序 1.1.1 • efa-config 程序包升级至 1.16 • efa-config 程序包升级至 1.7 • 升级到 `rdma-core 50.0` • 升级到 `libfabric 1.21.0amzn1.0` • 增强了 `efa_test.sh` 的功能，提供 `fi_pingpong` 端口选择功能 • 可需要时从系统安装较新版本的 rdma-core	2024 年 4 月 18 日
1.31.0	• 升级至 OpenPMIx 4.2.8 • 升级至 PRRTE 3.0.3 • 升级至 Open MPI 5.0.2，并将 MCA 组件转换为 DSO。 • 升级到 `Libfabric 1.20.1amzn1.0`	2024 年 3 月 7 日
1.30.0	• 包含 OpenPMIx 4.2.7，默认安装在 `/opt/amazon/pmix` • 包含 PRRTE 3.0.2，默认安装在 `/opt/amazon/prrte` • 包含 Open MPI 5.0.0，默认安装在 `/opt/amazon/openmpi5` • efa-profile 更新至 1.6	2023 年 12 月
1.29.1	• 引入 `libfabric 1.19.0amzn4.0`	2023 年 12 月
1.29.0	• 引入 efa 内核驱动程序 2.6.0 • 引入 `libfabric 1.19.0amzn3.0`	2023 年 11 月
1.28.0	• Open MPI 升级至 4.1.6 • 引入 `libfabric 1.19.0amzn2.0`	2023 年 10 月
1.27.0	• 安装程序增加了 `Libfabric 1.19.0amzn1.0` • Open MPI 4 实现了 GCC 内置原子	2023 年 9 月
1.26.1	• `efa_test.sh` ：修复了无限重试错误	2023 年 9 月

版本	更改	发行日期
1.26.0	• 增加支持 RHEL 9 • 修复了多个发行版上的调试生成 • 引入 `libfabric 1.18.2amzn1.0` • `efa_test.sh` ：增加了 `fi_pingpong` 重试功能	2023 年 9 月
1.25.1	• 引入 `libfabric 1.18.1amzn1.0` • 更新了 EFA 安装程序脚本中的超时	2023 年 9 月
1.25.0	• 增加了对 Amazon Linux 2023 的支持 • 修复了安装后测试中的错误	2023 年 7 月
1.24.1	• libfabric 升级至 1.18.1 • efa 驱动程序升级至 2.5.0	2023 年 7 月
1.24.0	• 引入 `rdma-core 46.0` • 引入 efa 内核驱动程序 2.4.1 • 支持 Debian 11	2023 年 6 月
1.23.1	• 引入 `libfabric 1.18.0amzn2.0`	2023 年 6 月
1.23.0	• 增加支持 Debian 10 • 停止支持 Ubuntu 18.04 LTS • efa-config 程序包升级至 1.14 • 引入 `libfabric 1.18.0amzn1.0`	2023 年 5 月
1.22.1	• libfabric 升级至 1.17.1	2023 年 3 月
1.22.0	• Open MPI 升级至 4.1.5 • libfabric 升级至 1.17.0 • efa-config 程序包升级至 1.13	2023 年 2 月

版本	更改	发行日期
1.21.0	• 增加支持 Rocky Linux 9 操作系统 • 引入 efa 内核驱动程序 2.1.1 • 引入 `libfabric 1.16.1amzn3.0` • efa-config 程序包升级至 1.12	2022 年 12 月
1.20.0	• 增加支持 Rocky Linux 8 操作系统。 • 引入 efa 内核驱动程序 2.1.0。 • 引入 `rdma-core 43.0`。 • 引入 `libfabric 1.16.1amzn1.0`	2022 年 11 月
1.19.0	• 引入 `libfabric 1.16.0` • 使用 `--enable-orterun-prefix-by-default` 设置构建 Open MPI	2022 年 10 月
1.18.0	• 增加支持 Ubuntu22.04	2022 年 8 月
1.17.3	• libfabric 更新至 `1.16.0~amzn4.0` 。~ 表示此为 libfabric 1.16.0 的预发行版。 • 将安装后的 pingpong 测试超时时间延长至 20 秒	2022 年 8 月
1.17.2	• libfabric 更新至 `1.16.0~amzn3.0` 。~ 表示此为 libfabric 1.16.0 的预发行版。	2022 年 7 月
1.17.1	• libfabric 更新至 `1.16.0~amzn2.0` 。~ 表示此为 libfabric 1.16.0 的预发行版。 • 在构建 libfabric 时禁用实验性网络提供者	2022 年 7 月
1.17.0	• `rdma-core` 更新至 v41.0 • Open MPI 更新至 4.1.4 • libfabric 更新至 `1.16.0~amzn1.0` 。~ 表示此为 libfabric 1.16.0 的预发行版。	2022 年 7 月

版本	更改	发行日期
1.16.0	• libfabric 更新至 `1.15.1amzn1.0` ，包含神经元库名称更改 • efa-config 升级至 1.10 • 在 libfabric 生成中排除 opx 和 rxd 提供者	2022 年 6 月
1.15.2	• libfabric 更新至 1.14.1	2022 年 5 月
1.15.1	• libfabric 更新至 `1.14.0amzn1.0`	2022 年 3 月
1.15.0	• 修复了一个会导致在 Open SuSE 15.3 上安装失败的错误 • 停止支持 Open suSE 15.2（因为 Open suSE 15.2 已接近生命周期终止） • 停止支持 CentOS 8（因为 CentOS 8 已达生命周期终止） • libfabric 更新至 1.14.0 • efa 内核驱动程序更新至 1.16.0 • `rdma-core` 更新至 v39.0 • Open MPI 更新至版本 4.1.2。	2022 年 2 月
1.14.1	• libfabric 更新至 `1.13.2amzn1.0` 。	2021 年 10 月
1.14.0	• 引入 efa 内核驱动程序 1.14.2。 • 使 `efa_installer.sh` 中的 `-g, --enable-gdr` 成为无操作选项，因为最新版本的 efa 内核驱动程序已默认启用 GDR 支持。 • 引入 `rdma-core v37.0`。 • 引入 `libfabric 1.13.2`。 • 增加了程序包列表并在安装过程中将 RPM/DEB 与列表进行比较，以防止安装未知程序包。 • 安装程序脚本中增加了 sleep，以等待 udev 规则在 EFA 驱动程序重新加载后生效。	2021 年 10 月

版本	更改	发行日期
1.13.0	• `rdma-core` 更新至 v35.0。 • `libfabric` 更新至 v1.13.0amzn1.0。 • Gravition2 平台增加了对 Centos/RHEL 8 的 EFA 支持。 • 安装程序脚本中增加了版本比较逻辑，以便在更高版本的系统中已经了安装本地软件包时跳过此步骤。	2021 年 8 月
1.12.3	• EFA 内核模块更新至 1.13.0。 • efa-config 程序包更新至版本 1.9。改进了巨型页预留量的计算，以满足默认巨型页大小较大的需要。	2021 年 7 月
1.12.2	• EFA 内核模块更新至 1.12.3。 • 使用 `--with-libevent=external` 和 `--with-hwloc=external` 设置构建 Open MPI debian 程序包。 • 将 Open MPI rpm 生成版本 ID 升级为 2，修复 CentOS 8 上的 HWLOC 向后兼容性问题。 • 取消在 SLES15SP2 和 openSUSE 15.2 上安装 kernel-devel 和 kernel-source 程序包。	2021 年 6 月
1.12.1	• Libfabric 更新至版本 1.11.2amzon1.1。 • EFA 内核模块更新至版本 1.12.1。	2021 年 5 月

版本	更改	发行日期
1.12.0	• Open MPI 更新至版本 4.1.1。 • Libfabric 更新至版本 1.11.2amzn1.0。 • 使用与 AL2 rdma-core 相同的打包配置构建适用于 Amazon Linux 2 的 rdma-core。 • 在构建 Open MPI RPM 时不强制实施 `-Wl,--enable-new-dtags` 。 • 使用适用于 hwloc 和 libevent 的系统库构建 Open MPI。 • EFA 内核模块更新至版本 1.12.0 • efa-config 程序包更新至版本 1.8。改进了巨型页预留量的计算，以满足长时间运行实例的需要。 • efa-profile 程序包更新至版本 1.5。移除用于临时解决 P4d 上的 Open MPI 4.1.0 挂起问题的 Open MPI 集体调整文件。 • `rdma-core` 更新至 v32.1。 • 停止支持 Amazon Linux 1 和 Ubuntu 16.04。	2021 年 5 月
1.11.2	• 通过配置文件更改了用于实现 MPI_BARRIER 的默认算法，从而解决了在 P4d 实例上使用 Open MPI 时出现 Open MPI 挂起的问题。 • 通过 `--disable-builtin-atomics` 在 ARM 上的 Open MPI 中禁用内置原子，从而解决编译器问题。	2021 年 2 月
1.11.1	• Open MPI 更新至版本 4.1.0。 • efa-config 程序包更新至版本 1.7。改进了巨型页预留量的计算。 • efa-profile 程序包更新至版本 1.3。移除了不需要的集体决策文件，因为现在使用的是 Open MPI 4.1.0。	2020 年 12 月
1.11.0	• 增加支持 Gravition2 平台。 • `rdma-core` 更新至版本 31.2amzn。 • Libfabric 更新至版本 1.11.1amzn1.0。 • efa-config 程序包更新至版本 1.6。 • efa-profile 程序包更新至版本 1.2。	2020 年 12 月

版本	更改	发行日期
1.10.1	• 增加支持 CentOS / RHEL 8。 • 增加支持 Ubuntu 20.04。 • 增加支持 SUSE Linux Enterprise 15。	2020 年 11 月
1.10.0	• 增加 P4d 平台对 GpuDirect RDMA 的支持。使用 `--enable-gdr` 安装程序选项来安装 GDR 感知内核模块和用户空间。 • EFA 内核模块更新至版本 1.10.2。 • `rdma-core` 更新至版本 31.amzn0。 • Libfabric 更新至版本 1.11.1。 • Open MPI 更新至版本 4.0.5。 • efa-config 程序包更新至版本 1.5。 • efa-profile 程序包更新至版本 1.1。包含改进的 Open MPI 集体决策文件。	2020 年 10 月
1.9.5	• efa-config 程序包更新至版本 1.4。修复了 Open MPI 集体决策文件中的错误。	2020 年 9 月
1.9.4	• Open MPI 更新至版本 4.0.3。 • Libfabric 更新至版本 1.10.1amazon1.1。 • `rdma-core` 更新至版本 28.amzn0。	2020 年 7 月
1.9.3	• EFA 内核模块更新至版本 1.6.0。 • `rdma-core` 更新至版本 28.amzn0。 • Libfabric 更新至版本 1.10.1amzn1.1。 • efa-config 程序包更新至版本 1.3。增加了 Open MPI 的集体调整文件。 • 可在已安装 dkms 时跳过安装。 • 修复了 `--skip-kmod` 安装模式，使其能够正常工作。	2020 年 6 月
1.8.4	• 将配置文件移至 efa-config 和 efa-profile 程序包中，以便操作系统程序包管理器对其进行跟踪。 • Open MPI 更新至版本 4.0.3。	2020 年 4 月

版本	更改	发行日期
1.8.3	• EFA 内核模块更新至版本 1.5.1。 • 在某些平台上分发 DKMS，而不是依赖 EPEL 存储库，从而提高安装可靠性。 • 在 RHEL 7 上，安装基于 CentOS 7 构建的 RPM，而不是基于 Amazon Linux 2 构建的 RPM。	2020 年 2 月
1.8.2	• 由于内核模块与 `rdma-core` 之间的设备命名不一致，因此将 `rdma-core` 恢复到版本 25。	2020 年 1 月
1.8.1	• Libfabric 更新至版本 1.9.0amzn1.1。	2020 年 1 月
1.8.0	• `rdma-core` 更新至版本 27.0。 • EFA 内核模块更新至版本 1.5.0。 • Libfabric 更新至版本 1.9.0amzn1.0。 • 增加了 `--minimal` 选项，以仅安装 EFA 内核模块和 `rdma-core` 。	2019 年 12 月
1.7.1	• Libfabric 更新至版本 1.8.1amzn1.3。	2019 年 12 月
1.7.0	• 增加了 Libfabric 模块文件。 • Libfabric 更新至版本 1.8.1amzn1.1。	2019 年 11 月
1.6.2	• Open MPI 更新至版本 1.6.2。	2019 年 10 月
1.6.1	• Libfabric 更新至版本 1.8.1amzn1.0。 • Open MPI 更新至版本 4.0.1。 • `rdma-core` 更新至版本 26.0。	2019 年 10 月
1.5.4	• EFA 内核模块更新至版本 1.4.1。	2019 年 9 月
1.5.3	• EFA 内核模块更新至版本 1.3.1。 • 避免安装 `kernel-devel` 或 `linux-headers` 程序包，除非要安装内核驱动程序。	2019 年 9 月
1.5.1	• 仅在有 EFA 设备时才配置巨型页。	2019 年 8 月

版本	更改	发行日期
1.5.0	• Libfabric 更新至版本 1.8.0amzn1.1。 • `rmda-core` 更新至版本 25.0。	2019 年 8 月
1.4.1	• 在 `/etc/ld.so.conf.d/efa.conf` 中添加了 Libfabric 和 Open MPI 库路径（`/opt/amazon/efa/lib64` 和 `/opt/amazon/efa/openmpi/lib`），确保 Open MPI 和 Libfabric 共享库的位置正确。	2019 年 7 月
1.4.0	• EFA 内核模块更新至版本 1.3.0。 • Libfabric 更新至版本 1.8.0amzn1.0 • 首个支持 Intel MPI 2019 Update 4 的版本。	2019 年 7 月

Amazon EC2 拓扑

Amazon EC2 拓扑提供了计算容量的相对邻近度的分层视图。您可以使用此信息来大规模管理高性能计算（HPC）、机器学习（ML）和生成式人工智能计算基础设施。

可用的 API

Amazon EC2 提供了两个 API 来了解您的 EC2 拓扑：

- DescribeInstanceTopology
  - 显示正在运行的实例在网络层次结构中相对于彼此的位置。
  - 帮助优化在现有实例上运行工作负载的位置。
- DescribeCapacityReservationTopology
  - 显示在启动实例之前预留容量在网络层次结构中相对于彼此的位置。
  - 告知您在启动实例之前预留容量的置放情况，从而帮助进行容量规划。

主要优势

EC2 拓扑具有以下主要优势：

- 容量管理 – 优化资源利用率。
- 任务计划工作安排 – 做出关于工作负载置放的明智决策。

- 节点排名 – 了解相对邻近度，以对紧密耦合的实例进行性能优化。

注意事项

- 拓扑视图仅适用于：
  - 处于 `running` 状态的实例
  - 处于 `pending` 或 `active` 状态的容量预留
- 每个 Amazon Web Services 账户 的每个拓扑视图都是唯一的。
- Amazon Web Services 管理控制台 不支持查看拓扑。
- 虽然拓扑信息可帮助您了解实例置放，但不能用其来启动物理上靠近现有实例的新实例。要影响实例放置，您可以在集群置放群组中创建容量预留。

定价

描述您的 EC2 拓扑不会产生额外的成本。

内容

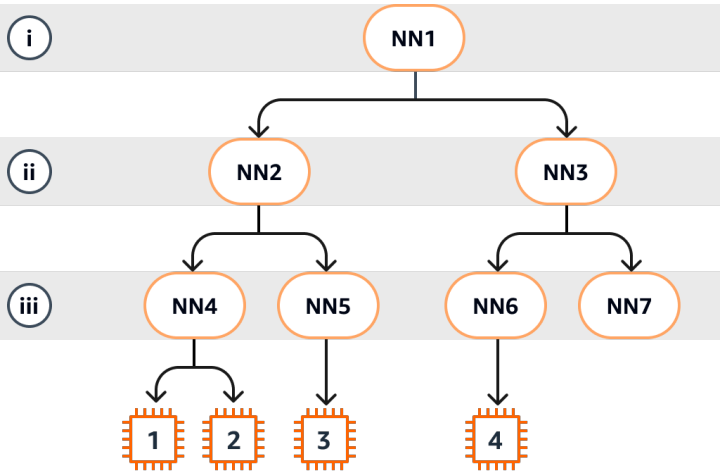
- Amazon EC2 拓扑工作原理
- Amazon EC2 拓扑的先决条件
- Amazon EC2 实例拓扑的示例

Amazon EC2 拓扑工作原理

Amazon 网络采用分层式层次结构。EC2 实例在第三层或更低层连接到网络，具体取决于实例类型。实例的拓扑结构可描述为一组节点，网络中的每层一个节点。借助 DescribeInstanceTopology 或 DescribeCapacityReservationTopology API 响应中返回的节点集，可以自上而下地总览网络层次结构，底层节点会连接到实例。

> **Note**
>
> 有些实例类型的节点组合包含 4 个网络节点，代表网络中共有 4 层；另一些实例类型的节点组合包含 3 个网络节点，代表网络中共有 3 层。有关受支持的实例类型，请参阅实例类型。您可能只会看到 1、2 或 3 个网络节点，具体取决于容量预留的类型。

下图提供了可视化表示，您可以使用其来理解 EC2 拓扑。网络节点被标识为 NN1 – NN7。数字符号 i、ii 和 iii 用于标识网络层。数字 1、2、3 和 4 用于标识 EC2 实例。实例连接到位于底层（下图中用 iii 标识）的节点。多个实例可以连接到同一个节点。



在本示例中：

- 实例 1 连接到第 iii 层中的网络节点 4（NN4）。NN4 连接到第 ii 层的网络节点 2（NN2），NN2 连接到第 i 层中的网络节点 1（NN1），第 i 层是本示例中网络层次结构的顶层。网络节点集包括 NN1、NN2 和 NN4，按从上层到底层的顺序分层表示。
- 实例 2 还连接到网络节点 4（NN4）。实例 1 和实例 2 共享相同的网络节点集：NN1、NN2 和 NN4。
- 实例 3 连接到网络节点 5（NN5）。NN5 连接到 NN2，而 NN2 连接到 NN1。实例 3 网络节点集是 NN1、NN2 和 NN5。
- 实例 4 连接到网络节点 6（NN6）。其网络节点集是 NN1、NN3 和 NN6。

考虑实例 1、实例 2 和实例 3 的邻近性时，实例 1 和实例 2 彼此靠得更近，因为其连接到同一个网络节点（NN4），而实例 3 则较远，因为其连接到不同的网络节点（NN5）。

在考虑此图中所有实例的邻近性时，实例 1、实例 2 和实例 3 彼此之间比其与实例 4 之间的距离更近，因为这几个实例在网络节点集中共享 NN2。

一般来说，如果连接到任意两个实例的网络节点相同，则这些实例在物理上彼此靠近，就像实例 1 和实例 2 一样。此外，网络节点之间的跳数越少，实例之间的距离就越近。例如，实例 1 和实例 3 到共同网络节点（NN2）的跳数少于其与实例 4 共同网络节点（NN1）的跳数，因此实例 1 和实例 3 彼此之间的距离比到实例 4 更近。

在此示例中，没有实例在网络节点 7（NN7）下运行，因此 API 输出将不包括 NN7。

如何解释 DescribeInstanceTopology 输出

您可以使用 DescribeInstanceTopology API 来描述实例拓扑。输出提供了实例底层网络拓扑的分层视图。

以下示例输出对应于上图中四个实例的网络拓扑信息。出于本示例的演示目的，示例中包含注释。

输出中的以下信息需要注意：

- NetworkNodes 描述了单个实例的网络节点集。
- 在每个网络节点集中，网络节点按从上到下的分层顺序列出。
- 连接到实例的网络节点是列表中最后的网络节点（底层）。
- 要确定哪些实例彼此靠近，请先在底层找到共同的网络节点。如果底层没有共同的网络节点，则在上层中查找共同的网络节点。

在以下示例输出中，i-1111111111example 和 i-2222222222example 与本示例中的其他实例相比彼此最靠近，因为其在底层具有共同的网络节点 nn-4444444444example。

> **Note**
>
> 响应中包含 3 个或以上网络节点。要了解每种受支持实例类型的响应中的网络节点数量，请参阅实例类型。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",  //Corresponds to instance 1
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 in layer i
                "nn-2222222222example",           //Corresponds to NN2 in layer ii
                "nn-4444444444example"            //Corresponds to NN4 in layer iii -
 bottom layer, connected to the instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
```

```
        {
            "InstanceId": "i-2222222222example",  //Corresponds to instance 2
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-4444444444example"            //Corresponds to NN4 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",  //Corresponds to instance 3
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",           //Corresponds to NN1 - layer i
                "nn-2222222222example",           //Corresponds to NN2 - layer ii
                "nn-5555555555example"            //Corresponds to NN5 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-444444444example",  //Corresponds to instance 4
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",          //Corresponds to NN1 - layer i
                "nn-3333333333example",          //Corresponds to NN3 - layer ii
                "nn-6666666666example"           //Corresponds to NN6 - layer iii -
 connected to instance
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

如何解释 DescribeCapacityReservationTopology 输出

您可以使用 DescribeCapacityReservationTopology API 来描述容量预留拓扑。输出提供了预留容量底层网络拓扑的分层视图。

以下示例输出对应于上图中的网络拓扑信息。出于本示例的演示目的，示例中包含注释。

输出中的以下信息需要注意：

- NetworkNodes 描述了单个容量预留的网络节点集。
- 在每个网络节点集中，网络节点按从上到下的分层顺序列出。
- 连接到容量预留的网络节点是列表中最后的网络节点（底层）。
- 要确定容量预留是否彼此靠近，请先在输出的底层中找到共同的网络节点。如果底层没有共同的网络节点，则在上层中查找共同的网络节点。

在以下示例输出中，cr-1111111111example 位于 nn-2222222222example 上，而 cr-2222222222example 位于 nn-3333333333example 上。由于容量预留位于 layer ii 中的不同网络节点上，因此从一个容量预留中的实例到其他容量预留中的实例的通信效率会很低。

> **Note**
>
> 响应包含 1、2 或 3 个网络节点，具体取决于容量预留的类型。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1111111111example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",      //Corresponds to NN1 - layer i
                "nn-2222222222example"       //Corresponds to NN2 - layer ii
                // Visibility of additional nodes requires an instance launch and
                // the DescribeInstanceTopology API
            ],
            "AvailabilityZone": "us-west-2a"
        },
```

```
        {
            "CapacityReservationId": "cr-2222222222example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",      //Corresponds to NN1 - layer i
                "nn-3333333333example"       //Corresponds to NN3 - layer ii
                // Visibility of additional nodes requires an instance launch and
                // the DescribeInstanceTopology API
            ],
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

DescribeInstanceTopology 与 DescribeCapacityReservationTopology 之间的区别

下表比较 DescribeInstanceTopology 与 DescribeCapacityReservationTopology API 之间的主要区别：

比较点	DescribeInstanceTopology	DescribeCapacityReservationTopology
使用阶段	启动后（执行模式）	预启动（规划和管理模式）
主要目的	优化正在运行的实例上的工作负载	实例启动之前的容量规划和容量预留管理（合并、拆分、分配）
网络节点数量	显示正在运行的实例的所有节点。如果实例位于容量预留中，则第一个节点将匹配相应的容量预留拓扑，然后是连接到该实例的其他节点。	显示部分节点集，而这些节点会因容量预留状态（pending 或 active）和类型而异。*
州	实例必须处于 running 状态	容量预留必须处于 pending 或 active 状态
使用案例	• 工作负载优化	• 容量规划

比较点	DescribeInstanceTopology	DescribeCapacityReservationTopology
	• 性能优化 • 运行时拓扑分析	• 容量预留管理（合并/拆分/分配） • 预启动拓扑评测

* 对于适用于 Ultraservers 的容量块，描述 `active` 容量预留或其正在运行的实例的拓扑时，网络节点集是相同的。

Amazon EC2 拓扑的先决条件

要描述您的 Amazon EC2 拓扑，请确保您的实例和容量预留满足以下先决条件。

以下项目的先决条件：

- Amazon Web Services 区域
- 实例类型
- 州
- IAM 权限

Amazon Web Services 区域

支持 Amazon Web Services 区域：

- 美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（加利福尼亚北部）、美国西部（俄勒冈州）
- 非洲（开普敦）
- 亚太地区（雅加达）、亚太地区（香港）、亚太地区（海得拉巴）、亚太地区（墨尔本）、亚太地区（孟买）、亚太地区（大阪）、亚太地区（首尔）、亚太地区（新加坡）、亚太地区（悉尼）、亚太地区（东京）
- 加拿大（中部）
- 欧洲地区（法兰克福）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（巴黎）、欧洲（西班牙）、欧洲地区（斯德哥尔摩）、欧洲（苏黎世）
- 以色列（特拉维夫）

- 中东（巴林）、中东（阿联酋）
- 南美洲（圣保罗）
- Amazon GovCloud（美国西部）

以色列（特拉维夫）和 Amazon GovCloud（美国西部）不支持 DescribeCapacityReservationTopology API。

实例类型

支持的实例类型：

- 响应中返回 3* 个网络节点
  - `g6e.xlarge | g6e.2xlarge | g6e.4xlarge | g6e.8xlarge | g6e.12xlarge | g6e.16xlarge | g6e.24xlarge | g6e.48xlarge | g7e.2xlarge | g7e.4xlarge | g7e.8xlarge | g7e.12xlarge | g7e.24xlarge | g7e.48xlarge`
  - `hpc6a.48xlarge | hpc6id.32xlarge | hpc7g.4xlarge | hpc7g.8xlarge | hpc7g.16xlarge | hpc7a.12xlarge | hpc7a.24xlarge | hpc7a.48xlarge | hpc7a.96xlarge | hpc8a.96xlarge`
  - `p3dn.24xlarge | p4d.24xlarge | p4de.24xlarge | p5.48xlarge | p5e.48xlarge | p5en.48xlarge | p6e-gb200.36xlarge`
  - `trn1.2xlarge | trn1.32xlarge | trn1n.32xlarge | trn2.48xlarge | trn2u.48xlarge`
- 响应中返回 4* 个网络节点
  - `p6-b200.48xlarge | p6-b300.48xlarge`

* 返回的网络节点数量仅在使用 DescribeInstanceTopology API 时适用。对于 DescribeCapacityReservationTopology API，返回的网络节点数量将因容量预留的类型和状态而异。

可用的实例类型因区域而异。有关更多信息，请参阅 Amazon EC2 instance types by Region。

州

- 对于 `DescribeInstanceTopology` – 示例必须处于 `running` 状态。
- 对于 `DescribeCapacityReservationTopology` – 容量预留必须处于 `pending` 或 `active` 状态。

您无法获取处于任何其他状态的实例或容量预留的拓扑信息。

IAM 权限

IAM 身份（用户、用户组或角色）需要以下权限：

- `ec2:DescribeInstanceTopology`
- `ec2:DescribeCapacityReservationTopology`

Amazon EC2 实例拓扑的示例

您可以使用 describe-instance-topology 命令来描述 EC2 实例的拓扑。另外，您可以使用 describe-capacity-reservation-topology 命令来描述容量预留的拓扑。

当您使用不带参数或筛选条件的 `describe-instance-topology` 或 `describe-capacity-reservation-topology` 命令时，响应将包括与指定区域中该命令支持的实例类型相匹配的所有实例或容量预留（具体取决于所使用的命令）。您可以通过包含 `--region` 参数或通过设置默认区域来指定区域。有关设置默认区域的更多信息，请参阅 为您的 Amazon EC2 资源选择区域。

您可以包含参数以返回与指定实例或容量预留 ID 或置放群组名称相匹配的实例或容量预留。您还可以包含筛选条件以返回与指定实例类型或实例系列匹配的实例或容量预留，或者返回指定可用区或本地区域中的实例或容量预留。您可以包含单个参数或筛选条件，也可以包含参数和筛选条件的组合。

输出是分页的，默认情况下每页最多 20 个实例或容量预留。您可以使用 `--max-results` 参数为每页指定最多 100 个实例或容量预留。

有关更多信息，请参阅describe-instance-topology和describe-reservation-topology-topology。

所需的权限

需要以下权限：

- `ec2:DescribeInstanceTopology` – 用于描述实例拓扑
- `ec2:DescribeCapacityReservationTopology` – 用于描述容量预留拓扑

示例

- 示例 1：DescribeInstanceTopology - 实例 ID
- 示例 2：DescribeInstanceTopology - 置放群组名称参数
- 示例 3：DescribeInstanceTopology - 实例类型筛选条件

- 示例 3a – 指定实例类型的精确匹配筛选条件
- 示例 3b – 实例系列的通配符筛选条件
- 示例 3c – 组合的实例系列和精确匹配筛选条件
- 示例 4：DescribeInstanceTopology - 区域 ID 筛选条件
  - 示例 4a – 可用区筛选条件
  - 示例 4b – 本地区域筛选条件
  - 示例 4c – 组合的可用区和本地区域筛选条件
- 示例 5：DescribeInstanceTopology - 实例类型和区域 ID 筛选条件
- 示例 6：DescribeCapacityReservationTopology - 容量预留 ID
- 示例 7：DescribeCapacityReservationTopology - 实例类型筛选条件

示例 1：DescribeInstanceTopology - 实例 ID

Amazon CLI

描述特定实例的拓扑

将 describe-instance-topology 命令与 `--instance-ids` 参数一起使用。输出内容会包含与指定实例 ID 匹配的实例。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --instance-ids i-1111111111example i-2222222222example
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
```

```
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -InstanceId i-1111111111example, i-2222222222example
```

示例 2：DescribeInstanceTopology - 置放群组名称参数

Amazon CLI

描述特定置放群组中实例的拓扑

将 describe-instance-topology 命令与 `group-names` 参数一起使用。输出内容会包含位于任一指定置放组中的实例。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --group-names ML-group HPC-group
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "GroupName": "HPC-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定置放群组中实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -GroupName ML-group, HPC-group
```

示例 3：DescribeInstanceTopology - 实例类型筛选条件

您可以按指定的实例类型（精确匹配）进行筛选，也可以按实例系列进行筛选（使用通配符）。您也可以将指定的实例类型筛选条件和实例系列筛选条件组合使用。

示例 3a – 指定实例类型的精确匹配筛选条件

Amazon CLI

描述特定实例类型的实例拓扑

将 describe-instance-topology 命令与 `instance-type` 筛选条件一起使用。输出内容仅包含指定实例类型的实例。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1n.32xlarge
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定实例类型的实例拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1n.32xlarge"}
```

示例 3b – 实例系列的通配符筛选条件

Amazon CLI

描述特定实例系列的实例拓扑

将 describe-instance-topology 命令与 instance-type 筛选条件一起使用。输出内容仅包含指定实例系列的实例。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
    --filters Name=instance-type,Values=trn1*
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
```

```
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
                "nn-1235301234example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定实例系列的实例拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="trn1*"}
```

示例 3c – 组合的实例系列和精确匹配筛选条件

Amazon CLI

描述某个实例系列或具有实例类型的实例的拓扑

将 describe-instance-topology 命令与 `instance-type` 筛选条件一起使用。输出内容仅包含符合指定条件的实例。

```
aws ec2 describe-instance-topology \
    --region us-west-2 \
```

```
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge"
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-4343434343example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述某个实例系列或具有实例类型的实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"}
```

示例 4：DescribeInstanceTopology - 区域 ID 筛选条件

您可以使用 zone-id 筛选条件按可用区或本地区域进行筛选。您也可以将可用区筛选条件和本地区域筛选条件组合使用。

示例 4a – 可用区筛选条件

Amazon CLI

描述特定可用区中的实例的拓扑

将 describe-instance-topology 命令与 zone-id 筛选条件一起使用。输出内容仅包含指定可用区中的实例。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定可用区中的实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1"}
```

示例 4b – 本地区域筛选条件

Amazon CLI

描述特定本地区域中的实例的拓扑

将 describe-instance-topology 命令与 zone-id 筛选条件一起使用。输出内容仅包含指定本地区域中的实例。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-atl2-az1
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定本地区域中的实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-atl2-az1"}
```

示例 4c – 组合的可用区和本地区域筛选条件

Amazon CLI

描述特定区域中实例的拓扑

将 describe-instance-topology 命令与 zone-id 筛选条件一起使用。输出内容仅包含位于任一指定区域中的实例。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters Name=zone-id,Values=use1-az1,use1-atl2-az1
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
```

```
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定区域中实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

示例 5：DescribeInstanceTopology - 实例类型和区域 ID 筛选条件

您可以将筛选条件组合在单个命令中。

Amazon CLI

描述具有特定实例类型、实例系列和区域的实例的拓扑

将 describe-instance-topology 命令与 `instance-type` 和 `zone-id` 筛选条件一起使用。响应包含任一指定实例类型的、位于任一指定区域的任何实例。

```
aws ec2 describe-instance-topology \
    --region us-east-1 \
    --filters "Name=instance-type,Values=p4d*,trn1n.32xlarge" \
              "Name=zone-id,Values=use1-az1,use1-atl2-az1"
```

下面是示例输出。

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
```

```
            "InstanceType": "p4d.24xlarge",
            "GroupName": "ML-group",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-atl2-az1",
            "AvailabilityZone": "us-east-1-atl-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "trn1n.32xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3214313214example"
            ],
            "CapacityBlockId": "null",
            "ZoneId": "use1-az1",
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述具有特定实例类型、实例系列和区域的实例的拓扑

使用 Get-EC2InstanceTopology cmdlet。

```
Get-EC2InstanceTopology `
    -Filter @{Name="instance-type"; Values="p4d*", "trn1n.32xlarge"} `
            @{Name="zone-id"; Values="use1-az1", "use1-atl2-az1"}
```

示例 6：DescribeCapacityReservationTopology - 容量预留 ID

Amazon CLI

描述特定容量预留的拓扑

将 describe-capacity-reservation-topology 命令与 capacity-reservation-id 参数一起使用。输出仅包含与指定容量预留 ID 匹配的容量预留。

```
aws ec2 describe-capacity-reservation-topology \
    --region us-east-1 \
    --capacity-reservation-id cr-1111111111example cr-2222222222example
```

下面是示例输出。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1111111111example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p5.48xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example"
            ],
            "AvailabilityZone": "us-east-1a"
        },
        {
            "CapacityReservationId": "cr-2222222222example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p5en.48xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example"
            ],
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述特定容量预留的拓扑

使用 Get-EC2CapacityReservationTopology cmdlet。

```
Get-EC2CapacityReservationTopology `
    -CapacityReservationId cr-1111111111example cr-2222222222example
```

示例 7：DescribeCapacityReservationTopology - 实例类型筛选条件

您可以按指定的实例类型（精确匹配）进行筛选，也可以按实例系列进行筛选（使用通配符）。您也可以将指定的实例类型筛选条件和实例系列筛选条件组合使用。

Amazon CLI

描述具有特定实例类型的容量预留的拓扑

将 describe-capacity-reservation-topology 命令与 `instance-type` 筛选条件一起使用。响应包含具有特定实例类型的所有实例。

```
aws ec2 describe-capacity-reservation-topology \
    --region us-east-1 \
    --filters Name=instance-type,Values=p5en.48xlarge
```

下面是示例输出。

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-2222222222example",
            "CapacityBlockId": "null",
            "State": "active",
            "InstanceType": "p5en.48xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example"
            ],
            "AvailabilityZone": "us-east-1a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

PowerShell

描述具有特定实例类型的容量预留的拓扑

使用 Get-EC2CapacityReservationTopology cmdlet。

```
Get-EC2CapacityReservationTopology `
    -Filter @{Name="instance-type"; Values="p5en.48xlarge"}
```

Amazon EC2 实例的置放群组

为了满足工作负载的需求，您可以将一组相互依赖的 EC2 实例启动到一个置放群组中，以影响它们的置放。

根据工作负载类型，您可以使用以下置放策略之一创建置放群组：

- 集群 – 将一个可用区内靠近的实例打包在一起。通过使用该策略，工作负载可以实现所需的低延迟网络性能，以满足高性能计算（HPC）应用程序通常使用的紧密耦合的节点到节点通信的要求。
- 分区 – 将实例分布在不同的逻辑分区上，以便一个分区中的实例组不会与不同分区中的实例组使用相同的基础硬件。该策略通常为大型分布式和重复的工作负载所使用，例如，Hadoop、Cassandra 和 Kafka。
- 分布 – 将一小组实例严格放置在不同的基础硬件上，以减少相关的故障。

置放群组是可选的。如果不将您的实例启动到一个置放群组中，EC2 会尝试以某种方式放置实例，以便将所有实例分布在基础硬件上，从而最大限度减少相关的故障。

定价

创建置放群组无需支付费用。

规则和限制

在使用置放群组之前，请注意以下规则：

- 一个实例一次只能放在一个置放群组中；您不能将一个实例放在多个置放群组中。
- 不能合并置放群组。
- 按需容量预留和区域预留实例可让您为可用区中的 EC2 实例提供容量预留。当您启动实例时，如果实例属性与按需容量预留或区域预留实例指定的属性相匹配，则该实例将自动使用预留容量。如果您将实例启动到置放群组，情况也是如此。
- 您无法在置放群组中启动专属主机。
- 您无法在置放群组中启动已配置为中断时停止或睡眠的竞价型实例。

内容

- 置放群组的置放策略
- 为 EC2 实例创建置放群组
- 更改 Amazon EC2 实例的置放
- 删除置放群组
- 共享置放群组
- Amazon Outposts 上的置放群组

置放群组的置放策略

可以使用以下某种置放策略来创建 EC2 实例的置放群组。

置放策略

- 集群置放群组
- 分区置放群组
- 分布置放群组

集群置放群组

集群置放群组是单个可用区中的实例的逻辑分组。实例未隔离到单个机架上。集群置放群组可横跨同一区域中的对等虚拟专用网络（VPC）。同一集群置放群组中的实例可针对 TCP/IP 流量享受更高的每流吞吐量限制，并且这些实例将处于网络的同一高等分带宽段中。

下图显示放入集群置放群组中的实例。

建议将集群置放群组用于可受益于低网络延迟和/或高网络吞吐量的应用程序。如果大部分网络流量在组中的实例之间进行，也建议使用集群置放群组。要为置放群组提供最低延迟和最高每秒数据包数的网络性能，请选择支持增强联网的实例类型。有关更多信息，请参阅增强联网。

我们建议您按照以下方法启动您的实例：

- 使用单个启动请求在置放群组中启动所需的实例数。
- 对置放群组中的所有实例使用相同的实例类型。

如果您以后尝试将更多实例添加到置放群组，或者如果您尝试在置放群组中启动多个实例类型，都会增大发生容量不足错误的可能性。

如果您停止置放群组中的某个实例，然后重启该实例，则其仍将在该置放群组中运行。但是，如果没有足够容量可用于该实例，则启动将会失败。

如果您在已有正在运行的实例的置放群组中启动实例时接收到容量错误信息，请在该置放群组中停止并启动所有实例，然后尝试再次启动。启动实例可能会将实例迁移至具有针对所有请求实例的容量的硬件。

规则和限制

以下规则适用于集群置放群组：

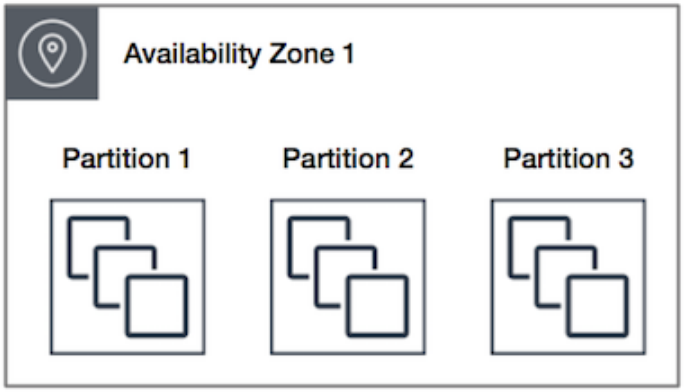
- 支持以下实例类型：
  - 最新一代实例，但不包括可突增性能实例（例如 T2）和 M7i-flex 实例。
  - 以下上一代实例：A1、C3、C4、I2、M4、R3 和 R4。
- 一个集群置放群组不能跨过多个可用区。
- 集群置放群组中的两个实例之间的最大网络吞吐量流量速度受两个实例中的较慢实例限制。对于具有高吞吐量要求的应用程序，请选择其网络连接满足您要求的实例类型。
- 对于启用了增强联网的实例，以下规则适用：
  - 对于单个流的流量，集群置放群组中的实例最多可以使用 10 Gbps。对于单个流的流量，不在集群置放群组中的实例最多可以使用 5 Gbps。
  - 在同一个区域中，通过公有 IP 地址空间或者通过 VPC 终端节点往返于 Amazon S3 存储桶之间的流量可以使用所有可用的实例聚合带宽。
- 您可以将多种类型的实例启动到集群置放群组中。不过，这会降低提供所需容量以成功完成启动的可能性。我们建议集群置放群组中的所有实例使用相同的实例类型。

- 我们建议您通过在集群置放群组中创建按需容量预留来明确集群置放群组中的预留容量。请注意，您无法使用区域预留实例预留容量，因为它们无法在置放群组中明确预留容量。
- 对于集群放置组，指向互联网的网络流量以及通过 Amazon Direct Connect 连接指向本地资源的流量限制为 5 Gbps。

分区置放群组

分区置放群组可帮助您的应用程序减少相关硬件故障的可能性。在使用分区置放群组时，Amazon EC2 将每个群组划分为多个逻辑段（称为“分区”）。Amazon EC2 确保置放群组中的每个分区具有自己的一组机架。每个机架具有自己的网络和电源。置放群组中的任何两个分区将不会分享相同的机架，从而让您可以在您的应用程序中隔离硬件故障的影响。

下图是单个可用区中的分区置放群组简单的直观表示。它显示了放入到一个分区置放群组的实例，该置放群组具有三个分区 — 分区 1、分区 2 和 分区 3。每个分区均包含多个实例。一个分区中的实例不与其他分区中的实例共享机架，这使您可以将单一硬件故障的影响限定在相关的分区内。



可使用分区置放群组跨不同机架部署大型分布式和重复的工作负载，例如 HDFS、HBase 和 Cassandra。当您在分区置放群组中启动实例时，Amazon EC2 将尝试跨您指定数量的分区均匀分发实例。您还可以在特定分区中启动实例，以更好地控制实例的放置位置。

分区置放群组可以在同一区域的多个可用区中具有分区。对于每个可用区，一个分区置放群组最多可具有 7 个分区。可在分区置放群组启动的实例的数量仅受账户限制的限制。

此外，分区置放群组提供对分区的可见性，您可以查看哪些实例位于哪些分区中。您可以与具有感知技术能力的应用程序共享此信息，例如 HDFS、HBase 和 Cassandra。这些应用程序使用此信息做出智能数据复制决策，用于提升数据的可用性和持久性。

如果在一个分区置放群组中启动一个实例，而没有足够的独特硬件来满足请求，则请求将失败。Amazon EC2 会随着时间的推移提供更多不同的硬件，因此，您稍后可以重试请求。

规则和限制

以下规则适用于分区置放群组：

- 对于每个可用区，一个分区置放群组最多支持 7 个分区。您可在分区置放群组中启动的实例的数量仅受账户限制的限制。
- 当您在分区置放群组中启动实例时，Amazon EC2 将尝试跨所有分区均匀分发实例。Amazon EC2 不保证跨所有分区均匀分发实例。
- 具有 专用实例 的分区置放群组最多可具有 2 个分区。
- 容量预留不会在分区放置组中预留容量。

分布置放群组

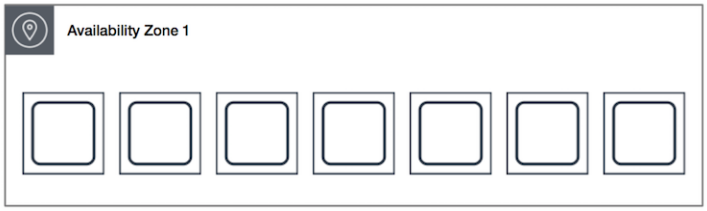
分布置放群组是将实例分别置放在不同硬件上的实例群组。

建议在具有少量应单独放置的重要实例的应用程序中使用分布置放群组。通过在分布置放群组中启动实例，可以降低在实例位于同一设备中时同时出现故障的风险。分布置放群组使用不同的硬件，因而适合混用不同类型的实例或随着时间推移逐步启动实例的场景。

如果在一个排列置放群组中启动一个实例，而没有足够的独特硬件来满足请求，则请求将失败。Amazon EC2 会随着时间的推移提供更多不同的硬件，因此，您稍后可以重试请求。置放群组可以跨机架或主机分布实例。机架级别的分布置放群组可在 Amazon 区域和 Amazon Outposts 中使用。主机级别的分布置放群组只能用于 Amazon Outposts。

机架级别的分布置放群组

下图显示单个可用区中的 7 个实例，这些实例已放入一个分布置放群组。将七个实例置放在七个不同的机架上，每个机架都具有自己的网络和电源。



机架级别的分布置放群组可以跨越同一区域中的多个可用区。在一个区域中，每个机架级别的分布置放群组在每个可用区中最多可以运行七个实例。通过 Outpost，机架级别的分布置放群组可以容纳与 Outpost 部署中机架数量一样多的实例。

主机级别分布置放群组

主机级别的分布置放群组仅适用于 Amazon Outposts。主机级别的分布置放群组可以容纳与 Outpost 部署中的主机数量一样多的实例。有关更多信息，请参阅 the section called “Amazon Outposts 上的置放群组”。

规则和限制

以下规则适用于分布置放群组：

- 机架分布置放群组最多支持在每个可用区运行七个实例。例如，在具有三个可用区的区域中，您可以在组中总共运行 21 个实例，每个可用区 7 个实例。如果您尝试在同一可用区和同一个分布置放群组中启动第八个实例，则该实例将无法启动。如果需要在一个可用区中运行超过七个实例，则建议使用多个分布置放群组。使用多个分布置放群组并不能保证实例在群组之间分布，但可帮助确保每个群组内的分布，从而限制某些故障类别的影响。
- 专用实例 不支持分布置放群组。
- 主机级别分布置放群组仅适用于 Amazon Outposts 上的置放群组。主机级别的分布置放群组可以容纳与 Outpost 部署中的主机数量一样多的实例。
- 在一个区域中，每个机架级别的分布置放群组在每个可用区中最多可以运行七个实例。通过 Amazon Outposts，机架级别的分布置放群组可以容纳与 Outpost 部署中机架数量一样多的实例。
- 容量预留不会在分布放置组中预留容量。

为 EC2 实例创建置放群组

您可以使用置放群组来控制实例相对于彼此的置放。创建置放群组后，您可以启动置放群组中的实例。

限制

每个区域最多可创建 500 个置放群组。

Console

创建置放群组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Placement Groups。
3. 选择创建置放群组。
4. 对于名称，请指定群组的名称。
5. 对于置放策略，为群组选择置放策略：集群、分布或分区。

a. 如果选择分布，则必须选择分布级别：主机或机架。

b. 如果选择分区，则在分区数中输入该群组的分区数。

6. （可选）要添加标签，请在标签下选择添加新标签，然后输入键和值。

7. 选择创建群组。

Amazon CLI

使用 create-placement-group 命令。

创建集群置放群组

以下示例创建一个使用 purpose 置放策略的置放群组，并应用键为 cluster 且值为 production 的标签。

```
aws ec2 create-placement-group \
    --group-name my-cluster \
    --strategy cluster \
    --tag-specifications 'ResourceType=placement-
group,Tags={Key=purpose,Value=production}'
```

创建分区置放群组

以下示例创建一个使用 partition 置放策略的置放群组，并使用 --partition-count 参数指定五个分区。

```
aws ec2 create-placement-group \
    --group-name HDFS-Group-A \
    --strategy partition \
    --partition-count 5
```

PowerShell

使用 New-EC2PlacementGroup cmdlet。

创建集群置放群组

以下示例创建了一个集群置放群组。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
```

```
    -Strategy cluster
```

创建分区置放群组

以下示例创建一个分区置放群组。

```
New-EC2PlacementGroup `
    -GroupName my-placement-group `
    -Strategy partition `
    -PartitionCount 5
```

更改 Amazon EC2 实例的置放

可以如下所示更改实例的置放群组：

- 将实例添加到置放群组
- 将实例从一个置放群组移动到另一个置放群组
- 从置放群组中删除实例

要求

在更改实例的置放群组之前，该实例必须处于 stopped 状态。

Console

更改实例置放

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 依次选择操作、实例设置、修改实例放置。
5. 对于置放群组，请执行以下操作之一：
   - 要将实例添加到置放群组，请选择该置放群组。
   - 要将实例从一个置放群组移动到另一个，请选择该置放群组。
   - 要从置放群组中移除实例，请选择无。
6. 选择保存。

Amazon CLI

将实例移动到置放群组

使用以下 modify-instance-placement 命令。

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name MySpreadGroup
```

从置放群组中删除实例

使用以下 modify-instance-placement 命令。当您为置放群组名称指定空字符串时，这会将实例从当前所在的置放群组中移除。

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

PowerShell

将实例移动到置放群组

使用 Edit-EC2InstancePlacement cmdlet 和置放群组的名称。

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName MySpreadGroup
```

从置放群组中删除实例

使用 Edit-EC2InstancePlacement cmdlet，并将置放群组名称设为空字符串。

```
Edit-EC2InstancePlacement `
    -InstanceId i-0123a456700123456 `
    -GroupName ""
```

删除置放群组

如果您需要替换或不再需要某个置放群组，您可以将其删除。在删除置放群组之前，该群组中不能包含任何实例。您可以终止实例，将这些实例移动到另一个置放群组，或者将其从置放群组中删除。

Console

删除置放群组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Placement Groups。
3. 选择该置放群组，然后依次选择操作、删除。
4. 提示进行确认时，输入 **Delete**，然后选择 Delete（删除）。

Amazon CLI

删除置放群组

使用 delete-placement-group 命令。

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

删除置放群组

使用 Remove-EC2PlacementGroup cmdlet。

```
Remove-EC2PlacementGroup -GroupName my-cluster
```

共享置放群组

通过置放群组共享，您可以影响由单独 Amazon Web Services 账户拥有的相互依赖实例的置放。所有者可在多个 Amazon Web Services 账户 之间或在其组织内共享一个置放群组。参与者可以在与其账户共享的置放群组中启动实例。

置放群组所有者可与以下人员共享置放群组：

- 其组织内部或外部的特定 Amazon 账户
- 其 组织内的组织部门
- 其整个 组织

可以使用 VPC 对等连接独立 Amazon 账户拥有的实例，获得共享集群置放群组提供的全部延迟优势。

内容

- 规则和限制
- 所需的权限
- 跨可用区共享
- 置放群组共享
- 取消置放群组共享

规则和限制

当您共享置放群组或与您共享置放群组时，请遵循以下规则和限制。

- 要共享置放群组，您必须在您的 Amazon 账户中拥有该置放群组。无法共享已与自己共享的置放群组。
- 共享分区或扩展分布置放群组时，置放群组限制不会改变。共享分区置放群组支持每个可用区最多七个分区，共享分布置放群组支持每个可用区最多七个运行实例。
- 要与您的组织或组织内的组织部门共享放置组，您必须允许与 Amazon Organizations 共享。有关更多信息，请参阅共享 Amazon 资源。
- 使用 Amazon Web Services 管理控制台 启动实例时，可以选择已与自己共享的任何置放群组。使用 Amazon CLI 启动实例时，必须按 ID 而非名称来指定共享置放群组。只有当您是共享置放群组的所有者时，才能使用置放群组的名称。
- 您负责管理共享置放群组中您拥有的实例。
- 您无法查看或修改与共享置放群组有关但不属于您的实例和容量预留。
- 置放群组的 Amazon 资源名称（ARN）包含拥有该置放群组的账户 ID。您可以使用置放群组 ARN 的账户 ID 部分来识别与您共享的置放群组的所有者。

所需的权限

要共享置放群组，用户必须具有执行以下操作的权限：

- `ec2:PutResourcePolicy`
- `ec2:DeleteResourcePolicy`

跨可用区共享

为确保资源分配到区域的各可用区，我们将可用区独立映射到每个账户的名称。这可能会导致账户之间的可用区命名差异。例如，您的 us-east-1a 账户的可用区 Amazon 可能与另一 us-east-1a 账户的 Amazon 不在同一位置。

要指定专属主机相对于账户的位置，您必须使用可用区 ID（AZ ID）。AZ ID 是跨所有 Amazon 账户的可用区的唯一且一致的标识符。例如，use1-az1 是 us-east-1 区域的可用区 ID，它在每个 Amazon 账户中的位置均相同。有关更多信息，请参阅 AZ IDs。

置放群组共享

要共享置放群组，您必须将其添加到资源共享。资源共享是一项 Amazon RAM 资源，可让您跨 Amazon 账户共享资源。资源共享指定要共享的资源以及与之共享资源的使用者。

如果您属于 Amazon Organizations 中的某个组织并且已在您的组织中启用共享，则组织中的使用者将获得对共享置放群组的访问权限。

如果将置放群组与您组织以外的 Amazon 账户共享，则 Amazon 账户所有者将会收到加入资源共享的邀请。他们可在接受邀请后访问共享的置放群组。

您可以使用 Amazon Resource Access Manager 在多个 Amazon 账户之间共享置放群组。有关更多信息，请参阅《Amazon RAM User Guide》中的 Creating a resource share。

取消置放群组共享

置放群组所有者随时可以将共享置放群组取消共享。将共享置放群组取消共享时，将发生以下更改：

- 与之共享置放群组的 Amazon 账户将无法再启动实例或预留容量。
- 任何正在共享置放群组中运行的实例都将与该置放群组断开关联，但会继续在 Amazon 账户中运行。
- 共享置放群组中的所有容量预留容量都将与该置放群组断开关联，但仍可在 Amazon 账户中可用。

有关更多信息，请参阅《Amazon RAM User Guide》中的 Deleting a resource share。

Amazon Outposts 上的置放群组

Amazon Outposts 是一项完全托管的服务，可将 Amazon 基础设施、服务、API 和工具扩展到客户场所。通过提供对 Amazon 托管基础设施的本地访问，Amazon Outposts 使客户能够使用与 Amazon 区

域中相同的编程接口在本地构建和运行应用程序，同时使用本地计算和存储资源来满足更低的延迟和本地数据处理需求。

Outpost 是部署在客户站点的 Amazon 计算和存储容量池。Amazon 作为 Amazon 区域的一部分运营、监控和管理此容量。

您可以在您在账户中创建的 Outpost 上创建置放群组。从而将实例分布在您站点中的 Outpost 上的底层硬件上。在 Outpost 上创建和使用置放群组的方式与您在常规可用区中创建和使用置放群组时相同。当您在 Outpost 上创建具有分布策略的置放群组时，您可以选择让置放群组跨主机或机架分布实例。跨主机分布实例允许您使用具有单个机架 Outpost 的分布策略。

注意事项

- 机架级别的分布置放群组可以容纳与 Outpost 部署中机架数量一样多的实例。
- 主机级别的分布置放群组可以容纳与 Outpost 部署中的主机数量一样多的实例。

先决条件

您的站点必须安装一个 Outpost。有关更多信息，请参阅 Amazon Outposts 用户指南中的创建 Outpost 并订购 Outpost 容量。

在 Outpost 上使用置放群组

1. 在 Outpost 上创建子网。有关更多信息，请参阅 Amazon Outposts 用户指南中的创建子网。
2. 在与该 Outpost 关联的区域中创建一个置放群组。如果您创建具有分布策略的置放群组，则可以选择主机或机架分布级别，以确定该群组将如何在 Outpost 上的底层硬件之间分布实例。有关更多信息，请参阅 the section called “创建置放群组”。
3. 将一个实例启动到置放群组。对于 Subnet（子网），选择您在第 1 步中创建的子网，然后对于 Placement group name（置放群组名称），选择您在第 2 步中创建的置放群组。有关更多信息，请参阅 Amazon Outposts 用户指南中的在 Outpost 中启动实例。

EC2 实例的网络最大传输单位 (MTU)

网络连接的最大传输单位 (MTU) 是能够通过该连接传递的最大可允许数据包的大小（以字节为单位）。连接的 MTU 越大，可在单个数据包中传递的数据越多。以太网帧由数据包（即您发送的实际数据）以及相关网络开销信息组成。

以太网帧有不同的格式，最常见的格式是标准以太网 v2 帧格式。它支持 1500 MTU，这是大部分 Internet 上支持的最大以太网数据包大小。实例支持的最大 MTU 取决于其实例类型。

所有 EC2 实例类型都支持 1500 MTU。

内容

- 巨型帧 (9001 MTU)
- 路径 MTU 发现
- 设置 Amazon EC2 实例的 MTU
- 故障排除

巨型帧 (9001 MTU)

借助巨型帧，您可以增加每个数据包的有效载荷大小，从而增加数据包中不属于数据包开销的百分比。使用巨型帧时，发送等量的可用数据所需要的数据包更少。但是，某些类型的流量受以下最大有效载荷的约束：

MTU 限制 1500 字节

- 互联网网关流量
- VPN 连接的流量
- Amazon 区域之间的流量，除非使用中转网关

MTU 限制 8500 字节

- 区域间 VPC 对等连接中的流量

如果数据包超过其 MTU 限制，则对数据包进行分段；如果在 IP 标头中设置了 `Don't Fragment` 标记，则丢弃数据包。

对于面向 Internet 的流量或离开 VPC 的任何流量，应谨慎使用巨型帧。中间系统会对数据包进行分段，从而减缓此流量。要使用 VPC 中的巨型帧而不减慢 VPC 外部的绑定流量的速度，您可按路由配置 MTU 大小，或者将弹性网络接口与不同 MTU 大小和不同路由结合使用。

对于在集群置放群组中并置的实例，巨型帧有助于实现可能的最大网络吞吐量，建议在这种情况下使用这些帧。有关更多信息，请参阅Amazon EC2 实例的置放群组。

您可以通过 Amazon Direct Connect 使用巨型帧在 VPC 与本地网络之间进行通信。有关更多信息以及验证巨型帧功能的方法，请参阅《Amazon Direct Connect 用户指南》中的 MTU for private virtual interfaces or transit virtual interfaces。

所有当前一代实例类型都支持巨型帧。下列上一代实例类型支持巨型帧：A1、C3、I2、M3 和 R3。

相关资源

- 对于 NAT 网关，请参阅《Amazon VPC 用户指南》中的 NAT 网关基础知识。
- 对于中转网关，请参阅 Amazon VPC 中转网关用户指南中的 Maximum transmission unit。
- 对于 Local Zones，请参阅《Amazon Local Zones User Guide》中的 Considerations。
- 对于 Amazon Wavelength，请参阅 Amazon Wavelength 用户指南中的 Maximum transmission unit。
- 对于 Outposts，请参阅 Amazon Outposts 用户指南中的 Service link maximum transmission unit requirements。

路径 MTU 发现

路径 MTU 发现（PMTUD）用于确定两台设备之间的路径 MTU。路径 MTU 是原始主机和接收主机之间的路径所支持的最大数据包大小。当两个主机的网络中 MTU 大小存在差异时，PMTUD 使接收主机能够使用 ICMP 消息响应原始主机。此 ICMP 消息将指示原始主机在网络路径中使用最低 MTU 大小并重新发送请求。若无此协商，当请求过大，导致接收主机无法接收时，数据包可能会丢失。

对于 IPv4，如果主机发送一个大于接收主机的 MTU 或大于路径上某台设备的 MTU 的数据包，则接收主机或设备将删除此数据包，然后返回以下 ICMP 消息：`Destination Unreachable: Fragmentation Needed and Don't Fragment was Set`（类型 3，代码 4）。这将指示传输主机将有效负载拆分为多个较小的数据包，然后重新传输。

IPv6 协议不支持网络中的分段。如果主机发送一个大于接收主机的 MTU 或大于路径上某台设备的 MTU 的数据包，则接收主机或设备将删除此数据包，然后返回以下 ICMP 消息：`ICMPv6 Packet Too Big (PTB)`（类型 2）。这将指示传输主机将有效负载拆分为多个较小的数据包，然后重新传输。

通过某些组件（例如 NAT 网关和负载均衡器）建立的连接会自动跟踪。这意味着将为您的出站连接尝试自动启用安全组跟踪。如果自动跟踪连接或者您的安全组规则允许入站 ICMP 流量，您可能会收到 PMTUD 响应。

请注意，即使在安全组级别允许流量，也可能会阻止 ICMP 流量；例如您的网络访问控制列表条目拒绝 ICMP 流量进入子网时。

> **Important**
>
> 路径 MTU 发现不能保证某些路由器不丢弃巨型帧。您 VPC 中的互联网网关仅将转发最多 1500 字节的数据包。建议对 Internet 流量使用 1500 MTU 数据包。

有关 NAT 网关的 MTU 规则，请参阅《Amazon VPC 用户指南》中的最大传输单元（MTU）。有关中转网关的 MTU 规则，请参阅《Amazon Transit Gateway User Guide》中的 Maximum transmission unit (MTU)。

设置 Amazon EC2 实例的 MTU

网络连接的最大传输单位 (MTU) 是能够通过该连接传递的最大可允许数据包的大小（以字节为单位）。所有 Amazon EC2 实例都支持标准帧（1500 MTU），所有当前一代实例类型都支持巨型帧（9001 MTU）。

您可以查看 Amazon EC2 实例的 MTU，查看您的实例与其他主机之间的路径 MTU，并将您的实例配置为使用标准帧或巨型帧。

任务

- 查看两个主机之间的路径 MTU
- 检查实例的 MTU
- 设置实例的 MTU

查看两个主机之间的路径 MTU

您可以检查 EC2 实例与另一主机之间的路径 MTU。可以指定 DNS 名称或 IP 地址作为目标。如果目标是另一个 EC2 实例，则验证其安全组是否允许入站 UDP 流量。

您使用的过程取决于实例的操作系统。

Linux 实例

在实例上运行 tracepath 命令来检查您的 EC2 实例与指定目标之间的路径 MTU。此命令是 `iputils` 软件包的一部分，默认情况下，许多 Linux 发行版中都提供该命令。

本例检查 EC2 实例与 `amazon.com` 之间的路径 MTU。

```
[ec2-user ~]$ tracepath amazon.com
```

在此示例输出中，路径 MTU 为 1500。

```
 1?: [LOCALHOST]      pmtu 9001
 1:  ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)   0.187ms pmtu 1500
 1:  no reply
 2:  no reply
 3:  no reply
 4:  100.64.16.241 (100.64.16.241)                              0.574ms
 5:  72.21.222.221 (72.21.222.221)                             84.447ms asymm 21
 6:  205.251.229.97 (205.251.229.97)                           79.970ms asymm 19
 7:  72.21.222.194 (72.21.222.194)                             96.546ms asymm 16
 8:  72.21.222.239 (72.21.222.239)                             79.244ms asymm 15
 9:  205.251.225.73 (205.251.225.73)                           91.867ms asymm 16
...
31:  no reply
     Too many hops: pmtu 1500
     Resume: pmtu 1500
```

Windows 实例

使用 mturoute 检查路径 MTU

1. 从 https://elifulkerson.com/projects/mturoute.php 将 mturoute.exe 下载到您的 EC2 实例。
2. 打开命令提示符窗口并将当前目录更改为 mturoute.exe 的下载目录。
3. 使用以下命令检查您的 EC2 实例与指定目标之间的路径 MTU。本例检查 EC2 实例与 `www.elifulkerson.com` 之间的路径 MTU。

   ```
   .\mturoute.exe www.elifulkerson.com
   ```

   在此示例输出中，路径 MTU 为 1500。

   ```
   * ICMP Fragmentation is not permitted. *
   * Speed optimization is enabled. *
   * Maximum payload is 10000 bytes. *
   + ICMP payload of 1472 bytes succeeded.
   - ICMP payload of 1473 bytes is too big.
   Path MTU: 1500 bytes.
   ```

检查实例的 MTU

您可以检查实例的 MTU 值。一些实例配置为使用巨型帧，另一些则配置为使用标准帧大小。

您使用的过程取决于实例的操作系统。

Linux 实例

查看 Linux 实例上的 MTU 设置

在您的 EC2 实例上运行以下 ip 命令。如果主网络接口不是 eth0，则请将 eth0 替换为您的网络接口。

```
[ec2-user ~]$ ip link show eth0
```

在此示例输出中，*mtu 9001* 指示实例使用了巨型帧。

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode
 DEFAULT group default qlen 1000
    link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

Windows 实例

您使用的过程取决于实例上的驱动程序。

ENA driver

版本 2.1.0 及更高版本

要获取 MTU 值，请在 EC2 实例上使用以下 Get-NetAdapterAdvancedProperty 命令。使用通配符（星号）获取所有以太网名称。检查接口名称 *JumboPacket 的输出。值为 9015 表示已启用巨型帧。默认情况下禁用巨型帧。

```
Get-NetAdapterAdvancedProperty -Name "Ethernet*"
```

版本 1.5 及更早版本

要获取 MTU 值，请在 EC2 实例上使用以下 Get-NetAdapterAdvancedProperty 命令。检查接口名称 MTU 的输出。值为 9001 表示已启用巨型帧。默认情况下禁用巨型帧。

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Intel SRIOV 82599 driver

要获取 MTU 值，请在 EC2 实例上使用以下 Get-NetAdapterAdvancedProperty 命令。检查接口名称 *JumboPacket 的条目。值为 9014 表示已启用巨型帧。(注意，MTU 的大小中包括标头和负载。) 默认情况下禁用巨型帧。

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Amazon PV driver

要获取 MTU 值，请在 EC2 实例上使用以下命令。接口的名称可能会有所不同。在输出中，查找名为“Ethernet”、“Ethernet 2”或“Local Area Connection”的条目。您需要接口名称来启用或禁用巨型帧。值为 9001 表示已启用巨型帧。

```
netsh interface ipv4 show subinterface
```

设置实例的 MTU

您可能希望将巨型帧用于 VPC 内的网络流量，并将标准帧用于互联网流量。无论是哪种用例，我们都建议您验证实例是否如预期那样运行。

您使用的过程取决于实例的操作系统。

Linux 实例

在 Linux 实例上设置 MTU 值

1. 在您的实例上运行以下 ip 命令。其将预期 MTU 值设置为 1500，但是您可以使用 9001 代替。如果主网络接口不是 eth0，请将 eth0 替换为实际网络接口。

   ```
   [ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
   ```

2. (可选) 要在重启后保留您的网络 MTU 设置，请根据您的操作系统类型修改配置文件。

   - Amazon Linux 2023：修改配置文件的 [Link] 部分。默认配置文件是 /usr/lib/systemd/network/80-ec2.network，您也可以更新在 /run/systemd/network/ 中创建的任何自定义配置文件，其中文件名为 *priority-interface*.network。有关更多信息，请参阅 Amazon Linux 文档中的 Networking service。

```
MTUBytes=1500
```

- Amazon Linux 2：将以下一行添加到 /etc/sysconfig/network-scripts/ifcfg-*eth0* 文件：

```
MTU=1500
```

将以下行添加到 /etc/dhcp/dhclient.conf 文件：

```
request subnet-mask, broadcast-address, time-offset, routers, domain-name,
 domain-search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-
servers;
```

- 其他 Linux 发行版：请参阅其具体文档。

3. (可选) 重启实例并验证 MTU 设置是否正确。

Windows 实例

您使用的过程取决于实例上的驱动程序。

ENA driver

您可以使用设备管理器或在实例上使用 Set-NetAdapterAdvancedProperty 命令更改 MTU 设置。

版本 2.1.0 及更高版本

使用以下命令启用巨型帧。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9015
```

使用以下命令禁用巨型帧。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

版本 1.5 及更早版本

使用以下命令启用巨型帧。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 9001
```

使用以下命令禁用巨型帧。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -
RegistryValue 1500
```

Intel SRIOV 82599 driver

您可以使用设备管理器或在实例上使用 Set-NetAdapterAdvancedProperty 命令更改 MTU 设置。

使用以下命令启用巨型帧。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 9014
```

使用以下命令禁用巨型帧。

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -
RegistryValue 1514
```

Amazon PV driver

您可以在实例上使用 netsh 命令更改 MTU 设置。您无法使用设备管理器更改 MTU。

使用以下命令启用巨型帧。

```
netsh interface ipv4 set subinterface "Ethernet" mtu=9001
```

使用以下命令禁用巨型帧。

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500
```

故障排除

如果在使用巨型帧时您的 EC2 实例和 Amazon Redshift 集群之间出现连接问题，则请参阅《Amazon Redshift 管理指南》中的查询似乎挂起，有时无法连接到集群。

适用于您的 EC2 实例的虚拟私有云

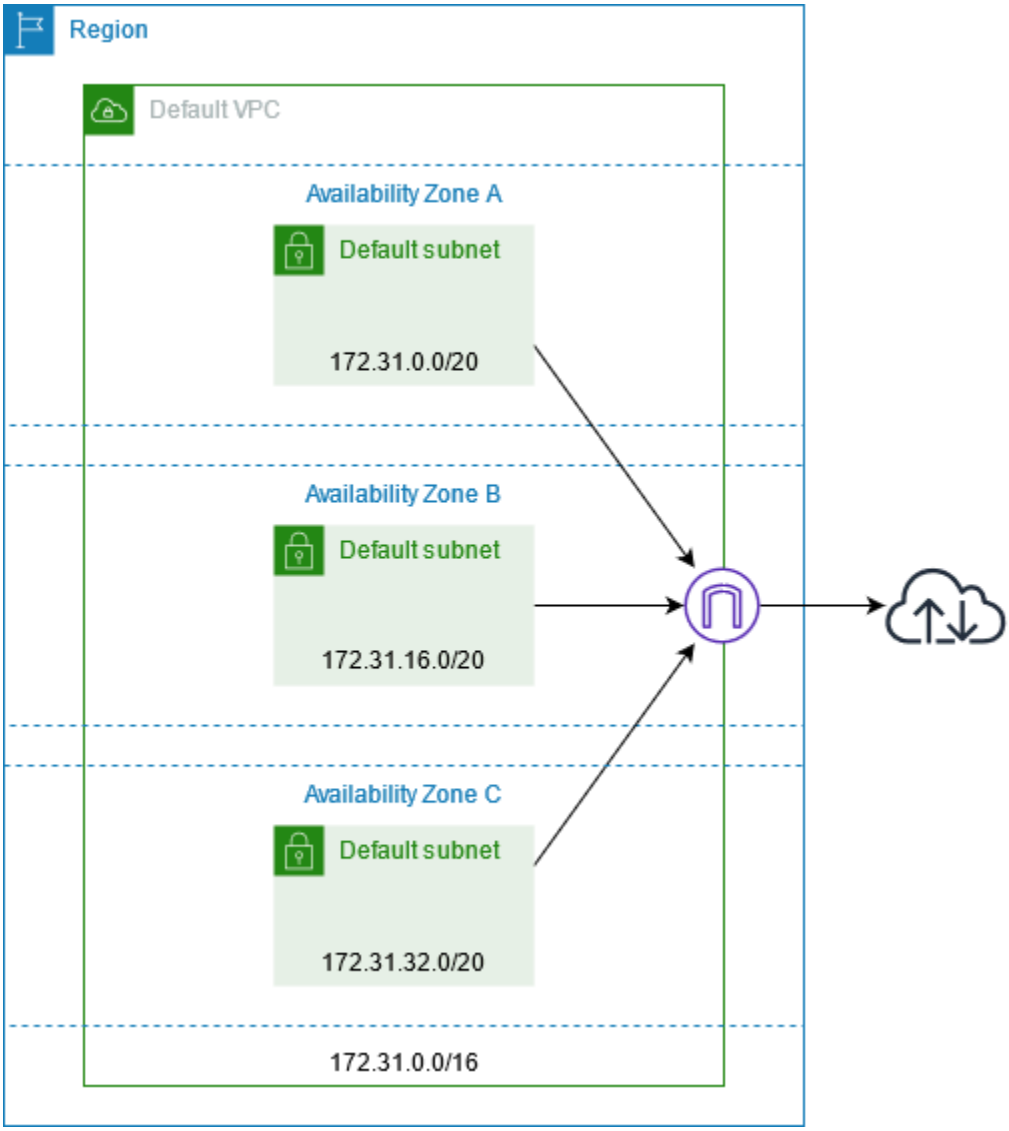
通过 Amazon Virtual Private Cloud (Amazon VPC)，您可以在 Amazon 云内您自己的逻辑隔离区域中定义虚拟网络，我们称之为 Virtual Private Cloud 或 VPC。您可将 Amazon 资源（如 Amazon EC2 实例）创建到 VPC 的子网中。您的 VPC 与您在自己的数据中心中运行的传统网络可能极为相似，同时享有使用来自 Amazon 的可扩展基础设施的优势。您可以配置您的 VPC；您可以选择它的 IP 地址范围、创建子网并配置路由表、网关和安全设置。现在您可以将您的 VPC 中的实例连接到 Internet 或您自己的数据中心。

内容

- 您的默认 VPC
- 非默认 VPC
- 互联网访问
- 共享子网
- 仅限 IPv6 子网

您的默认 VPC

在您创建 Amazon 账户时，我们会在每个区域中创建一个默认 VPC。默认 VPC 是已配置好可供您使用的 VPC。例如，在每个默认 VPC 中，每个可用区都有一个默认子网，一个连接到 VPC 的互联网网关，并且主路由表中有一条将所有流量 (0.0.0.0/0) 发送到互联网网关的路由。您可以按需修改默认 VPC 的配置。例如，您可以添加子网和路由表。



非默认 VPC

您可以创建自己的 VPC，而不是为您的资源使用默认 VPC，如《Amazon VPC 用户指南》的创建 VPC 中所述。

以下是为 EC2 实例创建 VPC 时要考虑的一些事项。

- 您可以使用 IPv4 CIDR 块的默认建议，也可以输入应用程序或网络所需的 CIDR 块。
- 为了确保高可用性，请在多个可用区中创建子网。
- 如果必须可从互联网访问您的实例，请执行以下操作之一：
  - 如果您的实例可以位于公共子网中，则请添加公共子网。保持两个 DNS 选项都处于启用状态。您可以选择立即或稍后添加私有子网。

- 如果您的实例必须位于私有子网中，则请仅添加私有子网。您可以添加 NAT 网关，为私有子网中的实例提供互联网访问。如果您的实例跨可用区发送或接收大量流量，则请在每个可用区创建一个 NAT 网关。否则，您可以仅在其中一个可用区中创建 NAT 网关，并在与 NAT 网关相同的可用区中启动发送或接收跨区域流量的实例。

互联网访问

在默认 VPC 中的默认子网中启动的实例可以访问互联网，因为默认 VPC 配置为分配公有 IP 地址和 DNS 主机名，并且主路由表配置了通往连接到 VPC 的互联网网关的路由。

对于在非默认子网和 VPC 中启动的实例，可以使用以下选项之一，以确保在这些子网中启动的实例可以访问互联网：

- 配置互联网网关。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用互联网网关连接到互联网。
- 配置公有 NAT 网关。有关更多信息，请参阅 Amazon VPC 用户指南中的从私有子网访问互联网。

共享子网

在共享 VPC 子网中启动 EC2 实例时，请注意以下几点：

- 参与者可以通过指定共享子网 ID 在共享子网中运行实例。参与者必须拥有其指定的任何网络接口。
- 参与者可以启动、停止、终止和描述其在共享子网中创建的实例。参与者无法启动、停止、终止或描述 VPC 所有者在共享子网中创建的实例。
- VPC 所有者无法启动、停止、终止或描述参与者在共享子网中创建的实例。
- 参与者可以使用 EC2 Instance Connect Endpoint 连接到共享子网中的实例。参与者必须在共享子网中创建 EC2 Instance Connect Endpoint。参与者不能使用 VPC 所有者在共享子网中创建的 EC2 Instance Connect Endpoint。

有关共享 Amazon EC2 资源的更多信息，请参阅以下资源：

- the section called “管理与组织或 OU 共享的 AMI”
- the section called “共享容量预留”
- the section called “共享置放群组”

有关共享子网的更多信息，请参阅 Amazon VPC 用户指南中的与其他账户共享 VPC。

仅限 IPv6 子网

在仅使用 IPv6 的子网中启动的 EC2 实例将会获得 IPv6 地址，但不会获得 IPv4 地址。您在仅使用 IPv6 的子网中启动的任何实例，必须是基于 Nitro 的实例。

辅助网络

辅助网络是专为特殊网络使用案例构建的虚拟网络。这些网络在 Amazon 云的各个分区内在逻辑上相互隔离。您可以在辅助网络中创建辅助子网之类的资源。辅助网络与 Amazon VPC 紧密关联，因此特定的实例会被设置为多网段连接，并同时部署在 VPC 和辅助网络中。

辅助网络目前可用于特定实例类型，并且需要通过长期承诺的容量预留方式来使用。如果您认为使用辅助网络可能对您的工作负载有所帮助，请联系您的客户团队获取更多信息。

内容

- 什么是辅助网络？
- 重要概念
- 架构
- 其他注意事项
- 开始使用
- 管理辅助网络资源
- 网络设计最佳实践
- 问题排查
- 限额和限制

什么是辅助网络？

辅助网络提供了一个与 VPC 网络结合使用的逻辑隔离网络，因此实例能够同时接入两个独立的网络。辅助网络的优势包括：

- 适用于特定使用案例和协议（如用于机器学习工作负载的东西向连接）的高性能网络解决方案
- 具有逻辑隔离功能的多租户支持
- 实例与 VPC 和 Amazon 服务无缝集成

重要概念

辅助网络

一种区域网络构造，它提供逻辑第 3 层网络，并采用 IPv4 CIDR 数据块（范围从 /28 到 /12）。辅助网络在物理分隔的网络基础设施上独立于 VPC 运行。

辅助子网

辅助网络中特定于某个可用区的构造，类似于 VPC 子网。辅助子网支持范围从 /28 到 /12 的 CIDR 数据块。

辅助接口

连接到辅助网卡的网络接口，用于在辅助子网内实现东西向连接。这些接口在物理和逻辑上与弹性网络接口（ENI）互相分隔。

架构

支持辅助网络的 EC2 实例是多网段的，这意味着它们能够同时在 VPC 和辅助网络之间进行通信：

- VPC：提供与 Amazon 服务、存储、数据库、网络服务和互联网的南北向 TCP/IP 连接
- 辅助网络：在支持的专用实例之间提供东西向连接

其他注意事项

- 辅助接口通过 RunInstances 进行管理，它们不能单独创建或删除。
- 实例启动后，辅助接口便无法再进行连接或分离。
- 辅助接口 IP 地址一旦启动便无法更改。
- 辅助网络不支持安全组、NACL、流日志等 VPC 功能。

开始使用

先决条件

在使用辅助网络启动实例之前，请确保您已在目标区域内配置您的 VPC，并在 EC2 容量的目标可用区中配置子网。

步骤 1：创建辅助网络

创建与 VPC 在同一区域的辅助网络。这是一种区域性资源，能够为您的 RDMA 流量提供逻辑隔离。

```
aws ec2 create-secondary-network \
  --network-type rdma \
  --ipv4-cidr-block 172.31.0.0/16 \
  --region us-east-2
```

参数：

- `--network-type`；网络类型（目前仅支持 rdma）
- `--ipv4-cidr-block`：介于 /28 和 /12 之间的 IPv4 CIDR 数据块
- `--region`：Amazon 区域（US-East-2）

> **Note**
>
> 最佳实践：选择与您的 VPC CIDR 不重叠的 CIDR 数据块，以便在实例层面简化路由操作。

步骤 2：创建辅助子网

在与您的 VPC 子网相同的可用区中创建辅助子网。这是特定于可用区的资源。

```
aws ec2 create-secondary-subnet \
  --secondary-network-id sn-1234567890abcdef0 \
  --ipv4-cidr-block 172.31.24.0/24 \
  --availability-zone us-east-2a
```

> **Note**
>
> IP 地址预留：与 VPC 子网一样，Amazon 为每个辅助子网预留了前 4 个 IP 地址和最后一个 IP 地址，供内部使用。

步骤 3：启动实例

在您的 VPC 子网和辅助子网中同时启动实例。该实例将具有多网连接功能，能够与两个网络实现连接。

```
aws ec2 run-instances \
  --image-id ami-12345678 \
  --count 1 \
  --instance-type <instance> \
  --key-name MyKeyPair \
  --instance-market-options '{"MarketType": "capacity-block"}' \
  --capacity-reservation-specification '{"CapacityReservationTarget": \
  {"CapacityReservationId": "cr-1234567890abcdef0"}}' \
  --network-interfaces \
    "NetworkCardIndex=0,DeviceIndex=0,Groups=sg-12345678,\
    SubnetId=subnet-0987654321fedcba0,InterfaceType=interface" \
  --secondary-interfaces \
    "NetworkCardIndex=1,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=2,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=3,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=4,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=5,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=6,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=7,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true", \
    "NetworkCardIndex=8,DeviceIndex=0,SecondarySubnetId=ss-98765421yxz,\
    InterfaceType=secondary,PrivateIpAddressCount=1,DeleteOnTermination=true"
```

关键参数：

- `--network-interfaces`：指定用于 VPC 连接的主 Nitro ENI（网卡索引 0）
- `--secondary-interfaces`：为辅助子网内的东西向连接指定 8 个辅助接口（网卡索引 1-8）
- `InterfaceType=secondary`：表示辅助接口

管理辅助网络资源

描述辅助网络

查看有关辅助网络的详细信息：

```
aws ec2 describe-secondary-networks \
  --secondary-network-id sn-1234567890abcdef0
```

描述辅助子网

查看有关辅助子网的详细信息：

```
aws ec2 describe-secondary-subnets \
  --secondary-subnet-id ss-98765421yxz
```

描述辅助接口

查看有关连接到您的实例的辅助网络接口的详细信息：

```
aws ec2 describe-secondary-interfaces \
  --filters "Name=attachment.instance-id,Values=i-1234567890abcdef0"
```

删除资源

删除辅助子网：

```
aws ec2 delete-secondary-subnet \
  --secondary-subnet-id ss-98765421yxz
```

删除辅助网络：

```
aws ec2 delete-secondary-network \
  --secondary-network-id sn-1234567890abcdef0
```

> **Important**
>
> 在删除辅助网络之前，您必须先终止所有实例并删除所有辅助子网。

网络设计最佳实践

CIDR 规划

避免 CIDR 重叠：尽管从物理层面上讲，辅助网络与 VPC 是相互隔离的，但使用不重叠的 CIDR 数据块可以简化实例操作系统层面的路由配置。

> **Note**
>
> Amazon 为每个子网预留 5 个 IP 地址。

流量分割

按辅助网络分割：为不同的项目、团队或安全边界创建单独的辅助网络。辅助网络实现了不同实例之间的逻辑隔离。实例无法在不同辅助网络之间进行通信。

使用多个子网：在辅助网络中，使用多个辅助子网按 GPU 索引、可用区或工作负载类型分割流量。例如，一种常见的架构模式是部署单个辅助网络，该网络包含 4 个或 8 个辅助子网，其中每个辅助子网都与一组具有相同索引的 GPU 相对应。

问题排查

实例启动失败

问题：使用辅助网络接口启动实例失败。

解决方案

- 验证您的 AMI 是否包含适当的驱动程序支持
- 确保您的辅助子网有足够的可用 IP 地址
- 确认您的容量预留处于“活动”状态
- 检查您的辅助子网是否与您的 VPC 子网位于相同的可用区中。

连接问题

问题：无法在各实例之间建立 RDMA 连接。

解决方案

- 确认所有实例都处于同一个辅助网络和辅助子网中
- 检查实例上是否正确加载了辅助接口驱动程序
- 确保您的应用程序绑定到正确的网络接口
- 同一辅助子网内的实例可通过直接路由进行访问。跨子网通信可通过由 DHCP 分配的静态路由实现。

API 错误

问题：辅助网络和辅助子网操作的 API 调用失败。

解决方案

- 请确认 ec2:CreateSecondaryNetwork、ec2:CreateSecondarySubnet 等的 IAM 权限
- 检查 CIDR 数据块是否在支持的范围内（/28 到 /12）
- 确认您在使用正确的区域和可用区

限额和限制

要申请提高配额，您可以使用 Amazon 服务配额或联系 Amazon Support。

辅助网络配额和限制

资源	限制	可调整
每个区域的辅助网络数	5	是
每个辅助网络的辅助子网数	200	是
CIDR 数据块大小	/28 至 /12	否

Amazon EC2 中的安全性

Amazon 十分重视云安全性。作为 Amazon 客户，您将从专为满足大多数安全敏感型组织的要求而打造的数据中心和网络架构中受益。

安全性是 Amazon 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全性：Amazon 负责保护在 Amazon 云中运行 Amazon 服务的基础结构。Amazon 还向您提供可安全使用的服务。作为Amazon合规性计划的一部分，第三方审计人员将定期测试和验证安全性的有效性。要了解适用于 Amazon EC2 的合规性计划，请参阅 Amazon 合规性计划范围内的的服务。
- 云中的安全性 – 您的责任包括以下各个方面：
  - 例如，通过配置 VPC 和安全组来控制对实例的网络访问。有关更多信息，请参阅控制网络流量。
  - 管理用于连接到您的实例的凭证。
  - 管理访客操作系统和部署到访客操作系统的软件，包括更新和安全补丁。有关更多信息，请参阅Amazon EC2 实例的更新管理。
  - 配置附加到实例的 IAM 角色以及与这些角色关联的权限。有关更多信息，请参阅适用于 Amazon EC2 的 IAM 角色。

该文档帮助您了解如何在使用 Amazon EC2 时应用责任共担模式。它说明了如何配置 Amazon EC2 以实现您的安全性和合规性目标。您还会了解如何使用其他Amazon服务以帮助您监控和保护 Amazon EC2 资源。

内容

- Amazon EC2 中的数据保护
- Amazon EC2 中的基础设施安全性
- Amazon EC2 中的恢复功能
- Amazon EC2 的合规性验证
- 适用于 Amazon EC2 的 Identity and Access Management
- Amazon EC2 实例的更新管理
- Windows 实例的安全最佳实践
- Amazon EC2 密钥对和 Amazon EC2 实例
- EC2 实例的 Amazon EC2 安全组
- 使用接口 VPC 端点访问 Amazon EC2

Amazon EC2 中的数据保护

Amazon 责任共担模式适用于 Amazon Elastic Compute Cloud 中的数据保护。如该模式中所述，Amazon 负责保护运行所有 Amazon Web Services 云 的全球基础结构。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 Amazon Web Services 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅Amazon 中国区域法律条款。

出于数据保护目的，建议您保护 Amazon Web Services 账户 凭证并使用 Amazon IAM Identity Center 或 Amazon Identity and Access Management（IAM）设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 使用 SSL/TLS 与 Amazon 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用 Amazon CloudTrail 设置 API 和用户活动日记账记录。有关使用 CloudTrail 跟踪来捕获 Amazon 活动的信息，请参阅《Amazon CloudTrail 用户指南》中的使用 CloudTrail 跟踪。
- 使用 Amazon 加密解决方案以及 Amazon Web Services 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在通过命令行界面或 API 访问 Amazon 时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://www.amazonaws.cn/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括当您通过控制台、API、Amazon CLI 或 Amazon SDK 使用 Amazon EC2 或其他 Amazon Web Services 服务 时。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

内容

- Amazon EBS 数据安全
- 静态加密
- 传输中加密

Amazon EBS 数据安全

Amazon EBS 卷作为未格式化的原始块设备呈现。这些设备是在 EBS 基础设施上创建的逻辑设备；Amazon EBS 服务可确保在客户使用或重复使用之前，这些设备为逻辑空白（即，原始数据块被归零或包含加密伪随机数据）。

如果您有要求在使用后和/或使用前使用特定方法擦除所有数据的程序，例如 DoD 5220.22-M（美国《国家工业安全计划操作手册》）或 NIST 800-88（《存储介质清理指南》）中详细说明的程序，您可以在 Amazon EBS 上执行此操作。该数据块级活动将反映到 Amazon EBS 服务的底层存储介质中。

静态加密

EBS 卷

Amazon EBS 加密是适用于 EBS 卷和快照的加密解决方案。它使用 Amazon KMS keys。有关更多信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 加密。

[Windows 实例] 您也可使用 Microsoft EFS 和 NTFS 权限来进行文件夹级和文件级加密。

实例存储卷

NVMe 实例存储卷上的数据是使用实例上的硬件模块中实施的 XTS-AES-256 密码加密的。用于加密写入本地连接的 NVMe 存储设备的数据的密钥是针对每个客户和每个卷的。密钥由硬件模块生成并且仅位于硬件模块内，Amazon 人员无法访问该模块。当实例停止或终止并且无法恢复时，将销毁所有加密密钥。无法禁用此加密，并且无法提供自己的加密密钥。

HDD 实例存储卷上的数据在 H1、D3 和 D3en 实例上使用 XTS-AES-256 和一次性密钥加密。

停止、休眠或终止实例时，实例存储卷中的每个存储数据块都会被重置。因此，无法通过另一实例的实例存储访问您的数据。

内存

以下实例上启用了内存加密：

- 使用 Amazon Graviton2 或更高版本 Amazon Graviton 处理器的实例支持始终在线内存加密。加密密钥是在主机系统内安全生成的，不要离开主机系统，并在重新启动或关闭主机电源时销毁的。有关更多信息，请参阅 Amazon Graviton 处理器。
- 采用第三代英特尔至强可扩展处理器（Ice Lake）的实例（如 M6i 实例）和采用第四代英特尔至强可扩展处理器（Sapphire Rapids）的实例（如 M7i 实例）。这些处理器支持使用英特尔总内存加密 (TME) 的永远在线内存加密。

- 采用第三代 AMD EPYC 处理器（Milan）的实例（如 M6a 实例）和采用第四代 AMD EPYC 处理器（Genoa）的实例（如 M7a 实例）。这些处理器支持使用 AMD 安全内存加密（SME）进行始终在线内存加密。
- 某些基于 AMD 的实例类型支持 AMD 安全加密虚拟化安全嵌套分页 (SEV-SNP)。有关更多信息，请参阅 查找支持 AMD SEV-SNP 的 EC2 实例类型。

传输中加密

物理层加密

所有在Amazon全球网络上跨Amazon区域流动的数据在离开Amazon保护的设施之前都会在物理层自动加密。可用区之间的所有流量均已加密。其他加密层（包括本节中列出的加密层）可能会提供额外保护。

Amazon VPC 对等连接和 Transit Gateway 跨区域对等连接提供的加密

使用 Amazon VPC 对等连接和 Transit Gateway 对等连接的所有跨区域流量在离开区域时都会自动批量加密。如本节之前所述，所有跨区域流量在离开Amazon保护的设施之前均会在物理层自动提供一层额外加密层。

实例之间的加密

Amazon 在所有类型的 EC2 实例之间提供安全的私有连接。此外，某些实例类型使用底层 Nitro 系统硬件的卸载功能，自动加密实例之间的传输中流量。此加密使用关联数据的身份验证加密（AEAD）算法，采用 256 位加密。这对网络性能没有影响。要在实例之间支持这种额外的传输中流量加密，必须满足以下要求：

- 使用以下实例类型：
  - 通用型：
    M5dn、M5n、M5zn、M6a、M6i、M6id、M6idn、M6in、M7a、M7g、M7gd、M7i、M7i-flex、M8a、M8azn、M8g、M8gb、M8gd、M8gn、M8i、M8id、M8i-flex、M8in、M8idn、M8ine、M8ib、M8idb、M9g、M9gd、Mac-m4、Mac-m4pro
  - 计算优化型：C5n、C6a、C6gn、C6i、C6id、C6in、C7a、C7g、C7gd、C7gn、C7i、C7i-flex、C8a、C8g、C8gb、C8gd、C8gn、C8i、C8id、C8i-flex、C8in、C8ine、C8ib
  - 内存优化型：R5dn、R5n、R6a、R6i、R6id、R6idn、R6in、R7a、R7g、R7gd、R7i、R7iz、R8a、R8g、R8gb flex、R8in、R8idn、R8ib、R8idb、U-3tb1、U-6tb1、U-9tb1、U-12tb1、U-18tb1、U-24tb1、U7i-6tb、U
  - 存储优化型：D3、D3en、I3en、I4g、I4i、I7i、I7ie、I8g、I8ge、Im4gn、Is4gen

  - 加速计算
    型：DL1、DL2q、F2、G4ad、G4dn、G5、G6、G6e、G6f、Gr6、Gr6f、G7e、Inf1、Inf2、P3dn、P4
    B200、P6-B300、P6e-GB200、Trn1、Trn1n、Trn2、Trn2u、VT1
  - 高性能计算：Hpc6a、Hpc6id、Hpc7a、Hpc7g、Hpc8a
- 这些实例位于同一区域。
- 这些实例位于相同 VPC 或对等的 VPC 中，并且流量不会通过虚拟网络设备或服务（如负载均衡器或中转网关）传输。

如本节之前所述，所有跨区域流量在离开Amazon保护的设施之前均会在物理层自动提供一层额外加密层。

使用 Amazon CLI 查看加密实例之间传输中流量的实例类型

使用以下 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
    --filters Name=network-info.encryption-in-transit-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

往返于 Amazon Outposts 的加密

Outpost 会创建名为 service links（服务链接）的特殊网络连接到其 Amazon 主区域，并且可以选择创建私有连接到您指定的 VPC 子网。通过这些连接的所有流量都完全加密。有关更多信息，请参阅 。通过服务链路进行连接和传输中加密中的Amazon Outposts用户指南。

远程访问加密

SSH 和 RDP 协议直接或通过 EC2 Instance Connect 提供了用于远程访问实例的安全通信通道。使用 Amazon Systems Manager Session Manage 或 Run Command 对实例的远程访问是使用 TLS 1.2 加密的，创建连接的请求是使用 SigV4 签名的，且由 Amazon Identity and Access Management 进行身份验证并授权。

使用传输层安全性协议（TLS）等加密协议加密在客户端和 Amazon EC2 实例之间传输的敏感数据是您的责任。

（Windows 实例）确保仅允许 EC2 实例和 Amazon API 端点或其他敏感远程网络服务之间的加密连接。您可以通过出站安全组或 Windows 防火墙规则强制实施这一点。

Amazon EC2 中的基础设施安全性

作为一项托管式服务，Amazon Elastic Compute Cloud 受 Amazon 全球网络安全保护。有关 Amazon 安全服务以及 Amazon 如何保护基础设施的信息，请参阅 Amazon 云安全性。要按照基础设施安全最佳实践设计您的 Amazon 环境，请参阅《安全性支柱 Amazon Well-Architected Framework》中的基础设施保护。

您可以使用Amazon发布的 API 调用通过网络访问 Amazon EC2。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

有关更多信息，请参阅《安全支柱：Amazon Well-Architected 框架》中的基础设施保护。

网络隔离

虚拟私有云（VPC）是 Amazon 云内您自己的逻辑隔离区域中的虚拟网络。可以使用单独的 VPC 按工作负载或组织实体隔离基础设施。

子网是 VPC 中的 IP 地址范围。在启动实例时，您可以在 VPC 上的子网中启动该实例。可以使用子网隔离单个 VPC 中的应用程序层（例如，Web、应用程序和数据库）。如果不应直接从 Internet 访问实例，请使用私有子网访问。

要使用私有 IP 地址从 VPC 调用 Amazon EC2 API，请使用 Amazon PrivateLink。有关更多信息，请参阅 使用接口 VPC 端点访问 Amazon EC2。

物理主机上的隔离

同一物理主机上的不同 EC2 实例彼此隔离，就好像它们位于不同的物理主机上一样。虚拟机监控程序隔离 CPU 和内存，并为实例提供虚拟化磁盘，而不是访问原始磁盘设备。

在停止或终止实例时，虚拟机监控程序将清理分配给实例的内存（设置为零），然后再将内存分配给新实例并重置每个存储块。这会确保不会意外向另一个实例泄露数据。

网络 MAC 地址由 Amazon 网络基础设施动态分配给实例。IP 地址由 Amazon 网络基础设施动态分配给实例，或者由 EC2 管理员通过经过身份验证的 API 请求进行分配。Amazon 网络允许实例仅从分配给它们的 MAC 和 IP 地址发送流量。否则，将会丢弃流量。

默认情况下，实例无法接收未明确将其指定为目标地址的流量。如果需要在实例上运行网络地址转换（NATI）、路由或防火墙服务，您可以为网络接口禁用源/目标检查。

控制网络流量

请考虑使用以下方法来控制到 EC2 实例的网络流量：

- 使用安全组限制实例访问。配置允许所需最小网络流量的规则。例如，您可以仅允许来自公司网络地址范围的流量或仅允许特定协议（如 HTTPS）的流量。对于 Windows 实例，允许 Windows 管理流量和最少出站连接。
- 使用安全组作为用于控制对 Amazon EC2 实例的网络访问的主要机制。必要时，尽量少使用网络 ACL，以提供无状态的粗略网络控制。安全组比网络 ACL 更为通用，因为它们能够执行有状态数据包筛选和创建引用其他安全组的规则。不过，网络 ACL 作为一项辅助控制，可以有效地拒绝特定流量子集或提供高级子网护栏。此外，由于网络 ACL 将应用于整个子网，因此，可以将它用作深度防御措施，以防实例在没有正确安全组的情况下意外启动。
- [Windows 实例] 使用组策略对象（GPO）集中管理 Windows 防火墙设置，以便进一步增强网络控制。通常，客户将使用 Windows 防火墙来进一步查看网络流量并对安全组筛选器进行补充，创建高级规则来阻止特定应用程序访问网络或筛选来自子集 IP 地址的流量。例如，Windows 防火墙可以将对 EC2 元数据服务 IP 地址的访问限制为特定的用户或应用程序。或者，面向公众的服务可能使用安全组来限制特定端口的流量，而 Windows 防火墙则维护显式阻止的 IP 地址的列表。
- 如果不应直接从 Internet 访问实例，请使用私有子网访问。使用堡垒主机或 NAT 网关从私有子网中的实例进行 Internet 访问。
- [Windows 实例] 使用安全管理协议，如通过 SSL/TLS 进行的 RDP 封装。Remote Desktop Gateway Quick Start 提供了有关部署远程桌面网关（包括将 RDP 配置为使用 SSL/TLS）的最佳实践。
- [Windows 实例] 使用 Active Directory 或 Amazon Directory Service 严格集中地控制和监控对 Windows 实例的交互式用户和组访问，并避免本地用户权限。此外，请避免使用域管理员，而是创建更精细的、应用程序特定的基于角色的账户。利用 Just Enough Administration (JEA)，可以在没有交互式访问或管理员访问的情况下管理对 Windows 实例进行的更改。此外，JEA 使组织能够锁定对实例管理所需的 Windows PowerShell 命令子集的管理访问权限。有关更多信息，请参阅Amazon 安全性最佳实践白皮书中关于“管理 Amazon EC2 的操作系统级访问权限”的部分。
- [Windows 实例] 系统管理员应使用带有限访问权限的 Windows 账户执行日常活动，并且仅在有必要执行特定配置更改时提升访问权限。此外，仅在绝对必要时直接访问 Windows 实例。相反，利用中央配置管理系统（例如 EC2 Run Command、Systems Center Configuration Manager (SCCM)、Windows PowerShell DSC 或 Amazon EC2 Systems Manager (SSM)）将更改推送到 Windows 服务器。

- 使用所需的最少网络路由配置 Amazon VPC 子网路由表。例如，仅将需要互联网的直接访问权限的 Amazon EC2 实例放置到带互联网网关路由的子网中，并且仅将需要内部网络的直接访问权限的 Amazon EC2 实例放置到带虚拟私有网关路由的子网中。
- 考虑使用其他安全组或网络接口，独立于常规应用程序流量来控制和审计 Amazon EC2 实例管理流量。此方法可让客户实施用于更改控制的特殊 IAM 策略，从而更轻松地审计对安全组规则或自动规则验证脚本进行的更改。使用多个网络接口还提供了额外的网络流量控制选择，包括能够创建基于主机的路由策略或根据网络接口分配的子网利用不同的 VPC 子网路由规则。
- 使用 Amazon Virtual Private Network 或 Amazon Direct Connect 建立从远程网络到 VPC 的私有连接。有关更多信息，请参阅网络到 Amazon VPC 的连接选项。
- 使用 VPC Flow Logs 监控到达实例的流量。
- 使用 GuardDuty 恶意软件防护来识别实例上表明恶意软件的可疑行为，这些行为可能会损害您的工作负载、重新利用资源用于恶意使用以及未经授权访问您的数据。
- 使用 GuardDuty 运行时监控来识别和应对您的实例面临的潜在威胁。有关更多信息，请参阅 How Runtime Monitoring works with Amazon EC2 instances（运行时监控如何与 Amazon EC2 实例协同工作）。
- 使用 Amazon Security Hub CSPM、可访问性分析器或网络访问分析器检查来自实例的意外网络访问。
- 使用 EC2 Instance Connect 通过 Secure Shell（SSH）连接到实例，而无需共享和管理 SSH 密钥。
- 使用 Amazon Systems Manager Session Manager 远程访问实例，而不是打开入站 SSH 或 RDP 端口并管理密钥对。
- 使用 Amazon Systems Manager Run Command 自动执行常见的管理任务，而不是连接到实例。
- [Windows 实例] 许多 Windows 操作系统角色和 Microsoft 业务应用程序还提供了增强功能，例如 IIS 中的 IP 地址范围限制、Microsoft SQL Server 中的 TCP/IP 筛选策略以及 Microsoft Exchange 中的连接筛选策略。应用层中的网络限制功能可以为关键业务应用程序服务器提供额外的防御层。

Amazon VPC 支持其他网络安全控制，例如网关、代理服务器和网络监控选项。有关更多信息，请参阅《Amazon VPC 用户指南》中的控制网络流量。

Amazon EC2 中的恢复功能

Amazon 全球基础设施围绕 Amazon 区域和可用区构建。区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作

在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

如果需要跨更大的地理距离复制数据或应用程序，请使用 AmazonLocal Zones。Amazon 本地扩展区是在地理上靠近您的用户的 Amazon 区域的扩展。Local Zones 有自己的 Internet 连接并支持。Amazon Direct Connect与所有 Amazon 区域类似，Amazon Local Zones 与其他 Amazon区域完全隔离。

如果您需要在 Amazon 本地扩展区中复制数据或应用程序，Amazon 建议您使用以下区域作为失效转移区域：

- 另一个本地扩展区
- 区域中不是父区域的可用区。您可以使用 describe-availability-zones 命令查看父区域。

有关Amazon区域和可用区的更多信息，请参阅Amazon全球基础设施。

除了 Amazon 全球基础设施以外，Amazon EC2 还提供以下功能以支持数据恢复：

- 跨区域复制 AMI
- 跨区域复制 EBS 快照
- 使用 Amazon Data Lifecycle Manager 自动化 EBS 支持的 AMI
- 使用 Amazon Data Lifecycle Manager 自动处理 EBS 快照
- 使用 Amazon EC2 Auto Scaling 保持队列的运行状况和可用性
- 使用 Elastic Load Balancing 在一个或多个可用区中的多个实例之间分配传入流量

Amazon EC2 的合规性验证

要了解某个 Amazon Web Services 服务是否在特定合规性计划范围内，请参阅合规性计划范围内的 Amazon Web Services 服务，然后选择您感兴趣的合规性计划。有关常规信息，请参阅 、、Amazon Web Services 合规性计划。

您可以使用 Amazon Artifact 下载第三方审计报告。有关更多信息，请参阅在 Amazon Artifact 中下载报告。

您在使用 Amazon Web Services 服务 时的合规性责任由您的数据的敏感性、您公司的合规性目标以及适用的法律法规决定。有关您在使用 Amazon Web Services 服务 时的合规责任的更多信息，请参阅 Amazon 安全性文档。

适用于 Amazon EC2 的 Identity and Access Management

Amazon Identity and Access Management（IAM）是一项，Amazon Web Services 服务可以帮助管理员安全地控制对 Amazon 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（具有权限）来使用 Amazon EC2 资源。IAM 是一项无需额外费用即可使用的。Amazon Web Services 服务

您的安全凭证使 Amazon 中的服务可以识别您，并授予您对 Amazon 资源（例如您 Amazon EC2 资源）的使用权限。您可以使用 Amazon EC2 和 IAM 的功能，在不共享您安全凭证情况下允许其他用户、服务和应用程序使用您 Amazon EC2 资源。您可以使用 IAM 控制其他用户对您 Amazon Web Services 账户 中资源的使用方式，并且您可以使用安全组来控制对您 Amazon EC2 实例的访问。您可以选择允许 Amazon EC2 资源的完全使用或有限使用权限。

如果您是开发人员，则可以使用 IAM 角色来管理在 EC2 实例上运行的应用程序所需的安全凭证。将 IAM 角色附加到实例后，在该实例上运行的应用程序可以从实例元数据服务（IMDS）检索凭证。

有关使用 IAM 保护您 Amazon 资源的最佳实践，请参阅《IAM 用户指南》中的 IAM 的安全防御最佳实操。

内容

- Amazon EC2 基于身份的策略
- 用于控制访问 Amazon EC2 API 的示例策略
- 用于控制对 Amazon EC2 控制台的访问的示例策略
- Amazon EC2 的 Amazon 托管式策略
- 适用于 Amazon EC2 的 IAM 角色

Amazon EC2 基于身份的策略

默认情况下，用户没有创建或修改 Amazon EC2 资源或使用 Amazon EC2 API、Amazon EC2 控制台或 CLI 执行任务的权限。要允许用户创建或修改资源和执行任务，您必须创建 IAM 策略以允许用户使用所需的特定资源和 API 操作，然后将这些策略与需要这些权限的用户、组或 IAM 角色关联起来。

在将策略附加到一个用户、一组用户或角色时，它会授权或拒绝用户使用指定资源执行指定任务。有关 IAM 策略的更多一般信息，请参阅《IAM 用户指南》中的 IAM 中的策略与权限。有关管理和创建 IAM 策略的更多信息，请参阅管理 IAM 策略。

IAM 策略必须授予或拒绝使用一个或多个 Amazon EC2 操作的权限。它还必须指定可以用于操作的资源 (可以是所有资源，在某些情况下可以是特定资源)。策略还可以包含应用于资源的条件。

首先，您可以检查 Amazon EC2 的 Amazon 托管式策略是否满足您的需求。若不满足，您可以创建自己的自定义策略。有关更多信息，请参阅 the section called “Amazon 托管策略”。

内容

- 策略语法
- Amazon EC2 操作
- Amazon EC2 API 操作支持的资源级权限
- 适用于 Amazon EC2 的 Amazon 资源名称（ARN）
- Amazon EC2 的条件键
- 使用基于属性的访问控制访问权限
- 向用户、组和角色授予权限
- 检查用户是否具有所需权限

策略语法

IAM 策略是包含一个或多个语句的 JSON 文档。每个语句的结构如下。

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
  ]
}
```

组成语句的各个元素如下：

- Effect：此 effect 可以是 `Allow` 或 `Deny`。默认情况下 用户没有使用资源和 API 操作的权限，因此，所有请求均会被拒绝。显式允许将覆盖默认规则。显式拒绝将覆盖任何允许。
- Action：action 是对其授予或拒绝权限的特定 API 操作。要了解有关指定 action 的信息，请参阅 Amazon EC2 操作。

- Resource：受操作影响的资源。有些 Amazon EC2 API 操作允许您在策略中包括该操作可以创建或修改的特定资源。您可使用 Amazon 资源名称（ARN）来指定资源，或使用通配符（*）以指明该语句适用于所有资源。有关更多信息，请参阅 Amazon EC2 API 操作支持的资源级权限。
- Condition：条件是可选的。它们可以用于控制策略生效的时间。想要了解更多有关为 Amazon EC2 指定条件的信息，请参阅 Amazon EC2 的条件键。

有关策略要求的更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素参考。有关适用于 Amazon EC2 的 IAM 策略语句示例，请参阅 用于控制访问 Amazon EC2 API 的示例策略。

Amazon EC2 操作

在 IAM 策略语句中，您可以从支持 IAM 的任何服务中指定任何 API 操作。对于 Amazon EC2，请使用以下前缀为 API 操作命名：`ec2:`。例如：`ec2:RunInstances` 和 `ec2:CreateImage`。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": ["ec2:action1", "ec2:action2"]
```

您也可以使用通配符指定多项操作。例如，您可以指定名称以单词“Describe”开头的所有操作，如下所示：

```
"Action": "ec2:Describe*"
```

> **Note**
>
> 当前，Amazon EC2 描述 * API 操作都不支持资源级权限。有关 Amazon EC2 资源级权限更多信息，请参阅 Amazon EC2 基于身份的策略。

要指定所有 Amazon EC2 API 操作，请使用 * 通配符，如下所示：

```
"Action": "ec2:*"
```

有关 Amazon EC2 操作的列表，请参阅 Amazon EC2 服务授权参考中定义的操作。

Amazon EC2 API 操作支持的资源级权限

资源级权限指的是能够指定允许用户对哪些资源执行操作的能力。Amazon EC2 部分支持资源级权限。这意味着对于某些 Amazon EC2 操作，您可以控制何时允许用户执行操作 (基于必须满足的条件)

或是允许用户使用的特定资源。例如，您可以向用户授予启动实例的权限，但是仅限特定类型的实例，并且只能使用特定的 AMI。

要在 IAM 策略语句中指定资源，请使用其 Amazon 资源名称（ARN）。有关指定 ARN 值的详细信息，请参阅 适用于 Amazon EC2 的 Amazon 资源名称（ARN）。如果 API 操作不支持单个 ARN，则必须使用通配符 (*) 来指定所有资源都可能受到操作的影响。

要查看标识哪些 Amazon EC2 API 操作支持资源级权限的表，以及您可以在策略中使用的 ARN 和条件键，请参阅 Amazon EC2 的操作、资源和条件键。

请记住，您可以在用于 Amazon EC2 API 操作的 IAM 策略中应用基于标签的资源级权限。这可让您更好地控制用户可创建、修改或使用哪些资源。有关更多信息，请参阅 在创建过程中授予标记 Amazon EC2 资源的权限。

适用于 Amazon EC2 的 Amazon 资源名称（ARN）

每个 IAM 策略语句适用于您使用资源的 ARN 指定的资源。

ARN 的一般语法如下：

```
arn:aws:[service]:[region]:[account-id]:resourceType/resourcePath
```

service

服务 (例如，ec2)。

region

资源所在区域（例如，us-east-1）。

account-id

Amazon 账户 ID，不包含连字符（例如，123456789012）。

resourceType

资源类型 (例如，instance)。

resourcePath

识别资源的路径。您可以在路径中使用 * 通配符。

例如，您可以使用特定实例 (i-1234567890abcdef0) 的 ARN 在语句中指定它，如下所示。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

您可以使用 * 通配符指定属于特定账户的所有实例，如下所示。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

还可以使用 * 通配符指定属于特定账户的所有 Amazon EC2 资源，如下所示。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:*"
```

要指定所有资源，或者如果特定 API 操作不支持 ARN，请在 `Resource` 元素中使用 * 通配符，如下所示。

```
"Resource": "*"
```

许多 Amazon EC2 API 操作涉及多种资源。例如，`AttachVolume` 将一个 Amazon EBS 卷附加到一个实例，从而使用户必须获得相应权限才能使用该卷和该实例。要在单个语句中指定多个资源，请使用逗号分隔其 ARN，如下所示。

```
"Resource": ["arn1", "arn2"]
```

有关 Amazon EC2 资源的 ARN 列表，请参阅 Amazon EC2 定义的资源类型。

Amazon EC2 的条件键

在策略语句中，您可以选择性指定控制策略生效时间的条件。每个条件都包含一个或多个键值对。条件键不区分大小写。我们已经定义了 Amazon 全局条件键以及其他特定于服务的条件键。

有关 Amazon EC2 的服务特定条件键的列表，请参阅 Amazon EC2 的条件键。Amazon EC2 还实施了 Amazon 全局条件键。有关更多信息，请参阅 IAM 用户指南中的在所有请求中可用的信息。

所有 Amazon EC2 操作都支持 `aws:RequestedRegion` 和 `ec2:Region` 条件键。有关更多信息，请参阅 示例：限制对特定区域的访问权限。

要在 IAM 策略中使用条件键，请使用 `Condition` 语句。例如，以下策略授予用户添加和删除任何安全组的入站和出站规则的权限。它使用 `ec2:Vpc` 条件键来指定只能对特定 VPC 中的安全组执行这些操作。

JSON

```
{
"Version":"2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action": [
       "ec2:AuthorizeSecurityGroupIngress",
       "ec2:AuthorizeSecurityGroupEgress",
       "ec2:RevokeSecurityGroupIngress",
       "ec2:RevokeSecurityGroupEgress"],
     "Resource": "arn:aws:ec2:us-east-1:111122223333:security-group/*",
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": "arn:aws:ec2:us-east-1:111122223333:vpc/
vpc-11223344556677889"
        }
      }
    }
  ]
}
```

如果您指定了多个条件或在单一条件中指定了多个密钥，我们将通过逻辑 AND 操作对其进行评估。如果您在单一条件中指定了一个具有多个值的密钥，我们将通过逻辑 OR 操作对其进行评估。必须满足所有条件才能授予权限。

在指定条件时，您也可使用占位符。有关更多信息，请参阅 IAM 用户指南 中的 IAM 策略元素：变量和标签。

> **Important**
>
> 许多条件键是特定于某个资源的，而某些 API 操作会使用多个资源。如果您使用条件键编写策略，请使用语句的 `Resource` 元素指定要应用该条件键的资源。否则，该策略可能会完全阻止用户执行操作，因为针对未应用条件键的资源的条件检查失败。如果您不想指定资源，或者如果您已将策略的 `Action` 元素编写为包含多个 API 操作，则必须使用 `...IfExists` 条件类型以确保对不使用条件键的资源忽略条件键。有关更多信息，请参阅《IAM 用户指南》中的 ...IfExists 条件。

条件键

- ec2:Attribute 条件键
- ec2:ResourceID 条件键
- ec2:SourceInstanceARN 条件键

ec2:Attribute 条件键

`ec2:Attribute` 条件键可用于按资源的属性筛选访问权限的条件。

此条件键仅支持原始数据类型（如字符串或整数）的属性，或者仅包含 Value 属性的复杂 AttributeValue 对象（如 ModifyImageAttribute API 操作的 Description 或 ImdsSupport 对象）。条件键不能用于包含多个属性的复杂对象，例如 ModifyImageAttribute 的 LaunchPermission 对象。

例如，以下策略使用 `ec2:Attribute/Description` 条件键按 ModifyImageAttribute API 操作的复杂 Description（描述）对象筛选访问权限。条件键仅允许将图像描述修改为 `Production` 或 `Development` 的请求。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute/Description": [
            "Production",
            "Development"
          ]
        }
      }
    }
  ]
}
```

以下示例策略使用 ec2:Attribute 条件键按 ModifyImageAttribute API 操作的主要 Attribute（属性）属性筛选访问权限。条件键拒绝尝试修改图像描述的所有请求。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:ModifyImageAttribute",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
          "ec2:Attribute": "Description"
        }
      }
    }
  ]
}
```

ec2:ResourceID 条件键

通过指定的 API 操作使用以下 ec2:*Resource*ID 条件键时，条件键值用于指定 API 操作创建的结果资源。ec2:*Resource*ID 条件键不能用于指定 API 请求中特定的源资源。如果您通过指定 API 使用以下其中一个 ec2:*Resource*ID 条件键，则必须始终指定通配符 (*)。如果您指定不同的值，则在运行期间条件始终解析为 *。例如，要通过 CopyImage API 使用 ec2:ImageId 条件键，则必须按如下方式指定条件键：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
      "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
      "Condition": {
        "StringEquals": {
```

```
          "ec2:ImageID": "*"
        }
      }
    }
  ]
}
```

我们建议您避免将以下条件键与以下 API 操作一起使用：

- ec2:DhcpOptionsID – CreateDhcpOptions
- ec2:ImageID – CopyImage、CreateImage、ImportImage 和 RegisterImage
- ec2:InstanceID – RunInstances 和 ImportInstance
- ec2:InternetGatewayID – CreateInternetGateway
- ec2:NetworkAclID – CreateNetworkAcl
- ec2:NetworkInterfaceID – CreateNetworkInterface
- ec2:PlacementGroupName – CreatePlacementGroup
- ec2:RouteTableID – CreateRouteTable
- ec2:SecurityGroupID – CreateSecurityGroup
- ec2:SnapshotID – CopySnapshot、CreateSnapshot、CreateSnapshots 和 ImportSnapshots
- ec2:SubnetID – CreateSubnet
- ec2:VolumeID – CreateVolume 和 ImportVolume
- ec2:VpcID – CreateVpc
- ec2:VpcPeeringConnectionID – CreateVpcPeeringConnection

要根据特定的资源 ID 筛选访问权限，我们建议您使用 Resource 策略元素，如下所示。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CopyImage",
```

```
      "Resource": "arn:aws:ec2:us-east-1::image/ami-01234567890abcdef"
    }
  ]
}
```

ec2:SourceInstanceARN 条件键

使用 ec2:SourceInstanceARN 指定从中发出请求的实例的 ARN。这是一个 Amazon 全局条件密钥，这意味着您可以将其用于 Amazon EC2 以外的服务。有关策略示例，请参阅示例：允许特定实例查看其他 Amazon 服务中的资源。

使用基于属性的访问控制访问权限

在创建向用户授予使用 EC2 资源的权限的 IAM 策略时，可以在该策略的 Condition 元素中包含标签信息，以根据标签控制访问权限。这称为基于属性的访问权限控制（ABAC）。ABAC 可以让您更好地控制用户可以修改、使用或删除哪些资源。有关更多信息，请参阅什么是适用于 Amazon 的 ABAC？

例如，您可以创建一个策略，允许用户终止实例，但在实例具有 environment=production 标签时拒绝此操作。为此，您可以使用 aws:ResourceTag 条件键来基于附加到资源的标签允许或拒绝对资源的访问。

```
"StringEquals": { "aws:ResourceTag/environment": "production" }
```

要了解 Amazon EC2 API 操作是否支持使用 aws:ResourceTag 条件键控制访问，请参阅 Amazon EC2 的操作、资源和条件建。请注意，Describe 操作不支持资源级权限，因此，您必须在不带条件的单独语句中指定它们。

有关示例 IAM policies，请参阅 用于控制访问 Amazon EC2 API 的示例策略。

如果您基于标签允许或拒绝用户访问资源，则必须考虑显式拒绝用户对相同资源添加或删除这些标签的能力。否则，用户可能通过修改资源标签来绕过您的限制并获得资源访问权限。

向用户、组和角色授予权限

要提供访问权限，请为您的用户、组或角色添加权限：

- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。

- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

检查用户是否具有所需权限

在您创建 IAM 策略后，建议您检查它是否允许用户使用策略生效前所需的特定 API 操作和资源。

首先，创建一个用于测试目的的用户，然后将您创建的 IAM 策略与该测试用户关联起来。然后，以测试用户身份提出请求。

如果您测试的 Amazon EC2 操作创建或修改了一种资源，您在提交请求时应该使用 `DryRun` 参数（或运行带有 `--dry-run` 选项的 Amazon CLI 命令）。在这种情况下，调用会完成身份验证检查，但是不会完成该操作。例如，您可以检查用户能否终止特定实例，但不会真的终止它。如果测试用户具有所需的权限，请求会返回 `DryRunOperation`；否则，它会返回 `UnauthorizedOperation`。

如果策略未授予用户您所期望的权限，您可以根据需要调节策略并重新测试，直到您获得预期的结果。

> **Important**
>
> 在其生效之前，它需要几分钟时间将策略更改为适合状态。因此，我们建议您在测试策略更新前，等候五分钟的时间。

如果身份验证检查失败，该请求将返回一个带有诊断信息的代码消息。您可以使用 `DecodeAuthorizationMessage` 操作对消息进行解码。有关更多信息，请参阅《Amazon Security Token Service API Reference》中的 DecodeAuthorizationMessage，以及 decode-authorization-message。

用于控制访问 Amazon EC2 API 的示例策略

您可以使用 IAM 策略向用户授予使用 Amazon EC2 所需的权限。有关分步说明，请参阅《IAM 用户指南》中的创建 IAM 策略。

以下示例显示了您可用于向用户授予使用 Amazon EC2 的权限的策略语句。这些策略设计用于采用 Amazon CLI 或 Amazon SDK 发出的请求。在以下示例中，将每个 *`user input placeholder`* 替换为您自己的信息。

示例

- 示例：只读访问权限
- 示例：限制对特定区域的访问权限
- 使用实例
- 启动实例 (RunInstances)
- 使用竞价型实例
- 示例：使用预留实例
- 示例：标记资源
- 示例：使用 IAM 角色
- 示例：使用路由表
- 示例：允许特定实例查看其他 Amazon 服务中的资源
- 示例：使用启动模板
- 使用实例元数据
- 使用 Amazon EBS 卷和快照

有关用于 Amazon EC2 控制台的策略示例，请参阅 用于控制对 Amazon EC2 控制台的访问的示例策略。

示例：只读访问权限

以下策略为用户授予使用名称以 `Describe` 开头的所有 Amazon EC2 API 操作的权限。`Resource` 元素使用通配符表示用户可以通过这些 API 操作指定所有资源。在 API 操作不支持资源级权限的情况下，也需要 * 通配符。有关哪些 ARN 可用于哪些 Amazon EC2 API 操作的更多信息，请参阅 Amazon EC2 的操作、资源和条件键。

用户无权对资源执行任何操作 (除非其他语句为用户授予执行此操作的权限)，因为在默认情况下会对用户拒绝使用 API 操作的权限。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
   ]
}
```

示例：限制对特定区域的访问权限

以下策略拒绝用户使用所有 Amazon EC2 API 操作的权限，除非区域为欧洲地区（法兰克福）。该区域使用全局条件键 `aws:RequestedRegion`，所有 Amazon EC2 API 操作均支持此条件键。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": "eu-central-1"
        }
      }
    }
  ]
}
```

或者，您也可以使用条件键 `ec2:Region`，此条件键是 Amazon EC2 特定的，所有 Amazon EC2 API 操作均支持它。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
      {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "ec2:Region": "eu-central-1"
        }
      }
    }
  ]
}
```

使用实例

示例

- 示例：描述、启动、停止和终止所有实例
- 示例：描述所有实例，以及仅停止、启动和终止特定实例

示例：描述、启动、停止和终止所有实例

以下策略为用户授予使用 `Action` 元素中指定的 API 操作的权限。`Resource` 元素使用 * 通配符表示用户可以通过这些 API 操作指定所有资源。在 API 操作不支持资源级权限的情况下，也需要 * 通配符。有关哪些 ARN 可用于哪些 Amazon EC2 API 操作的更多信息，请参阅 Amazon EC2 的操作、资源和条件键。

用户无权使用任何其他 API 操作 (除非其他语句允许用户执行此操作)，因为用户在默认情况下没有使用 API 操作的权限。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeImages",
```

```
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": "*"
    }
   ]
}
```

示例：描述所有实例，以及仅停止、启动和终止特定实例

以下策略允许用户描述所有实例，但只能启动和停止实例 i-1234567890abcdef0 和 i-0598c7d356eba48d7，且只能终止在 us-east-1 区域中具有“purpose=test”资源标签的实例。

第一条语句为 Resource 元素使用 * 通配符以指示用户可以在操作中指定所有资源；在本例中，用户可以列出所有实例。在 API 操作不支持资源级权限的情况下 (在此情况下，为 ec2:DescribeInstances)，也需要 * 通配符。有关哪些 ARN 可用于哪些 Amazon EC2 API 操作的更多信息，请参阅 Amazon EC2 的操作、资源和条件键。

第二条语句为 StopInstances 和 StartInstances 操作使用资源级权限。特定实例在 Resource 元素中通过其 ARN 进行指示。

第三条语句允许用户终止在 us-east-1 区域中属于指定 Amazon 账户的所有实例（但仅在实例具有 "purpose=test" 标签的情况下）。当策略语句生效时，Condition 元素具备资格。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
   },
   {
      "Effect": "Allow",
```

```
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:111122223333:instance/i-0598c7d356eba48d7"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/purpose": "test"
         }
      }
   }

   ]
}
```

启动实例 (RunInstances)

RunInstances API 操作可启动一个或多个按需实例或一个或多个 Spot 实例。RunInstances 需要 AMI 并创建实例。用户可以在请求中指定键对和安全组。启动到 VPC 中需要子网，会创建网络接口。从 Amazon EBS-backed AMI 启动将创建卷。因此，用户必须具有使用这些 Amazon EC2 资源的权限。您可以创建要求用户对 RunInstances 指定可选参数或限制用户针对某个参数使用特定值的策略语句。

有关启动实例所需的资源级权限的更多信息，请参阅 Amazon EC2 的操作、资源和条件键。

默认情况下，用户没有描述、启动、停止或终止生成的实例的权限。授予用户管理所生成实例的权限的一种方法是：为每个实例创建一个特定标签，然后创建一个允许用户使用该标签管理实例的语句。有关更多信息，请参阅 使用实例。

资源

- AMI
- 实例类型

- 子网
- EBS 卷
- 标签
- 启动模板中的标签
- Elastic GPUs
- 启动模板

AMI

以下策略仅允许用户使用指定的 AMI、ami-9e1670f7 和 ami-45cf5c3c 启动实例。用户无法使用其他 AMI 启动实例（除非其他语句允许用户执行此操作）。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:us-east-1::image/ami-9e1670f7",
        "arn:aws:ec2:us-east-1::image/ami-45cf5c3c",
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
        "arn:aws:ec2:us-east-1:111122223333:security-group/*",
        "arn:aws:ec2:us-east-1:111122223333:subnet/*",
        "arn:aws:ec2:us-east-1:111122223333:network-interface/*"
      ]
    }
   ]
}
```

另外，以下策略允许用户从 Amazon 或某些受信任和经过验证的合作伙伴拥有的所有 AMI 启动实例。第一个语句的 Condition 元素测试 ec2:Owner 是不是 amazon。用户无法使用其他 AMI 启动实例（除非其他语句允许用户执行此操作）。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
         {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:instance/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
         ]
      }
   ]
}
```

实例类型

以下策略仅允许用户使用 t2.micro 或 t2.small 实例类型启动实例，您也可以通过此操作控制成本。用户无法启动更大的实例，因为第一条语句的 Condition 元素会测试 ec2:InstanceType 是否是 t2.micro 或 t2.small。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
         ]
      }
   ]
}
```

或者，您也可以创建一个策略，以拒绝用户启动 `t2.micro` 和 `t2.small` 实例类型之外的任何实例的权限。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
```

```
        {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringNotEquals": {
            "ec2:InstanceType": ["t2.micro", "t2.small"]
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:instance/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:volume/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*"
         ]
      }
   ]
}
```

子网

以下策略仅允许用户使用指定子网 subnet-*12345678* 启动实例。组无法将实例启动到任何其他子网中 (除非其他语句授予执行此操作的用户权限)。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
```

```
        "arn:aws:ec2:us-east-1:111122223333:subnet/subnet-12345678",
        "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
        "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "arn:aws:ec2:us-east-1:111122223333:volume/*",
        "arn:aws:ec2:us-east-1::image/ami-*",
        "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
        "arn:aws:ec2:us-east-1:111122223333:security-group/*"
      ]
    }
  ]
}
```

或者，您也可以创建一个策略，以拒绝用户将实例启动到任何其他子网的权限。该语句通过拒绝创建网络接口的权限来执行此操作，除非指定了子网 subnet-*12345678*。此拒绝会覆盖创建的任何其他策略以允许将实例启动到其他子网中。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
         {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*"
      ],
      "Condition": {
         "ArnNotEquals": {
            "ec2:Subnet": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-12345678"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:us-east-1::image/ami-*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:instance/*",
```

```
            "arn:aws:ec2:us-east-1:111122223333:subnet/*",
            "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "arn:aws:ec2:us-east-1:111122223333:key-pair/*",
            "arn:aws:ec2:us-east-1:111122223333:security-group/*"
            ]
        }
    ]
}
```

EBS 卷

仅当实例的 EBS 卷为加密卷时，下面的策略才允许用户启动实例。用户必须从使用加密快照创建的 AMI 启动实例，以确保根卷是加密的。此外，用户在启动期间附加到此实例的任何其他卷也必须是加密的。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:key-pair/*",
```

```
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

标签

在创建时标记实例

下面的策略允许用户启动实例并在创建期间标记实例。对于应用标签的资源创建操作，用户必须具有使用 CreateTags 操作的权限。第二个语句使用 ec2:CreateAction 条件键使用户只能在 RunInstances 上下文中且只能为实例创建标签。用户无法标记现有资源，并且用户无法使用 RunInstances 请求标记卷。

有关更多信息，请参阅 在创建过程中授予标记 Amazon EC2 资源的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
    }
```

```
  ]
}
```

在创建时使用特定标签标记实例和卷

下面的策略包含 aws:RequestTag 条件键，该条件键要求用户标记使用标签 RunInstances 和 environment=production 通过 purpose=webserver 创建的任何卷。如果用户不传递这些特定标签，或者根本不指定任何标签，则请求失败。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1::image/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
          "arn:aws:ec2:us-east-1:111122223333:volume/*",
          "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
         "StringEquals": {
            "aws:RequestTag/environment": "production" ,
            "aws:RequestTag/purpose": "webserver"
          }
```

```
        }
      },
      {
        "Effect": "Allow",
        "Action": [
           "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
        "Condition": {
           "StringEquals": {
               "ec2:CreateAction" : "RunInstances"
            }
         }
      }
   ]
}
```

在创建时使用至少一个特定标记标记实例和卷

下面的策略对 ForAnyValue 条件使用了 aws:TagKeys 修饰符，以指示必须在请求中指定至少一个标签，并且其必须包含键 environment 或 webserver。标签必须应用于实例及卷。可以在请求中指定任何标签值。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-1::image/*",
         "arn:aws:ec2:us-east-1:111122223333:subnet/*",
         "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
         "arn:aws:ec2:us-east-1:111122223333:security-group/*",
         "arn:aws:ec2:us-east-1:111122223333:key-pair/*"
      ]
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
          "ec2:RunInstances"
      ],
      "Resource": [
          "arn:aws:ec2:us-east-1:111122223333:volume/*",
          "arn:aws:ec2:us-east-1:111122223333:instance/*"
      ],
      "Condition": {
          "ForAnyValue:StringEquals": {
              "aws:TagKeys": ["environment","webserver"]
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
          "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
          "StringEquals": {
              "ec2:CreateAction" : "RunInstances"
          }
       }
    }
  ]
}
```

如果在创建时标记实例，则必须使用特定标签标记它们

在下面的策略中，用户不必在请求中指定标签，但如果用户指定标签，则标签必须为 `purpose=test`。不允许使用任何其他标签。用户可以在 `RunInstances` 请求中向任何可标记资源应用标签。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/purpose": "test",
             "ec2:CreateAction" : "RunInstances"
          },
          "ForAllValues:StringEquals": {
              "aws:TagKeys": "purpose"
          }
       }
    }
  ]
}
```

禁止任何人在创建时为 RunInstances 调用标签

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
```

```
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

仅允许为 spot-instances-request 使用特定标签。第二个意外的不一致之处在这里发挥了作用。在正常情况下，不指定标签将导致 Unauthenticated 错误。对于 spot-instances-request，如果没有 spot-instances-request 标签，则不会评估此策略，因此无标签的 Spot on Run 请求将成功。

启动模板中的标签

在以下示例中，用户可以启动实例，但前提是他们使用特定的启动模板 (`lt-09477bcd97b0d310e`)。`ec2:IsLaunchTemplateResource` 条件键禁止用户覆盖在启动模板中指定的任何资源。语句的第二部分允许用户在创建时标记实例 — 如果在启动模板中为实例指定了标签，则该语句部分是必需的。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/lt-09477bcd97b0d310e"
```

```
            },
            "Bool": {
               "ec2:IsLaunchTemplateResource": "true"
            }
         }
      },
      {
        "Effect": "Allow",
        "Action": [
           "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
        "Condition": {
           "StringEquals": {
               "ec2:CreateAction" : "RunInstances"
            }
         }
      }
   ]
}
```

Elastic GPUs

在以下策略中，用户可以启动实例并指定要附加到实例的 Elastic GPU。用户可以在任何区域中启动实例，但他们只能在 us-east-2 区域中启动期间附加 Elastic GPU。

ec2:ElasticGpuType 条件键确保实例使用 eg1.medium 或 eg1.large 弹性 GPU 类型。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
             {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:elastic-gpu/*"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "ec2:Region": "us-east-2",
                    "ec2:ElasticGpuType": [
                        "eg1.medium",
                        "eg1.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/ami-*",
                "arn:aws:ec2:*:111122223333:network-interface/*",
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ec2:*:111122223333:subnet/*",
                "arn:aws:ec2:*:111122223333:volume/*",
                "arn:aws:ec2:*:111122223333:key-pair/*",
                "arn:aws:ec2:*:111122223333:security-group/*"
            ]
        }
    ]
}
```

启动模板

在以下示例中，用户可以启动实例，但前提是他们使用特定的启动模板 (lt-09477bcd97b0d310e)。用户可以在 RunInstances 操作中指定参数以覆盖启动模板中的任何参数。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
```

```
            "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/lt-09477bcd97b0d310e"
          }
       }
    }
  ]
}
```

在该示例中，只有在用户使用启动模板时，他们才能启动实例。该策略使用 `ec2:IsLaunchTemplateResource` 条件键防止用户覆盖启动模板中任何预先存在的 ARN。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "*",
      "Condition": {
         "ArnLike": {
            "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
          "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    }
  ]
}
```

以下示例策略允许用户启动实例，但前提是他们使用启动模板。用户无法覆盖请求中的子网和网络接口参数；只能在启动模板中指定这些参数。语句的第一部分使用 NotResource 元素允许子网和网络接口以外的所有其他资源。语句的第二部分允许子网和网络接口资源，但前提是它们来自于启动模板。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": ["arn:aws:ec2:us-east-1:111122223333:subnet/*",
                      "arn:aws:ec2:us-east-1:111122223333:network-interface/*" ],
      "Condition": {
         "ArnLike": {
            "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          }
       }
    },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": ["arn:aws:ec2:us-east-1:111122223333:subnet/*",
                   "arn:aws:ec2:us-east-1:111122223333:network-interface/*" ],
      "Condition": {
         "ArnLike": {
            "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
          "Bool": {
            "ec2:IsLaunchTemplateResource": "true"
          }
       }
    }
  ]
}
```

以下示例允许用户启动实例，但前提是他们使用启动模板，并且启动模板具有标签 `Purpose=Webservers`。用户无法覆盖 `RunInstances` 操作中的任何启动模板参数。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "NotResource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
         "ArnLike": {
             "ec2:LaunchTemplate": "arn:aws:ec2:us-east-1:111122223333:launch-
template/*"
          },
         "Bool": {
             "ec2:IsLaunchTemplateResource": "true"
          }
       }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
       "StringEquals": {
           "aws:ResourceTag/Purpose": "Webservers"
        }
       }
     }
  ]
}
```

使用竞价型实例

您可以使用 RunInstances 操作创建竞价型实例请求，并在创建时标记竞价型实例请求。要为 RunInstances 指定的资源为 `spot-instances-request`。

在 IAM 策略中评估 `spot-instances-request` 资源，如下所示：

- 如果您在创建时未标记竞价型实例请求，则 Amazon EC2 不会在 RunInstances 语句中评估 `spot-instances-request` 资源。

- 如果您在创建时标记竞价型实例请求，则 Amazon EC2 会在 RunInstances 语句中评估 `spot-instances-request` 资源。

因此，对于 `spot-instances-request` 资源，以下规则适用于 IAM 策略：

- 如果您使用 RunInstances 创建竞价型实例请求，并且您不打算在创建时标记竞价型实例请求，则无需明确允许 `spot-instances-request` 资源；调用将成功。
- 如果您使用 RunInstances 创建竞价型实例请求并打算在创建时标记竞价型实例请求，则必须在 RunInstances 允许语句中包含 `spot-instances-request` 资源，否则调用将失败。
- 如果您使用 RunInstances 创建竞价型实例请求并打算在创建时标记竞价型实例请求，则必须在 CreateTags 允许语句中指定 `spot-instances-request` 资源或 `*` 通配符，否则调用将失败。

您可以使用 RunInstances 或 RequestSpotInstances 请求竞价型实例。以下示例 IAM 策略仅在使用 RunInstances 请求竞价型实例时适用。

示例：使用 RunInstances 请求竞价型实例

以下策略允许用户使用 RunInstances 操作请求竞价型实例。由 RunInstances 创建的 `spot-instances-request` 资源将请求 Spot 实例。

> **Note**
>
> 要使用 RunInstances 创建竞价型实例请求，您可以从 `spot-instances-request` 列表中省略 `Resource`（如果您不打算在创建时标记竞价型实例请求）。这是因为，如果在创建时未标记竞价型实例请求，则 Amazon EC2 不会在 RunInstances 语句中评估 `spot-instances-request` 资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
```

```
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        }
    ]
}
```

Warning

不支持 – 示例：拒绝用户使用 RunInstances 请求竞价型实例的权限

`spot-instances-request` 资源不支持以下策略。

以下策略旨在向用户授予启动按需型实例的权限，但拒绝用户请求竞价型实例的权限。由 RunInstances 创建的 `spot-instances-request` 资源是请求 Spot 实例的资源。第二个语句旨在拒绝针对 `spot-instances-request` 资源的 RunInstances 操作。但不支持此条件，这是因为，如果在创建时未标记竞价型实例请求，则 Amazon EC2 不会在 RunInstances 语句中评估 `spot-instances-request` 资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
```

```
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequestsNOTSUPPORTEDDONOTUSE",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
        }
    ]
}
```

示例：在创建时标记竞价型实例请求

以下策略允许用户标记在实例启动期间创建的所有资源。第一个语句允许 RunInstances 创建列出的资源。由 RunInstances 创建的 `spot-instances-request` 资源是请求 Spot 实例的资源。第二个语句提供了一个 * 通配符，以允许在实例启动时创建所有资源时对其进行标记。

> **Note**
>
> 如果您在创建时标记竞价型实例请求，则 Amazon EC2 会在 RunInstances 语句中评估 `spot-instances-request` 资源。因此，您必须明确允许 RunInstances 操作的 `spot-instances-request` 资源，否则调用将失败。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
```

```
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

示例：拒绝在创建时标记竞价型实例请求

以下策略拒绝用户标记在实例启动期间创建的资源的权限。

第一个语句允许 RunInstances 创建列出的资源。由 RunInstances 创建的 `spot-instances-request` 资源是请求 Spot 实例的资源。第二个语句提供了一个 * 通配符，以拒绝在实例启动时创建所有资源时对其进行标记。如果在创建时标记 `spot-instances-request` 或任何其他资源，则 RunInstances 调用将失败。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
```

```
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenyTagResources",
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Warning

不支持 – 示例：仅在为竞价型实例请求分配了特定标签时允许创建该请求

`spot-instances-request` 资源不支持以下策略。

以下策略旨在向 RunInstances 授予权限以仅在使用特定标签标记竞价型实例请求时创建该请求。

第一个语句允许 RunInstances 创建列出的资源。

第二个语句旨在向用户授予权限以仅在竞价型实例请求具有标签 `environment=production` 时创建该请求。如果将此条件应用于由 RunInstances 创建的其他资源，则不指定标签会导致 `Unauthenticated` 错误。但是，如果没有为竞价型实例请求指定标签，则 Amazon EC2 不会在 RunInstances 语句中评估 `spot-instances-request` 资源，这会导致 RunInstances 创建未标记的竞价型实例请求。

请注意，请指定 `environment=production` 之外的其他标签会导致错误 `Unauthenticated`，这是因为，如果用户标记竞价型实例请求，则 Amazon EC2 会在 RunInstances 语句中评估 `spot-instances-request` 资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*"
            ]
        },
        {
            "Sid":
 "RequestSpotInstancesOnlyIfTagIsEnvironmentProductionNOTSUPPORTEDDONOTUSE",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/
*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }

    ]
}
```

示例：在为竞价型实例请求分配了特定标签的情况下拒绝创建该请求

如果使用 environment=production 标记了竞价型实例请求，则以下策略将拒绝 RunInstances 创建该请求的权限。

第一个语句允许 RunInstances 创建列出的资源。

第二个语句在竞价型实例请求具有标签 environment=production 时拒绝用户创建该请求的权限。指定 environment=production 作为标签会导致 Unauthenticated 错误。指定其他标签或不指定标签将导致创建竞价型实例请求。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1:*:subnet/*",
                "arn:aws:ec2:us-east-1:*:network-interface/*",
                "arn:aws:ec2:us-east-1:*:security-group/*",
                "arn:aws:ec2:us-east-1:*:key-pair/*",
                "arn:aws:ec2:us-east-1:*:volume/*",
                "arn:aws:ec2:us-east-1:*:instance/*",
                "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequests",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
```

```
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

示例：使用预留实例

下面的策略向用户授予在账户中查看、修改和购买预留实例的权限。

无法为个别的预留实例设置资源级别的许可。此策略表示用户可以访问账户中的所有预留实例。

Resource 元素使用 * 通配符指示用户可以在操作中指定所有资源；在本例中，他们可以列出并修改账户中的所有 预留实例。他们还可以使用账户凭证购买预留实例。在 API 操作不支持资源级权限的情况下，也需要 * 通配符。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
        "ec2:ModifyReservedInstances",
        "ec2:PurchaseReservedInstancesOffering",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
    }
   ]
}
```

要允许用户查看和修改账户中的 预留实例，但不允许购买新的 预留实例，请使用以下命令：

JSON

```
{
  "Version":"2012-10-17",
   "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeReservedInstances",
        "ec2:ModifyReservedInstances",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*"
    }
  ]
}
```

示例：标记资源

仅当标签包含键 CreateTags 和值 environment 时，下面的策略才允许用户使用 production 操作向实例应用标签。不允许使用其他标签，并且用户无法为任何其他资源类型添加标签。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        }
```

```
    ]
}
```

以下策略允许用户标记已具有键为 owner、值为用户名的标签的任何可标记资源。此外，用户还必须在请求中指定键为 anycompany:environment-type、值为 test 或 prod 的标签。用户可以在请求中指定其他的标签。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/anycompany:environment-type":
 ["test","prod"],
                    "aws:ResourceTag/owner": "${aws:username}"
                }
            }
        }
    ]
}
```

您可以创建允许用户删除资源的特定标签的 IAM 策略。例如，当在请求中指定的标签键为 environment 或 cost-center 时，下面的策略允许用户删除卷的标签。可以为此标签指定任何值，但标签键必须匹配某个指定键。

> **Note**
>
> 如果删除资源，则所有与资源相关的标签都将被删除。用户不需要使用 ec2:DeleteTags 操作删除具有标签的资源的权限，他们仅需要执行删除操作的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Effect": "Allow",
      "Action": "ec2:DeleteTags",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment","cost-center"]
        }
      }
    }
  ]
}
```

该策略仅允许用户删除任何资源上的 environment=prod 标签，但前提是已使用键为 owner、值为用户名的标签标记该资源。用户无法删除资源的任何其他标签。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "prod",
          "aws:ResourceTag/owner": "${aws:username}"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["environment"]
        }
```

```
      }
    }
  ]
}
```

示例：使用 IAM 角色

以下策略允许用户将 IAM 角色附加、替换到具有标签 `department=test` 的实例或与之分离。替换或分离 IAM 角色需要一个关联 ID，因此该策略还授予用户使用 `ec2:DescribeIamInstanceProfileAssociations` 操作的权限。

用户必须具有使用 `iam:PassRole` 操作的权限，才能将角色传递到实例。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation",
        "ec2:DisassociateIamInstanceProfile"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department":"test"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeIamInstanceProfileAssociations",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/DevTeam*"
```

```
    }
  ]
}
```

以下策略允许用户为所有实例附加或替换 IAM 角色。用户只能附加或替换名称以 TestRole- 开头的 IAM 角色。对于 iam:PassRole 操作，请确保您指定的是 IAM 角色的名称而不是实例配置文件的名称（如果名称不同）。有关更多信息，请参阅 实例配置文件。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/TestRole-*"
        }
    ]
}
```

示例：使用路由表

以下策略允许用户添加、删除和替换仅与 VPC vpc-ec43eb89 关联的路由表的路由。要为 ec2:Vpc 条件键指定 VPC，必须指定 VPC 的完整 ARN。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-
ec43eb89"
                }
            }
        }
    ]
}
```

示例：允许特定实例查看其他 Amazon 服务中的资源

下面是您可能附加到 IAM 角色的策略的示例。该策略允许实例查看不同 Amazon 服务中的资源。它使用 `ec2:SourceInstanceARN` 全局条件键指定从中发出请求的实例必须是实例 `i-093452212644b0dd6`。如果同一个 IAM 角色还与另一个实例关联，则另一个实例无法执行任何这些操作。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:DescribeVolumes",
                "s3:ListAllMyBuckets",
                "dynamodb:ListTables",
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ArnEquals": {
                    "ec2:SourceInstanceARN": "arn:aws:ec2:us-
east-1:111122223333:instance/i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

示例：使用启动模板

以下策略允许用户创建启动模板版本和修改启动模板，但仅适用于特定的启动模板 (`lt-09477bcd97b0d3abc`)。用户无法使用其他启动模板。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
   {
      "Action": [
        "ec2:CreateLaunchTemplateVersion",
        "ec2:ModifyLaunchTemplate"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/
lt-09477bcd97b0d3abc"
    }
  ]
}
```

以下策略允许用户删除任何启动模板和启动模板版本，但前提是启动模板具有标签 `Purpose=Testing`。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
       {
      "Action": [
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:us-east-1:111122223333:launch-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Testing"
        }
      }
    }
  ]
}
```

使用实例元数据

以下策略确保用户只能使用 实例元数据服务版本 2 (IMDSv2) 检索实例元数据。您可以将以下四个策略合并为一个具有四个语句的策略。当合并为一个策略时，您可以将该策略用作服务控制策略 (SCP)。它可以很好地用作应用于现有 IAM 策略的拒绝策略（取消和限制现有权限），也可以很好地用作在账户、组织单元（OU）或整个组织间全局应用的 SCP。

> **Note**
>
> 以下 RunInstances 元数据选项策略必须与授予主体使用 RunInstances 启动实例的权限的策略结合使用。如果主体没有同时具有 RunInstances 权限，则无法启动实例。有关更多信息，请参阅使用实例和启动实例 (RunInstances) 中的策略。

Important

如果您使用自动扩缩组且需要要求对所有新实例使用 IMDSv2，您的自动扩缩组必须使用启动模板。
当自动扩缩组使用启动模板时，会在创建新自动扩缩组时检查 IAM 主体的 `ec2:RunInstances` 权限。当更新现有自动扩缩组以使用新启动模板或新版本的启动模板时，也会检查这些内容。
只有在创建或更新正在使用启动模板的自动扩缩组时，才会检查对 `RunInstances` 的 IAM 主体使用 IMDSv1 的限制。对于配置为使用 `Latest` 或 `Default` 启动模板的自动扩缩组，在创建启动模板的新版本时不会检查权限。要检查权限，您必须将自动扩缩组配置为使用特定版本的启动模板。

要在自动扩缩组启动的实例上强制使用 IMDSv2，需要执行以下附加步骤：

1. 通过对创建的新主体使用服务控制策略（SCP）或 IAM 权限边界，请对组织中的所有账户禁用启动配置。对于具有自动扩缩组权限的现有 IAM 主体，请使用此条件键更新其关联策略。要禁用启动配置，请使用值指定为 `"autoscaling:LaunchConfigurationName"` 的 `null` 条件键创建或修改相关 SCP、权限边界或 IAM 策略。
2. 对于新启动模板，请在启动模板中配置实例元数据选项。对于现有启动模板，创建启动模板的新版本，并在新版本中配置实例元数据选项。
3. 在向任何主体授予使用启动模板的权限的策略中，通过指定 `$latest` 来限制 `$default` 和 `"autoscaling:LaunchTemplateVersionSpecified": "true"` 的关联。
通过限制只使用特定版本的启动模板，您可以确保使用在其中配置实例元数据选项的版本启动新实例。有关更多信息，请参阅 Amazon EC2 Auto Scaling API 参考 中的 LaunchTemplateSpecification，特别是 `Version` 参数。
4. 对于使用启动配置的自动扩缩组，请将启动配置替换为启动模板。有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Migrate your Auto Scaling groups to launch templates。
5. 对于使用启动模板的自动扩缩组，请确保它使用配置了实例元数据选项的新启动模板，或使用配置了实例元数据选项的新版本的当前启动模板。有关更多信息，请参阅 update-auto-scaling-group。

示例

- 要求使用 IMDSv2
- 拒绝退出 IMDSv2

- 指定最大跃点数限制
- 限制可以修改实例元数据选项的用户
- 要求从 IMDSv2 检索角色凭证

要求使用 IMDSv2

以下策略指定您不能调用 RunInstances API，除非该实例也选择需要使用 IMDSv2（由 `"ec2:MetadataHttpTokens": "required"` 指示）。如果您未指定实例需要 IMDSv2，则在调用 RunInstances API 时会收到 `UnauthorizedOperation` 错误。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Sid": "RequireImdsV2",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringNotEquals": {
                    "ec2:MetadataHttpTokens": "required"
                }
            }
        }
    ]
}
```

拒绝退出 IMDSv2

以下策略指定您不能调用 `ModifyInstanceMetadataOptions` API 并允许选择 IMDSv1 或 IMDSv2。如果您调用 `ModifyInstanceMetadataOptions` API，则必须将 `HttpTokens` 属性设置为 `required`。

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "DenyIMDSv1HttpTokensModification",
        "Effect": "Deny",
        "Action": "ec2:ModifyInstanceMetadataOptions",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringNotEquals": {
                "ec2:Attribute/HttpTokens": "required"
            },
            "Null": {
                "ec2:Attribute/HttpTokens": false
            }
        }
    }]
}
```

指定最大跃点数限制

以下策略指定您不能调用 RunInstances API，除非您还指定了跃点限制，且跃点限制不能超过 3。如果您无法执行此操作，则在调用 RunInstances API 时会收到 `UnauthorizedOperation` 错误。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
               {
            "Sid": "MaxImdsHopLimit",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "NumericGreaterThan": {
                    "ec2:MetadataHttpPutResponseHopLimit": "3"
                }
            }
        }
    ]
}
```

限制可以修改实例元数据选项的用户

以下策略仅允许具有 ec2-imds-admins 角色的用户对实例元数据选项进行更改。如果除 ec2-imds-admins 角色以外的任何主体尝试调用 ModifyInstanceMetadataOptions API，则会收到 Unauthorized0peration 错误。此语句可用于控制 ModifyInstanceMetadataOptions API 的使用；目前对于 ModifyInstanceMetadataOptions API 没有精细访问控制（条件）。

要求从 IMDSv2 检索角色凭证

以下策略指定如果将此策略应用于某个角色，并且该角色由 EC2 服务代入且生成的凭证用于对请求进行签名，则必须由从 IMDSv2 中检索的 EC2 角色凭证对该请求进行签名。否则，它的所有 API 调用都会收到 Unauthorized0peration 错误。此语句/策略可广泛应用，因为如果请求未由 EC2 角色证书签名，则其为无效。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
                {
            "Sid": "RequireAllEc2RolesToUseV2",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:RoleDelivery": "2.0"
                }
            }
        }
    ]
}
```

使用 Amazon EBS 卷和快照

有关使用 Amazon EBS 卷和快照的策略示例，请参阅 Amazon EBS 的基于身份的策略示例。

用于控制对 Amazon EC2 控制台的访问的示例策略

您可以使用 IAM 策略向用户授予使用 Amazon EC2 所需的权限。有关分步说明，请参阅《IAM 用户指南》中的创建 IAM 策略。

控制台使用其他 API 操作实现其功能，因此这些策略可能不会按预期方式起作用。例如，只拥有 `DescribeVolumes` API 操作使用权限的用户在控制台中查看卷时会遇到错误。此部分演示使用户可以使用控制台的特定部分的策略。有关创建 Amazon EC2 控制台的策略的更多信息，请参阅发布的以下Amazon安全博客：授予用户在 Amazon EC2 控制台中工作的权限。

以下示例显示了您可用于向用户授予使用 Amazon EC2 的权限的策略语句。将每个`#######`替换为您自己的信息。这些策略设计用于采用 Amazon Web Services 管理控制台 发出的请求。Amazon EC2 控制台可能会调用多个 API 操作来显示单个资源，但在用户尝试执行任务且控制台显示错误之前，这可能并不明显。有关更多信息，请参阅以下 Amazon Security Blog 文章：Granting Users Permission to Work in the Amazon EC2 Console。

示例

- 示例：只读访问权限
- 示例：使用 EC2 启动实例向导
- 示例：使用安全组
- 示例：使用弹性 IP 地址
- 示例：使用预留实例

为帮助了解在控制台中执行任务所需的相应 API 操作，您可以使用能够记录调用的服务，例如 Amazon CloudTrail。如果您的策略不授予创建或修改特定资源的权限，则控制台显示一个包含诊断信息的编码消息。您可以使用适用于 Amazon STS 的 DecodeAuthorizationMessage API 操作或 Amazon CLI 中的 decode-authorization-message 命令对该消息解码。

示例：只读访问权限

要允许用户在 Amazon EC2 控制台中查看所有资源，您可以使用与以下示例相同的策略：示例：只读访问权限用户无法对这些资源执行任何操作或创建新资源 (除非其他语句为用户授予执行此操作的权限)。

查看实例、AMI 和快照

或者，您可以提供对资源子集的只读访问权限。为此，请对每个资源将 `ec2:Describe` API 操作中的 * 通配符替换为特定 `ec2:Describe` 操作。以下策略允许用户在 Amazon EC2 控制台中查看所有实

例、AMI 和快照。ec2:DescribeTags 操作允许用户查看公用 AMI。控制台需要标记信息来显示公用 AMI；但是，您可以删除此操作以允许用户只查看私有 AMI。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeImages",
         "ec2:DescribeTags",
         "ec2:DescribeSnapshots"
      ],
      "Resource": "*"
   }
   ]
}
```

> **Note**
>
> Amazon EC2 ec2:Describe* API 操作不支持资源级权限，因此您无法控制用户可以在控制台中查看哪些单个资源。因此，在以上语句的 Resource 元素中需要 * 通配符。有关哪些 ARN 可用于哪些 Amazon EC2 API 操作的更多信息，请参阅 Amazon EC2 的操作、资源和条件键。

查看实例和 CloudWatch 指标

以下策略允许用户在 Amazon EC2 控制台中查看实例，以及在 Instances 页面的 Monitoring 选项卡中查看 CloudWatch 警报和指标。Amazon EC2 控制台使用 CloudWatch API 显示警报和指标，因此您必须授权用户执行 cloudwatch:DescribeAlarms、cloudwatch:DescribeAlarmsForMetric、cloudwatch:ListMet 和 cloudwatch:GetMetricData 操作。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
         "ec2:DescribeInstanceTypes",
         "cloudwatch:DescribeAlarms",
         "cloudwatch:DescribeAlarmsForMetric",
         "cloudwatch:ListMetrics",
         "cloudwatch:GetMetricStatistics",
         "cloudwatch:GetMetricData"
      ],
      "Resource": "*"
   }
   ]
}
```

示例：使用 EC2 启动实例向导

Amazon EC2 启动实例向导是一个屏幕，其中包含用于配置和启动实例的选项。您的策略必须包含允许用户使用向导选项的 API 操作使用权限。如果您的策略不包含使用这些操作的权限，则向导中的一些项目无法正确加载，用户无法完成启动。

启动实例向导基本访问权限

要成功完成启动，必须为用户授予使用 `ec2:RunInstances` API 操作以及至少以下 API 操作的权限：

- `ec2:DescribeImages`：查看并选择 AMI。
- `ec2:DescribeInstanceTypes`：查看并选择一种实例类型。
- `ec2:DescribeVpcs`：查看可用网络选项。
- `ec2:DescribeSubnets`：查看所选 VPC 的所有可用子网。
- `ec2:DescribeSecurityGroups` 或 `ec2:CreateSecurityGroup`：查看并选择现有安全组，或者创建新的安全组。
- `ec2:DescribeKeyPairs` 或 `ec2:CreateKeyPair`：选择现有密钥对，或者创建新密钥对。

- ec2:AuthorizeSecurityGroupIngress：添加入站规则。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:CreateSecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateKeyPair"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*"
        }
    ]
}
```

您可以向策略添加 API 操作以便为用户提供更多选项，例如：

- ec2:DescribeAvailabilityZones：查看并选择特定可用区。
- ec2:DescribeNetworkInterfaces：查看并选择所选子网的现有网络接口。
- 要向 VPC 安全组添加出站规则，必须为用户授予使用 ec2:AuthorizeSecurityGroupEgress API 操作的权限。要修改或删除现有规则，必须为用户授予使用相关 ec2:RevokeSecurityGroup* API 操作的权限。

- `ec2:CreateTags`：标记通过 `RunInstances` 创建的资源。有关更多信息，请参阅 在创建过程中授予标记 Amazon EC2 资源的权限。如果用户没有使用此操作的权限而又尝试在启动实例向导的标记页上应用标签，则启动失败。

  > **Important**
  >
  > 启动实例时指定 Name（名称）会创建一个标签并需要 `ec2:CreateTags` 操作。向用户授予使用 `ec2:CreateTags` 操作的权限时请谨慎，因为这样做会限制您使用 `aws:ResourceTag` 条件键限制用户使用其他资源的功能。如果您授予用户使用 `ec2:CreateTags` 操作的权限，则他们可以更改资源的标签以绕过这些限制。有关更多信息，请参阅 使用基于属性的访问控制访问权限。

- 要在选择 AMI 时使用 Systems Manager 参数，您必须将 `ssm:DescribeParameters` 和 `ssm:GetParameters` 添加到 IAM 策略中。`ssm:DescribeParameters` 会授予用户查看和选择 Systems Manager 参数的权限。`ssm:GetParameters` 则会授予用户获取 Systems Manager 参数值的权限。您还可以限制对特定 Systems Manager 参数的访问权限。有关更多信息，请参阅本节后文中的限制对特定 Systems Manager 参数的访问。

当前，Amazon EC2 `Describe*` API 操作不支持资源级权限，因此您无法限制用户可以在启动实例向导中查看的单个资源。但是，您可以对 `ec2:RunInstances` API 操作应用资源级权限，以限制用户可以用于启动实例的资源。如果用户选择未授权他们使用的选项，则启动会失败。

限制对特定实例类型、子网和区域的访问

以下策略允许用户使用 Amazon 拥有的 AMI 启动 `t2.micro` 实例，并且仅在特定子网 (`subnet-1a2b3c4d`) 中启动。用户只能在指定区域中启动。如果用户在启动实例向导中选择不同区域或选择不同实例类型、AMI 或子网，则启动会失败。

第一条语句为用户授予权限以查看启动实例向导中的选项或创建新选项，如上例所示。第二条语句为用户授予将网络接口、卷、密钥对、安全组和子网资源 (在 VPC 中启动实例需要这些资源) 用于 `ec2:RunInstances` 操作的权限。有关使用 `ec2:RunInstances` 操作的更多信息，请参阅 启动实例 (RunInstances)。第三和第四条语句分别为用户授予权限以使用实例（仅当实例是 `t2.micro` 实例时）和 AMI 资源（仅当 AMI 由 Amazon 或某些受信任或经过验证的合作伙伴所有时）。

JSON

```
{
   "Version":"2012-10-17",
```

```
    "Statement": [{
       "Effect": "Allow",
       "Action": [
          "ec2:DescribeInstances",
          "ec2:DescribeImages",
          "ec2:DescribeInstanceTypes",
          "ec2:DescribeKeyPairs",
          "ec2:CreateKeyPair",
          "ec2:DescribeVpcs",
          "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups",
          "ec2:CreateSecurityGroup",
          "ec2:AuthorizeSecurityGroupIngress"
    ],
    "Resource": "*"
    },
    {
       "Effect": "Allow",
       "Action":"ec2:RunInstances",
       "Resource": [
          "arn:aws:ec2:us-east-2:111122223333:network-interface/*",
          "arn:aws:ec2:us-east-2:111122223333:volume/*",
          "arn:aws:ec2:us-east-2:111122223333:key-pair/*",
          "arn:aws:ec2:us-east-2:111122223333:security-group/*",
          "arn:aws:ec2:us-east-2:111122223333:subnet/subnet-1a2b3c4d"
       ]
    },
    {
       "Effect": "Allow",
       "Action": "ec2:RunInstances",
       "Resource": [
          "arn:aws:ec2:us-east-2:111122223333:instance/*"
       ],
       "Condition": {
          "StringEquals": {
             "ec2:InstanceType": "t2.micro"
          }
       }
    },
    {
       "Effect": "Allow",
       "Action": "ec2:RunInstances",
       "Resource": [
             "arn:aws:ec2:us-east-2::image/ami-*"
       ],
```

```
      "Condition": {
         "StringEquals": {
            "ec2:Owner": "amazon"
         }
      }
   }
   ]
}
```

限制对特定 Systems Manager 参数的访问

以下策略授予访问权限以使用具有特定名称的 Systems Manager 参数。

第一条语句授予用户权限，可在启动实例向导中选择 AMI 时查看 Systems Manager 参数。第二条语句授予用户仅使用名为 `prod-*` 的参数的权限。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ssm:DescribeParameters"
      ],
      "Resource": "*"
   },
   {
      "Effect": "Allow",
      "Action": [
         "ssm:GetParameters"
      ],
     "Resource": "arn:aws:ssm:us-east-2:123456123456:parameter/prod-*"
   }
   ]
}
```

示例：使用安全组

查看安全组以及添加和删除规则

以下策略为用户授予相应的权限，以便在 Amazon EC2 控制台中查看安全组、添加和删除入站和出站规则，以及为具有标签 Department=Test 的现有安全组列出并修改规则描述。

在第一条语句中，ec2:DescribeTags 操作允许用户在控制台中查看标签，这样，用户更易于识别自己可修改的安全组。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeSecurityGroups",
         "ec2:DescribeSecurityGroupRules",
         "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:AuthorizeSecurityGroupIngress",
         "ec2:RevokeSecurityGroupIngress",
         "ec2:AuthorizeSecurityGroupEgress",
         "ec2:RevokeSecurityGroupEgress",
         "ec2:ModifySecurityGroupRules",
         "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
         "ec2:UpdateSecurityGroupRuleDescriptionsEgress"
      ],
      "Resource": [
         "arn:aws:ec2:us-east-2:111122223333:security-group/*"
      ],
      "Condition": {
         "StringEquals": {
            "aws:ResourceTag/Department": "Test"
         }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
         "ec2:ModifySecurityGroupRules"
```

```
      ],
      "Resource": [
         "arn:aws:ec2:us-east-2:111122223333:security-group-rule/*"
      ]
   }
]}
```

使用 Create Security Group（创建安全组）对话框

您可以创建一个策略，以允许用户使用 Amazon EC2 控制台中的 Create Security Group (创建安全组) 对话框。要使用此对话框，必须为用户授予使用至少以下 API 操作的权限：

- `ec2:CreateSecurityGroup`: 创建新安全组。
- `ec2:DescribeVpcs`查看 VPC 列表中的现有 VPC 列表。

借助这些权限，用户可以成功创建新安全组，但是他们不能向其中添加任何规则。要在 Create Security Group (创建安全组) 对话框中使用规则，您可以向策略添加以下 API 操作：

- `ec2:AuthorizeSecurityGroupIngress`：添加入站规则。
- `ec2:AuthorizeSecurityGroupEgress`：向 VPC 安全组添加出站规则。
- `ec2:RevokeSecurityGroupIngress`：修改或删除现有入站规则。如果要允许用户使用控制台中的 Copy to new 功能，这十分有用。此功能会打开 Create Security Group (创建安全组) 对话框，并使用所选安全组的规则进行填充。
- `ec2:RevokeSecurityGroupEgress`：修改或删除适用于 VPC 安全组的出站规则。若要允许用户修改或删除允许所有出站流量的默认出站规则，这十分有用。
- `ec2:DeleteSecurityGroup`：适用于无效规则无法保存的情况。控制台首先创建安全组，然后添加指定的规则。如果规则无效，则操作会失败，而控制台会尝试删除安全组。用户仍会停留在“Create Security Group”对话框中，这样就能更正无效规则和尝试重新创建安全组。此 API 操作不是必需的，但是如果用户在无权使用它的情况下尝试创建具有无效规则的安全组，则会创建不包含任何规则的安全组，用户必须在之后添加规则。
- `ec2:UpdateSecurityGroupRuleDescriptionsIngress`：添加或更新入口（入站）安全组规则的描述。
- `ec2:UpdateSecurityGroupRuleDescriptionsEgress`：添加或更新出口（出站）安全组规则的描述。
- `ec2:ModifySecurityGroupRules`：修改安全组规则。
- `ec2:DescribeSecurityGroupRules`：列出安全组规则。

以下策略向用户授予使用 Create Security Group (创建安全组) 对话框，以及为与特定 VPC (vpc-1a2b3c4d) 关联的安全组创建入站和出站规则的权限。用户可以为 VPC 创建安全组，但是无法向它们添加任何规则。同样，用户无法向不与 VPC vpc-1a2b3c4d 关联的任何现有安全组添加任何规则。还向用户授予了在控制台中查看所有安全组的权限。这样，用户更易于识别自己可添加入站规则的安全组。此策略还为用户授予删除与 VPC vpc-1a2b3c4d 关联的安全组的权限。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress"
      ],
      "Resource": "arn:aws:ec2:us-east-2:111122223333:security-group/*",
      "Condition":{
         "ArnEquals": {
            "ec2:Vpc": "arn:aws:ec2:us-east-2:111122223333:vpc/vpc-1a2b3c4d"
         }
      }
    }
   ]
}
```

示例：使用弹性 IP 地址

为了让用户能够查看 Amazon EC2 控制台中的弹性 IP 地址，您必须授予用户使用 ec2:DescribeAddresses 操作的权限。

要允许用户使用弹性 IP 地址，可将以下操作添加到您策略中。

- ec2:AllocateAddress：分配弹性 IP 地址。
- ec2:ReleaseAddress: 解除弹性 IP 地址。
- ec2:AssociateAddress：将弹性 IP 地址与实例或网络接口关联。
- ec2:DescribeNetworkInterfaces 和 ec2:DescribeInstances：使用 Associate Address (关联地址) 屏幕。屏幕显示了您可以将弹性 IP 地址关联到的可用实例或网络接口。
- ec2:DisassociateAddress：取消弹性 IP 地址与实例或网络接口的关联。

以下策略允许用户查看弹性 IP 地址并将其分配给实例和与实例相关联。用户不可以将弹性 IP 地址与网络接口关联、取消弹性 IP 地址的关联或释放弹性 IP 地址。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAddresses",
                "ec2:AllocateAddress",
                "ec2:DescribeInstances",
                "ec2:AssociateAddress"
            ],
            "Resource": "*"
        }
    ]
}
```

示例：使用预留实例

以下策略允许用户查看和修改账户中的预留实例，同时也能在 Amazon Web Services 管理控制台 内购买新的预留实例。

此策略允许用户查看账户中的所有预留实例和按需型实例。无法为单个预留实例设置资源级别的权限。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeReservedInstances",
         "ec2:ModifyReservedInstances",
         "ec2:PurchaseReservedInstancesOffering",
         "ec2:DescribeInstances",
         "ec2:DescribeInstanceTypes",
         "ec2:DescribeAvailabilityZones",
         "ec2:DescribeReservedInstancesOfferings"
      ],
      "Resource": "*"
   }
   ]
}
```

必须执行 ec2:DescribeAvailabilityZones 操作才能确保 Amazon EC2 控制台可以显示有关您能够购买预留实例的可用区的信息。ec2:DescribeInstances 操作不是必须的，但是请确保用户可查看账户内的实例并且能够购买预留实例，以匹配正确的规格。

您可以调整 API 操作，以限制用户访问，例如删除ec2:DescribeInstances，而ec2:DescribeAvailabilityZones表示用户有只读形式的访问权。

Amazon EC2 的 Amazon 托管式策略

要向用户、组和角色添加权限，与自己编写策略相比，使用 Amazon 托管式策略更简单。创建仅为团队提供所需权限的 IAM 客户管理型策略需要时间和专业知识。要快速入门，您可以使用我们的 Amazon 托管式策略。这些策略涵盖常见使用案例，可在您的 Amazon 账户中使用。有关 Amazon 托管式策略的更多信息，请参阅《IAM 用户指南》中的 Amazon 托管式策略。

Amazon 服务负责维护和更新 Amazon 托管式策略。您无法更改 Amazon 托管式策略中的权限。服务偶尔会向 Amazon 托管式策略添加额外权限以支持新特征。此类更新会影响附加策略的所有身份（用户、组和角色）。当启动新特征或新操作可用时，服务最有可能会更新 Amazon 托管式策略。服务不会从 Amazon 托管式策略中删除权限，因此策略更新不会破坏您的现有权限。

此外，Amazon 还支持跨多种服务的工作职能的托管式策略。例如，ReadOnlyAccess Amazon 托管式策略提供对所有 Amazon 服务和资源的只读访问权限。当服务启动新特征时，Amazon 会为新操作和资源添加只读权限。有关工作职能策略的列表和说明，请参阅《IAM 用户指南》中的适用于工作职能的 Amazon 托管策略。

Amazon 托管策略：AmazonEC2FullAccess

您可以将 `AmazonEC2FullAccess` 策略附加得到 IAM 身份。此策略授予允许完全访问 Amazon EC2 的权限。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AmazonEC2FullAccess。

Amazon 托管式策略：AmazonEC2ReadOnlyAccess

您可以将 `AmazonEC2ReadOnlyAccess` 策略附加到 IAM 身份。此策略授予允许对 Amazon EC2 进行只读访问的权限。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AmazonEC2ReadOnlyAccess。

Amazon 托管式策略：AmazonEC2ImageReferencesAccessPolicy

您可以将 `AmazonEC2ImageReferencesAccessPolicy` 策略附加到 IAM 身份。此策略授予使用 EC2 DescribeImageReferences API 所需的权限，包括查看 EC2 实例、启动模板、Systems Manager 参数和 Image Builder 配方的权限。该策略支持 `IncludeAllResourceTypes` 标志，并且当 Amazon 添加对新资源类型的支持时将继续工作，从而无需进行未来的策略更新。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AmazonEC2ImageReferencesAccessPolicy。

Amazon 托管式策略：AWSEC2CapacityReservationFleetRolePolicy

此策略将附加到名为 AWSServiceRoleForEC2CapacityReservationFleet 的服务相关角色，以允许服务代表您在容量预留实例集中创建、修改和取消容量预留。有关更多信息，请参阅 将服务相关角色用于容量预留实例集。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AWSEC2CapacityReservationFleetRolePolicy。

Amazon 托管式策略：AWSEC2FleetServiceRolePolicy

此策略将附加到名为 AWSServiceRoleForEC2Fleet 的服务相关角色，以允许 EC2 实例集代表您请求、启动、终止和标记实例。有关更多信息，请参阅 EC2 实例集的服务相关角色。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AWSEC2FleetServiceRolePolicy。

Amazon 托管式策略：AWSEC2SpotFleetServiceRolePolicy

此策略将附加到名为 AWSServiceRoleForEC2SpotFleet 的服务相关角色，以允许竞价型实例集代表您启动和管理实例。有关更多信息，请参阅 竞价型实例集的服务相关角色。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AWSEC2SpotFleetServiceRolePolicy。

Amazon 托管式策略：AWSEC2SpotServiceRolePolicy

此策略将附加到名为 AWSServiceRoleForEC2Spot 的服务相关角色，以允许 Amazon EC2 代表您启动和管理 Spot 实例。有关更多信息，请参阅 竞价型实例请求的服务相关角色。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AWSEC2SpotServiceRolePolicy。

Amazon 托管式策略：AWSEC2VssSnapshotPolicy

您可以将此托管策略附加到用于 Amazon EC2 Windows 实例的 IAM 实例配置文件角色。该策略授予允许 Amazon EC2 代表您创建和管理 VSS 快照的权限。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AWSEC2VssSnapshotPolicy。

Amazon 托管式策略：DeclarativePoliciesEC2Report

此策略将附加到名为 `AWSServiceRoleForDeclarativePoliciesEC2Report` 的服务相关角色，以提供对为声明式策略生成账户状态报告所需的只读 API 的访问权限。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 DeclarativePoliciesEC2Report。

Amazon 托管式策略：EC2FastLaunchFullAccess

您可以将 `EC2FastLaunchFullAccess` 策略附加到实例配置文件或其他 IAM 角色。此策略会授予对 EC2 Fast Launch 操作的完全访问权限以及目标权限，如下所示。

权限详细信息

- EC2 Fast Launch – 授予管理访问权限，便于角色启用或禁用 EC2 Fast Launch 并描述 EC2 Fast Launch 映像。
- Amazon EC2：授予对 Amazon EC2 RunInstances、CreateTags，以及启动模版描述、创建和修改操作的访问权限。此外还授予了创建网络和安全组资源、授权入口规则以及删除 EC2 Fast Launch 所创建资源的访问权限。

- IAM – 授予访问权限，以获取和使用名称包含 `ec2fastlaunch` 的实例配置文件，进而创建 EC2FastLaunchServiceRolePolicy 服务相关角色。
- Amazon CloudFormation：向 EC2 Fast Launch 授予了描述和创建 CloudFormation 堆栈以及删除所创建堆栈的访问权限。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 EC2FastLaunchFullAccess。

Amazon 托管式策略：AWSEC2CapacityManagerServiceRolePolicy

此策略将附加到名为 AWSServiceRoleForEC2CapacityManager 的服务相关角色，以允许 EC2 容量管理器代表您管理容量资源并与 Amazon Organizations 集成。有关更多信息，请参阅 EC2 容量管理器的服务相关角色。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 AWSEC2CapacityManagerServiceRolePolicy。

Amazon 托管式策略：EC2FastLaunchServiceRolePolicy

此策略将附加到名为 AWSServiceRoleForEC2FastLaunch 的服务相关角色，以允许 Amazon EC2 创建和管理一组预置快照，从而减少从启用了 EC2 Fast Launch 的 AMI 启动实例所需的时间。有关更多信息，请参阅 the section called “服务相关角色”。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 EC2FastLaunchServiceRolePolicy。

Amazon 托管式策略：Ec2InstanceConnect

您可以将 `Ec2InstanceConnect` 策略附加到 IAM 身份。此策略授予权限，允许客户调用 EC2 Instance Connect 向其 EC2 实例发布临时密钥并通过 ssh 或 EC2 Instance Connect CLI 进行连接。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 Ec2InstanceConnect。

Amazon 托管式策略：Ec2InstanceConnectEndpoint

此策略附加到名为 AWSServiceRoleForEC2InstanceConnect 的服务相关角色，以允许 EC2 Instance Connect 端点代表您执行操作。有关更多信息，请参阅 EC2 Instance Connect Endpoint 的服务相关角色。

要查看此策略的权限，请参阅《Amazon 托管式策略参考》中的 Ec2InstanceConnectEndpoint。有关此策略更新的说明，请参阅 Amazon 托管策略的 Amazon EC2 更新。

Amazon 托管策略的 Amazon EC2 更新

查看有关 Amazon EC2 的 Amazon 托管策略的更新的详细信息（此服务开始跟踪这些更改）。

更改	描述	日期
AWSEC2CapacityManagerServiceRolePolicy - 新策略	Amazon EC2 新增了此策略，以便可以代表您管理容量资源并与 Amazon Organizations 集成。	2025 年 10 月 15 日
AmazonEC2ImageReferencesAccessPolicy：新策略	Amazon EC2 添加了此策略，以提供扫描 EC2 DescribeImageReferences API 支持的所有资源类型的权限。	2025 年 8 月 26 日
Ec2InstanceConnectEndpoint - 更新的策略	为了支持修改现有实例连接端点，Amazon EC2 更新了此策略，以添加分配和取消分配 IPv6 地址的权限，并修改 EC2 实例连接端点创建的网络接口上的安全组。Amazon EC2 还更新了此策略，用 `StringLike` 条件运算符替换了 `Null` 条件运算符。	2025 年 7 月 31 日
EC2FastLaunchServiceRolePolicy - 更新的策略	为了防止出现孤立资源，Amazon EC2 对此策略进行了更新，增加了描述卷、卷属性和网络接口的权限，并增加了删除 EC2 Fast Launch 创建的卷和网络接口的权限。	2025 年 7 月 17 日
EC2FastLaunchFullAccess - 更新的策略	Amazon EC2 对此策略进行了更新，包含了启动模板创建和修改操作的权限，以及创建网络和安全组资源、授权入口规则以及删除 EC2 Fast Launch	2025 年 5 月 14 日

更改	描述	日期
	所创建资源的权限。此外还可以描述和创建 CloudFormation 堆栈，以及删除 EC2 Fast Launch 创建的堆栈。	
EC2FastLaunchServiceRolePolicy - 更新的策略	Amazon EC2 对此策略进行了更新，增加了可让 EC2 Fast Launch 创建和管理事件规则的 Amazon EventBridge 访问权限。此外，EC2 Fast Launch 现在还可以描述 Amazon CloudFormation 堆栈，利用关联到 Amazon License Manager 的 AMI 启动实例，获取由其创建并且可以停用的 Amazon KMS 授权列表，以及删除由其创建的启动模板。	2025 年 5 月 14 日
AWSEC2CapacityReservationFleetRolePolicy – 更新了权限	Amazon EC2 更新了 `AWSEC2CapacityReservationFleetRolePolicy` 托管式策略，使用 `ArnLike` 条件运算符，而不是 `StringLike` 条件运算符。	2025 年 3 月 3 日
AmazonEC2ReadOnlyAccess：添加了权限	Amazon EC2 添加了一项权限，允许使用 `GetSecurityGroupsForVpc` 操作检索安全组。	2024 年 12 月 27 日

更改	描述	日期
EC2FastLaunchFullAccess：新策略	Amazon EC2 添加了此策略，以便在实例中执行与 EC2 Fast Launch 功能相关的 API 操作。对从启用了 EC2 Fast Launch 的 AMI 启动的实例，该策略可以附加到其实例配置文件。	2024 年 5 月 14 日
AWSEC2VssSnapshotPolicy：新策略	Amazon EC2 添加了 `AWSEC2VssSnapshotPolicy` 策略，该策略包含创建标签并将标签添加到亚马逊机器映像（AMI）和 EBS 快照的权限。	2024 年 3 月 28 日
Ec2InstanceConnectEndpoint：新策略	Amazon EC2 添加了 `Ec2InstanceConnectEndpoint` 策略。此策略附加到 AWSServiceRoleForEC2InstanceConnect 服务相关角色，以允许 Amazon EC2 在您创建 EC2 Instance Connect Endpoint 时代表您执行操作。	2023 年 1 月 24 日
EC2FastLaunchServiceRolePolicy：新策略	Amazon EC2 添加了 EC2 Fast Launch 功能，如此一来，只需创建一组预置快照，便能让 Windows AMI 更快地启动实例。	2021 年 11 月 26 日
Amazon EC2 开始跟踪更改	Amazon EC2 已开始跟踪 Amazon 托管式策略的更改	2021 年 3 月 1 日

适用于 Amazon EC2 的 IAM 角色

应用程序必须通过 Amazon 凭证签署其 API 请求。因此，如果您是应用程序开发人员，您需要一个策略来为 EC2 实例上运行的应用程序管理凭证。例如，您可以安全地将您的 Amazon 凭证分配至实例，从而允许这些实例上的应用程序使用您的凭证签署请求，并保护您的凭证不被其他用户使用。但是，要将凭证安全地分配至每个实例是有难度的，尤其是 Amazon 以您的名义创建的实例，例如竞价型实例或自动扩缩组中的实例。当您轮换 Amazon 凭证时，您还必须能够更新每项实例上的凭证。

我们设计了 IAM 角色，以便您的应用程序能够安全地从实例发出 API 请求，而无需管理应用程序使用的安全凭证。您可以使用 IAM 角色委托授权以发出 API 请求，而不用创建并分配您的 Amazon 凭证，如下所示：

1. 创建一个 IAM 角色。
2. 定义能够担任此角色的账户或 Amazon 服务。
3. 定义担任角色后应用程序可以使用的 API 操作和资源。
4. 在您启动实例时指定角色，或者将角色附加到现有实例。
5. 让应用程序检索一组临时证书并使用它们。

例如，您可以使用 IAM 角色为在实例上运行的应用程序授予使用 Amazon S3 中的存储桶所需的权限。您可以通过创建 JSON 格式的策略为 IAM 角色指定权限。这些类似于您为 用户创建的策略。如果您更改了某个角色，系统会将此更改传播到所有实例。

> **Note**
>
> Amazon EC2 IAM 角色凭证不受角色中配置的会话持续时间上限的限制。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

在创建 IAM 角色时，请关联最低权限 IAM policies，这些策略将限制对应用程序所需的特定 API 调用的访问权限。对于 Windows 到 Windows 的通信，请使用明确定义且良好记录的 Windows 组和角色以授予 Windows 实例之间的应用程序级访问权限。组和角色允许客户定义最低权限的应用程序和 NTFS 文件夹级别的权限，从而限制对应用程序特定的要求的访问。

您只能将一个 IAM 角色附加到实例，但可以将同一角色附加到多个实例。有关创建和使用 IAM 角色的更多信息，请参阅 IAM 用户指南 中的角色。

您可以将资源级权限应用到您的 IAM 策略，以便控制用户为一个实例附加、替换或分离 IAM 角色的能力。有关更多信息，请参阅 Amazon EC2 API 操作支持的资源级权限 以及以下示例：示例：使用 IAM 角色。

目录

- 实例配置文件
- 使用案例的权限
- 从实例元数据中检索安全凭证
- 授予将 IAM 角色附加到实例的权限
- 将 IAM 角色附加到实例
- Amazon EC2 实例的实例身份角色

实例配置文件

Amazon EC2 使用实例配置文件 作为 IAM 角色的容器。使用 IAM 控制台创建 IAM 角色时，控制台自动创建实例配置文件，按相应的角色为文件命名。如果您使用 Amazon EC2 控制台启动一个带 IAM 角色的实例或将一个 IAM 角色附加到实例，则请根据实例配置文件名称列表选择角色。

如果您使用 Amazon CLI、API 或 Amazon 开发工具包创建角色，则是以单独操作的形式创建角色和实例配置文件，可以为它们提供不同的名称。如果您使用 Amazon CLI、API 或 Amazon 开发工具包启动带有 IAM 角色的实例，或将 IAM 角色附加到实例，请指定实例配置文件名称。

一个实例配置文件只能包含一个 IAM 角色。您可以将一个 IAM 角色包含在多个实例配置文件中。

要更新实例的权限，请替换其实例配置文件。我们不建议从实例配置文件中移除角色，因为此更改最长需要一小时才会生效。

有关更多信息，请参阅《IAM 用户指南》中的使用实例配置文件。

使用案例的权限

首次为应用程序创建 IAM 角色时，有时授予的权限可能超出所需权限。在生产环境中启动应用程序之前，您可以根据 IAM 角色的访问活动生成 IAM 策略。IAM 访问分析器会查看您的 Amazon CloudTrail 日志并生成一个策略模板，其中包含角色在指定日期范围内使用的权限。您可以使用模板创建具有精细权限的托管策略，然后将其附加到 IAM 角色。这样，您仅需授予角色与特定使用案例中的 Amazon 资源进行交互所需的权限。这可帮助您遵循授予最低权限的最佳实践。有关更多信息，请参阅《IAM 用户指南》中的 IAM Access Analyzer 策略生成。

从实例元数据中检索安全凭证

实例上的应用程序通过实例元数据条目 `iam/security-credentials/role-name` 检索角色提供的安全证书。该应用程序具有使用您通过与角色关联的安全凭证为其定义的操作和资源的权限。这些安全凭证是临时的，我们会自动更换它们。我们会在旧凭证过期前至少五分钟提供可用的新凭证。

有关实例元数据的更多信息，请参阅 使用实例元数据管理 EC2 实例。

> **Warning**
>
> 如果您使用的服务采用了带有 IAM 角色的实例元数据，请确保服务代表您进行 HTTP 调用时不会泄露您的凭证。可能泄露您的凭证的服务类型包括 HTTP 代理、HTML/CSS 验证程序服务和支持 XML 包含的 XML 处理程序。

对于您的 Amazon EC2 工作负载，我们建议您使用下述方法检索会话凭证。借助这些凭证，您的工作负载能够发出 Amazon API 请求，而无需使用 `sts:AssumeRole` 来代入已与实例关联的相同角色。除非您需要为基于属性的访问权限控制 (ABAC) 传递会话标签或传递会话策略以进一步限制角色的权限，否则您无需调用角色代入，因为此操作会创建一组新的相同临时角色会话凭证。

如果您的工作负载使用某个角色代入自己，则必须创建明确允许该角色代入自己的信任策略。如果您没有创建信任策略，您会收到 `AccessDenied` 错误。有关更多信息，请参阅《IAM 用户指南》中更新角色信任策略。

IMDSv2

Linux

从 Linux 实例运行以下命令，检索某个 IAM 角色的安全凭证。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

从 Windows 实例运行以下 cmdlet，检索某个 IAM 角色的安全凭证。

```
[string]$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
```

```
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/iam/security-
credentials/role-name
```

IMDSv1

Linux

从 Linux 实例运行以下命令，检索某个 IAM 角色的安全凭证。

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Windows

从 Windows 实例运行以下 cmdlet，检索某个 IAM 角色的安全凭证。

```
Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/iam/security-
credentials/role-name
```

下面是示例输出。如果您无法检索安全凭证，请参阅《IAM 用户指南》中的我无法访问我的 EC2 实例的临时安全凭证。

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

对于实例上运行的应用程序、Amazon CLI 和 Tools for Windows PowerShell 命令，您无需显式获取临时安全凭证 – Amazon开发工具包、Amazon CLI 和 Windows PowerShell 工具会自动从 EC2 实例元数据服务中获取凭证并使用这些凭证。要使用临时安全凭证在实例外部发出调用 (例如，为了测试 IAM 策略)，您必须提供访问密钥、私有密钥和会话令牌。有关更多信息，请参阅 IAM 用户指南中的使用临时安全凭证以请求对 Amazon 资源的访问权限。

授予将 IAM 角色附加到实例的权限

您的 Amazon Web Services 账户中的身份（例如 IAM 用户）必须具有特定权限才能使用 IAM 角色启动 Amazon EC2 实例、将 IAM 角色附加到实例、替换实例的 IAM 角色或从实例分离 IAM 角色。您必须根据需要授予使用以下 API 操作的权限：

- `iam:PassRole`
- `ec2:AssociateIamInstanceProfile`
- `ec2:DisassociateIamInstanceProfile`
- `ec2:ReplaceIamInstanceProfileAssociation`

> **Note**
>
> 如果您将 `iam:PassRole` 的资源指定为 `*`，这将授予将任何 IAM 角色传递给实例的访问权限。要遵循最低权限的最佳实践，请使用 `iam:PassRole` 指定特定 IAM 角色的 ARN，如以下示例策略所示。

编程访问策略示例

以下 IAM 策略授予使用 IAM 角色启动实例、将 IAM 角色附加到实例或者使用 Amazon CLI 或 Amazon EC2 API 替换实例的 IAM 角色的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances",
         "ec2:AssociateIamInstanceProfile",
         "ec2:DisassociateIamInstanceProfile",
         "ec2:ReplaceIamInstanceProfileAssociation"
      ],
      "Resource": "*"
    },
    {
```

```
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/DevTeam*"
    }
  ]
}
```

控制台访问权限的其他要求

要授予使用 Amazon EC2 控制台完成相同任务的权限，您还必须包含 `iam:ListInstanceProfiles` API 操作。

将 IAM 角色附加到实例

在启动过程中或启动之后，您可以创建一个 IAM 角色并将其附加到实例。您也可以替换或分离 IAM 角色。

在实例启动期间创建和附加 IAM 角色（推荐）

1. 在 EC2 实例启动期间，展开高级详细信息。
2. 在 IAM 实例配置文件部分中，选择创建新的 IAM 角色。
3. 将打开内联角色创建表单，您可以：
   - 指定角色名称（例如 `EC2-S3-Access-Role`）
   - 通过选择 Amazon 托管式策略或创建自定义策略来为实例定义权限

     例如，要授予 S3 访问权限，请选择 `AmazonS3ReadOnlyAccess` 托管式策略
   - 查看允许 `ec2.amazonaws.com` 代入该角色的信任策略
   - 为元数据添加可选标签
4. 选择创建角色。

   系统会自动选择新创建的角色，并在实例启动时通过实例配置文件将其附加到您的实例。

> **Note**
>
> 在实例启动期间使用控制台创建角色时，系统会自动创建一个与该角色同名的实例配置文件。实例配置文件是一个容器，用于在启动时将 IAM 角色信息传递给实例。

> **Important**
>
> - 您只能将一个 IAM 角色附加到一个实例，但可以将同一角色附加到多个实例。
> - 关联具有最低权限的 IAM 策略，从而仅允许访问应用程序所需的特定 API 调用。

有关创建和使用 IAM 角色的更多信息，请参阅 IAM 用户指南 中的角色。

在实例启动期间附加现有的 IAM 角色

要在启动时使用 Amazon EC2 控制台将现有的 IAM 角色附加到实例，请展开高级详细信息。对于 IAM 实例配置文件，请从下拉列表中选择该 IAM 角色。

> **Note**
>
> 如果您是使用 IAM 控制台创建的 IAM 角色，则为您创建了实例配置文件，并提供了与角色相同的名称。如果使用 Amazon CLI、API 或 Amazon SDK 创建了 IAM 角色，则可能为实例配置文件指定了与角色不同的名称。

您可以将 IAM 角色附加到正在运行或已停止的实例。如果实例已附加 IAM 角色，则必须将其替换为新的 IAM 角色。

Console

将 IAM 角色附加到实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 依次选择 Actions（操作）、Security（安全）和 Modify IAM role（修改 IAM 角色）。
5. 对于 IAM 角色，选择 IAM 实例配置文件。
6. 选择更新 IAM 角色。

Amazon CLI

将 IAM 角色附加到实例

使用 associate-iam-instance-profile 命令将 IAM 角色附加到该实例。指定实例配置文件时，您可以使用实例配置文件的 Amazon 资源名称（ARN），也可以使用其名称。

```
aws ec2 associate-iam-instance-profile \
    --instance-id i-1234567890abcdef0 \
    --iam-instance-profile Name="TestRole-1"
```

PowerShell

将 IAM 角色附加到实例

使用 Register-EC2IamInstanceProfile cmdlet。

```
Register-EC2IamInstanceProfile `
    -InstanceId i-1234567890abcdef0 `
    -IamInstanceProfile_Name TestRole-1
```

要在已附加了 IAM 角色的实例上替换该 IAM 角色，该实例必须处于正在运行状态。如果要更改实例的 IAM 角色而先不分离现有角色，则您可以执行此操作。例如，您可以执行此操作，以确保正在实例上运行的应用程序所执行的 API 操作不会被中断。

Console

替换实例的 IAM 角色

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 依次选择 Actions（操作）、Security（安全）和 Modify IAM role（修改 IAM 角色）。
5. 对于 IAM 角色，选择 IAM 实例配置文件。
6. 选择更新 IAM 角色。

Amazon CLI

替换实例的 IAM 角色

1. 需要时使用 describe-iam-instance-profile-associations 命令来获取关联 ID。

```
aws ec2 describe-iam-instance-profile-associations \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. 使用 replace-iam-instance-profile-association 命令。指定现有实例配置文件的关联 ID，以及新实例配置文件的 ARN 或名称。

```
aws ec2 replace-iam-instance-profile-association \
    --association-id iip-assoc-0044d817db6c0a4ba \
    --iam-instance-profile Name="TestRole-2"
```

PowerShell

替换实例的 IAM 角色

1. 需要时使用 Get-EC2IamInstanceProfileAssociation cmdlet 来获取关联 ID。

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. 使用 Set-EC2IamInstanceProfileAssociation cmdlet。指定现有实例配置文件的关联 ID，以及新实例配置文件的 ARN 或名称。

```
Set-EC2IamInstanceProfileAssociation `
    -AssociationId iip-assoc-0044d817db6c0a4ba `
    -IamInstanceProfile_Name TestRole-2
```

您可以将 IAM 角色从正在运行或已停止的实例上分离。

Console

从实例中分离 IAM 角色

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 依次选择 Actions（操作）、Security（安全）和 Modify IAM role（修改 IAM 角色）。

5. 对于 IAM 角色，请选择无 IAM 角色。
6. 选择更新 IAM 角色。
7. 提示进行确认时，输入分离，然后选择分离。

Amazon CLI

从实例中分离 IAM 角色

1. 需要时使用 describe-iam-instance-profile-associations，来获取要替换的 IAM 实例配置文件的关联 ID。

```
aws ec2 describe-iam-instance-profile-associations \
    --filters Name=instance-id,Values=i-1234567890abcdef0 \
    --query IamInstanceProfileAssociations.AssociationId
```

2. 使用 disassociate-iam-instance-profile 命令。

```
aws ec2 disassociate-iam-instance-profile --association-id iip-
assoc-0044d817db6c0a4ba
```

PowerShell

从实例中分离 IAM 角色

1. 需要时使用 Get-EC2IamInstanceProfileAssociation，来获取要分离的 IAM 实例配置文件的关联 ID。

```
(Get-EC2IamInstanceProfileAssociation -Filter @{Name="instance-id";
 Values="i-0636508011d8e966a"}).AssociationId
```

2. 使用 Unregister-EC2IamInstanceProfile cmdlet。

```
Unregister-EC2IamInstanceProfile -AssociationId iip-assoc-0044d817db6c0a4ba
```

Amazon EC2 实例的实例身份角色

您启动的每个 Amazon EC2 实例都有一个代表其身份的实例身份角色。实例身份角色是一种 IAM 角色。集成使用实例身份角色的 Amazon 服务和功能可以使用它来标识服务的实例。

实例身份角色凭证可通过 /identity-credentials/ec2/security-credentials/ec2-instance 的实例元数据服务（IMDS）访问。凭证由 Amazon 临时访问密钥对和会话令牌组成。它们用于对使用实例身份角色的 Amazon 服务签署 Amazon Sigv4 请求。无论实例上是否启用了使用实例身份角色的服务或功能，凭证都存在于实例元数据中。

实例身份角色会在实例启动时自动创建，没有 role-trust 策略文档，并且不受任何身份或资源策略的约束。

支持的服务

以下 Amazon 服务使用实例身份角色：

- Amazon EC2：EC2 Instance Connect 使用实例身份角色来更新 Linux 实例的主机密钥。
- Amazon GuardDuty：GuardDuty 运行时监控使用实例身份角色，允许运行时代理向 GuardDuty VPC 端点发送安全遥测数据。
- Amazon Lambda：Lambda 托管实例使用实例身份角色进行生命周期挂钩、遥测和构件分发。
- Amazon Security Token Service（Amazon STS）– 实例身份角色凭证可与 Amazon STS GetCallerIdentity 操作结合使用。
- Amazon Systems Manager – 使用默认主机管理配置时，Amazon Systems Manager 将使用实例身份角色提供的身份来注册 EC2 实例。标识实例后，Systems Manager 可将 AWSSystemsManagerDefaultEC2InstanceManagementRole IAM 角色传递给实例。

实例身份角色不能与其他 Amazon 服务或功能结合使用，因为其并未与实例身份角色集成。

实例身份角色 ARN

实例身份角色 ARN 采用以下格式：

```
arn:aws-partition:iam::account-number:assumed-role/aws:ec2-instance/instance-id
```

例如：

```
arn:aws:iam::0123456789012:assumed-role/aws:ec2-instance/i-1234567890abcdef0
```

有关 ARN 的更多信息，请参阅《IAM 用户指南》中的 Amazon 资源名称（ARN）。

Amazon EC2 实例的更新管理

我们建议您定期修补、更新和保护 EC2 实例上的操作系统和应用程序。您可以使用 Amazon Systems Manager Patch Manager 自动执行为操作系统和应用程序安装安全相关更新的过程。

对于自动扩缩组中的 EC2 实例，您可以使用 AWS-PatchAsgInstance 运行手册以避免正在修补的实例被替换。或者，您也可以使用任何自动更新服务或建议的过程安装应用程序供应商提供的更新。

资源

- AL2023：《Amazon Linux 2023 User Guide》中的 Updating AL2023
- AL2：《Amazon Linux 2 User Guide》中的 Manage software on your Amazon Linux 2 instance
- Windows 实例：the section called “更新管理”

Windows 实例的安全最佳实践

建议您对 Windows 实例遵循以下安全最佳实践。

内容

- 高级安全最佳实践
- 更新管理
- 配置管理
- 变更管理
- Amazon EC2 Windows 实例的审计和问责

高级安全最佳实践

对于 Windows 实例，您应遵循以下高级安全最佳实践：

- 最小访问权限：仅授予对受信任的和预期的系统和位置的访问权限。这适用于所有 Microsoft 产品，例如 Active Directory、Microsoft 业务生产力服务器以及基础设施服务（例如远程桌面服务、反向代理服务器、IIS Web 服务器等）。使用 Amazon 功能（例如 Amazon EC2 实例安全组、网络访问控制列表 (ACL) 和 Amazon VPC 公有/私有子网）跨架构中的多个位置对安全性进行分层。在 Windows 实例中，客户可以使用 Windows 防火墙在其部署中进一步对深度防御策略进行分层。只安装系统按设计运行所需的操作系统组件和应用程序。将 IIS 等基础设施服务配置为在服务账户下运行，或使用应用程序池身份等功能以跨基础设施本地和远程访问资源。

- 最低权限：确定实例和账户执行其功能所需的最低权限集。限制服务器和用户以仅允许这些已定义的权限。使用基于角色的访问控制等技术来减小管理账户的表面面积，并创建最受限的角色来完成任务。使用 NTFS 中的加密文件系统（EFS）等操作系统功能来对敏感数据进行静态加密，并控制应用程序和用户对该数据的访问。
- 配置管理：创建一个基准服务器配置，其中包含最新的安全修补程序和基于主机的保护套件，其中包括防病毒、防恶意软件、入侵检测/防护和文件完整性监控。根据当前记录的基准评测每个服务器，以确定和标记任何偏差。确保将每个服务器配置为生成和安全存储适当的日志和审计数据。
- 更改管理：创建流程来控制对服务器配置基准进行的更改，并致力于完全自动化的更改流程。此外，通过将 Just Enough Administration (JEA) 与 Windows PowerShell DSC 结合使用来限制对最少所需功能的管理访问。
- 补丁管理：实施定期安装补丁、更新和保护 EC2 实例上的操作系统和应用程序的流程。
- 审计日志：审计访问权限和对 Amazon EC2 实例进行的所有更改，以验证服务器完整性并确保仅进行授权更改。利用 IIS 增强日志记录等功能来增强默认日志记录功能。Amazon 功能（例如 VPC 流日志和 Amazon CloudTrail）也可用于审计网络访问权限，包括允许/拒绝的请求和 API 调用。

更新管理

为确保在 Amazon EC2 上运行 Windows Server 时的最佳结果，我们建议您实施以下最佳实践：

- Configure Windows Update
- Update drivers
- Use the latest Windows AMIs
- Test performance before migration
- Update launch agents
- 在安装更新之后重启 Windows 实例。有关更多信息，请参阅 重启 Amazon EC2 实例。

有关如何将 Windows 实例升级或迁移到 Windows Server 新版本的信息，请参阅将 EC2 Windows 实例升级到更高版本的 Windows Server。

配置 Windows 更新

默认情况下，从 Amazon Windows Server AMI 启动的实例不会通过 Windows 更新接收更新。

更新 Windows 驱动程序

维护所有 Windows EC2 实例上的最新驱动程序，以确保在您的实例集中应用最新的问题修复和性能增强。您需要更新 Amazon PV、Amazon ENA 和 Amazon NVMe 驱动程序，具体取决于您的实例类型。

- 使用 Amazon Systems Manager Automation 运行手册 AWSSupport-UpgradeWindowsAWSDrivers 跨实例中轻松应用更新。

使用最新 Windows AMI 启动实例

Amazon 每月发布新的 Windows AMI，其中包含最新的操作系统补丁、驱动程序和启动代理。您应该在启动新实例或构建自己的自定义映像时利用最新的 AMI。

- 要查看每个版本的 Amazon Windows AMI 的更新，请参阅 Amazon Windows AMI 版本历史记录。
- 要使用最新 AMI 进行构建，请参阅使用 Systems Manager Parameter Store 查询最新的 Windows AMI。
- 有关可用于启动数据库实例的专用 Windows AMI 以及合规性强化用例的更多信息，请参阅《Amazon Windows AMI Reference》中的 Specialized Windows AMIs。

在迁移前测试系统/应用程序性能

将企业应用程序迁移到 Amazon 可能涉及很多变量和配置。始终对 EC2 解决方案进行性能测试，以确保：

- 已正确配置实例类型，包括实例大小、增强联网和租赁（共享或专用）。
- 实例拓扑适用于工作负载，并在必要时利用高性能功能（如专用租赁、放置组、实例存储卷和裸机）。

安装最新版本的 EC2Launch v2

安装最新的 EC2Launch v2 代理，以确保在您的实例集中应用最新的增强。有关更多信息，请参阅 安装 EC2Launch v2。

如果有混合实例集，或者要继续使用 EC2Launch（Windows Server 2016 和 2019）或 EC2 Config（仅限旧式操作系统）代理，请更新到相应代理的最新版本。

以下 Windows Server 版本和启动代理的组合支持自动更新。您可以在 Amazon EC2 启动代理下的 SSM 快速设置主机管理控制台中选择自动更新。

Windows 版本	EC2Launch v1	EC2Launch v2
2016	✓	✓
2019	✓	✓
2022		✓

- 有关更新到 EC2Launch v2 的更多信息，请参阅 the section called “安装 EC2Launch v2”。
- 有关手动更新 EC2Config 的信息，请参阅 the section called “安装 EC2Config”。

配置管理

亚马逊机器映像（AMI）为 Amazon EC2 实例提供了初始配置，其中包括 Windows OS 和可选的客户特定的自定义项，例如应用程序和安全控制。创建包含自定义安全配置基准的 AMI 目录可确保使用标准安全控制启动所有 Windows 实例。可以将安全基准烘焙成 AMI，在启动 EC2 实例时动态引导它，或将它打包为产品，以便通过 Amazon Service Catalog 产品组合进行统一分发。有关保护 AMI 的更多信息，请参阅构建 AMI 的最佳实践。

每个 Amazon EC2 实例都应遵循组织安全标准。请勿安装任何不需要的 Windows 角色和功能，并安装软件来抵御恶意代码（防病毒软件、防恶意软件、利用缓解）、监控主机完整性并执行入侵检测。配置安全软件来监控和维护 OS 安全设置，保护关键 OS 文件的完整性，并在偏离安全基准时发出警报。考虑实施由 Microsoft、互联网安全中心 (CIS) 或美国国家标准与技术研究院 (NIST) 发布的建议安全配置基准。考虑对特定的应用程序服务器使用其他 Microsoft 工具，例如 SQL Server 的最佳实践分析器。

Amazon 客户还可以运行 Amazon Inspector 评估，以改进 Amazon EC2 实例上部署的应用程序的安全性和合规性。Amazon Inspector 将自动评估应用程序以查看是否存在漏洞或偏离最佳实践，并且包含一个知识库，其中包含映射到常见安全合规性标准（例如，PCI DSS）和漏洞定义的几百条规则。内置规则的示例包括检查是否启用了远程根登录或是否安装了易受攻击的软件版本。Amazon 安全研究员会定期更新这些规则。

在保护 Windows 实例时，建议您实施 Active Directory 域服务，以便为分布式位置启用可扩展的、安全的且可管理的基础设施。此外，在通过 Amazon EC2 控制台或使用 Amazon EC2 预配置工具（例如 Amazon CloudFormation）启动实例后，最好是使用本机操作系统功能（例如 Microsoft Windows PowerShell DSC），以便在发生配置偏差时维护配置状态。

变更管理

在启动时将初始安全基准应用于 Amazon EC2 实例后，可以控制正在进行的 Amazon EC2 更改来维护虚拟机的安全性。建立变更管理流程以授权和合并对 Amazon 资源（例如，安全组、路由表和网络 ACL）以及操作系统和应用程序配置（例如 Windows 或应用程序修补、软件升级或配置文件更新）进行的更改。

Amazon 提供了多种工具来帮助管理对 Amazon 资源（包括 Amazon CloudTrail、Amazon Config、Amazon CloudFormation 和 Amazon Elastic Beanstalk 以及 Systems Center Operations Manager 和 System Center Virtual Machine Manager 的管理包）进行的更改。请注意，Microsoft 会在每月的第二个星期二（或根据需要）发布 Windows 补丁，而 Amazon 将在 Microsoft 发布补丁后五天内更新所有由 Amazon 管理的 Windows AMI。因此，必须持续修补所有基准 AMI、使用最新的 AMI ID 更新 Amazon CloudFormation 模板和自动扩缩组配置，并实施工具来自动实施运行的实例修补管理。

Microsoft 提供了几个用于管理 Windows 操作系统和应用程序更改的选项。例如，SCCM 提供了环境修改的完整生命周期覆盖。选择相应工具，以便满足业务需求并控制更改如何影响应用程序 SLA、容量、安全性和灾难恢复过程。避免手动更改，并改用自动化配置管理软件或命令行工具（例如 EC2 Run Command 或 Windows PowerShell）来实施脚本化的、可重复的变更流程。要帮助达到此要求，请使用带增强日志记录的堡垒主机来进行与 Windows 实例的所有交互，以确保自动记录所有事件和任务。

Amazon EC2 Windows 实例的审计和问责

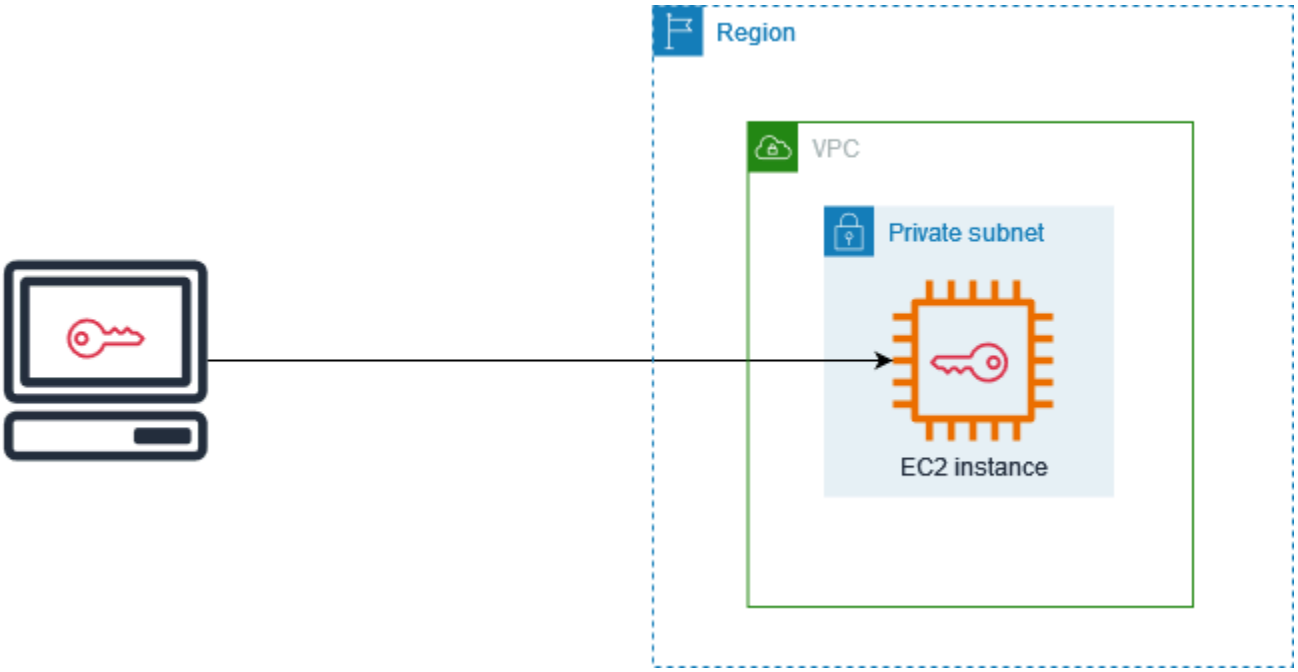
Amazon CloudTrail、Amazon Config 和 Amazon Config 规则 提供了审计和更改跟踪功能来审计 Amazon 资源更改。将 Windows 事件日志配置为将本地日志文件发送到集中式日志管理系统，以便保留日志数据来进行安全和操作行为分析。Microsoft System Center Operations Manager (SCOM) 聚合有关部署到 Windows 实例的 Microsoft 应用程序的信息，并根据应用程序角色和服务来应用预配置的规则集和自定义规则集。System Center Management Packs 基于 SCOM 而构建，旨在提供特定于应用程序的监控和配置准则。这些管理包支持 Windows Server Active Directory、SharePoint Server 2013、Exchange Server 2013、Lync Server 2013、SQL Server 2014 和许多其他的服务器和技术。

除了 Microsoft 系统管理工具之外，客户还可以使用 Amazon CloudWatch 监控实例 CPU 利用率、磁盘性能、网络 I/O 以及执行主机和实例状态检查。EC2Config、EC2Launch 和 EC2Launch v2 启动代理提供对 Windows 实例的其他高级功能的访问。例如，它们可以将 Windows 系统、安全性、应用程序和 Internet Information Services (IIS) 日志导出到 CloudWatch Logs，这些日志随后可与 Amazon CloudWatch 指标和警报集成。此外，客户可以创建脚本来将 Windows 性能计数器导出到 Amazon CloudWatch 自定义指标。

Amazon EC2 密钥对和 Amazon EC2 实例

密钥对（由公有密钥和私有密钥组成）是一组安全凭证，可在连接到 Amazon EC2 实例时用来证明您的身份。对于 Linux 实例，私有密钥可使您安全地 SSH 到您的实例。对于 Windows 实例，需要用私有密钥才能解密管理员密码，然后您可以使用该密码连接到您的实例。

Amazon EC2 在您的实例上存储公有密钥，您需要存储私有密钥，如下图所示。重要的是，您要将您的私有密钥存储在一个安全的位置，因为拥有您的私有密钥的任何人都可以连接到使用密钥对的实例。



启动实例时，您可以指定密钥对，这样便可使用需要密钥对的方法连接到您的实例。根据安全管理方式的不同，您可以为所有实例指定相同的密钥对，也可以指定不同的密钥对。

对于 Linux 实例，实例首次启动时，您在启动时指定的公有密钥将放在 Linux 实例 `~/.ssh/authorized_keys` 内的条目中。在使用 SSH 连接到 Linux 实例时，要进行登录，您必须指定与公有密钥对应的私有密钥。

有关连接到 EC2 实例的更多信息，请参阅 连接到您的 EC2 实例。

> **Important**
>
> 由于 Amazon EC2 不保存私有密钥的副本，因此，如果您丢失私有密钥，则无法恢复它。但是，仍可通过一种方法连接到丢失了密钥对的实例。有关更多信息，请参阅 我丢失了私有密钥。如何连接到我的实例？。

作为密钥对的替代，您可以使用 Amazon Systems Manager Session Manager，通过基于浏览器的交互式、一键式 Shell 或 Amazon Command Line Interface（Amazon CLI）连接到您的实例。

内容

- 为您的 Amazon EC2 实例创建密钥对
- 描述密钥对
- 删除您的密钥对
- 在 Linux 实例上添加或更换公有密钥
- 验证您的密钥对的指纹

为您的 Amazon EC2 实例创建密钥对

您可以使用 Amazon EC2 创建密钥对，或者可以使用第三方工具创建密钥对，然后将它们导入 Amazon EC2。

Amazon EC2 支持适用于 Linux 和 Windows 实例的 2048 位 SSH-2 RSA 密钥。Amazon EC2 还支持 Linux 实例的 ED25519 密钥。

有关在创建密钥对后如何连接到实例的说明，请参阅the section called “使用 SSH 连接到 Linux 实例”和the section called “使用 RDP 连接到 Windows 实例”。

内容

- 使用 Amazon EC2 创建密钥对
- 使用 Amazon CloudFormation 创建密钥对
- 使用第三方工具创建密钥对，并将公有密钥导入 Amazon EC2

使用 Amazon EC2 创建密钥对

当您使用 Amazon EC2 创建密钥对时，公有密钥将存储在 Amazon EC2 中，您需要存储私有密钥。

您最多可以为每个区域创建 5000 个密钥对。要申请增加，请创建支持案例。有关更多信息，请参阅 Amazon Web Services 支持 用户指南中的创建支持案例。

Console

要使用 Amazon EC2 创建密钥对

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中的 Network & Security 下，选择 Key Pairs。
3. 选择 Create key pair (创建密钥对)。
4. 对于 Name (名称)，为模板输入一个描述性名称。Amazon EC2 将公有密钥与您指定的密钥名称相关联。密钥名称最多可包含 255 个 ASCII 字符。它不能包含前导空格或尾随空格。
5. 选择适合您操作系统的密钥对类型：

   （Linux 实例）对于密钥对类型，选择 RSA 或者 ED25519。

   （Windows 实例）对于密钥对类型，请选择 RSA。Windows 实例不支持 ED25519 密钥。
6. 对于 Private key file format（私有密钥文件格式），选择要保存私有密钥的格式。要以可与 OpenSSH 一起使用的格式保存私有密钥，请选择 pem。要以可与 PuTTY 一起使用的格式保存私有密钥，请选择 ppk。
7. 要添加标签到公有密钥，请选择 Add tag（添加标签），然后输入标签的键和值。对每个标签重复此操作。
8. 选择 Create key pair (创建密钥对)。
9. 您的浏览器会自动下载私有密钥文件。基本文件名是指定为密钥对名称的名称，文件扩展名由您选择的文件格式确定。将私有密钥文件保存在安全位置。

   > **Important**
   >
   > 这是您保存私有密钥文件的唯一机会。

10. 如果您计划在 macOS 或 Linux 计算机上使用 SSH 客户端连接到 Linux 实例，请使用以下命令设置私有密钥文件的权限，以确保只有您可以读取该文件。

    ```
    chmod 400 key-pair-name.pem
    ```

    如果不设置这些权限，则无法使用此密钥对连接到实例。有关更多信息，请参阅 错误：未保护的私有密钥文件。

Amazon CLI

要使用 Amazon EC2 创建密钥对

1. 按如下方式使用 create-key-pair 命令生成密钥对，并将私有密钥保存到 `.pem` 文件中。`--query` 选项会将私有密钥材料打印到输出中。`--output` 选项会将私有密钥材料保存到指定的

文件中。扩展名应为 .pem 或 .ppk，具体取决于密钥格式。私有密钥名称可以与公有密钥名称不同，但为方便起见，建议使用相同的名称。

```
aws ec2 create-key-pair \
    --key-name my-key-pair \
    --key-type rsa \
    --key-format pem \
    --query "KeyMaterial" \
    --output text > my-key-pair.pem
```

2. 如果您计划在 macOS 或 Linux 计算机上使用 SSH 客户端连接到 Linux 实例，请使用以下命令设置私有密钥文件的权限，以确保只有您可以读取该文件。

```
chmod 400 key-pair-name.pem
```

如果不设置这些权限，则无法使用此密钥对连接到实例。有关更多信息，请参阅 错误：未保护的私有密钥文件。

PowerShell

要使用 Amazon EC2 创建密钥对

按如下所示使用 New-EC2KeyPair cmdlet 生成密钥，并将其保存到 .pem 或 .ppk 文件中。Out-File cmdlet 会将私有密钥材料保存到具有指定扩展名的文件中。扩展名应为 .pem 或 .ppk，具体取决于密钥格式。私有密钥名称可以与公有密钥名称不同，但为方便起见，建议使用相同的名称。

```
(New-EC2KeyPair `
    -KeyName "my-key-pair" `
    -KeyType "rsa" `
    -KeyFormat "pem").KeyMaterial | Out-File -Encoding ascii -FilePath C:\path\my-key-pair.pem
```

使用 Amazon CloudFormation 创建密钥对

当您使用 Amazon CloudFormation 创建新的密钥对时，私有密钥会保存到 Amazon Systems Manager Parameter Store。参数名称有以下形式：

```
/ec2/keypair/key_pair_id
```

有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的 Amazon Systems Manager Parameter Store。

使用 Amazon CloudFormation 创建密钥对

1. 在您的模板中指定 AWS::EC2::KeyPair 资源。

```
Resources:
  NewKeyPair:
    Type: 'AWS::EC2::KeyPair'
    Properties:
      KeyName: new-key-pair
```

2. 按如下方式使用 describe-key-pairs 命令获取密钥对的 ID。

```
aws ec2 describe-key-pairs --filters Name=key-name,Values=new-key-pair --query
 KeyPairs[*].KeyPairId --output text
```

下面是示例输出。

```
key-05abb699beEXAMPLE
```

3. 按如下方式使用 get-parameter 命令获取密钥的参数，并将密钥材料保存在 .pem 文件中。

```
aws ssm get-parameter --name /ec2/keypair/key-05abb699beEXAMPLE --with-decryption
 --query Parameter.Value --output text > new-key-pair.pem
```

所需的 IAM 权限

要启用 Amazon CloudFormation 以代表您管理 Parameter Store 参数，由 Amazon CloudFormation 或您用户担任的 IAM 角色必须拥有以下权限：

- `ssm:PutParameter` – 授予权限以创建私有密钥材料的参数。
- `ssm:DeleteParameter` – 授予权限以删除存储私有密钥材料的参数。无论密钥对是由 Amazon CloudFormation 导入还是创建的，都需要此权限。

当 Amazon CloudFormation 删除由堆栈创建或导入的密钥对时，它会执行权限检查，以确定您是否有权删除参数，即使 Amazon CloudFormation 仅在创建密钥对而非导入密钥对时才创建参数。Amazon

CloudFormation 使用与您账户中的任何参数都不匹配的虚构参数名称测试所需的权限。因此，您可能会在 `AccessDeniedException` 错误消息中看到虚构的参数名称。

使用第三方工具创建密钥对，并将公有密钥导入 Amazon EC2

如果不使用 Amazon EC2 创建密钥对，则可以使用第三方工具创建一个 RSA 或 ED25519 密钥对，然后将公有密钥导入 Amazon EC2。

密钥对的要求

- 支持的类型：
  - （Linux 和 Windows）RSA
  - （仅限 Linux）ED25519

    > **Note**
    >
    > Windows 实例不支持 ED25519 密钥。

  - Amazon EC2 不接受 DSA 密钥。
- 支持的格式：
  - OpenSSH 公有密钥格式（对于 Linux，格式为 `~/.ssh/authorized_keys`）。
  - （仅限 Linux）如果您在使用 EC2 Instance Connect API 时使用 SSH 进行连接，则也支持 SSH2 格式。
  - SSH 私有密钥文件格式必须为 PEM 或 PPK
  - （仅 RSA）Base64 编码的 DER 格式
  - （仅 RSA）SSH 公有密钥文件格式如 RFC 4716 所指定
- 支持的长度：
  - 1024、2048 和 4096。
  - （仅限 Linux）如果您在使用 EC2 Instance Connect API 时使用 SSH 进行连接，则支持的长度为 2048 和 4096。

要使用第三方工具创建密钥对

1. 使用您选择的第三方工具生成密钥对。例如，您可以使用 ssh-keygen（通过标准 OpenSSH 安装提供的工具）。也可以使用 Java、Ruby、Python 以及多种其他提供标准库的编程语言来创建密钥对。

> **Important**
>
> 私有密钥必须采用 PEM 或 PPK 格式。例如，使用 `ssh-keygen -m PEM` 生成 PEM 格式的 OpenSSH 密钥。

2. 将公有密钥保存至本地文件。例如，`~/.ssh/my-key-pair.pub`（Linux、macOS）或 `C:\keys\my-key-pair.pub`（Windows）。此文件的文件扩展名并不重要。
3. 将私有密钥保存至扩展名为 `.pem` 或 `.ppk` 的本地文件。例如，`~/.ssh/my-key-pair.pem` 或 `~/.ssh/my-key-pair.ppk`（Linux、macOS）或 `C:\keys\my-key-pair.pem` 或 `C:\keys\my-key-pair.ppk`（Windows）。文件扩展名很重要，因为根据您用来连接实例的工具不同，您将需要特定的文件格式。OpenSSH 需要一个 `.pem` 文件，而 PuTTY 需要一个 `.ppk` 文件。

> **Important**
>
> 将私有密钥文件保存在安全位置。当您启动实例时，您将需要提供公有密钥的名称；当您每次连接到实例时，您将需要提供相应的私有密钥。

创建密钥对后，使用以下方法之一将公有密钥导入到 Amazon EC2。

Console

要将公有密钥导入至 Amazon EC2

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Key Pairs (密钥对)。
3. 选择 Import key pair (导入密钥对)。
4. 对于 Name（名称），为公有密钥输入一个描述性名称。该名称最多可包含 255 个 ASCII 字符。它不能包含前导空格或尾随空格。

> **Note**
>
> 当您从 EC2 控制台连接到实例时，控制台会建议使用此名称作为私有密钥文件的名称。

5. 选择 Browse (浏览) 以导航到您的公有密钥并选择它，或者将公有密钥的内容粘贴到 Public key contents (公有密钥内容) 字段中。

6. 选择 Import key pair (导入密钥对)。
7. 验证您导入的公有密钥是否显示在密钥对列表中。

Amazon CLI

要将公有密钥导入至 Amazon EC2

使用 import-key-pair 命令。

```
aws ec2 import-key-pair \
    --key-name my-key-pair \
    --public-key-material fileb://path/my-key-pair.pub
```

验证密钥对是否已成功导入

使用 describe-key-pairs 命令。

```
aws ec2 describe-key-pairs --key-names my-key-pair
```

PowerShell

要将公有密钥导入至 Amazon EC2

使用 Import-EC2KeyPair cmdlet。

```
$publickey=[Io.File]::ReadAllText("C:\Users\TestUser\.ssh\id_rsa.pub")
Import-EC2KeyPair `
    -KeyName my-key-pair `
    -PublicKey $publickey
```

验证密钥对是否已成功导入

使用 Get-EC2KeyPair cmdlet。

```
Get-EC2KeyPair -KeyName my-key-pair
```

描述密钥对

您可以描述存储在 Amazon EC2 中的密钥对。您还可以检索公有密钥材料并识别启动时指定的公有密钥。

任务

- 描述密钥对
- 检索公有密钥材料
- 确定启动时指定的公有密钥

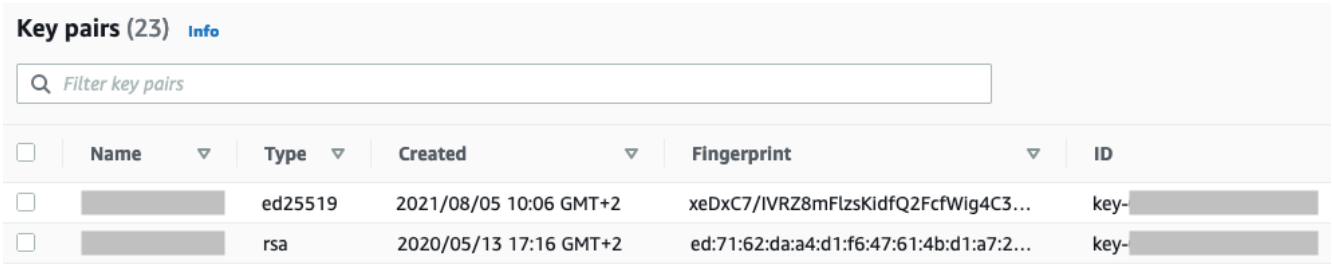
描述密钥对

您可以查看有关存储在 Amazon EC2 中的公有密钥的以下信息：公有密钥名称、ID、密钥类型、指纹、公有密钥材料、Amazon EC2 创建密钥的日期和时间（采用 UTC 时区）（如果密钥是由第三方工具创建的，则是将该密钥导入 Amazon EC2 的日期和时间），以及与该公有密钥相关联的所有标签。

Console

查看有关密钥对的信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航器中，选择 Key Pairs（密钥对）。
3. 您可以在Key pairs（密钥对）表中查看每个公有密钥的相关信息。



4. 要查看公有密钥的标签，请选中密钥旁边的复选框，然后选择操作、管理标签。

Amazon CLI

查看有关密钥对的信息

使用 describe-key-pairs 命令。

```
aws ec2 describe-key-pairs --key-names key-pair-name
```

PowerShell

查看有关密钥对的信息

使用 Get-EC2KeyPair cmdlet。

```
Get-EC2KeyPair -KeyName key-pair-name
```

检索公有密钥材料

您可以获取密钥对的公有密钥材料。以下是一个示例公有密钥。请注意，添加的换行符是为了提高可读性。

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZ
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Private key

使用 ssh-keygen 检索公有密钥材料（Linux）

在本地 Linux 或 macOS 计算机上，使用 ssh-keygen 命令。指定您已在其中下载私有密钥的路径（.pem 文件）。

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

如果此 ssh-keygen 命令运行失败，请运行以下 chmod 命令，确保私有密钥文件具有所需的权限。

```
chmod 400 key-pair-name.pem
```

使用 PuTTYgen 检索公有密钥材料（Windows）

在本地 Windows 计算机上，启动 PuTTYgen。选择 Load（加载）。选择 .ppk 或者 .pem 私有密钥文件。PuTTYgen 在 Public key for pasting into OpenSSH authorized_keys file (粘贴到 OpenSSH authorized_keys 文件的公有密钥) 下方显示公有密钥。也可以通过以下方式查看公有密钥：选择 Save public key (保存公有密钥)，请指定文件的名称，然后打开文件。

Amazon CLI

检索公有密钥材料

使用以下 describe-key-pairs 命令并指定 --include-public-key 选项。

```
aws ec2 describe-key-pairs \
    --key-names key-pair-name \
    --include-public-key \
    --query "KeyPairs[].PublicKey"
```

PowerShell

检索公有密钥材料

使用 Get-EC2KeyPair cmdlet。

```
(Get-EC2KeyPair -KeyName key-pair-name -IncludePublicKey $true).PublicKey
```

IMDSv2

Linux

在 Linux 实例上运行以下命令。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key
```

Windows

在 Windows 实例上运行以下 cmdlet。

```
[string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds"
 = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method GET -Uri
 http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

IMDSv1

Linux

在 Linux 实例上运行以下命令。

```
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

Windows

在 Windows 实例上运行以下 cmdlet。

```
Invoke-RestMethod -uri  http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

确定启动时指定的公有密钥

如果您在启动实例时指定公有密钥，则该实例将记录公有密钥名称。即使更改了实例上的公有密钥或添加了公有密钥，报告的实例公有密钥对名称也不会更改。

Console

识别在实例启动时指定的公有密钥

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 在详细信息选项卡的实例详细信息下，找到启动时分配的密钥对。

Amazon CLI

识别在实例启动时指定的公有密钥

使用以下 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].KeyName" \
    --output text
```

下面是示例输出。

```
key-pair-name
```

PowerShell

识别在实例启动时指定的公有密钥

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-1234567890abcdef0).Instances | Select KeyName
```

下面是示例输出。

```
KeyName
-------
key-pair-name
```

删除您的密钥对

您可以删除密钥对，这将移除存储在 Amazon EC2 中的公有密钥。删除密钥对不会删除匹配的私有密钥。

使用以下方法删除公有密钥时，您仅删除了创建或导入密钥对时在 Amazon EC2 中存储的公有密钥。删除公有密钥并不会从您添加公有密钥的任何实例中删除公有密钥，无论是在启动实例时还是启动后。也不会删除本地计算机上的私有密钥。您可以继续连接到使用已从 Amazon EC2 删除的公有密钥启动的实例，只要您仍然拥有私有密钥（.pem）文件。

> **Important**
>
> 如果您使用的是自动扩缩组（例如，在 Elastic Beanstalk 环境中），请确保您要删除的公有密钥未在关联的启动模板或启动配置中指定。如果 Amazon EC2 Auto Scaling 检测到运行不正常的实例，它将启动替代实例。但是，如果找不到公有密钥，则实例启动失败。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的启动模板。

Console

要在 Amazon EC2 上删除您的公有密钥

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 Key Pairs (密钥对)。
3. 选择要删除的密钥对，然后依次选择 Actions（操作）、Delete（删除）。
4. 在确认字段中，输入 `Delete`，然后选择 Delete (删除)。

Amazon CLI

要在 Amazon EC2 上删除您的公有密钥

使用 delete-key-pair 命令。

```
aws ec2 delete-key-pair --key-name my-key-pair
```

PowerShell

要在 Amazon EC2 上删除您的公有密钥

使用 Remove-EC2KeyPair cmdlet。

```
Remove-EC2KeyPair -KeyName my-key-pair
```

在 Linux 实例上添加或更换公有密钥

如果您丢失了私有密钥，则无法访问任何使用该密钥对的实例。有关使用与您在启动时指定的密钥对不同的密钥对连接到实例的更多信息，请参阅我丢失了私有密钥。

启动实例时，您可以指定密钥对。如果您在启动时指定密钥对，当您的实例首次启动时，公有密钥将放置在 `~/.ssh/authorized_keys` 内的一个条目中的 Linux 实例中。使用 SSH 初次连接到 Linux 实例时，您需要指定默认用户和与存储在 Linux 实例上的公有密钥对应的私有密钥。

连接到实例后，您可以更改用于访问实例默认系统账户的密钥对，方法是在实例上添加新的公有密钥，或者替换实例上的公有密钥（删除现有公有密钥并添加新的公有密钥）。您还可以从实例中删除所有公有密钥。

您可能会出于以下原因添加或替换密钥对：

- 如果组织中有用户需要使用单独的密钥访问系统用户，您可以将此公有密钥添加到实例。

- 如果某人有私有密钥的副本（.pem 文件），而您想要防止他们连接到实例（例如在他们已离开企业的情况下），您可以删除实例上的公有密钥，并将其替换为新的密钥。
- 如果您从实例创建 Linux AMI，则公有密钥材料将从实例复制到 AMI。如果您从 AMI 启动实例，则新实例将包含来自原始实例的公有密钥。要防止拥有私有密钥的用户连接到新实例，您可以在创建 AMI 之前从原始实例中删除公有密钥。

使用以下程序修改默认用户的密钥对，例如 ec2-user。有关向实例添加用户的信息，请参阅实例上的操作系统的文档。

添加或替换密钥对

1. 使用 Amazon EC2 控制台或第三方工具创建新的密钥对。
2. 从新密钥对检索公有密钥。有关更多信息，请参阅 检索公有密钥材料。
3. 连接到您的 实例。
4. 在实例上，使用您选择的文本编辑器，打开 .ssh/authorized_keys 文件。将新密钥对的公有密钥信息粘贴到现有公有密钥信息下，然后保存该文件。
5. 从实例断开连接。测试您是否可以使用新密钥对的私有密钥文件连接到实例。
6. 如果您使用 Auto Scaling、EC2 实例集或启动模板来启动实例，请检查您要替换的密钥对是否在启动模板或启动配置中已指定。如果未指定，实例启动将失败。
7. (可选) 如果您要替换现有密钥对，请连接到实例并从 .ssh/authorized_keys 文件中删除原始密钥对的公有密钥信息。

要从实例中删除公有密钥

1. 连接到您的实例。
2. 使用您选择的文本编辑器，在实例上打开 .ssh/authorized_keys 文件。删除公有密钥信息，然后保存该文件。

> **Warning**
>
> 如果从实例中删除所有公有密钥并断开与实例的连接，除非配置了其他登录方式，否则将无法再次连接该实例。

验证您的密钥对的指纹

要验证您的密钥对的指纹，请将在 Amazon EC2 控制台的密钥对页面上显示的或由 describe-key-pairs 命令返回的指纹与您在本地计算机上使用私有密钥生成的指纹进行比较。这些指纹应该一致。

当 Amazon EC2 计算指纹时，Amazon EC2 可能会用 = 字符在指纹上添加填充。ssh-keygen 等其他工具可能会省略此填充。

如果您尝试验证 Linux EC2 实例的指纹，而不是密钥对的指纹，则请参阅获取实例指纹。

指纹如何计算

Amazon EC2 使用不同的哈希函数来计算 RSA 和 ED25519 密钥对的指纹。此外，对于 RSA 密钥对，Amazon EC2 使用不同的哈希函数以不同方式计算指纹，具体取决于密钥对是由 Amazon EC2 创建还是导入到 Amazon EC2 的。

下表所列的哈希函数用于计算由 Amazon EC2 创建以及导入到 Amazon EC2 的 RSA 和 ED25519 密钥对的指纹。

（Linux 实例）用于计算指纹的哈希函数

密钥对来源	RSA 密钥对（Windows 和 Linux）	ED25519 密钥对（Linux）
由 Amazon EC2 创建	SHA-1	SHA-256
导入到 Amazon EC2	MD5[1]	SHA-256

[1] 如果您将公有的 RSA 密钥导入到 Amazon EC2，系统会使用 MD5 哈希函数计算指纹。无论您如何创建密钥对（例如，通过使用第三方工具或者从使用 Amazon EC2 创建的现有私有密钥生成新的公有密钥），都是如此。

在不同区域使用相同的密钥对时

如果您计划使用相同的密钥对连接到不同 Amazon Web Services 区域 中的实例，必须将公有密钥导入到将在其中使用公有密钥的所有区域。如果使用 Amazon EC2 创建密钥对，您可以 检索公有密钥材料 以便将公有密钥导入到其他区域。

> **Note**
>
> - 如果您使用 Amazon EC2 创建 RSA 密钥对，然后从 Amazon EC2 私有密钥生成公有密钥，则导入的公有密钥的指纹将与原始公有密钥不同。这是因为使用 Amazon EC2 创建的原始 RSA 密钥的指纹是使用 SHA-1 哈希函数计算的，而导入的 RSA 密钥的指纹则使用 MD5 哈希函数计算。
> - 对于 ED25519 密钥对，无论它们是由 Amazon EC2 创建还是导入到 Amazon EC2 的，指纹都将相同，因为两种情况下都使用相同的 SHA-256 哈希函数来计算指纹。

从私有密钥生成指纹

使用以下命令之一可从本地机器上的私有密钥生成指纹。

如果您使用的是 Windows 本地计算机，您可以使用 Windows Subsystem for Linux (WSL) 运行以下命令。按照如何使用 WSL 在 Windows 上安装 Linux 中的说明来安装 WSL 和 Linux 发行版。说明中的示例安装的是 Linux 的 Ubuntu 发行版，但您可以安装任意发行版。系统会提示您重新启动电脑以使更改生效。

- 如果您使用 Amazon EC2 创建密钥对

  如以下示例中所示，使用 OpenSSL 工具生成指纹。

  对于 RSA 密钥对：

  ```
  openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt |
   openssl sha1 -c
  ```

  （Linux 实例）对于 ED25519 密钥对：

  ```
  ssh-keygen -l -f path_to_private_key
  ```

- （仅限 RSA 密钥对）如果您将公有密钥导入 Amazon EC2

  无论密钥对的创建方式如何（例如，使用第三方工具或者从使用 Amazon EC2 创建的现有私有密钥生成新的公有密钥），都可以遵循此过程。

  如以下示例中所示，使用 OpenSSL 工具生成指纹。

```
openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

- 如果您使用 OpenSSH 7.8 或更高版本创建 OpenSSH 密钥对，并将公有密钥导入到 Amazon EC2

  如以下示例中所示，使用 ssh-keygen 生成指纹。

  对于 RSA 密钥对：

  ```
  ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER
   | openssl md5 -c
  ```

  （Linux 实例）对于 ED25519 密钥对：

  ```
  ssh-keygen -l -f path_to_private_key
  ```

EC2 实例的 Amazon EC2 安全组

安全组 充当 EC2 实例的虚拟防火墙，用于控制传入和传出流量。入站规则控制传入到实例的流量，出站规则控制从实例传出的流量。启动实例时，您可以指定一个或多个安全组。如果您未指定安全组，则 Amazon EC2 将对 VPC 使用默认安全组。启动实例后，您可以更改其安全组。

安全性是 Amazon 和您的共同责任。有关更多信息，请参阅 Amazon EC2 中的安全性。Amazon 提供安全组作为一项工具用于保护您的实例，您需要对其进行配置以满足您的安全需求。如果安全组无法满足您的要求，除了使用安全组外，您还可以在任何一个实例上保持自己的防火墙。

定价

使用安全组不会收取任何额外费用。

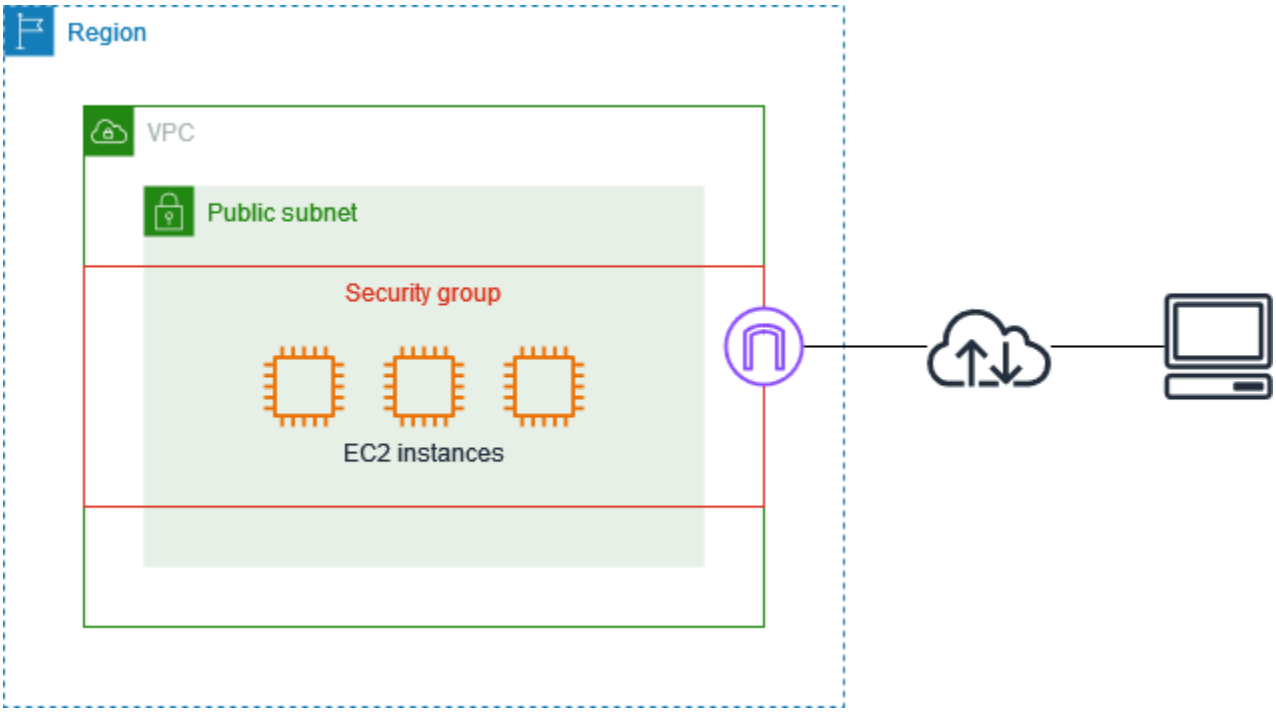
内容

- 概述
- 为 Amazon EC2 实例创建安全组
- 更改 Amazon EC2 实例的安全组
- 删除 Amazon EC2 安全组
- Amazon EC2 安全组连接跟踪
- 针对不同使用案例的安全组规则

概述

您可以将每个实例与多个安全组关联，也可以将每个安全组与多个实例关联。为每个安全组添加规则，规定流入或流出其关联实例的流量。您可以随时修改安全组的规则。新规则和修改后的规则将自动应用到与安全组相关联的所有实例。在决定是否允许流量到达某个实例时，Amazon EC2 会评估与该实例关联的所有安全组中的所有规则。有关更多信息，请参阅 Amazon VPC 用户指南中的安全组规则。

下图显示了一个具有一个子网、一个互联网网关和一个安全组的 VPC。子网包含 EC2 实例。安全组与实例关联。到达实例的唯一流量是得到安全组规则允许的流量。例如，如果安全组包含一条允许来自您的网络的 SSH 流量的规则，则您可以使用 SSH 从您的计算机连接到实例。如果安全组包含一条规则，允许来自与其关联的资源的所有流量，则每个实例都可以接收从其他实例发送的任何流量。



安全组是有状态的 — 如果您从实例发送一个请求，则无论入站安全组规则如何，都将允许该请求的响应流量流入。此外，如果是为响应已允许的入站流量，则该响应可以出站，此时可忽略出站规则。有关更多信息，请参阅 连接跟踪。

为 Amazon EC2 实例创建安全组

安全组用作相关实例的防火墙，可在实例级别控制入站和出站的数据流。您可以向安全组添加规则，以便能够使用 SSH（Linux 实例）或 RDP（Windows 实例）连接到实例。您还可以添加允许客户端流量的规则，例如发往 Web 服务器的 HTTP 和 HTTPS 流量。

在启动某个实例时，您可以将安全组与该实例关联。在关联的安全组中添加或删除规则时，这些更改将自动应用于与安全组关联的所有实例。

启动实例后，即可关联其他安全组。有关更多信息，请参阅 更改 Amazon EC2 实例的安全组。

您可以在创建安全组时添加入站和出站安全组规则，也可以稍后添加规则。有关更多信息，请参阅 配置安全组规则。有关可以添加到安全组的规则示例，请参阅针对不同使用案例的安全组规则。

注意事项

- 新安全组起初只有一条出站规则，即允许所有通信离开资源。您必须添加规则，以便允许任何入站数据流或限制出站数据流。
- 在为允许 SSH 或 RDP 访问实例的规则配置来源时，切勿允许从任何地方进行访问，因为这将允许从互联网上的所有 IP 地址访问实例。这在测试环境中可以接受一小段时间，但是在生产环境中并不安全。
- 如果特定端口具有多个规则，Amazon EC2 将应用最宽松的规则。例如，如果有一条规则允许从 IP 地址 203.0.113.1 访问 TCP 端口 22（SSH），而另一条规则允许从任何地方访问 TCP 端口 22，则所有人都可以访问 TCP 端口 22。
- 您可以将多个安全组与一个实例关联。因此，一个实例可以有数百条适用的规则。访问该实例时，这可能会导致问题。因此，我们建议您尽可能使规则简洁。
- 当您指定一个安全组作为规则的源或目标时，规则会影响与该安全组关联的所有实例。允许的传入流量基于与源安全组相关联的实例的私有 IP 地址 (而不是公有 IP 或弹性 IP 地址)。有关 IP 地址的更多信息，请参阅 Amazon EC2 实例 IP 寻址。
- 默认情况下，Amazon EC2 会阻止端口 25 上的流量。有关更多信息，请参阅 对使用端口 25 发送的电子邮件的限制。

Console

创建安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Security Groups（安全组）。
3. 选择 Create security group（创建安全组）。
4. 输入安全组的描述性名称和简要说明。在创建安全组后，您无法更改其名称和描述。
5. 对于 VPC，请选择将在其中运行 Amazon EC2 实例的 VPC。

6. （可选）要添加入站规则，请选择入站规则。对于每条规则，请选择添加规则并指定协议、端口和来源。例如，要允许 SSH 流量，请在类型中选择 SSH，并为来源指定计算机或网络的公有 IPv4 地址。
7. （可选）要添加出站规则，请选择出站规则。对于每条规则，请选择添加规则并指定协议、端口和目标。若不这样做，您可以保留允许所有出站流量的默认规则。
8. （可选）若要添加标签，请选择 Add new tag（添加新标签），然后输入该标签的键和值。
9. 选择创建安全组。

Amazon CLI

创建安全组

使用以下 create-security-group 命令。

```
aws ec2 create-security-group \
    --group-name my-security-group \
    --description "my security group" \
    --vpc-id vpc-1234567890abcdef0
```

有关添加规则的示例，请参阅the section called “配置安全组规则”。

PowerShell

创建安全组

使用 New-EC2SecurityGroup cmdlet。

```
New-EC2SecurityGroup `
    -GroupName my-security-group `
    -Description "my security group" `
    -VpcId vpc-1234567890abcdef0
```

有关添加规则的示例，请参阅the section called “配置安全组规则”。

更改 Amazon EC2 实例的安全组

您可以在启动 Amazon EC2 实例时指定安全组。启动实例后，即可添加或删除安全组。您也可以随时添加、删除或编辑关联安全组的安全组规则。

安全组与网络接口关联。添加或删除安全组会更改与主网络接口关联的安全组。您还可以更改与任何辅助网络接口关联的安全组。有关更多信息，请参阅 修改网络接口属性。

任务

- 添加或删除安全组
- 配置安全组规则

添加或删除安全组

启动实例后，即可在关联的安全组列表中添加或删除安全组。当您将多个安全组与一个实例相关联时，将有效汇总每个安全组的规则，以创建一组规则。Amazon EC2 使用这组规则确定是否允许流量。

要求

- 实例必须处于 `running` 或 `stopped` 状态。

Console

更改实例的安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择您的实例，依次选择 Actions（操作）、Security（安全）和 Change security groups（更改安全组）。
4. 对于 Associated security groups (关联的安全组)，从列表中选择一个安全组，然后选择 Add security group (添加安全组)。

   要删除已关联的安全组，请为该安全组选择 Remove (删除)。
5. 选择保存。

Amazon CLI

更改实例的安全组

使用以下 modify-instance-attribute 命令。

```
aws ec2 modify-instance-attribute \
```

```
    --instance-id i-1234567890abcdef0 \
    --groups sg-1234567890abcdef0
```

PowerShell

更改实例的安全组

使用 Edit-EC2InstanceAttribute cmdlet。

```
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -Group sg-1234567890abcdef0
```

配置安全组规则

创建安全组后，即可添加、更新或删除安全组的规则。在添加、更新或删除规则时，更改将自动应用于与安全组关联的资源。

有关可以添加到安全组的规则示例，请参阅针对不同使用案例的安全组规则。

所需的权限

在开始之前，请确保您拥有所需权限。有关更多信息，请参阅 示例：使用安全组。

协议和端口

- 使用控制台，当您选择预定义的类型时，将为您指定协议和端口范围。要输入端口范围，必须选择以下自定义类型之一：自定义 TCP 或自定义 UDP。
- 使用 Amazon CLI，您可以使用 `--protocol` 和 `--port` 选项添加带有单个端口的单个规则。要添加多个规则或具有端口范围的规则，请改用 `--ip-permissions` 选项。

来源和目标

- 使用控制台，您可以将以下内容指定为入站规则的源或出站规则的目标：
  - 自定义：IPv4 CIDR 数据块、IPv6 CIDR 数据块、安全组或前缀列表。
  - Anywhere-IPv4 – 0.0.0.0/0 IPv4 CIDR 数据块。
  - Anywhere-IPv6 – ::/0 IPv6 CIDR 数据块。
  - 我的 IP – 本地计算机的公有 IPv4 地址。

- 使用 Amazon CLI，您可以使用 `--cidr` 选项指定 IPv4 CIDR 数据块，也可以使用 `--source-group` 选项指定安全组。要指定前缀列表或 IPv6 CIDR 数据块，请使用 `--ip-permissions` 选项。

> **Warning**
>
> 如果要为端口 22（SSH）或 3389（RDP）添加入站规则，强烈建议仅授权特定 IP 地址或特定范围内的 IP 地址访问实例。如果选择 Anywhere-IPv4，则允许来自所有 IPv4 地址的流量使用指定协议访问实例。如果选择 Anywhere-IPv6，则允许来自所有 IPv6 地址的流量使用指定协议访问实例。

Console

配置安全组规则

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Security Groups（安全组）。
3. 选择安全组。
4. 要编辑入站规则，请从操作或入站规则选项卡中选择编辑入站规则。
   a. 要添加规则，请选择添加规则，再输入规则的类型、协议、端口和来源。

      如果类型为 TCP 或 UDP，则必须输入允许的端口范围。对于自定义 ICMP，您必须从 Protocol（协议）中选择 ICMP 类型名称，并从 Port range（端口范围）中选择代码名称（如果适用）。对于任何其他类型，则会为您配置协议和端口范围。
   b. 要更新规则，请根据需要更改规则的协议、描述和来源。但是，您无法更改来源类型。例如，若来源是 IPv4 CIDR 数据块，则无法指定 IPv6 CIDR 数据块、前缀列表或安全组。
   c. 要删除规则，请选择规则的删除按钮。
5. 要编辑出站规则，请从操作或出站规则选项卡中选择编辑出站规则。
   a. 要添加规则，请选择添加规则，再输入规则的类型、协议、端口和目标。您也可以输入可选描述。

      如果类型为 TCP 或 UDP，则必须输入允许的端口范围。对于自定义 ICMP，您必须从 Protocol（协议）中选择 ICMP 类型名称，并从 Port range（端口范围）中选择代码名称（如果适用）。对于任何其他类型，则会为您配置协议和端口范围。

b. 要更新规则，请根据需要更改规则的协议、描述和来源。但是，您无法更改来源类型。例如，若来源是 IPv4 CIDR 数据块，则无法指定 IPv6 CIDR 数据块、前缀列表或安全组。

c. 要删除规则，请选择规则的删除按钮。

6. 选择保存规则。

Amazon CLI

添加安全组规则

使用 authorize-security-group-ingress 命令添加入站规则。以下示例会允许来自指定前缀列表中的 CIDR 数据块的入站 SSH 流量。

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds=[{PrefixListId=pl-
f8a6439156EXAMPLE}]'
```

使用 authorize-security-group-egress 命令添加出站规则。以下示例会允许端口 80 上指向具有指定安全组的实例的出站 TCP 流量。

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bb6EXAMPLE}]'
```

移除安全组规则

使用以下 revoke-security-group-ingress 命令移除入站规则。

```
aws ec2 revoke-security-group-egress \
    --group id sg-1234567890abcdef0 \
    --security-group-rule-ids sgr-09ed298024EXAMPLE
```

使用以下 revoke-security-group-egress 命令移除出站规则。

```
aws ec2 revoke-security-group-ingress \
    --group id sg-1234567890abcdef0 \
```

```
    --security-group-rule-ids sgr-0352250c1aEXAMPLE
```

修改安全组规则

使用 modify-security-group-rules 命令。以下示例会更改指定安全组规则的 IPv4 CIDR 数据块。

```
aws ec2 modify-security-group-rules \
    --group id sg-1234567890abcdef0 \
    --security-group-rules
 'SecurityGroupRuleId=sgr-09ed298024EXAMPLE,SecurityGroupRule={IpProtocol=tcp,FromPort=80,To
```

PowerShell

添加安全组规则

使用 Grant-EC2SecurityGroupIngress cmdlet 添加入站规则。以下示例会允许来自指定前缀列表中的 CIDR 数据块的入站 SSH 流量。

```
$plid = New-Object -TypeName Amazon.EC2.Model.PrefixListId
$plid.Id = "pl-f8a6439156EXAMPLE"
Grant-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=22; ToPort=22; PrefixListIds=$plid}
```

使用 Grant-EC2SecurityGroupEgress cmdlet 添加出站规则。以下示例会允许端口 80 上指向具有指定安全组的实例的出站 TCP 流量。

```
$uigp = New-Object -TypeName Amazon.EC2.Model.UserIdGroupPair
$uigp.GroupId = "sg-0aad1c26bb6EXAMPLE"
Grant-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -IpPermission @{IpProtocol="tcp"; FromPort=80; ToPort=80; UserIdGroupPairs=
$uigp}
```

移除安全组规则

使用 Revoke-EC2SecurityGroupIngress cmdlet 移除入站规则。

```
Revoke-EC2SecurityGroupIngress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-09ed298024EXAMPLE
```

使用 Revoke-EC2SecurityGroupEgress cmdlet 移除出站规则。

```
Revoke-EC2SecurityGroupEgress `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRuleId sgr-0352250c1aEXAMPLE
```

修改安全组规则

使用 Edit-EC2SecurityGroupRule cmdlet。以下示例会更改指定安全组规则的 IPv4 CIDR 数据块。

```
$sgrr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleRequest
$sgrr.IpProtocol = "tcp"
$sgrr.FromPort = 80
$sgrr.ToPort = 80
$sgrr.CidrIpv4 = "0.0.0.0/0"
$sgr = New-Object -TypeName Amazon.EC2.Model.SecurityGroupRuleUpdate
$sgr.SecurityGroupRuleId = "sgr-09ed298024EXAMPLE"
$sgr.SecurityGroupRule = $sgrr
Edit-EC2SecurityGroupRule  `
    -GroupId sg-1234567890abcdef0 `
    -SecurityGroupRule $sgr
```

删除 Amazon EC2 安全组

使用完为使用 Amazon EC2 实例创建的安全组后，可以将其删除。

要求

- 安全组不能与实例或网络接口关联。
- 其他安全组中的规则无法引用安全组。

Console

删除安全组

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. （可选）要验证安全组是否未与实例关联，请执行以下操作：
   a. 在导航窗格中，选择安全组。

b. 复制要删除的安全组的 ID。

c. 在导航窗格中，选择 Instances (实例)。

d. 在搜索栏中，添加安全组 ID 等于筛选条件，再粘贴安全组的 ID。若没有结果，则安全组未与实例关联。若有结果，则必须取消关联安全组，才能将其删除。

3. 在导航窗格中，选择安全组。
4. 选择安全组，然后依次选择操作、删除安全组。
5. 如果选择了多个安全组，系统会提示您进行确认。如果某些安全组无法删除，我们会显示每个安全组的状态，表明是否会被删除。要确认删除操作，请输入删除。
6. 选择删除。

Amazon CLI

删除安全组

使用以下 delete-security-group 命令。

```
aws ec2 delete-security-group --group-id sg-1234567890abcdef0
```

PowerShell

删除安全组

使用 Remove-EC2SecurityGroup cmdlet。

```
Remove-EC2SecurityGroup -GroupId sg-1234567890abcdef0
```

Amazon EC2 安全组连接跟踪

您的安全组使用连接跟踪来跟踪有关进出实例的流量的信息。将基于流量的连接状态应用规则以确定允许还是拒绝流量。有了这种方法，安全组就具有状态。这意味着无论出站安全组规则如何都允许对入站流量的响应流出实例，反之亦然。

例如，假设您从家中的计算机对您的实例启动 netcat 或与您的实例类似的命令，并且您的入站安全组规则允许 ICMP 流量。跟踪有关连接的信息（包括端口信息）。来自 命令的实例的响应流量不会作为新请求来跟踪，而是作为已建立的连接来跟踪，并且可以流出实例，即使您的出站安全组规则限制出站 ICMP 流量也是如此。

对于除 TCP、UDP 或 ICMP 以外的协议，仅跟踪 IP 地址和协议编号。如果您的实例将流量发送到另一个主机，且该主机在 600 秒内将同一类型的流量发送到您的实例，则无论入站安全组规则如何，您的实例的安全组都将接受该请求。安全组接受它，因为它被视为原始流量的响应流量。

更改安全组规则时，其跟踪的连接不会立即中断。在现有连接超时之前，安全组将继续允许数据包。要确保该流量立即中断，或确保所有流量无论跟踪状态如何，均遵循防火墙规则，您可以使用子网的网络 ACL。网络 ACL 是无状态的，因此不会自动允许响应流量。添加阻止任一方向的流量的网络 ACL 会破坏现有连接。有关更多信息，请参阅 Amazon VPC 用户指南中的网络 ACL。

> **Note**
>
> 安全组对传入或传出 Route 53 Resolver 的 DNS 流量没有影响，Route 53 Resolver 有时称为“VPC+2 IP 地址”（请参阅《Amazon Route 53 开发者指南》中的什么是 Amazon Route 53 Resolver？）或“AmazonProvidedDNS”（请参阅《Amazon Virtual Private Cloud 用户指南》中的使用 DHCP 选项集）。如果希望通过 Route 53 Resolver 筛选 DNS 请求，则可以启用 Route 53 Resolver DNS Firewall（请参阅《Amazon Route 53 开发人员指南》中的 Route 53 Resolver DNS Firewall）。

未跟踪的连接

并非所有通信流都会被跟踪。如果有安全组规则允许所有流量（0.0.0.0/0 或 ::/0）的 TCP 或 UDP 流，而另一个方向上存在对应的规则，允许任何端口（0-65535）的所有响应流量（0.0.0.0/0 或 ::/0），则系统不会跟踪该流量流，除非它属于自动跟踪连接的一部分。基于允许响应流量的入站或出站规则允许未跟踪流的响应流量，而不是基于跟踪信息。

如果删除或修改了支持该流的规则，则会立即中断未被跟踪的通信流。例如，如果您有一个开放 (0.0.0.0/0) 出站规则，并且删除了允许所有 (0.0.0.0/0) 入站 SSH（TCP 端口 22）流量传输到实例的规则（或修改它以使不再允许连接），则您与实例的现有 SSH 连接将立即删除。该连接以前未跟踪，因此更改将断开连接。另一方面，如果您有一个较细化的入站规则，最初允许 SSH 连接（意味着已跟踪该连接），但将该规则更改为不再允许来自当前 SSH 客户端地址的新连接，则由于现有的 SSH 连接被跟踪，它不会中断。

自动跟踪的连接

即使安全组配置无需跟踪，也会自动跟踪通过以下各项建立的连接：

- 仅出口互联网网关
- Global Accelerator 加速器

- NAT 网关
- Network Firewall 防火墙端点
- 网络负载均衡器
- Amazon PrivateLink（接口 VPC 端点）
- Amazon Lambda（Hyperplane 弹性网络接口）
- DynamoDB 网关端点 – 与 DynamoDB 的每个连接都会消耗 2 个 conntrack 条目。

连接跟踪限额

Amazon EC2 定义每个实例可以跟踪的最大连接数量。达到最大值后，由于无法建立新连接，所有发送或接收的数据包都将丢失。发生这种情况时，发送和接收数据包的应用程序无法正常通信。使用 `conntrack_allowance_available` 网络性能指标来确定该实例类型仍然可用的跟踪连接数。

要确定数据包是否因为您实例的网络流量超过了可跟踪的最大连接数量而丢弃，请使用 `conntrack_allowance_exceeded` 网络性能指标。有关更多信息，请参阅 监控 EC2 实例上 ENA 设置的网络性能。

如果您超过了每个实例可跟踪的最大连接数量，建议您借助 Elastic Load Balancing 扩展负载均衡器注册的实例数或实例大小。

连接跟踪最佳实践

非对称路由（即流量通过一个网络接口进入实例，然后通过另一个网络接口离开）可能会降低实例在跟踪流量时所能达到的峰值性能。

为在针对安全组启用连接跟踪时保持峰值性能并优化连接管理，建议您使用以下配置：

- 如有可能，请避免使用非对称路由拓扑。
- 并非使用安全组进行筛选，而是使用网络 ACL。
- 如果必须将安全组与连接跟踪结合使用，请配置尽可能短的空闲连接跟踪超时。有关空闲连接跟踪超时的详细信息，请参阅下一节。
- 由于 Nitrov6 实例的默认超时时间较短，具有长期连接的应用程序（例如数据库连接池、持久 HTTP 连接或流式工作负载）应在实例启动时配置适当的 `TcpEstablishedTimeout` 值。
- 对于长期连接，请将 TCP keepalive 配置为每隔少于 5 分钟发送一次，进而确保连接保持打开状态并保持其跟踪状态。这有助于防止由于空闲超时而导致连接中断，并减少重新建立连接的开销。

有关 Nitro 系统性能优化的更多信息，请参阅 Nitro 系统性能调整注意事项。

空闲连接跟踪超时

安全组会跟踪建立的每个连接，以确保返回数据包按预期交付。每个实例都有可以跟踪的最大连接数量。保持空闲状态的连接可能导致连接跟踪耗尽，并导致无法跟踪连接和丢弃数据包。您可以在弹性网络接口上设置空闲连接跟踪的超时时间。

> **Note**
>
> 此功能仅适用于基于 Nitro 的实例。在部署到生产环境之前，您应该在 Nitrov6 代实例上测试应用程序，减少 350 秒默认连接跟踪超时。

有三种可配置的超时：

- TCP 建立超时：处于已建立状态的空闲 TCP 连接的超时时间（以秒为单位）。
  - 最小值：60 秒
  - 最大值：432000 秒
  - 默认值：Nitro v6 实例类型（不包括 p6e-GB200）为 350 秒；所有其他实例类型（包括 p6e-GB200）为 432000 秒。
  - 建议值：小于 432000 秒
- UDP 超时：空闲 UDP 流的超时时间（以秒为单位），这些流仅在单个方向或单个请求-响应事务上看到流量。
  - 最小值：30 秒
  - 最大值：60 秒
  - 原定设置：30 秒
- UDP 流超时：空闲 UDP 流的超时（以秒为单位），这些流被归类为已看到多个请求-响应事务的流。
  - 最小值：60 秒
  - 最大值：180 秒
  - 原定设置：180 秒

对于以下任何一种情况，您可能需要修改默认超时：

- 如果您使用 Amazon EC2 网络性能指标监控跟踪的连接，那么 conntrack_allowance_exceeded 和 conntrack_allowance_available 指标使您能够监控丢弃的数据包并跟踪连接利用率，通过纵向或横

向扩展操作主动管理 EC2 实例容量，从而在丢弃数据包之前帮助满足网络连接需求。如果您观察到正在 EC2 实例上丢弃 conntrack_allowance_exceeded，您可以设置较低的 TCP 建立超时，以解决因不正确的客户端或网络中继盒导致 TCP/UDP 会话失效的问题，您可能会受益于此操作。

- 通常，负载均衡器或防火墙的 TCP 建立空闲超时在 60 到 90 分钟之间。如果您正在运行的工作负载需要处理来自网络防火墙等设备的大量连接（大于 100k），则建议在 EC2 网络接口上配置类似的超时时间。
- 如果您运行的工作负载使用非对称路由拓扑，我们建议将 TCP 建立空闲超时配置为 60 秒。
- 如果您正在运行具有大量连接的工作负载，例如 DNS、SIP、SNMP、Syslog、Radius 和主要使用 UDP 来处理请求的其他服务，则将“UDP-Stream”超时设置为 60 秒可以提高现有容量的规模/性能，并防止灰色故障。
- 对于通过网络负载均衡器进行的 TCP/UDP 连接，将追踪所有连接。TCP 流的空闲超时值为 350 秒，而 UDP 流为 120 秒，因接口级别的超时值而异。您可能希望在网络接口级别配置超时，以便比负载均衡器的默认值具有更大的超时灵活性。

当您执行以下操作时，可以选择配置连接跟踪超时：

- 创建网络接口
- 修改网络接口属性
- 启动 EC2 实例
- 创建 EC2 实例启动模板

示例

在以下示例中，安全组具有允许 TCP 和 ICMP 流量的入站规则，并具有允许所有出站流量的出站规则。

入站

协议类型	端口号	来源
TCP	22 (SSH)	203.0.113.1/32
TCP	80 (HTTP)	0.0.0.0/0
TCP	80 (HTTP)	::/0
ICMP	全部	0.0.0.0/0

出站

协议类型	端口号	目标
全部	全部	0.0.0.0/0
All	All	::/0

通过与实例或网络接口的直接网络连接，跟踪行为如下：

- 将会跟踪端口 22（SSH）上的入站和出站 TCP 流量，因为入站规则只允许来自 203.0.113.1/32 的流量，而不是所有 IP 地址（0.0.0.0/0）。
- 不会跟踪端口 80（HTTP）上的入站和出站 TCP 流量，因为入站和出站规则允许来自所有 IP 地址的流量。
- 始终跟踪 ICMP 流量。

如果您删除 IPv4 流量的出站规则，则将跟踪所有入站和出站 IPv4 流量，包括端口 80（HTTP）上的流量。如果您删除 IPv6 流量的出站规则，这同样适用于 IPv6 流量。

针对不同使用案例的安全组规则

您可以创建安全组，并添加可反映与安全组关联的实例角色的规则。例如，配置作为 Web 服务器的实例需要允许入站 HTTP 和 HTTPS 访问的安全组规则。同样，数据库实例需要允许数据库类型访问的规则，例如，对 MySQL 允许通过端口 3306 进行访问。

以下是您可以添加到允许特定类型访问的安全组的规则类型示例。

示例

- Web 服务器规则
- 数据库服务器规则
- 用于从您的计算机连接到实例的规则
- 用于在具有相同安全组的实例之间进行连接的规则
- 用于 ping/ICMP 的规则
- DNS 服务器规则
- Amazon EFS 规则
- Elastic Load Balancing 规则

有关说明，请参阅创建安全组和the section called “配置安全组规则”。

Web 服务器规则

以下入站规则允许来自任何 IP 地址的 HTTP 和 HTTPS 访问。如果您为 VPC 启用了 IPv6，则可添加规则以控制来自 IPv6 地址的入站 HTTP 和 HTTPS 流量。

协议类型	协议编号	端口	源 IP	备注
TCP	6	80 (HTTP)	0.0.0.0/0	允许来自任何 IPv4 地址的入站 HTTP 访问。
TCP	6	443 (HTTPS)	0.0.0.0/0	允许来自任何 IPv4 地址的入站 HTTPS 访问
TCP	6	80 (HTTP)	::/0	允许来自任何 IPv6 地址的入站 HTTP 访问
TCP	6	443 (HTTPS)	::/0	允许来自任何 IPv6 地址的入站 HTTPS 访问

数据库服务器规则

以下入站规则是您可以为数据库访问添加的规则示例，具体取决于您在实例运行的数据库类型。有关 Amazon RDS 实例的更多信息，请参阅 Amazon RDS 用户指南。

对于源 IP，请指定以下其中一项：

- 本地网络中的特定 IP 地址或 IP 地址范围（采用 CIDR 数据块表示法）
- 访问数据库的一组实例的安全组 ID

协议类型	协议编号	端口	备注
TCP	6	1433 (MS SQL)	访问 Microsoft SQL Server 数据库的默认端口，例如，在 Amazon RDS 实例上

协议类型	协议编号	端口	备注
TCP	6	3306 (MYSQL/Aurora)	访问 MySQL 或 Aurora 数据库的默认端口，例如，在 Amazon RDS 实例上
TCP	6	5439 (Redshift)	访问 Amazon Redshift 集群数据库的默认端口。
TCP	6	5432 (PostgreSQL)	访问 PostgreSQL 数据库的默认端口，例如，在 Amazon RDS 实例上
TCP	6	1521 (Oracle)	访问 Oracle 数据库的默认端口，例如，在 Amazon RDS 实例上

您可以选择限制来自数据库服务器的出站流量。例如，您可能希望允许访问 Internet 进行软件更新，但限制所有其他类型的流量。您必须先删除允许所有出站流量的默认出站规则。

协议类型	协议编号	端口	目的地 IP	备注
TCP	6	80 (HTTP)	0.0.0.0/0	允许对任何 IPv4 地址进行出站 HTTP 访问
TCP	6	443 (HTTPS)	0.0.0.0/0	允许对任何 IPv4 地址进行出站 HTTPS 访问
TCP	6	80 (HTTP)	::/0	(仅限已启用 IPv6 的 VPC) 允许对任何 IPv6 地址进行出站 HTTP 访问
TCP	6	443 (HTTPS)	::/0	(仅限已启用 IPv6 的 VPC) 允许对任何 IPv6 地址进行出站 HTTPS 访问

用于从您的计算机连接到实例的规则

要连接到您的实例，您的安全组必须拥有允许 SSH 访问 (适用于 Linux 实例) 或 RDP 访问 (适用于 Windows 实例) 的入站规则。

协议类型	协议编号	端口	源 IP
TCP	6	22 (SSH)	您电脑的公有 IPv4 地址或您的本地网络中的 IP 地址范围。如果您为 VPC 启用了 IPv6，并且您的实例有一个 IPv6 地址，则可以输入一个 IPv6 地址或范围。
TCP	6	3389 (RDP)	您电脑的公有 IPv4 地址或您的本地网络中的 IP 地址范围。如果您为 VPC 启用了 IPv6，并且您的实例有一个 IPv6 地址，则可以输入一个 IPv6 地址或范围。

用于在具有相同安全组的实例之间进行连接的规则

要允许与同一安全组关联的实例之间相互通信，您必须明确添加实现此目的的规则。

> **Note**
>
> 如果您将路由配置为通过中间设备在不同子网中的两个实例之间转发流量，则必须确保这两个实例的安全组允许流量在实例之间流动。每个实例的安全组必须引用另一个实例的私有 IP 地址或包含另一个实例的子网的 CIDR 范围作为源。如果您引用另一个实例的安全组作为源，则安全组不允许流量在实例之间流动。

下表描述了允许关联的实例相互通信的安全组的入站规则。该规则允许所有类型的流量。

协议类型	协议编号	端口	源 IP
-1 (All)	-1 (All)	-1 (All)	安全组的 ID 或包含另一个实例的子网的 CIDR 范围（请参阅注释）。

用于 ping/ICMP 的规则

ping 命令是一种 ICMP 流量。要对实例执行 ping 操作，您必须添加以下入站 ICMP 规则之一。

Type	协议	来源		
自定义 ICMP - IPv4	Echo 请求	计算机的公有 IPv4 地址、特定 IPv4 地址、或是来自任何位置的 IPv4 或 IPv6 地址。		
所有 ICMP - IPv4	IPv4 ICMP (1)	计算机的公有 IPv4 地址、特定 IPv4 地址、或是来自任何位置的 IPv4 或 IPv6 地址。		

要使用 ping6 命令对您的实例的 IPv6 地址执行 ping 操作，您必须添加以下入站 ICMPv6 规则。

Type	协议	来源		
所有 ICMP - IPv6	IPv6 ICMP (58)	计算机的 IPv6 地址、特定 IPv4 地址、或是来自任何位置的 IPv4 或 IPv6 地址。		

DNS 服务器规则

如果您已将 EC2 实例设置为 DNS 服务器，则必须确保 TCP 和 UDP 流量可通过端口 53 访问您的 DNS 服务器。

对于源 IP，请指定以下其中一项：

- 网络中的 IP 地址或 IP 地址范围（采用 CIDR 数据块表示法）
- 您网络中需要访问 DNS 服务器的实例组的安全组 ID。

协议类型	协议编号	端口
TCP	6	53
UDP	17	53

Amazon EFS 规则

如果您将 Amazon EFS 文件系统与 Amazon EC2 实例结合使用，与 Amazon EFS 挂载目标关联的安全组必须允许使用 NFS 协议传输的流量。

协议类型	协议编号	端口	源 IP	备注
TCP	6	2049 (NFS)	安全组 ID	允许从与该安全组关联的资源 (包括挂载目标) 进行入站 NFS 访问。

要在 Amazon EC2 实例上挂载 Amazon EFS 文件系统，您必须连接到您的实例。因此，与您的实例关联的安全组必须拥有允许来自本地计算机或本地网络的入站 SSH 的规则。

协议类型	协议编号	端口	源 IP	备注
TCP	6	22 (SSH)	本地计算机的 IP 地址范围或网络的 IP 地址范围（采用 CIDR 数据块表示法）。	允许从您的本地计算机进行入站 SSH 访问。

Elastic Load Balancing 规则

如果在负载均衡器中注册 EC2 实例，与负载均衡器关联的安全组必须允许与实例进行通信。有关更多信息，请参阅弹性负载均衡文档中的以下内容。

- 应用程序负载均衡器的安全组
- 网络负载均衡器的安全组
- 为经典负载均衡器配置安全组

使用接口 VPC 端点访问 Amazon EC2

您可以通过在 VPC 中的资源和 Amazon EC2 API 之间创建私有连接来改善 VPC 的安保状况。您可以像在 VPC 中一样访问 Amazon EC2 API，而无需使用互联网网关、NAT 设备、VPN 连接或 Amazon Direct Connect 连接。VPC 中的 EC2 实例不需要公有 IP 地址即可访问 Amazon EC2 API。

有关更多信息，请参阅 Amazon PrivateLink 指南中的通过 Amazon PrivateLink 访问 Amazon Web Services 服务。

内容

- 创建接口 VPC 端点
- 创建端点策略

创建接口 VPC 端点

使用以下服务名称为 Amazon EC2 创建接口端点：

- com.amazonaws.***region***.ec2 - 为 Amazon EC2 API 操作创建端点。

有关更多信息，请参阅 Amazon PrivateLink 指南中的使用接口 VPC 端点访问 Amazon Web Services 服务。

创建端点策略

端点策略是一种 IAM 资源，您可以将其附加到接口端点。默认端点策略允许通过接口端点访问 Amazon EC2 API 的完全权限。要控制允许从 VPC 访问 Amazon EC2 API 的权限，请将自定义端点策略附加到接口端点。

端点策略指定以下信息：

- 可以执行操作的主体。
- 可执行的操作。
- 可对其执行操作的资源。

> **Important**
>
> 当将非默认策略应用于 Amazon EC2 的接口 VPC 端点时，某些失败的 API 请求（例如 `RequestLimitExceeded` 中失败的请求）可能不会记录到 Amazon CloudTrail 或 Amazon CloudWatch。

有关更多信息，请参阅 Amazon PrivateLink 指南中的使用端点策略控制对服务的访问权限。

以下示例显示一个 VPC 端点策略，该策略拒绝创建未加密的卷或启动具有未加密卷的实例的权限。示例策略还授予执行所有其他 Amazon EC2 操作的权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Action": "ec2:*",
        "Effect": "Allow",
        "Resource": "*",
        "Principal": "*"
    },
    {
        "Action": [
            "ec2:CreateVolume"
        ],
        "Effect": "Deny",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    },
    {
        "Action": [
            "ec2:RunInstances"
        ],
        "Effect": "Deny",
```

```
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "Bool": {
                "ec2:Encrypted": "false"
            }
        }
    }]
}
```

适用于 Amazon EC2 实例的存储选项

Amazon EC2 为您的实例提供了灵活、经济且易于使用的数据存储选项。各选项都具有独特的性能和耐久性。这些存储选项既可以单独使用，也可以组合使用，以便满足您的需求。

数据块存储

- Amazon EBS – Amazon EBS 提供持久的块级存储卷，您可以将其附加到实例或从实例中分离。您可以将多个 EBS 卷附加到一个实例。EBS 卷始终不受其关联实例的生命周期影响。您可以对 EBS 卷进行加密。要保留您数据的备份副本，您可以从 EBS 卷创建快照。快照会存储在 Amazon S3 中。您可以从快照创建 EBS 卷。
- 适用于 EC2 实例的实例存储临时块存储 – 实例存储可以为实例提供临时性块级存储。实例存储卷的数量、大小和类型由实例类型和实例大小决定。实例存储卷上的数据仅在关联实例的生命周期内保留；如果您停止、休眠或终止实例，则实例存储卷上的所有数据都会丢失。

对象存储

- Amazon S3 – Amazon S3 提供可靠且廉价的数据存储基础设施。它的设计理念是通过支持您随时从 Amazon EC2 内部或从网络上的任何地方存储和检索任何数量的数据，从而简化整个网络计算。例如，您可以使用 Amazon S3 存储数据和应用程序的备份副本。Amazon EC2 使用 Amazon S3 存储 EBS 快照以及 Amazon S3 支持的 AMI。

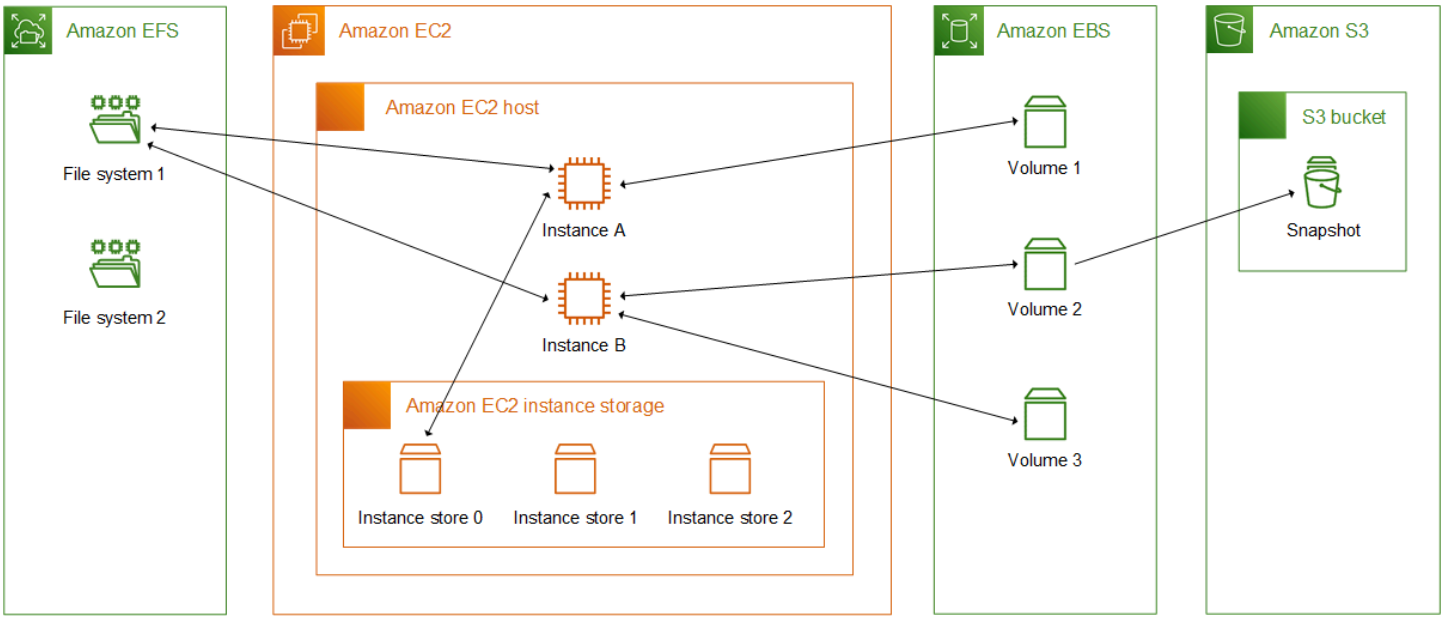
文件存储

- Amazon EFS（仅适用于 Linux 实例）– Amazon EFS 提供适用于 Amazon EC2 的可扩展文件存储。您可以创建 EFS 文件系统并配置实例来装载文件系统。您可以使用 EFS 文件系统作为在多个实例上运行的工作负载和应用程序的通用数据来源。
- Amazon FSx – 借助 Amazon FSx，您可以在云中启动、运行和扩展功能丰富的高性能文件系统。Amazon FSx 是一项完全托管的服务，支持各种工作负载。您可以在这些广泛使用的文件系统之间进行选择：Lustre、NetApp ONTAP、OpenZFS 和 Windows File Server。

文件缓存

- 将 Amazon File Cache 与 Amazon EC2 实例结合使用 – Amazon File Cache 在 Amazon 上提供临时的高性能缓存，用于处理文件数据。缓存提供对 Amazon EC2 上计算工作负载的读写数据访问权限，以及亚毫秒级别的延迟、高达数百 GB/s 的吞吐量以及高达数百万的 IOPS。

下图显示了这些存储选项和您的实例之间的关系。



Amazon 存储定价

打开 Amazon 定价，滚动到 Amazon 产品定价，然后选择存储。选择存储产品以打开其定价页面。

适用于 Amazon EC2 实例的 Amazon EBS 永久块存储

Amazon Elastic Block Store（Amazon EBS）提供了可扩展、高性能的块存储资源，可以与 Amazon EC2 实例一起使用。使用 Amazon EBS，您可以创建和管理以下数据块存储资源：

- Amazon EBS 卷 – 挂载到 Amazon EC2 实例的存储卷。将卷附加到实例后，您可以像使用数据块存储一样使用卷。实例就像与本地驱动器交互一样与该卷交互。
- Amazon EBS 快照 – 独立于卷本身持续存在的 Amazon EBS 卷的时间点备份。您可以通过创建快照来备份 Amazon EBS 卷上的数据。而后可以随时从这些快照还原新卷。

您可以在启动期间创建 EBS 卷并将其附加到实例，也可以在启动后随时创建 EBS 卷并将其附加到实例。您还可以在不分离卷或重启实例的情况下增加 EBS 卷的大小或性能。

创建后，您可以随时从 EBS 卷中创建 EBS 快照。您可以使用 EBS 快照来备份卷上存储的数据。然后，您可以使用这些快照立即还原卷或跨 Amazon Web Services 账户、Amazon 区域或可用区迁移数据。您可以使用 Amazon Data Lifecycle Manager 或 Amazon Backup 自动创建、保留和删除 EBS 快照。

托管 EBS 卷是指由服务提供商（例如 Amazon EKS 自动模式）管理的卷。您不能直接修改托管 EBS 卷的设置。托管 EBS 卷由托管字段中的 true 值标识。有关更多信息，请参阅 Amazon EC2 托管式实例。

有关使用卷和快照的更多信息，请参阅《Amazon EBS 用户指南》https://docs.amazonaws.cn/ebs/latest/userguide/what-is-ebs.html。

Amazon EC2 实例的 Amazon EBS 卷限制

您可以挂载到实例的最大 Amazon EBS 卷数取决于实例类型和实例规模。如果超过了实例的卷挂载限制，则连接请求将会失败并显示 `AttachmentLimitExceeded` 错误。

考虑应将多少个卷添挂载到实例时，应考虑是否需要增加 I/O 带宽或存储容量。

带宽与容量

为获得一致且可预测的带宽使用案例，请将经 Amazon EBS 优化的实例与通用型 SSD 卷或预调配 IOPS SSD 卷结合使用。要获取最大性能，请将您为卷预置的 IOPS 与实例类型带宽匹配。

对于 RAID 配置，您可能会发现大于 8 个卷的阵列由于 I/O 开销提高而降低了性能回报。测试您的各个应用程序性能并根据需要优化。

目录

- 在 Nitro 系统上构建的实例卷的限制
  - 专用 EBS 卷限制
  - 共享 EBS 卷限制
- 基于 Xen 的实例的卷限制
  - Linux 实例
  - Windows 实例

在 Nitro 系统上构建的实例卷的限制

基于 Nitro System 构建的实例的卷限制取决于实例类型。某些 Nitro 实例类型具有专用 EBS 卷限制，而大多数实例类型具有共享卷限制。要查看每种实例类型的卷附加限制，请参阅以下内容：

- Amazon EBS 规格 – 通用型
- Amazon EBS 规格 – 计算优化型

- Amazon EBS 规格 – 内存优化型
- Amazon EBS 规格 – 存储优化型
- Amazon EBS 规格 – 加速计算型
- Amazon EBS 规格 – 高性能计算型
- Amazon EBS 规格 – 上一代

专用 EBS 卷限制

以下 Nitro 实例类型的专用 EBS 卷限制取决于实例大小。此限制不与其他设备挂载共享。换句话说，无论连接的设备（如 NVMe 实例存储卷和网络接口）有多少，您都可以连接不超过卷挂载限制的任意数量的 EBS 卷。

- 通用型：M7a | M7i | M7i-flex | M8a | M8azn | M8g | M8gb | M8gd | M8gn | M8i | M8id | M8i-flex | M8in | M8idn | M8ine | M8ib | M8idb | M9g | M9gd
- 计算优化型：C7a | C7i | C7i-flex | C8a | C8g | C8gb | C8gd | C8gn | C8i | C8id | C8i-flex | C8in | C8ine | C8ib
- 内存优化型：R7a | R7i | R7iz | R8a | R8g | R8gb | R8gd | R8gn | R8i | R8id | R8i-flex | R8in | R8idn | R8ib | R8idb | U7i-6tb | U7i-8tb | U7i-12tb | U7in-16tb | U7in-24tb | U7in-32tb | U7inh-32tb | X8g | X8aedz | X8i
- 存储优化型：I7i | I7ie | I8g | I8ge
- 加速计算型：F2 | G6 | G6e | G6f | Gr6 | Gr6f | G7e | P4d | P4de | P5 | P5e | P5en | P6-B200 | P6-B300 | P6e-GB200 | Trn2 | Trn2u
- 高性能计算型：Hpc7a | Hpc8a

共享 EBS 卷限制

所有其他 Nitro 实例类型（专用 EBS 卷限制 中未列出）都有卷挂载限制，该限制在 Amazon EBS 卷、网络接口和 NVMe 实例存储卷之间共享。您可以连接任意数量的 Amazon EBS 卷，但是不能超过该限制（减去连接的网络接口和 NVMe 实例存储卷的数量）。请记住，每个实例必须至少有一个网络接口，并且 NVMe 实例存储卷在启动时会自动附加。

大多数 Nitro 实例最多支持 28 个附加项。以下示例演示如何计算可以附加的 EBS 卷数量。

示例

- 对于只有主网络接口的 `m5.xlarge` 实例，您可以附加 27 个 EBS 卷。

28 个卷 - 1 个网络接口 = 27 个

- 使用带有两个额外网络接口的 `m5.xlarge` 实例时，您可以附加 25 个 EBS 卷。

  28 个卷 - 3 个网络接口 = 25 个

- 使用带有两个额外网络接口的 `m5d.xlarge` 实例时，您可以附加 24 个 EBS 卷。

  28 个卷 - 3 个网络接口 - 1 个 NVMe 实例存储卷 = 24 个

基于 Xen 的实例的卷限制

对于 T2 等基于 Xen 的实例，卷限制取决于操作系统。

有关更多信息，请参阅基于 Xen 的实例。

Linux 实例

将超过 40 个卷附加到基于 Xen 的 Linux 实例可能会导致启动失败。此数字包括根卷以及所有附加的实例存储卷和 Amazon EBS 卷。

如果连接了大量卷的实例出现启动问题，请停止该实例，分离所有在启动过程中不必要的卷，启动该实例，然后在实例运行之后重新附加这些卷。

> **Important**
>
> 如果将 40 个以上的卷附加到基于 Xen 的 Linux 实例，系统只会尽力支持，不对此进行保证。

Windows 实例

下表基于所使用的驱动程序显示基于 Xen 的 Windows 实例的卷限制。这些数字包括根卷以及所有附加的实例存储卷和 Amazon EBS 卷。

驱动程序	卷限制
Amazon PV	26
Citrix 半虚拟化驱动程序	26
Red Hat 半虚拟化	17

建议使用 Amazon PV 或 Citrix PV 驱动程序连接至基于 Xen 的 Windows 实例的卷不要超过 26 个，否则可能导致性能问题。要确定您的实例所使用的半虚拟化驱动程序，或是要将 Windows 实例从 Red Hat 升级到 Citrix 半虚拟化驱动程序，请参阅 the section called “升级半虚拟化驱动程序”。

> **Important**
>
> 如果附加到基于 Xen 的 Windows 实例的卷的数量超过下面的数字，系统只会尽力支持，不对此提供保证。

有关设备名称如何与卷相关联的更多信息，请参阅 如何为 Amazon EC2 Windows 实例附加和映射卷。

EBS 卡

大多数实例类型都支持一个 EBS 卡。支持多个 EBS 卡的实例类型在 EBS 吞吐量和 IOPS 方面均可提供更高的 EBS 优化性能。您的 Amazon EC2 实例的最大性能在每个 EBS 卡上平均分配。例如，在通过 2 个 EBS 卡支持总共 `1,000,000` 个 IOPS 的 EC2 实例上，每个 EBS 卡最多可以支持 `500,000` 个 IOPS。有关您的 Amazon EC2 实例支持的 Amazon EBS 性能的信息，请参阅 Amazon EBS 优化的实例类型。

在将 Amazon EBS 卷附加到支持多个 EBS 卡的实例时，可以通过指定 EBS 卡索引为该卷选择 EBS 卡。根卷必须附加到 EBS 卡索引 0。

具有多个 EBS 卡的实例类型

以下实例类型支持多个 EBS 卡。有关实例类型支持的 Amazon EBS 卷数量的信息，请参阅 Amazon EC2 实例的 Amazon EBS 卷限制。

实例类型	EBS 卡的数量
通用	
m8gb.48xlarge	2
m8gb.metal-48xl	2
m8gn.48xlarge	2
m8gn.metal-48xl	2

实例类型	EBS 卡的数量
m8in.96xlarge	2
m8in.metal-96xl	2
m8idn.96xlarge	2
m8idn.metal-96xl	2
m8ib.96xlarge	2
m8ib.metal-96xl	2
m8idb.96xlarge	2
m8idb.metal-96xl	2
计算优化	
c8gb.48xlarge	2
c8gb.metal-48xl	2
c8gn.48xlarge	2
c8gn.metal-48xl	2
c8in.96xlarge	2
c8in.metal-96xl	2
c8ib.96xlarge	2
c8ib.metal-96xl	2
内存优化	
r8gb.48xlarge	2
r8gb.metal-48xl	2

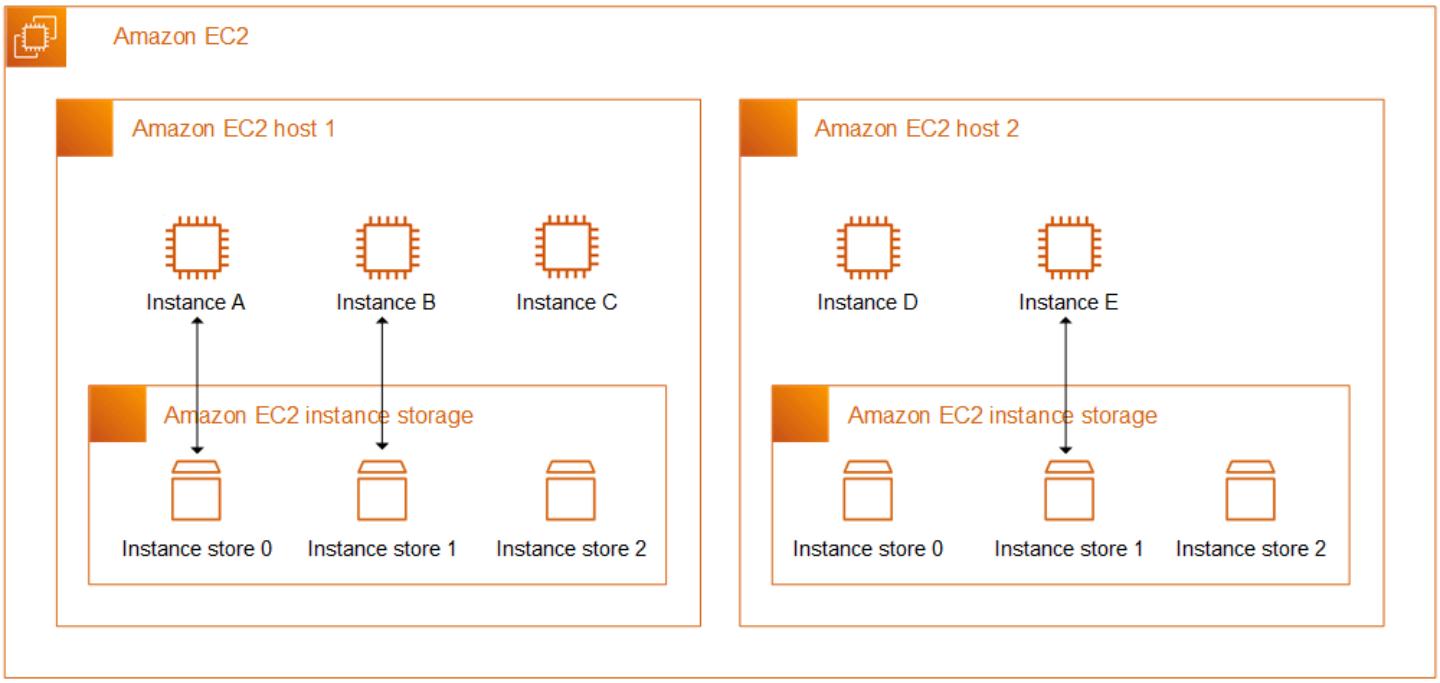
实例类型	EBS 卡的数量
r8gn.48xlarge	2
r8gn.metal-48xl	2
r8in.96xlarge	2
r8in.metal-96xl	2
r8idn.96xlarge	2
r8idn.metal-96xl	2
r8ib.96xlarge	2
r8ib.metal-96xl	2
r8idb.96xlarge	2
r8idb.metal-96xl	2

适用于 EC2 实例的实例存储临时块存储

实例存储为您的 EC2 实例提供临时块级存储。此存储由已物理附加到主机的磁盘提供。实例存储非常适合临时存储频繁更改的信息，例如缓冲区、缓存、Scratch 数据和其他临时内容。它还可用于存储您在一组实例中复制的临时数据，例如负载均衡的 Web 服务器池。

实例存储由一个或多个显示为块设备的实例存储卷组成。实例存储的大小以及可用设备的数量因实例类型和实例大小而异。例如，并非每种实例类型都提供实例存储卷。有关更多信息，请参阅 EC2 实例的实例存储卷限制。

实例存储卷的虚拟设备按从 `ephemeral0` 到 `ephemeral23` 的顺序指定虚拟设备名称。例如，对于支持一个实例存储卷的实例类型，该卷的虚拟设备名称为 `ephemeral0`。对于支持四个实例存储卷的实例类型，四个卷的虚拟设备名称如下所示：`ephemeral0`、`ephemeral1`、`ephemeral2` 和 `ephemeral3`。



实例存储定价

使用为您的实例提供的实例存储卷无需支付额外费用。实例存储卷包含在实例使用成本中。

内容

- Amazon EC2 实例存储卷的数据持久性
- EC2 实例的实例存储卷限制
- 适用于 EC2 实例的 SSD 实例存储卷
- 将实例存储卷添加到 EC2 实例
- 为 M1 和 C1 EC2 实例启用实例存储交换卷
- 在 EC2 实例上初始化实例存储卷
- Amazon EC2 实例存储卷的详细性能统计数据

Amazon EC2 实例存储卷的数据持久性

只有在实例启动时才会附加实例存储卷。无法在启动实例后附加实例存储卷。您无法将实例存储卷与一个实例分离并将该卷附加到另一个实例。

实例存储卷仅在其连接的实例的生命周期内存在。您无法将实例存储卷的生命周期配置为在其关联实例的生命周期之后保留。

即使实例重启，实例存储卷上的数据仍会保留。但是，如果实例停止、休眠或终止，则数据不会保留。当实例停止、休眠或终止后，实例存储卷的每个块都会通过加密方式删除。

因此，切勿依赖实例存储卷来存储珍贵且需要长期保存的数据。如果您需要在实例的生命周期之后保留存储在实例存储卷上的数据，需要手动将该数据复制到更持久的存储中，例如 Amazon EBS 卷、Amazon S3 存储桶或 Amazon EFS 文件系统。

有些事件可能导致数据无法在实例的整个生命周期内持续存在。下表显示了虚拟化实例和裸机实例的实例存储卷上的数据在特定事件期间是否会永久保存。

事件	您的数据会如何处理？
用户启动的实例生命周期事件	
实例重启	The data persists
实例停止	The data does not persist
实例休眠	The data does not persist
实例终止	The data does not persist
实例类型更改	The data does not persist *
从实例创建 EBS 支持的 AMI	The data does not persist in the created AMI **
从实例创建 Amazon S3 支持的 AMI (Linux instances)	The data persists in the AMI bundle uploaded to Amazon S3 ***
用户启动的操作系统事件	
A shutdown is initiated	The data does not persist †
A restart is initiated	The data persists
Amazon 计划的事件	
实例停止	The data does not persist
实例重启	The data persists
系统重启	The data persists

事件	您的数据会如何处理？
实例指令引退	The data does not persist
计划外事件	
简化的自动恢复	The data does not persist
CloudWatch 基于操作的恢复	The data does not persist
The underlying disk fails	The data on the failed disk does not persist
Power failure	The data persists upon reboot

* 如果新实例类型支持实例存储，该实例将获得新实例类型支持的实例存储卷数，但数据不会传输到新实例。如果新的实例类型不支持实例存储，实例将无法获得实例存储卷。

** 数据不包含在 EBS 支持的 AMI 中，也不包含在连接到从该 AMI 启动的实例的实例存储卷中。

*** 数据会包含在上传至 Amazon S3 的 AMI 捆绑包中。当您从该 AMI 启动实例时，该实例会将实例存储卷与创建 AMI 时它们所包含的数据捆绑在 AMI 中。

† 终止保护和停止保护不能让实例避免因实例上操作系统启动的关机引起的实例停止或终止。存储在实例存储卷上的数据不会在实例停止和终止事件中持续存在。

EC2 实例的实例存储卷限制

实例存储卷的数量、大小和类型由实例类型决定。某些实例类型（例如，C8i、M8i 和 R8i）不支持实例存储卷，而其他实例类型（例如，C8id、M8id 和 R8id）可支持实例存储卷。您向实例附加的实例存储卷不能超过其实例类型所支持的数量。对于支持实例存储卷的实例类型，实例存储卷的数量和大小因实例大小而异。例如，`r8id.large` 支持 1 x 118 GB 的实例存储卷，而 `r8id.32xlarge` 支持 2 x 3800GB 的实例存储卷。

对于具有 NVMe 实例存储卷的实例类型，所有支持的实例存储卷都会在启动时自动连接到实例。对于具有非 NVMe 实例存储卷的实例类型（例如，C1、C3、M1、M2、M3、R3、D2、H1、I2、X1 和 X1e），您必须在启动时为要连接的实例存储卷手动指定块设备映射。然后，在实例启动后，必须先格式化并装载连接的实例存储卷，然后才能使用它们。无法在启动实例后附加实例存储卷。

某些实例类型使用基于 NVMe 或 SATA 的固态硬盘（SSD），而其他实例类型使用基于 SATA 的硬盘驱动器（HDD）。SSD 提供高随机 I/O 性能，且延迟非常低；但实例终止时不需要保留数据，或者可以利用容错架构。有关更多信息，请参阅 适用于 EC2 实例的 SSD 实例存储卷。

NVMe 实例存储卷和某些 HDD 实例存储卷上的数据采用静态加密。有关更多信息，请参阅 Amazon EC2 中的数据保护。

可用的实例存储卷

《Amazon EC2 实例类型指南》提供每种支持的实例类型可以使用的实例存储卷的数量、大小、类型和性能优化。有关更多信息，请参阅下列内容：

- 实例存储规格 – 通用型
- 实例存储规格 – 计算优化型
- 实例存储规格 – 内存优化型
- 实例存储规格 – 存储优化型
- 实例存储规格 – 加速计算型
- 实例存储规格 – 高性能计算型
- 实例存储规格 – 上一代

Console

检索实例存储卷信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instance Types (实例类型)。
3. 添加筛选条件本地实例存储 = true。存储列指示该实例类型的实例存储总大小。
4. （可选）单击首选项图标，然后打开存储磁盘计数。此列指示实例存储卷的数量。
5. （可选）添加筛选条件以进一步筛选相关的特定实例类型。

Amazon CLI

检索实例存储卷信息

请使用 describe-instance-types 命令。以下示例显示了在具有实例存储卷的 R8i 实例系列中，每种实例类型的实例存储总大小。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r8i*" "Name=instance-storage-
supported,Values=true" \
    --query 'sort_by(InstanceTypes, &InstanceStorageInfo.TotalSizeInGB)[].
{InstanceType:InstanceType,TotalSizeInGB:InstanceStorageInfo.TotalSizeInGB}' \
    --output table
```

下面是示例输出。

```
-------------------------------------
|        DescribeInstanceTypes       |
+------------------+-----------------+
|   InstanceType   |  TotalSizeInGB  |
+------------------+-----------------+
|  r8id.large      |  118            |
|  r8id.xlarge     |  237            |
|  r8id.2xlarge    |  474            |
|  r8id.4xlarge    |  950            |
|  r8id.8xlarge    |  1900           |
|  r8id.12xlarge   |  2850           |
|  r8id.16xlarge   |  3800           |
|  r8id.24xlarge   |  5700           |
|  r8id.32xlarge   |  7600           |
|  r8id.48xlarge   |  11400          |
|  r8id.metal-48xl |  11400          |
|  r8id.96xlarge   |  22800          |
|  r8id.metal-96xl |  22800          |
+------------------+-----------------+
```

获取实例类型的完整实例存储详细信息

请使用 describe-instance-types 命令。

```
aws ec2 describe-instance-types \
    --filters "Name=instance-type,Values=r8id.32xlarge" \
    --query 'InstanceTypes[0].InstanceStorageInfo' \
    --output json
```

示例输出显示，此实例类型有两个 3800 GB 的 NVMe SSD 卷，总共 7600 GB 的实例存储空间。

```
{
```

```
    "TotalSizeInGB": 7600,
    "Disks": [
        {
            "SizeInGB": 3800,
            "Count": 2,
            "Type": "ssd"
        }
    ],
    "NvmeSupport": "required",
    "EncryptionSupport": "required"
}
```

PowerShell

检索实例存储卷信息

使用 Get-EC2InstanceType cmdlet。以下示例显示了在具有实例存储卷的 R8i 实例系列中，每种实例类型的实例存储总大小。

```
(Get-EC2InstanceType -Filter `
    @{Name="instance-type"; Values="r8i*"},
    @{Name="instance-storage-supported"; Values="true"}) |
    Sort-Object {$_.InstanceStorageInfo.TotalSizeInGB} |
    Format-Table InstanceType,

 @{Name="Disks.SizeInGB";Expression={$_.InstanceStorageInfo.Disks[0].SizeInGB}},
        @{Name="Disks.Count";Expression={$_.InstanceStorageInfo.Disks[0].Count}},
        @{Name="TotalSizeInGB";Expression={$_.InstanceStorageInfo.TotalSizeInGB}}
```

下面是示例输出。

```
InstanceType    Disks.SizeInGB Disks.Count TotalSizeInGB
------------    -------------- ----------- -------------
r8id.large                118           1           118
r8id.xlarge               237           1           237
r8id.2xlarge              474           1           474
r8id.4xlarge              950           1           950
r8id.8xlarge             1900           1          1900
r8id.12xlarge            2850           1          2850
r8id.16xlarge            3800           1          3800
r8id.24xlarge            2850           2          5700
r8id.32xlarge            3800           2          7600
```

```
r8id.48xlarge             3800          3         11400
r8id.metal-48xl           3800          3         11400
r8id.96xlarge             3800          6         22800
r8id.metal-96xl           3800          6         22800
```

获取实例类型的完整实例存储详细信息

使用 Get-EC2InstanceType cmdlet。

```
(Get-EC2InstanceType `
    -Filter @{Name="instance-type"; Values="r8id.32xlarge"}).InstanceStorageInfo |
    Format-List *,
        @{Name="Disks.Count";Expression={$_.Disks[0].Count}},
        @{Name="Disks.SizeInGB";Expression={$_.Disks[0].SizeInGB}},
        @{Name="Disks.Type";Expression={$_.Disks[0].Type.Value}}
```

示例输出显示，此实例类型有两个 3800 GB 的 NVMe SSD 卷，总共 7600 GB 的实例存储空间。

```
Disks             : {Amazon.EC2.Model.DiskInfo}
EncryptionSupport : required
NvmeSupport       : required
TotalSizeInGB     : 7600
Disks.Count       : 2
Disks.SizeInGB    : 3800
Disks.Type        : ssd
```

适用于 EC2 实例的 SSD 实例存储卷

像其他实例存储卷一样，您必须在启动实例时为其映射 SSD 实例存储卷。SSD 实例卷上的数据仅在其关联实例的生命周期内保留。有关更多信息，请参阅将实例存储卷添加到 EC2 实例。

NVMe SSD 卷

有些实例提供符合非易失性存储器规范（NVMe）的固态硬盘（SSD）实例存储卷。有关每种实例类型支持的实例存储卷类型的更多信息，请参阅EC2 实例的实例存储卷限制。

NVMe 实例存储上的数据是使用在实例上的硬件模块中实施的 XTS-AES-256 数据块密码加密的。加密密钥是使用硬件模块生成的，并且对每台 NVMe 实例存储设备都是唯一的。当实例停止或终止并且无法恢复时，将销毁所有加密密钥。无法禁用此加密，并且无法提供自己的加密密钥。

Linux 实例

要访问 NVMe 卷，必须安装 NVMe 驱动程序。以下 AMI 满足此要求：

- AL2023
- Amazon Linux 2
- Amazon Linux AMI 2018.03 和更高版本
- Ubuntu 14.04 或更高版本（采用 `linux-aws` 内核）

> **Note**
>
> Amazon 基于 Graviton 的实例类型需要 Ubuntu 18.04 或更高版本（采用 `linux-aws` 内核）

- Red Hat Enterprise Linux 7.4 或更高版本
- SUSE Linux Enterprise Server 12 SP2 或更高版本
- CentOS 7.4.1708 或更高版本
- FreeBSD 11.1 或更高版本
- Debian GNU/Linux 9 或更高版本

- Bottlerocket

连接到实例后，您可以使用 lspci 命令列出 NVMe 设备。以下是支持 4 台 NVMe 设备的 `i3.8xlarge` 实例的示例输出。

```
[ec2-user ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Device 1d0f:ec20
00:17.0 Non-Volatile memory controller: Device 1d0f:cd01
00:18.0 Non-Volatile memory controller: Device 1d0f:cd01
00:19.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1a.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1f.0 Unassigned class [ff80]: XenSource, Inc. Xen Platform Device (rev 01)
```

如果您使用了受支持的操作系统但未看到 NVMe 设备，请使用以下命令验证是否已加载 NVMe 模块。

- Amazon Linux、Amazon Linux 2、Ubuntu 14/16、Red Hat Enterprise Linux、SUSE Linux Enterprise Server、CentOS 7

```
$ lsmod | grep nvme
nvme           48813  0
```

- Ubuntu 18

```
$ cat /lib/modules/$(uname -r)/modules.builtin | grep nvme
s/nvme/host/nvme-core.ko
kernel/drivers/nvme/host/nvme.ko
kernel/drivers/nvmem/nvmem_core.ko
```

NVMe 卷符合 NVMe 1.0e 规范。您可以对 NVMe 卷使用 NVMe 命令。利用 Amazon Linux，您可以使用 `nvme-cli` 命令从存储库安装 yum install 程序包。利用其他受支持的 Linux 版本，您可以下载 `nvme-cli` 包（如果包在映像中不可用）。

Windows 实例

以下操作系统的最新 Amazon Windows AMI 包含用于与显示为 NVMe 块设备的 SSD 实例存储卷进行交互的 Amazon NVMe 驱动程序，以提高性能：

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2

连接到实例后，您可以验证 Disk Manager 中是否显示了 NVMe 卷。在任务栏上，打开 Windows 徽标的上下文（右键单击）菜单，然后选择磁盘管理。

由 Amazon 提供的 Amazon Windows AMI 包含 Amazon NVMe 驱动程序。如果您使用的不是最新 Amazon Windows AMI，则可以安装最新的 Amazon NVMe 驱动程序。

非 NVMe SSD 卷

以下实例支持使用非 NVMe SSD 来提供高随机 /O 性能的实例存储卷：C3、I2、M3、R3 和 X1。有关每种实例类型支持的实例存储卷的更多信息，请参阅EC2 实例的实例存储卷限制。

基于 SSD 的实例存储卷的 I/O 性能

随着您不断在您的实例的基于 SSD 的实例存储卷中填充数据，您可以达到的写入 IOPS 将不断减少。这是因为，SSD 控制器必须执行额外的工作，即查找可用空间、重写现有数据，以及擦除未使用的空间以使之可供重写。这一垃圾回收过程将导致对 SSD 的内部写入放大影响，这以 SSD 写入操作数相对于用户写入操作数的比率形式来表示。如果写入操作数并非 4096 字节的倍数，或不在 4096 字节这一边界上，则性能的降低会更明显。如果您写入的字节数较少或不在边界上，则 SSD 控制器必须读取周围的数据并在新位置存储结果。这种模式会大大增加写入放大的影响，加长延迟，并显著降低 I/O 性能。

SSD 控制器可以使用多种策略来减少写入放大的影响。其中的一个策略是在 SSD 实例存储中预订空间，以便控制器更高效地管理可用于写入操作的空间。这称为超额配置。为实例提供的基于 SSD 的实例存储卷不会为超额预置预保留任何空间。要减少写入放大问题造成的影响，建议您留出 10% 的卷空间不进行分区，以便 SSD 控制器使用这部分空间进行超额预置。虽然这会减少您可使用的存储空间，但可提高性能，即使磁盘容量快用完也是如此。

对于支持 TRIM 的实例存储卷，可在不再需要已写入的数据时，使用 TRIM 命令告知 SSD 控制器此情况。这将为控制器提供更多可用空间，从而可以减少写入放大的影响并提高性能。有关更多信息，请参阅 实例存储卷 TRIM 支持。

实例存储卷 TRIM 支持

某些实例类型支持带有 TRIM 的 SSD 卷。有关更多信息，请参阅 EC2 实例的实例存储卷限制。

> **Note**
>
> （仅限 Windows 实例）从 Amazon PV 驱动程序版本 7.3.0 起，运行 Windows Server 2012 R2 的实例支持 TRIM。运行更低版本的 Windows Server 的实例不支持 TRIM。

支持 TRIM 的实例存储卷先经全面删减，然后再分配到您的实例。这些卷在实例启动时未经过文件系统的格式化处理，因此，您必须先进行格式化，而后才能挂载和使用。为了更快地访问这些卷，您在格式化它们时应跳过 TRIM 操作。

（Windows 实例）要在初始格式化过程中暂时禁用 TRIM 支持，请使用 `fsutil behavior set DisableDeleteNotify 1` 命令。完成格式化后，使用 `fsutil behavior set DisableDeleteNotify 0` 重新启用 TRIM 支持。

利用支持 TRIM 的实例存储卷，您可在不再需要已写入的数据时使用 TRIM 命令告知 SSD 控制器此情况。这将为控制器提供更多可用空间，从而可以减少写入放大的影响并提高性能。在 Linux 实例上，使用 `fstrim` 命令启用定期 TRIM。在 Windows 实例上，使用 `fsutil behavior set DisableDeleteNotify 0` 命令确保在正常操作期间启用 TRIM 支持。

将实例存储卷添加到 EC2 实例

对于具有 NVMe 实例存储卷的实例类型，所有支持的实例存储卷都会在启动时自动连接到实例。将在实例启动时自动枚举 NVMe 实例存储卷，并为其分配设备名称。

对于具有非 NVMe 实例存储卷的实例类型（例如，C1、C3、M1、M2、M3、R3、D2、H1、I2、X1 和 X1e），您必须在启动时为要连接的实例存储卷手动指定块设备映射。可以在实例启动请求中指定块设备映射，也可以在用于启动实例的 AMI 中指定。块设备映射包括设备名称和映射到其上的卷。有关更多信息，请参阅 Amazon EC2 实例上卷的块设备映射。

> **Important**
>
> 只能在启动实例时附加实例的实例存储卷。无法在启动实例后将实例存储卷附加到该实例。

在启动实例后，您必须先确保已格式化和挂载实例的实例存储卷，然后才能使用这些存储卷。将自动挂载实例存储根卷。

根卷注意事项

块设备映射始终指定实例的根卷。根卷始终自动挂载。

Linux 实例：根卷是一个 Amazon EBS 卷或实例存储卷。对于根卷的具有实例存储卷的实例，该卷的大小因 AMI 而异，但最大大小为 10 GB。有关更多信息，请参阅 根卷类型。

Windows 实例：根卷必须是 Amazon EBS 卷。根卷不支持实例存储。

内容

- 将实例存储卷添加到 Amazon EC2 AMI
- 启动期间将实例存储卷添加到 EC2 实例
- 使实例存储卷可供 EC2 实例使用

将实例存储卷添加到 Amazon EC2 AMI

您可创建带包括实例存储卷的块设备映射的 AMI。

如果使用指定实例存储卷块设备映射的 AMI 启动支持非 NVMe 实例存储卷的实例，该实例将包括实例存储卷。如果 AMI 中实例存储卷块设备映射的数量超过实例可用的实例存储卷的数量，则会忽略其他实例存储卷块设备映像。

如果使用指定实例存储卷块设备映射的 AMI 启动支持 NVMe 实例存储卷的实例，则会忽略实例存储卷块设备映射。无论在实例启动请求和 AMI 中指定的块设备映射如何，支持 NVMe 实例存储卷的实例都会获得其支持的所有实例存储卷。此类卷的设备映射取决于操作系统枚举这些卷的顺序。

注意事项

- 可用的实例存储卷数量取决于实例类型。有关更多信息，请参阅 the section called “可用的实例存储卷”。
- 必须为每个块设备指定一个设备名称。有关更多信息，请参阅 Amazon EC2 实例上卷的设备名称。
- 启动实例时，可忽略 AMI 块设备映射中指定的非 NVMe 实例存储卷，或添加实例存储卷。
- 对于 M3 实例，请在实例的块设备映射中指定实例存储卷，而非 AMI。Amazon EC2 可能会忽略 AMI 中的实例存储卷块设备映射。

Console

向 Amazon EBS-backed AMI 添加实例存储卷

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择实例。
3. 依次选择操作、映像和模板和创建映像。
4. 在创建映像页面上，为您的映像添加一个有意义的名称和描述。
5. 对于要添加的每个实例存储卷，选择添加新卷，从卷类型中选择实例存储卷，并从设备中选择设备名称。
6. 选择创建映像。

Amazon CLI

将实例存储卷添加到 AMI

使用 create-image 命令和 --block-device-mappings 选项，为 EBS-backed AMI 指定块设备映射。使用 register-image 命令和 --block-device-mappings 选项，为 iAmazon S3 支持的 AMI 指定块设备映射。

```
--block-device-mappings file://mapping.json
```

以下块设备映射会添加两个实例存储卷。

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

将实例存储卷添加到 AMI

使用 New-EC2Image cmdlet 和 -BlockDeviceMapping 参数，为 EBS-backed AMI 指定块设备映射。使用 Register-EC2Image cmdlet 和 -BlockDeviceMapping 参数，为 Amazon S3 支持的 AMI 指定块设备映射。

```
-BlockDeviceMapping $bdm
```

以下块设备映射会添加两个实例存储卷。

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
```

```
$bdm += $sdd
```

启动期间将实例存储卷添加到 EC2 实例

启动具有非 NVMe 实例存储卷的实例类型（例如，C1、C3、M1、M2、M3、R3、D2、H1、I2、X1 和 X1e）时，您必须在启动时为要附加的实例存储卷指定块设备映射。必须在实例启动请求中指定块设备映射，或者在用于启动实例的 AMI 中指定。

如果 AMI 包含实例存储卷的块设备映射，则无需在实例启动请求中指定块设备映射，除非您需要的实例存储卷超过 AMI 中包含的容量。

如果 AMI 不包含实例存储卷的块设备映射，则必须在实例启动请求中指定块设备映射。

对于具有 NVMe 实例存储卷的实例类型，所有支持的实例存储卷都会在启动时自动连接到实例。

注意事项

- 可用的实例存储卷数量取决于实例类型。有关更多信息，请参阅 the section called “可用的实例存储卷”。
- 必须为每个块设备指定一个设备名称。有关更多信息，请参阅 Amazon EC2 实例上卷的设备名称。
- 对于 M3 实例，即使您未在实例的块设备映射中指定实例存储卷，您也可能收到这些卷。

Console

在实例启动请求中指定块设备映射

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在控制面板中，选择启动实例。
3. 在 Application and OS Images（应用程序和操作系统映像）部分中，请选择要使用的 AMI。
4. 在配置存储部分中，实例存储卷部分列出了可附加到实例的实例存储卷。
5. 对于要附加的每个实例存储卷，对于设备名称，选择要使用的设备名称。
6. 根据需要配置剩余的实例设置，然后选择启动实例。

Amazon CLI

在实例启动请求中指定块设备映射

使用 run-instances 命令和 `--block-device-mappings` 选项。

```
--block-device-mappings file://mapping.json
```

以下块设备映射会添加两个实例存储卷。

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral0"
    },
    {
        "DeviceName": "/dev/sdd",
        "VirtualName": "ephemeral1"
    }
]
```

PowerShell

在实例启动请求中指定块设备映射

使用 New-EC2Instance cmdlet 和 `-BlockDeviceMapping` 选项。

```
-BlockDeviceMapping $bdm
```

以下块设备映射会添加两个实例存储卷。

```
$bdm = @()

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral0"
$bdm += $sdc

$sdd = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdd.DeviceName = "/dev/sdd"
$sdd.VirtualName = "ephemeral1"
$bdm += $sdd
```

使实例存储卷可供 EC2 实例使用

启动带有附加实例存储卷的实例后，必须先挂载这些卷，然后才能访问它们。

Linux 实例

您可以在启动实例后，使用您选择的文件系统将卷格式化。

使实例存储卷在 Linux 上可用

1. 使用 SSH 客户端连接到实例。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
2. 使用 `df -h` 命令查看已格式化并挂载的卷。

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   72K  3.8G   1% /dev
tmpfs           3.8G     0  3.8G   0% /dev/shm
/dev/nvme0n1p1  7.9G  1.2G  6.6G  15% /
```

3. 使用 `lsblk` 查看在启动时已映射但未格式化和装载的所有卷。

```
$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1    0    8G  0 disk
##nvme0n1p1   259:2    0    8G  0 part /
##nvme0n1p128 259:3    0    1M  0 part
nvme1n1       259:0    0 69.9G  0 disk
```

4. 要格式化并挂载仅映射的实例存储卷，请执行以下操作：

   a. 使用 `mkfs` 命令在设备上创建文件系统。

   ```
   $ sudo mkfs -t xfs /dev/nvme1n1
   ```

   b. 使用 `mkdir` 命令创建要将设备挂载到的目录。

   ```
   $ sudo mkdir /data
   ```

   c. 使用 `mount` 命令在新建目录上挂载设备。

   ```
   $ sudo mount /dev/nvme1n1 /data
   ```

Windows 实例

对于 Windows 实例，我们利用 NTFS 文件系统重新格式化实例存储卷。

您可以使用 Windows 磁盘管理来查看实例存储卷。有关更多信息，请参阅 列出非 NVMe 磁盘。

手动挂载实例存储卷

1. 选择开始，输入计算机管理，然后按 Enter。
2. 在左侧面板中，选择磁盘管理。
3. 如果系统提示您初始化卷，请选择要初始化的卷，根据使用案例选择所需的分区类型，然后选择确定。
4. 在卷列表中，右键单击要挂载的卷，然后选择新建简单卷。
5. 在向导中，选择下一步。
6. 在“指定卷大小”页面上，选择下一步，以使用最大卷大小。或者，也可以选择介于最小磁盘空间和最大磁盘空间之间的卷大小。
7. 在“分配驱动器号或路径”页面上，执行以下某项操作，然后选择下一步。
   - 要使用驱动器号挂载卷，请选择分配以下驱动器号，然后选择要使用的驱动器号。
   - 要将卷挂载为文件夹，请选择装入以下空白 NTFS 文件夹中，然后选择浏览，以创建或选择要使用的文件夹。
   - 要想不使用驱动器号或路径挂载卷，请选择不分配驱动器号或驱动器路径。
8. 在“格式化分区”页面上，请指定是否格式化卷。如果选择格式化卷，请选择所需的文件系统和单位大小，然后指定卷标。
9. 选择下一步、完成。

为 M1 和 C1 EC2 实例启用实例存储交换卷

> **Note**
>
> 本主题仅适用于 `c1.medium` 和 `m1.small` Linux 实例。

`c1.medium` 和 `m1.small` 实例类型的物理内存量有限。因此，启动时作为 Linux 系统虚拟内存的是 900 MiB 交换卷或交换空间。当系统所需内存超过实际分配内存时，可以在 Linux 中使用交换空间。启用交换空间后，Linux 系统可以将很少使用的内存页面从物理内存交换至交换空间（现有文件系统中的专用分区或交换文件），并为需要高速访问的内存页面释放空间。

> **Note**
>
> - 使用交换空间进行内存分页并不像使用 RAM 那样快速高效。如果您的工作负载定期将内存分页为交换空间，您应考虑迁移到具有更多 RAM 的较大实例类型。有关更多信息，请参阅 Amazon EC2 实例类型更改。
> - 尽管 Linux 内核将此交换空间看作根卷的一部分，但是它实际上是一个独立的实例存储卷，与根卷类型无关。

Amazon Linux 可以自动启用和使用此交换空间，但是您的 AMI 可能需要一些额外的步骤来识别和使用此交换空间。要查看您的实例是否正在使用交换空间，可以使用 swapon -s 命令。

```
[ec2-user ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/xvda3                              partition       917500  0       -1
```

上述实例拥有一个已附加并启用的 900 MiB 交换卷。如果您没有通过该命令看到列出的交换卷，则可能需要启用该设备的交换空间。使用 lsblk 命令检查您的可用磁盘。

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk
```

在这里，交换卷 xvda3 对该实例可用，但是尚未启用（请注意 MOUNTPOINT 字段为空）。您可以使用 swapon 命令启用交换卷。

> **Note**
>
> 您必须在 /dev/ 列出的设备名称前加上 lsblk。设备的命名可以不同，例如 sda3、sde3 或 xvde3。在以下命令中使用系统的设备名称。

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

现在交换空间应该显示在 lsblk 和 swapon -s 输出中。

```
[ec2-user ~]$ lsblk
```

```
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0    8G  0 disk /
xvda3 202:3    0  896M  0 disk [SWAP]
[ec2-user ~]$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/xvda3                              partition       917500  0       -1
```

您还需要编辑您的 /etc/fstab 文件，以便在每次系统启动时自动启用此交换空间。

```
[ec2-user ~]$ sudo vim /etc/fstab
```

将以下行附加到您的 /etc/fstab 文件中（使用系统的交换设备名称）：

```
/dev/xvda3       none    swap    sw  0       0
```

使用实例存储卷作为交换空间

所有实例存储卷都可用作交换空间。例如，m3.medium 实例类型包含一个适用于交换空间的 4 GB SSD 实例存储卷。如果您的实例存储卷大很多（例如 350GB），则可以考虑将卷分区为一个较小的 4-8GB 交换分区，其余部分用作数据卷。

> **Note**
>
> 此过程仅适用于支持实例存储的实例类型。有关受支持实例类型的列表，请参阅EC2 实例的实例存储卷限制。

1. 列出附加到您的实例的块设备以获取实例存储卷的设备名称。

   ```
   [ec2-user ~]$ lsblk -p
   NAME       MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   /dev/xvdb  202:16   0   4G  0 disk /media/ephemeral0
   /dev/xvda1 202:1    0   8G  0 disk /
   ```

   在此示例中，实例存储卷为 /dev/xvdb。因为这是 Amazon Linux 实例，所以实例存储卷在 /media/ephemeral0 处格式化并挂载；并不是所有 Linux 操作系统都自动执行这一操作。

2. （可选）如果您挂载了实例存储卷（它将在 MOUNTPOINT 命令输出中列出 lsblk），您需要使用以下命令卸载它。

```
[ec2-user ~]$ sudo umount /dev/xvdb
```

3. 使用 mkswap 命令在设备上设置一个 Linux 交换区域。

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swapspace version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. 启用新的交换空间。

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. 验证所使用的新交换空间。

```
[ec2-user ~]$ swapon -s
Filename    Type  Size Used Priority
/dev/xvdb                               partition 4188668 0 -1
```

6. 编辑您的 /etc/fstab 文件，以在每次系统启动时自动启用此交换空间。

```
[ec2-user ~]$ sudo vim /etc/fstab
```

如果您的 /etc/fstab 文件拥有 /dev/xvdb（或 /dev/sdb）条目，请将其更改为与下面的行匹配；如果没有针对此设备的条目，请将以下行附加到您的 /etc/fstab 文件（使用您系统的交换设备名称）：

```
/dev/xvdb       none    swap    sw  0       0
```

> **Important**
>
> 当实例停止或休眠后，实例存储卷数据将丢失；这包括在Step 3中创建的实例存储交换空间格式设置。如果您停止并重新启动已配置为使用实例存储交换空间的实例，则必须在新的实例存储卷上重复 Step 1 到 Step 5。

在 EC2 实例上初始化实例存储卷

由于 Amazon EC2 采用特殊方式将磁盘虚拟化，所以第一次在部分实例存储卷上执行写入操作的速度会比之后的写入操作慢。对于大部分应用程序，可将此成本分摊到实例的整个使用期限。然而，如果您需要较高的磁盘性能，我们建议您在生产使用之前对每个磁盘位置执行一次性写入操作，以此来实现硬盘初始化。

> **Note**
>
> 使用直接附加的固态硬盘（SSD）且支持 TRIM 的实例类型，可以在启动时提供最大性能，无需初始化。有关每种实例类型的实例存储的信息，请参阅EC2 实例的实例存储卷限制。

如果您需要在延迟或吞吐量方面具有更大灵活性，我们建议您使用 Amazon EBS。

要初始化实例存储卷，请使用以下 dd 命令，具体取决于要初始化的存储（例如 /dev/sdb 或 /dev/nvme1n1）。

> **Note**
>
> 请确保先卸载硬盘，然后再执行该命令。
> 初始化可能需要很长一段时间（对于超大型实例，约为 8 小时）。

要将实例存储卷初始化，可使用 m1.large、m1.xlarge、c1.xlarge、m2.xlarge、m2.2xlarge 和 m2.4xlarge 实例类型上的以下命令：

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

要同时对所有实例存储卷执行初始化，可使用以下命令：

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

配置硬盘以便通过对每个硬盘位置执行写入操作对其执行 RAID 初始化。当配置基于软件的 RAID 时，请务必更改最低重建速度：

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

Amazon EC2 实例存储卷的详细性能统计数据

Amazon EC2 为连接到基于 Nitro 的 Amazon EC2 实例的 NVMe 实例存储卷提供实时、高分辨率的性能统计数据。这些统计数据以聚合计数器的形式显示，将在实例的生命周期内保留。统计数据提供有关累积操作数、发送和接收的字节数、读取和写入 I/O 操作所花费的时间以及读取和写入 I/O 操作直方图的详细信息。这些统计数据与 Amazon EBS 详细性能统计数据保持一致，同时还包含按 I/O 大小细分的详细延迟直方图，可以更精细地洞察您的存储性能模式。这种增强的可见性使您能够精确识别哪些特定的 I/O 大小遇到了延迟问题，从而可以优化应用程序性能并更有效地解决问题。

您可以按最高 1 秒的间隔收集这些统计数据。如果发出请求的频率超过 1 秒的间隔，NVMe 驱动程序可能会将这些请求与其他管理命令一起排队，以便稍后处理。

注意事项

- 仅挂载到基于 Nitro 的实例的 NVMe 实例存储卷支持这些统计数据。
- 在实例停止和重启期间，计数器不会保留。
- 这些统计数据无需额外费用即可使用。

统计信息

NVMe 块设备提供以下统计信息：

统计数据名称	全名	Type	说明
total_read_ops	总读取操作数	计数器	已完成的读取操作总数。
total_write_ops	总写入操作数	计数器	已完成的写入操作总数。
total_read_bytes	总读取字节数	计数器	传输的读取字节总数。
total_write_bytes	总写入字节数	计数器	传输的写入字节总数。

统计数据名称	全名	Type	说明
total_rea d_time	总读取时间	计数器	所有已完成读取操作的总耗时（单位：微秒）。
total_wri te_time	总写入时间	计数器	所有已完成写入操作的总耗时（单位：微秒）。
instance_ store_vol ume_perfo rmance_ex ceeded_io ps	总需求时间超出卷的最大 IOPS	计数器	IOPS 请求超出卷最大 IOPS 的总时间（单位：微秒）。大于 0 的任何值都表明您的工作负载需要的 IOPS 超出卷的交付能力。理想情况下，该指标的增量计数在两次快照时间之间应最少。
instance_ store_vol ume_perfo rmance_ex ceeded_tp	总需求时间超出卷的最大吞吐量	计数器	吞吐量请求超出卷最大吞吐量的总时间（单位：微秒）。大于 0 的任何值都表明您的工作负载需要的吞吐量超出卷的交付能力。理想情况下，该指标的增量计数在两次快照时间之间应最少。
volume_qu eue_lengt h	卷队列长度	时间点	待完成的读取与写入操作的数量。
read_io_l atency_hi stogram	读取 I/O 直方图	直方图*	每个延迟区间内完成的读取操作数（单位：微秒）。
write_io_ latency_h istogram	写入 I/O 直方图	直方图*	每个延迟区间内完成的写入操作数（单位：微秒）。

> **Note**
>
> * 直方图统计数据仅表示成功完成的 I/O 操作数。停滞或受损的 I/O 操作不包括在内，但将在 `volume_queue_length` 统计数据中显示，以时间点统计数据的形式呈现。

访问统计数据

必须直接从实例存储卷所挂载的实例访问统计数据。您可以使用以下方法之一访问统计数据。

Linux 实例

Amazon CloudWatch

您可以将 Amazon CloudWatch 代理配置为从实例收集统计数据，然后将其作为自定义指标提供给 CloudWatch。然后，您可以使用 CloudWatch 中的指标来分析 I/O 模式、跟踪性能趋势、创建自定义控制面板以及根据性能阈值设置自动警报。

有关配置 CloudWatch 代理的更多信息，请参阅收集 Amazon EC2 实例存储卷指标。

nvme-cli tool

访问统计数据

1. 连接到卷所挂载的实例。
2. 2025 年 9 月 15 日之后发布的 Amazon Linux 2023 AMI 包含最新版本的 `nvme-cli` 工具。如果您使用的是较旧的 Amazon Linux AMI，请更新 `nvme-cli` 工具。

   ```
   sudo yum install nvme-cli
   ```
3. 运行以下命令并指定卷的设备名称。

   ```
   sudo nvme amzn stats /dev/nvme0n1
   ```

统计数据还提供了按 I/O 大小细分的详细延迟直方图。要查看按 I/O 大小细分的统计数据，请包含 `--details` 选项。例如：

```
sudo nvme amzn stats --details /dev/nvme0n1
```

通过指定 `--help` 选项，您可以获得有关如何使用该工具的更多信息。例如：

```
sudo nvme amzn stats --help
```

Windows 实例

nvme_amzn.exe tool

访问统计数据

1. 连接到卷所挂载的实例。
2. 确保您使用的是 AWSNVMe 驱动程序版本 `1.7.0` 或更高版本。有关更新 AWSNVMe 驱动程序的更多信息，请参阅 Amazon NVMe 驱动程序。
3. 获取卷的磁盘编号。有关更多信息，请参阅将 Amazon EC2 Windows 实例上的 NVMe 磁盘映射到卷。
4. 以管理员身份运行以下命令，并指定卷的磁盘编号。

```
.\nvme_amzn.exe stats disk_number
```

统计数据还提供了按 I/O 大小细分的详细延迟直方图。要查看按 I/O 大小细分的统计数据，请包含 `--details` 选项。例如：

```
.\nvme_amzn.exe stats --details disk_number
```

Amazon EC2 实例的根卷

当您启动一个实例时，我们会为该实例创建根卷。根卷包含用于启动该实例的映像。每个实例都有单独的根卷。您可以在启动期间或启动后向您的实例添加存储卷。

您用于启动实例的 AMI 决定根卷的类型。您可以从 Amazon EBS-backed AMI（Linux 和 Windows 实例）或 Amazon S3 支持的 AMI（仅限 Linux 实例）启动实例。使用每种 AMI 可进行的操作有很大区别。有关这些区别的更多信息，请参阅 根卷类型。

我们建议您使用由 Amazon EBS 提供支持的 AMI，因为这些实例启动速度更快，而且采用了持久性存储。

我们为根卷保留特定的设备名称。有关更多信息，请参阅 Amazon EC2 实例上卷的设备名称。

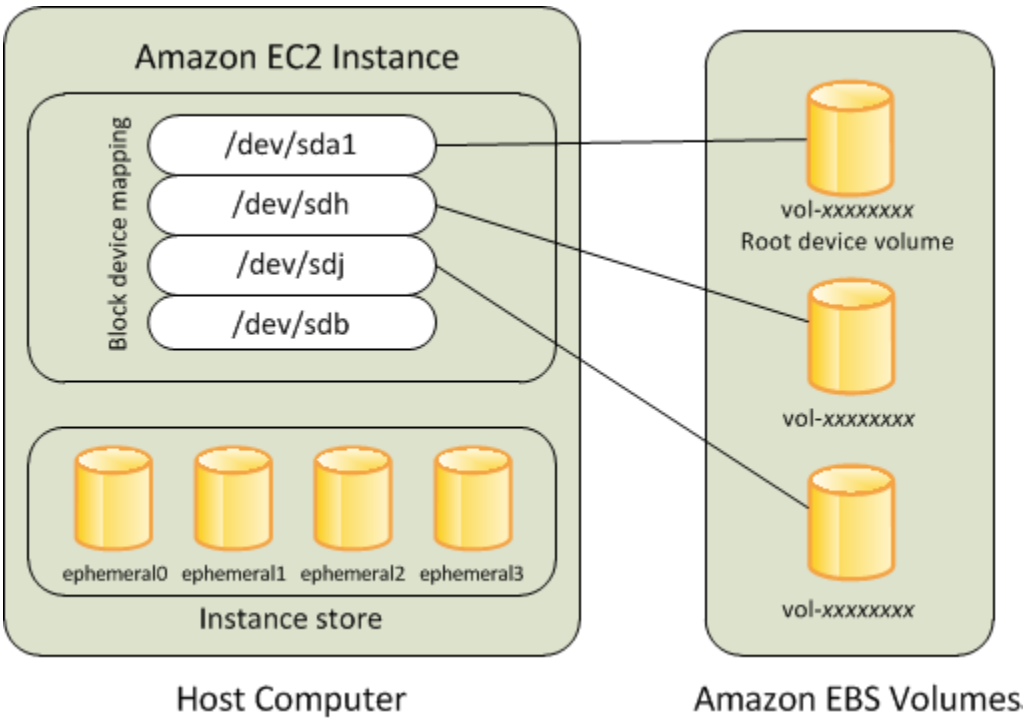
内容

- 具有 Amazon EBS 根卷的实例
- 具有实例存储根卷的实例（仅限 Linux 实例）
- Amazon EC2 实例终止后会保留 Amazon EBS 根卷
- 在不停止 Amazon EC2 实例的情况下替换该实例的根卷

具有 Amazon EBS 根卷的实例

使用 Amazon EBS 作为根卷的实例会自动附加一个 Amazon EBS 卷。当您启动具有 Amazon EBS-AMI 的实例时，系统会为 AMI 所引用的每一个 Amazon EBS 快照创建 Amazon EBS 卷。您可以根据实例类型选择使用其他Amazon EBS卷或实例存储卷。

具有 EBS 根卷的实例可以停止然后再重新启动，附加的卷中存储的数据不会受影响。当具有 EBS 根卷的实例处于停止状态时，您可以执行各种与该实例和卷有关的任务。例如，您可以修改实例的属性、更改其大小或更新其使用的内核，或者您可以将其根卷附加到另一个正在运行的实例，以进行调试或实现任何其他目的。有关更多信息，请参阅 Amazon EBS 卷。



限制

您不能使用 `st1` 或 `sc1` EBS 卷作为根卷。

实例失败

如果具有 EBS 根卷的实例失败，您可以通过以下方法之一恢复您的会话：

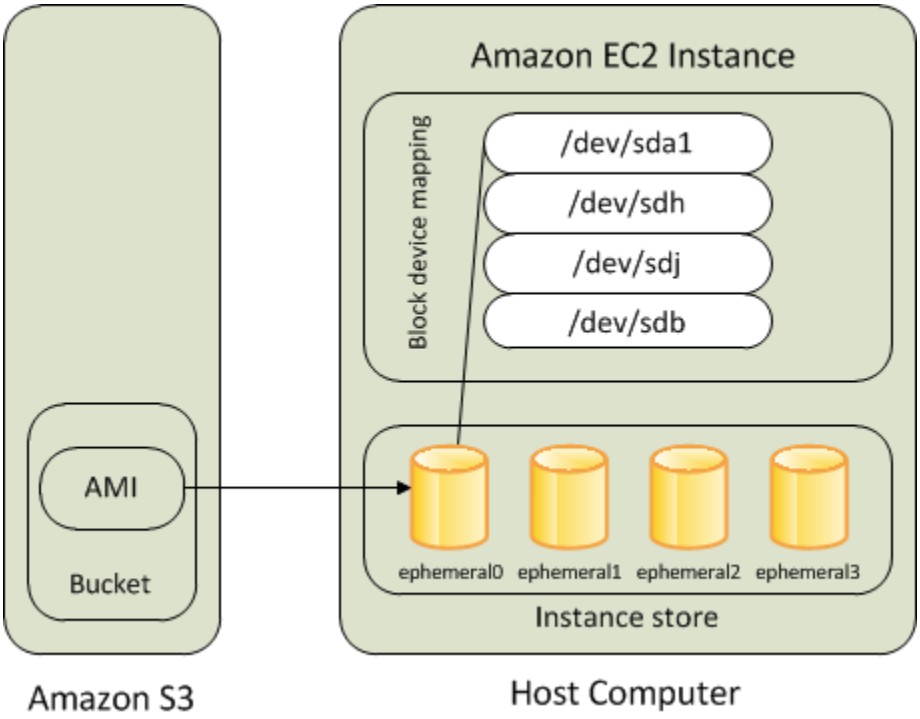
- 停止，然后再次启动（先尝试此方法）。
- 自动为相关卷拍摄快照并创建新的 AMI。有关更多信息，请参阅创建 Amazon EBS-backed AMI。
- 通过以下步骤将卷附加到新实例：
  1. 创建根卷的快照。
  2. 使用快照注册一个新的 AMI。
  3. 从新的 AMI 启动一个新实例。
  4. 将其余 Amazon EBS 卷与旧实例分离。
  5. 将这些 Amazon EBS 卷重新附加到新实例。

具有实例存储根卷的实例（仅限 Linux 实例）

> **Note**
>
> Windows 实例不支持实例存储根卷。

使用实例存储作为根卷的实例会自带可用的一个或多个实例存储卷，其中一个卷充当根卷。从 Amazon S3 支持的 AMI 启动实例时，即会将 AMI 复制到根卷。请注意，您可以根据实例类型选择使用更多实例存储卷。

只要实例正在运行，实例存储卷上的所有数据便会存在，但是在实例终止时（具有实例存储根卷的实例不支持停止操作）或是实例失败时（例如底层硬盘有问题时），会删除这些数据。有关更多信息，请参阅 适用于 EC2 实例的实例存储临时块存储。



支持的实例类型

只有以下实例类型支持将实例存储卷作为根卷：C1、C3、D2、I2、M1、M2、M3、R3 和 X1。

实例失败

具有实例存储根卷的实例失败或终止后，该实例不能被恢复。如果您打算使用具有实例存储根卷的实例，我们强烈建议您将数据跨多个可用区分散到不同的实例存储卷中。您还应该定期将您的实例存储卷上的关键数据备份至持久性存储。

Amazon EC2 实例终止后会保留 Amazon EBS 根卷

默认情况下，当实例终止时，实例的 Amazon EBS 根卷会被删除。您可以更改默认行为，确保 Amazon EBS 根卷在实例终止后仍保留。要更改默认行为，请将 `DeleteOnTermination` 属性设置为 `false`。您可以在实例启动时或稍后执行此操作。

任务

- 将根卷配置为在实例启动期间保留
- 配置根卷以便为现有实例保留
- 确认已将根卷配置为保留

将根卷配置为在实例启动期间保留

您可以将根卷配置为在启动实例时保留。

Console

在启动实例时将根卷配置为持久保留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例，然后选择启动实例。
3. 依次选择亚马逊机器映像（AMI）、实例类型、密钥对，然后配置网络设置。
4. 在配置存储中，选择高级。
5. 扩展根卷。
6. 对于终止时删除，选择是。
7. 配置完实例后，选择启动实例。

Amazon CLI

在启动实例时将根卷配置为持久保留

使用 run-instances 命令并包含以下选项。

```
--block-device-mappings file://mapping.json
```

在 mapping.json 中，指定一个将 DeleteOnTermination 属性设置为 false 的块设备映射。

```
[
    {
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "DeleteOnTermination": false
        }
    }
]
```

PowerShell

在启动实例时将根卷配置为持久保留

使用 New-EC2Instance cmdlet 并包含以下参数。

```
-BlockDeviceMapping $bdm
```

创建一个将 DeleteOnTermination 属性设置为 $false 的块设备映射。

```
$ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
$ebs.DeleteOnTermination = $false
$bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "dev/xvda"
$bdm.Ebs = $ebs
```

配置根卷以便为现有实例保留

您可以将根卷配置为对运行的实例保留。请注意，您无法使用 Amazon EC2 控制台完成此任务。

Amazon CLI

将现有实例的根卷配置为持久保留

使用 modify-instance-attribute 命令以及将 DeleteOnTermination 属性设置为 false 的块设备映射。

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --block-device-mappings file://mapping.json
```

在 mapping.json 中指定以下内容。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "DeleteOnTermination": false
        }
    }
]
```

PowerShell

将现有实例的根卷配置为持久保留

使用 Edit-EC2InstanceAttribute cmdlet 以及将 DeleteOnTermination 属性设置为 $false 的块设备映射。

```
$ebs = New-Object Amazon.EC2.Model.EbsInstanceBlockDeviceSpecification
$ebs.DeleteOnTermination = $false
$bdm = New-Object Amazon.EC2.Model.InstanceBlockDeviceMappingSpecification
$bdm.DeviceName = "/dev/xvda"
$bdm.Ebs = $ebs
Edit-EC2InstanceAttribute `
    -InstanceId i-1234567890abcdef0 `
    -BlockDeviceMapping $bdm
```

确认已将根卷配置为保留

您可以确认已将根卷配置为持久保留。

Console

确认已将根卷配置为持久保留

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例，然后选择实例。
3. 在存储选项卡的块设备下，找到根卷的条目。如果终止时删除为 No，则表示卷已配置为保留。

Amazon CLI

确认已将根卷配置为持久保留

使用 describe-instances 命令，并确认 DeleteOnTermination 属性设置为 false。

```
aws ec2 describe-instances \
    --instance-id i-1234567890abcdef0 \
    --query "Reservations[].Instances[].BlockDeviceMappings"
```

下面是示例输出。

```
[
    [
        {
            "DeviceName": "/dev/xvda",
```

```
            "Ebs": {
                "AttachTime": "2024-07-12T04:05:33.000Z",
                "DeleteOnTermination": false,
                "Status": "attached",
                "VolumeId": "vol-1234567890abcdef0"

        }
    ]
]
```

PowerShell

确认已将根卷配置为持久保留

使用 Get-EC2Instance cmdlet，并确认 `DeleteOnTermination` 属性设置为 `False`。

```
(Get-EC2Instance -InstanceId i-
i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

下面是示例输出。

```
AssociatedResource  :
AttachTime          : 7/12/2024 4:05:33 AM
DeleteOnTermination : False
Operator            :
Status              : attached
VolumeId            : vol-1234567890abcdef0
```

在不停止 Amazon EC2 实例的情况下替换该实例的根卷

Amazon EC2 使您能够替换正在运行的实例的根 Amazon EBS 卷，同时保留以下内容：

- 存储在实例存储卷上的数据 – 还原根卷后，实例存储卷仍附加到实例。
- 存储在数据（非根）Amazon EBS 卷上的数据 – 在还原根卷后，非根 Amazon EBS 卷仍附加到实例。
- 网络配置 – 所有网络接口均仍附加到实例，并保留其 IP 地址、标识符和附件 ID。当实例变为可用时，将刷新所有待处理的网络流量。此外，实例保留在同一个物理主机上，因此它会保留其公有和私有 IP 地址以及 DNS 名称。
- IAM policy – 将保留和强制实施与实例相关联的 IAM 配置文件和策略（例如基于标签的策略）。

内容

- 根卷替换的工作原理
- 注意事项
- 替换根卷

根卷替换的工作原理

当您替换实例的根卷时，将会创建根卷替换任务。原始根卷与实例分离，新根卷将在其位置附加到实例。将更新实例的块设备映射，以反映替换根卷的 ID。

替换实例的根卷时，必须指定新卷的快照源。以下是可能的选项。

将根卷还原到其原始状态

此选项将当前根卷替换为一个卷，该卷基于创建根卷时使用的快照。

使用启动状态的注意事项

替换根卷的类型、大小和终止时删除属性与原始根卷相同。

使用快照替换根卷

此选项将当前根卷替换为一个卷，该卷基于您指定的快照。例如，您之前从该根卷创建的特定快照。如果您需要从根卷损坏或客户操作系统中的网络配置错误引起的问题中恢复，这将非常有用。

替换根卷的类型、大小和终止时删除属性与原始根卷相同。

使用快照的注意事项

- 您只能使用直接从实例当前或以前的根卷创建的快照。
- 您不能使用从根卷拍摄的快照创建的快照副本。
- 成功替换根卷后，您仍可使用从原始根卷获取的快照来替换新的（替换）根卷。

使用 AMI 替换根卷

此选项使用您指定的 AMI 替换当前根卷。如果需要执行操作系统和应用程序修补或升级，这将非常有用。AMI 必须具有与实例相同的产品代码、账单信息、架构类型和虚拟化类型。

如果实例启用了 ENA 或 sriov-net，则必须使用支持这些功能的 AMI。如果实例未启用 ENA 或 sriov-net，则可以选择不支持这些功能的 AMI，或者如果选择支持 ENA 或 sriov net 的 AMI，则可以自动添加支持。

如果实例启用了 NitroTPM，则必须使用启用了 NitroTPM 的 AMI。无论选择哪种 AMI，如果没有实例配置它，都不会启用 NitrotPM 支持。

您可以选择与实例的启动模式不同的 AMI，前提是该实例支持 AMI 的启动模式。如果实例不支持启动模式，则该请求将失败。如果实例支持启动模式，则新的启动模式将传播到该实例，并相应更新其 UEFI 数据。如果您手动修改了启动顺序或添加了私有 UEFI 安全启动密钥来加载私有内核模块，则在根卷替换期间，更改将丢失。

替换根卷获得与原始根卷相同的卷类型和终止时删除属性，并获取 AMI 根卷块设备映射的大小。

> **Note**
>
> AMI 根卷块设备映射的大小必须等于或大于原始根卷的大小。如果 AMI 根卷块设备映射的大小小于原始根卷的大小，则请求失败。

在根卷替换任务完成后，当您使用控制台 Amazon CLI 或 Amazon SDK 描述实例时，将反映以下新的和经过更新的信息：

- 新 AMI ID
- 根卷的新卷 ID
- 经过更新的启动模式配置（如果由 AMI 更改）
- 经过更新的 NitroTPM 配置（如果由 AMI 启用）
- 经过更新的 ENA 配置（如果由 AMI 启用）
- 经过更新的 sriov-net 配置（如果由 AMI 启用）

新 AMI ID 也反映在实例元数据中。

使用 AMI 的注意事项：

- 如果您使用拥有多个块设备映射的 AMI，则仅使用该 AMI 的根卷。其他（非根）卷将被忽略。
- 只有当您拥有 AMI 及其关联根卷快照的权限时，才能使用此功能。您不能对 Amazon Web Services Marketplace AMI 使用此功能。
- 仅当实例没有产品代码时，您才能使用没有产品代码的 AMI。
- AMI 根卷块设备映射的大小必须等于或大于原始根卷的大小。如果 AMI 根卷块设备映射的大小小于原始根卷的大小，则请求失败。

- 将自动更新实例的实例身份文档。
- 如果实例支持 NitroTPM，则将重置该实例的 NitroTPM 数据，并将生成新密钥。

您可以选择在根卷替换过程完成后是否保留原始根卷。如果您在替换过程完成后选择删除原始根卷，则原始根卷将被自动删除，并变得不可恢复。如果您选择在该过程完成后保留原始根卷，则该卷仍会在您的账户中预调配；当您不再需要该卷时，必须手动将其删除。

根卷替换任务会经历以下几个状态：

- `pending`：正在创建替换卷。
- `in-progress`：正在分离原始卷，正在附加替换卷。
- `succeeded`：已将替换卷成功附加到实例并且实例可用。
- `failing`：替换任务即将失败。
- `failed`：替换任务已失败，但根卷仍处于附加状态。
- `failing-detached`：替换任务即将失败，并且实例可能未附加根卷。
- `failed-detached`：替换任务已失败，并且实例未附加根卷。

如果根卷替换任务失败，则实例将重新启动，原始根卷仍附加到该实例。

注意事项

开始之前，请注意以下因素：

要求

- 该实例必须处于 `running` 状态。
- 在此过程中，实例将自动重新启动。在重启过程中，将清空内存（RAM）的内容。无需手动重启。
- 如果根卷是实例存储卷，则无法替换它。仅支持具有 Amazon EBS 根卷的实例。
- 您可以替换所有虚拟化实例类型和 EC2 Mac 裸机实例的根卷。不支持其他裸机实例类型。
- 您只能使用直接从实例当前或以前的根卷创建的快照。
- 如果您的账户在当前区域启用了 Amazon EBS 默认加密密，无论指定快照的加密状态或指定 AMI 的根卷如何，由根卷替换任务创建的替换根卷始终处于加密状态。

加密结果

下表汇总了可能的加密结果。

	原始根卷	指定的快照或 AMI	默认加密	替换根卷	用于替换根卷的加密密钥
将替换根卷还原到初始启动状态	已加密	不适用	不考虑	已加密	与原始根卷相同的 KMS 密钥
	未加密	不适用	已禁用	未加密	不适用
	未加密	不适用	已启用	已加密	账户的 Amazon EBS 加密的默认 KMS 密钥
从快照或 AMI 还原替换根卷	已加密	未加密	不考虑	已加密	与原始根卷相同的 KMS 密钥
	已加密	已加密	不考虑	已加密	与原始根卷相同的 KMS 密钥
	未加密	未加密	已禁用	未加密	不适用
	未加密	未加密	已启用	已加密	账户的 Amazon EBS 加密的默认 KMS 密钥
	未加密	已加密	不考虑	已加密	如果 AMI 或快照归账户所有，则使用 AMI 或快照的 KMS 密钥对替换卷进行加密。如果与账户共享 AMI

	原始根卷	指定的快照或 AMI	默认加密	替换根卷	用于替换根卷的加密密钥
					或快照，则使用该账户的 Amazon EBS 加密的默认 KMS 密钥加密替换卷。

替换根卷

当您替换实例的根卷时，将创建根卷替换任务。您可以使用根卷更换任务来监控替换过程的进度和结果。

Console

替换根卷

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择要替换其根卷的实例，然后选择 Actions（操作）、Monitor and troubleshoot（监控和故障排除）、Replace root volume（替换根卷）。

   > **Note**
   >
   > 如果选定的实例不在 `running` 状态，替换根卷操作将被禁用。

4. 对于还原，选择以下选项之一：
   - 启动状态：从用于创建当前根卷的快照中还原替换根卷。
   - 快照：将替换根卷还原到您指定的快照。对于快照，选择要使用的快照。
   - 映像：使用您指定的 AMI 还原替换根卷。对于映像，选择要使用的 AMI。
5. （可选）对于卷初始化速率，您可以指定 Amazon EBS 预置卷初始化速率（卷初始化速率），单位为 MiB/s，这是将快照块从 Amazon S3 下载到卷的速率。有关更多信息，请参阅 Use an Amazon EBS Provisioned Rate for Volume Initialization。要使用默认初始化速率或快速快照还原功能（如果已为所选快照启用此功能），请不要指定速率。

6. （可选）要删除要替换的根卷，请选择删除替换的根卷。
7. 选择创建替换任务。
8. 要监控替换任务，请选择实例的存储选项卡，然后展开最近的根卷替换任务。

Amazon CLI

将替换根卷还原到启动状态

使用 create-replace-root-volume-task 命令。对于 `--instance-id`，指定要为其替换根卷的实例的 ID。省略 `--snapshot-id` 和 `--image-id` 参数。若要在替换原始根卷后将其删除，请包括 `--delete-replaced-root-volume` 并指定 `true`。要指定将快照块从 Amazon S3 下载到卷的卷初始化速率，则对于 `--volume-initialization-rate`，请指定一个介于 `100` 到 `300` MiB/s 之间的值。

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

将替换根卷还原到特定快照

使用 create-replace-root-volume-task 命令。对于 `--instance-id`，指定要为其替换根卷的实例的 ID。对于 `--snapshot-id`，请定要使用的快照的 ID。若要在替换原始根卷后将其删除，请包括 `--delete-replaced-root-volume` 并指定 `true`。要指定将快照块从 Amazon S3 下载到卷的卷初始化速率，则对于 `--volume-initialization-rate`，请指定一个介于 `100` 到 `300` MiB/s 之间的值。

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--snapshot-id snap-9876543210abcdef0 \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

使用 AMI 还原替换根卷

使用 create-replace-root-volume-task 命令。对于 `--instance-id`，指定要为其替换根卷的实例的 ID。对于 `--image-id`，指定要使用的 AMI 的 ID。若要在替换原始根卷后将其删除，请包括 `--delete-replaced-root-volume` 并指定 `true`。要指定将快照块从 Amazon S3 下载到卷的

卷初始化速率，则对于 `--volume-initialization-rate`，请指定一个介于 `100` 到 `300` MiB/s 之间的值。

```
aws ec2 create-replace-root-volume-task \
--instance-id i-1234567890abcdef0 \
--image-id ami-09876543210abcdef \
--delete-replaced-root-volume \
--volume-initialization-rate 150
```

查看根卷替换任务的状态

使用 describe-replace-root-volume-tasks 命令并指定要查看的根卷替换任务的 ID。

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-1234567890abcdef0 \
    --query ReplaceRootVolumeTasks[].TaskState
```

下面是示例输出。

```
[
    "succeeded"
]
```

或者，请指定 `instance-id` 筛选条件以按实例筛选结果。

```
$ aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-1234567890abcdef0
```

PowerShell

将替换根卷还原到启动状态

使用 New-EC2ReplaceRootVolumeTask 命令。对于 `-InstanceId`，指定要为其替换根卷的实例的 ID。省略 `-SnapshotId` 和 `-ImageId` 参数。若要在替换原始根卷后将其删除，请包括 `-DeleteReplacedRootVolume` 并指定 `$true`。要指定将快照块从 Amazon S3 下载到卷的卷初始化速率，则对于 `-VolumeInitializationRate`，请指定一个介于 `100` 到 `300` MiB/s 之间的值。

```
New-EC2ReplaceRootVolumeTask `
```

```
    -InstanceId i-1234567890abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

将替换根卷还原到特定快照

使用 New-EC2ReplaceRootVolumeTask 命令。对于 --InstanceId，指定要为其替换根卷的实例的 ID。对于 -SnapshotId，请定要使用的快照的 ID。若要在替换原始根卷后将其删除，请包括 -DeleteReplacedRootVolume 并指定 $true。要指定将快照块从 Amazon S3 下载到卷的卷初始化速率，则对于 -VolumeInitializationRate，请指定一个介于 100 到 300 MiB/s 之间的值。

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -SnapshotId snap-9876543210abcdef0 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

使用 AMI 还原替换根卷

使用 New-EC2ReplaceRootVolumeTask 命令。对于 -InstanceId，指定要为其替换根卷的实例的 ID。对于 -ImageId，指定要使用的 AMI 的 ID。若要在替换原始根卷后将其删除，请包括 -DeleteReplacedRootVolume 并指定 $true。要指定将快照块从 Amazon S3 下载到卷的卷初始化速率，则对于 -VolumeInitializationRate，请指定一个介于 100 到 300 MiB/s 之间的值。

```
New-EC2ReplaceRootVolumeTask `
    -InstanceId i-1234567890abcdef0 `
    -ImageId ami-0abcdef1234567890 `
    -VolumeInitializationRate 150 `
    -DeleteReplacedRootVolume $true
```

查看根卷替换任务的状态

使用 Get-EC2ReplaceRootVolumeTask 命令并指定要查看的根卷替换任务的 ID。

```
(Get-EC2ReplaceRootVolumeTask `
    -ReplaceRootVolumeTaskIds replacevol-1234567890abcdef0).TaskState.Value
```

下面是示例输出。

```
Succeeded
```

或者，请指定 `instance-id` 筛选条件以按实例筛选结果。

```
PS C:\> Get-EC2ReplaceRootVolumeTask -Filters @{Name = 'instance-id'; Values =
 'i-1234567890abcdef0'} | Format-Table
```

Amazon EC2 实例上卷的设备名称

当您将卷附加到实例时，需要为卷提供设备名称。该设备名称由 Amazon EC2 使用。实例的块储存设备驱动程序会在装载卷时分配实际的卷名称，指定的名称可以与 Amazon EC2 使用的名称不同。

您的实例可支持的卷的数量取决于操作系统。有关更多信息，请参阅Amazon EC2 实例的 Amazon EBS 卷限制。

目录

- 可用设备名称
- 设备名称注意事项

可用设备名称

Linux 实例

对 Linux 实例提供两种类型的虚拟化：半虚拟化（PV）和硬件虚拟机（HVM）。实例的虚拟化类型由用于启动实例的 AMI 确定。支持 HVM AMI 的所有实例类型。上一代的某些实例类型支持半虚拟化 AMI。请务必注意您的 AMI 的虚拟化类型，因为推荐的和您可以使用的可用设备名称取决于您的实例的虚拟化类型。有关更多信息，请参阅 虚拟化类型。

下表列出了在块储存设备映射中或附加 EBS 卷时您可指定的可用设备名称。

虚拟化类型	可用	为根卷预留	建议用于 EBS 数据卷	实例存储卷
半虚拟化	/dev/sd[a-z]	/dev/sda1	/dev/sd[f-p] /dev/sd[f-p][1-6]	/dev/sd[b-e]

虚拟化类型	可用	为根卷预留	建议用于 EBS 数据卷	实例存储卷
	/dev/sd[a-z] [1-15] /dev/hd[a-z] /dev/hd[a-z] [1-15]			
全虚拟化	/dev/sd[a-z] /dev/xvd[a-c][a-z] /dev/xvdd[a-x]	不同的 AMI /dev/sda1 或 /dev/xvda	/dev/sd[b-z] /dev/xvdb[b-z] *	/dev/sd[b-e] /dev/sd[b-h]（h1.16xlarge） /dev/sd[b-y]（d2.8xlarge） /dev/sd[b-i]（i2.8xlarge） **

* 您在块设备映射中为 NVMe EBS 卷指定的设备名称将使用 NVMe 设备名称（`/dev/nvme[0-26]n1`）进行重命名。块储存设备驱动程序可以使用不同于您在块储存设备映射中为卷指定的顺序来分配 NVMe 设备名称。

** 将自动枚举 NVMe 实例存储卷并为其分配 NVMe 设备名称。

Windows 实例

Amazon Windows AMI 使用其中一组驱动程序来允许访问虚拟化硬件：

- Amazon PV：Windows 实例的半虚拟化驱动程序
- Amazon NVMe：Amazon NVMe 驱动程序

基于 Nitro 的实例的设备名称

下表列出了可在块设备映射中或将 EBS 卷附加到基于 Nitro 的实例时指定的可用设备名称。

驱动程序类型	可用	为根卷预留	建议用于 EBS 卷	实例存储卷
Amazon NVMe	xvd[a-z] xvd[a-c][a-z] xvdd[a-x] /dev/sda1	/dev/sda1	xvd[b-z] xvdb[b-z]	*

*系统将自动枚举 NVMe 实例存储卷并为其分配 Windows 驱动器号。

基于 Xen 的实例的设备名称

下表列出了可在块设备映射中或将 EBS 卷附加到基于 Xen 的实例时指定的可用设备名称。

驱动程序类型	可用	为根卷预留	建议用于 EBS 卷	实例存储卷
Amazon PV	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]
Citrix PV（不再支持）	xvd[b-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-z]	xvdc[a-x] xvd[a-e]
Red Hat PV（不再支持）	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-p]	xvdc[a-x] xvd[a-e]

有关实例存储卷的更多信息，请参阅 适用于 EC2 实例的实例存储临时块存储。有关 NVMe EBS 卷（基于 Nitro 的实例）的更多信息，包括如何识别 EBS 设备，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 和 NVMe。

设备名称注意事项

在选择设备名称时请记住以下原则：

- 您使用的设备名称的结尾部分不应重叠，因为这可能会在您启动实例时导致问题。例如，避免对连接到同一实例的卷使用 /dev/xvdf 和 xvdf 之类的组合。
- 尽管您可以使用用于附加实例存储卷的设备名附加 EBS 卷，我们还是强烈建议您不要这样做，因为这种操作具有不可预测性。
- 实例的 NVMe 实例存储卷数取决于该实例的大小。将自动枚举 NVMe 实例存储卷并为其分配 NVMe 设备名称（Linux 实例）或 Windows 驱动器号（Windows 实例）。
- （Windows 实例）Amazon Windows AMI 附带其他软件，用于在首次启动时准备实例。此软件为 EC2Config 服务（Windows Server 2016 之前的 Windows AMI）或 EC2Launch（Windows Server 2016 及更高版本）。设备映射到驱动器后，驱动器即完成初始化和装载。根硬盘经过初始化后装载为 C:\。默认情况下，当 EBS 卷附加到 Windows 实例时，它在实例上可能会显示为任何盘符。您可以更改设置，以便按您的规范来设置卷的盘符。对于实例存储卷，默认设置则取决于驱动程序。AmazonPV 驱动程序和 Citrix PV 驱动程序以从 Z: 到 A: 的顺序分配实例存储卷驱动器盘符。Red Hat 驱动程序按 D: 到 Z: 的顺序分配实例存储卷盘符。有关更多信息，请参阅 Amazon EC2 Windows 实例上的 Windows 启动代理 和 如何为 Amazon EC2 Windows 实例附加和映射卷。
- （Linux 实例）根据内核的块设备驱动程序，附加的设备所采用的名称可能与您指定的名称不同。例如，如果您指定 /dev/sdh 的设备名称，则设备可能命名为 /dev/xvdh 或/dev/hdh。在大多数情况下，尾部字母保持不变。在某些版本的 Red Hat Enterprise Linux（及其变体，例如，CentOS）中，尾部字母可能发生变化（/dev/sda 可能变为 /dev/xvde）。在这些情况下，每个设备名称的尾部字母都会递增相同次数。例如，如果 /dev/sdb 重命名为 /dev/xvdf，则 /dev/sdc 重命名为 /dev/xvdg。Amazon Linux 为您对重命名设备指定的名称创建符号链接。其他操作系统的行为方式可能有所不同。
- （Linux 实例）HVM AMI 不支持在设备名称中使用尾部数字，除为根卷保留的 /dev/sda1 和 /dev/sda2 以外。尽管可以使用 /dev/sda2，但我们不建议将此设备映射与 HVM 实例结合使用。
- （Linux 实例）使用 PV AMI 时，您不能连接共享相同设备字母的卷，无论是否带有尾部数字都是如此。例如，如果您将一个卷附加为 /dev/sdc，另一个卷附加为 /dev/sdc1，则只有 /dev/sdc 将对实例可见。要在设备名称中使用尾部数字，您必须对所有基础字母相同的设备名称使用尾部数字（例如 /dev/sdc1、/dev/sdc2、/dev/sdc3）。

- （Linux 实例）一些自定义内核可能会包含限制，限制使用 /dev/sd[f-p] 或 /dev/sd[f-p][1-6]。如果您在使用 /dev/sd[q-z] 或 /dev/sd[q-z][1-6] 时遇到问题，请尝试切换为 /dev/sd[f-p] 或 /dev/sd[f-p][1-6]。

在指定所选设备名称之前，请确认其是否可用。否则，您将收到设备名称已被使用的错误。要查看磁盘设备及其挂载点，请使用 lsblk 命令（Linux 实例）、磁盘管理实用程序或 diskpart 命令（Windows 实例）。

Amazon EC2 实例上卷的块设备映射

您启动的每个实例都有一个关联根卷，它是 Amazon EBS 卷或实例存储卷。您可以使用块设备映射来指定实例启动时要连接的其他 EBS 卷或实例存储卷。您还可以将其他 EBS 卷附加到运行中的实例。然而，将实例存储卷附加到实例的唯一办法是，在实例启动时，使用块设备映射来附加这些卷。

内容

- 块储存设备映射的概念
- 将块设备映射添加到 AMI
- 将块设备映射添加到 Amazon EC2 实例

块储存设备映射的概念

块储存设备是一种以字节或位（块）为单位移动数据的存储设备。这些设备支持随机访问并广泛使用缓存 I/O，例如，包括硬盘、CD-ROM 驱动和闪存盘。块设备可以实际连接到计算机，或者就像实际连接到计算机一样对进行远程访问。

Amazon EC2 支持两种类型的块储存设备：

- 实例存储卷（虚拟设备，其底层硬件实际附加到该实例的主机）
- EBS 卷（远程存储设备）

块设备映射定义了挂载到某个实例的块储存设备（实例存储卷和 EBS 卷）。您可以指定块储存设备映射作为创建 AMI 的一部分，以便使从该 AMI 启动的所有实例均可使用该映射。或者，您还可以在启动实例时指定块储存设备映射，这样该映射会覆盖您在启动实例的 AMI 中指定的块储存设备映射。请注意，某个实例类型支持的所有 NVMe 实例存储卷将在实例启动时自动枚举并为其分配设备名称；将这些卷包含在您的块储存设备映射中不起作用。

目录

- 块储存设备映射条目
- 块储存设备映射实例存储注意事项
- 块储存设备映射示例
- 如何使设备在操作系统可用

块储存设备映射条目

当您创建块储存设备映射时，可以为需要附加到该实例的每个块储存设备指定以下信息：

- 在 Amazon EC2 内使用的设备名称。在装载卷时，实例的块储存设备驱动程序将分配实际卷名称。分配的名称可以与 Amazon EC2 建议的名称不同。有关更多信息，请参阅Amazon EC2 实例上卷的设备名称。

对于实例存储卷，您还可以指定以下信息：

- 虚拟设备：`ephemeral[0-23]`。请注意，对您的实例可用的实例存储卷的数量和大小因实例类型而异。

对于 NVMe 实例存储卷，以下信息也适用：

- 这些卷将自动枚举并分配设备名称；将这些卷包含在块储存设备映射中将不起作用。

对于 EBS 卷，您还可以指定以下信息：

- 用于创建块设备的快照的 ID（snap-xxxxxxxx）。只要您指定卷大小，此值为可选。您不能指定已归档快照的 ID。
- 卷的大小（以 GiB 计算）。所指定的大小必须大于等于指定快照的大小。
- 是否在实例终止时删除卷（`true` 或 `false`）。根卷的默认值为 `true`，附加的卷的默认值为 `false`。当您创建 AMI 时，其块储存设备映射会从该实例继承此设置。当您启动某个实例时，该实例会从 AMI 继承此设置。
- 卷类型：`gp2` 和（`gp3`适用于通用型 SSD）、`io1` 和（`io2`适用于预置 IOPS SSD）、`st1`（适用于吞吐量优化型 HDD）、`sc1`（适用于 Cold HDD）或（`standard`适用于磁介质卷）。
- 该卷支持的每秒进行读写操作的次数（IOPS）。（仅与 `io1` 和 `io2` 卷一起使用。）

- 某些实例类型支持多个 EBS 卡。您可以通过指定 EBS 卡索引来选择要附加的卷的 EBS 卡。有关更多信息，请参阅 EBS 卡。

块储存设备映射实例存储注意事项

使用在其块储存设备映射中具有实例存储卷的 AMIs 启动实例时，要考虑一些注意事项。

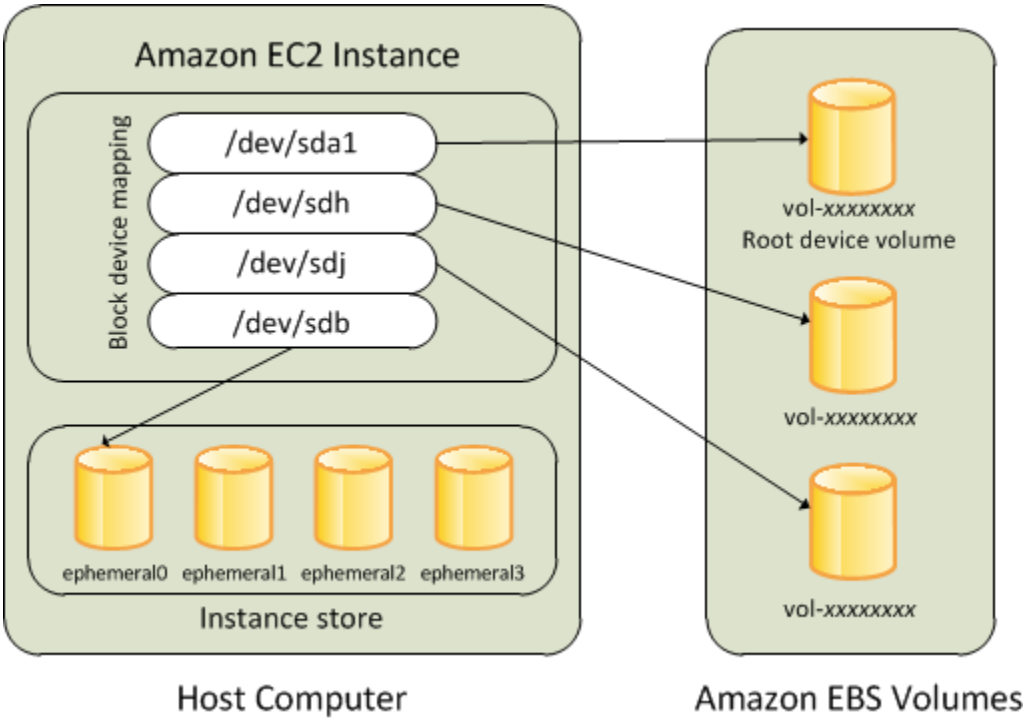
- 有些实例类型包含的实例存储卷多于其他类型，而有些实例类型根本不包含实例存储卷。如果实例类型支持一个实例存储卷，而且 AMI 具有用于两个实例存储卷的映射，则实例会在启动时带有一个实例存储卷。
- 实例存储卷只能在启动时进行映射。不能停止没有实例存储卷的实例（例如 `t2.micro`），将实例更改为支持实例存储卷的类型，然后重新启动带有实例存储卷的实例。但是，您可以从实例创建 AMI 并以支持实例存储卷的实例类型启动它，然后将这些实例存储卷映射到实例。
- 如果您启动映射了实例存储卷的实例，然后停止实例，将它更改为具有较少实例存储卷的实例类型并重新启动它，则来自初始启动的实例存储卷映射会出现在实例元数据中。但是，实例使用的实例存储卷不能超出该实例类型支持的最大数量。

> **Note**
>
> 实例停止时，实例存储卷上的所有数据都会丢失。

- 根据启动时的实例存储容量，M3 实例可能会在启动时忽略 AMI 实例存储块设备映射（除非在启动时指定它们）。您应在启动时指定实例存储块储存设备映射（即使启动的 AMI 在 AMI 中映射了实例存储卷），以确保实例存储卷在实例启动时可用。

块储存设备映射示例

此图显示了 EBS 支持的实例的块储存设备映射示例。它将 `/dev/sdb` 映射到 `ephemeral0` 并映射两个 EBS 卷，一个映射到 `/dev/sdh`，另一个映射到 `/dev/sdj`。它还显示了作为根卷的 EBS 卷，即 `/dev/sda1`。



请注意，此块储存设备映射示例是在本主题中的示命令和 API 中例使用的。您可以在为 AMI 指定块储存设备映射和在启动实例时更新块储存设备映射中找到创建块储存设备映射的示例命令和 API。

如何使设备在操作系统可用

Amazon EC2 使用设备名称（如 `/dev/sdh` 和 `xvdh`）来描述块储存设备。Amazon EC2 使用块储存设备映射来指定要附加到 EC2 实例的块储存设备。当块储存设备附加到实例后，您必须先将其装载到操作系统，然后才可以访问此存储设备。当块储存设备从实例分离后，就被操作系统卸载下来，而您也不能再访问该存储设备。

Linux 实例：该实例首次启动时，在块设备映射中指定的设备名称会被映射到其相应的块设备。默认情况下，实例类型决定要格式化并装载哪个实例存储卷。您可以在启动时装载额外的实例存储卷，前提是不得超过您的实例类型所允许的实例存储卷数量。有关更多信息，请参阅适用于 EC2 实例的实例存储临时块存储。实例的块储存设备驱动程序决定在格式化和装载卷时要使用哪些设备。

Windows 实例：首次启动实例时，将在块设备映射中指定的设备名称映射到其相应的块设备，然后 Ec2Config 服务会初始化并挂载这些驱动器。根卷装载为 `C:\`。实例存储卷装载为 `Z:\`、`Y:\` 等。安装 EBS 卷时，可使用任何可用盘符进行装载。然而，您可以配置如何将驱动器号分配给 EBS 卷；要了解更多信息，请参阅 the section called “Windows 启动代理”。

将块设备映射添加到 AMI

各个 AMI 都拥有块储存设备映射，指定实例启动时要附加的块储存设备。要向 AMI 添加更多块储存设备，必须创建自己的 AMI。

目录

- 为 AMI 指定块储存设备映射
- 查看 AMI 块储存设备映射中的 EBS 卷

为 AMI 指定块储存设备映射

创建 AMI 时，您可以使用两种方法来指定除根卷以外的卷。如果您在从该实例创建 AMI 前已将卷附加到运行中的实例，则 AMI 的块储存设备映射将包括这些相同的卷。对于 EBS 卷，这些现存的数据会保存在一个新的快照中，而且是块储存设备映射指定的新快照。而实例存储卷的数据无法保存。

对于 EBS 支持的 AMI，您可以使用块储存设备映射来添加 EBS 卷和实例存储卷。对于 Amazon S3 支持的 AMI，您只能添加实例存储卷，方法是在注册镜像时修改镜像清单文件中的块设备映射条目。

> **Note**
>
> 对于 M3 实例，您必须在启动实例时，在块储存设备映射中指定适用于实例的实例存储卷。当您启动 M3 实例时，如果在块储存设备映射中为 AMI 指定的实例存储卷未指定为块储存设备映射的一部分，则该卷可能会被忽略。

Console

将卷添加到 AMI

1. 打开 Amazon EC2 控制台。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择一个实例，再依次选择 Actions（操作）Image and templates（映像和模板）、Create image（创建映像）。
4. 输入映像的名称和描述。
5. 实例卷将显示在 Instance volumes（实例卷）下。要添加另一个卷，请选择 Add volume（添加卷）。

6. 对于 Volume Type（卷类型），选择卷类型。对于 Device（设备），请选择设备名称。对于 EBS 卷，您可以指定其他详细信息，例如快照、卷大小、卷类型、IOPS 和加密状态。
7. 选择创建映像。

Amazon CLI

将卷添加到 AMI

使用 create-image 命令可为由 EBS 支持的 AMI 指定块设备映射。使用 register-image 命令可为 Amazon S3 支持的 AMI 指定块设备映射。

使用 `--block-device-mappings` 参数指定块储存设备映射。您可以直接通过命令行指定 JSON 编码格式的参数，也可以通过引用 JSON 文件指定参数，如下所示。

```
--block-device-mappings file://mapping.json
```

要添加实例存储卷，请使用以下映射。请注意，NVMe 实例存储卷会自动挂载。

```
{
    "DeviceName": "device_name",
    "VirtualName": "ephemeral0"
}
```

要添加空的 100 GiB 卷，请使用以下映射。

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "VolumeSize": 100
    }
}
```

要添加基于快照的 EBS 卷，请使用以下映射。

```
{
    "DeviceName": "device_name",
    "Ebs": {
      "SnapshotId": "snap-1234567890abcdef0"
    }
}
```

要对设备省略映射，请使用以下映射。

```
{
    "DeviceName": "device_name",
    "NoDevice": ""
}
```

PowerShell

使用 New-EC2Image cmdlet 为由 EBS 支持的 AMI 指定块设备映射。使用 Register-EC2Image cmdlet，为 Amazon S3 支持的 AMI 指定块设备映射。

添加 `-BlockDeviceMapping` 选项，并在 `bdm` 中指定更新：

```
-BlockDeviceMapping $bdm
```

以下映射添加了一个基于快照的卷。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.SnapshotId = "snap-1234567890abcdef0"
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

以下映射添加了一个 100 GB 的空卷。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.VolumeSize = 100
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "device_name"
$bdm.Ebs = $ebd
```

查看 AMI 块储存设备映射中的 EBS 卷

您可以轻松列举块储存设备映射中适用于 AMI 的 EBS 卷。

Console

查看 AMI 的 EBS 卷

1. 打开 Amazon EC2 控制台。

2. 在导航窗格中，选择 AMIs。
3. 从 Filter 列表中选择 EBS images 以获取 EBS 支持的 AMI 的列表。
4. 选择所需的 AMI，然后查看详细信息选项卡。至少，以下信息适用于根卷（其中术语根设备等同于根卷）：

   - 根设备类型（ebs）
   - 根设备名称（例如 /dev/sda1）
   - 块设备（例如 /dev/sda1=snap-1234567890abcdef0:8:true）

   如果使用块设备映射创建的 AMI 带有额外 卷，则块设备字段会显示针对这些额外 EBS 卷的映射。（此屏幕不显示实例存储卷。）

Amazon CLI

查看 AMI 的 EBS 卷

使用 describe-images 命令。

```
aws ec2 describe-images \
    --image-ids ami-0abcdef1234567890 \
    --query Image[0].BlockDeviceMappings
```

PowerShell

查看 AMI 的 EBS 卷

使用 Get-EC2Image cmdlet。

```
(Get-EC2Image -ImageId ami-0abcdef1234567890).BlockDeviceMappings
```

将块设备映射添加到 Amazon EC2 实例

默认情况下，您启动的实例包含所有在 AMI 的块设备映射中指定的存储设备（您是从该 AMI 启动实例的）。您可以在启动实例时，为实例指定要对块储存设备映射执行的更改，而这些更新会覆盖 AMI 的块储存设备映射或与其合并。

限制

- 对于根卷，您只能修改下列内容：卷大小、卷类型和 Delete on Termination 标志。
- 修改 EBS 卷时，无法减小其大小。因此，您必须指定大小等于或大于 AMI 的块储存设备映射中指定的快照大小的快照。

任务

- 在启动实例时更新块储存设备映射
- 更新正在运行的实例的块储存设备映射
- 查看实例块储存设备映射中的 EBS 卷
- 查看实例存储卷的实例块储存设备映射

在启动实例时更新块储存设备映射

您可以在启动实例时向其添加 EBS 卷和实例存储卷。请注意，针对实例更新块储存设备映射不会对启动实例的 AMI 的块储存设备映射造成永久性更改。

Console

在启动时更新实例的卷

1. 按照启动实例的步骤操作，但在完成以下步骤更新卷前不要启动实例。
2. （可选）要添加卷，请选择配置存储、添加新卷。选择卷大小和卷类型。
3. （可选）要隐藏由 AMI 的块设备映射指定的卷，请选择配置存储、移除。
4. （可选）要修改 EBS 卷的配置，请在配置存储窗格上选择高级。展开该卷的信息，然后根据需要进行任何更改。
5. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

Amazon CLI

在启动时更新实例的卷

使用 run-instances 命令和 `--block-device-mappings` 选项。

```
--block-device-mappings file://mapping.json
```

例如，假定某个 AMI 块设备映射指定的设置如下：

- /dev/xvda – EBS 根卷
- /dev/sdh – 从 snap-1234567890abcdef0 创建的 EBS 卷
- /dev/sdj – 空 EBS 卷，大小为 100
- /dev/sdb – 实例存储卷 ephemeral0

假设以下为 mapping.json 中的实例块设备映射。

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "VolumeSize": 100
        }
    },
    {
        "DeviceName": "/dev/sdj",
        "NoDevice": ""
    },
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 300
        }
    },
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral1"
    }
]
```

此实例块设备映射会执行以下操作：

- 覆盖根卷 /dev/xvda 的大小，将其增加到 100 GiB。
- 阻止 /dev/sdj 连接到该实例。
- 覆盖 /dev/sdh 的大小，将其增加到 300 GiB。请注意，不需要再次指定快照 ID。
- 添加临时卷 /dev/sdc。如果实例类型不支持多个实例存储卷，则这将无效。如果实例类型支持 NVMe 实例存储卷，则会自动枚举这些卷并将其包含在实例块设备映射中，并且无法覆盖。

PowerShell

在启动时更新实例的卷

使用 New-EC2Instance cmdlet 以及 -BlockDeviceMapping 参数和 -BlockDeviceMapping 参数。

```
-BlockDeviceMapping $bdm
```

假设以下为 $bdm 中的实例块设备映射。

```
$bdm = @()

$root = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$root.DeviceName = "/dev/xvda"
$ebs1 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs1.VolumeSize = 100
$root.Ebs = $ebs1
$bdm += $root

$sdj = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdj.DeviceName = "/dev/sdj"
$sdj.NoDevice = ""
$bdm += $sdj

$sdh = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdh.DeviceName = "/dev/sdh"
$ebs2 = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebs2.VolumeSize = 300
$sdh.Ebs = $ebs2
$bdm += $sdh

$sdc = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$sdc.DeviceName = "/dev/sdc"
$sdc.VirtualName = "ephemeral1"
$bdm += $sdc
```

此实例块设备映射会执行以下操作：

- 覆盖根卷 /dev/xvda 的大小，将其增加到 100 GiB。
- 阻止 /dev/sdj 连接到该实例。
- 覆盖 /dev/sdh 的大小，将其增加到 300 GiB。请注意，不需要再次指定快照 ID。

- 添加临时卷 /dev/sdc。如果实例类型不支持多个实例存储卷，则这将无效。如果实例类型支持 NVMe 实例存储卷，则会自动枚举这些卷并将其包含在实例块设备映射中，并且无法覆盖。

更新正在运行的实例的块储存设备映射

在更改此属性之前，您不需要停止该实例。

Amazon CLI

更新正在运行的实例的块设备映射

使用 modify-instance-attribute 命令。

添加 --block-device-mappings 选项：

```
--block-device-mappings file://mapping.json
```

在 mapping.json 中，指定更新。例如，以下更新会将根卷更改为持久保留。

```
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
```

PowerShell

更新正在运行的实例的块设备映射

使用 Edit-EC2InstanceAttribute cmdlet。

添加 -BlockDeviceMapping 选项：

```
-BlockDeviceMapping $bdm
```

在 bdm 中，指定更新。例如，以下更新会将根卷更改为持久保留。

```
$ebd = New-Object -TypeName Amazon.EC2.Model.EbsBlockDevice
$ebd.DeleteOnTermination = false
```

```
$bdm = New-Object -TypeName Amazon.EC2.Model.BlockDeviceMapping
$bdm.DeviceName = "/dev/sda1"
$bdm.Ebs = $ebd
```

查看实例块储存设备映射中的 EBS 卷

您可以轻松枚举映射到实例的 EBS 卷。

Console

查看实例的 EBS 卷

1. 打开 Amazon EC2 控制台。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后查看存储选项卡中显示的详细信息。至少，以下信息适用于根卷（其中术语根设备等同于根卷）：

   - Root device type（例如 EBS ）
   - Root Device Name（例如，/dev/xvda）
   - Block devices（例如，/dev/xvda、/dev/sdf 和 /dev/sdj）

   如果使用块储存设备映射启动了带有其他 EBS 卷的实例，则它们将显示在 Block devices（块储存设备）下。此选项卡上不会显示任何实例存储卷。
4. 要显示有关 EBS 卷的其他信息，请选择其卷 ID 以转到卷页面。

Amazon CLI

查看实例的 EBS 卷

可以使用 describe-instances 命令。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[*].Instances[0].BlockDeviceMappings
```

PowerShell

查看实例的 EBS 卷

使用 Get-EC2Instance cmdlet。

```
(Get-EC2Instance -InstanceId i-0bac57d7472c89bac).Instances.BlockDeviceMappings
```

查看实例存储卷的实例块储存设备映射

实例类型将决定对实例可用的实例存储卷的数量和类型。如果块储存设备映射中的实例存储卷数超过了对实例可用的实例存储卷数，则其他卷将被忽略。要查看实例的实例存储卷，请运行 lsblk 命令（Linux 实例）或打开 Windows 磁盘管理（Windows 实例）。要了解每种实例类型支持的实例存储卷数，请参阅 Amazon EC2 instance type specifications。

当您查看实例的块储存设备映射时，可以只查看 EBS 卷，但是不能查看实例存储卷。您用于查看实例的实例存储卷的方法取决于卷类型。

NVMe 实例存储卷

Linux 实例

您可以使用 NVMe 命令行程序包 nvme-cli 来查询块储存设备映射中的 NVMe 实例存储卷。运行以下命令，下载程序包，并在实例上安装该程序包。

```
[ec2-user ~]$ sudo nvme list
```

以下是实例的示例输出。“模型”列中的文本表示卷是 EBS 卷还是实例存储卷。在这个例子中，/dev/nvme1n1 和 /dev/nvme2n1 均为实例存储卷。

```
Node             SN                   Model
 Namespace
---------------- -------------------- ----------------------------------------
 ---------
/dev/nvme0n1     vol06afc3f8715b7a597 Amazon Elastic Block Store               1

/dev/nvme1n1     AWS2C1436F5159EB6614 Amazon EC2 NVMe Instance Storage         1

/dev/nvme2n1     AWSB1F4FF0C0A6C281EA Amazon EC2 NVMe Instance Storage         1
  ...
```

Windows 实例

您可以使用磁盘管理或 PowerShell 列出 EBS 和实例存储 NVMe 卷。有关更多信息，请参阅 the section called “将 NVME 磁盘映射到卷”。

HDD 或 SSD 实例存储卷

您可以使用实例元数据在块储存设备映射中查询 HDD 或 SSD 实例存储卷。未包含 NVMe 实例储存卷。

所有针对实例元数据的请求的基本 URI 均为 `http://169.254.169.254/latest/`。有关更多信息，请参阅 使用实例元数据管理 EC2 实例。

Linux 实例

首先，连接到运行中的实例。从该实例中，使用此查询获取其块储存设备映射。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/block-device-mapping/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

该响应包含实例的块储存设备名称。举例来说，由实例存储支持的 `m1.small` 实例的输出如下所示。

```
ami
ephemeral0
root
swap
```

`ami` 设备是实例所看到的根卷。实例存储卷命名为 `ephemeral[0-23]`。`swap` 设备用于存储页面文件。如果您还映射了一些 EBS 卷，它们会依次显示为 `ebs1`、`ebs2` 等。

要了解块储存设备映射中的单个块储存设备的详细信息，可将其名称添加到上述查询，如下所示。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

Windows 实例

首先，连接到运行中的实例。从该实例中，使用此查询获取其块储存设备映射。

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/
```

该响应包含实例的块储存设备名称。举例来说，由实例存储支持的 m1.small 实例的输出如下所示。

```
ami
ephemeral0
root
swap
```

ami 设备是实例所看到的根卷。实例存储卷命名为 ephemeral[0-23]。swap 设备用于存储页面文件。如果您还映射了一些 EBS 卷，它们会依次显示为 ebs1、ebs2 等。

要了解块储存设备映射中的单个块储存设备的详细信息，可将其名称添加到上述查询，如下所示。

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

如何为 Amazon EC2 Windows 实例附加和映射卷

> **Note**
>
> 本主题仅适用于 Windows 实例。

您的 Windows 实例附带一个作为根卷的 EBS 卷。如果您的 Windows 实例使用 Amazon PV 或 Citrix PV 驱动程序，您可以选择最多添加 25 个卷，最终一共连接 26 个卷。有关更多信息，请参阅Amazon EC2 实例的 Amazon EBS 卷限制。

根据您的实例的实例类型，您将有 0 到 24 个可能的实例存储卷可用于实例。要使用可用于您的实例的任何实例存储卷，您必须在创建 AMI 或启动实例时指定这些卷。您还可以在创建 AMI 或启动实例时添加 EBS 卷，或在您的实例正在运行时连接这些卷。

将卷添加到实例时，您可以指定 Amazon EC2 使用的设备名称。有关更多信息，请参阅Amazon EC2 实例上卷的设备名称。AmazonWindows 亚马逊机器映像（AMI）包含 Amazon EC2 所使用的一系列驱动程序，用于将实例存储和 EBS 卷映射到 Windows 磁盘和驱动器盘符。

将磁盘映射到 EBS 卷的方法

- 将 Amazon EC2 Windows 实例上的 NVMe 磁盘映射到卷
- 将 Amazon EC2 Windows 实例上的非 NVMe 磁盘映射到卷

将 Amazon EC2 Windows 实例上的 NVMe 磁盘映射到卷

对于基于 Nitro 的实例，EBS 卷显示为 NVMe 设备。本主题介绍如何查看您的实例上可用于 Windows 操作系统的 NVMe 磁盘。它还展示了如何将这些 NVMe 磁盘映射到底层 Amazon EBS 卷以及为 Amazon EC2 使用的块设备映射指定的设备名称。

主题

- 列出 NVMe 磁盘
- 将 NVMe 磁盘映射到卷

列出 NVMe 磁盘

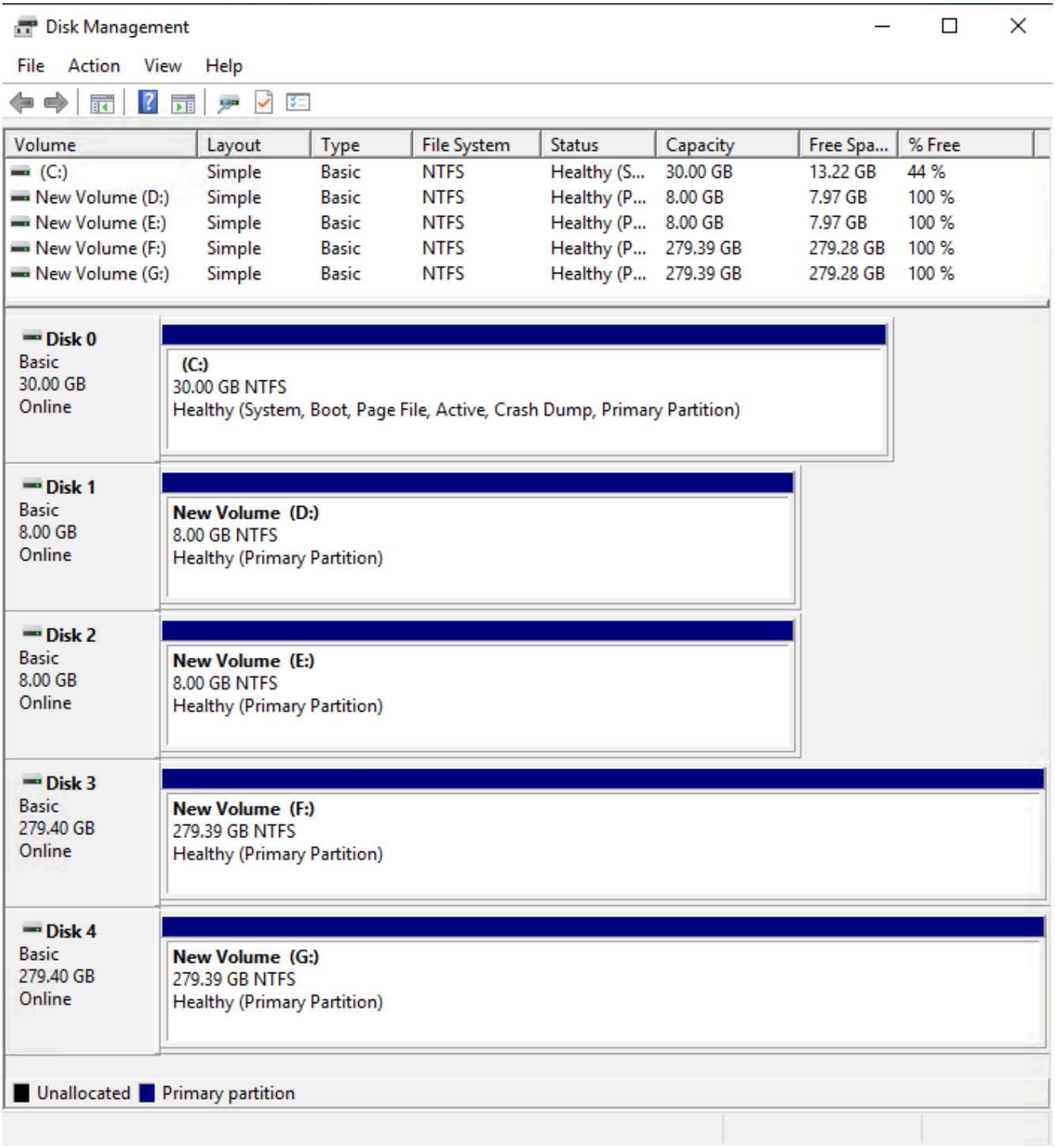
您可以使用磁盘管理工具或 Powershell，在您的 Windows 实例上查找磁盘。

Disk Management

查找 Windows 实例上的磁盘

1. 使用远程桌面登录 Windows 实例。有关更多信息，请参阅使用 RDP 连接到 Windows 实例。
2. 启动磁盘管理实用工具。
3. 检查磁盘。根卷是一个装载为 `C:\` 的 EBS 卷。如果未显示其他磁盘，表示您在创建 AMI 或启动实例时没有指定其他卷。

下面的示例介绍了在您启动一个包含两个其他 EBS 卷的 `r5d.4xlarge` 实例时可用的磁盘。



PowerShell

以下 PowerShell 脚本列出每个磁盘以及相应的设备名称和卷。旨在与基于 Nitro 的实例结合使用，此类实例使用 NVMe EBS 和实例存储卷。

连接到 Windows 实例并运行以下命令以启用 PowerShell 脚本执行。

```
Set-ExecutionPolicy RemoteSigned
```

复制以下脚本并将其保存为 Windows 实例 `mapping.ps1` 上。

```
# List the disks for NVMe volumes

function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function GetEBSVolumeId {
    param($Path)
    $SerialNumber = (Get-Disk -Path $Path).SerialNumber
    if($SerialNumber -clike 'vol*'){
        $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("vol","vol-")
    }
    else {
       $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("AWS","AWS-")
    }
    return $EbsVolumeId
}

function GetDeviceName{
    param($EbsVolumeId)
    if($EbsVolumeId -clike 'vol*'){

        $Device  = ((Get-EC2Volume -VolumeId $EbsVolumeId ).Attachment).Device
        $VolumeName = ""
    }
     else {
        $Device = "Ephemeral"
        $VolumeName = "Temporary Storage"
    }
    Return $Device,$VolumeName
}
```

```
function GetDriveLetter{
    param($Path)
    $DiskNumber =  (Get-Disk -Path $Path).Number
    if($DiskNumber -eq 0){
        $VirtualDevice = "root"
        $DriveLetter = "C"
        $PartitionNumber = (Get-Partition -DriveLetter C).PartitionNumber
    }
    else
    {
        $VirtualDevice = "N/A"
        $DriveLetter = (Get-Partition -DiskNumber $DiskNumber).DriveLetter
        if(!$DriveLetter)
        {
            $DriveLetter = ((Get-Partition -DiskId $Path).AccessPaths).Split(",")[0]
        }
        $PartitionNumber = (Get-Partition -DiskId $Path).PartitionNumber
    }

    return $DriveLetter,$VirtualDevice,$PartitionNumber

}

$Report = @()
foreach($Path in (Get-Disk).Path)
{
    $Disk_ID = ( Get-Partition -DiskId $Path).DiskId
    $Disk = ( Get-Disk -Path $Path).Number
    $EbsVolumeId  = GetEBSVolumeId($Path)
    $Size =(Get-Disk -Path $Path).Size
    $DriveLetter,$VirtualDevice, $Partition = (GetDriveLetter($Path))
    $Device,$VolumeName = GetDeviceName($EbsVolumeId)
    $Disk = New-Object PSObject -Property @{
      Disk          = $Disk
      Partitions    = $Partition
      DriveLetter   = $DriveLetter
      EbsVolumeId   = $EbsVolumeId
      Device        = $Device
      VirtualDevice = $VirtualDevice
      VolumeName= $VolumeName
    }
 $Report += $Disk
}
```

```
$Report | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, VolumeName
```

按如下方式编辑脚本：

```
PS C:\> .\mapping.ps1
```

以下是具有一个根卷、两个 EBS 卷和两个实例存储卷的实例的输出示例。

```
Disk Partitions DriveLetter EbsVolumeId           Device    VirtualDevice VolumeName
---- ---------- ----------- -----------           ------    ------------- ----------
   0          1 C           vol-03683f1d861744bc7 /dev/sda1 root
   1          1 D           vol-082b07051043174b9 xvdb      N/A
   2          1 E           vol-0a4064b39e5f534a2 xvdc      N/A
   3          1 F           AWS-6AAD8C2AEEE1193F0 Ephemeral N/A           Temporary
 Storage
   4          1 G           AWS-13E7299C2BD031A28 Ephemeral N/A           Temporary
 Storage
```

如果您没有在 Windows 实例上为 Tools for Windows PowerShell 提供凭证，则脚本将无法获取 EBS 卷 ID，并会在 `EbsVolumeId` 列中使用 N/A。

将 NVMe 磁盘映射到卷

您可以使用 Get-Disk 命令将 Windows 磁盘编号映射到 Amazon EBS 卷和 Amazon EC2 实例存储卷。

```
PS C:\> Get-Disk
Number Friendly Name Serial Number                   HealthStatus
 OperationalStatus      Total Size Partition

                      Style
------ ------------- -------------                   ------------
 -----------------      ---------- ----------
3      NVMe Amazo... AWS6AAD8C2AEEE1193F0_00000001.   Healthy              Online
              279.4 GB MBR
4      NVMe Amazo... AWS13E7299C2BD031A28_00000001.   Healthy              Online
              279.4 GB MBR
2      NVMe Amazo... vol0a4064b39e5f534a2_00000001.   Healthy              Online
                  8 GB MBR
```

```
0      NVMe Amazo... vol03683f1d861744bc7_00000001.   Healthy              Online
                30 GB MBR
1      NVMe Amazo... vol082b07051043174b9_00000001.   Healthy              Online
                 8 GB MBR
```

您还可以运行 ebsnvme-id 命令将 nVMe 磁盘号映射到 EBS 卷 ID 和设备名称。

```
PS C:\> C:\PROGRAMDATA\Amazon\Tools\ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-03683f1d861744bc7
Device Name: sda1

Disk Number: 1
Volume ID: vol-082b07051043174b9
Device Name: xvdb

Disk Number: 2
Volume ID: vol-0a4064b39e5f534a2
Device Name: xvdc
```

将 Amazon EC2 Windows 实例上的非 NVMe 磁盘映射到卷

对于从使用 Amazon PV 或 Citrix PV 驱动程序的 Windows AMI 启动的实例，可以使用本页介绍的关系，将 Windows 磁盘映射到实例存储和 EBS 卷。本主题介绍如何查看您的实例上可用于 Windows 操作系统的非 NVMe 磁盘。它还展示了如何将这些非 NVMe 磁盘映射到底层 Amazon EBS 卷以及为 Amazon EC2 使用的块设备映射指定的设备名称。

> **Note**
>
> 如果启动实例，并且如果 Windows AMI 使用 Red Hat PV 驱动程序，则可以更新实例来使用 Citrix 驱动程序。有关更多信息，请参阅 the section called “升级半虚拟化驱动程序”。

主题

- 列出非 NVMe 磁盘
- 将非 NVMe 磁盘映射到卷

列出非 NVMe 磁盘

您可以使用磁盘管理工具或 PowerShell 在您的 Windows 实例上查找磁盘。

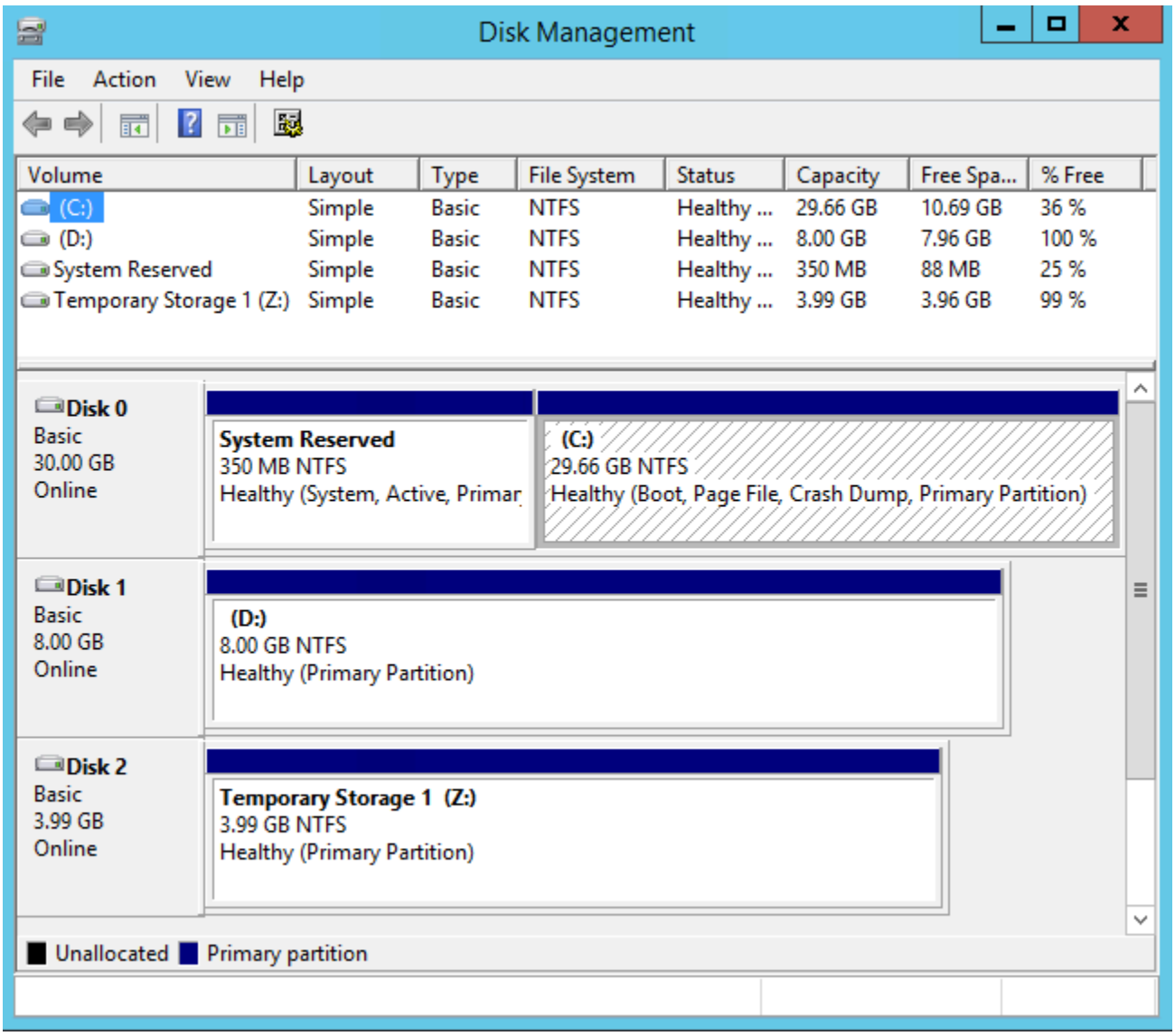
Disk Management

查找 Windows 实例上的磁盘

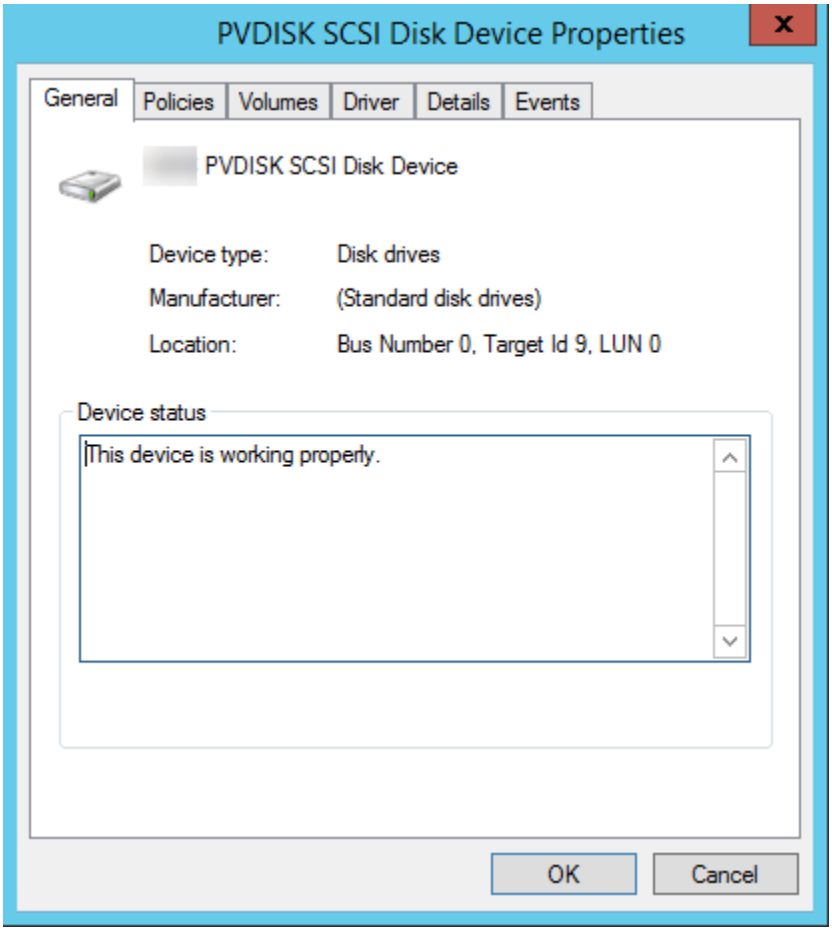
1. 使用远程桌面登录 Windows 实例。有关更多信息，请参阅使用 RDP 连接到 Windows 实例。
2. 开启磁盘管理实用工具。

   在任务栏上，右键单击 Windows 徽标，然后选择磁盘管理。
3. 检查磁盘。根卷是一个装载为 `C:\` 的 EBS 卷。如果未显示其他磁盘，表示您在创建 AMI 或启动实例时没有指定其他卷。

   下面的示例介绍了在您启动一个包含实例存储卷（磁盘 2）和其他 EBS 卷（磁盘 1）的 `m3.medium` 实例时可用的磁盘。



4. 右键单击标记为“磁盘 1”的灰色窗格，然后选择属性。记下位置值，并在将非 NVMe 磁盘映射到卷中的表中查找该值。例如，以下磁盘的位置值为“总线编号 0”、“目标 ID 9”、“LUN 0”。根据 EBS 卷的表，此位置的设备名称为 `xvdj`。



PowerShell

以下 PowerShell 脚本列出每个磁盘以及相应的设备名称和卷。

要求和限制

- 需要 Windows 服务器 2012 或更高版本。
- 需要凭据才能获取 EBS 卷 ID。您可以使用 Tools for PowerShell 配置配置文件，也可以将 IAM 角色附加到实例。
- 不支持 nVMe 卷。
- 不支持动态磁盘。

连接到 Windows 实例并运行以下命令以启用 PowerShell 脚本执行。

```
Set-ExecutionPolicy RemoteSigned
```

复制以下脚本并将其保存为 Windows 实例 `mapping.ps1` 上。

```
# List the disks
function Convert-SCSITargetIdToDeviceName {
  param([int]$SCSITargetId)
  If ($SCSITargetId -eq 0) {
    return "sda1"
  }
  $deviceName = "xvd"
  If ($SCSITargetId -gt 25) {
    $deviceName += [char](0x60 + [int]($SCSITargetId / 26))
  }
  $deviceName += [char](0x61 + $SCSITargetId % 26)
  return $deviceName
}

[string[]]$array1 = @()
[string[]]$array2 = @()
[string[]]$array3 = @()
[string[]]$array4 = @()

Get-WmiObject Win32_Volume | Select-Object Name, DeviceID | ForEach-Object {
  $array1 += $_.Name
  $array2 += $_.DeviceID
}

$i = 0
While ($i -ne ($array2.Count)) {
  $array3 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).SerialNumber) -replace "_[^ ]*$" -replace "vol", "vol-"
  $array4 += ((Get-Volume -Path $array2[$i] | Get-Partition | Get-
Disk).FriendlyName)
  $i ++
}

[array[]]$array = $array1, $array2, $array3, $array4

Try {
  $InstanceId = Get-EC2InstanceMetadata -Category "InstanceId"
  $Region = Get-EC2InstanceMetadata -Category "Region" | Select-Object -
ExpandProperty SystemName
```

```
}
Catch {
  Write-Host "Could not access the instance Metadata using AWS Get-
EC2InstanceMetadata CMDLet.
Verify you have AWSPowershell SDK version '3.1.73.0' or greater installed and
 Metadata is enabled for this instance." -ForegroundColor Yellow
}
Try {
  $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
 $InstanceId).Instances.BlockDeviceMappings
  $VirtualDeviceMap = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping").GetEnumerator() | Where-Object { $_.Key -ne "ami" }
}
Catch {
  Write-Host "Could not access the AWS API, therefore, VolumeId is not available.
Verify that you provided your access keys or assigned an IAM role with adequate
 permissions." -ForegroundColor Yellow
}

Get-disk | ForEach-Object {
  $DriveLetter = $null
  $VolumeName = $null
  $VirtualDevice = $null
  $DeviceName = $_.FriendlyName

  $DiskDrive = $_
  $Disk = $_.Number
  $Partitions = $_.NumberOfPartitions
  $EbsVolumeID = $_.SerialNumber -replace "_[^ ]*$" -replace "vol", "vol-"
  if ($Partitions -ge 1) {
    $PartitionsData = Get-Partition -DiskId $_.Path
    $DriveLetter = $PartitionsData.DriveLetter | Where-object { $_ -notin @("",
 $null) }
    $VolumeName = (Get-PSDrive | Where-Object { $_.Name -in
 @($DriveLetter) }).Description | Where-object { $_ -notin @("", $null) }
  }
  If ($DiskDrive.path -like "*PROD_PVDISK*") {
    $BlockDeviceName = Convert-SCSITargetIdToDeviceName((Get-WmiObject -Class
 Win32_Diskdrive | Where-Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" +
 $DiskDrive.Number) }).SCSITargetId)
    $BlockDeviceName = "/dev/" + $BlockDeviceName
    $BlockDevice = $BlockDeviceMappings | Where-Object { $BlockDeviceName -like "*"
 + $_.DeviceName + "*" }
    $EbsVolumeID = $BlockDevice.Ebs.VolumeId
```

```
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON_EC2_NVME*") {
    $BlockDeviceName = (Get-EC2InstanceMetadata -Category
 "BlockDeviceMapping")."ephemeral$((Get-WmiObject -Class Win32_Diskdrive | Where-
Object { $_.DeviceID -eq ("\\.\PHYSICALDRIVE" + $DiskDrive.Number) }).SCSIPort - 2)"
    $BlockDevice = $null
    $VirtualDevice = ($VirtualDeviceMap.GetEnumerator() | Where-Object { $_.Value -
eq $BlockDeviceName }).Key | Select-Object -First 1
  }
  ElseIf ($DiskDrive.path -like "*PROD_AMAZON*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  ElseIf ($DiskDrive.path -like "*NETAPP*") {
    if ($DriveLetter -match '[^a-zA-Z0-9]') {
      $i = 0
      While ($i -ne ($array3.Count)) {
        if ($array[2][$i] -eq $EbsVolumeID) {
          $DriveLetter = $array[0][$i]
          $DeviceName = $array[3][$i]
        }
        $i ++
      }
    }
    $EbsVolumeID = "FSxN Volume"
    $BlockDevice = ""
    $BlockDeviceName = ($BlockDeviceMappings | Where-Object { $_.ebs.VolumeId -eq
 $EbsVolumeID }).DeviceName
  }
  Else {
    $BlockDeviceName = $null
```

```
    $BlockDevice = $null
  }
  New-Object PSObject -Property @{
    Disk          = $Disk;
    Partitions    = $Partitions;
    DriveLetter   = If ($DriveLetter -eq $null) { "N/A" } Else { $DriveLetter };
    EbsVolumeId   = If ($EbsVolumeID -eq $null) { "N/A" } Else { $EbsVolumeID };
    Device        = If ($BlockDeviceName -eq $null) { "N/A" } Else
 { $BlockDeviceName };
    VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
    VolumeName    = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
    DeviceName    = If ($DeviceName -eq $null) { "N/A" } Else { $DeviceName };
  }
} | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions,
 DriveLetter, EbsVolumeId, Device, VirtualDevice, DeviceName, VolumeName
```

按如下方式编辑脚本：

```
PS C:\> .\mapping.ps1
```

下面是示例输出。

```
Disk  Partitions  DriveLetter   EbsVolumeId            Device      VirtualDevice
 DeviceName              VolumeName
----  ----------  -----------   -----------            ------      -------------
 ----------              ----------
   0           1            C   vol-0561f1783298efedd   /dev/sda1   N/A
 NVMe Amazon Elastic B   N/A
   1           1            D   vol-002a9488504c5e35a   xvdb        N/A
 NVMe Amazon Elastic B   N/A
   2           1            E   vol-0de9d46fcc907925d   xvdc        N/A
 NVMe Amazon Elastic B   N/A
```

如果您没有在 Windows 实例上提供凭据，则脚本将无法获取 EBS 卷 ID，并会在 EbsVolumeId 列中使用 N/A。

将非 NVMe 磁盘映射到卷

在装载卷时，实例的块储存设备驱动程序将分配实际卷名称。

映像

- 实例存储卷
- EBS 卷

实例存储卷

下表描述了 Citrix PV 和 Amazon PV 驱动程序是如何将非 NVMe 实例存储卷映射到 Windows 卷的。可用的实例存储卷数量由实例类型决定。有关更多信息，请参阅EC2 实例的实例存储卷限制。

位置	设备名称
总线编号 0，目标 ID 78，LUN 0	xvdca
总线编号 0，目标 ID 79，LUN 0	xvdcb
总线编号 0，目标 ID 80，LUN 0	xvdcc
总线编号 0，目标 ID 81，LUN 0	xvdcd
总线编号 0，目标 ID 82，LUN 0	xvdce
总线编号 0，目标 ID 83，LUN 0	xvdcf
总线编号 0，目标 ID 84，LUN 0	xvdcg
总线编号 0，目标 ID 85，LUN 0	xvdch
总线编号 0，目标 ID 86，LUN 0	xvdci
总线编号 0，目标 ID 87，LUN 0	xvdcj
总线编号 0，目标 ID 88，LUN 0	xvdck
总线编号 0，目标 ID 89，LUN 0	xvdcl

EBS 卷

下表描述了 Citrix PV 和 Amazon PV 驱动程序是如何将非 NVME EBS 卷映射到 Windows 卷的。

位置	设备名称
总线编号 0，目标 ID 0，LUN 0	/dev/sda1
总线编号 0，目标 ID 1，LUN 0	xvdb
总线编号 0，目标 ID 2，LUN 0	xvdc
总线编号 0，目标 ID 3，LUN 0	xvdd
总线编号 0，目标 ID 4，LUN 0	xvde
总线编号 0，目标 ID 5，LUN 0	xvdf
总线编号 0，目标 ID 6，LUN 0	xvdg
总线编号 0，目标 ID 7，LUN 0	xvdh
总线编号 0，目标 ID 8，LUN 0	xvdi
总线编号 0，目标 ID 9，LUN 0	xvdj
总线编号 0，目标 ID 10，LUN 0	xvdk
总线编号 0，目标 ID 11，LUN 0	xvdl
总线编号 0，目标 ID 12，LUN 0	xvdm
总线编号 0，目标 ID 13，LUN 0	xvdn
总线编号 0，目标 ID 14，LUN 0	xvdo
总线编号 0，目标 ID 15，LUN 0	xvdp
总线编号 0，目标 ID 16，LUN 0	xvdq
总线编号 0，目标 ID 17，LUN 0	xvdr
总线编号 0，目标 ID 18，LUN 0	xvds
总线编号 0，目标 ID 19，LUN 0	xvdt

位置	设备名称
总线编号 0，目标 ID 20，LUN 0	xvdu
总线编号 0，目标 ID 21，LUN 0	xvdv
总线编号 0，目标 ID 22，LUN 0	xvdw
总线编号 0，目标 ID 23，LUN 0	xvdx
总线编号 0，目标 ID 24，LUN 0	xvdy
总线编号 0，目标 ID 25，LUN 0	xvdz

Amazon EC2 Linux 实例上的防撕裂写入

> **Note**
>
> 仅 Linux 实例支持防撕裂写入。

撕裂写防护是由 Amazon 设计的块存储功能，可提高 I/O 密集型关系数据库工作负载的性能并减少延迟，而不会对数据弹性产生负面影响。使用 InnoDB 或 XtraDB 作为数据库引擎的关系数据库，例如 MySQL 和 MariaDB，将受益于撕裂写防护这一功能。

通常，使用大于存储设备断电原子性的页面的关系数据库使用数据记录机制来防止撕裂写。MariaDB 和 MySQL 在将数据写入数据表之前使用双写缓冲区文件来记录数据。如果写入事务期间由于操作系统崩溃或断电导致写入不完整或撕裂，数据库可以从双写缓冲区恢复数据。与写入双写缓冲区相关的 I/O 额外开销会影响数据库性能和应用程序延迟，并减少每秒可处理的事务数量。有关双写缓冲区的更多信息，请参阅 MariaDB 和 MySQL 文档。

借助撕裂写防护功能，可以在全有或全无的写入事务中将数据写入存储，这样就无需使用双写缓冲区。这样可以防止在写入事务期间因操作系统崩溃或断电将部分或撕裂的数据写入存储。在不影响工作负载弹性的情况下，每秒处理的事务数量最多可增加 30%，写入延迟最多可减少 50%。

定价

使用撕裂写防护功能不会产生额外费用。

内容

- Amazon EC2 上防撕裂写入的块大小
- 在 Amazon EC2 上使用防撕裂写入的要求
- 检查 Amazon EC2 实例对防撕裂写入的支持
- 配置 Amazon EC2 上工作负载的防撕裂写入

Amazon EC2 上防撕裂写入的块大小

撕裂写防护支持 4 KiB、8 KiB 和 16 KiB 数据块的写入操作。数据块起始逻辑块地址（LBA）必须与 4 KiB、8 KiB 或 16 KiB 的相应块的边界大小对齐。例如，对于 16 KiB 写入操作，数据块起始 LBA 必须与 16 KiB 的块边界大小对齐。

下表显示了跨存储和实例类型的支持。

	4 KiB 块	8 KiB 块	16 KiB 块
实例存储卷	所有 NVMe 实例存储卷都附加到当前 I 代新系列实例。	Amazon Nitro SSD 支持的 I4i、Im4gn、Is4gen、I7i、I7ie、I8g 和 I8ge 实例。	
Amazon EBS 卷	挂载到基于 nitro 的实例的所有 Amazon EBS 卷。		

要确认您的实例和卷是否支持撕裂写防护，请查询以检查该实例是否支持撕裂写防护以及其他详细信息，例如支持的块和边界大小。有关更多信息，请参阅 检查 Amazon EC2 实例对防撕裂写入的支持。

在 Amazon EC2 上使用防撕裂写入的要求

要使撕裂防护功能正常运行，I/O 操作必须符合 NTWPU、NTWGU、NTWBU 字段中指定的大小、对齐方式和边界要求。在提交到设备之前，必须配置操作系统，确保特定的存储子系统（文件系统、LVM、RAID 等）不会修改存储堆栈中的 I/O 属性，包括块合并、拆分或块地址重新定位。

已使用以下配置测试了撕裂写防护功能：

- 支持所需块大小的实例类型和存储类型。
- Amazon Linux 2 或更高版本（内核版本为 5.10）。
- 已启用 `bigalloc` 的 ext4，集群大小为 16 KiB，还有最新的 ext4 实用程序（e2fsprogs 1.46.5 或更高版本）。

- O_DIRECT 文件访问模式可绕过 Linux 内核缓冲区缓存。

> **Note**
>
> 您无需禁用 MySQL 和 MariaDB 工作负载的 I/O 合并。

检查 Amazon EC2 实例对防撕裂写入的支持

要确认您的实例和卷是否支持防撕裂写入，并查看包含防撕裂写入信息的 NVMe 命名空间供应商特定数据，请使用以下命令。

```
$ sudo nvme id-ns -v device_name
```

> **Note**
>
> 该命令以带有 ASCII 解释的十六进制形式返回供应商特定信息。您可能需要在应用程序中构建类似 ebsnvme-id 的工具，以读取和解析输出。

例如，以下命令返回 NVMe 命名空间供应商特定数据，其中包含 /dev/nvme1n1 的防撕裂写入信息。

```
$ sudo nvme id-ns -v /dev/nvme1n1
```

如果您的实例和卷支持防撕裂写入，其会在 NVMe 命名空间供应商特定数据中返回以下 Amazon 防撕裂写入信息。

> **Note**
>
> 下表中的字节表示从 NVMe 命名空间供应商特定数据开头起的偏移量（以字节为单位）。

字节	说明
0:31	设备附件挂载点的名称，例如 /dev/xvda 。您在卷附件请求期间提供此信息，Amazon EC2 实例可用其来创建指向 NVMe 区块设备的符号链接（nvmeXn1）。
32:63	卷 ID。例如 vol01234567890abcdef 。此字段可用于将 NVMe 设备映射到附加的卷。
64:255	留待将来使用。
256:257	命名空间撕裂写防护单元大小（NTWPU）。此字段表示在断电或发生错误时，保证以原子方式写入 NVM 的写入操作的命名空间特定大小。此字段在逻辑块中指定，从以零开始的值表示。
258:259	命名空间撕裂写防护粒度大小（NTWPG）。此字段表示在断电或发生错误时，保证以原子方式写入 NVM 的写入操作的 NTWPU 下的命名空间特定大小增量。也就是说，大小应为 NTWPG * n <= NTWPU，其中 n 为正整数。写入操作 LBA 偏移量也必须与该字段对齐。此字段在逻辑块中指定，从以零开始的值表示。
260:263	命名空间撕裂写防护边界大小（NTWPB）。此字段表示该 NTWPU 值的此命名空间的原子边界大小。在断电或发生错误时，无法将保证跨原子边界写入此命名空间的操作以原子方式写入 NVM。值为 0h 表示断电或发生错误时没有原子边界。所有其他值使用与 NTWPU 字段相同的编码以逻辑块形式指定大小。

配置 Amazon EC2 上工作负载的防撕裂写入

具有支持卷的受支持的实例类型上默认启用撕裂写防护功能。您无需进行任何其他设置即可启用防撕裂写入。

> **Note**
>
> 不支持防撕裂写入的工作负载不会受到性能影响。您无需对这些工作负载进行任何更改。
> 支持防撕裂写入但未配置使用该功能的工作负载会继续使用双写缓冲区，并且不会获得任何性能优势。

要将 MySQL 或 MariaDB 软件堆栈配置为禁用双写缓冲区并使用撕裂写防护功能，请完成以下步骤：

1. 将您的卷配置为使用带有 BigAlloc 选项的 ext4 文件系统，并将集群大小设置为 4 KiB、8 KiB 或 16 KiB。使用集群大小为 4 KiB、8 KiB 或 16 KiB 的 BigAlloc，可确保文件系统分配与各自边界对齐的文件。

   ```
   $  mkfs.ext4 -O bigalloc -C 4096|8192|16384 device_name
   ```

   > **Note**
   >
   > 对于 MySQL 和 MariaDB，必须使用 `-C 16384` 来匹配数据库页面大小。将分配粒度设置为页面大小倍数以外的值可能会导致分配与存储设备撕裂写防护边界不匹配。

   例如：

   ```
   $  mkfs.ext4 -O bigalloc -C 16384 /dev/nvme1n1
   ```

2. 将 InnoDB 配置为使用 `O_DIRECT` 刷新方法并关闭 InnoDB 双写模式。使用首选文本编辑器打开 `/etc/my.cnf`，更新 `innodb_flush_method` 和 `innodb_doublewrite` 参数，如下所示：

   ```
   innodb_flush_method=O_DIRECT
   innodb_doublewrite=0
   ```

> **Important**
>
> 如果您使用的是逻辑卷管理器（LVM）或其他存储虚拟化层，请确保卷的起始偏移量以 16 KiB 倍数对齐。这是相对于底层 NVMe 存储而言，用以考虑存储虚拟化层使用的元数据标头和超级块。如果您向 LVM 物理卷添加偏移量，则可能引起文件系统分配与 NVMe 设备的偏移量有偏差，从而导致撕裂写防护失效。有关更多信息，请参阅 Linux manual page（Linux 手册页面）中的 `--dataalignmentoffset`。

基于 Windows VSS 的应用程序一致性 Amazon EBS 快照

您可以使用 Amazon Systems Manager Run Command，为附加到 Amazon EC2 Windows 实例的所有 Amazon EBS 卷拍摄应用程序一致性快照。快照过程使用 Windows 卷影复制服务（VSS）为 VSS

感知应用程序拍摄 EBS 卷级备份。这些快照包括这些应用程序和磁盘之间的挂起事务中的数据。在需要备份所有附加的卷时，您无需关闭或断开连接实例。

使用基于 VSS 的 EBS 快照不会产生额外费用。您只需为备份过程创建的 EBS 快照付费。有关更多信息，请参阅我的 Amazon EBS 快照是如何计费的？

> **Note**
>
> 只有 Windows 实例支持基于应用程序一致的 Windows VSS 快照。

内容

- 什么是 VSS？
- 基于 VSS 的 Amazon EBS 快照解决方案工作原理
- 创建基于 Windows VSS 的 EBS 快照的先决条件
- 为您的 EC2 Windows 实例创建基于 VSS 的 EBS 快照
- 对基于 Windows VSS 的 EBS 快照进行问题排查
- 使用 Amazon VSS 解决方案恢复实例数据
- Amazon VSS 解决方案版本历史记录

什么是 VSS？

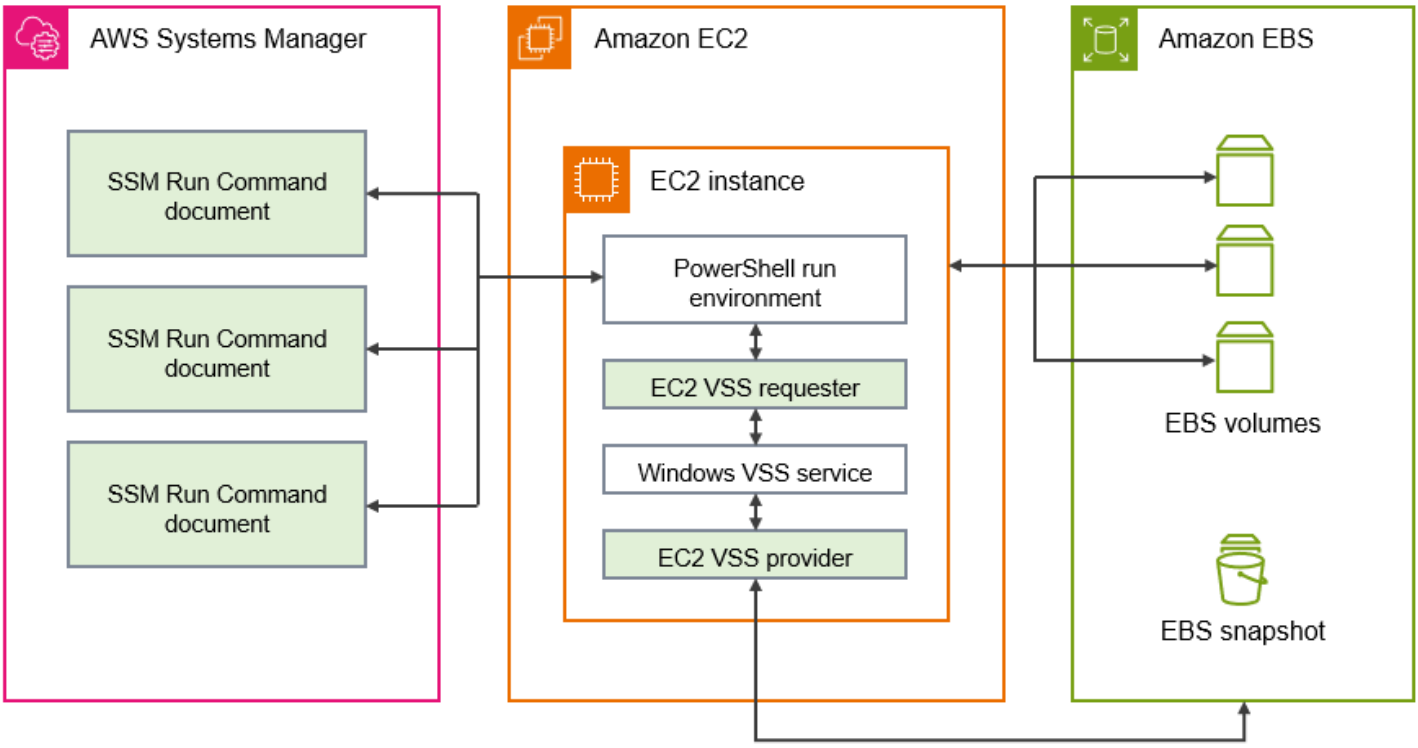
卷影复制服务（VSS）是 Microsoft Windows 中包含的一项备份和恢复技术。它可以在计算机文件或卷处于使用状态时创建备份副本或快照。有关更多信息，请参阅卷影复制服务。

要创建应用程序一致性快照，需要使用以下软件组件。

- VSS 服务 – Windows 操作系统的一部分
- VSS 请求程序 – 请求创建卷影副本的软件
- VSS 写入器 – 通常作为应用程序（例如 SQL Server）的一部分提供，用于确保备份一致的数据集
- VSS 提供程序 – 创建底层卷影副本的组件

基于 Windows VSS 的 Amazon EBS 快照解决方案包括多个 Systems Manager（SSM）Run Command 文档，这些文档有助于创建备份，还有一个名为 `AwsVssComponents` 的 Systems

Manager Distributor 包，其中包含 EC2 VSS 请求程序和 EC2 VSS 提供程序。必须将 `AwsVssComponents` 包安装在 EC2 实例上，才能拍摄 EBS 卷的应用程序一致性快照。下图说明了这些软件组件之间的关系。



基于 VSS 的 Amazon EBS 快照解决方案工作原理

拍摄基于 VSS 的应用程序一致性 EBS 快照脚本的过程包括以下步骤。

1. 完成 创建基于 Windows VSS 的 EBS 快照的先决条件。
2. 输入 `AWSEC2-VssInstallAndSnapshot` SSM 文档的参数，并使用 Run Command 运行该文档。有关更多信息，请参阅 运行 AWSEC2-VssInstallAndSnapshot 命令文档（推荐）。
3. 您的实例上的 Windows VSS 服务会协调正在运行的应用程序的所有正在进行的 I/O 操作。
4. 系统会刷新所有 I/O 缓冲区并临时暂停所有 I/O 操作。暂停最多持续 10 秒钟。
5. 在暂停期间，系统会为附加到实例的所有卷创建快照。
6. 暂停解除，I/O 恢复操作。
7. 系统将所有新建快照添加到 EBS 快照列表。系统会为此过程成功创建的所有基于 VSS 的 EBS 快照添加 AppConsistent:true 标签。
8. 如果需要从快照中还原，您可以使用从快照中创建卷的标准 EBS 过程，也可以使用示例脚本将所有卷还原到实例，如 使用 Amazon VSS 解决方案恢复实例数据 中所述。

创建基于 Windows VSS 的 EBS 快照的先决条件

可以使用 Systems Manager Run Command、Amazon Backup 或 Amazon Data Lifecycle Manager 创建基于 VSS 的 EBS 快照。以下是使用两种解决方案的先决条件。

[系统要求]

确保您的 EC2 Windows 实例满足创建基于 VSS 的快照的所有系统要求，包括支持的 Windows 操作系统版本、.NET 框架、PowerShell、Amazon Tools for Windows PowerShell 和 Amazon Systems Manager 代理版本。

[IAM 权限]

附加到您的 Amazon EC2 Windows 实例的 IAM 角色必须有使用 VSS 创建应用程序一致性快照的权限。要授予必要的权限，您可以将 `AWSEC2VssSnapshotPolicy` 托管策略附加到实例配置文件。

[VSS 组件]

要在 Windows 操作系统上创建应用程序一致性快照，必须在实例上安装 `AwsVssComponents` 软件包。此软件包包含一个实例 EC2 VSS 代理，它充当 VSS 请求程序以及 EBS 卷的 EC2 VSS 提供程序。

系统要求

安装 Systems Manager 代理

VSS 由 Systems Manager Agent 使用 PowerShell 编排。请确保您已在您的 Amazon EC2 实例上安装了 SSM Agent 版本 `3.0.502.0` 或更高版本。如果使用的是较旧版本的 SSM Agent，请使用 Run Command 进行更新。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的为 Amazon EC2 实例设置 Systems Manager 和在适用于 Windows Server 的 Amazon EC2 实例上使用 SSM Agent。

Amazon EC2 Windows 实例要求

运行 Windows Server 2016 及更高版本的实例可支持基于 VSS 的 EBS 快照。

.NET Framework 版本

`AwsVssComponents` 包需要 .NET Framework 版本 `4.6` 及更高版本。Windows Server 2016 之前的 Windows 操作系统版本默认使用早期版本的 .NET Framework。如果实例使用早期版本的 .NET Framework，则必须使用 Windows Update 安装 `4.6` 或更高版本。

Amazon Tools for Windows PowerShell 版本

确保实例运行的是版本 3.3.48.0 或更高版本的 Amazon Tools for Windows PowerShell。要检查版本，请在 PowerShell 终端中的实例上运行以下命令。

```
C:\> Get-AWSPowerShellVersion
```

如果您需要在实例上更新 Amazon Tools for Windows PowerShell，请参阅《Amazon Tools for PowerShell 用户指南》中的安装 Amazon Tools for Windows PowerShell。

Windows Powershell 版本

确保您的实例运行的是 Windows PowerShell 主要版本 3、4 或 5。要检查版本，请在 PowerShell 终端中的实例上运行以下命令。

```
C:\> $PSVersionTable.PSVersion
```

PowerShell 语言模式

确保实例将 PowerShell 语言模式设置为 FullLanguage。有关更多信息，请参阅 Microsoft 文档中的 about_Language_Modes。

使用 IAM 托管策略为基于 VSS 的快照授予权限

AWSEC2VssSnapshotPolicy 托管策略允许 Systems Manager 在 Windows 实例上执行以下操作：

- 创建和标记 EBS 快照
- 创建和标记亚马逊机器映像（AMI）
- 将元数据（如设备 ID）附加到 VSS 创建的默认快照标签。

本主题介绍 VSS 托管策略的权限详细信息，以及如何将其附加到您的 EC2 实例配置文件 IAM 角色。

内容

- AWSEC2VssSnapshotPolicy 托管策略详细信息
- 将 VSS 快照托管策略附加到实例配置文件角色

AWSEC2VssSnapshotPolicy 托管策略详细信息

Amazon 托管策略是 Amazon 为 Amazon 客户提供的独立策略。Amazon 托管策略旨在为常见使用案例授予权限。您无法更改 Amazon 托管策略中定义的权限。但是，您可以复制该策略并将其用作特定于您的使用案例的客户管理型策略的基准。

有关 Amazon 托管式策略的更多信息，请参阅《IAM 用户指南》中的 Amazon 托管式策略。

要使用 AWSEC2VssSnapshotPolicy 托管策略，您可以将其附加到 IAM 角色，该角色已附加到您的 EC2 Windows 实例。此策略允许 EC2 VSS 解决方案为亚马逊机器映像（AMI）和 EBS 快照创建和添加标签。要附加策略，请参阅 将 VSS 快照托管策略附加到实例配置文件角色。

AWSEC2VssSnapshotPolicy 授予的权限

AWSEC2VssSnapshotPolicy 策略包括以下 Amazon EC2 权限，允许 Amazon EC2 代表您创建和管理 VSS 快照。您可以将此托管策略附加到用于 EC2 Windows 实例的 IAM 实例配置文件角色。

- ec2:CreateTags – 向 EBS 快照和 AMI 添加标签，以帮助识别和分类资源。
- ec2:DescribeInstanceAttribute – 检索附加到目标实例的 EBS 卷和相应的块设备映射。
- ec2:CreateSnapshots – 创建 EBS 卷的快照。
- ec2:CreateImage – 从正在运行的 EC2 实例创建 AMI。
- ec2:DescribeImages – 检索 EC2 AMI 和快照的信息。
- ec2:DescribeSnapshots – 确定快照的创建时间和状态以验证应用程序一致性。

> **Note**
>
> 要查看此策略的权限详细信息，请参阅《Amazon Managed Policy Reference》中的 AWSEC2VssSnapshotPolicy。

简化特定使用案例的权限：高级

`AWSEC2VssSnapshotPolicy` 托管策略包括创建基于 VSS 的快照的所有方式的权限。您可创建仅包含所需权限的自定义策略。

使用案例：创建 AMI，使用案例：使用 Amazon Backup 服务

如果您只使用 `CreateAmi` 选项，或者仅通过 Amazon Backup 服务创建基于 VSS 的快照，则可以按如下方式简化策略语句。

- 忽略由以下语句 ID（SID）标识的策略语句：
  - CreateSnapshotsWithTag
  - CreateSnapshotsAccessInstance
  - CreateSnapshotsAccessVolume
- 按如下方式调整 CreateTagsOnResourceCreation 语句：
  - 从资源中删除 arn:aws:ec2:*:*:snapshot/*。
  - 从 ec2:CreateAction 条件中删除 CreateSnapshots。
- 调整 CreateTagsAfterResourceCreation 语句以从资源中删除 arn:aws:ec2:*:*:snapshot/*。
- 调整 DescribeImagesAndSnapshots 语句以从语句操作中删除 ec2:DescribeSnapshots。

使用案例：仅限快照

如果您不使用 CreateAmi 选项，则可以按如下方式简化策略语句。

- 忽略由以下语句 ID（SID）标识的策略语句：
  - CreateImageAccessInstance
  - CreateImageWithTag
- 按如下方式调整 CreateTagsOnResourceCreation 语句：
  - 从资源中删除 arn:aws:ec2:*:*:image/*。
  - 从 ec2:CreateAction 条件中删除 CreateImage。
- 调整 CreateTagsAfterResourceCreation 语句以从资源中删除 arn:aws:ec2:*:*:image/*。
- 调整 DescribeImagesAndSnapshots 语句以从语句操作中删除 ec2:DescribeImages。

> **Note**
>
> 为确保您的自定义策略按预期执行，建议您定期查看托管策略并将其更新纳入其中。

将 VSS 快照托管策略附加到实例配置文件角色

要为您的 EC2 Windows 实例授予基于 VSS 的快照的权限，您可以按如下方式将 AWSEC2VssSnapshotPolicy 托管策略附加到您的实例配置文件角色。确保您的实例满足所有 系统要求 很重要。

> **Note**
>
> 要使用托管策略，您的实例必须安装 AwsVssComponents 软件包版本 2.3.1 或更高版本。有关版本历史记录，请参阅 AwsVssComponents 包版本。

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在导航窗格中，选择角色以查看您有权访问的 IAM 角色列表。
3. 为附加到您的实例的角色选择角色名称链接。这将打开角色详细信息页面。
4. 要附加托管策略，请选择位于列表面板右上角的添加权限。然后，从下拉列表中选择附加策略。
5. 要简化搜索结果，请在搜索栏中输入策略名称（AWSEC2VssSnapshotPolicy）。
6. 选中要附加的策略名称旁边的复选框，然后选择添加权限。

管理基于 Windows VSS 的 EBS 快照的 VSS 组件包

在创建基于 VSS 的 EBS 快照之前，请确保在 Windows 实例上已安装最新版本的 VSS 组件包。可使用多种方法将 AwsVssComponents 包安装到现有实例上，如下所示：

- （建议）运行 AWSEC2-VssInstallAndSnapshot 命令文档（推荐）。在每次运行时都会根据需要自动安装或更新。
- 在 EC2 Windows 实例上手动安装 VSS 组件.
- 更新 EC2 Windows 实例上的 VSS 组件包.

您还可以使用 EC2 Image Builder 创建一个 AMI，该 AMI 将使用 aws-vss-components-windows 托管组件为映像安装 AwsVssComponents 软件包。托管组件使用 Amazon Systems Manager Distributor 安装软件包。Image Builder 创建映像后，从关联 AMI 启动的每个实例都将安装 VSS 软件包。有关如何创建已安装 VSS 软件包的 AMI 的更多信息，请参阅《EC2 Image Builder 用户指南》中的适用于 Windows 的 Distributor 软件包托管组件。

内容

- 在 EC2 Windows 实例上手动安装 VSS 组件
- 更新 EC2 Windows 实例上的 VSS 组件包

在 EC2 Windows 实例上手动安装 VSS 组件

EC2 Windows 实例必须安装 VSS 组件，然后您才能使用 Systems Manager 创建应用程序一致性快照。如果没有在每次创建应用程序一致性快照时，运行 `AWSEC2-VssInstallAndSnapshot` 命令文档来自动安装或更新软件包，则必须手动安装该软件包。

如果您计划使用以下方法之一从 EC2 实例创建应用程序一致性快照，则也必须进行手动安装。

- 使用 Amazon Backup 创建 VSS 快照
- 使用 Amazon Data Lifecycle Manager 创建 VSS 快照

如果需要执行手动安装，我们建议使用最新的 Amazon VSS 组件包，以提高 EC2 Windows 实例上应用程序一致性快照的可靠性和性能。

> **Note**
>
> 要在每次创建应用程序一致性快照时自动安装或更新 `AwsVssComponents` 软件包，建议使用 Systems Manager 运行 `AWSEC2-VssInstallAndSnapshot` 文档。有关更多信息，请参阅运行 AWSEC2-VssInstallAndSnapshot 命令文档（推荐）。

要在 Amazon EC2 Windows 实例上安装 VSS 组件，请按照适用于您的首选环境的步骤操作。

Console

要使用 SSM Distributor 安装 VSS 组件

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 在导航窗格中，选择Run Command。
3. 选择运行命令。
4. 对于 Command document（命令文档），选择 AWS-ConfigureAWSPackage 旁的按钮。
5. 对于命令参数，执行以下操作：

   a. 确认操作设置为安装。

b. 对于名称，请输入 `AwsVssComponents`。

c. 对于版本，输入一个版本或将该字段保留空白，以便 Systems Manager 安装最新的版本。

6. 对于目标，请指定标签或手动选择实例以确定要在其中运行该操作的实例。

> **Note**
>
> 如果您选择手动选择实例，并且您要查看的实例未包含在列表中，请参阅 Amazon Systems Manager 用户指南中的我的实例在哪里？以获得故障排除提示。

7. 对于其他参数：

- （可选）对于注释，请键入有关该命令的信息。
- 对于超时（秒），请指定在整个命令执行失败之前系统等待的秒数。

8. （可选）对于速率控制：

- 对于并发，请指定要同时运行该命令的实例数或百分比。

> **Note**
>
> 如果通过选择 Amazon EC2 标签选择了目标，但不确定有多少个实例使用所选标签，则可以通过指定百分比来限制可同时运行此文档的实例的数量。

- 对于错误阈值，请指定在一定数量或百分比的实例上失败后何时在其他实例上停止运行该命令。例如，如果您指定三个错误，Systems Manager 将在收到第四个错误时停止发送该命令。仍在处理命令的实例也可能发送错误。

9. （可选）对于输出选项部分，如果您要将命令输出保存到文件，请选定启用 S3 存储桶写入旁边的选项框。指定存储桶和（可选）前缀（文件夹）名称。

> **Note**
>
> 授予将数据写入 S3 存储桶的能力的 S3 权限是分配给实例的实例配置文件的权限，而不是执行此任务的 用户的权限。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的配置 EC2 实例权限。

10. （可选）为 SNS 通知指定选项。

有关为 Run Command 配置 Amazon SNS 通知的信息，请参阅为 Amazon Systems Manager 配置 Amazon SNS 通知。

11. 选择运行。

Amazon CLI

可以使用以下过程下载 AwsVssComponents pa 程序包，并从 Amazon CLI 中使用 Run Command 在实例上安装该程序包。该程序包安装两个组件：VSS 请求程序和 VSS 提供程序。系统将这些组件复制到实例上的某个目录，然后将提供程序 DLL 注册为 VSS 提供程序。

安装 VSS 包

运行以下命令下载并安装 Systems Manager 所需的 VSS 组件。

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"action":["Install"],"name":["AwsVssComponents"]}'
```

PowerShell

使用以下过程用 Run Command 从 Tools for Windows PowerShell 下载 AwsVssComponents 程序包并在实例上进行安装。该程序包安装两个组件：VSS 请求程序和 VSS 提供程序。系统将这些组件复制到实例上的某个目录，然后将提供程序 DLL 注册为 VSS 提供程序。

安装 VSS 包

运行以下命令下载并安装 Systems Manager 所需的 VSS 组件。

```
Send-SSMCommand `
    -DocumentName "AWS-ConfigureAWSPackage" `
    -InstanceId "i-1234567890abcdef0" `
    -Parameter @{'action'='Install';'name'='AwsVssComponents'}
```

验证 Amazon VSS 组件上的签名

按照以下过程验证 AwsVssComponents 程序包上的签名。

1. 连接到您的 Windows 实例。有关更多信息，请参阅 使用 RDP 连接到 Windows 实例。

2. 导航到 C:\Program Files\Amazon\AwsVssComponents。
3. 打开 ec2-vss-agent.exe 的上下文菜单（单击右键），然后选择属性。
4. 导航到数字签名选项卡，验证签名者的名称是否为 Amazon Web Services Inc.。
5. 按照前述步骤验证 Ec2VssInstaller 和 Ec2VssProvider.dll 上的签名。

更新 EC2 Windows 实例上的 VSS 组件包

建议始终将 VSS 组件更新到最新的推荐版本。当发布了一个新版本的 AwsVssComponents 软件包时，可以通过多种不同的方式更新组件。

更新方法

- 当发布了新版本的 Amazon VSS 组件时，可以重复 在 EC2 Windows 实例上手动安装 VSS 组件 中所述的步骤。
- 可以将一个 Systems Manager State Manager 关联配置为在 AwsVssComponents 软件包可用时，自动下载并安装全新或更新后的 VSS 组件。
- 如果使用 Systems Manager 运行 AWSEC2-VssInstallAndSnapshot 文档，每次创建应用程序一致性快照时，都可以自动安装或更新 AwsVssComponents 软件包。

> **Note**
>
> 建议使用 Systems Manager 运行 AWSEC2-VssInstallAndSnapshot 命令文档，它会在创建应用程序一致性快照之前自动安装或更新 AwsVssComponents 软件包。有关更多信息，请参阅 运行 AWSEC2-VssInstallAndSnapshot 命令文档（推荐）。

要创建 Systems Manager State Manager 关联，请按照适用于您的首选环境的步骤操作。

Console

创建 Systems Manager 状态管理器关联时，有两个选项可用于更新 AwsVssComponents 程序包，如下所示：

卸载并重新安装

此方法会下载并安装程序包，无需任何其他先决条件。

就地更新

这将对程序包执行就地更新，并具有以下先决条件：

- 实例上安装的 SSM 代理版本必须是版本 3.3.808.0 或更高版本。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的在适用于 Windows Server 的 Amazon EC2 实例上使用 SSM Agent。
- 如果指定，则 AwsVssComponents 程序包版本必须为版本 2.5.0 或更高版本。早期版本不支持就地更新。

> **Note**
>
> 如果您的实例不满足这些先决条件，则就地更新将失败。改用卸载并重新安装选项。

创建 State Manager 关联

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 在导航窗格中，选择状态管理器。

   或者，如果一开始打开了 Systems Manager 主页，请打开导航窗格，然后选择 State Manager。
3. 选择 Create association（创建关联）。
4. 在 Name（名称）字段中，输入一个描述性名称。
5. 在文档列表中，选择 AWS-ConfigureAWSPackage。
6. 在参数部分，选择在操作列表中安装。
7. 对于安装类型，选择卸载并重新安装或就地更新。
8. 在名称字段中，输入 AwsVssComponents。您可以保持版本和其他参数字段为空。
9. 在 Targets 部分中，选择选项。

   > **Note**
   >
   > 如果您选择使用标签将实例设为目标，并指定映射到 Linux 实例的标签，则关联在 Windows 实例上将成功，但在 Linux 实例上将失败。关联的总体状态将显示 Failed。
10. 在指定计划部分，选择一个选项。

11. 在 高级选项 部分中，对于合规性严重级别，选择关联的严重级别。有关更多信息，请参阅关于关联合规性。对于 Change Calendar，请选择预配置的更改日历。有关更多信息，请参阅 Amazon Systems Manager Change Calendar。
12. 对于速率控制，请执行以下操作：
    - 对于 Concurrency（并发），请指定要同时运行该命令的托管式节点的数量或百分比。
    - 对于 Error threshold（错误阈值），请指定当命令在一定数量或百分比的节点上失败后，何时在其他托管式节点上停止运行该命令。
13. （可选）对于输出选项，要将命令输出保存到文件，请选中 启用将输出写入 S3 方框。在输入框中输入存储桶和前缀（文件夹）名称。
14. 选择 Create Association，然后选择 Close。系统将尝试在实例上创建关联并立即应用状态。

> **Note**
>
> 如果适用于 Windows Server 的 EC2 实例显示失败状态，请验证 SSM Agent 是否在该实例上运行，并验证该实例是否配置有适用于 Systems Manager 的 Amazon Identity and Access Management（IAM）角色。有关更多信息，请参阅设置 Amazon Systems Manager。

Amazon CLI

使用 create-association 命令以按计划更新 Distributor 包，而无需将关联的应用程序脱机。仅替换软件包中的新文件或更新的文件。

创建 State Manager 关联

运行以下命令以创建关联。`--name`的值（文档名称）始终为 `AWS-ConfigureAWSPackage`。以下命令使用键 `InstanceIds` 指定目标实例。

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters action=Install,installationType="Uninstall and
 reinstall",name=AwsVssComponents \
    --targets Key=InstanceIds,Values=i-1234567890abcdef0,i-000011112222abcde
```

PowerShell

创建 State Manager 关联

使用 New-SSMAssociation cmdlet。

```
New-SSMAssociation `
    -Name "AWS-ConfigureAWSPackage" `
    -Parameter  @{
        "action" = "Install"
        "installationType" = "Uninstall and reinstall"
        "name" = "AwsVssComponents"
    } `
    -Target @{
        "Key" = "InstanceIds"
        "Values" = @("i-1234567890abcdef0", "i-000011112222abcde")
    }
```

为您的 EC2 Windows 实例创建基于 VSS 的 EBS 快照

满足所有创建基于 Windows VSS 的 EBS 快照的先决条件后，可以使用以下任一方法从 EC2 实例创建基于 VSS 的快照。

Amazon Systems Manager 命令文档

使用 Systems Manager 命令文档以创建基于 VSS 的快照。

要自动执行备份，可以创建一个使用 AWSEC2-VssInstallAndSnapshot 命令文档的 Amazon Systems Manager 维护时段任务。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的使用维护时段（控制台）。

Amazon Backup

使用 Amazon Backup 时，您可以在控制台或 CLI 中通过启用 VSS 来创建 VSS 备份。有关更多信息，请参阅《Amazon Backup Developer Guide》中的 Creating Windows VSS Backup。

> **Note**
>
> Amazon Backup 不会在实例上自动安装 AwsVssComponents 软件包。您必须在实例上执行手动安装。有关更多信息，请参阅 在 EC2 Windows 实例上手动安装 VSS 组件。

Amazon Data Lifecycle Manager

您可以通过在快照生命周期策略中启用前置和后置脚本，使用 Amazon Data Lifecycle Manager 创建 VSS 快照。有关更多信息，请参阅《Amazon EBS User Guide》中的 Automating application-consistent snapshots。

> **Note**
>
> Amazon Data Lifecycle Manager 不会在实例上自动安装 `AwsVssComponents` 软件包。您必须在实例上执行手动安装。有关更多信息，请参阅 在 EC2 Windows 实例上手动安装 VSS 组件。

使用 Systems Manager 命令文档创建基于 VSS 的快照

可以使用 Amazon Systems Manager 命令文档创建基于 VSS 的快照。以下内容介绍可用的命令文档，以及这些文档用于创建快照的运行时系统参数。

在使用任何 Systems Manager 命令文档之前，请确保已满足所有 创建基于 Windows VSS 的 EBS 快照的先决条件。

主题

- Systems Manager VSS 快照文档的参数
- 运行 Systems Manager VSS 快照命令文档

Systems Manager VSS 快照文档的参数

用于创建 VSS 快照的 Systems Manager 文档均使用以下参数，除非另有说明：

AmiName（字符串，可选）

如果 CreateAmi 设置为 `True`，请指定备份创建的 AMI 名称。

description（字符串，可选）

为此过程创建的快照或映像指定一个描述。

CollectDiagnosticLogs（字符串，可选）

要在快照和 AMI 创建步骤中收集更多信息，请将此参数设置为“`True`”。此参数的默认值为“`False`”。整合后的诊断日志将以 `.zip` 存档格式保存在实例的以下位置：

```
C:\ProgramData\Amazon\AwsVss\Logs\timestamp.zip
```

CopyOnly（字符串，可选）

如果除了 Amazon VSS 之外还使用本地 SQL Server 备份，执行“仅复制”备份可防止 Amazon VSS 中断本地差异备份链。要执行一个“仅复制”备份操作，请将此参数设置为 `True`。

此参数的默认值为 `False`，这样会导致 Amazon VSS 执行完整备份操作。

CreateAmi（字符串，可选）

要创建基于 VSS 的亚马逊机器映像（AMI）以备份您的实例，请将此参数设置为 `True`。此参数的默认值为 `False`，它将使用 EBS 快照备份您的实例。

有关创建从实例创建 AMI 的更多信息，请参阅 创建 Amazon EBS-backed AMI。

executionTimeout（字符串，可选）

指定在实例上执行快照创建过程或从实例创建 AMI 的最长时间（以秒为单位）。增大此超时将允许此命令等待更长时间，以使 VSS 开始冻结并完成对它创建的资源执行的标记操作。此超时只适用于快照或 AMI 创建步骤。此超时不包括用来安装或更新 `AwsVssComponents` 软件包的初始步骤所花费的时间。

ExcludeBootVolume（字符串，可选）

如果是创建快照，此设置将从备份过程中排除启动卷。要从快照中排除启动卷，请将 ExcludeBootVolume 设置为 **`True`**、将 CreateAmi 设置为 `False`。

如果是为备份创建 AMI，应将此参数设置为 `False`。此参数的默认值为 `False`。

NoWriters（字符串，可选）

要从快照进程中排除应用程序 VSS 写入器，请将此参数设置为 `True`。排除应用程序 VSS 写入器可以帮助您解决与第三方 VSS 备份组件的冲突。此参数的默认值为 `False`。

如果 `SaveVssMetadata` 为 `True`，则此参数必须设置为 `False`。

SaveVssMetadata（字符串，可选）

要在每次快照拍摄期间保存 VSS 元数据文件，请将此参数设置为 `True`。默认值为 `False`。这些文件有助于深入了解备份操作中包含的具体组件或写入器，以及每个快照的关联文件和卷。使用 VSS 还原解决方案还原 SQL 数据库时会使用元数据文件。有关如何从 VSS 快照还原 SQL 数据库的更多信息，请参阅使用自动化运行手册从 Amazon VSS 解决方案快照还原数据库。

元数据文件的名称中包含关联快照集的 ID。可以在实例上的以下位置找到这些信息：

```
C:\ProgramData\Amazon\AwsVss\VssMetadata\
```

> **Warning**
>
> - 保存 VSS 元数据文件需要 AwsVssComponents 软件包版本 2.4.0 或更高版本。如果实例安装的是早期版本，则将 SaveVssMetadata 设置为 True 会导致快照创建失败。
> - NoWriters 和 SaveVssMetadata 参数是互斥的。如果两者都设置为 True，则快照创建将会失败。

tags（字符串，可选）

建议您标记快照和映像，以帮助您定位和管理资源，例如，从快照列表中恢复卷。系统添加带有空值的 Name 键，您可以在其中指定要应用于输出快照或映像的名称。

如果您要指定更多标签，请使用分号分隔标签。例如 Key=Environment,Value=Test;Key=User,Value=TestUser1。

> **Note**
>
> 标签键和值只能包含字母数字字符和以下特殊字符：() ./\-"'@_+:={}。

默认情况下，系统会为基于 VSS 的快照和映像添加以下保留标签。

- 设备：对于基于 VSS 的快照，这是快照捕获的 EBS 卷的设备名称。
- AppConsistent：此标签表示已成功创建基于 VSS 的快照或 AMI。
- AwsVssConfig – 此标签用于识别在启用 VSS 的情况下创建的快照和 AMI。此标签包含 AwsVssComponents 版本等元信息以及快照集 ID。

> **Warning**
>
> 在参数列表中指定这些保留标签中的任何一个都将导致错误。

VssVersion（字符串，可选）

仅对于 AWSEC2-VssInstallAndSnapshot 文档，可以指定 VssVersion 参数，以便在实例上安装特定版本的 AwsVssComponents 软件包。如果将此参数留空，将安装建议的默认版本。

如果已经安装了 AwsVssComponents 软件包的指定版本，此脚本将跳过安装步骤并转到备份步骤。有关 AwsVssComponents 软件包版本和操作支持的列表，请参阅 Amazon VSS 解决方案版本历史记录。

运行 Systems Manager VSS 快照命令文档

可以使用 Amazon Systems Manager 命令文档创建基于 VSS 的 EBS 快照，如下所示。

运行 AWSEC2-VssInstallAndSnapshot 命令文档（推荐）

当使用 Amazon Systems Manager 运行 AWSEC2-VssInstallAndSnapshot 文档时，此脚本将执行以下步骤。

1. 此脚本首先在实例上安装或更新 AwsVssComponents 软件包，具体情况取决于是否已安装此软件包。
2. 此脚本将在完成第一步之后创建应用程序一致性快照。

要运行 AWSEC2-VssInstallAndSnapshot 文档，请按照适用于您的首选环境的步骤操作。

Console

从控制台创建基于 VSS 的 EBS 快照

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 从导航窗格中选择运行命令。这将显示您的账户中当前正在运行的命令的列表（如果适用）。
3. 选择 Run command（运行命令）。这将打开您有权访问的命令文档的列表。
4. 从命令文档列表中选择 AWSEC2-VssInstallAndSnapshot。为了简化结果，您可以输入完整或部分文档名称。也可以按所有者、平台类型或标签进行筛选。

   当选择命令文档时，将在列表下方填充详细信息。
5. 从文档版本列表中选择 Default version at runtime。
6. 配置命令参数，以定义 AWSEC2-VssInstallAndSnapshot 如何安装 AwsVssComponents 软件包以及如何使用 VSS 快照或 AMI 进行备份。有关参数的详细信息，请参阅 Systems Manager VSS 快照文档的参数。
7. 对于目标选择，请指定标签或者手动选择实例，以确定要在其上执行此操作的实例。

> **Note**
>
> 如果手动选择实例，而且要查看的实例未包含在列表中，请参阅我的实例在哪里？以获得故障排除提示。

8. 要获得用来定义 Systems Manager Run Command 行为（例如速率控制）的其他参数，请输入从控制台运行命令中所述的值。
9. 选择运行。

   如果成功，则该命令使用新快照填充 EBS 快照列表。您可以通过搜索指定的标签或搜索 AppConsistent 在 EBS 快照列表中查找这些快照。如果命令执行失败，请查看 Systems Manager 命令输出，了解执行失败的详细原因。如果命令成功完成，但特定卷备份失败，您可以在 EBS 卷列表中排查失败的原因。

Amazon CLI

可以在 Amazon CLI 中运行以下命令，以创建基于 VSS 的 EBS 快照，并获取快照创建的状态。

创建基于 VSS 的 EBS 快照

运行以下命令以创建基于 VSS 的 EBS 快照。要创建快照，必须使用 `--instance-ids` 参数标识实例。有关可以使用的其他参数的更多信息，请参阅 Systems Manager VSS 快照文档的参数。

```
aws ssm send-command \
 --document-name "AWSEC2-VssInstallAndSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"],"VssVersion":[""]}'
```

如果成功，则该命令文档使用新快照填充 EBS 快照列表。您可以通过搜索指定的标签或搜索 AppConsistent 在 EBS 快照列表中查找这些快照。如果命令执行失败，请查看命令输出，了解执行失败的详细原因。

获取命令状态

要获取快照的当前状态，请使用从 send-command 返回的命令 ID 运行如下命令。

```
aws ssm get-command-invocation
```

```
--instance-ids "i-01234567890abcdef" \
--command-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
--plugin-name "CreateVssSnapshot"
```

PowerShell

使用 Amazon Tools for Windows PowerShell 运行以下命令，以创建基于 VSS 的 EBS 快照，并获取输出创建过程的当前运行时状态。指定前面的列表中介绍的参数，以修改快照进程的行为。

使用 Tools for Windows PowerShell 创建基于 VSS 的 EBS 快照

运行如下命令，以创建基于 VSS 的 EBS 快照或 AMI。

```
Send-SSMCommand -DocumentName "AWSEC2-VssInstallAndSnapshot" -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value';'VssVersion'=''}
```

获取命令状态

要获取快照的当前状态，请使用从 Send-SSMCommand 返回的命令 ID 运行如下命令。

```
Get-SSMCommandInvocationDetail -InstanceId "i-01234567890abcdef" -CommandId
 "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" -PluginName "CreateVssSnapshot"
```

如果成功，则该命令使用新快照填充 EBS 快照列表。您可以通过搜索指定的标签或搜索 `AppConsistent` 在 EBS 快照列表中查找这些快照。如果命令执行失败，请查看命令输出，了解执行失败的详细原因。

运行 AWSEC2-CreateVssSnapshot 命令文档

要运行 `AWSEC2-CreateVssSnapshot` 文档，请按照适用于您的首选环境的步骤操作。

Console

从控制台创建基于 VSS 的 EBS 快照

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 从导航窗格中选择运行命令。这将显示您的账户中当前正在运行的命令的列表（如果适用）。

3. 选择 Run command（运行命令）。这将打开您有权访问的命令文档的列表。
4. 从命令文档列表中选择 AWSEC2-CreateVssSnapshot。为了简化结果，您可以输入完整或部分文档名称。也可以按所有者、平台类型或标签进行筛选。

   当选择命令文档时，将在列表下方填充详细信息。
5. 从文档版本列表中选择 Default version at runtime。
6. 配置命令参数，以定义 AWSEC2-CreateVssSnapshot 使用 VSS 快照或 AMI 进行备份的方式。有关参数的详细信息，请参阅 Systems Manager VSS 快照文档的参数。
7. 对于目标选择，请指定标签或者手动选择实例，以确定要在其上执行此操作的实例。

   > **Note**
   >
   > 如果手动选择实例，而且要查看的实例未包含在列表中，请参阅我的实例在哪里？以获得故障排除提示。

8. 要获得用来定义 Systems Manager Run Command 行为（例如速率控制）的其他参数，请输入从控制台运行命令中所述的值。
9. 选择运行。

   如果成功，则该命令使用新快照填充 EBS 快照列表。您可以通过搜索指定的标签或搜索 AppConsistent 在 EBS 快照列表中查找这些快照。如果命令执行失败，请查看 Systems Manager 命令输出，了解执行失败的详细原因。如果命令成功完成，但特定卷备份失败，您可以在 EBS 卷列表中排查失败的原因。

Amazon CLI

在 Amazon CLI 中运行以下命令来创建基于 VSS 的 EBS 快照。

创建基于 VSS 的 EBS 快照

运行以下命令以创建基于 VSS 的 EBS 快照。要创建快照，必须使用 --instance-ids 参数标识实例。有关可以使用的其他参数的更多信息，请参阅 Systems Manager VSS 快照文档的参数。

```
aws ssm send-command \
 --document-name "AWSEC2-CreateVssSnapshot" \
 --instance-ids "i-01234567890abcdef" \
 --parameters '{"ExcludeBootVolume":["False"],"description":["Description"],"tags":
["Key=key_name,Value=tag_value"]}'
```

如果成功，则该命令文档使用新快照填充 EBS 快照列表。您可以通过搜索指定的标签或搜索 `AppConsistent` 在 EBS 快照列表中查找这些快照。如果命令执行失败，请查看命令输出，了解执行失败的详细原因。

PowerShell

使用 Amazon Tools for Windows PowerShell 运行以下命令以创建基于 VSS 的 EBS 快照。

使用 Tools for Windows PowerShell 创建基于 VSS 的 EBS 快照

运行以下命令以创建基于 VSS 的 EBS 快照。要创建快照，必须使用 `InstanceId` 参数标识实例。可以指定多个实例为其创建快照。有关可以使用的其他参数的更多信息，请参阅 Systems Manager VSS 快照文档的参数。

```
Send-SSMCommand -DocumentName AWSEC2-CreateVssSnapshot -InstanceId
 "i-01234567890abcdef" -Parameter
 @{'ExcludeBootVolume'='False';'description'='a_description'
 ;'tags'='Key=key_name,Value=tag_value'}
```

如果成功，则该命令使用新快照填充 EBS 快照列表。您可以通过搜索指定的标签或搜索 `AppConsistent` 在 EBS 快照列表中查找这些快照。如果命令执行失败，请查看命令输出，了解执行失败的详细原因。如果命令成功完成，但特定卷备份失败，您可以在 EBS 快照列表中排查失败的原因。

为具有共享 EBS 存储的 Windows 失效转移群集运行命令文档

您可以使用上一部分中描述的任何命令行步骤来创建基于 VSS 的快照。命令文档（`AWSEC2-VssInstallAndSnapshot` 或 `AWSEC2-CreateVssSnapshot`）必须在集群中的主节点上运行。由于辅助节点无权访问共享磁盘，因此该文档将在辅助节点上失败。如果您的主节点和辅助节点动态更改，则可以在多个节点上运行 Amazon Systems Manager Run Command 文档，并且预期该命令将在主节点上成功，而在辅助节点上会失败。

> **Note**
>
> 要自动执行备份，可以创建一个使用 `AWSEC2-VssInstallAndSnapshot` 文档的 Amazon Systems Manager 维护时段任务。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的使用维护时段（控制台）。

对基于 Windows VSS 的 EBS 快照进行问题排查

在尝试任何其他故障排除步骤之前，我们建议您验证以下详细信息。

- 确保已满足所有 创建基于 Windows VSS 的 EBS 快照的先决条件。
- 确认操作系统使用的是最新 Windows 操作系统版本支持 的 AwsVssComponents 软件包。您发现的问题可能已在较新的版本中得到解决。

主题

- 检查日志文件
- 收集其他诊断日志
- 在已配置代理的实例上使用 VSS
- 错误：解冻管道连接超时、解冻时出错、等待 VSS 冻结超时或其他超时错误
- 错误：无法调用方法。在此语言模式下，仅核心类型支持方法调用。

检查日志文件

如果您在创建基于 VSS 的 EBS 快照时遇到问题或收到错误消息，可以在 Systems Manager 控制台中查看命令输出。

对于可创建 VSS 快照的 Systems Manager 文档，可在运行时将 CollectDiagnosticLogs 参数设置为“True”。当 CollectDiagnosticLogs 参数设置为“True”时，VSS 会收集其他日志以帮助调试。有关更多信息，请参阅 收集其他诊断日志。

如果您收集诊断日志，Systems Manager 文档会将其存储在实例的以下位置：C:\ProgramData\Amazon\AwsVss\Logs*timestamp*.zip。CollectDiagnosticLogs 参数的默认值为“False”。

> **Note**
>
> 有关调试的其他帮助，可将 .zip 文件发送到 Amazon Web Services 支持。

无论您是否收集诊断日志，以下其他日志均可访问：

- %ProgramData%\Amazon\SSM\InstanceData*InstanceID*\document\orchestration*SSMCommandID*\awsrunPowerShellScript\runPowerShellScript\stdout

- %ProgramData%\Amazon\SSM\InstanceData*InstanceID*\document\orchestration*SSMCommandID*\awsrunPowerShellScript\runPowerShellScript\stderr

您还可以打开事件查看器 Windows 应用程序，然后选择 Windows 日志、应用程序以查看其他日志。要专门查看来自 EC2 Windows VSS 提供程序和卷影复制服务的事件，请根据条款 **Ec2VssSoftwareProvider** 和 **VSS** 按源进行筛选。

如果您将 Systems Manager 与 VPC 端点结合使用，并且 Systems Manager send-command API 操作（控制台中的 Run Command）失败，请确认您是否正确配置了如下端点：com.amazonaws.***region***.ec2。

如果没有定义 Amazon EC2 端点，枚举附加的 EBS 卷的调用将会失败，进而导致 Systems Manager 命令失败。有关使用 Systems Manager 设置 VPC 端点的更多信息，请参阅《Amazon Systems Manager 用户指南》中的创建 Virtual Private Cloud 端点。

收集其他诊断日志

要在通过 Systems Manager 发送命令以运行 VSS 快照文档时收集其他诊断日志，请在运行时将 `CollectDiagnosticLogs` 输入参数设置为“`True`”。建议您在排查问题时将该参数设置为“`True`”。

要查看命令行示例，请选择以下选项卡之一。

Amazon CLI

以下示例将在 Amazon CLI 中运行 `AWSEC2-CreateVssSnapshot` Systems Manager 文档：

```
aws ssm send-command \
    --document-name "AWSEC2-CreateVssSnapshot" \
    --instance-ids "i-1234567890abcdef0" \
    --parameters '{"description":["Example - create diagnostic logs at
 runtime."],"tags":["Key=tag_name,Value=tag_value"],"CollectDiagnosticLogs":
["True"]}'
```

PowerShell

以下示例将在 PowerShell 中运行 `AWSEC2-CreateVssSnapshot` Systems Manager 文档：

```
Send-SSMCommand `
    -DocumentName "AWSEC2-CreateVssSnapshot" `
    -InstanceId "i-1234567890abcdef0" `
```

```
    -Parameter @{'description'='Example - create diagnostic logs at
runtime.';'tags'='Key=tag_name,Value=tag_value';'CollectDiagnosticLogs'='True'}
```

在已配置代理的实例上使用 VSS

如果在使用代理访问 EC2 端点的实例上创建基于 VSS 的 EBS 快照时遇到问题，请验证实例上的以下设置：

- 验证是否配置代理，以便以 SYSTEM 身份运行的 Amazon Tools for Windows PowerShell 可以访问实例区域和 IMDS 中的 EC2 服务端点。
- 要支持使用系统配置的 WinHTTP 代理，请确保实例上已安装最新的 `AwsVssComponents` 版本。有关配置 WinHTTP 代理的更多信息，请参阅 Microsoft 网站上的 Windows 超文本传输协议（WINHTTP）的 Netsh 命令。

错误：解冻管道连接超时、解冻时出错、等待 VSS 冻结超时或其他超时错误

EC2 Windows VSS 提供程序可能会由于实例上的活动或服务阻止基于 VSS 的快照及时继续而超时。Windows VSS 框架提供了一个不可配置的 10 秒钟窗口，在此期间暂停与文件系统的通信。在此期间，对卷执行 `AWSEC2-CreateVssSnapshot` 快照。

以下问题可能会导致 EC2 Windows VSS Provider 在快照期间受到时间限制：

- 对卷的 I/O 过多
- 实例上 EC2 API 的响应速度缓慢
- 片段化的卷
- 与某些杀毒软件不兼容
- VSS 应用程序编写者的问题
- 当为大量 PowerShell 模块启用模块日志记录时，这样可能会导致 PowerShell 脚本运行缓慢

运行 `AWSEC2-CreateVssSnapshot` 命令文档时出现的大多数超时问题，与备份时实例上的工作负载过高有关。以下操作可以帮助您成功拍摄快照：

- 重试 `AWSEC2-CreateVssSnapshot` 命令以查看快照尝试是否成功。如果在某些情况下重试成功，减少实例负载可能会使快照成功几率更高。
- 等待片刻，让实例上的工作负载减少，然后重试该 `AWSEC2-CreateVssSnapshot` 命令。或者，当已知实例处于低压力时，您也可以尝试拍摄快照。

- 在系统上的杀毒软件关闭时尝试拍摄 VSS 快照。如果这样解决了问题，请参阅杀毒软件说明并将其配置为允许 VSS 快照。
- 如果在运行快照的同一区域内的账户中有大量 Amazon EC2 API 调用，则 API 节流可能会延迟快照操作。要降低节流带来的影响，请使用最新的 `AwsVssComponents` 程序包。该软件包利用 EC2 `CreateSnapshots` API 操作来减少变更操作的数量，例如按卷创建快照和标记。
- 如果同时运行多个 `AWSEC2-CreateVssSnapshot` 命令脚本，则可以采取以下步骤来减少并发问题。
  - 考虑在 API 活动较低的时段安排快照。
  - 如果在 Systems Manager 控制台中（或 API SendCommand 中）使用 Run Command 来运行命令脚本，则可以使用 Systems Manager 速率控制来减少并发。

    还可以使用 Systems Manager 速率控制，来减少诸如 Amazon Backup 等服务（使用 Systems Manager 运行命令脚本）的并发。
- 在 Shell 中运行命令 `vssadmin list writers`，看它是否在系统上任何写入程序的最后一个错误字段中报告任何错误。如果有写入程序报告超时错误，请考虑在实例负载较小时重新拍摄快照。
- 当使用较小的实例类型（例如，*t2 | t3 | t3a*.nano 或 *t2 | t3 | t3a*.micro）时，可能会由于内存和 CPU 限制而出现超时。以下操作可能有助于减少超时问题。
  - 在拍摄快照之前，请尝试关闭内存或 CPU 密集型应用程序。
  - 尝试在实例活动较低的时段拍摄快照。

错误：无法调用方法。在此语言模式下，仅核心类型支持方法调用。

PowerShell 语言模式未设置为 `FullLanguage` 时，会发生此错误。`AWSEC2-CreateVssSnapshot` SSM 文档要求将 PowerShell 配置为 `FullLanguage` 模式。

要验证语言模式，请在 PowerShell 控制台中的实例上运行以下命令：

```
$ExecutionContext.SessionState.LanguageMode
```

有关语言模式的更多信息，请参阅 Microsoft 文档中的 about_Language_Modes。

使用 Amazon VSS 解决方案恢复实例数据

您可以从 Amazon VSS 解决方案创建的基于 VSS 的快照恢复 Windows 实例的 EBS 卷。如果您的 Amazon VSS 解决方案快照包含 Microsoft SQL Server 数据库的备份，则可以使用 `AWSEC2-`

RestoreSqlServerDatabaseWithVss Amazon Systems Manager Automation 运行手册恢复该数据库。

数据库恢复运行手册可自动执行整个恢复过程，包括从快照创建卷并将其附加到实例。自动化利用 VSS 技术恢复数据库，使您无需停止 SQL Server 应用程序或断开任何活动连接即可进行恢复。

有关如何使用 Microsoft SQL Server 数据库运行手册的详细说明，请参阅《Microsoft SQL Server on Amazon EC2 User Guide》中的 Restore from VSS based snapshots。

自定义脚本以从 Amazon VSS 解决方案快照恢复 EBS 卷

您可以使用 RestoreVssSnapshotSampleScript.ps1 脚本作为模型来创建自己的自定义脚本，用于从 Amazon VSS 解决方案快照恢复 EBS 卷。示例脚本将执行以下任务：

- 停止实例
- 从实例中删除所有现有驱动器（如果已排除引导卷，则引导卷除外）
- 从快照创建新卷
- 通过在快照上使用设备 ID 标签将卷附加到实例
- 重新启动实例

> **Important**
>
> 以下脚本将断开附加到实例的所有卷，然后从快照创建新卷。确保已正确备份实例。不会删除旧卷。如果需要，您可以编辑脚本来删除旧卷。

从基于 VSS 的 EBS 快照还原卷

1. 下载 RestoreVssSnapshotSampleScript.zip 文件并提取文件内容。
2. 在文本编辑器中打开 RestoreVssSnapshotSampleScript.ps1，编辑脚本底部的示例调用以包含有效的 EC2 实例 ID 和 EBS 快照 ID，然后从 PowerShell 中运行该脚本。

Amazon VSS 解决方案版本历史记录

此页面包含 Amazon VSS 组件程序包的版本发行说明，以及每个受支持的 Windows Server 版本的组件和脚本版本要求。

主题

- AwsVssComponents 包版本
- Windows 操作系统版本支持

AwsVssComponents 包版本

下表介绍 Amazon VSS 组件包的已发布版本。

Version	详细信息	发行日期	可下载
2.5.1	修复了指定目标数据库参数时 SQL 数据库还原可能失败的情况。	2025 年 3 月 13 日	是
2.5.0	• 增加了读取 VSS 元数据文件和在实例上恢复 Microsoft SQL Server 数据库的功能。有关更多信息，请参阅《Microsoft SQL Server on Amazon EC2 User Guide》中的 Restore from VSS based snapshots。 • 在安装或升级 `AwsVssComponents` 软件包时增加了对就地更新选项的支持。	2025 年 1 月 17 日	是
2.4.0	增加了在创建快照时保存 VSS 元数据文件的功能。要启用此功能，请参阅 Systems Manager VSS 快照文档的参数 中的 SaveVssMetadata。	2024 年 10 月 7 日	是
2.3.3	已更新 VSS 代理以确保在快照创建期间使用 `Ec2VssProvider` 。	2024 年 6 月 25 日	是
2.3.2	修复了卸载时未删除 VSS 提供程序注册的情况。	2024 年 5 月 9 日	是
2.3.1	添加了新的默认标签 `AwsVssConfig` 以识别 Amazon VSS 创建的快照和 AMI。	2024 年 3 月 7 日	是

Version	详细信息	发行日期	可下载
2.2.1	• 增加了对使用 `DescribeInstanceAttribute` API 的支持。 • 错误修复与可靠性改进。 • 已放弃对 Windows Server 2012 和 2012 R2 的支持。AmazonVSS 组件版本 2.2.1 在 Windows Server 2012 和 2012 R2 上的安装将失败。AmazonVSS 组件版本 2.1.0 是最后一个支持 Windows Server 2012 和 2012 R2 的版本。	2024 年 1 月 18 日	是
2.1.0	增加了对使用 `CreateSnapshots` API 的支持。	2023 年 11 月 6 日	是
2.0.1	增加了对使用 WinHTTP 代理设置的支持。	2023 年 10 月 26 日	否
2.0.0	Amazon VSS 组件增加了创建快照和 AMI 的功能，从而实现了与 PowerShell 模块日志记录、脚本块日志记录和转录功能的兼容。	2023 年 4 月 28 日	否
1.3.2.0	修复了未正确报告安装失败的情况。	2022 年 5 月 10 日	否
1.3.1.0	• 修复了域控制器上与 NTDS VSS 写入器日志记录错误相关的快照失败。 • 修复了卸载版本 1.0 VSS 提供程序时的 VSS 代理错误。	2020 年 2 月 6 日	是

Version	详细信息	发行日期	可下载
1.3.00	• 通过减少不需要的详细程度来改进日志记录。 • 修复了安装过程中的区域化问题。 • 修正了某些供应商注册错误条件的返回代码。 • 修复了各种安装问题。	2019 年 3 月 19 日	否
1.2.00	• 将命令行参数 -nw（无写入器）和 -copy（仅复制）添加到代理。 • 修复了由不正确的内存分配调用引起的 EventLog 错误。	2018 年 11 月 15 日	否
1.1	修复了 Amazon VSS 组件被错误用作默认 Windows 备份和还原提供程序的问题。	2017 年 12 月 12 日	否
1.0	初始版本。	2017 年 11 月 20 日	否

Windows 操作系统版本支持

下表显示了应在 Amazon EC2 上每个版本的 Windows Server 上运行的 Amazon VSS 解决方案版本。

Windows Server 版本	AwsVssComponents 版本	AWSEC2-VssInstallAndSnapshot 版本名称	AWSEC2-CreateVssSnapshot 版本名称
Windows Server 2025	默认	默认	默认

Windows Server 版本	AwsVssComponents 版本	AWSEC2-VssInstallAndSnapshot 版本名称	AWSEC2-CreateVssSnapshot 版本名称
Windows Server 2022	默认	默认	默认
Windows Server 2019	默认	默认	默认
Windows Server 2016	默认	默认	默认
Windows Server 2012 R2	2.1.0	不支持	2012R2
Windows Server 2012	2.1.0	不支持	2012R2
Windows Server 2008 R2	1.3.1.0	不支持	2008R2

Amazon EC2 上的对象存储、文件存储及文件缓存

云文件存储是一种在云中存储数据的方法，允许服务器和应用程序通过共享文件系统访问数据。这种兼容性使得云文件存储非常适合依赖共享文件系统的工作负载，并且实现了无需更改代码的简单集成。

现在有许多文件存储解决方案，从使用块存储作为基础的计算实例上的单节点文件服务器（没有扩展性或冗余来保护数据），到自助集群解决方案，再到完全托管的解决方案，不一而足。以下内容介绍了一些由 Amazon 提供的用于与 Amazon EC2 实例结合使用的存储服务。

内容

- 将 Amazon S3 与 Amazon EC2 实例结合使用
- 将 Amazon EFS 与 Amazon EC2 Linux 实例结合使用
- 将 Amazon FSx 与 Amazon EC2 实例结合使用
- 将 Amazon File Cache 与 Amazon EC2 实例结合使用

将 Amazon S3 与 Amazon EC2 实例结合使用

Amazon Simple Storage Service（Amazon S3）是一种对象存储服务，提供行业领先的可扩展性、数据可用性、安全性和性能。您可以使用 Amazon S3 为数据湖、网站、备份和大数据分析等一系列用例存储和检索任意数量的数据，这些数据可以来自 Amazon EC2 实例或互联网上的任何地方。有关更多信息，请参阅什么是 Amazon S3？

有两种方法可以从 Amazon EC2 实例访问 Amazon S3 数据：

- 文件访问：使用 Amazon S3 Files 将 S3 存储桶作为高性能文件系统挂载到实例。
- 对象访问：使用 Amazon S3 API、Amazon CLI、Amazon SDK 或 wget 等工具在 S3 之间复制对象。

使用 Amazon S3 Files 进行文件访问

Amazon S3 Files 是一个无服务器文件系统，允许您将 S3 通用存储桶作为高性能文件系统挂载到计算实例。借助 S3 Files，您可以通过本地挂载路径上的标准文件系统操作（例如读取和写入），将 S3 对象作为文件进行访问。

您可以在启动时或启动后正在运行的实例上，将 S3 文件系统挂载到 EC2 实例。

先决条件

在使用 EC2 实例设置 S3 Files 之前，务必先满足以下条件：

- 拥有一个 S3 文件系统，以及至少一个处于可用状态的挂载目标。有关创建 S3 文件系统的信息，请参阅《Amazon S3 用户指南》中的使用 Amazon S3 Files。
- EC2 Linux 实例附加到实例配置文件。有关挂载文件系统所需权限的信息，请参阅《Amazon S3 用户指南》中的 IAM 角色和策略。
- 安全组允许实例与文件系统挂载目标之间的 NFS 流量（端口 2049）。有关所需安全组设置的信息，请参阅《Amazon S3 用户指南》中的安全组。

在启动时使用 EC2 控制台将文件系统挂载到 EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 选择启动实例。
3. 在 Network settings（网络设置）下，执行以下操作：
   a. 选择编辑。
   b. 对于 Subnet (子网)，选择一个子网。
   c. 选择默认安全组，以确保 EC2 实例可以访问 S3 文件系统。您不能使用该安全组通过安全外壳 (SSH) 访问您的 EC2 实例。要通过 SSH 进行访问，您可以稍后编辑默认安全组，并添加一个允许 SSH 的规则/新安全组。您可以使用以下设置：
      i. 类型：SSH
      ii. 协议：TCP
      iii. 端口范围：22
      iv. 源：任何位置 0.0.0.0/0
4. 在配置存储下，执行以下操作：
   a. 在文件系统下，选择 S3 Files。
   b. 请选择 Add shared file system（添加共享文件系统）。
   c. 对于 S3 文件系统，文件系统会根据您在网络设置中选择的子网显示在可用区中。选择要挂载的 S3 文件系统。如果没有任何文件系统，请选择创建新的文件系统创建一个新的。
   d. 在 EC2 实例上输入要在其中挂载文件系统的本地挂载路径（例如 `/mnt/s3files`）。
   e. 将生成一个命令来挂载文件系统并将其添加到 fstab。您可以将此命令添加到高级详细信息下的用户数据字段中。然后，EC2 实例将配置为在启动或重启时挂载 S3 文件系统。您也可以在 EC2 实例启动后运行这些命令。
5. 在高级详细信息下，将实例配置文件附加到实例。IAM 角色必须具有挂载文件系统和访问 S3 存储桶的权限。有关所需权限的更多信息，请参阅《Amazon S3 用户指南》中的 IAM 角色和策略。
6. 选择启动实例。

   实例启动后，将安装所需的软件实用程序并挂载文件系统。您可以通过导航到本地挂载路径来查看文件系统。

在启动后将文件系统挂载到 EC2 实例

1. 通过 Secure Shell（SSH）或 EC2 控制台中的 EC2 Instance Connect 连接到 EC2 实例。
2. 要挂载 S3 文件系统，请使用挂载帮助程序 `amazon-efs-utils`。根据 Linux 发行版，使用以下命令之一安装 `amazon-efs-utils` 软件包：

   a. 如果您使用的是 Amazon Linux，请运行以下命令从 Amazon 存储库安装 efs-utils：

   ```
   sudo yum -y install amazon-efs-utils
   ```

   b. 如果您使用的是其他受支持的 Linux 发行版，请运行以下命令：

   ```
   curl https://amazon-efs-utils.aws.com/efs-utils-installer.sh | sudo sh -s -- --
   install
   ```

   c. 有关其他 Linux 发行版，请参阅 GitHub 上的 efs-utils 存储库。
3. 使用以下命令为文件系统挂载点创建一个目录：

   ```
   sudo mkdir {path/to/mount}
   ```
4. 挂载 S3 文件系统：

   ```
   FS="{YOUR_FILE_SYSTEM_ID}"
   sudo mount -t s3files $FS:/ {path/to/mount}
   ```
5. 确认文件系统已挂载：

   ```
   df -h {path/to/mount}
   ```

将 S3 存储桶中的对象作为文件查看

完成上述步骤后，您现在可以使用标准文件系统操作在本地挂载路径上将 S3 对象作为文件进行读取和写入。如果 S3 存储桶中有对象，则可以使用以下命令将其作为文件进行查看：

```
ls {path/to/mount}
```

基于对象的访问

您可以使用 S3 API、Amazon CLI、Amazon SDK 或标准 HTTP 工具在 Amazon S3 之间复制文件。如果您具有所需的权限，可以使用以下方法之一在 Amazon S3 和实例之间复制文件。

wget

> **Note**
>
> 此方法仅适用于公有对象。如果对象不是公有的，您会收到 `ERROR 403: Forbidden` 消息。如果您收到此错误，您必须使用 Amazon S3 控制台、Amazon CLI、Amazon API、Amazon 开发工具包或 Amazon Tools for Windows PowerShell，并且您必须拥有所需的权限。有关更多信息，请参阅《Amazon S3 用户指南》中的 Amazon S3 的身份和访问管理以及下载对象。

wget 实用程序是一个 HTTP 和 FTP 客户端，可用于从 Amazon S3 下载公有对象。该实用工具在 Amazon Linux 和大多数其他分发版中均为默认安装，可在 Windows 上下载安装。要下载 Amazon S3 对象，请使用以下命令（替换要下载的对象的 URL）。

```
[ec2-user ~]$ wget https://amzn-s3-demo-bucket.s3.amazonaws.com/path-to-file
```

PowerShell

您可以使用 Amazon Tools for Windows PowerShell 将对象移入和移出 Amazon S3。

您可以使用 Copy-S3Object cmdlet 将 Amazon S3 对象复制到 Windows 实例，如下所示。

```
Copy-S3Object `
    -BucketName amzn-s3-demo-bucket `
    -Key path-to-file `
    -LocalFile my_copied_file.ext
```

您也可以在 Windows 实例上使用 Web 浏览器打开 Amazon S3 控制台。

Amazon CLI

您可以使用 Amazon Command Line Interface（Amazon CLI）从 Amazon S3 下载受限制的项目和上传项目。有关更多信息（例如如何安装和配置这些工具），请参阅 Amazon Command Line Interface 详细信息页。

aws s3 cp 命令与 Unix cp 命令类似。您可以将文件从 Amazon S3 复制到您的实例，从您的实例复制到 Amazon S3，可以将文件在不同 Amazon S3 位置之间复制。

使用以下命令可将对象从 Amazon S3 复制到实例：

```
aws s3 cp s3://amzn-s3-demo-bucket/my_folder/my_file.ext my_copied_file.ext
```

使用以下命令可将对象从实例重新复制到 Amazon S3：

```
aws s3 cp my_copied_file.ext s3://amzn-s3-demo-bucket/my_folder/my_file.ext
```

aws s3 sync 命令可以将整个 Amazon S3 存储桶同步到本地目录位置。这可以用于下载数据集并使本地副本随远程集保持更新。如果您对 Amazon S3 存储桶拥有合适权限，则当您最后在命令中将源与目标位置反转时，可以将本地目录备份推送到云。

使用以下命令可将整个 Amazon S3 存储桶下载到实例上的本地目录：

```
aws s3 sync s3://amzn-s3-demo-source-bucket local_directory
```

Amazon S3 API

您可以使用 API 访问 Amazon S3 中的数据。您可以使用此 API 帮助开发应用程序，并且可以将其与其他 API 和 SDK 集成。有关更多信息，请参阅《Amazon Simple Storage Service API Reference》中的 Code examples for Amazon S3 using Amazon SDKs。

将 Amazon EFS 与 Amazon EC2 Linux 实例结合使用

> **Note**
>
> Amazon EFS 在 Windows 实例上不受支持。

Amazon EFS 提供可扩展文件存储以供和 Amazon EC2 一起使用。您可以使用 EFS 文件系统作为在多个实例上运行的工作负载和应用程序的通用数据源。有关更多信息，请参阅Amazon Elastic File System产品页。

本教程向您展示如何在实例启动期间，使用 Amazon EFS 快速创建向导创建和附加 Amazon EFS 文件系统。有关如何使用 Amazon EFS 控制台创建文件系统的教程，请参阅 Amazon Elastic File System

User Guide（《Amazon File System 用户指南》）中的 Getting started with Amazon Elastic File System（Amazon Elastic File System 入门）。

> **Note**
>
> 使用 EFS 快速创建 EFS 文件系统时，将使用以下服务推荐设置创建文件系统：
>
> - 已启用自动备份。
> - 在所选 VPC 中管理挂载目标。
> - 通用性能模式。
> - 突增吞吐量模式。
> - 使用 Amazon EFS 的默认密钥 (aws/elasticfilesystem) 启用静态数据加密。
> - 使用 30 天策略启用 Amazon EFS 生命周期管理。

任务

- 测试 EFS 文件系统
- 删除 EFS 文件系统

测试 EFS 文件系统

您可以连接到实例并验证文件系统是否已挂载到您指定的目录（例如，/mnt/efs）。

验证文件系统是否已装载

1. 连接到您的实例。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。
2. 从每个实例的终端窗口，运行 df -T 命令以验证是否已挂载 EFS 文件系统。

```
$ df -T
Filesystem     Type              1K-blocks    Used          Available Use% Mounted
 on
/dev/xvda1     ext4                8123812 1949800            6073764  25% /
devtmpfs       devtmpfs            4078468      56            4078412   1% /dev
tmpfs          tmpfs               4089312       0            4089312   0% /dev/shm
efs-dns        nfs4       9007199254740992       0   9007199254740992   0% /mnt/efs
```

请注意，文件系统的名称（在示例输出中显示为 *efs-dns*）具有以下格式。

```
file-system-id.efs.aws-region.amazonaws.com:/
```

3. （可选）在该实例的文件系统中创建文件，然后验证您是否可以从另一实例查看该文件。

   a. 在该实例中，运行以下命令来创建文件。

   ```
   $ sudo touch /mnt/efs/test-file.txt
   ```

   b. 在另一个实例中，运行以下命令来查看文件。

   ```
   $ ls /mnt/efs
   test-file.txt
   ```

删除 EFS 文件系统

如果您不再需要文件系统，可将其删除。

要删除文件系统

1. 访问 https://console.aws.amazon.com/efs/，打开 Amazon Elastic File System 控制台。
2. 选择要删除的文件系统。
3. 选择 Actions、Delete file system。
4. 当系统提示您确认时，输入文件系统 ID 并选择删除文件系统。

将 Amazon FSx 与 Amazon EC2 实例结合使用

借助 Amazon FSx 系列服务，可以轻松启动、运行和扩缩由常见的商业和开源文件系统提供支持的共享存储。您可以使用新的启动实例向导在启动时自动将以下类型的 Amazon FSx 文件系统附加到您的 Amazon EC2 实例：

- Amazon FSx for NetApp ONTAP 使用 NetApp ONTAP 的常见数据访问和管理功能在 Amazon 云中提供完全托管式共享存储。
- Amazon FSx for OpenZFS 提供了由常见 OpenZFS 文件系统提供支持的完全托管的、经济高效的共享存储。

> **Note**
>
> - 此功能仅在新启动实例向导中可用。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例
> - 启动时无法挂载 Amazon FSx for Windows File Server 和 Amazon FSx for Lustre 文件系统。启动后必须手动挂载这些文件系统。

您可以选择挂载先前创建的现有文件系统，也可以创建一个新的文件系统以在启动时挂载到实例。

主题

- 安全组和用户数据脚本
- 在启动时挂载 Amazon FSx 文件系统

安全组和用户数据脚本

当您使用启动实例向导将 Amazon FSx 文件系统挂载到实例时，您可以选择是否自动创建和附加启用对文件系统的访问所需的安全组，以及是否自动包含挂载文件系统和使其可用所需的用户数据脚本。

主题

- 安全组
- 用户数据脚本

安全组

如果您选择自动创建启用对文件系统的访问所需的安全组，启动实例向导将创建并附加两个安全组 - 一个安全组附加到实例，另一个安全组附加到文件系统。有关安全组要求的更多信息，请参阅使用 Amazon VPC 进行 FSx for ONTAP 文件系统访问控制和使用 Amazon VPC 进行 FSx for OpenZFS 文件系统访问控制。

我们将标签 `Name=instance-sg-1` 添加到已创建并附加到实例的安全组中。标签中的值在启动实例向导每次为 Amazon FSx 文件系统创建安全组时都会自动递增。

安全组包括以下输出规则，但不包括入站规则。

出站规则

协议类型	端口号	目标位置
UDP	111	*######*
UDP	20001 – 20003	*######*
UDP	4049	*######*
UDP	2049	*######*
UDP	635	*######*
UDP	4045 – 4046	*######*
TCP	4049	*######*
TCP	635	*######*
TCP	2049	*######*
TCP	111	*######*
TCP	4045 – 4046	*######*
TCP	20001 – 20003	*######*
全部	全部	*######*

已创建并附加到文件系统的安全组都通过 `Name=fsx-sg-`*1* 进行标记。标签中的值在启动实例向导每次为 Amazon FSx 文件系统创建安全组时都会自动递增。

该安全组包括以下规则。

入站规则

协议类型	端口号	来源
UDP	2049	*#####*
UDP	20001 – 20003	*#####*

协议类型	端口号	来源
UDP	4049	*#####*
UDP	111	*#####*
UDP	635	*#####*
UDP	4045 – 4046	*#####*
TCP	4045 – 4046	*#####*
TCP	635	*#####*
TCP	2049	*#####*
TCP	4049	*#####*
TCP	20001 – 20003	*#####*
TCP	111	*#####*

出站规则

协议类型	端口号	目标位置
全部	全部	0.0.0.0/0

用户数据脚本

如果您选择自动附加用户数据脚本，启动实例向导会将以下用户数据添加到实例。此脚本安装必要的软件包、挂载文件系统并更新您的实例设置，以便在实例重新启动时，文件系统将自动重新挂载。

```
#cloud-config
package_update: true
package_upgrade: true
runcmd:
- yum install -y nfs-utils
- apt-get -y install nfs-common
- svm_id_1=svm_id
```

```
- file_system_id_1=file_system_id
- vol_path_1=/vol1
- fsx_mount_point_1=/mnt/fsx/fs1
- mkdir -p "${fsx_mount_point_1}"
- if [ -z "$svm_id_1" ]; then printf "\n${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev
 0 0\n" >> /etc/fstab; else printf "\n${svm_id_1}.${file_system_id_1}.fsx.eu-
north-1.amazonaws.com:/${vol_path_1} ${fsx_mount_point_1} nfs4
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0
 0\n" >> /etc/fstab; fi
- retryCnt=15; waitTime=30; while true; do mount -a -t nfs4 defaults; if [ $? = 0 ] ||
 [ $retryCnt -lt 1 ]; then echo File system mounted successfully; break; fi; echo File
 system not available, retrying to mount.; ((retryCnt--)); sleep $waitTime; done;
```

在启动时挂载 Amazon FSx 文件系统

要在启动时挂载新的或现有的 Amazon FSx 文件系统

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 Instances（实例），然后选择 Launch instance（启动实例）以打开启动实例向导。
3. 在 Application and OS Images（应用程序和操作系统映像）部分中，请选择要使用的 AMI。
4. 在 Instance type（实例类型）部分中，选择实例类型。
5. 在 Key pair（密钥对）部分中，选择现有密钥对或创建新密钥对。
6. 在 Network settings（网络设置）部分，执行以下操作：

    a. 选择编辑。

    b. 如果您想挂载现有文件系统，对于 Subnet（子网），请选择文件系统的首选子网。我们建议您将实例作为文件系统的首选子网启动到同一可用区中，以优化性能。

    如果您想创建新的文件系统以挂载到实例，对于 Subnet（子网），请选择要在其中启动实例的子网。

    > **Important**
    >
    > 您必须选择一个子网才能在新启动实例向导中启用 Amazon FSx 功能。如果不选择子网，将无法挂载现有文件系统或创建新的文件系统。

7. 在 Storage（存储）部分中，执行以下操作：

   a. 根据需要配置卷。

   b. 展开 File systems（文件系统）部分，然后选择 FSx。

   c. 请选择 Add shared file system（添加共享文件系统）。

   d. 对于 File system（文件系统），请选择要挂载的文件系统。

   > **Note**
   >
   > 该列表显示所选区域内您账户中的所有 Amazon FSx for NetApp ONTAP 和 Amazon FSx for OpenZFS 文件系统。

   e. 要自动创建并附加启用对文件系统的访问所需的安全组，请选择 Automatically create and attach security groups（自动创建和附加安全组）。如果您更喜欢手动创建安全组，请清除该复选框。有关更多信息，请参阅 安全组。

   f. 要自动附加挂载文件系统所需的用户数据脚本，请选择 Automatically mount shared file system by attaching required user data script（通过附加所需的数据用户数据脚本自动挂载共享文件系统）。如果更喜欢手动提供用户数据脚本，请清除该复选框。有关更多信息，请参阅用户数据脚本。

8. 在 Advanced（高级）部分中，根据需要配置其他实例设置。

9. 选择启动。

将 Amazon File Cache 与 Amazon EC2 实例结合使用

Amazon File Cache 在 Amazon 上提供一个完全托管的高速缓存，可以更轻松地处理文件数据，无论数据存储在何处。Amazon File Cache 可作为临时的高性能存储位置，供存储在本地文件系统、Amazon 文件系统和 Amazon Simple Storage Service (Amazon S3)存储桶中的数据使用。您可以利用这一功能，以统一的视图和极快的速度(亚毫秒级延迟和高吞吐量)将分散的数据集提供给 Amazon 上基于文件的应用程序。有关更多信息，请参阅《Amazon File Cache 用户指南》。

Amazon File Cache 可与最流行的 Linux AMI 配合使用，并与基于 x86 的实例类型和 Graviton 实例类型兼容。您可以使用开源的 Lustre 客户端从 Amazon EC2 实例访问缓存。您可以挂载缓存，然后使用标准 Linux 命令处理缓存中的文件和目录。Amazon EC2 实例可以从同一虚拟私有云（VPC）内的其他可用区访问您的缓存，但前提是您的网络配置允许在 VPC 内跨子网访问。您也可以在共享 VPC 中创建缓存。

要开始使用，请参阅《Amazon File Cache User Guide》中的 Getting started with Amazon File Cache。

管理 Amazon EC2 资源

资源是可以使用的实体。Amazon EC2 会在您使用服务的功能时创建资源。例如，Amazon EC2 资源包括映像、实例、实例集、密钥对和安全组。所有 Amazon EC2 资源类型都包含可描述资源的属性，例如名称、描述、资源标识符和 Amazon 资源名称（ARN）。

Amazon EC2 资源取决于所在的 Amazon 区域或可用区。例如，亚马逊机器映像（AMI）是特定于某个 Amazon 区域，但从 AMI 启动的实例是特定于您启动该实例的区域。您可以使用 Amazon EC2 资源的 ARN 在权限策略中指定 Amazon EC2 资源。

您的 Amazon Web Services 账户具有 Amazon EC2 的默认配额。这些配额定义了您可以创建的最大资源数量。例如，正在运行的实例中 vCPU 的最大数量有配额。如果启动实例或启动已停止的实例会导致您超出配额，则操作将失败。

您可以使用资源 ID 或标签按区域搜索 Amazon Web Services 账户中的特定资源。要搜索多个区域中的特定资源或资源类型，请使用 Amazon EC2 全局视图。

内容

- 为您的 Amazon EC2 资源选择区域
- 查找 Amazon EC2 资源
- 使用 Amazon 全局视图查看跨区域的资源
- 标记 Amazon EC2 资源
- Amazon EC2 Service Quotas

为您的 Amazon EC2 资源选择区域

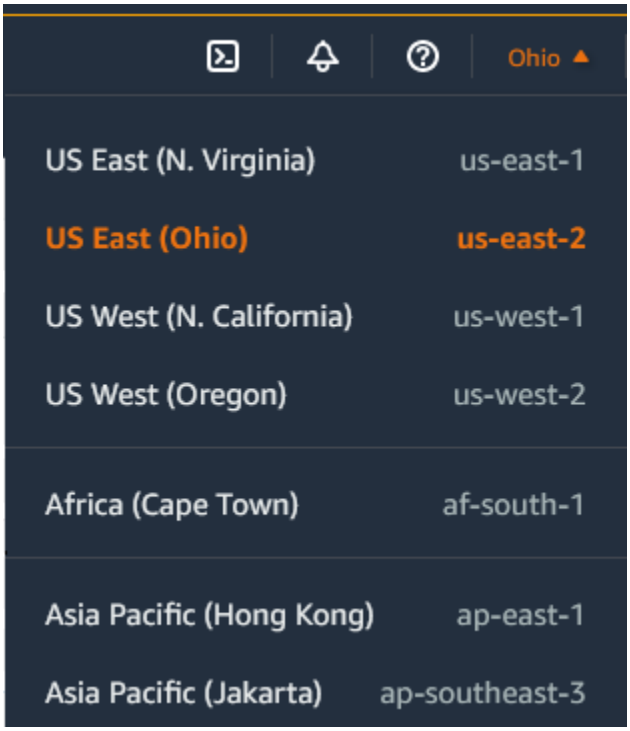
Amazon EC2 资源取决于所在的 Amazon 区域或可用区。创建 Amazon EC2 资源时，您可选择该资源的区域。

注意事项

- 某些 Amazon 资源可能并非在所有区域都可用。在区域中开始创建资源之前，确保您可以在区域中创建所需的所有 Amazon 资源。
- 默认情况下，会禁用某些区域。您必须先启用这些区域，然后才能使用。有关更多信息，请参阅 Amazon Regions。

使用控制台为资源选择区域

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航栏中，选中 Regions（区域）选择器，然后选择区域。



3. 区域选择器包括可在您的 Amazon Web Services 账户中使用的所有资源。选择列表底部附近带下划线的文本，查看您的账户未启用的区域。

使用 Amazon CLI 为资源选择区域

您可以将 Amazon CLI 配置为使用默认区域。如果您未在命令中指定区域，则使用默认区域。要对特定命令使用不同的区域，请添加以下选项。

```
--region us-east-1
```

使用 Tools for PowerShell 为资源选择区域

您可以将 Tools for Windows PowerShell 配置为使用默认区域。如果您未在命令中指定区域，则使用默认区域。要对特定命令使用不同的区域，请添加以下参数。

```
-Region us-east-1
```

查找 Amazon EC2 资源

您可以使用 Amazon EC2 控制台获取某些类型的资源的列表。您可以使用相应命令或 API 操作获取每种类型的资源的列表。如果您拥有许多资源，可以筛选结果以仅包含或排除符合特定标准的资源。

目录

- 使用控制台列出并筛选资源
- 使用命令行列出并筛选
- 使用 Amazon EC2 全局视图查看跨区域的资源

使用控制台列出并筛选资源

目录

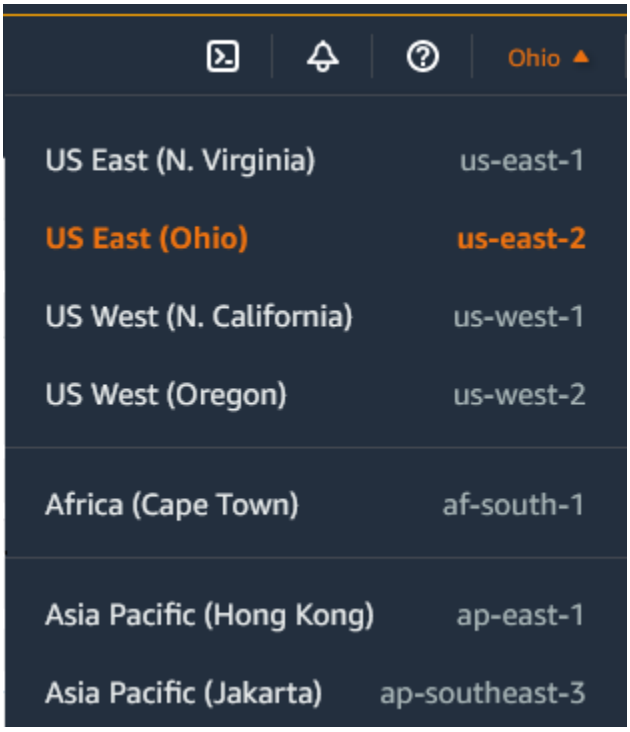
- 使用控制台列出资源
- 使用控制台筛选资源
  - 支持的筛选条件
- 使用控制台保存筛选条件
  - 主要功能
  - 创建筛选条件集
  - 修改筛选条件集
  - 删除筛选条件集

使用控制台列出资源

您可以使用控制台查看最常用的 Amazon EC2 资源类型。要查看其他资源，请使用命令行界面或 API 操作。

要使用控制台列出 EC2 资源

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. Amazon EC2 资源是特定于 Amazon Web Services 区域。在导航栏中，从区域选择器中选择一个区域。

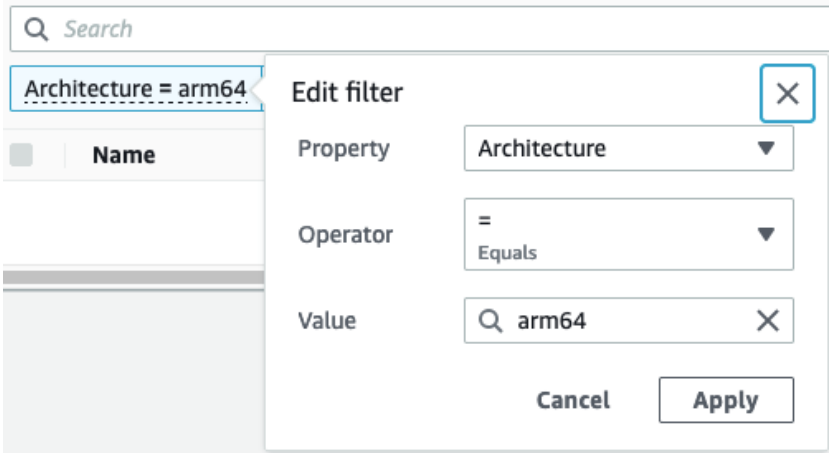


3. 在导航窗格中，选择与资源类型对应的选项。例如，要列出所有实例，请选择实例。

使用控制台筛选资源

筛选资源列表

1. 在导航窗格中，选择资源类型 (例如，Instances)。
2. 选择搜索字段。
3. 从列表中选择筛选条件。
4. 选择运算符，例如 = (等于) 。某些属性具有更多可供选择的运算符。请注意，并非所有页面都支持选择运算符。
5. 选择筛选条件值。
6. 要编辑选定的筛选条件，请选择过滤条件令牌 (蓝色框) ，进行所需的编辑，然后选择 Apply (应用) 。请注意，并非所有页面都支持编辑选定的筛选条件。



7. 完成后，删除筛选条件。

支持的筛选条件

Amazon EC2 控制台支持两种类型的筛选。

- API 筛选发生在服务器端。该筛选应用于 API 调用，可减少服务器返回的资源数量。它可以对大型资源集进行快速筛选，并可降低服务器和浏览器之间的数据传输时间和成本。API 筛选支持 =（等于）和 :（包含）运算符，并且始终区分大小写。
- 客户端筛选发生在客户端。它可以对浏览器中现有的数据（也就是 API 已返回的数据）进行筛选。客户端筛选与 API 筛选结合使用，可以在浏览器中筛选出较小的数据集。除 =（等于）和 :（包含）运算符外，客户端筛选还支持范围运算符 [例如 >=（大于等于）] 和否定（反向）运算符 [例如 !=（不等于）]。

Amazon EC2 控制台支持以下类型的搜索：

按关键字搜索

按关键词搜索是一种自由文本搜索，可让您在所有资源属性或标签中搜索特定的值，而无需指定要搜索的属性或标签键。

> **Note**
>
> 所有关键字搜索都使用客户端筛选。

要按关键字进行搜索，请在搜索框中输入或粘贴要查找的内容，然后按 Enter。例如，搜索 123 时将会比对任何属性（例如 IP 地址、实例 ID、VPC ID 或 AMI ID）或任何标签（例如 Name）中包含 123 的所有实例。如果自由文本搜索返回不需要的匹配项，请再应用其他筛选条件。

按属性搜索

按属性搜索可让您在所有资源中搜索特定属性。

> **Note**
>
> 属性搜索可以使用 API 筛选或客户端筛选，具体取决于所选的属性。执行属性搜索时，会相应地对属性进行分组。

例如，您可以搜索所有实例的 Instance state 属性，以便仅返回处于 `stopped` 状态的实例。要实现此目的，应按照以下步骤进行：

1. 在实例屏幕上的搜索字段中，开始输入 `Instance state`。输入字符时，实例状态将显示两种类型的筛选条件：API 筛选条件和客户端筛选条件。
2. 要在服务器端进行搜索，请在 API 筛选条件下选择 实例状态。要在客户端进行搜索，请在客户端筛选条件下选择实例状态（客户端）。

   这时会显示所选属性的可能运算符列表。
3. 选择 =（等于）运算符。

   这时会显示所选属性和运算符的可能值列表。
4. 从列表中选择已停止。

按标签搜索

借助按标签搜索功能，您可以按标签键或标签值来筛选当前显示的表中的资源。

标签搜索使用 API 筛选或客户端筛选方式，具体取决于 Preferences（首选项）窗口中的设置。

确保对标签执行 API 筛选

1. 打开 Preferences（首选项）窗口。
2. 清除使用正则表达式匹配复选框。如果选中此复选框，则将执行客户端筛选。
3. 选择使用区分大小写的匹配复选框。如果清除此复选框，则将执行客户端筛选。
4. 选择确认。

按标签搜索时，您可以使用以下值：

- （空）– 查找具有指定标签键，但必须不具有标签值的所有资源。
- All values（所有值）– 查找具有指定标签键和任何标签值的所有资源。
- Not tagged（未添加标签）– 搜索不具有指定标签键的所有资源。
- 显示的值 – 查找具有指定标签键和指定标签值的所有资源。

您可以使用以下技术来增强或优化搜索：

逆向搜索

反向搜索可让您搜索与指定的值不匹配的资源。在实例和 AMI 屏幕中，反向搜索的方法是首先选择 != （不等于）或 !: （不包含）运算符，然后选择一个值。在其他屏幕中，反向搜索可通过在搜索关键词前加上感叹号（!）字符来执行。

> **Note**
>
> 只有客户端筛选条件上的关键字搜索和属性搜索支持反向搜索。API 筛选条件上的属性搜索不支持它。

例如，您可以搜索所有实例的实例状态属性，以便包含处于 `terminated` 状态的所有实例。要实现此目的，应按照以下步骤进行：

1. 在实例屏幕上的搜索字段中，开始输入 `Instance state`。输入字符时，实例状态将显示两种类型的筛选条件：API 筛选条件和客户端筛选条件。
2. 在 Client filters（客户端筛选条件）下，选择 Instance state (client) [实例状态（客户端）]。只有客户端筛选条件支持反向搜索。

   这时会显示所选属性的可能运算符列表。
3. 选择 != （不等于），然后选择 terminated（已终止）。

要根据实例状态属性筛选实例，您还可以使用实例状态列中的搜索图标 ( )。带加号 (+) 的搜索图标将显示与该属性匹配的所有实例。带减号 (-) 的搜索图标将排除与该属性匹配的所有实例。

以下是使用反向搜索的另一个示例：要列出未分配名为 `launch-wizard-1` 的安全组的所有实例，则在 Client filters（客户端筛选条件）下，按 Security group name（安全组名称）属性进行搜索，然后选择 != 并在搜索栏中输入 `launch-wizard-1`。

部分搜索

使用部分搜索，可以搜索部分字符串值。要执行部分搜索，请仅输入要搜索的关键字的一部分。在 Instances（实例）和 AMI 屏幕中，部分搜索只能通过 :（包含）运算符来执行。在其他屏幕中，您可以选择该客户端筛选条件属性，然后立即仅输入要搜索的关键词的一部分。例如，在 Instance type（实例类型）屏幕上，要搜索所有 `t2.micro`、`t2.small` 和 `t2.medium` 实例，则按 Instance Type（实例类型）属性进行搜索，然后输入关键词 `t2`。

正则表达式搜索

要使用正则表达式搜索，必须在首选项窗口中选中使用正则表达式匹配复选框。

当需要匹配字段中具有特定模式的值时，可以使用正则表达式。例如，要搜索以 `s` 开头的值，请搜索 `^s`。要搜索以 `xyz` 结尾的值，请搜索 `xyz$`。或者要搜索以数字开头，后跟一个或多个字符的值，请搜索 `[0-9]+.*`。

> **Note**
>
> 只有客户端筛选条件上的关键字搜索和属性搜索支持正则表达式搜索。API 筛选条件上的属性搜索不支持它。

区分大小写搜索

要使用区分大小写搜索，必须在首选项窗口中选中使用区分大小写的匹配复选框。区分大小写的首选项仅适用于客户端筛选条件和标签筛选条件。

> **Note**
>
> API 筛选条件始终区分大小写。

通配符搜索

使用 * 通配符匹配零个或多个字符。使用 ？通配符匹配零个或一个字符。例如，假设您有一个包含值 `prod`、`prods`、和 `production` 的数据集，搜索 `prod*` 将会匹配所有值，而 `prod?` 只匹配 `prod` 和 `prods`。要使用文字值，请使用反斜杠（`\`）进行转义。例如，“`prod\*`”将匹配 `prod*`。

> **Note**
>
> 仅 API 筛选条件上的属性和标签搜索支持通配符搜索。客户端筛选条件上的关键词搜索以及属性和标签搜索不支持通配符。

合并搜索

通常，具有相同属性的多个筛选条件会自动以 OR 连接。例如，搜索 `Instance State : Running` 和 `Instance State : Stopped` 会返回正在运行或已停止的所有实例。要使用 AND 连接搜索，请跨不同的属性进行搜索。例如，搜索 `Instance State : Running` 和 `Instance Type : c4.large` 只会返回为类型为 `c4.large` 且处于运行状态的实例。

使用控制台保存筛选条件

保存的筛选条件集是一组自定义的筛选条件，您可以创建并重复使用这些筛选条件来高效查看自己的 Amazon EC2 资源。此功能有助于简化您的工作流，实现快速访问特定的资源视图。

保存的筛选条件集仅受 Amazon EC2 控制台支持，并且当前仅适用于卷页面。

主要功能

- 自定义：创建根据您的需求量身定制的筛选条件集。例如，您可以创建一个筛选条件集来仅显示在指定日期之后创建的 gp3 卷。
- 默认筛选条件：为某页面设置默认筛选条件集，当您导航到该页面时，系统会自动应用默认筛选条件。如果未设置默认值，则不应用任何筛选条件。
- 轻松应用：选择保存的筛选条件集即可立即应用。然后，Amazon EC2 会显示相关资源，其中活动筛选条件用蓝色标记表示。
- 灵活性：根据需要创建、修改和删除筛选条件集。

创建筛选条件集

创建新的筛选条件集

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Volumes。

> **Note**
>
> 保存的筛选器集目前仅适用于卷。

3. 在搜索字段中，为您的筛选条件集选择筛选条件。
4. 选择清除筛选条件按钮旁边的箭头，然后选择保存新的筛选条件集。
5. 在保存筛选条件集窗口中，执行下面的操作：

   a. 对于筛选条件集名称，输入筛选条件集的名称。

   b. （可选）对于筛选条件集描述，输入筛选条件集的描述。

   c. （可选）要将筛选条件集设置为默认筛选条件，请选中设为默认值复选框。

   > **Note**
   >
   > 每次您打开控制台页面时，都会自动应用默认筛选条件。

   d. 选择保存。

修改筛选条件集

修改筛选条件集

1. 从保存的筛选条件集列表中，选择要修改的筛选条件。
2. 要添加筛选条件，请在搜索字段中选择要添加到筛选条件集中的筛选条件。要删除集中的筛选条件，请选择筛选条件标记上的 X。
3. 选择清除筛选条件按钮旁边的箭头，然后选择修改筛选条件集。
4. 在修改筛选条件集窗口中，执行下面的操作：

   a. （可选）要将筛选条件集设置为默认筛选条件，请选中设为默认值复选框。

   > **Note**
   >
   > 每次您打开控制台页面时，都会自动应用默认筛选条件。

   b. 选择 Modify(修改)。

删除筛选条件集

删除筛选条件集

1. 从保存的筛选条件集列表中，选择要删除的筛选条件。
2. 选择清除筛选条件按钮旁边的箭头，然后选择删除筛选条件集。
3. 在删除筛选条件集窗口中，查看筛选条件以确认这是要删除的筛选条件，然后选择删除。

使用命令行列出并筛选

每个资源类型都有对应的 API 操作，以供您用来描述、列出或获取该类型的资源。生成的资源列表可能很长，因此筛选结果以仅包括符合特定条件的资源可能会更快、更有用。

筛选注意事项

- 在单个请求中，您最多可以指定 50 个筛选条件，以及为每个筛选条件最多指定 200 个值。
- 筛选条件字符串的长度最多为 255 个字符。
- 您可以将通配符与筛选值一同使用。星号 (*) 匹配零个或多个字符，而问号 (?) 匹配零个或一个字符。
- 筛选值区分大小写。
- 您的搜索中可包含通配符的字面值；您只需要在字符前用反斜线隔开字符。例如，用 `\*amazon\?\\` 值搜索文字字符串 `*amazon?\`。
- 您不能将筛选条件值指定为 null，但可以使用客户端筛选。例如，以下命令使用 --query 选项，返回在无密钥对的情况下启动之实例的 ID。

```
aws ec2 describe-instances \
    --query 'Reservations[*].Instances[?!not_null(KeyName)].InstanceId' \
    --output text
```

Amazon CLI

Example示例：指定单个筛选条件

您可以使用 describe-instances 列出 Amazon EC2 实例。如果没有筛选条件，响应将包含您的所有资源的信息。您可以使用以下选项，在输出中仅包含正在运行的实例。

```
--filters Name=instance-state-name,Values=running
```

要仅列出正在运行的实例的实例 ID，请按如下所示添加 `--query` 选项。

```
--query "Reservations[*].Instances[*].InstanceId"
```

Example示例：指定多个筛选条件或筛选条件值

如果指定多个筛选条件或多个筛选条件值，则资源必须在与所有筛选条件匹配时才会包含在结果中。

您可以使用以下选项来列出类型为 `m5.large` 或 `m5d.large` 的所有实例。

```
--filters Name=instance-type,Values=m5.large,m5d.large
```

您可以使用以下选项来列出类型为 `t2.micro` 的所有已停止实例。

```
--filters Name=instance-state-name,Values=stopped Name=instance-type,Values=t2.micro
```

Example示例：在筛选条件值中使用通配符

您可以使用 describe-snapshots 和以下选项，从而仅返回描述为“database”的快照。

```
--filters Name=description,Values=database
```

* 通配符可与零个或多个字符匹配。您可以使用以下选项，仅返回描述中包含单词 database 的快照。

```
--filters Name=description,Values=*database*
```

? 通配符完全匹配 1 个字符。您可以使用以下选项，仅返回描述为“database”或“database”并后接一个字符的快照。

```
--filters Name=description,Values=database?
```

您可以使用以下选项，仅返回描述为“database”并后接不超过四个字符的快照。这会排除包含“database”并后接五个或以上字符的描述。

```
--filters Name=description,Values=database????
```

Example示例：基于日期进行筛选

使用 Amazon CLI，您可以使用 JMESPath 通过表达式来筛选结果。例如，以下 describe-snapshots 命令会显示指定 Amazon Web Services 账户在指定日期之前创建的所有快照的 ID。如果未指定所有者，则结果将包括所有公有快照。

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime<='2024-03-31')].[SnapshotId]" \
    --output text
```

以下示例会显示在指定日期范围内创建的所有快照的 ID。

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime>='2024-01-01') && (StartTime<='2024-12-31')].
[SnapshotId]" \
    --output text
```

示例：基于标签进行筛选

有关如何根据资源标签筛选资源列表的示例，请参阅按标签筛选 Amazon EC2 资源。

PowerShell

Example示例：指定单个筛选条件

您可以使用 Get-EC2Instance 列出 Amazon EC2 实例。如果没有筛选条件，响应将包含您的所有资源的信息。您可以使用以下参数，在输出中仅包含正在运行的实例。

```
-Filter @{Name="instance-state-name"; Values="running"}
```

以下示例仅会列出正在运行的实例的实例 ID。

```
(Get-EC2Instance -Filter @{Name="instance-state-name"; Values="stopped"}).Instances
 | Select InstanceId
```

Example示例：指定多个筛选条件或筛选条件值

如果指定多个筛选条件或多个筛选条件值，则资源必须与所有筛选条件匹配才能包括在结果中。

您可以使用以下选项来列出类型为 m5.large 或 m5d.large 的所有实例。

```
-Filter @{Name="instance-type"; Values="m5.large", "m5d.large"}
```

您可以使用以下选项来列出类型为 `t2.micro` 的所有已停止实例。

```
-Filter @{Name="instance-state-name"; Values="stopped"}, @{Name="instance-type";
 Values="t2.micro"}
```

使用 Amazon EC2 全局视图查看跨区域的资源

通过 Amazon EC2 全局视图，您可以在单个控制台中跨单个 Amazon 区域或跨多个区域同时查看和搜索 Amazon EC2 和 Amazon VPC 资源。有关更多信息，请参阅 使用 Amazon 全局视图查看跨区域的资源。

使用 Amazon 全局视图查看跨区域的资源

通过 Amazon 全局视图，您可以在单个控制台中跨单个 Amazon 区域或跨多个区域查看一些 Amazon EC2 和 Amazon VPC 资源。Amazon全局视图还提供全局搜索功能，让您能同时搜索多个区域中的特定资源或特定资源类型。

Amazon 全局视图不允许您以任何方式修改资源。

支持的资源

使用 Amazon 全局视图，您可以查看已启用 Amazon Web Services 账户 的所有区域中以下资源的全局摘要。

- 自动扩缩组
- 可用区
- 容量预留和容量块
- 数据库集群
- 数据库实例数
- DHCP 选项集
- 仅出口互联网网关
- 弹性 IP
- 端点服务

- 实例
- 互联网网关
- 托管前缀列表
- NAT 网关
- 网络 ACL
- 网络接口
- Outposts
- 路由表
- S3 存储桶
- 安全组
- 子网
- 卷
- VPC
- VPC 端点
- VPC 对等连接

所需的权限

用户必须具有以下权限才能使用 Amazon 全局视图。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "autoscaling:DescribeAutoScalingGroups",
      "ec2:DescribeRegions",
      "ec2:DescribeCapacityReservations",
      "ec2:DescribeDhcpOptions",
      "ec2:DescribeEgressOnlyInternetGateways",
      "ec2:DescribeAddresses",
      "ec2:DescribeVpcEndpointServices",
      "ec2:DescribeInstances",
```

```
        "ec2:DescribeInternetGateways",
        "ec2:DescribePrefixLists",
        "ec2:DescribeNatGateways",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeVpcEndpointServiceConfigurations",
        "ec2:DescribeManagedPrefixLists",
        "outposts:ListOutposts",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
    }]
}
```

使用 Amazon 全局视图

登录 Amazon 全局视图控制台。

> **Important**
>
> 您不能在 Firefox 中使用私密窗口来访问 Amazon 全局视图。

控制台由以下各项组成：

- 区域资源管理器：此页面包括以下各节：
  - 摘要 – 提供您在所有区域中的资源的简要概述。

    展开显示所有资源摘要以表示已启用 Amazon Web Services 账户的区域数。剩余字段表示您当前在这些区域中拥有的资源数量。选择任何链接以查看所有区域该类型的资源。例如，如果

Instances（实例）标签以下的链接是 29 in 10 Regions（10 个区域中的 29 个），它表示您当前在 10 个区域有 29 个实例。选择该链接以查看所有 29 个实例的列表。

- 区域资源管理器：列出所有 Amazon Web Services 区域（包括没有为其启用账户的那些区域），并提供每个区域的每种资源类型的总计。

  选择区域名称，以查看该特定区域所有类型的所有资源。例如，选择 Africa (Cape Town) af-south-1（非洲（开普敦）af-south-1）可查看该区域中的所有 VPC、子网、实例、安全组、卷和自动扩缩组。或者，选择一个区域，然后选择 View resources for selected Region（查看所选区域的资源）。

  选择特定区域中特定资源类型的值，以便仅查看该区域中该类型的资源。例如，选择 Africa (Cape Town) af-south-1（非洲（开普敦）af-south-1）实例的值，以仅查看该区域中的实例。

- 全局搜索 – 在此页面上，您可以跨单个区域或跨多个区域搜索特定资源或特定资源类型。它还使您能够查看特定资源的详细信息。

  要搜索资源，请在网格前面的字段中输入搜索条件。您可以按区域、按资源类型以及按分配给资源的标签进行搜索。

  要查看特定资源的详细信息，请在网格中选中。您也可以选择资源的资源 ID，以在各自的控制台中打开该资源。例如，选择实例 ID 以在 Amazon EC2 控制台中打开实例，或选择子网 ID 以在 Amazon VPC 控制台中打开子网。

- 区域和可用区 - 在此页面上，您可以查看和管理所有可用区域、可用区、本地区域和 Wavelength 区域。

  在区域选项卡中，可以查看所有 Amazon 区域。状态列显示为您的 Amazon Web Services 账户启用的区域。在此页面中，您可以选择一个区域来执行以下操作：

  - 查看区域的详细信息，例如区域代码、地理位置和每种区域的数量。

    您还可以查看可用区、本地区域和 Wavelength 区域的列表以及区域中的资源列表。

  - 启用或禁用区域。

  从每个区域选项卡中，您可以查看该区域类型的列表。在本地区域选项卡中，您可以选择加入某个本地区域。

> **Tip**
>
> 如果您仅使用特定区域或资源类型，则可以自定义 Amazon 全局视图，使其仅显示这些区域和资源类型。要自定义显示的区域和资源类型，请在导航面板中选择设置，然后在资源和区域选项卡上，选择不希望在 Amazon 全局视图中显示的区域和资源类型。

标记 Amazon EC2 资源

为了方便您管理实例、映像以及其他 Amazon EC2 资源，您可通过标签 的形式为每个资源分配元数据。标签可让您按各种标准（例如用途、所有者或环境）对 Amazon 资源进行分类。这在您具有相同类型的很多资源时会很有用 – 您可以根据分配给特定资源的标签快速识别该资源。本主题介绍标签并说明如何创建标签。

> **Warning**
>
> 很多不同的 API 调用返回标签键及其值。拒绝访问 `DescribeTags` 不会自动拒绝访问其他 API 返回的标签。作为最佳实践，我们建议您不要在标签中包含敏感数据。

目录

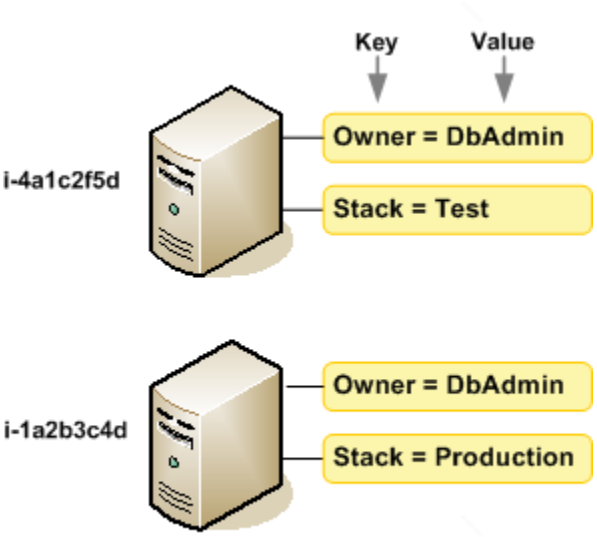
- 有关标签的基本知识
- 标记 资源
- 标签限制
- 标签和访问管理
- 为资源添加标签以便于进行计费
- 在创建过程中授予标记 Amazon EC2 资源的权限
- 添加或删除 Amazon EC2 资源的标签
- 按标签筛选 Amazon EC2 资源
- 使用实例元数据来查看 EC2 实例的标签

有关标签的基本知识

标签是为Amazon资源分配的标记。每个标签都包含您定义的一个键 和一个可选值。

标签可让您按各种标准（例如用途、所有者或环境）对 Amazon 资源进行分类。例如，您可以为账户中的 Amazon EC2 实例定义一组标签，以跟踪每个实例的所有者和堆栈级别。

下图说明了标签的工作方式。在此示例中，您为每个实例分配了两个标签 — 一个标签使用键 `Owner`，另一个使用键 `Stack`。每个标签都拥有相关的值。



我们建议您针对每类资源设计一组标签，以满足您的需要。使用一组连续的标签键，管理 资源时会更加轻松。您可以根据添加的标签搜索和筛选资源。有关如何实施有效资源标记策略的更多信息，请参阅标记最佳实践 Amazon 白皮书。

标签对 Amazon EC2 没有任何语义意义，应严格按字符串进行解析。同时，标签不会自动分配至您的资源。您可以修改标签的密钥和值，还可以随时删除资源的标签。您可以将标签的值设为空的字符串，但是不能将其设为空值。如果您添加的标签的值与该实例上现有标签的值相同，新的值就会覆盖旧值。如果删除资源，资源的所有标签也会被删除。

> **Note**
>
> 删除资源后，其标签可能会在短时间内在控制台、API 和 CLI 输出中仍保持可见。这些标签将逐步解除与资源的关联，并永久删除。

标记 资源

当您使用 Amazon EC2 控制台时，可以使用相关资源屏幕上的标签选项卡或 Amazon Resource Groups 控制台中的标签编辑器，将标签应用到资源。在您创建资源时，某些资源屏幕能让您为资源指

定标签；例如，具有 Name 键并且具有您指定的值的标签。在大多数情况下，控制台会在资源创建后 (而不是在资源创建期间) 立即应用标签。控制台可能根据 Name 标签对资源进行组织，但此标签对于 Amazon EC2 服务没有任何语义意义。

如果使用的是 Amazon EC2 API、Amazon CLI 或 Amazon 软件开发工具包，则您可以使用 CreateTags EC2 API 操作向现有资源应用标签。此外，某些资源创建操作允许您在创建资源时为其指定标签。如果无法在资源创建期间应用标签，系统会回滚资源创建过程。这样可确保要么创建带有标签的资源，要么根本不创建资源，即任何时候都不会创建出未标记的资源。通过在创建时标记资源，您不需要在资源创建后运行自定义标记脚本。有关允许用户在创建时标记资源的更多信息，请参阅 在创建过程中授予标记 Amazon EC2 资源的权限。

对于支持在创建时进行标记的 Amazon EC2 API 操作，您可以在 IAM policies 中应用基于标签的资源级权限，以对可在创建时标记资源的用户和组实施精细控制。您的资源从创建开始会受到适当的保护 — 标签会立即用于您的资源，因此控制资源使用的任何基于标签的资源级权限都会立即生效。可以更准确地对您的资源进行跟踪和报告。您可以强制对新资源使用标记，可以控制对资源设置哪些标签键和值。

此外，您还可以在 IAM policies 中对 CreateTags 和 DeleteTags Amazon EC2 API 操作应用资源级权限，从而控制对现有资源设置哪些标签键和值。有关更多信息，请参阅示例：标记资源。

有关标记资源以便于计费的更多信息，请参阅 Amazon Billing 用户指南中的使用成本分配标签。

标签限制

下面是适用于标签的基本限制：

- 每个资源的标签数上限：50
- 对于每个资源，每个标签键都必须是唯一的，每个标签键只能有一个值。
- 最大键长度：128 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度：256 个 Unicode 字符 (采用 UTF-8 格式)
- 允许使用的字符
  - 虽然 EC2 允许在其标签中使用任何字符，但其他 Amazon 服务具有更严格的限制。允许在所有 Amazon 服务中使用的字符包括：可以使用 UTF-8 表示的字母（a-z、A-Z）、数字（0-9）和空格以及以下字符：+ - = . _ : / @。
  - 如果在实例元数据中启用实例标签，实例标签键只能使用字母 (a-z、A-Z)、数字 (0-9) 及以下字符：+ - = . , _ : @。实例标签 keys 不能包含空格或 /，也不能只包含 .（一个句号）、..（两个句号）或 _index。有关更多信息，请参阅 使用实例元数据来查看 EC2 实例的标签。

- 标签键和值区分大小写。
- aws: 前缀专门预留供 Amazon 使用。如果某个标签具有带有此标签键，则您无法编辑该标签的键或值。具有 aws: 前缀的标签不计入每个资源的标签数限制。

您不能仅依据标签终止或删除资源，而必须指定资源的标识符。例如，要删除您使用名为 DeleteMe 的标签键标记的快照，您必须将 DeleteSnapshots 操作与快照的资源标识符 (如 snap-1234567890abcdef0) 结合使用。

当您为公有或共享资源添加标签时，您分配的标签仅对您的 Amazon 账户可用；其他 Amazon 账户无权访问这些标签。为了对共享资源进行基于标签的访问控制，每个 Amazon 账户必须分配自己的一组标签来控制对资源的访问。

标签和访问管理

如果您使用的是 Amazon Identity and Access Management (IAM)，则可以控制Amazon账户中的哪些用户拥有创建、编辑或删除标签的权限。有关更多信息，请参阅在创建过程中授予标记 Amazon EC2 资源的权限。

您还可以使用资源标签来实现基于属性的控制 (ABAC)。您可以创建 IAM policies，基于资源标签允许操作。有关更多信息，请参阅使用基于属性的访问控制访问权限。

为资源添加标签以便于进行计费

您也可以使用标签来组织您的 Amazon 账单，使其反映您的成本结构。要执行此操作，请注册以获取包含标签键值的 Amazon 账户账单。有关设置带有标签的成本分配报告的更多信息，请参阅 Amazon Billing 用户指南中的月度成本分配报告。如需查看组合资源的成本，请按具有相同标签键值的资源组织您的账单信息。例如，您可以将特定的应用程序名称用作几个资源的标签，然后组织账单信息，以查看在数个服务中的使用该应用程序的总成本。有关更多信息，请参阅 Amazon Billing 用户指南中的使用成本分配标签。

> **Note**
>
> 如果您已启用报告，则可以在 24 小时后查看当月的数据。

成本分配标签可指示哪些资源导致产生成本，而删除或停用资源并不总是能降低成本。例如，即使删除包含原始数据的快照，其他快照引用的快照数据也将保留。有关更多信息，请参阅 Amazon Billing 用户指南中的 Amazon Elastic Block Store 卷和快照。

> **Note**
>
> 标记的弹性 IP 地址不会显示在成本分配报告中。

在创建过程中授予标记 Amazon EC2 资源的权限

某些资源创建 Amazon EC2 API 操作允许您在创建资源时指定标签。您可以使用资源标签来实现基于属性的控制 (ABAC)。有关更多信息，请参阅标记 资源和使用基于属性的访问控制访问权限。

为使用户能够在创建时为资源添加标签，他们必须具有使用创建该资源的操作（如 `ec2:RunInstances` 或 `ec2:CreateVolume`）的权限。如果在资源创建操作中指定了标签，则 Amazon 会对 `ec2:CreateTags` 操作执行额外的授权，以验证用户是否具备创建标签的权限。因此，用户还必须具有使用 `ec2:CreateTags` 操作的显式权限。

在 `ec2:CreateTags` 操作的 IAM 策略定义中，使用带有 `Condition` 条件键的 `ec2:CreateAction` 元素，为创建资源的操作授予添加标签的权限。

例如，下面的策略允许用户启动实例并在启动期间向实例和卷应用任何标签。用户无权标记任何现有资源 (他们无法直接调用 `ec2:CreateTags` 操作)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "RunInstances"
          }
       }
```

```
    }
  ]
}
```

同样，下面的策略允许用户创建卷并在创建卷期间向卷应用任何标签。用户无权标记任何现有资源 (他们无法直接调用 ec2:CreateTags 操作)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateVolume"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:111122223333:*/*",
      "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "CreateVolume"
          }
       }
    }
  ]
}
```

仅当用户在资源创建操作中应用了标签时，系统才会评估 ec2:CreateTags 操作。因此，如果未在此请求中指定任何标签，则拥有创建资源权限（假定没有标记条件）的用户无需具备使用 ec2:CreateTags 操作的权限。但是，如果用户不具备使用 ec2:CreateTags 操作的权限而又试图创建带标签的资源，则请求将失败。

如果在启动模板中提供了标签，也会对 ec2:CreateTags 操作进行评估。有关策略示例，请参阅启动模板中的标签。

控制对特定标签的访问

您可以在 IAM 策略的 Condition 元素中使用其他条件来控制可应用到资源的标签键和值。

以下条件键可用于上一节中的示例：

- aws:RequestTag：指示请求中必须存在特定的标签键或标签键和值。也可在此请求中指定其他标签。
  - 与 StringEquals 条件运算符配合使用，以强制实施特定的标签键和值组合，例如强制实施标签 cost-center=cc123：

    ```
    "StringEquals": { "aws:RequestTag/cost-center": "cc123" }
    ```

  - 与 StringLike 条件运算符配合使用，以在请求中强制实施特定的标签键；如强制实施标签键 purpose：

    ```
    "StringLike": { "aws:RequestTag/purpose": "*" }
    ```

- aws:TagKeys：强制实施在请求中使用的标签键。
  - 与 ForAllValues 修饰符配合使用，以只强制实施请求中提供的特定标签键（如果在请求中指定了标签，则只允许特定的标签键；不允许任何其他标签）。例如，允许标签键 environment 或 cost-center：

    ```
    "ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }
    ```

  - 与 ForAnyValue 修饰符配合使用，以强制请求中至少存在一个指定的标签键。例如，强制请求中至少存在标签键 environment 或 webserver 中的一个：

    ```
    "ForAnyValue:StringEquals": { "aws:TagKeys": ["environment","webserver"] }
    ```

上述条件键可应用于支持标记的资源创建操作，以及 ec2:CreateTags 和 ec2:DeleteTags 操作。要了解 Amazon EC2 API 操作是否支持添加标签，请参阅 Amazon EC2 的操作、资源和条件建。

为强制用户指定标签，在创建资源时，您必须使用 aws:RequestTag 条件键或 aws:TagKeys 条件键，并在资源创建操作中使用修饰符 ForAnyValue。如果用户没有为资源创建操作指定标签，则不会对 ec2:CreateTags 操作进行评估。

对于条件，条件键不区分大小写，条件值区分大小写。因此，要强制标签键区分大小写，请使用 aws:TagKeys 条件键，其中标签键指定为条件中的值。

有关示例 IAM policies，请参阅 用于控制访问 Amazon EC2 API 的示例策略。有关更多信息，请参阅《IAM 用户指南》中的具有多个上下文键或值的条件。

添加或删除 Amazon EC2 资源的标签

在创建 Amazon EC2 资源（例如 Amazon EC2 实例）时，可以指定要添加到资源的标签。Amazon EC2 控制台可用于显示特定 Amazon EC2 资源的标签。您也可以在现有的 Amazon EC2 资源中添加或删除标签。

使用 Amazon Resource Groups 控制台中的标签编辑器，可以查看、添加或删除所有区域中所有 Amazon 资源的标签。您可以同时将标签应用到多个类型的资源，也可以同时从多个类型的资源删除标签。有关更多信息，请参阅 标记 Amazon 资源用户指南。

任务

- 使用控制台添加标签
- 使用 Amazon CLI 添加标签
- 使用 PowerShell 添加标签
- 使用 CloudFormation 添加标签

使用控制台添加标签

您可以从资源页面直接为现有的资源添加标签。

向现有资源添加标签

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航栏中选择资源所在的区域。
3. 在导航窗格中，选择资源类型 (例如，Instances)。
4. 从列表中选择相应资源。
5. 从标签选项卡中选择管理标签。
6. 选择添加新标签，然后输入标签键和标签值。
7. 选择保存。

使用 Amazon CLI 添加标签

您可以在创建资源时添加标签，也可以向现有资源添加标签。

在创建资源时添加标签

使用 `-tag-specifications` 选项在创建资源时标记资源。标签规范规定了要标记的资源类型、标签键和标签值。以下示例会创建一个标签并将其添加到标签规范中。

```
--tag-specifications 'ResourceType=instance,Tags=[{Key=stack,Value=production}]'
```

向现有资源添加标签

以下示例演示了如何使用 create-tags 命令向现有资源添加标签。

Example示例：将标签添加到资源

以下示例将标签 **Stack=production** 添加到指定的映像，或者覆盖该 AMI 的现有标签（其中标签键为 stack）。如果命令成功，则不返回任何输出。

```
aws ec2 create-tags \
    --resources ami-0abcdef1234567890 \
    --tags Key=stack,Value=production
```

Example示例：将标签添加到多个资源

此示例为 AMI 和实例添加（或覆盖）两个标签。其中一个标签仅包含键 (webserver)，不包含值（我们将值设置为空字符串）。另一个标签则包含键 (stack) 和值 (**Production**)。如果命令成功，则不返回任何输出。

```
aws ec2 create-tags \
    --resources ami-0abcdef1234567890 i-1234567890abcdef0 \
    --tags Key=webserver,Value=  Key=stack,Value=Production
```

Example示例：使用特殊字符添加标签

此示例将标签 [Group]=test 添加到实例。方括号（[和]）是特殊字符，必须对其进行转义。

如果您使用的是 Linux 或 OS X，要转义特殊字符，请用双引号 (") 将具有特殊字符的元素引起来，然后用单引号 (') 将整个键和值结构引起来。

```
aws ec2 create-tags \
    --resources i-1234567890abcdef0 \
    --tags 'Key="[Group]",Value=test'
```

如果您使用的是 Windows，要转义特殊字符，请用双引号 (") 将具有特殊字符的元素引起来，然后在每个双引号字符前面添加反斜杠 (****)，如下所示：

```
aws ec2 create-tags ^
    --resources i-1234567890abcdef0 ^
    --tags Key=\"[Group]\",Value=test
```

如果您使用的是 Windows PowerShell，要转义特殊字符，请用双引号 (**"**) 将具有特殊字符的值引起来，在每个双引号字符前面添加反斜杠 (****)，然后用单引号 (**'**) 将整个键和值结构引起来，如下所示：

```
aws ec2 create-tags `
    --resources i-1234567890abcdef0 `
    --tags 'Key=\"[Group]\",Value=test'
```

使用 PowerShell 添加标签

您可以在创建资源时添加标签，也可以向现有资源添加标签。

在创建资源时添加标签

使用 `-TagSpecification` 参数在创建资源时标记资源。标签规范规定了要标记的资源类型、标签键和标签值。以下示例会创建一个标签并将其添加到标签规范中。

```
$tag = @{Key="stack"; Value="production"}
$tagspec = new-object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance"
$tagspec.Tags.Add($tag)
```

以下示例会在 `-TagSpecification` 参数中指定此标签。

```
-TagSpecification $tagspec
```

向现有资源添加标签

使用 New-EC2Tag cmdlet。您必须指定资源、标签键和标签值。

```
New-EC2Tag `
    -Resource i-1234567890abcdef0 `
    -Tag @{Key="purpose"; Value="production"}
```

使用 CloudFormation 添加标签

对于 Amazon EC2 资源类型，您可以使用 `Tags` 或 `TagSpecifications` 属性指定标签。

以下示例使用其 `Tags` 属性将标签 **`Stack=Production`** 添加到 AWS::EC2::Instance。

Example示例：YAML 中的 Tags

```
Tags:
  - Key: "Stack"
    Value: "Production"
```

Example示例：JSON 中的 Tags

```
"Tags": [
    {
        "Key": "Stack",
        "Value": "Production"
    }
]
```

以下示例使用其 `TagSpecifications` 属性将标签 **`Stack=Production`** 添加到 AWS::EC2::LaunchTemplate LaunchTemplateData。

Example示例：YAML 中的 TagSpecifications

```
TagSpecifications:
  - ResourceType: "instance"
    Tags:
    - Key: "Stack"
      Value: "Production"
```

Example示例：JSON 中的 TagSpecifications

```
"TagSpecifications": [
    {
        "ResourceType": "instance",
        "Tags": [
            {
                "Key": "Stack",
                "Value": "Production"
            }
        ]
    }
]
```

按标签筛选 Amazon EC2 资源

添加标签后，即可基于标签键和标签值来筛选 Amazon EC2 资源。

Console

按标签筛选资源

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择资源类型 (例如，Instances)。
3. 选择搜索字段。
4. 在列表中的标签下，选择标签键。
5. 从列表中选择相应的标签值。
6. 完成后，删除筛选条件。

有关在 Amazon EC2 控制台中使用筛选条件的更多信息，请参阅 查找 Amazon EC2 资源。

Amazon CLI

描述具有指定标签键的单个类型的资源

将以下筛选条件添加到 `describe` 命令中，来描述具有 Stack 标签的该类型资源，无论标签值为何。

```
--filters Name=tag-key,Values=Stack
```

描述具有指定标签的单个类型的资源

将以下筛选条件添加到 `describe` 命令中，来描述具有标签 Stack=production 的该类型资源。

```
--filters Name=tag:Stack,Values=production
```

描述具有指定标签值的单个类型的资源

将以下筛选条件添加到 `describe` 命令中，来描述具有值为 production 的标签的该类型资源，无论标签键为何。

```
--filters Name=tag-value,Values=production
```

描述具有指定标签的所有 EC2 资源

将以下筛选条件添加到 describe-tags 命令中，来描述具有标签 Stack=test 的所有 EC2 资源。

```
--filters Name=key,Values=Stack Name=value,Values=test
```

PowerShell

按标签键筛选单个类型的资源

将以下筛选条件添加到 Get cmdlet 中，来描述具有 Stack 标签的该类型资源，无论标签值为何。

```
-Filter @{Name="tag-key"; Values="Stack"}
```

按标签筛选单个类型的资源

将以下筛选条件添加到 Get cmdlet 中，来描述具有标签 Stack=production 的该类型资源。

```
-Filter @{Name="tag:Stack"; Values="production"}
```

按标签值筛选单个类型的资源

将以下筛选条件添加到 Get 命令中，来描述具有值为 production 的标签的该类型资源，无论标签键的值为何。

```
-Filter @{Name="tag-value"; Values="production"}
```

按标签筛选所有 EC2 资源

将以下筛选条件添加到 Get-EC2Tag cmdlet 中，来描述具有标签 Stack=test 的所有 EC2 资源。

```
-Filter @{Name="tag:Stack"; Values="test"}
```

使用实例元数据来查看 EC2 实例的标签

您可以从实例元数据访问实例的标签。通过从实例元数据访问标签，您无需再使用 DescribeInstances 或 DescribeTags API 调用来检索标签信息，这可以减少每秒的 API 事务

量，并允许您的标签检索随您控制的实例数量扩展。此外，在实例上运行的本地进程可以直接从实例元数据中查看实例的标签信息。

默认情况下，实例元数据中不提供标签；您必须明确允许访问。您可以在实例启动时或启动后在正在运行或停止的实例上允许访问。您还可以通过在启动模板中指定标签来允许访问标签。使用模板启动的实例允许访问实例元数据中的标签。

如果您添加或移除实例标签，则实例元数据将在实例运行期间更新，无需停止后再启动实例。

任务

- 启用对实例元数据中标签的访问权限
- 从实例元数据中检索标签
- 禁用对实例元数据中标签的访问权限

启用对实例元数据中标签的访问权限

默认情况下，无法访问实例元数据中的实例标签。对于每个实例，您必须显式启用访问权限。

> **Note**
>
> 如果您允许访问实例元数据中的标签，则实例标签键存在特定的限制。不合规将导致新实例启动失败或现有实例出错。这些限制包括：
>
> - 只能包含字母（a-z、A-Z）、数字 (0-9) 和以下字符：+ - = . , _ : @。
> - 不能包含空格或 /。
> - 不能只包含 .（一个句点）、..（两个句点）或 _index。
>
> 有关更多信息，请参阅 标签限制。

Console

在实例启动期间启用对实例元数据中标签的访问权限

1. 按照程序启动实例。
2. 展开高级详细信息，对于允许元数据中的标签，选择启用。
3. 在 Summary（摘要）面板中查看实例配置，然后选择 Launch instance（启动实例）。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。

在实例启动后启用对实例元数据中标签的访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择操作、实例设置、允许实例元数据中的标签。
4. 要允许访问实例元数据中的标签，请选择允许复选框。
5. 选择保存。

Amazon CLI

在实例启动期间启用对实例元数据中标签的访问权限

使用 run-instances 命令并添加以下 `--metadata-options` 选项。

```
--metadata-options "InstanceMetadataTags=enabled"
```

在实例启动后启用对实例元数据中标签的访问权限

使用以下 modify-instance-metadata-options 命令。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags enabled
```

验证对实例元数据中标签的访问权限已启用

使用 describe-instances 命令并检查 `InstanceMetadataTags` 的值。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query "Reservations[*].Instances[].MetadataOptions[].InstanceMetadataTags"
```

下面是示例输出。该值为 `enabled` 或 `disabled`。

```
[
    "enabled"
]
```

PowerShell

在实例启动期间启用对实例元数据中标签的访问权限

使用 New-EC2Instance cmdlet 并添加以下 `-MetadataOptions_InstanceMetadataTags` 参数。

```
-MetadataOptions_InstanceMetadataTags enabled
```

在实例启动后启用对实例元数据中标签的访问权限

使用 Edit-EC2InstanceMetadataOption cmdlet。

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTags enabled
```

验证对实例元数据中标签的访问权限已启用

使用 Get-EC2Instance cmdlet 并检查 `InstanceMetadataTags` 的值。

```
(Get-EC2Instance `
    -
InstanceId i-1234567890abcdef0).Instances.MetadataOptions.InstanceMetadataTags.Value
```

下面是示例输出。该值为 `enabled` 或 `disabled`。

```
enabled
```

从实例元数据中检索标签

在允许访问实例元数据中的实例标签后，您可以从实例元数据访问 `tags/instance` 类别。有关更多信息，请参阅 访问 EC2 实例的实例元数据。

IMDSv2

Linux

在 Linux 实例中运行以下命令，列出该实例的所有标签键。

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance
```

此示例会获取上一示例中获取的键的值。IMDSv2 请求使用在前面的示例中使用命令中创建的存储令牌。令牌不得过期。

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

在 Windows 实例中运行以下 cmdlet，列出该实例的所有标签键。

```
$token = Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" = "21600"} `
    -Method PUT -Uri http://169.254.169.254/latest/api/token
```

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

此示例会获取上一示例中获取的键的值。IMDSv2 请求使用在前面的示例中使用命令中创建的存储令牌。令牌不得过期。

```
Invoke-RestMethod `
    -Headers @{"X-aws-ec2-metadata-token" = $token} `
    -Method GET -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

IMDSv1

Linux

在 Linux 实例中运行以下命令，列出该实例的所有标签键。

```
curl http://169.254.169.254/latest/meta-data/tags/instance
```

此示例会获取上一示例中获取的键的值。

```
curl http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

Windows

在 Windows 实例中运行以下 cmdlet，列出该实例的所有标签键。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance
```

此示例会获取上一示例中获取的键的值。

```
Invoke-RestMethod -Uri http://169.254.169.254/latest/meta-data/tags/instance/tag-key
```

禁用对实例元数据中标签的访问权限

您可以禁用对实例元数据中实例标签的访问权限。您无需在启动时禁用对实例元数据中实例标签的访问权限，因此在默认情况下已禁用此权限。

Console

禁用对实例元数据中标签的访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，然后依次选择 Actions（操作）、Instance settings（实例设置）、Allow tags in instance metadata（允许实例元数据中的标签）。
4. 要关闭对实例元数据中的标签的访问，请清除允许复选框。
5. 选择保存。

Amazon CLI

禁用对实例元数据中标签的访问权限

使用以下 modify-instance-metadata-options 命令。

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567890abcdef0 \
    --instance-metadata-tags disabled
```

PowerShell

禁用对实例元数据中标签的访问权限

使用 Edit-EC2InstanceMetadataOption cmdlet。

```
Edit-EC2InstanceMetadataOption `
    -InstanceId i-1234567890abcdef0 `
    -InstanceMetadataTag disabled
```

Amazon EC2 Service Quotas

在创建 Amazon Web Services 账户 时，我们会根据区域设置 Amazon 资源的默认限额（也称为限制）。如果超出资源的限额，请求会失败。例如，在某一区域中，可为按需型实例预置的 Amazon EC2 vCPU 数量存在最大数量限制。如果尝试在某一区域中启动实例，但此请求会导致使用量超出限额，则请求会失败。如果发生这种情况，可以减少资源使用量或请求增加限额。

您可以通过服务配额控制台集中查看和管理 Amazon 服务配额，以及请求提高所使用各种资源的限额。使用我们提供的限制信息来管理您的 Amazon 基础设施。请根据需要请提前计划以请求提高限额。

查看当前限额

您可以使用 服务配额控制台来查看您在每个区域的配额。

使用服务配额控制台查看当前配额

1. 访问 https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/，打开服务配额控制台。
2. 在导航栏（位于屏幕顶部）中，选择一个区域。
3. 在管理配额窗格上，选择一项服务；例如，Amazon Elastic Compute Cloud（Amazon EC2）。选择查看配额。
4. 使用筛选字段按资源名称筛选列表。例如，输入 **On-Demand** 即可查找按需型实例的限额。
5. 要查看更多信息，请选择限额名称以打开限额的详细信息页面。

请求提高

您可以请求为每个区域提高限额。

使用服务配额控制台请求提高限制

1. 访问 https://console.aws.amazon.com/servicequotas/home/services/ec2/quotas/，打开服务配额控制台。
2. 在导航栏（位于屏幕顶部）中，选择一个区域。
3. 使用筛选字段按资源名称筛选列表。例如，输入 **On-Demand** 即可查找按需型实例的限额。
4. 如果限额可调，则选中该限额并选择请求提高限额。
5. 对于更改限额值，请输入新的限额值。
6. 选择请求。
7. 要在控制台中查看任何待处理或最近已解决的请求，请从导航窗格中选择控制面板。对于待处理的请求，请选择请求状态以打开收到的请求。请求的初始状态为 Pending。状态更改为已请求限额后，您将在 Amazon Web Services 支持 中看到工单编号。选择案例编号以打开请求服务单。

有关更多信息，包括如何使用 Amazon CLI 或 SDK 请求提高配额的信息，请参阅《服务配额用户指南》中的请求增加配额。

对使用端口 25 发送的电子邮件的限制

默认情况下，Amazon EC2 仅允许通过端口 25 的出站流量指向私有 IPv4 地址。通过端口 25 指向公有 IPv4 地址和 IPv6 地址的流量会被阻止。

您可以请求删除此限制。有关更多信息，请参阅如何删除 Amazon EC2 实例或 Lambda 函数的端口 25 的限制？

此限制不适用于通过端口 25 指向下列地址的出站流量：

- 存在原始网络接口的 VPC 主 CIDR 数据块中的 IP 地址。
- RFC 1918、RFC 6598 和 RFC 4193 中定义的 CIDR 中的 IP 地址。

监控 Amazon EC2 资源

监控是保持 Amazon EC2 实例和 Amazon 解决方案的可靠性、可用性以及性能的重要方面。您的 Amazon 解决方案的所有组成部分都应收集监控数据，以便更轻松地调试出现的多点故障。

Amazon提供各种可以用来监控 Amazon EC2 的工具。Amazon EC2 和 CloudWatch 控制台控制面板提供您的 Amazon EC2 环境状态的概览视图。此外，我们提供以下各项：

- 系统状态检查 – 监控使用您的实例所需的 Amazon 系统，以确保这些系统正常工作。这些检查会检测出需要 Amazon 参与修复的实例问题。当一个系统状态检查故障时，您可以等待 Amazon 修复故障，或者您也可以亲自解决该故障（例如，通过停止和重启或终止和替换实例）。导致系统状态检查出现故障的问题示例包括：
  - 网络连接丢失
  - 系统电源损耗
  - 物理主机上的软件问题
  - 物理主机上影响到网络连接状态的硬件问题

  有关更多信息，请参阅Amazon EC2 实例的状态检查。
- 实例状态检查 – 监控单个实例的软件和网络配置。这些检查检测需要您参与修复的问题。一旦发生实例状态检查故障，一般需要都您亲自解决这些问题（例如，通过重启实例或者在您的操作系统中进行修改）。可能导致实例状态检查出现故障的问题示例包括：
  - 系统状态检查故障
  - 网络或启动配置错误
  - 内存耗尽
  - 文件系统损坏
  - 内核不兼容

  有关更多信息，请参阅Amazon EC2 实例的状态检查。
- Amazon CloudWatch 警报 – 在您指定的时间段内监控单个指标，并根据指标值在一些时间段内与给定阈值的对比情况执行一个或多个操作。操作是一个发送到 Amazon Simple Notification Service (Amazon SNS) 主题或 Amazon EC2 Auto Scaling 策略的通知。警报只会调用操作进行持续的状态变更。CloudWatch 警报将不会调用操作，因为这些操作处于特定状态，该状态必须改变并在指定数量的时间段内一直保持。有关更多信息，请参阅 使用 CloudWatch 监控您的实例。

- Amazon EventBridge 事件 – 自动执行 Amazon 服务并自动响应系统事件。Amazon服务中的事件将近实时传输到 EventBridge，并且您可以指定要在事件匹配您编写的规则时执行的自动操作。有关更多信息，请参阅 the section called “使用 EventBridge 自动执行”。
- Amazon CloudTrail 日志 – 捕获有关向 Amazon EC2 API 发出的调用的详细信息，并将其以日志文件的形式存储在 Amazon S3 中。您可以使用 CloudTrail 日志确定已发出的调用、调用的源 IP 地址、调用的发出方、调用的发出时间。有关更多信息，请参阅 the section called “使用 CloudTrail 记录 API 调用”。
- CloudWatch 代理 – 从 EC2 实例和本地服务器上的主机及访客中收集日志和系统级指标。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的使用 CloudWatch 代理从 Amazon EC2 实例和本地服务器中收集指标和日志。

监控 Amazon EC2 实例的状态

您可以通过查看实例的状态检查和计划事件来监控您的实例状态。

状态检查反映 Amazon EC2 自动检查的结果信息。这些自动检查会检测出指定的问题是否影响您的实例。该状态检查信息与 Amazon CloudWatch 提供的数据一起为您的每一个实例提供详细的操作可视性。

您还可以查看为实例计划的特定事件的状态。事件状态提供了有关为实例计划的未来各项活动的信息，例如重启或停用。它们还提供了各个事件的计划开始时间和结束时间。

内容

- Amazon EC2 实例的状态检查
- Amazon EC2 实例的状态更改事件
- Amazon EC2 实例的计划事件

Amazon EC2 实例的状态检查

使用实例状态监控，您可以快速确定 Amazon EC2 是否已经检测到可能阻止您的实例运行应用程序的任何问题。Amazon EC2 会对每个运行的 EC2 实例执行自动检查以识别硬件和软件问题。您可以通过查看这些状态检查的结果来识别特定的和可检测的问题。事件状态数据扩充了 Amazon EC2 已提供的有关每个实例状态（如 `pending`、`running`、`stopping`）的信息以及 Amazon CloudWatch 监控的利用率指标（CPU 利用率、网络流量和磁盘活动）。

状态检查每分钟进行一次，会返回一个通过或失败状态。如果所有的检查都通过，则实例的整体状态是OK。如果有一个或多个检查故障，则整体状态为受损。状态检查是内置到 Amazon EC2 中的，所以不能禁用或删除。

当状态检查失败时，状态检查的相应 CloudWatch 指标将增加。有关更多信息，请参阅状态检查指标。您可以使用这些指标创建基于状态检查结果触发的 CloudWatch 警报。例如，您可以创建一个警报来提醒您在一个指定实例上的状态检查中返回了故障状态。有关更多信息，请参阅为未能通过状态检查的 Amazon EC2 实例创建 CloudWatch 警报。

您也可以创建 Amazon CloudWatch 警报，用于监控 Amazon EC2 实例并在实例由于潜在问题而受损时自动恢复实例。有关更多信息，请参阅实例自动恢复。

目录

- 状态检查类型
- 查看 Amazon EC2 实例的状态检查
- 为未能通过状态检查的 Amazon EC2 实例创建 CloudWatch 警报

状态检查类型

状态检查有三种类型。

- 系统状态检查
- 实例状态检查
- 附加的 EBS 状态检查

系统状态检查

系统状态检查监控您的实例在其上运行的 Amazon 系统。这些检查会检测出需要 Amazon 参与修复的深层实例问题。如果系统状态检查失败，您可以选择等待 Amazon 修复问题，也可以自行解决问题。对于由 Amazon EBS 支持的实例，您也可自行停止和启动实例，在大多数情况下，这会导致实例被迁移至新的主机。对于由实例存储支持的实例（仅 Linux 实例支持），您可以终止并替换实例。请注意，实例存储卷是临时的，实例停止后所有数据都会丢失。

以下是可能导致系统状态检查失败的问题的示例：

- 网络连接丢失
- 系统电源损耗

- 物理主机上的软件问题
- 物理主机上影响到网络连接状态的硬件问题

如果未通过系统状态检查，我们会微增 StatusCheckFailed_System 指标。

裸机实例

如果在裸机实例上重新启动，系统状态检查可能会暂时返回失败状态。当实例变为可用时，系统状态检查应返回通过状态。

实例状态检查

实例状态检查会监控各个实例的软件和网络连接情况。Amazon EC2 通过向网络接口 (NIC) 发送地址解析协议 (ARP) 请求，检查实例的运行状况。这些检查检测需要您参与修复的问题。如果实例状态检查失败，通常必须由您自行解决问题（例如，重启实例或更改实例配置）。

> **Note**
>
> 使用 `systemd-networkd` 进行网络配置的最新 Linux 发行版在报告运行状况检查时可能与早期发行版不同。在启动过程中，这种类型的网络可以更早地启动，并可能在其他启动任务之前完成，这些任务也可能会影响实例运行状况。依赖于网络可用性的状态检查可以在其他任务完成之前报告运行状况。

以下是可能导致实例状态检查失败的问题的示例：

- 系统状态检查故障
- 网络或启动配置不正确
- 内存耗尽
- 文件系统损坏
- 内核不兼容
- 在重启期间，在实例变得再次可用之前，实例状态检查都会报告失败。

如果未通过实例状态检查，我们会微增 StatusCheckFailed_Instance 指标。

裸机实例

如果在裸机实例上重新启动，实例状态检查可能会暂时返回失败状态。当实例变为可用时，实例状态检查应返回通过状态。

附加的 EBS 状态检查

附加的 EBS 状态检查可监控附加到实例的 Amazon EBS 卷是否可以访问并能够完成 I/O 操作。如果附加到实例的一个或多个 EBS 卷无法完成 I/O 操作，`StatusCheckFailed_AttachedEBS` 指标是表示受损的二进制值。这些状态检查可检测计算或 Amazon EBS 基础设施的底层问题。当附加的 EBS 状态检查指标失败时，您可以等待 Amazon 解决问题，也可以自行采取措施，例如更换受影响的卷或停止并重启实例。

以下是可能导致附加的 EBS 状态检查失败的问题的示例：

- EBS 卷底层的存储子系统的硬件或软件问题
- 物理主机上影响到 EBS 卷的可访问性的硬件问题
- 实例和 EBS 卷之间的连接问题

您可以使用 `StatusCheckFailed_AttachedEBS` 指标来帮助提高工作负载的弹性。您可以使用此指标创建基于状态检查结果触发的 Amazon CloudWatch 告警。例如，当您检测到长时间影响时，可以故障转移到辅助实例或可用区。或者，您可以使用 EBS CloudWatch 指标监控每个附加的卷的 I/O 性能，以检测和替换受损的卷。如果工作负载不会驱动挂载到实例的任何 EBS 卷的 I/O，并且 EBS 状态检查表明存在受损，则可以停止并启动该实例，以将工作负载转移到新的主机。这可以解决影响 EBS 卷可访问性的底层主机问题。有关更多信息，请参阅 Amazon EBS 的 Amazon CloudWatch 指标。

您还可以配置 Amazon EC2 Auto Scaling 组来检测附加 EBS 状态检查失败情况，然后用新的实例替换受影响的实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Monitor and replace Auto Scaling instances with impaired Amazon EBS volumes。

> **Note**
>
> 附加的 EBS 状态检查指标仅适用于 Nitro 实例。

查看 Amazon EC2 实例的状态检查

如果实例状态检查失败，通常必须由您自行解决问题（例如，通过重启实例或更改实例配置）。要自行解决系统或实例状态检查失败问题，请参阅 通过失败状态检查来排查 Amazon EC2 Linux 实例问题。

Console

查看状态检查，需要进行以下操作

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 实例页面上的状态检查列中会列出每个实例的运行状态。
4. 要查看特定实例的状态，请选择该实例，然后选择状态和警报选项卡。
5. 要查看 CloudWatch 指标以进行状态检查，请在状态和警报选项卡上，展开指标以查看以下指标的图表：
   - 系统状态检查失败
   - 实例状态检查失败
   - 附加 EBS 状态检查失败

   有关更多信息，请参阅 the section called “状态检查指标”。

Amazon CLI

查看状态检查，需要进行以下操作

使用 describe-instance-status 命令。

示例：获取所有正在运行的实例的状态

```
aws ec2 describe-instance-status
```

示例：获取所有实例的状态

```
aws ec2 describe-instance-status --include-all-instances
```

示例：获取单个正在运行的实例的状态

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

示例：获取所有状态为 `impaired` 的实例

```
aws ec2 describe-instance-status \
--filters Name=instance-status.status,Values=impaired
```

PowerShell

查看状态检查，需要进行以下操作

使用 Get-EC2InstanceStatus 命令。

示例：获取所有正在运行的实例的状态

```
Get-EC2InstanceStatus
```

示例：获取所有实例的状态

```
Get-EC2InstanceStatus -IncludeAllInstance $true
```

示例：获取单个正在运行的实例的状态

```
Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0
```

示例：获取所有状态为 `impaired` 的实例

```
Get-EC2InstanceStatus \
-Filter @{Name="instance-status.status"; Values="impaired"}
```

为未能通过状态检查的 Amazon EC2 实例创建 CloudWatch 警报

您可以使用状态检查指标创建 CloudWatch 警报，以在实例的状态检查失败时向您发送通知。

如果缺少指标数据点，状态检查和状态检查警报可能会暂时进入数据不足状态。尽管很少见，但当指标报告系统出现中断时，即使实例运行正常，也会发生这种情况。我们建议将此状态视为数据丢失，而不是状态检查失败或警报违例。为了响应事件而对实例执行停止、终止、重启或恢复操作时，这一点尤其重要。

Console

此示例配置了当实例未通过状态检查时发送通知的警报。您可以选择停止、终止或恢复实例。

创建状态检查警报

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择实例，选择状态检查选项卡，然后依次选择操作、创建状态检查警报。
4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）页面的 Add or edit alarm（添加或编辑警报）下，选择 Create an alarm（创建警报）。
5. 对于警报通知，打开开关以配置 Amazon Simple Notification Service (Amazon SNS) 通知。选择现有 Amazon SNS 主题或输入名称来创建新主题。

   如果您向收件人列表添加了电子邮件地址或创建了新的主题，则 Amazon SNS 会向每个新地址发送一封确认电子邮件。每位收件人都必须在电子邮件中选择确认链接。仅已确认的地址会收到提醒通知。
6. 对于警报操作，打开开关以指定触发警报时要执行的操作。选择操作。
7. 对于警报阈值，请指定警报的指标和条件。

   您可以保留样本分组依据（平均值）和样本数据类型（状态检查 failed:either）的默认设置，也可以根据自己的需求进行更改。

   在 Consecutive period（连续周期）中，设置要评估的周期数，然后在 Period（时间段）中输入评估时间段的持续时间，此评估时间段结束后才会触发警报并发送电子邮件。
8. （可选）对于样本指标数据，选择添加到控制面板。
9. 选择创建。

如果需要更改实例状态警报，则可以对其进行编辑。

编辑状态检查警报

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择所需实例，然后依次选择操作、监控、管理 CloudWatch 警报。
4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）页面的 Add or edit alarm（添加或编辑警报）下，选择 Edit an alarm（编辑警报）。
5. 对于 Search for alarm（搜索警报），选择警报。
6. 完成更改后，选择 Update（更新）。

Amazon CLI

在以下示例中，当实例的实例检查或系统状态检查在至少两个期间连续失败后，警报将向 SNS 主题发送通知。使用的 CloudWatch 指标为 `StatusCheckFailed`。

创建状态检查警报

1. 选择一个现有 SNS 主题或创建一个新的主题。有关更多信息，请参阅《Amazon Command Line Interface 用户指南》中的在 Amazon CLI 中访问 Amazon SNS。
2. 使用以下 list-metrics 命令查看 Amazon EC2 的可用 Amazon CloudWatch 指标。

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. 使用以下 put-metric-alarm 命令创建警报。

```
aws cloudwatch put-metric-alarm \
    --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 \
    --metric-name StatusCheckFailed \
    --namespace AWS/EC2 \
    --statistic Maximum \
    --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
    --unit Count \
    --period 300 \
    --evaluation-periods 2 \
    --threshold 1 \
    --comparison-operator GreaterThanOrEqualToThreshold \
    --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

周期为收集 Amazon CloudWatch 指标所需的时间范围（以秒为单位）。此示例使用 300，这是 60 秒乘以 5 分钟得到的结果。评估期是必须将指标数值与阈值相比较的连续周期数。此示例使用 2。警报操作是要在此警报触发时执行的操作。

PowerShell

创建状态检查警报

当实例的状态检查在至少两个期间连续失败后，按如下方式使用 Write-CWMetricAlarm cmdlet 将通知发送到 SNS 主题。

```
Write-CWMetricAlarm `
    -AlarmName "StatusCheckFailed-Alarm-for-i-1234567890abcdef0" `
```

```
    -MetricName "StatusCheckFailed" `
    -Namespace "AWS/EC2" `
    -Statistic "Maximum" `
    -Dimension @{Name="InstanceId"; Values="i-1234567890abcdef0"} `
    -Unit "Count" `
    -Period 300 `
    -EvaluationPeriod 2 `
    -Threshold 1 `
    -ComparisonOperator "GreaterThanOrEqualToThreshold" `
    -AlarmAction "arn:aws:sns:us-west-2:111122223333:my-sns-topic"
```

周期为收集 Amazon CloudWatch 指标所需的时间范围（以秒为单位）。此示例使用 300，这是 60 秒乘以 5 分钟得到的结果。评估期是必须将指标数值与阈值相比较的连续周期数。此示例使用 2。警报操作是要在此警报触发时执行的操作。

Amazon EC2 实例的状态更改事件

当实例状态发生更改时，Amazon EC2 会向 Amazon EventBridge 发送 EC2 Instance State-change Notification 事件。

以下是此事件的示例数据。在此示例中，实例进入了 pending 状态。

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2021-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
   ],
   "detail":{
      "instance-id":"i-1234567890abcdef0",
      "state":"pending"
   }
}
```

state 可使用的值为：

- pending

- running
- stopping
- stopped
- shutting-down
- terminated

当您启用或启动实例时，它会进入 pending 状态，然后进入 running 状态。当您停止实例时，它会进入 stopping 状态，然后进入 stopped 状态。当您终止实例时，它会进入 shutting-down 状态，然后进入 terminated 状态。有关更多信息，请参阅 Amazon EC2 实例状态更改。

创建会在 Amazon EC2 实例更改状态时发送电子邮件的警报

若要在实例状态更改时接收电子邮件通知，请创建一个 Amazon SNS 主题，然后为 EC2 Instance State-change Notification 事件创建一条 EventBridge 规则。

创建 SNS 主题

1. 通过 https://console.aws.amazon.com/sns/v3/home 打开 Amazon SNS 控制台。
2. 在导航窗格中，选择主题。
3. 选择创建主题。
4. 对于类型，选择标准。
5. 对于 Name（名称），请为主题输入一个名称。
6. 选择创建主题。
7. 选择创建订阅。
8. 对于协议，选择电子邮件。
9. 对于 Endpoint（端点），请输入接收通知的电子邮件地址。
10. 选择创建订阅。
11. 您将收到电子邮件消息，其主题行为：Amazon Notification - Subscription Confirmation。请按照说明确认订阅。

创建 EventBridge 规则

1. 打开位于 https://console.aws.amazon.com/events/ 的 Amazon EventBridge 控制台。
2. 选择 Create rule（创建规则）。

3. 对于 Name（名称），请为规则输入一个名称。
4. 对于规则类型，选择具有事件模式的规则。
5. 选择下一步。
6. 对于 Event pattern（事件模式），执行以下操作：
   a. 对于事件源，选择 Amazon Web Services 服务。
   b. 对于 Amazon Web Services 服务，请选择 EC2。
   c. 对于事件类型，请选择 EC2 实例状态更改通知。
   d. 默认情况下，我们会就任何实例的任何状态更改发送通知。如果愿意，您可以选择特定状态或特定实例。
7. 选择下一步。
8. 按以下操作指定目标：
   a. 对于 Target types（目标类型），选择 Amazon Web Services 服务。
   b. 对于 Select a target（选择一个目标），选择 SNS topic（SNS 主题）。
   c. 对于 Topic（主题），请选择您在上一步中创建的 SNS 主题。
9. 选择下一步。
10. （可选）向规则添加标签。
11. 选择下一步。
12. 选择 Create rule（创建规则）。
13. 为测试规则，请发起状态更改。例如，开始已停止的实例、停止正在运行的实例或启动实例。您将收到电子邮件消息，其主题行为：Amazon Notification Message。电子邮件正文包含事件数据。

Amazon EC2 实例的计划事件

为确保基础设施的可靠性和性能，Amazon 可以计划事件来重启、停止和停用您的实例。这些事件不会频繁发生。

如果您的一个实例将受某计划事件影响，则 Amazon 将使用与您的 Amazon 账户关联的电子邮件地址提前通知您。该电子邮件将提供有关该事件的详细信息，包括开始和结束日期。根据事件类型的不同，您也许能够执行操作来控制事件的发生时间。Amazon 还会发送 Amazon Health 事件，您可以使用 Amazon EventBridge 对该事件进行监控和管理。有关更多信息，请参阅 Monitoring events in Amazon Health with Amazon EventBridge。

计划的事件由 Amazon 管理。您无法为实例计划事件。不过，您可以：

- 查看实例的计划事件。
- 重新计划某些已计划的事件。
- 为计划事件创建自定义事件窗口。
- 在实例按计划重启、停止或停用时执行操作。

为确保您会收到计划事件的通知，请在账户页面上验证您的联系信息。

> **Note**
>
> 当实例受计划事件的影响并且它属于自动扩缩组的一部分时，Amazon EC2 Auto Scaling 最终将其替换为其运行状况检查的一部分，您无需采取进一步措施。有关由 Amazon EC2 Auto Scaling 执行的运行状况检查的更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 Health checks for instances in an Auto Scaling group。

计划的事件类型

Amazon EC2 可以创建以下类型的计划实例事件，这些事件会在计划的时间发生：

事件类型	事件代码	事件操作
实例停止	`instance-stop`	实例将在计划的时间停止。再次启动实例时，实例会迁移至新主机。仅适用于具有 Amazon EBS 根卷的实例。
实例指令引退	`instance-retirement`	如果实例具有 Amazon EBS 根卷，则实例将在计划的时间停止；如果实例具有实例存储根卷，则实例将在计划的时间终止。
实例重启	`instance-reboot`	实例将在计划的时间重启。实例将保留在主机上，并且主机将在重启期间进行维护。这称为就地重启。

事件类型	事件代码	事件操作
系统重启	system-reboot	实例将在计划的时间重启并迁移至新的主机。这称为重启迁移。
系统维护	system-maintenance	实例可能会在计划的时间因网络维护或电源维护而临时受到影响。

确定事件类型

您可以检查为实例计划的事件类型。

Console

确定事件类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 事件代码会显示在表中的事件类型列中。
4. 要对该表进行筛选以仅显示涉及相关实例的事件，请在搜索字段的筛选条件列表中选择资源类型：实例。

Amazon CLI

确定实例的事件类型

使用 describe-instance-status 命令。如果实例具有关联的计划事件，则输出会提供有关该计划事件的信息。

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0 \
    --query InstanceStatuses[].Events
```

下面是示例输出。计划的事件代码是 system-reboot。

```
[
```

```
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2020-03-14T22:00:00.000Z",
            "NotBefore": "2020-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2020-04-05T11:00:00.000Z"
        }
    ]
]
```

PowerShell

确定实例的事件类型

使用 Get-EC2InstanceStatus cmdlet。如果实例具有关联的计划事件，则输出会提供有关该计划事件的信息。

```
(Get-EC2InstanceStatus `
    -InstanceId i-1234567890abcdef0).Events
```

下面是示例输出。计划的事件代码是 system-reboot。

```
Code              : system-reboot
Description       : The instance is scheduled for a reboot
InstanceEventId   : instance-event-0d59937288b749b32
NotAfter          : 2020-03-14T22:00:00.000Z
NotBefore         : 2020-03-14T20:00:00.000Z
NotBeforeDeadline : 2020-04-05T11:00:00.000Z
```

内容

- 管理计划停止或停用的 Amazon EC2 实例
- 管理计划重启的 Amazon EC2 实例
- 管理计划维护的 Amazon EC2 实例
- 查看会影响 Amazon EC2 实例的计划事件
- 重新计划 EC2 实例的计划事件
- 为会影响 Amazon EC2 实例的计划事件创建自定义事件窗口

管理计划停止或停用的 Amazon EC2 实例

当 Amazon 检测到您的实例的基础主机存在无法修复的故障时，它会计划停止或终止实例，这取决于实例的根卷类型。

- 如果实例具有 Amazon EBS 根卷，则将计划停止实例。
- 如果实例具有实例存储根卷，则将计划终止实例。

有关更多信息，请参阅 实例指令引退。

> **Important**
>
> 实例停止、休眠或终止后，实例存储卷上存储的所有数据都会丢失。这包括附加到具有 EBS 根卷的实例的实例存储卷。在实例停止、休眠或终止之前，请务必保存实例存储卷中以后可能需要的数据。

您可以执行的操作

可以对具有 EBS 根卷的实例执行的操作

收到计划 `instance-stop` 事件通知时，您可以执行以下操作之一：

- 等待计划停止：您可等待实例在其计划维护时段内停止。
- 执行手动停止并启动：您也可在适合的时间自行停止并启动实例，这会将实例迁移至新的主机。这与重启实例不同。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。
- 自动停止并启动：您可以自动立即停止并启动以响应计划的 `instance-stop` 事件。有关更多信息，请参阅《Amazon Health 用户指南》中的自动在 EC2 实例上运行操作以响应 Amazon Health 中的事件。

可以对具有实例存储根卷的实例执行的操作

收到计划 `system-retirement` 事件通知，并且您希望保留数据时，您可以执行以下操作之一：

1. 从最新的 AMI 启动替换实例。
2. 在实例按计划终止之前，将所有必需的数据迁移至替换实例。
3. 终止原始实例，或等待其按计划终止。

有关您可以执行的操作的更多信息，请参阅实例指令引退。

管理计划重启的 Amazon EC2 实例

当 Amazon 必须执行安装更新或维护底层主机等任务时，可以计划实例重启。在计划重启期间，实例可以保留在同一主机上，也可迁移到其他主机，具体取决于事件，如下所示：

- `instance-reboot` 事件
  - 在重启期间，实例仍保留在主机上。这称为就地重启。
  - 当前主机正在执行维护。
  - 这通常会在几秒钟内完成。
- `system-reboot` 事件
  - 实例将在重启期间迁移至新的主机。这称为重启迁移，
  - 通常在几分钟内完成。

要检查为实例安排计划的事件类型，请参阅确定事件类型。

您可以执行的操作

收到计划的 `instance-reboot` 或 `system-reboot` 事件通知时，您可以执行以下操作之一：

- 等待计划重启：您可以等待实例在计划的维护时段内重启。
- 将计划重启改期：您可以将实例重启改期为适合自己的其他日期和时间。
- 执行用户发起的重启：您可以在自己方便的时间自行手动重启实例。但结果因事件而异：
  - `instance-reboot` 事件：实例将保留在当前硬件上（就地重启），不会进行主机维护，并且事件将保持打开状态。
  - `system-reboot` 事件
    - 如果实例启用了重启迁移，则用户发起的重启会尝试将实例迁移到新硬件上。如果操作成功，事件将被清除。如果操作不成功，则会执行就地重启并保留计划事件。
    - 如果实例禁用了重启迁移，则用户发起的重启会使实例保留在相同的硬件上（就地重启），不会进行主机维护但会保留计划事件。当计划事件最终发生时，Amazon 会将实例移至新硬件（重启迁移）。

在 Amazon 重启实例后

在 Amazon 重启实例后，将会发生以下情况：

- 该计划事件将被清除。
- 事件描述将被更新。
- 对于 `instance-reboot` 事件：
  - 底层主机的维护将会完成。
- 对于 `system-reboot` 事件：
  - 实例将迁移至新主机。
  - 实例将保留其 IP 地址和 DNS 名称。
  - 本地实例存储卷上的所有数据都会保留。
- 您可在实例完全启动之后使用实例。

替代方案

如果无法将重启事件改期或为用户发起的重启启用重启迁移，但希望在计划维护时段内保持正常运行，则可以执行以下操作：

- 对于具有 EBS 根卷的实例
  - 手动停止并启动实例，以将其迁移至新的主机。这与手动重启实例不同，手动重启时实例会保留在同一主机上。
  - 此外也可以自动立即停止并启动实例以响应计划重启事件。有关更多信息，请参阅《Amazon Health 用户指南》中的自动在 EC2 实例上运行操作以响应 Amazon Health 中的事件。

  > **Important**
  >
  > 停止实例后，实例存储卷上的数据将会丢失。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。

- 对于具有实例存储根卷的实例
  1. 从最新的 AMI 启动替换实例。
  2. 在计划维护时段开始之前，将所有必需的数据迁移至替换实例。
  3. 终止原始实例。

启用或禁用重启迁移

在为实例计划某个 `system-reboot` 事件后，您可以在事件到来之前重启该实例。用户发起的重启的结果取决于实例的重启迁移设置：

- 已启用：用户发起的重启会尝试将实例迁移到新硬件上（重启迁移）。如果操作成功，事件将被清除。如果操作不成功，则会执行就地重启并保留计划事件。请注意，即使启用了重启迁移，也只能在实例满足重启迁移要求的前提下进行重启迁移。
- 已禁用：用户发起的重启会使实例保留在相同的硬件上（就地重启），不会进行主机维护但会保留计划事件。当计划事件最终发生时，Amazon 会将实例移至新硬件（重启迁移）。

迁移后重启所需的时间比就地重启的时间长：

- 就地重启：大约 30 秒
- 迁移后重启：几分钟

> **Note**
>
> 收到 `system-reboot` 事件通知的实例会为用户发起的重启默认启用此功能。

启用重启迁移的要求

只有满足以下条件的实例才能启用重启迁移：

实例类型

并非所有实例类型都支持启用重启迁移。您可以查看支持启用重启迁移的实例类型。

Console

查看支持启用重启迁移的实例类型

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instance Types（实例类型）。
3. 在筛选条件栏中，输入重启迁移支持：已支持。选择输入字符时出现的筛选条件名称。

   实例类型表会显示支持启用重启迁移的所有实例类型。

Amazon CLI

查看支持启用重启迁移的实例类型

使用 describe-instance-types 命令和 `reboot-migration-support` 筛选条件。

```
aws ec2 describe-instance-types \
    --filters Name=reboot-migration-support,Values=supported \
    --query "InstanceTypes[*].[InstanceType]" \
    --output text | sort
```

PowerShell

查看支持启用重启迁移的实例类型

使用 Get-EC2InstanceType cmdlet 和 `reboot-migration-support` 筛选条件。

```
Get-EC2InstanceType `
    -Filter @{Name="reboot-migration-support";Values="true"} | `
    Select InstanceType | Sort-Object InstanceType
```

租赁

- 已共享
- 专用实例

有关更多信息，请参阅 Amazon EC2 专用实例。

限制

具有以下特征的实例不支持重启迁移：

- 平台：在 Xen 虚拟机监控程序上原生运行的实例
- 实例大小：`metal` 实例
- 租赁：专属主机。对专属主机使用专属主机自动恢复
- 存储：具有实例存储卷的实例
- 网络：使用 Elastic Fabric Adapter 的实例
- 自动扩缩：属于自动扩缩组的实例

启用或禁用重启迁移的步骤

实例收到 `system-reboot` 事件后，则会默认启用重启迁移。您可以禁用重启迁移，以便在用户发起的重启期间将实例保留在相同的硬件上（就地重启）。

`default` 配置不能为不受支持的实例启用重启迁移。有关更多信息，请参阅 启用重启迁移的要求。

您可以在正在运行或已停止的实例上禁用或启用重启迁移。

Amazon CLI

禁用重启迁移

使用e modify-instance-maintenance-options 命令，并将 `--reboot-migration` 参数设置为 `disabled`。

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration disabled
```

启用重启迁移

使用e modify-instance-maintenance-options 命令，并将 `--reboot-migration` 参数设置为 `default`。

```
aws ec2 modify-instance-maintenance-options \

    --instance-id i-0abcdef1234567890 \

    --reboot-migration default
```

PowerShell

禁用重启迁移

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId  `
    -RebootMigration Disabled
```

启用重启迁移

使用 Edit-EC2InstanceMaintenanceOption cmdlet。

```
Edit-EC2InstanceMaintenanceOption `
    -InstanceId i-1234567890abcdef0 `
    -RebootMigration Enabled
```

管理计划维护的 Amazon EC2 实例

当 Amazon 必须维护实例的基础主机时，它将计划实例的维护。维护事件有两种：网络维护和电源维护。

- 在网络维护期间，计划的实例会在短时间内失去网络连接。在维护完成后，将恢复与实例的正常网络连接。
- 在电源维护期间，计划的实例将短时间脱机，然后重启。执行重启后，将保留您的所有实例的配置设置。

在实例重启后 (这通常需要几分钟)，验证您的应用程序是否按预期运行。此时，您的实例应该不再具有与之关联的计划事件，或者如果有，该计划事件的描述应该以 [Completed] 开头。实例状态说明的刷新有时需要长达 1 个小时。已完成的维护事件将在 Amazon EC2 控制台面板上显示长达一周时间。

您可以执行的操作

可以对具有 EBS 根卷的实例执行的操作

收到 `system-maintenance` 事件通知时，您可以执行以下操作之一：

- 等待执行计划维护：您可等待按计划执行维护。
- 执行手动停止并启动：您可以停止并启动实例，这会将实例迁移至新的主机。这与重启实例不同。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。
- 自动停止并启动：您可以自动立即停止并启动以响应计划维护事件。有关更多信息，请参阅《Amazon Health 用户指南》中的自动在 EC2 实例上运行操作以响应 Amazon Health 中的事件。

可以对具有实例存储根卷的实例执行的操作

收到 `system-maintenance` 事件通知时，您可以执行以下操作之一：

- 等待执行计划维护：您可等待按计划执行维护。
- 启动替换实例：如果想在计划维护时段内保持正常运行，请执行以下操作：
  1. 从最新的 AMI 启动替换实例。

2. 在计划维护时段开始之前，将所有必需的数据迁移至替换实例。
3. 终止原始实例。

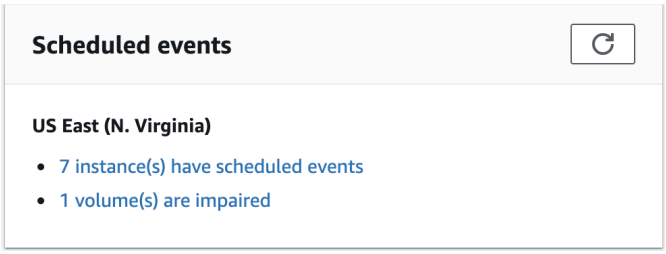
查看会影响 Amazon EC2 实例的计划事件

除了通过电子邮件接收计划事件的通知外，您还可以查看计划的事件。

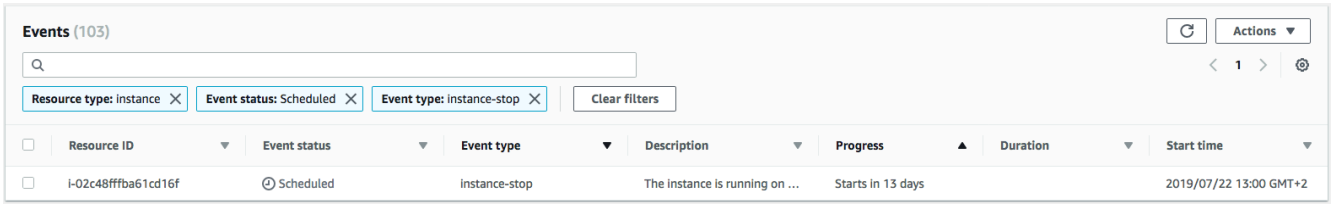
Console

查看实例的计划事件

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 控制面板的计划的事件下将显示与事件关联的所有资源。



3. 有关更多详细信息，在导航窗格中，选择事件。将显示与事件关联的所有资源。您可以按事件类型、资源类型和可用区等特征进行筛选。



Amazon CLI

查看实例的计划事件

使用 describe-instance-status 命令。

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0 \
    --query "InstanceStatuses[].Events"
```

以下示例输出显示重启事件：

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0d59937288b749b32",
            "Code": "system-reboot",
            "Description": "The instance is scheduled for a reboot",
            "NotAfter": "2019-03-15T22:00:00.000Z",
            "NotBefore": "2019-03-14T20:00:00.000Z",
            "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
         }

    ]
]
```

以下示例输出显示实例停用事件。

```
[
    "Events": [
        {
            "InstanceEventId": "instance-event-0e439355b779n26",

            "Code": "instance-stop",
            "Description": "The instance is running on degraded hardware",
            "NotBefore": "2015-05-23T00:00:00.000Z"
        }
    ]
]
```

PowerShell

查看实例的计划事件

使用以下 Get-EC2InstanceStatus 命令。

```
(Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0).Events
```

以下示例输出显示实例停用事件。

```
Code         : instance-stop
Description  : The instance is running on degraded hardware
NotBefore    : 5/23/2015 12:00:00 AM
```

Instance metadata

使用实例元数据查看实例的计划事件

您可以使用实例元数据服务版本 2 或实例元数据服务版本 1 从实例元数据中检索有关实例的活动维护事件的信息。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

以下是 JSON 格式的计划系统重启事件信息的示例输出。

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "active"
  }
]
```

使用实例元数据查看有关实例的已完成或已取消事件的事件历史记录

您可以使用实例元数据服务版本 2 或实例元数据服务版本 1 从实例元数据中检索有关已完成或已取消的事件的信息。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/events/maintenance/history
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/history
```

以下是 JSON 格式的已取消和已完成系统重启事件相关信息的示例输出。

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Canceled] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "canceled"
  },
  {
    "NotBefore" : "29 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Completed] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "29 Jan 2019 09:17:23 GMT",
    "State" : "completed"
  }
]
```

Amazon Health

您可以使用 Amazon Health Dashboard 了解可能影响您的实例的事件。Amazon Health Dashboard 会分三组管理问题：未处理问题、已计划更改和其他通知。已计划更改组包含正在进行或即将进行的项目。

有关更多信息，请参阅 Amazon Health 用户指南中的开始使用 Amazon Health Dashboard。

重新计划 EC2 实例的计划事件

您可以重新安排一个事件，以便它在适合您的特定日期和时间发生。重新计划事件后，可能需要一两分钟才能显示更新后的日期。

限制

- 只有具有事件截止日期的事件才可以重新计划。可以将事件重新计划到事件截止日期之前的日期。截止期限列（控制台）和 `NotBeforeDeadline` 字段（Amazon CLI）会指示事件的截止日期。
- 只有尚未开始的事件才可以重新计划。开始时间列（控制台）和 `NotBefore` 字段（Amazon CLI）会指示事件的开始时间。距离开始时间只剩 5 分钟的计划事件无法改期。
- 新的事件开始时间离当前时间必须至少有 60 分钟。
- 如果您使用控制台重新计划多个事件，则事件截止期限由具有最早的事件截止期限的事件决定。

Console

重新安排事件

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 从筛选器列表中选择 Resource type: instance (资源类型：实例)。
4. 选择一个或多个实例，然后依次选择 Actions (操作)、Schedule Event (计划事件)。

   只有具有事件截止期限（由 Deadline (截止期限) 值指示）的事件才可以重新计划。如果某个选定的事件没有截止期限，则会禁用 Actions (操作) 下的 Schedule Event (计划事件)。
5. 对于 New start time (新的开始时间)，为事件输入新的日期和时间。新的日期和时间必须早于 Event deadline (事件截止期限)。
6. 选择保存。

   更新的事件开始时间可能需要一两分钟才会反映在控制台中。

Amazon CLI

重新安排事件

使用 modify-instance-event-start-time 命令。

```
aws ec2 modify-instance-event-start-time \
    --instance-id i-1234567890abcdef0 \
    --instance-event-id instance-event-0d59937288b749b32 \
    --not-before 2020-03-25T10:00:00.000
```

PowerShell

重新安排事件

使用 Edit-EC2InstanceEventStartTime cmdlet。

```
Edit-EC2InstanceEventStartTime `
    -InstanceId i-1234567890abcdef0 `
    -InstanceEventId instance-event-0d59937288b749b32 `
    -NotBefore 2020-03-25T10:00:00.000
```

为会影响 Amazon EC2 实例的计划事件创建自定义事件窗口

您可以为重新启动、停止或终止 Amazon EC2 实例的计划事件定义自定义事件窗口。您可以将一个或多个实例与事件窗口关联。如果计划了这些实例的计划事件，Amazon 将在关联事件窗口中调度事件。

通过指定工作负载在非高峰期间发生的事件窗口，可以使用事件窗口最大化工作负载可用性。您还可以将事件窗口与内部维护计划保持一致。

您可以通过指定一组时间范围来定义事件窗口。最短持续时间为 2 小时。总计组合时间范围必须至少为 4 小时。

您可以使用实例 ID 或实例标签将一个或多个实例与事件窗口相关联。您还可以使用主机 ID 将专属主机与事件窗口相关联。

> **Warning**
>
> 事件窗口仅适用于停止、重新启动或终止实例的计划事件。
> 事件窗口不适用于：
>
> - 加速计划事件和网络维护事件。
> - 自动实例恢复等非计划维护和非计划重启。

使用事件窗口

- 注意事项
- 创建事件窗口
- 将目标与事件窗口关联
- 取消目标与事件窗口的关联

- 修改事件窗口
- 删除事件窗口

注意事项

- 所有事件窗口均采用 UTC 时间。
- 一个事件窗口可以包含多个时间范围。虽然每个范围必须至少为 2 小时，但所有范围的总持续时间必须至少为 4 小时。
- 一个事件窗口只能关联一个目标类型（实例 ID、专属主机 ID 或实例标签）。
- 一个目标（实例 ID、专属主机 ID 或实例标签）只能关联一个事件窗口。
- 一个事件窗口最多可以关联 100 个实例 ID 或 50 个专属主机 ID 或 50 个实例标签。实例标签可以关联任意数量的实例。
- 每个 Amazon 区域最多可以创建 200 个事件窗口。
- 与事件窗口关联的多个实例可能会同时发生计划的事件。
- 如果 Amazon 已计划事件，修改事件窗口不会更改计划事件的时间。如果事件有截止日期，您可以重新计划事件。
- 您可以在计划事件开始之前停止并启动实例。这会将实例迁移至新主机并清除该事件。

创建事件窗口

您可以创建一个或多个事件窗口。对于每个事件窗口，您可以指定一个或多个时间块。例如，您可以创建一个带有时间块的事件窗口，发生于每天凌晨 4 点，持续 2 个小时。或者，您可以创建一个带有时间块的事件窗口，发生于星期日凌晨 2 点至凌晨 4 点以及星期三凌晨 3 点至凌晨 5 点。

事件窗口每周重复出现，直至将其删除。

Console

创建事件窗口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 选择 Actions (操作)，然后选择 Manage event windows (管理事件窗口)。
4. 选择 Create instance event window (创建实例事件窗口)。
5. 对于 Event window name (事件窗口名称)，输入事件窗口的描述性名称。

6. 对于 Event window schedule (事件窗口时间表)，选择通过使用 Cron schedule builder（Cron 计划生成器）或指定时间范围来指定事件窗口中的时间块。

   - 如果选择 Cron schedule builder (Cron 计划生成器)，请指定以下内容：
     1. 对于 Days (UTC) (天数(UTC))，请指定发生事件窗口的具体日期。
     2. 对于 Start time (UTC) (开始时间(UTC))，请指定事件窗口开始的时间。
     3. 对于 Duration (持续时间)，请指定事件窗口中时间块的持续时间。每个时间块的最短持续时间为 2 小时。事件窗口的最短持续时间总计必须等于或超过 4 小时。所有时间均采用 UTC。
   - 如果选择 Time ranges (时间范围)，选择 Add new time range (添加新时间范围)，并指定开始以及结束日期和时间。每个时间范围重复此操作。每个时间范围的最短持续时间为 2 小时。所有时间范围总计最小持续时间必须等于或超过 4 小时。
7. （可选）对于目标详细信息，将一个或多个实例关联到该事件时段。使用实例 ID 或实例标签来关联实例。使用主机 ID 来关联专属主机。当这些目标列入计划维护时，该事件将在此事件时段内出现。

   请注意，您可以在不将目标与窗口关联的情况下创建事件窗口。稍后，您可以修改窗口以关联一个或多个目标。
8. （可选）对于 Event window tags (事件窗口标签)，选择 Add tag (添加标签)，然后输入标签键和值。对每个标签重复此操作。
9. 选择 Create event window (创建事件窗口)。

Amazon CLI

创建有时间范围的事件窗口

使用 create-instance-event-window 命令。

```
aws ec2 create-instance-event-window \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
```

```
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

创建有 Cron 表达式的事件窗口

使用 create-instance-event-window 命令。

```
aws ec2 create-instance-event-window \
    --cron-expression "* 21-23 * * 2,3" \
    --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
    --name myEventWindowName
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
```

```
        ]
    }
}
```

PowerShell

创建有时间范围的事件窗口

使用 New-EC2InstanceEventWindow cmdlet。

```
$timeRange = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange.StartWeekDay = "monday"
$timeRange.EndWeekDay = "wednesday"
$timeRange.StartHour = 2
$timeRange.EndHour = 8
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
    -Name my-event-window `
    -TagSpecification $tagspec `
    -TimeRange @($timeRange)
```

下面是示例输出。

```
AssociationTarget     :
CronExpression        :
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {Amazon.EC2.Model.InstanceEventWindowTimeRange}
```

创建有 Cron 表达式的事件窗口

使用 New-EC2InstanceEventWindow cmdlet。

```
$tag = @{Key="key1"; Value="value1"}
$tagspec = New-Object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "instance-event-window"
$tagspec.Tags.Add($tag)
New-EC2InstanceEventWindow `
```

```
    -Name my-event-window `
    -TagSpecification $tagspec`
    -CronExpression "* 21-23 * * 2,3"
```

下面是示例输出。

```
AssociationTarget     :
CronExpression        : * 21-23 * * 2,3
InstanceEventWindowId : iew-0abcdef1234567890
Name                  : my-event-window
State                 : creating
Tags                  : {key1}
TimeRanges            : {}
```

将目标与事件窗口关联

创建事件窗口后，您可以将目标与事件窗口相关联。一个事件窗口只能关联一个目标类型。您可以指定实例 ID、专属主机 ID 或实例标签。

Console

将目标与事件窗口相关联

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 选择要修改的事件窗口。
4. 选择操作、修改实例事件窗口。

Amazon CLI

将实例标签与事件窗口相关联

使用 associate-instance-event-window 命令。

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [
                {
                    "Key": "k2",
                    "Value": "v2"
                },
                {
                    "Key": "k1",
                    "Value": "v1"
                }
            ],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

将实例 ID 与事件窗口相关联

使用 associate-instance-event-window 命令。

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
```

```
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

将专属主机与事件窗口相关联

使用 associate-instance-event-window 命令。

```
aws ec2 associate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
            ]
        },
        "State": "creating"
    }
}
```

PowerShell

将实例标签与事件窗口相关联

使用 Register-EC2InstanceEventWindow cmdlet。

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Register-EC2InstanceEventWindow `
```

```
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1,$tag2)
```

将实例 ID 与事件窗口相关联

使用 Register-EC2InstanceEventWindow cmdlet。

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

将专属主机与事件窗口相关联

使用 Register-EC2InstanceEventWindow cmdlet。

```
Register-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

取消目标与事件窗口的关联

Console

取消目标与事件窗口的关联

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 选择要修改的事件窗口。
4. 选择操作、修改实例事件窗口。

Amazon CLI

取消实例标签与事件窗口的关联

使用disassociate-instance-event-window 命令。

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

取消实例 ID 与事件窗口的关联

使用disassociate-instance-event-window 命令。

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

取消专属主机与事件窗口的关联

使用disassociate-instance-event-window 命令。

```
aws ec2 disassociate-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

PowerShell

取消实例标签与事件窗口的关联

使用 Unregister-EC2InstanceEventWindow cmdlet。

```
$tag1 = @{Key="key1"; Value="value1"}
$tag2 = @{Key="key2"; Value="value2"}
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceTag @($tag1, $tag2)
```

取消实例 ID 与事件窗口的关联

使用 Unregister-EC2InstanceEventWindow cmdlet。

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_InstanceId i-1234567890abcdef0, i-0598c7d356eba48d7
```

取消专属主机与事件窗口的关联

使用 Unregister-EC2InstanceEventWindow cmdlet。

```
Unregister-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -AssociationTarget_DedicatedHostId h-029fa35a02b99801d
```

修改事件窗口

您可以修改事件窗口中除 ID 之外的所有字段。例如，夏令时开始时，您可能需要修改事件窗口计划。对于现有事件窗口，您可能需要添加或移除目标。

您可以在修改事件窗口时修改时间范围或 Cron 表达式，但不能同时修改两者。

Console

修改事件窗口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 选择 Actions (操作)，然后选择 Manage event windows (管理事件窗口)。
4. 选择要修改的事件窗口，然后选择 Actions (操作)、Modify instance event window (修改实例事件窗口)。
5. 在事件窗口中修改字段，然后选择 Modify event window (修改事件窗口)。

Amazon CLI

修改事件窗口的时间范围

使用 modify-instance-event-window 命令。

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
```

```
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

修改事件窗口的一组时间范围

使用 modify-instance-event-window 命令。

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":
 "wednesday", "EndHour": 8},
   {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",
 "EndHour": 8}]'
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
```

```
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            },
            {
                "StartWeekDay": "thursday",
                "StartHour": 2,
                "EndWeekDay": "friday",
                "EndHour": 8
            }
        ],
        "Name": "myEventWindowName",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

修改事件窗口的 Cron 表达式

使用 modify-instance-event-window 命令。

```
aws ec2 modify-instance-event-window
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

下面是示例输出。

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-0abcdef1234567890",
                "i-0be35f9acb8ba01f0"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

PowerShell

修改事件窗口的时间范围

使用 Edit-EC2InstanceEventWindow cmdlet。

```
$timeRange1 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange1.StartWeekDay = "monday"
$timeRange1.EndWeekDay = "wednesday"
$timeRange1.StartHour = 2
$timeRange1.EndHour = 8
$timeRange2 = New-Object Amazon.EC2.Model.InstanceEventWindowTimeRangeRequest
$timeRange2.StartWeekDay = "thursday"
$timeRange2.EndWeekDay = "friday"
$timeRange2.StartHour = 1
$timeRange2.EndHour = 6
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -TimeRange @($timeRange1, $timeRange2)
```

修改事件窗口的 Cron 表达式

使用 Edit-EC2InstanceEventWindow cmdlet。

```
Edit-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -CronExpression "* 21-23 * * 2,3"
```

删除事件窗口

您可以一次删除一个事件窗口。

Console

删除事件窗口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Events。
3. 选择 Actions (操作)，然后选择 Manage event windows (管理事件窗口)。
4. 选择要删除的事件窗口，然后选择 Actions (操作)、Delete instance event window (删除实例事件窗口)。
5. 当系统提示时，输入 **delete**，然后选择删除。

Amazon CLI

删除事件窗口

使用 delete-instance-event-window 命令并指定要删除的事件窗口。

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890
```

强制删除事件时段

使用 `--force-delete` 参数（如果事件窗口当前与目标相关联）。

```
aws ec2 delete-instance-event-window \
    --instance-event-window-id iew-0abcdef1234567890 \
```

```
    --force-delete
```

PowerShell

删除事件窗口

使用 Remove-EC2InstanceEventWindow cmdlet。

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890
```

强制删除事件时段

使用 Remove-EC2InstanceEventWindow cmdlet。

```
Remove-EC2InstanceEventWindow `
    -InstanceEventWindowId iew-0abcdef1234567890 `
    -ForceDelete $true
```

使用 CloudWatch 监控您的实例

您可以使用 Amazon CloudWatch 监控您的实例，此工具可从 Amazon EC2 收集原始数据，并将数据处理为易读的近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。

默认情况下，Amazon EC2 每隔 5 分钟向 CloudWatch 发送一次指标数据。要每隔 1 分钟向 CloudWatch 发送一次实例的指标数据，可以对实例启用详细监控。有关更多信息，请参阅 管理 EC2 实例的详细监控。

Amazon EC2 控制台将根据来自 Amazon CloudWatch 的原始数据显示一系列图表。根据您的需求，您可能更愿意从 Amazon CloudWatch 而非控制台中的图表中获取实例数据。

有关 Amazon CloudWatch 账单和成本信息，请参阅《Amazon CloudWatch 用户指南》中的 CloudWatch 账单和成本。

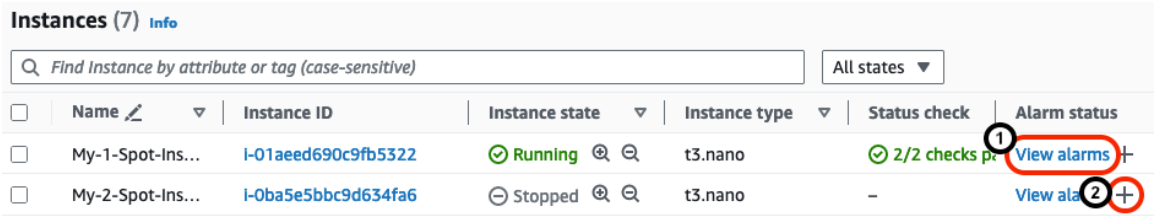
内容

- 在 Amazon EC2 控制台中管理 EC2 实例的 CloudWatch 警报
- 管理 EC2 实例的详细监控
- 可用于实例的 CloudWatch 指标

- 使用 Amazon EC2 控制台安装和配置 CloudWatch 代理以添加其他指标
- 实例的 CloudWatch 指标的统计数据
- 查看实例的监控图形
- 为实例创建 CloudWatch 警报
- 创建停止、终止、重启或恢复实例的警报

在 Amazon EC2 控制台中管理 EC2 实例的 CloudWatch 警报

从 Amazon EC2 控制台的实例屏幕中，您可以管理实例的 Amazon CloudWatch 警报。在实例表中，警报状态列提供了两个控制台控件：一个用于查看警报，一个用于创建或编辑警报。以下屏幕截图显示了这些控制台控件，编号为 1（查看警报）和 2（+ 号表示创建或编辑警报）。



从“实例”屏幕查看警报

您可以从实例屏幕查看每个实例的警报。

从“实例”屏幕查看实例的警报

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 在实例表中，对于您选择的实例，选择查看警报（上一屏幕截图中编号为 1）。
4. 在 ***i-1234567890abcdef0*** 的警报详细信息窗口中，选择该警报名称以在 CloudWatch 控制台查看该警报。

从“实例”屏幕创建警报

您可以从实例屏幕为每个实例创建警报。

从“实例”屏幕为实例创建警报

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。

3. 在实例表中，对于您选择的实例，选择加号（上一屏幕截图中编号为 2）。
4. 在管理 CloudWatch 警报屏幕中，创建警报。有关更多信息，请参阅 为实例创建 CloudWatch 警报。

从实例屏幕编辑警报

您可以从实例屏幕编辑实例的警报。

要从实例屏幕编辑实例的警报

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 在实例表中，对于您选择的实例，选择加号（上一屏幕截图中编号为 2）。
4. 在管理 CloudWatch 警报屏幕中，编辑警报。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的编辑或删除 CloudWatch 警报。

管理 EC2 实例的详细监控

Amazon CloudWatch 提供两类监控：基本监控和详细监控。默认情况下，您的实例已配置为基本监控。您可以选择启用详细监控，以帮助您更快地识别运行问题并对其采取措施。可以在实例启动时或在实例运行或停止时，启用或关闭详细监控。

在实例上启用详细监控不会影响其附加的 EBS 卷的监控。有关更多信息，请参阅 Amazon EBS 的 Amazon CloudWatch 指标。

下面突出显示了实例的基本监控和详细监控之间的区别。

监控类型	描述	收费
基本监控	状态检查指标在 1 分钟内可用。所有其他指标均在 5 分钟内可用。	免费。
详细监控	只要您为实例开启详细监控功能，便每隔 1 分钟获取一次指标。 一旦您启用了详细监控功能，就可以聚合相似实例组的数据。	您需要按 Amazon EC2 发送到 CloudWatch 的每个指标付费。您无需为数据存储付费。有关更多信息，

监控类型	描述	收费
		请参阅 Amazon CloudWatch 定价页面中的付费套餐。

内容

- 所需权限
- 启动时启用详细监控
- 管理详细监控

所需权限

要启用对实例的详细监控，用户必须具有使用 MonitorInstances API 操作的权限。要关闭对实例的详细监控，用户必须具有使用 UnmonitorInstances API 操作的权限。

启动时启用详细监控

使用以下过程在启动时启用详细监控。默认情况下，您的实例使用基本监控。

Console

要在启动实例时启用详细监视

当使用 Amazon EC2 控制台启动实例时，请在高级详细信息下，选中详细 CloudWatch 监控复选框。

Amazon CLI

要在启动实例时启用详细监视

使用 run-instances 命令和 `--monitoring` 选项。

```
--monitoring Enabled=true
```

PowerShell

要在启动实例时启用详细监视

将 New-EC2Instance cmdlet 与 `-Monitoring` 参数结合使用。

```
-Monitoring $true
```

管理详细监控

使用以下过程管理正在运行或已停止实例的详细监控。

Console

管理详细监控

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择实例。
3. 选择实例。
4. 依次选择操作、监控和故障排除以及管理详细监控。
5. 在详细监控页面上，对于详细监控，执行以下操作之一：
   - 详细监控：选择启用。
   - 基本监控：清除启用。
6. 选择确认。

Amazon CLI

启用详细监控

使用以下 monitor-instances 命令。

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

禁用详细监控

使用 unmonitor-instances 命令。

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

PowerShell

启用详细监控

使用 Start-EC2InstanceMonitoring cmdlet。

```
Start-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

禁用详细监控

使用 Stop-EC2InstanceMonitoring cmdlet。

```
Stop-EC2InstanceMonitoring -InstanceId i-1234567890abcdef0
```

可用于实例的 CloudWatch 指标

Amazon EC2 将指标发送到 Amazon CloudWatch。可使用 Amazon Web Services 管理控制台、Amazon CLI 或 API 列出 Amazon EC2 发送到 CloudWatch 的指标。默认情况下，每个数据点中包含的是实例自启动后的 5 分钟内的活动。如果您启用了详细监控，则每个数据点包含自启动后的 1 分钟内的活动。请注意，对于统计数据 Minimum、Maximum 和 Average，EC2 提供的指标的最小粒度为 1 分钟。

有关如何使用 Amazon Web Services 管理控制台或 Amazon CLI 查看可用指标的信息，请参阅《Amazon CloudWatch 用户指南》中的查看可用的指标。

有关获取这些指标的统计数据的信息，请参阅 实例的 CloudWatch 指标的统计数据。

目录

- 实例指标
- 加速器指标
- CPU 积分指标
- 专属主机指标
- 基于 Nitro 的实例的 Amazon EBS 指标
- 状态检查指标
- 流量镜像指标
- 自动扩缩组指标
- Amazon EC2 指标维度
- Amazon EC2 使用情况指标

实例指标

AWS/EC2 命名空间包括以下实例指标。

指标	说明	单位	有意义的统计数据
CPUUtilization	Amazon EC2 用于运行 EC2 实例的物理 CPU 时间的百分比，包括运行用户代码和 Amazon EC2 代码所花费的时间。 在很高的级别上，CPUUtilization 是 guest CPUUtilization 和 hypervisor CPUUtilization 的总和。 由于旧设备模拟、非旧设备配置、中断密集型工作负载、实时迁移和实时更新等因素，操作系统中的工具显示的百分比可能与 CloudWatch 不同。	Percent	• Average • Minimum • Maximum
DiskReadOps	在指定时间段内从可供实例使用的所有实例存储卷完成的读取操作数。 要计算该周期的每秒平均 I/O 操作数 (IOPS)，请将该周期的总操作数除以总秒数。 如果没有实例存储卷，则值为 0 或不报告指标。	Count	• Sum • Average • Minimum • Maximum
DiskWriteOps	在指定时间段内向可供实例使用的所有实例存储卷完成的写入操作数。 要计算该周期的每秒平均 I/O 操作数 (IOPS)，请将该周期的总操作数除以总秒数。 如果没有实例存储卷，则值为 0 或不报告指标。	Count	• Sum • Average • Minimum • Maximum
DiskReadBytes	从可供实例使用的所有实例存储卷读取的字节数。	Bytes	• Sum • Average • Minimum

指标	说明	单位	有意义的统计数据
	该指标用来确定应用程序从实例的硬盘读取的数据量。它可以用来确定应用程序的速度。 报告的数量是该期间内接收的字节数。如果您使用的是基本（5 分钟）监控，则可以将此数字除以 300 以获得字节/秒。如果您使用的是详细（1 分钟）监控，请将其除以 60。您也可以使用 CloudWatch 指标数学函数 `DIFF_TIME` 来查找每秒字节数。例如，如果您在 CloudWatch 中绘制 `DiskReadBytes` 为 `m1`，指标数学公式 `m1/(DIFF_TIME(m1))` 会返回以字节/秒为单位的指标。有关 `DIFF_TIME` 和其他指标数学函数的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。 如果没有实例存储卷，则值为 0 或不报告指标。		• Maximum

指标	说明	单位	有意义的统计数据
DiskWriteBytes	向可供实例使用的所有实例存储卷写入的字节数。 该指标用来确定应用程序向实例的硬盘写入的数据量。它可以用来确定应用程序的速度。 报告的数量是该期间内接收的字节数。如果您使用的是基本（5 分钟）监控，则可以将此数字除以 300 以获得字节/秒。如果您使用的是详细（1 分钟）监控，请将其除以 60。您也可以使用 CloudWatch 指标数学函数 DIFF_TIME 来查找每秒字节数。例如，如果您在 CloudWatch 中绘制 DiskWriteBytes 为 m1，指标数学公式 m1/(DIFF_TIME(m1)) 会返回以字节/秒为单位的指标。有关 DIFF_TIME 和其他指标数学函数的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。 如果没有实例存储卷，则值为 0 或不报告指标。	Bytes	• Sum • Average • Minimum • Maximum
MetadataNoToken	在没有令牌的情况下成功访问实例元数据服务（IMDS）的次数。 该指标用于确定是否有任何进程正在使用实例元数据服务版本 1（IMDSv1）访问实例元数据，但未使用令牌。如果所有请求都使用支持令牌的会话（即实例元数据服务版本 2（IMDSv2））），则该值为 0。有关更多信息，请参阅 转换为使用 实例元数据服务版本 2。	Count	• Sum • Percentiles

指标	说明	单位	有意义的统计数据
MetadataNoTokenRejected	在 IMDSv1 被禁用后尝试进行 IMDSv1 调用的次数。 如果出现此指标，则表示尝试了 IMDSv1 调用但被拒绝。您可以重新启用 IMDSv1，也可以确保所有调用都使用 IMDSv2。有关更多信息，请参阅 转换为使用 实例元数据服务版本 2。	Count	• Sum • Percentiles
NetworkIn	实例在所有网络接口上收到的字节数。此指标用于确定流向单个实例的传入网络流量。 报告的数量是该期间内接收的字节数。如果您使用的是基本（5 分钟）监控且统计数据为 Sum，则可以将此数字除以 300 以获得字节/秒。如果您使用的是详细（1 分钟）监控且统计数据为 Sum，请将其除以 60。	Bytes	• Sum • Average • Minimum • Maximum
NetworkOut	实例在所有网络接口上发送的字节数。此指标用于确定来自单个实例的传出网络流量。 报告的数字是该时间段内发送的字节数。如果您使用的是基本（5 分钟）监控且统计数据为 Sum，则可以将此数字除以 300 以获得字节/秒。如果您使用的是详细（1 分钟）监控且统计数据为 Sum，请将其除以 60。	Bytes	• Sum • Average • Minimum • Maximum
NetworkPacketsIn	实例在所有网络接口上收到的数据包数。此指标依据单个实例上的数据包数量来标识传入流量的量。 此指标仅可用于基本监控（5 分钟期间）。要计算实例 5 分钟内每秒收到的数据包数 (PPS)，请将 Sum 统计数据除以 300。	Count	• Sum • Average • Minimum • Maximum

指标	说明	单位	有意义的统计数据
NetworkPacketsOut	实例在所有网络接口上发送的数据包数。此指标依据单个实例上的数据包数量标识传出流量的量。 此指标仅可用于基本监控（5 分钟期间）。要计算实例 5 分钟内每秒发送的数据包数（PPS），请将 Sum 统计数据除以 300。	Count	• Sum • Average • Minimum • Maximum

加速器指标

AWS/EC2 命名空间包括您的加速型计算实例的以下加速器指标。仅在部分加速型计算实例类型中受支持。

指标	说明	单位	有意义的统计数据
GPUPowerUtilization	有功功率使用量占最大有功功率的百分比。	百分比	• Average • Minimum • Maximum

CPU 积分指标

AWS/EC2 命名空间包括 可突增性能实例的以下 CPU 积分指标。

指标	说明	单位	有意义的统计数据
CPUCreditUsage	实例为保持 CPU 使用率而花费的 CPU 积分数。一个 CPU 积分等于一个 vCPU 按 100% 利用率运行一分钟，或者 vCPU、利用率和时间的等效组合（例如，一个 vCPU 按 50% 利用率运行两分钟，或者两个 vCPU 按 25% 利用率运行两分钟）。	积分 (vCPU 分钟)	• Sum • Average • Minimum • Maximum

指标	说明	单位	有意义的统计数据
	CPU 信用指标仅每 5 分钟提供一次。如果您指定一个大于五分钟的时间段，请使用Sum 统计数据，而非 Average 统计数据。		
CPUCreditBalance	实例自启动后已累积获得的 CPU 积分数。对于 T2 标准，CPUCreditBalance 还包含已累积的启动积分数。 在获得积分后，积分将在积分余额中累积；在花费积分后，将从积分余额中扣除积分。积分余额具有最大值限制，这是由实例大小决定的。在达到限制后，将丢弃获得的任何新积分。对于 T2 标准，启动积分不计入限制。 实例可以花费 CPUCreditBalance 中的积分，以便突增到基准 CPU 使用率以上。 在实例运行过程中，CPUCreditBalance 中的积分不会过期。在 T3 或 T3a 实例停止时，CPUCreditBalance 值将保留七天。之后，所有累积的积分都将丢失。在 T2 实例停止时，CPUCreditBalance 值不会保留，并且所有累积的积分都将丢失。 CPU 信用指标仅每 5 分钟提供一次。	积分（vCPU 分钟）	• Sum • Average • Minimum • Maximum
CPUSurplusCreditBalance	在 unlimited 值为零时，CPUCreditBalance 实例花费的超额积分数。 CPUSurplusCreditBalance 值由获得的 CPU 积分支付。如果超额积分数超出实例可在 24 小时周期内获得的最大积分数，则超出最大积分数的已花费超额积分将产生额外费用。 CPU 信用指标仅每 5 分钟提供一次。	积分（vCPU 分钟）	• Sum • Average • Minimum • Maximum

指标	说明	单位	有意义的统计数据
CPUSurplusCreditsCharged	未由获得的 CPU 积分支付并且会产生额外费用的已花费超额积分数。 在出现以下任一情况时，将对花费的超额积分收费： • 花费的超额积分超出实例可在 24 小时周期内获得的最大积分数。对于超出最大积分数的所花费超额积分，将在该小时结束时向您收费。 • 实例已停止或终止。 • 实例从 unlimited 切换为 standard。 CPU 信用指标仅每 5 分钟提供一次。	积分（vCPU 分钟）	• Sum • Average • Minimum • Maximum

专属主机指标

AWS/EC2 命名空间包括 T3 专属主机的下列指标。

指标	说明	单位	有意义的统计数据
DedicatedHostCPUUtilization	在专属主机上运行的实例当前正在使用的已分配计算容量的百分比。	Percent	• Sum • Average • Minimum • Maximum

基于 Nitro 的实例的 Amazon EBS 指标

AWS/EC2 命名空间包括所添加的卷的基于 Nitro 的实例（非裸机实例）的其他 Amazon EBS 指标。

指标	说明	单位	有意义的统计数据
InstanceEBSIOPSExceededCheck	报告应用程序是否在最后一分钟内尝试驱动超过实例最大 EBS IOPS 限制的 IOPS。此指标可以是 0（未超过 IOPS），也可以是 1（已超过 IOPS）。	无	• 总和 • 平均值 • 最小值 • 最大值
InstanceEBSThroughputExceededCheck	报告应用程序是否在最后一分钟内尝试驱动超过实例最大 EBS 吞吐量限制的吞吐量。此指标可以是 0（未超过吞吐量），也可以是 1（超出吞吐量）。	无	• 总和 • 平均值 • 最小值 • 最大值
EBSReadOps	在指定时间段内在挂载到实例的所有 Amazon EBS 卷中完成的读取操作数。 要计算该时间段的平均每秒读取 I/O 操作数（读取 IOPS），请将该时间段的总操作数除以秒数。如果使用基本（5 分钟）监控，您可以将该数字除以 300 以计算读取 IOPS。如果您使用的是详细（1 分钟）监控，请将其除以 60。您也可以使用 CloudWatch 指标数学函数 DIFF_TIME 来查找每秒操作数。例如，如果您在 CloudWatch 中绘制 EBSReadOps 为 m1，指标数学公式 m1/(DIFF_TIME(m1)) 会返回以操作/秒为单位的指标。有关 DIFF_TIME 和其他指标数学函数的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。	计数	• 总和 • 平均值 • 最小值 • 最大值
EBSWriteOps	在指定时间段内在附加到实例的所有 EBS 卷中完成的写入操作数。 要计算该时间段的平均每秒写入 I/O 操作数（写入 IOPS），请将该时间段的总操作数除以秒数。如果使用基本（5 分钟）监控，您可以将该数字除以 300 以计算写入 IOPS。	计数	• 总和 • 平均值 • 最小值 • 最大值

指标	说明	单位	有意义的统计数据
	如果您使用的是详细（1 分钟）监控，请将其除以 60。您也可以使用 CloudWatch 指标数学函数 DIFF_TIME 来查找每秒操作数。例如，如果您在 CloudWatch 中绘制 EBSWriteOps 为 m1，指标数学公式 m1/(DIFF_TIME(m1)) 会返回以操作/秒为单位的指标。有关 DIFF_TIME 和其他指标数学函数的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。		
EBSReadBytes	在指定时间段内从附加到实例的所有 EBS 卷中读取的字节数。 报告的数字是在该时间段内读取的字节数。如果使用基本（5 分钟）监控，您可以将该数字除以 300 以计算每秒读取的字节数。如果您使用的是详细（1 分钟）监控，请将其除以 60。您也可以使用 CloudWatch 指标数学函数 DIFF_TIME 来查找每秒字节数。例如，如果您在 CloudWatch 中绘制 EBSReadBytes 为 m1，指标数学公式 m1/(DIFF_TIME(m1)) 会返回以字节/秒为单位的指标。有关 DIFF_TIME 和其他指标数学函数的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。	字节	• 总和 • 平均值 • 最小值 • 最大值

指标	说明	单位	有意义的统计数据
EBSWriteBytes	在指定时间段内写入附加到实例的所有 EBS 卷的字节数。 报告的数字是在该时间段内写入的字节数。如果使用基本（5 分钟）监控，您可以将该数字除以 300 以计算每秒写入的字节数。如果您使用的是详细（1 分钟）监控，请将其除以 60。您也可以使用 CloudWatch 指标数学函数 DIFF_TIME 来查找每秒字节数。例如，如果您在 CloudWatch 中绘制 EBSWriteBytes 为 m1，指标数学公式 m1/(DIFF_TIME(m1)) 会返回以字节/秒为单位的指标。有关 DIFF_TIME 和其他指标数学函数的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用指标数学。	字节	• 总和 • 平均值 • 最小值 • 最大值
EBSIOBalance%	提供有关突增存储桶中剩余的 I/O 积分百分比的信息。此指标仅对基本监控可用。 此指标仅适用于某些 *.4xlarge 大小和更小的实例，这样的实例仅需 30 分钟便可突增到最高性能，且至少每 24 小时发生一次。 Sum 统计数据不适用于该指标。	百分比	• 最小值 • 最大值
EBSByteBalance%	提供有关突增存储桶中剩余的吞吐量积分百分比的信息。此指标仅对基本监控可用。 此指标仅适用于某些 *.4xlarge 大小和更小的实例，这样的实例仅需 30 分钟便可突增到最高性能，且至少每 24 小时发生一次。 Sum 统计数据不适用于该指标。	百分比	• 最小值 • 最大值

有关为 EBS 卷提供的指标的信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS 卷的指标。有关为 EC2 实例集和竞价型实例集提供的指标的信息，请参阅使用 CloudWatch 监控 EC2 实例集或竞价型实例集。

状态检查指标

默认情况下，状态检查指标可在 1 分钟的频率下免费提供。对于新启动的实例，状态检查指标数据仅在实例完成初始化状态之后（实例进入 `running` 状态的几分钟之内）提供。有关 EC2 状态检查的更多信息，请参阅Amazon EC2 实例的状态检查。

`AWS/EC2` 命名空间包括以下状态检查指标。

指标	说明	单位	有意义的统计数据
`StatusCheckFailed`	报告实例在上一分钟内是否通过了所有系统状态检查。 此指标可以是 `0`（通过）或 `1`（失败）。 默认情况下，此指标可在 1 分钟的频率下免费提供。	Count	• Average • Minimum • Maximum
`StatusCheckFailed_Instance`	报告实例在上个 1 分钟内是否通过了 实例状况检查。 此指标可以是 `0`（通过）或 `1`（失败）。 默认情况下，此指标可在 1 分钟的频率下免费提供。	Count	• Average • Minimum • Maximum
`StatusCheckFailed_System`	报告实例在上一分钟内是否通过了 系统状况检查。 此指标可以是 `0`（通过）或 `1`（失败）。 默认情况下，此指标可在 1 分钟的频率下免费提供。	Count	• Average • Minimum • Maximum
`StatusCheckFailed_`	报告实例在上一分钟内是否通过了附加的 EBS 卷状态检查。	Count	• Average

指标	说明	单位	有意义的统计数据
AttachedEBS	此指标可以是 0（通过）或 1（失败）。 默认情况下，此指标可在 1 分钟的频率下免费提供。		• Minimum • Maximum

AWS/EBS 命名空间包括以下状态检查指标。

指标	说明	单位	有意义的统计数据
VolumeStalledIOCheck	注意：仅适用于 Nitro 实例。对附加到 Amazon ECS 和 Amazon Fargate 任务的卷未发布。 报告卷在最后一分钟是否通过停滞的 IO 检查。此指标可以是 0（通过）或 1（失败）。	无	• Average • Minimum • Maximum

流量镜像指标

AWS/EC2 命名空间包含镜像流量的指标。有关更多信息，请参阅《Amazon VPC Traffic Mirroring 指南》中的使用 Amazon CloudWatch 监控镜像的流量。

自动扩缩组指标

AWS/AutoScaling 命名空间包括自动扩缩组的指标。有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Monitor CloudWatch metrics for your Auto Scaling groups and instances。

Amazon EC2 指标维度

您可以使用以下维度来优化上表中列出的指标。

维度	描述
AutoScalingGroupName	该维度筛选您为指定容量组中的所有实例请求的数据。如果您使用自动扩缩，自动扩缩组就是您定义的实例集合。当实例在上述

维度	描述
	自动扩缩组中时，该维度仅供 Amazon EC2 指标使用。可供启用了详细或基本监控的实例使用。
ImageId	该维度筛选您为运行此 Amazon EC2 亚马逊机器映像（AMI）的所有实例而请求的数据。可供启用了详细监控功能的实例使用。
InstanceId	该维度筛选您仅为已识别实例请求的数据。这样有助于您精确定位要对其监控数据的确切实例。
InstanceType	该维度筛选您为以这一指定实例类型运行的所有实例请求的数据。这样有助于您按运行的实例类型给数据分类。例如，您可以比较 m1.small 实例和 m1.large 实例的数据，以确定哪一个对您的应用程序具有更好的商业价值。可供启用了详细监控功能的实例使用。

Amazon EC2 使用情况指标

您可以使用 CloudWatch 用量指标来提供账户资源使用情况的可见性。使用这些指标在 CloudWatch 图表和控制面板上可视化当前服务用量。

Amazon EC2 用量指标与 Amazon 服务配额对应。您可以配置警报，以在用量接近服务配额时向您发出警报。有关 CloudWatch 与 Service Quotas 集成的更多信息，请参阅《Amazon CloudWatch 用户指南》中的 Amazon 使用情况指标。

Amazon EC2 在 AWS/Usage 命名空间中发布以下指标。

指标	描述
ResourceCount	您账户中运行的指定资源的数量。资源由与指标关联的维度定义。 此指标最有用的统计数据是 MAXIMUM，这表示 1 分钟期间内使用的最大资源数。

以下维度用于优化由 Amazon EC2 发布的用量指标。

维度	描述
Service	包含该资源的 Amazon 服务的名称。对于 Amazon EC2 用量指标，此维度的值为 EC2。
Type	正在报告的实体的类型。目前，Amazon EC2 用量指标的唯一有效值为 Resource。
Resource	正在运行的资源的类型。目前，Amazon EC2 用量指标的唯一有效值是 vCPU，它返回有关正在运行的实例的信息。
Class	所跟踪的资源的类。对于以 vCPU 作为 Resource 维度的值的 Amazon EC2 用量指标，有效值为 Standard/OnDemand 、F/OnDemand 、G/OnDemand 、Inf/OnDemand 、P/OnDemand 和 X/OnDemand 。 此维度的值定义由该指标报告的实例类型的第一个字母。例如，Standard/OnDemand 返回有关类型以 A、C、D、H、I、M、R、T 和 Z 开头的所有正在运行的实例的信息，并且 G/OnDemand 返回有关类型以 G 开头的所有正在运行的实例的信息。

使用 Amazon EC2 控制台安装和配置 CloudWatch 代理以添加其他指标

对于 Amazon EC2，使用 Amazon EC2 控制台安装和配置 CloudWatch 代理的功能为试用版，可能随时更改。

默认情况下，Amazon CloudWatch 提供基本指标（例如 CPUUtilization 和 NetworkIn）用于监控 Amazon EC2 实例。要收集其他指标，您可以在 EC2 实例上安装 CloudWatch 代理，然后将该代理配置为发出所选指标。您可以使用 Amazon EC2 控制台执行此操作，而不必在每个 EC2 实例上手动安装和配置 CloudWatch 代理。

您可以使用 Amazon EC2 控制台在实例上安装 CloudWatch 代理以及配置代理以发出所选指标。

或者，要手动完成此过程，请参阅《Amazon CloudWatch 用户指南》中的安装 CloudWatch 代理。有关 CloudWatch 代理的更多信息，请参阅使用 CloudWatch 代理采集指标、日志和跟踪数据。

内容

- 前提条件
- 工作原理
- 成本
- 安装和配置 CloudWatch 代理

前提条件

要使用 Amazon EC2 安装和配置 CloudWatch 代理，您必须满足本节所述的用户和实例前提条件。

> **Tip**
>
> 此功能并非在所有 Amazon Web Services 区域 中都可用。如果 Amazon EC2 控制台中不存在本页安装过程中描述的菜单项，并且您可以灵活决定实例的运行位置，请尝试其他区域。否则，可以使用《Amazon CloudWatch 用户指南》中的手动说明安装和配置代理。

用户先决条件

要使用此功能，您的 IAM 控制台用户或角色必须具有使用 Amazon EC2 所需的权限以及下列 IAM 权限：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/EC2-Custom-Metrics-*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:SendCommand",
                "ssm:ListCommandInvocations",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetInstanceProfile",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

实例先决条件

- 实例状态：`running`
- 支持的操作系统：Linux
- Amazon Systems Manager Agent（SSM Agent）：已安装。关于 SSM Agent 的两个注意事项：
  - SSM Agent 预装在由 Amazon 和受信任的第三方提供的一些亚马逊机器映像（AMI）上。有关支持的 AMI 和 SSM Agent 安装说明的信息，请参阅《Amazon Systems Manager 用户指南》中的预装了 SSM Agent 的亚马逊机器映像（AMI）。
  - 如果在使用 SSM Agent 时遇到问题，请参阅《Amazon Systems Manager 用户指南》中的 SSM Agent 故障排除。
- 实例的 IAM 权限：必须将以下 Amazon 托管策略添加到附加到实例的 IAM 角色中：
  - AmazonSSMManagedInstanceCore – 使实例能够使用 Systems Manager 安装和配置 CloudWatch 代理。
  - CloudWatchAgentServerPolicy – 使实例能够使用 CloudWatch 代理将数据写入 CloudWatch。

  有关如何为实例添加 IAM 权限的信息，请参阅《IAM 用户指南》中的使用实例配置文件。

工作原理

在使用 Amazon EC2 控制台安装和配置 CloudWatch 代理之前，必须确保 IAM 用户或角色以及要添加指标的实例满足某些先决条件。然后，可以使用 Amazon EC2 控制台在所选实例上安装和配置 CloudWatch 代理。

首先满足先决条件

- 需要所需的 IAM 权限 – 开始之前，请确保控制台用户或角色具有使用此功能所需的 IAM 权限。
- 实例 – 要使用该功能，EC2 实例必须是 Linux 实例，安装了 SSM Agent，具有所需的 IAM 权限并且正在运行。

然后可以使用该功能

1. 选择实例 – 在 Amazon EC2 控制台中，选择要在其上安装和配置 CloudWatch 代理的实例。然后，可以通过选择“配置 CloudWatch 代理”来开始该过程。
2. 验证 SSM Agent – Amazon EC2 会检查每个实例上是否安装并启动了 SSM Agent。未通过此检查的任何实例都将从过程中排除。在此过程中，SSM Agent 用于对实例执行操作。
3. 验证 IAM 权限 – Amazon EC2 会检查每个实例是否具有此过程所需的 IAM 权限。未通过此检查的任何实例都将从过程中排除。IAM 权限使 CloudWatch 代理能够从实例收集指标并与Amazon Systems Manager 集成以使用 SSM Agent。
4. 验证 CloudWatch 代理 – Amazon EC2 检查 CloudWatch 代理是否已安装并在每个实例上运行。如果任何实例未通过此检查，Amazon EC2 会主动为您安装和启动 CloudWatch 代理。此过程完成后，CloudWatch 代理将收集每个实例上的选定指标。
5. 选择指标配置 – 选择 CloudWatch 代理要从实例中发出的指标。选择后，Amazon EC2 会将配置文件存储在 Parameter Store 中，该文件将一直保留到过程完成。除非过程中断，否则 Amazon EC2 将从 Parameter Store 中删除配置文件。请注意，如果没有选择某指标，但之前已将其添加到实例中，则在此过程完成后，将从实例中删除该指标。
6. 更新 CloudWatch 代理配置 – Amazon EC2 将指标配置发送到 CloudWatch 代理。这是过程的最后一步。如果成功，则实例可以为所选指标发出数据，Amazon EC2 将从 Parameter Store 中删除配置文件。

成本

在此过程中添加的其他指标将作为自定义指标计费。有关 CloudWatch 指标定价的信息，请参阅 Amazon CloudWatch 定价。

安装和配置 CloudWatch 代理

可以使用 Amazon EC2 控制台来安装和配置 CloudWatch 代理，以添加其他指标。

> **Note**
>
> 每次执行此过程时，都将覆盖现有 CloudWatch 代理配置。如果没有选择之前选中的指标，则将从实例中删除该指标。

使用 Amazon EC2 控制台安装和配置 CloudWatch 代理

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择要在其上安装和配置 CloudWatch 代理的实例。
4. 选择操作、监控和故障排除、配置 CloudWatch 代理。
5. 对于该过程中的每个步骤，请阅读控制台文本，然后选择下一步。
6. 要完成该过程，请在最后一步中选择完成。

更新 Amazon EC2 控制台创建的代理配置

您可以手动自定义 EC2 控制台创建的配置。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的手动创建或编辑 CloudWatch 代理配置文件。

实例的 CloudWatch 指标的统计数据

您可以获取有关实例的 CloudWatch 指标的统计信息。统计数据是指定时间段内的指标数据汇总。CloudWatch 提供统计数据的依据是您的自定义数据所提供的指标数据点，或其他 Amazon 服务向 CloudWatch 提供的指标数据点。聚合通过使用命名空间、指标名称、维度以及数据点度量单位在您指定的时间段内完成。下表介绍了可用的统计信息。

统计数据	描述
`Minimum`	指定时间段内的最低观察值。可以使用此值来决定应用程序的活动量是否较低。
`Maximum`	指定时间段内的最高观察值。可以使用此值来决定应用程序的活动量是否较高。
`Sum`	为匹配指标所提交的所有的值添加在一起。此统计信息的作用是决定指标的总量。

统计数据	描述
Average	指定时间段内 Sum / SampleCount 的值。通过将此统计信息与 Minimum 和 Maximum 进行比较，可以决定指标的完整范围以及平均使用率与 Minimum 和 Maximum 的接近程度。这样的比较可以帮助了解何时应该根据需要增加或减少资源。
SampleCount	数据点计数 (数量) 用于统计信息的计算。
pNN.NN	指定的百分位数的值。您可以指定任何百分位数，最多使用两位小数 (例如 p95.45)。

内容

- 获取特定实例的统计数据
- 聚合不同实例的统计数据
- 按自动扩缩组聚合统计数据
- 按 AMI 聚合统计数据

获取特定实例的统计数据

您可以使用 Amazon Web Services 管理控制台或 Amazon CLI 来获取特定实例的统计数据。以下示例显示了如何使用 Amazon Web Services 管理控制台 或 Amazon CLI 来确定特定 EC2 实例的最大 CPU 利用率。

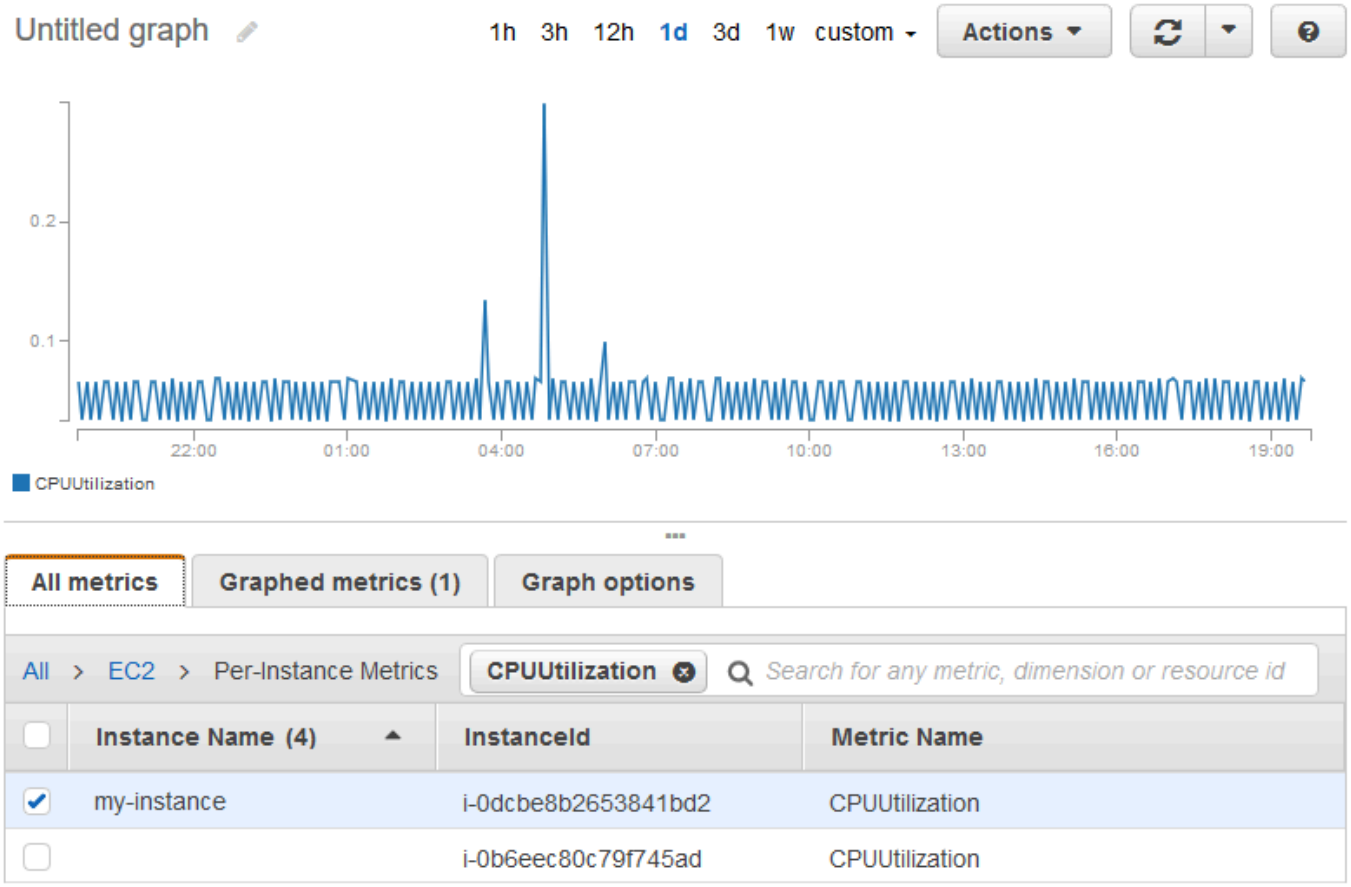
要求

- 您必须拥有实例的 ID。可使用 Amazon Web Services 管理控制台 或 describe-instances 命令获取实例 ID。
- 默认情况下，基本监控已启用，但您可以启用详细监控。有关更多信息，请参阅 管理 EC2 实例的详细监控。

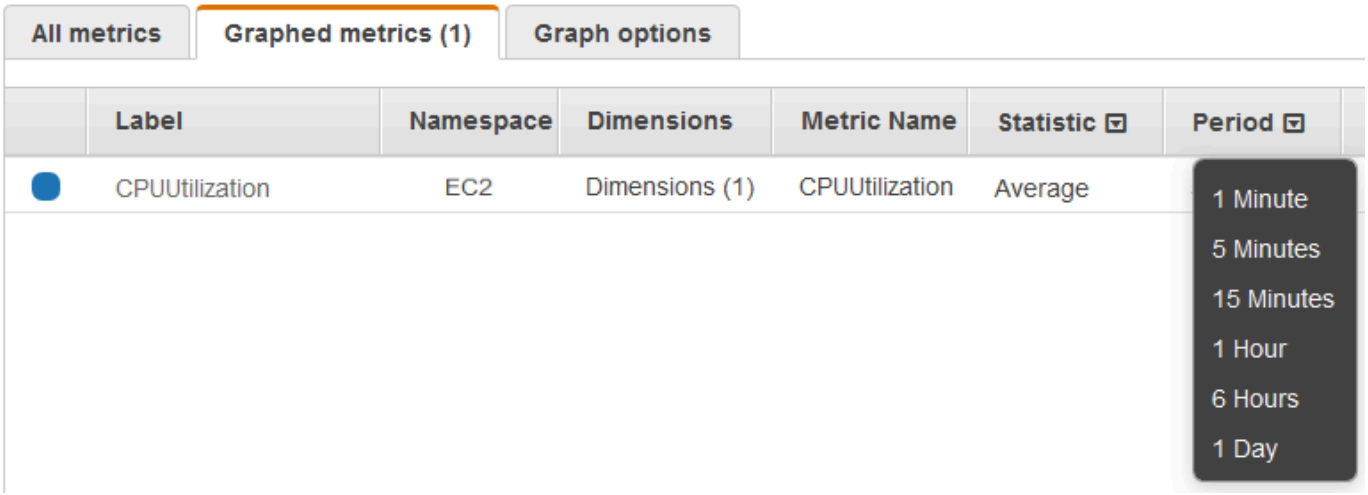
显示指定实例的 CPU 利用率（控制台）

1. 通过 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 在导航窗格中，选择指标。

3. 选择 EC2 指标命名空间。
4. 选择 Per-Instance Metrics (每个实例的指标) 维度。
5. 在搜索框中，输入 **`CPUUtilization`** 并按 Enter。选择特定实例所在的行，这将显示该实例的 CPUUtilization 指标的图表。要为该图标命名，请选择铅笔图标。要更改时间范围，请选择某个预定义的值或选择 custom。



6. 要更改指标的统计数据或时间段，请选择 Graphed metrics 选项卡。选择列标题或单个值，然后选择其他值。



获取特定实例的 CPU 利用率 (Amazon CLI)

使用以下 get-metric-statistics 命令获取指定实例的 CPUUtilization 指标（使用指定周期和时间间隔）：

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --period 3600 \
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
--start-time 2022-10-18T23:18:00 --end-time 2022-10-19T23:18:00
```

下面是示例输出。每个数值代表一个 EC2 实例的最大 CPU 使用率百分比。

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
```

```
        },
        {
            "Timestamp": "2022-10-19T12:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

聚合不同实例的统计数据

聚合统计信息适用于已经启用详细监控的实例。聚合中不包含使用基本监控的实例。在获取多实例聚合统计数据前，必须启用详细监控（另外收费），以提供以 1 分钟为间隔的数据。

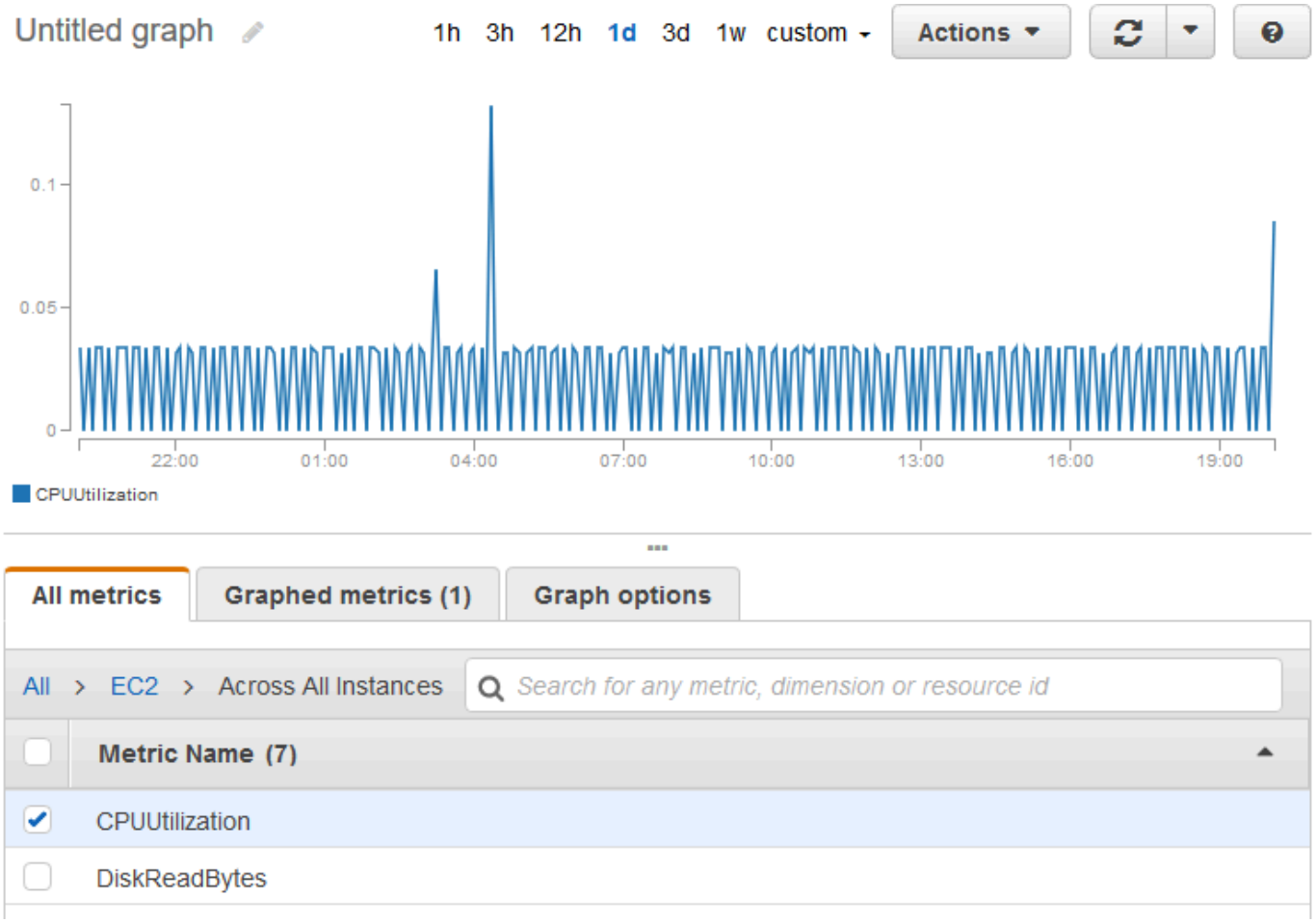
请注意，Amazon CloudWatch 不能跨各个 Amazon 区域聚合数据。指标在各区域间彼此独立。

此示例显示了如何使用详细监控来获取 EC2 实例的平均 CPU 使用率。因为未指定任何维度，所以 CloudWatch 会返回 AWS/EC2 命名空间中所有维度的统计数据。

> **Important**
>
> 此方法可以在 Amazon 命名空间中检索所有维度，但不适用于发布到 Amazon CloudWatch 的自定义命名空间。对于自定义命名空间，必须指定与任意给定数据关联的完整的维度组，以检索包含数据点的统计数据。

显示实例的平均 CPU 利用率（控制台）

1. 通过 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 在导航窗格中，选择指标。
3. 选择 EC2 命名空间，然后选择 Across All Instances (跨所有实例)。
4. 选择包含 CPUUtilization 的行，这将显示所有 EC2 实例的指标的图表。要为该图标命名，请选择铅笔图标。要更改时间范围，请选择某个预定义的值或选择 custom。



5. 要更改指标的统计数据或时间段，请选择 Graphed metrics 选项卡。选择列标题或单个值，然后选择其他值。

获取实例的平均 CPU 利用率 (Amazon CLI)

使用 get-metric-statistics 命令 (如下所示) 获取实例的平均 CPUUtilization 指标。

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/EC2 \
    --metric-name CPUUtilization \
    --period 3600  --statistics "Average" "SampleCount" \
    --start-time 2022-10-11T23:18:00 \
    --end-time 2022-10-12T23:18:00
```

下面是示例输出：

```
{
```

```
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2022-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2022-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


按自动扩缩组聚合统计数据

您可以聚合自动扩缩组中 EC2 实例的统计数据。请注意，Amazon CloudWatch 不能跨各个 Amazon 区域聚合数据。指标在各区域间彼此独立。

此示例说明如何检索为一个自动扩缩组写入磁盘的字节总数。总数以 1 分钟为周期 24 小时为间隔针对指定自动扩缩组中的所有 EC2 实例计算得出。

显示一个自动扩缩组中的实例的 DiskWriteBytes（控制台）

1. 通过 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 在导航窗格中，选择指标。
3. 选择 EC2 命名空间，然后选择 By Auto Scaling Group。
4. 选择 DiskWriteBytes 指标和特定自动扩缩组所在的行，这将显示自动扩缩组中实例的指标的图表。要为该图标命名，请选择铅笔图标。要更改时间范围，请选择某个预定义的值或选择 custom。
5. 要更改指标的统计数据或时间段，请选择 Graphed metrics 选项卡。选择列标题或单个值，然后选择其他值。

显示一个自动扩缩组中的实例的 DiskWriteBytes（Amazon CLI）

使用 get-metric-statistics 命令，如下所示。

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
 --period 360 \
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --
start-time 2022-10-16T23:18:00 --end-time 2022-10-18T23:18:00
```

下面是示例输出：

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2022-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2022-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

按 AMI 聚合统计数据

您可以通过 AMI 聚合已启用详细监控的实例的统计数据。聚合中不包含使用基本监控的实例。在获取多实例聚合统计数据前，必须启用详细监控（另外收费），以提供以 1 分钟为间隔的数据。

请注意，Amazon CloudWatch 不能跨各个 Amazon 区域聚合数据。指标在各区域间彼此独立。

此示例显示了如何确定使用特定亚马逊机器映像（AMI）的所有实例的平均 CPU 使用率。平均值以 60 秒为时间间隔 1 天为周期。

按 AMI 显示平均 CPU 利用率（控制台）

1. 通过 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。

2. 在导航窗格中，选择指标。
3. 选择 EC2 命名空间，然后选择 By Image (AMI) Id。
4. 选择 CPUUtilization 指标和特定 AMI 所在的行，这将显示指定 AMI 的指标的图表。要为该图标命名，请选择铅笔图标。要更改时间范围，请选择某个预定义的值或选择 custom。
5. 要更改指标的统计数据或时间段，请选择 Graphed metrics 选项卡。选择列标题或单个值，然后选择其他值。

获取某个映像 ID 的平均 CPU 利用率 (Amazon CLI)

使用 get-metric-statistics 命令，如下所示。

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --period 3600 \
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-
time 2022-10-10T00:00:00 --end-time 2022-10-11T00:00:00
```

下面是示例输出。每个数值代表运行指定 AMI 的 EC2 实例的平均 CPU 使用率百分比。

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2022-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

查看实例的监控图形

在您启动实例后，可以打开 Amazon EC2 控制台并在 Monitoring（监控）选项卡上查看实例的监控图表。每个图表以一个可用的 Amazon EC2 指标为基础。

可供使用图形如下：

- CPU 平均使用率 (%)
- 平均读磁盘数 (字节)
- 平均写磁盘数 (字节)
- 最大网络输入 (字节)
- 最大网络输出 (字节)
- 读磁盘操作概括 (计数)
- 写磁盘操作概括 (计数)
- 状态概括 (任意)
- 实例状态概括 (计数)
- 系统状态概括 (计数)

有关指标及其向图表提供的数据的更多信息，请参阅 可用于实例的 CloudWatch 指标。

使用 CloudWatch 控制台绘制指标图标

您还可以使用 CloudWatch 控制台将 Amazon EC2 和其他 Amazon 服务生成的指标数据绘制成图表。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的绘制指标图表。

为实例创建 CloudWatch 警报

您可以创建 CloudWatch 警报来监控您的任一实例的 CloudWatch 指标。当该指标达到指定阈值时，CloudWatch 自动向您发送通知。您可以使用 Amazon EC2 控制台创建 CloudWatch 警报，或者使用 CloudWatch 控制台提供的更多高级选项。

使用 CloudWatch 控制台创建警报

有关示例，请参阅Amazon CloudWatch 用户指南中的创建 Amazon CloudWatch 警报。

使用 Amazon EC2 控制台创建警报

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 Instances (实例)。
3. 选择所需实例，然后依次选择 Actions（操作）、Monitor and troubleshoot（监控和问题排查）、Manage CloudWatch alarms（管理 CloudWatch 警报）。
4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）详细信息页面中的 Add or edit alarm（添加或编辑警报）下，选择 Create an alarm（创建警报）。
5. 对于警报通知，选择是否配置 Amazon Simple Notification Service（Amazon SNS）通知。输入现有 Amazon SNS 主题或输入名称来创建新主题。
6. 对于警报操作，选择是否指定触发警报时要执行的操作。从列表中选择操作。
7. 对于警报阈值，选择警报的指标和条件。例如，要创建一个当 CPU 利用率在 5 分钟内达到 80% 时触发的警报，请执行以下操作：

   a. 为样本分组依据（平均）和要采样的数据类型（CPU 利用率）保留默认设置。

   b. 对于警报触发条件，选择 **>=** 并输入 **`0.80`** 作为百分比。

   c. 对于连续周期，输入 **`1`** 并选择 5 分钟作为周期。
8. （可选）对于样本指标数据，选择添加到控制面板。
9. 选择创建。

您可以从 Amazon EC2 控制台或 CloudWatch 控制台编辑 CloudWatch 警报设置。如果要删除警报，可以从 CloudWatch 控制台执行此操作。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的编辑或删除 CloudWatch 警报。

创建停止、终止、重启或恢复实例的警报

利用 Amazon CloudWatch 警报操作，您可创建自动停止、终止、重启或恢复实例的警报。当不再需要某个实例运行时，您可使用停止或终止操作来帮助您节省资金。如果发生了系统损害，您可使用重启和恢复操作自动重启这些实例或将它们恢复到新硬件上。

> **Note**
>
> 有关 Amazon CloudWatch 告警账单和成本信息，请参阅《Amazon CloudWatch 用户指南》中的 CloudWatch 账单和成本。

`AWSServiceRoleForCloudWatchEvents` 服务相关角色使 Amazon 能够代表您执行警报操作。当您首次在 Amazon Web Services 管理控制台、Amazon CLI 或 IAM API 中创建告警时，CloudWatch 会为您创建服务相关角色。

在许多情况下，您可能需要自动终止或停止实例。例如，您可能拥有专用于批工资单处理作业或科学计算任务的实例，这些实例在运行一段时间后就完成了其工作。与其让这些实例空闲 (并产生费用)，不如将其停止或终止以节省开支。使用停止警报操作和使用终止警报操作的主要区别在于，停止的实例在以后需要再次运行时，可以轻松地启动，并且可以保留相同的实例 ID 和根卷，而终止的实例则无法再启动。如此就必须启动一个新的实例。停止或终止实例后，实例存储卷上的数据将丢失。

您可以向为 Amazon EC2 每个实例指标设置的任何警报添加停止、终止、重启或恢复操作，这些指标包括 Amazon CloudWatch 提供的基本和详细监控指标（在 AWS/EC2 命名空间中），以及包含 InstanceId 维度的任何自定义指标，只要其值引用有效运行的 Amazon EC2 实例。

> **Important**
>
> 如果缺少指标数据点，状态检查警报可能会暂时进入 INSUFFICIENT_DATA 状态。尽管很少见，但当指标报告系统出现中断时，即使实例运行正常，也会发生这种情况。我们建议您将 INSUFFICIENT_DATA 状态视为数据丢失而不是警报违例，尤其是在配置警报以停止、终止、重启或恢复实例时。

控制台支持

可使用 Amazon EC2 控制台或 CloudWatch 控制台创建警报。本文档中的过程使用 Amazon EC2 控制台。有关使用 CloudWatch 控制台的过程，请参阅《Amazon CloudWatch 用户指南》中的创建停止、终止、重新启动或恢复实例的告警。

权限

您必须拥有 iam:CreateServiceLinkedRole 才能创建或修改执行 EC2 告警操作的告警。服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 Amazon Web Services 服务 委派权限的角色。

内容

- 在 Amazon CloudWatch 警报中添加停止操作
- 在 Amazon CloudWatch 警报中添加终止操作
- 在 Amazon CloudWatch 警报中添加重启操作
- 在 Amazon CloudWatch 警报中添加恢复操作
- Amazon CloudWatch 警报操作场景

在 Amazon CloudWatch 警报中添加停止操作

可以创建当达到一定阈值后停止 Amazon EC2 实例的警报。例如，您可能运行了开发或测试实例而偶尔忘记将其关闭。可以创建当平均 CPU 使用率低于 10% 达 24 小时时触发的警报，同时告知其为空闲并不再使用。可以根据需要调整阈值、持续时间和周期，还可以添加 Amazon Simple Notification Service (Amazon SNS) 通知，以便在触发警报后，您能够收到电子邮件。

可以停止或终止将 Amazon EBS 卷用作根卷的实例，但只能终止将实例存储用作根卷的实例。停止或终止实例后，实例存储卷上的数据将丢失。

创建停止空闲实例的警报（Amazon EC2 控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择所需实例，然后依次选择 Actions（操作）、Monitor and troubleshoot（监控和问题排查）、Manage CloudWatch alarms（管理 CloudWatch 警报）。

   或者，您可以在 Alarm status（警报状态）列中选择加号
   (
)。
4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）页面上，执行以下操作：
   a. 选择 Create an alarm（创建警报）。
   b. 要在警报触发时接收电子邮件，对于 Alarm notification（警报通知），选择现有的 Amazon SNS 主题。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。有关更多信息，请参阅Amazon Simple Notification Service 开发人员指南中的使用 Amazon SNS 进行应用程序对人 (A2P) 消息传送。
   c. 开启 Alarm action（警报操作），然后选择 Stop（停止）。
   d. 对于 Group samples by（样本分组方式）和 Type of data to sample（要采样的数据类型），选择统计数据和指标。在此示例中，选择 Average（平均值）和 CPU utilization（CPU 利用率）。
   e. 对于 Alarm When（警报时间）和 Percent（百分比），请指定指标阈值。在此示例中，请指定 <= 和 10%。
   f. 对于 Consecutive period（连续周期）和 Period（周期），请指定警报的评估周期。在此示例中，将 1 个连续周期指定为 5 分钟。
   g. Amazon CloudWatch 自动为您创建警报名称。要更改名称，请在 Alarm name（警报名称）中输入新名称。警报名称必须仅包含 ASCII 字符。

> **Note**
>
> 可以在创建警报前根据自己的要求调整警报配置，也可以在之后编辑配置。这包括指标、阈值、时长、操作和通知等设置。但是，警报创建后其名称无法再次编辑。

h. 选择创建。

在 Amazon CloudWatch 警报中添加终止操作

可以创建当达到一定阈值时自动终止 EC2 实例的警报 (只要该实例未启用终止保护)。例如，某个实例已经完成工作，您不再需要此实例而想将其终止。如果可能在之后使用该实例，则应该选择停止而不是终止。终止实例后，实例存储卷上的数据将丢失。有关为实例启用和禁用终止保护的信息，请参阅 更改实例终止保护。

创建终止空闲实例的警报（Amazon EC2 控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择所需实例，然后依次选择 Actions（操作）、Monitor and troubleshoot（监控和问题排查）、Manage CloudWatch alarms（管理 CloudWatch 警报）。

   或者，您可以在 Alarm status（警报状态）列中选择加号
   ( )。
4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）页面上，执行以下操作：
   a. 选择 Create an alarm（创建警报）。
   b. 要在警报触发时接收电子邮件，对于 Alarm notification（警报通知），选择现有的 Amazon SNS 主题。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。有关更多信息，请参阅Amazon Simple Notification Service 开发人员指南中的使用 Amazon SNS 进行应用程序对人 (A2P) 消息传送。
   c. 开启 Alarm action（警报操作），然后选择 Terminate（终止）。
   d. 对于 Group samples by（样本分组方式）和 Type of data to sample（要采样的数据类型），选择统计数据和指标。在此示例中，选择 Average（平均值）和 CPU utilization（CPU 利用率）。

e. 对于 Alarm When（警报时间）和 Percent（百分比），请指定指标阈值。在此示例中，请指定 => 和 10%。

f. 对于 Consecutive period（连续周期）和 Period（周期），请指定警报的评估周期。在此示例中，将 24 个连续周期指定为 1 小时。

g. Amazon CloudWatch 自动为您创建警报名称。要更改名称，请在 Alarm name（警报名称）中输入新名称。警报名称必须仅包含 ASCII 字符。

> **Note**
>
> 可以在创建警报前根据自己的要求调整警报配置，也可以在之后编辑配置。这包括指标、阈值、时长、操作和通知等设置。但是，警报创建后其名称无法再次编辑。

h. 选择创建。

在 Amazon CloudWatch 警报中添加重启操作

您可创建监控 Amazon EC2 实例并自动重启此实例的 Amazon CloudWatch 警报。在实例运行状况检查失败时，推荐重启警报操作（与恢复警报操作相反，该操作适合系统运行状况检查失败的情况）。实例重启相当于操作系统重启。在许多情况下，只需要几分钟时间即可重启您的实例。重启实例时，其仍驻留在相同的物理主机上，因此您的实例将保留其公有 DNS 名称、私有 IP 地址及其实例存储卷上的任何数据。

与停止并重新启动您的实例不同，重启实例不会启动新的实例计费周期（最低收取一分钟的费用）。重启实例后，实例存储卷上的数据会保留。重启后，必须将实例存储卷重新挂载到文件系统中。有关更多信息，请参阅 重启 Amazon EC2 实例。

> **Important**
>
> 为了避免重启操作与恢复操作之间的竞争情况，请避免为重启警报和恢复警报设置相同的评估周期数。我们建议您将重启警报设置为 3 个 1 分钟的评估期。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的评估警报。

创建重启实例的警报（Amazon EC2 控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。

3. 选择所需实例，然后依次选择 Actions（操作）、Monitor and troubleshoot（监控和问题排查）、Manage CloudWatch alarms（管理 CloudWatch 警报）。

   或者，您可以在 Alarm status（警报状态）列中选择加号

   ( )。

4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）页面上，执行以下操作：

   a. 选择 Create an alarm（创建警报）。

   b. 要在警报触发时接收电子邮件，对于 Alarm notification（警报通知），选择现有的 Amazon SNS 主题。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。有关更多信息，请参阅Amazon Simple Notification Service 开发人员指南中的使用 Amazon SNS 进行应用程序对人 (A2P) 消息传送。

   c. 开启 Alarm action（警报操作），然后选择 Reboot（重启）。

   d. 对于 Group samples by（样本分组方式）和 Type of data to sample（要采样的数据类型），选择统计数据和指标。在此示例中，选择 Average（平均值）和 Status check failed: instance（状态检查失败：实例）。

   e. 对于 Consecutive period（连续周期）和 Period（周期），请指定警报的评估周期。在此示例中，输入 3 个 1 分钟的连续周期。如果禁用 1 分钟，则必须启用详细监控，您也可以选择 5 分钟。

   f. Amazon CloudWatch 自动为您创建警报名称。要更改名称，请在 Alarm name（警报名称）中输入新名称。警报名称必须仅包含 ASCII 字符。

   g. 选择创建。

在 Amazon CloudWatch 警报中添加恢复操作

您可以创建 Amazon CloudWatch 警报来监控 Amazon EC2 实例。如果实例因需要 Amazon 参与才能修复的基础硬件故障或问题而受损，您可自动恢复实例。无法恢复终止的实例。恢复的实例与原始实例相同，包括实例 ID、私有 IP 地址、弹性 IP 地址以及所有实例元数据。

CloudWatch 会阻止您将恢复操作添加到位于不支持恢复操作的实例上的警报。

当 `StatusCheckFailed_System` 警报触发且恢复操作启动时，您在创建警报及相关恢复操作时所选择的 Amazon SNS 主题将向您发出通知。在实例恢复过程中，实例将在重启时迁移，并且内存中的所有数据都将丢失。当该过程完成后，会向您已配置警报的 SNS 主题发布信息。任何订阅此 SNS 主题的用户都将收到一封电子邮件通知，其中包括恢复尝试的状态以及任何进一步的指示。您将注意到，实例会在已恢复的实例上重启。

> **Note**
>
> 恢复操作仅适用于 `StatusCheckFailed_System`，而不能用于 `StatusCheckFailed_Instance`。

下列问题可能导致系统状态检查失败：

- 网络连接丢失
- 系统电源损耗
- 物理主机上的软件问题
- 物理主机上影响到网络连接状态的硬件问题

仅具有以下特性的实例支持恢复操作。有关更多信息，请参阅 实例自动恢复。

如果您的实例具有公有 IP 地址，它会在恢复后保留公有 IP 地址。

> **Important**
>
> 为了避免重启操作与恢复操作之间的竞争情况，请避免为重启警报和恢复警报设置相同的评估周期数。我们建议您将恢复警报设置为 2 个 1 分钟的评估期。有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的评估警报。

创建恢复实例的警报（Amazon EC2 控制台）

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择所需实例，然后依次选择 Actions（操作）、Monitor and troubleshoot（监控和问题排查）、Manage CloudWatch alarms（管理 CloudWatch 警报）。

   或者，您可以在 Alarm status（警报状态）列中选择加号
   ( )。
4. 在 Manage CloudWatch alarms（管理 CloudWatch 警报）页面上，执行以下操作：

   a. 选择 Create an alarm（创建警报）。

b. 要在警报触发时接收电子邮件，对于 Alarm notification（警报通知），选择现有的 Amazon SNS 主题。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。有关更多信息，请参阅Amazon Simple Notification Service 开发人员指南中的使用 Amazon SNS 进行应用程序对人 (A2P) 消息传送。

> **Note**
>
> 用户必须订阅指定的 SNS 主题才能在触发警报时收到电子邮件通知。执行实例自动恢复操作时，Amazon Web Services 账户根用户 始终会收到电子邮件通知，即使未指定 SNS 主题或者根用户未订阅指定 SNS 主题也是如此。

c. 开启 Alarm action（警报操作），然后选择 Recover（恢复）。

d. 对于 Group samples by（样本分组方式）和 Type of data to sample（要采样的数据类型），选择统计数据和指标。在此示例中，选择 Average（平均值）和 Status check failed: system（状态检查失败：系统）。

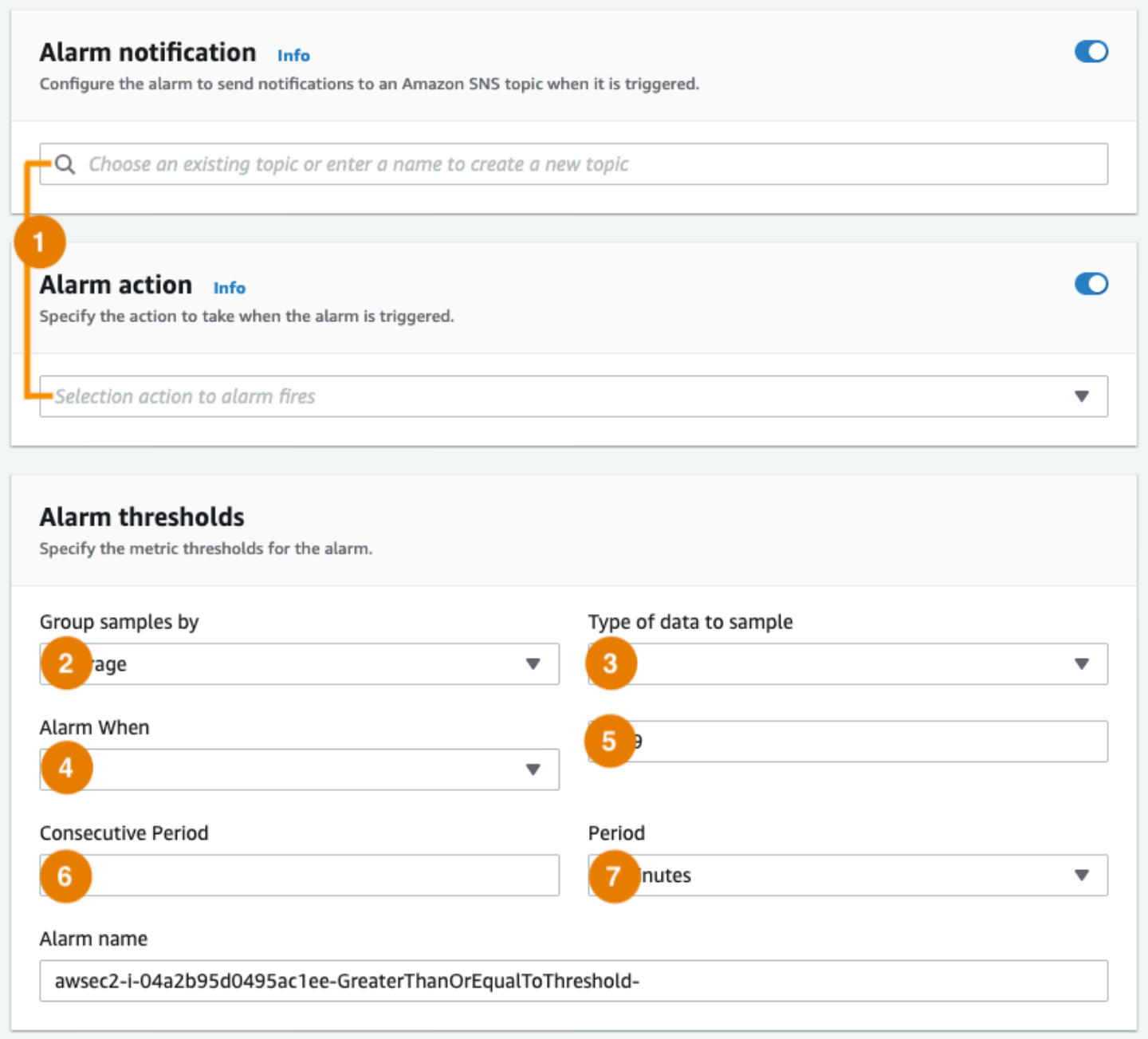
e. 对于 Consecutive period（连续周期）和 Period（周期），请指定警报的评估周期。在此示例中，输入 2 个 1 分钟的连续周期。如果禁用 1 分钟，则必须启用详细监控，您也可以选择 5 分钟。

f. Amazon CloudWatch 自动为您创建警报名称。要更改名称，请在 Alarm name（警报名称）中输入新名称。警报名称必须仅包含 ASCII 字符。

g. 选择创建。

Amazon CloudWatch 警报操作场景

可以使用 Amazon EC2 控制台创建当满足一定条件时停止或终止 Amazon EC2 实例的警报操作。在下方的控制台页面屏幕截图中，您设置了警报操作，我们对设置进行了编号。我们还对后续场景中的设置进行了编号，帮助您创建合适的操作。



场景 1：停止空闲开发和测试实例

创建当用于软件开发或测试的实例空闲达到至少 1 小时时停止该实例的警报。

设置	值
1	停止
2	最高

设置	值
3	CPU 利用率
4	<=
5	10%
6	1
7	1 小时

场景 2：停止空闲实例

创建一个当实例空闲达到 24 小时时停止该实例并发送电子邮件的警报。

设置	值
1	停止并发送电子邮件
2	平均值
3	CPU 利用率
4	<=
5	5%
6	24
7	1 小时

场景 3：发送有关出现异常高流量的 Web 服务器的电子邮件

创建一个当实例的出站网络流量每天超过 10 GB 时发送电子邮件的警报。

设置	值
1	电子邮件

设置	值
2	总计
3	网络输出
4	>
5	10GB
6	24
7	1 小时

场景 4：停止出现异常高流量的 Web 服务器

创建当出站流量超过每小时 1 GB 时停止实例并发送短消息 (SMS) 的警报。

设置	值
1	停止并发送短消息
2	总计
3	网络输出
4	>
5	1GB
6	1
7	1 小时

场景 5：停止受损的实例

创建当实例连续 3 次状态检查 (每隔 5 分钟执行一次) 皆为故障时将其停止的警报。

设置	值
1	停止
2	平均值
3	状态检查失败：系统
4	-
5	-
6	1
7	15 分钟

场景 6：在批处理作业完成时终止实例

创建当实例不再发送结果数据时终止运行批处理作业的实例的警报。

设置	值
1	终止
2	最高
3	网络输出
4	<=
5	100000 字节
6	1
7	5 分钟

使用 EventBridge 自动执行 Amazon EC2

您可以使用 Amazon EventBridge 自动执行您的 Amazon Web Services 服务 和自动响应系统事件，例如应用程序可用性问题或资源更改。Amazon 服务中的事件将近乎实时传输到 EventBridge。您可以创建规则来指示要关注的事件，以及在事件匹配规则时要执行的操作。可自动触发的操作包括：

- 调用 Amazon Lambda 函数
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 Amazon Step Functions 状态机
- 通知 Amazon SNS 主题
- 通知 Amazon SQS 队列

以下示例说明如何将 EventBridge 与 Amazon EC2 结合使用：

- 每当实例进入运行状态时激活 Lambda 函数。
- 在创建或修改 Amazon EBS 卷时通知 Amazon SNS 主题。
- 当另一个 Amazon 服务中发生特定事件时，使用 Amazon EC2 Run Command 向一个或多个 Amazon EC2 实例发送命令。

有关更多信息，请参阅 Amazon EventBridge 用户指南。

Amazon EC2 事件类型

Amazon EC2 支持以下事件类型：

- EC2 AMI 状态变化
- EC2 快速启动状态更改通知
- EC2 实例集错误
- EC2 实例集信息
- EC2 实例集实例更改
- EC2 实例集竞价型实例请求更改
- EC2 实例集状态更改

- EC2 实例再平衡建议
- EC2 实例状态更改通知
- EC2 竞价型实例集错误
- EC2 竞价型实例集信息
- EC2 竞价型实例集实例更改
- EC2 竞价型实例集竞价型实例请求更改
- EC2 竞价型实例集状态更改
- EC2 竞价型实例中断警告
- EC2 竞价型实例请求履行
- EC2 ODCR 利用不足通知

有关 Amazon EBS 支持的事件类型的信息，请参阅 Amazon EventBridge for Amazon EBS。

使用 Amazon CloudTrail 记录 Amazon EC2 API 调用

Amazon EC2 API 与 Amazon CloudTrail 集成，后者是一项提供用户、角色或 Amazon Web Services 服务所采取操作记录的服务。CloudTrail 将所有 Amazon EC2 API 调用作为事件捕获。捕获的调用包括控制台发出的调用。借助 CloudTrail 收集的信息，您可以确定向 Amazon EC2 API 发出的请求、发出请求的 IP 地址、发出请求的时间。

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户凭证还是用户凭证发出的。
- 请求是否代表 IAM Identity Center 用户发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 Amazon Web Services 服务 发出。

当您创建账户并可以自动访问 CloudTrail 事件历史记录时，CloudTrail 在您的 Amazon Web Services 账户 中处于活动状态。CloudTrail 事件历史记录提供对 Amazon Web Services 区域 中过去 90 天的已记录管理事件的可查看、可搜索、可下载和不可变记录。有关更多信息，请参见《Amazon CloudTrail 用户指南》的 使用 CloudTrail 事件历史记录。查看事件历史记录不会收取 CloudTrail 费用。

要持续记录您的 Amazon Web Services 账户 过去 90 天的事件，请创建跟踪或 CloudTrail Lake 事件数据存储。

CloudTrail 跟踪

通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service (Amazon S3) 存储桶。使用 Amazon Web Services 管理控制台 创建的所有跟踪均具有多区域属性。您可以通过使用 Amazon CLI 创建单区域或多区域跟踪。建议创建多区域跟踪，因为您可记录您账户中的所有 Amazon Web Services 区域 的活动。如果您创建单区域跟踪，则只能查看跟踪的 Amazon Web Services 区域 中记录的事件。有关跟踪的更多信息，请参阅《Amazon CloudTrail 用户指南》中的为您的 Amazon Web Services 账户 创建跟踪和为组织创建跟踪。

通过创建跟踪，您可以从 CloudTrail 免费向您的 Amazon S3 存储桶传送一份正在进行的管理事件的副本，但会收取 Amazon S3 存储费用。有关 CloudTrail 定价的更多信息，请参阅 Amazon CloudTrail 定价。有关 Amazon S3 定价的信息，请参阅 Amazon S3 定价。

CloudTrail Lake 事件数据存储

CloudTrail Lake 允许您对事件运行基于 SQL 的查询。CloudTrail Lake 可将基于行的 JSON 格式的现有事件转换为 Apache ORC 格式。ORC 是一种针对快速检索数据进行优化的列式存储格式。事件将被聚合到事件数据存储中，它是基于您通过应用高级事件选择器选择的条件的不可变的事件集合。应用于事件数据存储的选择器用于控制哪些事件持续存在并可供您查询。有关 CloudTrail Lake 的更多信息，请参阅《Amazon CloudTrail 用户指南》中的使用 Amazon CloudTrail Lake。

CloudTrail Lake 事件数据存储和查询会产生费用。创建事件数据存储时，您可以选择要用于事件数据存储的定价选项。定价选项决定了摄取和存储事件的成本，以及事件数据存储的默认和最长保留期。有关 CloudTrail 定价的更多信息，请参阅 Amazon CloudTrail 定价。

CloudTrail 中的 Amazon EC2 API 管理事件

管理事件提供对您的 Amazon Web Services 账户内资源所执行管理操作的相关信息。这些也称为控制层面操作。默认情况下，CloudTrail 会记录管理事件。

所有 Amazon EC2 API 操作都被记录为管理事件。有关记录到 CloudTrail 中的 API 操作的列表，请参阅《Amazon EC2 API 参考》。例如，对 RunInstances、DescribeInstances 和 StopInstances 操作的调用被记录为管理事件。

Amazon EC2 API 事件示例

一个事件表示一个来自任何源的请求，包括有关所请求的 API 操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此事件不会按任何特定顺序显示。

下面的日志文件记录显示，用户终止了一个实例。


```
{
   "Records":[
      {
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"user"
         },
         "eventTime":"2016-05-20T08:27:45Z",
         "eventSource":"ec2.amazonaws.com",
         "eventName":"TerminateInstances",
         "awsRegion":"us-west-2",
         "sourceIPAddress":"198.51.100.1",
         "userAgent":"aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
         "requestParameters":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d"
               }]
            }
         },
         "responseElements":{
            "instancesSet":{
               "items":[{
                  "instanceId":"i-1a2b3c4d",
                  "currentState":{
                     "code":32,
                     "name":"shutting-down"
                  },
                  "previousState":{
                     "code":16,
                     "name":"running"
                  }
               }]
            }
         },
         "requestID":"be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
         "eventID":"6e12345-2a4e-417c-aa78-7594fEXAMPLE",
```

```
        "eventType":"AwsApiCall",
        "recipientAccountId":"123456789012"
    }
  ]
}
```

有关 CloudTrail 记录内容的信息，请参阅《Amazon CloudTrail 用户指南》中的 CloudTrail 记录内容。

审核使用 EC2 Instance Connect 建立的连接

您可以使用 Amazon CloudTrail 审核通过 EC2 Instance Connect 连接到实例的用户。

使用 Amazon CloudTrail 控制台审计通过 EC2 Instance Connect 的 SSH 活动

1. 访问 https://console.aws.amazon.com/cloudtrail/，打开 CloudTrail 控制台。
2. 验证您是否位于正确的区域中。
3. 在导航窗格中，选择事件历史记录。
4. 对于筛选条件，请选择事件源，然后选择 ec2-instance-connect.amazonaws.com。
5. （可选）对于时间范围，请选择一个时间范围。
6. 选择刷新事件图标。
7. 该页面显示与 SendSSHPublicKey API 调用对应的事件。使用箭头展开一个事件以查看其他详细信息，例如，用于建立 SSH 连接的用户名和 Amazon 访问密钥以及源 IP 地址。
8. 要以 JSON 格式显示完整事件信息，请选择查看事件。requestParameters 字段包含用于建立 SSH 连接的目标实例 ID、操作系统用户名和公有密钥。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
        "userName": "IAM-friendly-name",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-09-21T21:37:58Z"}
        }
```

```
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey ",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNO01234567890EXAMPLE"
        }
     },
    "responseElements": null,
    "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",
    "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",
    "eventType": "AwsApiCall",
    "recipientAccountId": "0987654321"
}
```

如果已将 Amazon 账户配置为在 S3 存储桶中收集 CloudTrail 事件，您可以按编程方式下载和审计该信息。有关更多信息，请参阅《Amazon CloudTrail 用户指南》中的获取和查看 CloudTrail 日志文件。

使用 CloudWatch Application Insights 监控 .NET 和 SQL Server 应用程序

CloudWatch Application Insights 可帮助您监控使用 Amazon EC2 实例以及其他 Amazon 应用程序资源的 .NET 和 SQL Server 应用程序。该服务在应用程序资源和技术堆栈 [例如：Microsoft SQL Server 数据库、Web（IIS）和应用程序服务器、操作系统、负载均衡器以及队列] 中，指定并设置关键指标日志和警报。它持续监控指标和日志，以检测异常情况和错误并将它们关联起来。在检测到错误和异常情况时，Application Insights 会生成可用于设置通知或执行操作的事件。为了帮助进行故障排除，它为检测到的问题创建自动化控制面板，其中包括关联的指标异常情况和日志错误以及指出潜在根本原因的其他信息。自动化控制面板帮助您快速采取修复措施，以将应用程序保持正常运行并防止对应用程序的最终用户造成影响。

为检测到的问题提供的相关信息：

- 问题的简短摘要

- 问题的开始时间和日期
- 问题严重性：高/中/低
- 检测到的问题的状态：正在进行/已解决
- 信息：自动生成有关检测到的问题和可能的根本原因的信息
- 信息反馈：您为有关适用于 .NET 和 SQL Server 的 CloudWatch Application Insights 生成的信息是否有用提供的反馈
- 相关的观察结果：与各种应用程序组件中的问题相关的指标异常情况和相关日志错误片段的详细视图

反馈

您可以指定为检测到的问题自动生成的信息是否有用以提供反馈。将使用您对这些信息的反馈以及应用程序诊断（指标异常情况和日志异常情况）改进将来对类似问题的检测。

有关更多信息，请参阅 Amazon CloudWatch 用户指南中的 CloudWatch Application Insights 文档。

了解账单和使用情况报告中的 Amazon EC2 代码

当您使用 Amazon EC2 时，我们会在您的 Amazon 账单和使用情况报告中包含相关代码。查看这些代码有助于您了解 Amazon EC2 的成本和使用模式。跟踪和管理您的支出对于优化成本至关重要。

下表描述您的账单和使用情况报告中出现的 Amazon EC2 代码。有关账单和使用情况报告中使用的区域代码列表，请参阅 Amazon Region billing codes。

以下各项的账单代码：

- 实例
- 裸机实例
- 专属主机
- Dedicated Instances
- EBS 优化
- 容量预留

相关资源

- the section called “计费和购买选项”
- the section called “了解 AMI 账单”

- Amazon EC2 定价

实例

代码	描述	单位
region-BoxUsage :*instance-type*	按需型实例的运行时间。	Hours
region-HostBoxUsage :*instance-type*	专属主机上实例的运行时间。	Hours
region-SpotUsage :*instance-type*	竞价型实例的运行时间。	Hours

裸机实例

代码	描述	单位
region-BoxUsage :*instance-family* .metal	裸机按需型实例的运行时间。	Hours
region-HostBoxUsage :*instance-family* .metal	专属主机上裸机实例的运行时间。	Hours
region-SpotUsage :*instance-family* .metal	裸机竞价型实例的运行时间。	Hours

专属主机

代码	描述	单位
region-HostUsage :*host-type*	预置专属主机的时间。	Hours
region-ReservedHostUsage :*host-type*	应用专属主机预留的时间。	Hours

Dedicated Instances

代码	描述	单位
region-DedicatedUsage :*instance-type*	专用实例的运行时间。	小时 + 每个区域的费用

EBS 优化

代码	描述	单位
region-EBSOptimized :*instance-type*	启用 EBS 优化的时间。	Hours

容量预留

代码	描述	单位
region-Reservation :*instance-type*	容量预留的预留实例时间。	Hours
region-UnusedBox :*instance-type*	容量预留的未使用预留实例时间。	Hours
region-DedicatedRes :*instance-type*	专用容量预留的预留实例时间。	Hours
region-UnusedDed :*instance-type*	专用容量预留的未使用预留实例时间。	Hours

创建 EC2 实例的清单

Amazon EC2 提供按需、可扩展的计算容量。EC2 实例是 Amazon 网络中运行的虚拟服务器。

下表描述 EC2 实例的主要特征。系统管理员可以使用本指南中提供的指导来获取有关这些特征的信息，并使用它来配置在本地或另一个云提供商网络中运行的功能等效服务器。

特征	描述
地理位置	Amazon EC2 托管在全球所有 Amazon 区域。您可以为实例选择靠近客户的位置。您可以在多个位置中启动实例。
硬件规格	Amazon EC2 提供针对不同使用案例优化的各种不同的实例类型。您为实例选择的实例类型决定了其计算、内存、网络和存储资源。

特征	描述
映像	亚马逊机器映像（AMI）包含启动时要在实例上安装的软件。这包括操作系统、软件包和自定义配置。
IP 地址和 DNS 主机名	实例接收私有 IP 地址和私有 DNS 主机名。如果您为实例配置了公有 IP 地址，它还会收到公有 DNS 主机名。
安全组规则	与您的实例关联的安全组规则决定允许哪些入站流量和出站流量。
用户数据	用户数据在启动时可供实例使用。它包含 shell 脚本（Linux）或 PowerShell 脚本（Windows）。

地理位置

Amazon EC2 在全球每个 Amazon 区域均可用。每个区域都是一个单独的地理区域。为服务器选择靠近大多数用户的区域，可以降低网络延迟。

您可以使用 Amazon EC2 全球视图列出所有区域的 Amazon EC2 实例。使用 Amazon Web Services 管理控制台 和 API 时，您必须单独列出每个区域的实例。

为什么这很重要

确定实例所在的位置后，您可以根据需要决定是在相同位置还是不同位置部署功能等效的服务器。

获取所有区域中 EC2 实例的摘要

1. 通过以下网址打开 Amazon EC2 全局视图控制台：https://console.aws.amazon.com/ec2globalview/home。
2. 在区域资源管理器选项卡的摘要下，查看实例的资源计数，其中包括实例数量和区域数量。单击带下划线的文本，查看实例计数在各区域的分布情况。
3. 在全球搜索选项卡上，选择客户端筛选器资源类型 = 实例。您可通过指定区域或标签进一步筛选结果。

使用 Amazon CLI 获取某个区域中的 EC2 实例数量

使用以下 describe-instances 命令统计指定区域中的实例数。您必须在您有实例的每个区域运行此命令。

```
aws ec2 describe-instances \
    --region us-east-2 \
    --query "length(Reservations[*].Instances[])"
```

下面是示例输出。

```
27
```

要获取实例 ID 列表而不是实例计数，请改用以下 --query 参数。

```
--query "Reservations[*].Instances[].InstanceId"
```

硬件规格

您为 EC2 实例指定的实例类型决定了其可用的计算、内存、存储和联网资源。每个实例系列以不同方式兼顾计算、内存、网络和存储资源。有关更多信息，请参阅 Amazon EC2 实例类型指南。

为什么这些很重要

确定实例的硬件规格后，您可以决定功能等效的服务器所需的最低规格。

获取您的实例所使用实例类型的摘要

使用以下 describe-instances 命令。sed 命令删除会带有方括号的行和空行。您必须在您有实例的每个区域运行此命令。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].InstanceType" | sed 's/[][]//g;/^$/d' | sort |
 uniq -c | sort -nr
```

下面是示例输出。

```
  20    "c6i.4xlarge",
   5    "t2.micro",
```

```
    2    "g6e.2xlarge",
```

获取有关特定实例类型的信息

使用以下 describe-instance-types 命令描述实例类型。--query 参数在输出中仅包括指定的字段。这些是基本特征。您可以包括所需的任何其他字段，也可以参阅《Amazon EC2 Instance Types Guide》中的 Amazon EC2 instance type specifications。

```
aws ec2 describe-instance-types \
    --instance-types c6i.4xlarge t2.micro g6e.2xlarge \
    --query "InstanceTypes[*].
[InstanceType,VCpuInfo.DefaultVCpus,MemoryInfo.SizeInMiB,NetworkInfo.NetworkPerformance]"
 \
    --output table
```

下面是示例输出。这些列包括实例类型、vCPU、内存（GiB）和网络带宽（Gbps）。

```
-----------------------------------------------------
|                DescribeInstanceTypes                |
+--------------+-----+--------+----------------------+
|  t2.micro    |  1  |  1024  |  Low to Moderate     |
|  c6i.4xlarge |  16 |  32768 |  Up to 12.5 Gigabit  |
|  g6e.2xlarge |  8  |  65536 |  Up to 20 Gigabit    |
+--------------+-----+--------+----------------------+
```

映像

亚马逊机器映像（AMI）提供启动时安装在实例上的软件，例如操作系统、服务、开发工具和应用程序。您可以将 AMI 导出为其他格式以供其他服务器使用。

为什么这些很重要

确定实例的 AMI 后，您可以规划功能等效的服务器所需的映像。可以将您的 AMI 导出为可以在其他地方使用的格式。或者，您可能需要连接到从每个 AMI 启动的实例，确认已安装的内容及其配置方式，并确保拥有所需所有内容的副本。

获取实例的 AMI

使用以下 describe-instances 命令。sed 命令删除会带有方括号的行和空行。您必须在您有实例的每个区域运行此命令。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].ImageId" | sed 's/[][]//g;/^$/d' | sort | uniq
 -c | sort -nr
```

下面是示例输出。

```
20     "ami-0a70b9d193ae8a79",
 5     "ami-07d9cf938edb0739b",
 2     "ami-09245d5773578a1d6",
```

获取有关特定 AMI 的信息

使用以下 describe-images 命令。--query 参数在输出中仅包括指定的字段。您可以包括所需的任何其他字段。

```
aws ec2 describe-images \
    --image-id ami-0a70b9d193ae8a79 ami-07d9cf938edb0739b ami-09245d5773578a1d6 \
    --query "Images[*].
{ID:ImageId,CreateDate:CreationDate,Name:Name,Description:Description}"
```

下面是示例输出。

```
[
    {
        "ID": "ami-0a70b9d193ae8a799",
        "CreateDate": "2025-03-01T02:22:41.000Z",
        "Name": "web-server-3.1",
        "Description": "Image for web servers"
    },
    {
        "ID": "ami-07d9cf938edb0739b",
        "CreateDate": "2025-02-01T23:59:03.000Z",
        "Name": "awesome-application-11.5",
        "Description": "Image for Awesome Application"
    },
    {
        "ID": "ami-09245d5773578a1d6",
        "CreateDate": "2025-01-31T02:22:41.000Z",
        "Name": "monitoring-4.2",
        "Description": "Monitoring software"
    }
```

```
]
```

从 AMI 创建 VM 文件

使用 export-image 命令。支持的映像格式包括 VHD（与 Citrix Xen 和 Microsoft Hyper-V 兼容）、VMDK（与 VMware ESX 和 VMware vSphere 兼容）以及原始格式 raw（与 KVM 和 Xen 虚拟机监控程序兼容）。有关 VM Import/Export 要求和限制的信息，请参阅 VM Import/Export 要求。

IP 地址和 DNS 主机名

您的用户使用其公有 DNS 主机名通过互联网连接到 EC2 实例。EC2 实例的公有 DNS 主机名解析为其公有 IP 地址。EC2 实例可以使用其私有 IP 地址相互通信。

为什么这些很重要

IP 地址允许设备通过本地网络或互联网相互通信。通过记下实例的 IP 地址和 DNS 名称，可帮助确保功能等效的服务器可以与实例相同的客户端或设备进行通信。例如，您可以更新负载均衡器配置或更新您为 DNS 提供商创建的 DNS 记录。

获取实例的 IP 地址和 DNS 主机名

使用以下 describe-instances 命令。您必须在您有实例的每个区域运行此命令。`--query` 参数在输出中仅包括指定的字段。您可以包括所需的任何其他字段。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].[InstanceId,PrivateIpAddress,PublicDnsName]" \
    --output table
```

下面是示例输出。这些列包括实例 ID、私有 IPv4 地址和公有 IPv4 DNS 名称。

```
----------------------------------------------------------------------------------
|                                DescribeInstances
 |
+---------------------+------------+-------------------------------------------------
+
| i-0bac57d7472c89bac | 10.0.2.139 | ec2-192-0-2-142.us-east-2.compute.amazonaws.com
 |
| i-0fa8b7678975a3fff | 10.0.14.85 | ec2-198-51-100-56.us-east-2.compute.amazonaws.com
 |
  ...
  ...
```

```
| i-08fd74f3f1595fdbd | 10.0.1.241 | ec2-203.0.113.13.us-east-2.compute.amazonaws.com
 |
+---------------------+------------+-------------------------------------------------
+
```

安全组规则

安全组起到虚拟防火墙的作用。在将安全组与 EC2 实例关联后，其规则会允许该 EC2 实例通过特定端口和协议的入站和出站流量。

为什么这些很重要

确定允许到达服务器的入站流量和允许离开服务器的出站流量之后，您可以规划功能等效的服务器所需的防火墙规则。

获取实例的安全组

使用以下 describe-instances 命令。sed 命令删除会带有方括号的行和空行。您必须在您有实例的每个区域运行此命令。

```
aws ec2 describe-instances \
    --query "Reservations[*].Instances[].SecurityGroups[].GroupId" | sed 's/[][]//g;/^
$/d' | sort | uniq -c | sort -nr
```

下面是示例输出。

```
27     "sg-01dd3383691d02f42",
10     "sg-08c77cc910c0b3b2c",
 2     "sg-00f4e409629f1a42d",
```

获取安全组的入站规则

使用以下 describe-security-group-rules 命令显示 `IsEgress` 为 `false` 的指定安全组的规则。

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-01dd3383691d02f42 \
    --query 'SecurityGroupRules[?IsEgress==`false`]'
```

下面是示例输出。第一条规则是默认入站规则，其允许分配给此安全组的所有资源的入站流量。另一条规则允许来自指定前缀列表中 IP 地址的 SSH 流量。要描述前缀列表中的 CIDR 块，请使用 describe-prefix-lists 命令。

```
[
    {
        "SecurityGroupRuleId": "sgr-06c8b42574a91db1b",
        "GroupId": "sg-01dd3383691d02f42",
        "GroupOwnerId": "123456789012",
        "IsEgress": false,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "ReferencedGroupInfo": {
            "GroupId": "sg-01dd3383691d02f42",
            "UserId": "123456789012"
        },
        "Tags": [],
        "SecurityGroupRuleArn": "arn:aws:ec2:us-west-2:123456789012:security-group-
rule/sgr-06c8b42574a91db1b"
    },
    {
        "SecurityGroupRuleId": "sgr-0886a5d46afcd1758",
        "GroupId": "sg-01dd3383691d02f42",
        "GroupOwnerId": "123456789012",
        "IsEgress": false,
        "IpProtocol": "tcp",
        "FromPort": 22,
        "ToPort": 22,
        "PrefixListId": "pl-f8a6439125e7bf465",
        "Tags": [],
        "SecurityGroupRuleArn": "arn:aws:ec2:us-east-2:123456789012:security-group-
rule/sgr-0886a5d46afcd1758"
    }
]
```

获取安全组的出站规则

使用以下 describe-security-group-rules 命令显示 IsEgress 为 true 的指定安全组的规则。

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-01dd3383691d02f42 \
    --query 'SecurityGroupRules[?IsEgress==`true`]'
```

下面是示例输出。其仅包含默认出站规则，允许所有出站 IPv4 流量。

```
[
```

```
    {
        "SecurityGroupRuleId": "sgr-048f09a719247dce7",
        "GroupId": "sg-01dd3383691d02f42",
        "GroupOwnerId": "123456789012",
        "IsEgress": true,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "CidrIpv4": "0.0.0.0/0",
        "Tags": [],
        "SecurityGroupRuleArn": "arn:aws:ec2:us-east-2:123456789012:security-group-
rule/sgr-048f09a719247dce7"
    }
]
```

用户数据

在启动 EC2 实例时，您可以使用用户数据将 shell 脚本传递给该实例。请注意，用户数据采用 base64 编码，因此您需要解码用户数据才能读取脚本。

为什么这很重要

如果您在启动时运行命令作为设置实例的一部分，则在设置功能等效的服务器时，可能需要执行相同的任务。

查看实例的解码用户数据

使用以下 describe-instance-attribute 命令。base64 命令会解码用户数据。

```
aws ec2 describe-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --attribute userData \
    --output text \
    --query "UserData.Value" | base64 --decode
```

下面是示例输出。

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

相关资源

以下是 EC2 实例的其他特征：

- 密钥对
- 存储
- 标签

您可以验证自己是使用以下方式启动 EC2 实例，还是在 EC2 实例之间分配流量：

- 自动扩缩组
- EC2 Fleet
- Elastic Load Balancing

排查 Amazon EC2 实例的问题

以下程序和技巧可以帮助您对 Amazon EC2 实例问题进行排查。

问题

- 排查 Amazon EC2 实例启动问题
- 排查 Amazon EC2 实例的停止问题
- 排查 Amazon EC2 实例的终止问题
- 排查 Amazon EC2 实例无法访问的问题
- 排查 Amazon EC2 Linux 实例的连接问题
- 通过失败状态检查来排查 Amazon EC2 Linux 实例问题
- 排查从错误的卷启动 Amazon EC2 Linux 实例的问题
- 排查连接到 Amazon EC2 Windows 实例时遇到的问题
- 排查 Amazon EC2 Windows 实例的启动问题
- 排查 Amazon EC2 Windows 实例问题
- 通过网络对 Windows 实例进行内核调试
- 重置 Amazon EC2 Windows 实例的 Windows 管理员密码
- 排查 Amazon EC2 Windows 实例的 Sysprep 问题
- 使用 EC2Rescue 对受损的 Amazon EC2 Linux 实例进行问题排查
- 使用 EC2Rescue 对受损的 Amazon EC2 Windows 实例进行问题排查
- 适用于实例的 EC2 Serial Console
- 发送诊断中断来调试无法访问的 Amazon EC2 实例

排查 Amazon EC2 实例启动问题

以下问题排查技巧有助于解决启动 Amazon EC2 实例时出现的问题。

启动问题

- 设备名称无效
- 超出实例限制

- 实例容量不足
- 当前不支持请求的配置。请查看文档以了解支持的配置。
- 实例立即终止
- 权限不足
- Windows 启动后，CPU 使用率短时增高（仅限 Windows 实例）
- 启动启用 IMDSv1 的实例时失败

设备名称无效

说明

在尝试启动新实例时，您将收到 `Invalid device name device_name` 错误。

原因

如果您在尝试启动实例时遇到此错误，则请求中为一个或多个卷指定的设备名称无效。可能的原因包括：

- 所选 AMI 目前可能在使用该设备名称。
- 该设备名称可能是为根卷保留的。
- 该设备名称可能用于请求中的另一个卷。
- 该设备名称可能对操作系统无效。

解决方案

要解决问题，请执行以下操作：

- 确保所选 AMI 未使用该设备名称。运行以下命令，查看 AMI 使用的设备名称。

```
aws ec2 describe-images --image-id ami-0abcdef1234567890 --query
 'Images[*].BlockDeviceMappings[].DeviceName'
```

- 确保没有使用为根卷保留的设备名称。有关更多信息，请参阅 可用设备名称。
- 确保在请求中指定的每个卷都有唯一的设备名称。
- 确保指定的设备名称采用正确格式。有关更多信息，请参阅 可用设备名称。

超出实例限制

说明

在尝试启动新实例或重新启动已停止的实例时，您将收到 `InstanceLimitExceeded` 错误。

原因

在尝试启动新实例或重新启动已停止的实例时，如果您已达到可在区域中启动的实例的数目限制，则将收到 `InstanceLimitExceeded` 错误。在创建 Amazon 账户时，我们根据区域设置可运行的实例数的默认限制。

解决方案

您可以根据区域请求提高实例限制。有关更多信息，请参阅 Amazon EC2 Service Quotas。

实例容量不足

说明

在尝试启动新实例或重新启动已停止的实例时，您将收到 `InsufficientInstanceCapacity` 错误。

原因

如果您在尝试启动实例或重新启动已停止的实例时收到此错误，则表示 Amazon 当前没有足够的可用按需容量来服务您的请求。

解决方案

要解决该问题，请尝试以下操作：

- 等待几分钟，然后再次提交您的请求；容量可能经常转移。
- 提交减少了实例数的新请求。例如，如果您要提交 1 个启动包含 15 个实例的请求，请改为尝试提交 3 个包含 5 个实例的请求或 15 个包含 1 个实例的请求。
- 如果您要启动实例，请提交新请求，无需指定可用区。
- 如果您要启动实例，请使用其他实例类型 (可在后期调整大小) 提交新请求。有关更多信息，请参阅 Amazon EC2 实例类型更改。

- 如果您将实例启动到集群置放群组中，则会获得容量不足错误。

当前不支持请求的配置。请查看文档以了解支持的配置。

说明

当您尝试启动新实例时会出现 Unsupported 错误，因为不支持实例配置。

原因

错误消息提供了更多详细信息。例如，指定的区域或可用区可能不支持实例类型或实例购买选项。

解决方案

尝试其他实例配置。要搜索符合您要求的实例类型，请参阅 查找 Amazon EC2 实例类型。

实例立即终止

说明

您的实例会从 pending 状态变为 terminated 状态。

原因

下面是实例可能立即终止的一些原因：

- 您已超出 EBS 卷限制。有关更多信息，请参阅 Amazon EC2 实例的 Amazon EBS 卷限制。
- EBS 快照损坏。
- 根 EBS 卷已加密，但您无权访问用于解密的 KMS 密钥。
- 在块储存设备映射中为 AMI 指定的快照已加密，但您无权访问用于解密的 KMS 密钥，或者您无权访问 KMS 密钥 来加密还原的卷。
- 您用来启动实例的由 Amazon S3 支持的 AMI 缺少必需部分（一个 image.part.xx 文件）

有关更多信息，请通过以下某种方法了解终止原因。

使用 Amazon EC2 控制台了解终止原因

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 Instances，然后选择实例。
3. 在第一个选项卡上，在状态转换原因旁边查看原因。

使用 Amazon CLI 控制台了解终止原因

1. 使用 describe-instances 命令并指定实例 ID。

```
aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

2. 检查命令返回的 JSON 响应，并记下 StateReason 响应元素中的值。

   下面的代码块显示了 StateReason 响应元素的示例。

```
"StateReason": {
  "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",
  "Code": "Server.InternalError"
},
```

使用 Amazon CloudTrail 了解终止原因

有关更多信息，请参阅 Amazon CloudTrail 用户指南中的使用 CloudTrail 事件历史记录查看事件。

解决方案

根据找到的终止原因，执行以下某项操作：

- **Client.VolumeLimitExceeded: Volume limit exceeded** - 删除未使用的卷。您可以提交请求，要求提高卷限制。
- **Client.InternalError: Client error on launch** – 确保您具有访问用于解密和加密卷的 Amazon KMS keys 所需的权限。有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的在 Amazon KMS 中使用密钥策略。

权限不足

说明

在尝试启动新实例时出现 "*errorMessage*": "You are not authorized to perform this operation." 错误，并且启动失败。

原因

如果您在尝试启动实例时出现此错误，则说明您没有启动该实例所需的 IAM 权限。

可能缺少以下权限：

- `ec2:RunInstances`
- `iam:PassRole`

也可能缺少其他权限。有关启动实例所需的权限列表，请参阅 示例：使用 EC2 启动实例向导 和 启动实例 (RunInstances) 下的示例 IAM policy。

解决方案

要解决问题，请执行以下操作：

- 如果您以 IAM 用户身份发出请求，请验证您拥有以下权限：
  - 对通配符资源 ("*") 执行 `ec2:RunInstances` 操作的许可
  - `iam:PassRole` 对匹配角色 ARN (例如 `arn:aws:iam::999999999999:role/ExampleRoleName`) 的资源执行操作的权限。
- 如果您没有上述权限，请编辑与 IAM 角色或用户关联的 IAM policy 以添加缺少的必需权限。

如果问题并未解决且仍然出现启动失败错误，您可以解码错误中包含的授权失败消息。解码后的消息包含 IAM policy 中缺少的权限。有关更多消息，请参阅在 EC2 实例启动期间收到“UnauthorizedOperation”错误消息后，如何解码授权失败消息？

Windows 启动后，CPU 使用率短时增高（仅限 Windows 实例）

> **Note**
>
> 此故障排除技巧仅适用于 Windows 实例。

如果将 Windows Update 设置为 Check for updates but let me choose whether to download and install them (检查更新，但允许我选择是否下载并安装它们)（默认实例设置），则此检查可能会在实例上消耗 50 - 99% 的 CPU。如果此 CPU 消耗会导致您的应用程序出现问题，您可手动在 Control Panel 中更改 Windows Update 设置或者使用 Amazon EC2 用户数据字段中的以下脚本：

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v
 AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

运行此脚本时，为 /d 指定一个值。默认值为 3。可能的值包括：

1. 从未检查更新
2. 检查更新，但允许我选择是否下载并安装它们
3. 下载更新，但允许我选择是否安装它们
4. 自动安装更新

在为您的实例修改用户数据后，可运行它。有关更多信息，请参阅启动时在 Windows 实例上运行命令。

启动启用 IMDSv1 的实例时失败

说明

您会获得 UnsupportedOperation 异常并收到以下消息：

```
You can't launch instances with IMDSv1 because httpTokensEnforced
is enabled for this account. Either launch the instance with
httpTokens=required or contact your account owner to disable
httpTokensEnforced using the ModifyInstanceMetadataDefaults API or the
account settings in the EC2 console.
```

原因

在 EC2 账户设置或 Amazon 组织声明性策略强制使用 IMDSv2 (`httpTokensEnforced = enabled`) 的账户中，尝试启动新实例并启用 IMDSv1 (`httpTokens = optional`) 时，将会引发此错误。

解决方案

如果您已准备好仅使用 IMDSv2，请在禁用 IMDSv1 的情况下启动实例 (`httpTokens = required`)。要检查准备就绪，请参阅转换为使用 实例元数据服务版本 2。

如果仍需要在新实例或现有实例上支持 IMDSv1，则需要在该区域为该账户禁用 IMDSv2 强制使用。要禁用 IMDSv2 强制使用，请将 `HttpTokensEnforced` 设置为 `disabled`。有关更多信息，请参阅

《Amazon EC2 API 参考》中的 ModifyInstanceMetadataDefaults。如果您更喜欢使用控制台来配置此设置，请参阅在账户级别强制使用 IMDSv2。

排查 Amazon EC2 实例的停止问题

如果您的 Amazon EBS 支持的实例卡在 `stopping` 状态，则底层主机可能存在问题。

要解决此问题，请按照下列步骤操作：

1. 强制停止实例

   使用 Amazon EC2 控制台或 Amazon CLI 强制停止实例。有关步骤，请参阅强制停止实例。

   实例将首先尝试正常关闭，其中包括刷新文件系统缓存和元数据（虽然您可以选择绕过正常关闭）。如果在超时期限内无法完成正常关闭，则实例将强制关闭，而不会刷新文件系统缓存和元数据。
2. 强制停止后

   执行文件系统检查和修复过程。

   > **Important**
   >
   > 执行这些过程至关重要，因为强制的停止会阻止刷新文件系统缓存和元数据。
3. 如果强制停止失败

   如果 10 分钟后实例仍未停止，请执行以下操作：

   a. 请在 Amazon Web Services re:Post 上发布求助请求。为了帮助加快解决问题，请提供实例 ID 并描述已采取的步骤。
   b. 此外，如果您有支持计划，则可在支持中心创建技术支持案例。
   c. 在等待帮助时，您可以根据需要创建替代实例。有关步骤，请参阅（可选）创建替代实例。

当实例处于 `stopping` 状态或处于除 `running` 外的任何其他状态时，不会收取任何实例使用费用。只有当实例处于 `running` 状态时，您使用实例时才需要付费。

内容

- 强制停止实例

- （可选）创建替代实例

强制停止实例

您可以强制停止实例。如果在 10 分钟后，实例未停止，请在 Amazon Web Services re:Post 上寻求帮助。为了帮助加快解决问题，请提供实例 ID 并描述已采取的步骤。此外，如果您有支持计划，则可在支持中心创建技术支持案例。

> **Note**
>
> 使用控制台时，只能在实例处于 `stopping` 状态时强制停止实例。使用 Amazon CLI 时，可以在实例处于 `pending`、`running` 或 `stopping` 状态时强制停止实例。

Console

强制停止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择卡住实例。
3. 选择实例状态、强制停止实例。

   请注意，只有当您的实例处于 `stopping` 状态时，Force stop instance（强制停止实例）才可用。如果您的实例处于另一状态（除 `shutting-down` 和 `terminated` 之外的状态），您可以使用 Amazon CLI 强制停止您的实例。
4. （可选）要绕过强制停止期间操作系统的正常关闭，请选中跳过操作系统关闭复选框。
5. 选择强制停止。

Amazon CLI

强制停止实例

使用 stop-instances 命令和 `--force` 选项。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

要绕过强制停止期间操作系统的正常关闭，请包括 `--skip-os-shutdown` 选项。

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

强制停止实例

使用 Stop-EC2Instance cmdlet 并将 `-Enforce` 设置为 `true`。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

要绕过强制停止期间操作系统的正常关闭，请包括 `-SkipOsShutdown $true`。

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

（可选）创建替代实例

在等待 Amazon Web Services re:Post 或支持中心的帮助时，您可以根据需要创建替代实例。创建卡住实例的 AMI，再使用新的 AMI 启动新实例。

> **Important**
>
> 如果卡住的实例仅生成系统状态检查，则可以创建替换实例，因为实例状态检查将导致 AMI 复制损坏操作系统的精确副本。确认状态消息后，创建 AMI 并使用新的 AMI 启动新实例。

Console

创建替代实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在导航窗格中，选择 Instances，然后选择卡住实例。
3. 依次选择操作、映像和模板和创建映像。
4. 在 Create image (创建映像) 页面上，执行以下操作：

   a. 输入 AMI 的名称和说明。

   b. 清除重启实例。

   c. 选择创建映像。

   有关更多信息，请参阅 the section called “从实例创建 AMI”。
5. 从 AMI 启动新实例，验证新实例是否正常运行。
6. 选择卡住的实例，然后依次选择操作、实例状态、终止（删除）实例。如果该实例也因卡住而终止，则 Amazon EC2 会自动强制其在几个小时内终止。

如果无法按上一步骤所述从该实例创建 AMI，则可以设置替代实例，如下所示：

(替代) 使用控制台创建替代实例

1. 选择实例并选择 Description、Block devices。选择每个卷并记下其卷 ID。请务必注意哪个卷是根卷。
2. 在导航窗格中，选择 Volumes。选择该实例的各个卷，然后依次选择 Actions、Create Snapshot。
3. 在导航窗格中，选择快照。选择您刚刚创建的快照，然后依次选择 Actions、Create Volume。
4. 使用与粘滞的实例相同的操作系统启动实例。注意其根卷的卷 ID 和设备名称。
5. 在导航窗格中，选择Instances（实例），选择刚才启动的实例，然后依次选择 Instance state（实例状态）、Stop Instance（停止实例）。
6. 在导航窗格中，选择 Volumes，选择已停止实例的根卷，然后依次选择 Actions、Detach Volume。
7. 选择您从卡住的实例创建的根卷，依次选择 Actions、Attach Volume，然后将其附加到新实例以作为其根卷（使用记下的设备名称）。将任何其他非根卷附加到该实例。
8. 在导航窗格中，选择 Instances，然后选择替代实例。依次选择实例状态、启动实例。验证该实例是否正常运行。
9. 选择卡住的实例，然后依次选择实例状态、终止（删除）实例。如果该实例也因卡住而终止，则 Amazon EC2 会自动强制其在几个小时内终止。

Amazon CLI

创建替代实例

1. 使用 create-image 命令和 --no-reboot 选项，根据卡滞的实例创建一个 AMI。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-replacement-ami" \
    --description "AMI for replacement instance" \
    --no-reboot
```

2. 使用 run-instances 命令，从您刚刚创建的 AMI 启动一个新实例。
3. 验证新实例是否正常运行。
4. （可选）使用 terminate-instances 命令，终止卡滞的实例。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

创建替代实例

1. 使用 New-EC2Image cmdlet 并将 -NoReboot 设置为 true，从卡滞的实例创建一个 AMI。

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-replacement-ami" `
    -Description "AMI for replacement instance" `
    -NoReboot $true
```

2. 使用 New-EC2Instance cmdlet，从您刚刚创建的 AMI 启动一个新实例。
3. 验证新实例是否正常运行。
4. （可选）使用 Remove-EC2Instance cmdlet，终止卡滞的实例。

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

排查 Amazon EC2 实例的终止问题

关闭或删除实例被称为实例终止。以下信息可以帮助您排查终止实例时遇到的问题。

当实例未处于 `running` 状态时，不会向您收取任何实例使用费用。换言之，当您终止实例时，一旦实例的状态变为 `shutting-down`，就不再产生与该实例相关的费用。

实例立即终止

有多个问题可能会导致您的实例在启动后立即终止。请参阅实例立即终止了解更多信息。

延迟的实例终止

如果您的实例处于`shutting-down`状态超过几分钟，这可能是因为以下原因：

- 该实例正在运行关闭脚本。
- 底层主机有问题。

实例处于`shutting-down`状态已有几小时之后，Amazon EC2 会视之为卡住，并会强制终止该实例。

要自行解决卡住的实例，请执行以下操作：

1. 强制终止实例

   使用 Amazon EC2 控制台或 Amazon CLI 强制终止实例。有关步骤，请参阅强制终止实例。

   实例将首先尝试正常关闭，其中包括刷新文件系统缓存和元数据（虽然您可以选择绕过正常关闭）。如果在超时期限内无法完成正常关闭，则实例将强制关闭，而不会刷新文件系统缓存和元数据。

2. 如果强制终止失败

   如果在几个小时后，实例仍未终止，并且看上去终止被卡住，请执行以下操作：

   a. 请在 Amazon Web Services re:Post 上发布求助请求。为了帮助加快解决问题，请提供实例 ID 并描述已采取的步骤。

   b. 此外，如果您有支持计划，则可在支持中心创建技术支持案例。

强制终止实例

如果您的实例似乎卡在终止状态，您可以强制终止实例。如果几个小时后实例仍未终止，请向 Amazon re:Post 发送求助请求。为了帮助加快解决问题，请提供实例 ID 并描述已采取的步骤。此外，如果您有支持计划，则可在支持中心创建技术支持案例。

Console

要强制终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择卡住实例。
3. 依次选择实例状态和强制终止实例。

   请注意，只有当您的实例处于 `stopping` 状态时，强制终止实例才可用。如果您的实例处于其他状态（`shutting-down` 和 `terminated` 除外），您可以使用 Amazon CLI 强制终止实例。
4. （可选）要绕过强制终止期间操作系统的正常关闭，请选中跳过操作系统关闭复选框。
5. 选择强制终止。

Amazon CLI

要强制终止实例

使用 terminate-instances 命令和 `--force` 选项。

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

要绕过强制终止期间操作系统的正常关闭，请包含 `--skip-os-shutdown` 选项。

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

要强制终止实例

使用 Remove-EC2Instance cmdlet 并将 -Enforce 设置为 true。

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

要绕过强制终止期间操作系统的正常关闭，请包含 -SkipOsShutdown $true。

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

已终止实例仍然显示

在您终止某个实例之后，它会在删除之前的短时间内保持可见。状态显示为 terminated。如果该条目在几小时之后未删除，请联系 Support。

错误：此实例可能无法终止。修改其“disableApiTermination”实例属性

如果您尝试终止实例并收到 The instance *i-1234567890abcdef0* may not be terminated. Modify its 'disableApiTermination' instance attribute 错误消息，则表明此实例已启用终止保护。终止保护可防止实例意外终止。

终止实例前，必须先禁用终止保护。

有关更多信息，请参阅 更改实例终止保护。

自动启动或终止的实例

通常，以下行为意味着您已使用 Amazon EC2 Auto Scaling、EC2 队列或 Spot 队列，根据已定义的条件自动扩展计算资源：

- 您终止实例，新实例将自动启动。
- 您启动一个实例，并且其中一个实例将自动终止。
- 您停止实例且此实例终止，新实例会自动启动。

要停止自动扩缩，请找到正在启动实例的自动扩缩组或实例集，然后将其容量设置为 0 或将其删除。

排查 Amazon EC2 实例无法访问的问题

以下信息可帮助您排查无法访问的 Amazon EC2 实例的问题。您可以捕获屏幕截图或获取控制台输出，以便帮助诊断问题并确定您是否应重启实例。对于无法访问的 Windows 实例，请根据服务返回的屏幕截图进行问题排查。

内容

- 实例重启
- 实例控制台输出
- 捕获无法访问的实例的屏幕截图
- 用于排查无法访问的 Windows 实例问题的常见屏幕截图
- 主机发生故障时的实例恢复
- 实例出现脱机后又意外重启

实例重启

能够重启无法访问的实例对于故障排除和一般实例管理都非常有用。

就像可以通过按下重置按钮来重置计算机一样，您可以使用 Amazon EC2 控制台、CLI 或 API 来重置 EC2 实例。有关更多信息，请参阅 重启 Amazon EC2 实例。

实例控制台输出

控制台输出对于问题诊断是非常有价值的工具。它尤其适合用于排查内核问题和服务配置问题，它们可能会导致实例在 SSH 后台程序启动前终止或变得不可达到。

- Linux 实例：实例控制台输出显示了确切的控制台输出，在正常情况下，这些输出会在连接到计算机的物理显示器上显示。控制台输出返回缓冲的信息，该信息在实例转变状态 (启动、停止、重新引导和终止) 之后很快发布。发布的输出不会持续更新；仅当它可能是最大值时。
- Windows 实例：实例控制台输出包括最后三个系统事件日志错误。

只有实例的所有人可以访问控制台输出。

您可以在实例生命周期中检索最新的串行控制台输出。仅基于 Nitro 的实例支持此选项。

Console

获取控制台输出

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择实例。
4. 依次选择 Actions (操作)、Monitor and troubleshoot (监控和问题排查)、Get system log (获取系统日志)。

Amazon CLI

获取控制台输出

使用 get-console-output 命令。

```
aws ec2 get-console-output --instance-id i-1234567890abcdef0
```

PowerShell

获取控制台输出

使用 Get-EC2ConsoleOutput cmdlet。

```
Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0
```

捕获无法访问的实例的屏幕截图

如果无法连接到实例，则您可以捕获实例的屏幕截图并将其作为图像查看。该图像可以让您查看实例的状态，更快地处理问题。

您可在实例运行时或在其崩溃后生成屏幕截图。生成的图像为 JPG 格式，大小不超过 100 kb。屏幕截图不会产生数据传输费用。

限制

以下客户端不支持此功能：

- 裸机实例（类型 `*.metal` 的实例）

- 实例正在使用 NVIDIA GRID 驱动程序
- 由基于 Arm 的 Graviton 处理器支持的实例
- Amazon Outposts 上的 Windows 实例
- Amazon 本地区域上的 Windows 实例

区域支持

此功能在下列区域尚未开放：

- 亚太地区（泰国）
- 墨西哥（中部）
- GovCloud 区域

Console

获取实例的屏幕截图

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航窗格中，选择 Instances (实例)。
3. 选择要捕获的实例。
4. 依次选择 Actions (操作)、Monitor and troubleshoot (监控和问题排查)、Get instance log (获取实例屏幕截图)。
5. 选择 Download (下载)，或右键单击图像，以下载并保存该图像。

Amazon CLI

捕获实例的屏幕截图

使用 get-console-screenshot 命令。输出为 base64 编码。

```
aws ec2 get-console-screenshot --instance-id i-1234567890abcdef0
```

PowerShell

捕获实例的屏幕截图

使用 Get-EC2ConsoleScreenshot cmdlet。输出为 base64 编码。

```
Get-EC2ConsoleScreenshot -InstanceId i-1234567890abcdef0
```

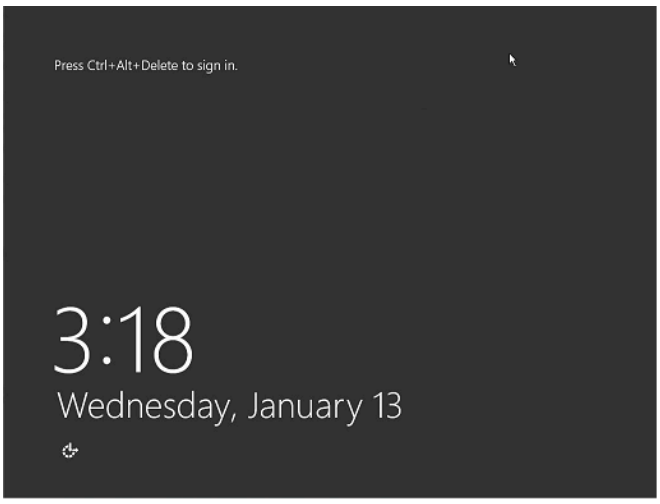
用于排查无法访问的 Windows 实例问题的常见屏幕截图

可以根据服务返回的屏幕截图，利用以下信息帮助对无法访问的 Windows 实例进行故障排除。

- 登录屏幕 (Ctrl+Alt+Delete)
- 恢复控制台屏幕
- Windows 引导管理器屏幕
- Sysprep 屏幕
- Getting Ready 屏幕
- Windows Update 屏幕
- Chkdsk

登录屏幕 (Ctrl+Alt+Delete)

控制台屏幕截图服务返回以下截图。



如果实例在登录时无法访问，可能是因为网络配置或 Windows 远程桌面服务存在问题。如果某个进程占用了大量 CPU，实例也可能没有响应。

网络配置

使用以下信息验证您的 Amazon、Microsoft Windows 和本地（或内部部署）网络配置是否阻止了对实例的访问。

Amazon 网络配置

配置	验证
安全组配置	验证端口 3389 是否向您的安全组开放。验证您是否连接到正确的公有 IP 地址。如果实例未与弹性 IP 关联，公有 IP 就会在实例停止或启动后发生更改。有关更多信息，请参阅远程桌面无法连接到远程计算机。
VPC 配置 (网络 ACL)	验证您的 Amazon VPC 的访问控制列表 (ACL) 是否阻止访问。有关信息，请参阅 Amazon VPC 用户指南 中的网络 ACL。
VPN 配置	如果您使用虚拟专用网络（VPN）连接您的 VPC，请验证 VPN 隧道的连通性。有关更多信息，请参阅 Troubleshooting Amazon Client VPN: Tunnel connectivity issues to a VPC。

Windows 网络配置

配置	验证
Windows 防火墙	验证 Windows 防火墙是否阻止与您的实例建立连接。按照远程桌面故障排除部分第 7 项的说明禁用 Windows 防火墙：远程桌面无法连接到远程计算机。
高级 TCP/IP 配置 (使用静态 IP)	实例可能因您配置了静态 IP 地址而没有响应。对于 VPC，请创建网络接口并将其连接到实例。

本地或内部部署网络配置

验证本地网络配置是否阻止访问。尝试连接无法访问的实例所在 VPC 中的其他实例。如果您无法访问其他实例，请联系本地网络管理员确定是否有本地策略限制了访问。

远程桌面服务问题

如果实例在登录时无法访问，可能是因为实例存在远程桌面服务 (RDS) 问题。

> **Tip**
>
> 您可以使用 `AWSSupport-TroubleshootRDP` 运行手册检查和修改可能影响远程桌面协议 (RDP) 连接的各种设置。有关更多信息，请参阅《Amazon Systems Manager Automation Runbook Reference》中的 `AWSSupport-TroubleshootRDP`。

远程桌面服务配置

配置	验证
RDS 正在运行	验证实例上的 RDS 是否在运行。使用 Microsoft 管理控制台 (MMC) 的服务管理单元 (`services.msc`) 连接实例。在服务列表中，验证Remote Desktop Services (远程桌面服务) 是否处于正在运行状态。如果不是，请将其开启，然后将启动类型设置为 Automatic。如果使用服务管理单元无法连接到实例，请将根卷从该实例分离并为其创建快照或 AMI，将原始卷作为辅助卷附加到同一个可用区中的其他实例，并修改 Start 注册表项。完成后，请重新将根卷附加到原始实例。
RDS 已启用	即使该服务已启动，它也有可能被禁用。将根卷从实例分离并为其创建快照或 AMI，将原始卷作为辅助卷挂载到同一个可用区中的其他实例，然后按照 在具有远程注册表的 EC2 实例上启用远程桌面中的说明，通过修改 Terminal Server (终端服务器) 注册表项来启用该服务。 完成后，请重新将根卷附加到原始实例。

较高的 CPU 使用率

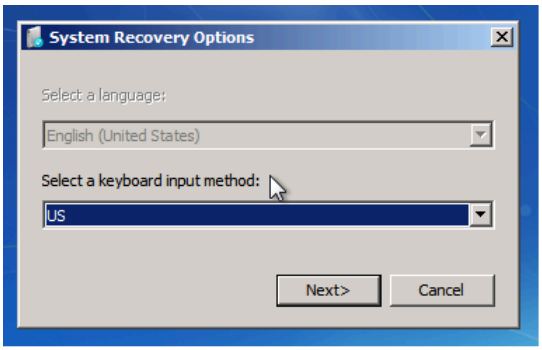
使用 Amazon CloudWatch 检查您实例上的 CPUUtilization (Maximum) 指标。如果 CPUUtilization (Maximum) 的值比较大，请等待 CPU 使用率下降后尝试重新连接。CPU 使用率高可能由以下原因导致：

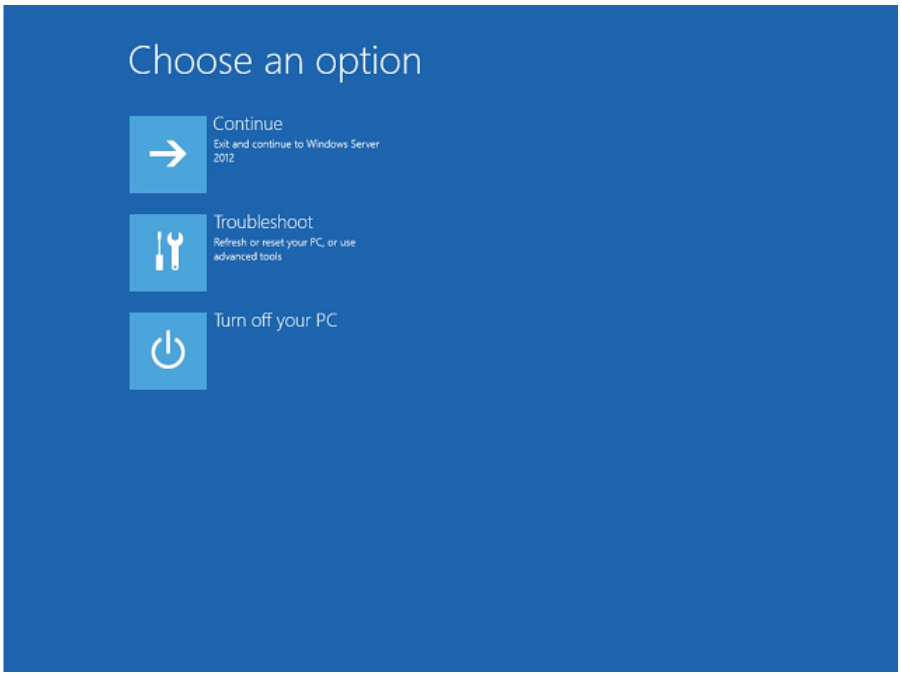
- Windows 更新
- 安全软件扫描
- 自定义启动脚本
- 任务计划程序

有关更多信息，请参阅 Amazon CloudWatch 用户指南 中的获取特定资源的统计信息。有关其他故障排除提示，请参阅 Windows 启动后，CPU 使用率短时增高（仅限 Windows 实例）。

恢复控制台屏幕

控制台屏幕截图服务返回以下截图。





如果 `bootstatuspolicy` 未设置为 `ignoreallfailures`，操作系统可能启动至“Recovery”（恢复）控制台，并停滞在这一状态。请按照以下步骤将 `bootstatuspolicy` 配置更改为 `ignoreallfailures`。

默认情况下，Amazon 提供的公有 Windows AMI 的策略配置设置为 `ignoreallfailures`。

1. 停止无法访问的实例。
2. 创建根卷的快照。根卷是作为 `/dev/sda1` 附加到实例的。

   将根卷从无法访问的实例分离并为其创建快照或 AMI，然后将根卷作为辅助卷附加到同一个可用区中的其他实例。

   > **Warning**
   >
   > 如果您的临时实例和原始实例是使用相同的 AMI 启动的，则您必须完成额外步骤，否则在您恢复原始实例的根卷之后，由于磁盘签名冲突，您将无法启动该原始实例。如果您必须使用相同的 AMI 创建临时实例，为了避免磁盘签名冲突，请完成 磁盘签名冲突 中的步骤。
   >
   > 或者，可以为临时实例选择不同的 AMI。例如，如果原始实例使用适用于 Windows Server 2016 的 AMI，则使用适用于 Windows Server 2019 的 AMI 来启动临时实例。

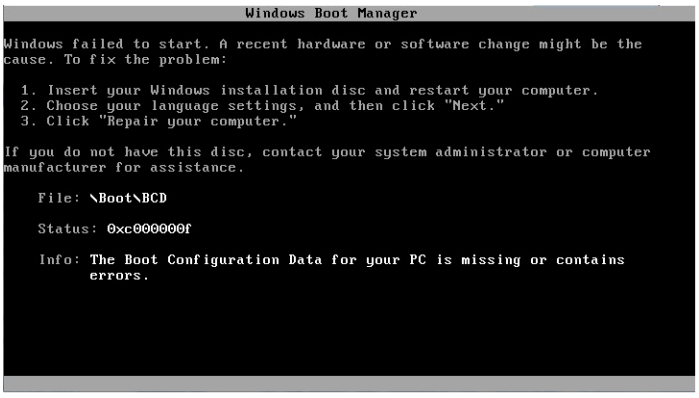
3. 登录实例并从命令提示符运行以下命令，以将 `bootstatuspolicy` 配置更改为 `ignoreallfailures`。

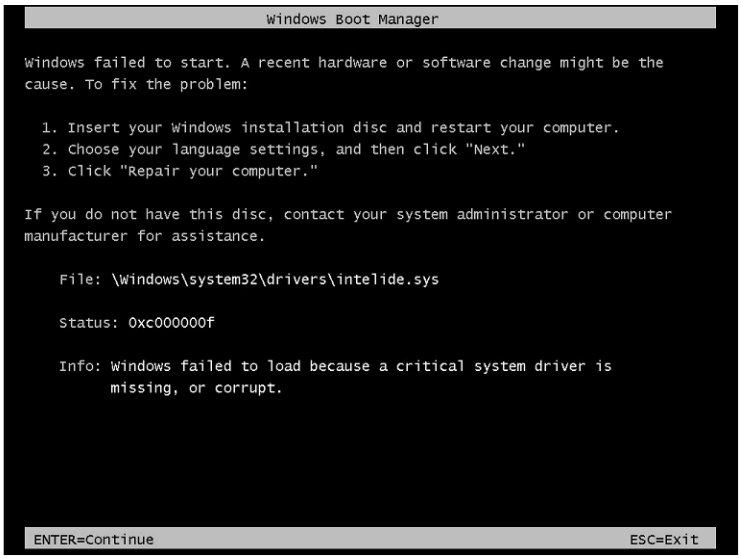
   ```
   bcdedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy
    ignoreallfailures
   ```

4. 重新将卷附加到无法访问的实例并重新启动实例。

Windows 引导管理器屏幕

控制台屏幕截图服务返回以下截图。

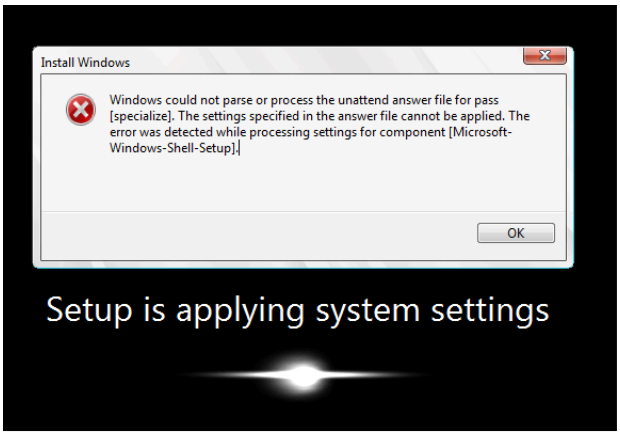


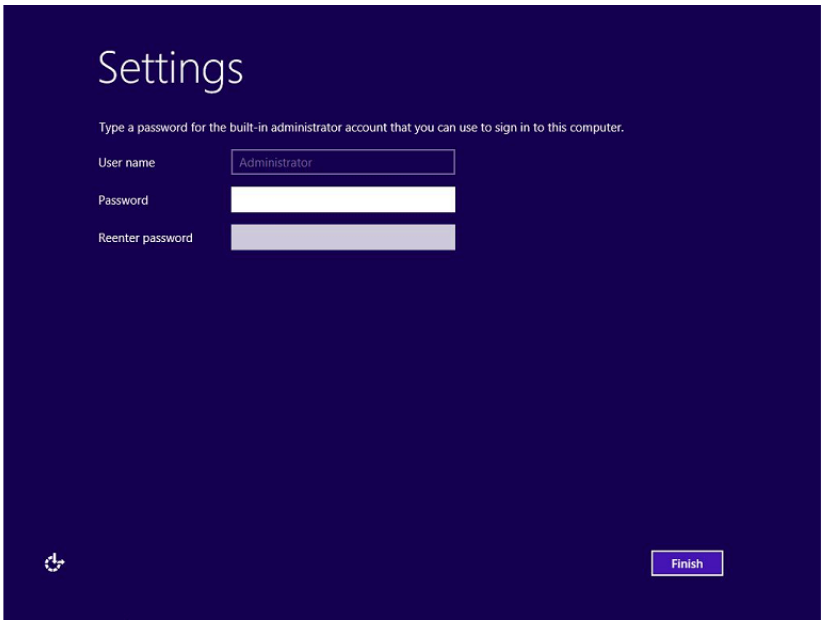


操作系统的系统文件和/或注册表遭到严重损坏。当实例停滞在这一状态时，您应该从最近备份的 AMI 恢复实例或启动一个替代实例。如果您需要访问实例上的数据，请将所有根卷从无法访问的实例上分离并为这些卷创建快照或 AMI，然后将其作为辅助卷附加到同一个可用区中的其他实例。

Sysprep 屏幕

控制台屏幕截图服务返回以下截图。

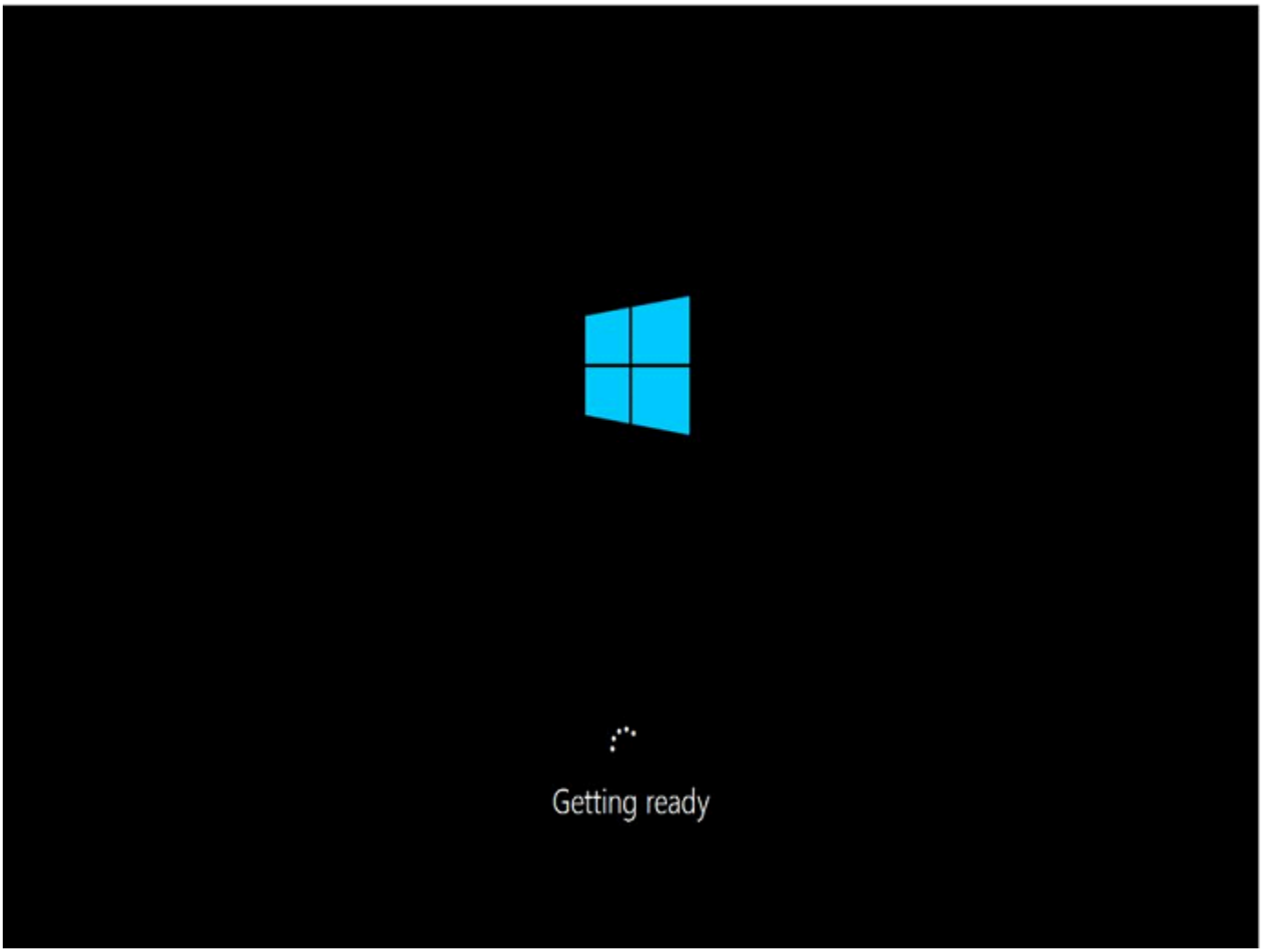




如果您没有使用 EC2Config 服务调用 Sysprep，或者操作系统在运行 Sysprep 时出现故障，您可能会看到此屏幕。您可以使用 EC2Rescue 重置密码。否则，请参阅使用 Windows Sysprep 创建 Amazon EC2 AMI。

Getting Ready 屏幕

控制台屏幕截图服务返回以下截图。



反复刷新实例控制台屏幕截图服务，验证环形进度条是否转动。如果环形进度条转动，请等待操作系统启动。您也可以使用 Amazon CloudWatch 查看您实例上的 CPUUtilization (Maximum) 指标，以确认操作系统是否处于活动状态。如果环形进度条不转，则实例可能停滞在启动过程。重启实例。如果重新启动无法解决问题，请从最近备份的 AMI 中恢复实例或启动一个替代实例。如果需要访问实例上的数据，请将根卷从无法访问的实例中分离，并为其创建快照或 AMI。然后，将根卷作为辅助卷附加到同一个可用区中的其他实例。

Windows Update 屏幕

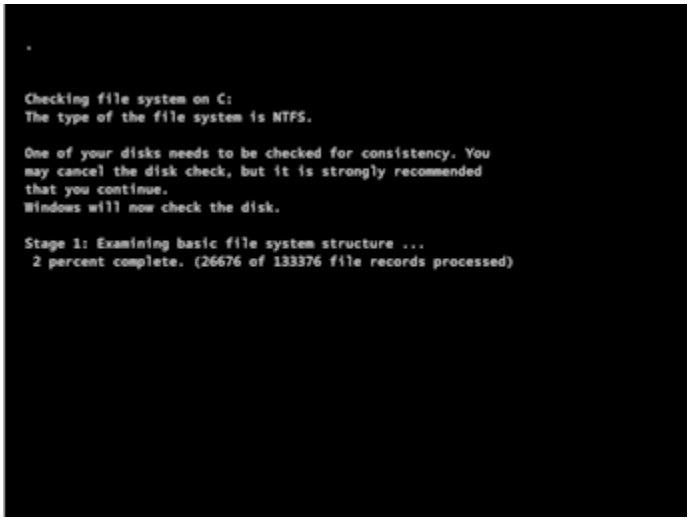
控制台屏幕截图服务返回以下截图。

Windows 更新进程正在更新注册表。请等待更新完成。请勿在更新期间重新启动或停止实例，因为这可能会导致数据损坏。

> **Note**
>
> 在更新时，Windows 更新进程会占用服务器上的资源。若您频繁遇到这一问题，可以考虑使用速度更快的实例类型和 EBS 卷。

Chkdsk

控制台屏幕截图服务返回以下截图。



Windows 正在驱动器上运行 chkdsk 系统工具，以验证文件系统的完整性并修复逻辑文件系统错误。请等待该进程完成。

主机发生故障时的实例恢复

如果底层主机上的硬件出现不可恢复性问题，Amazon 可能会预定实例停止事件。我们会通过电子邮件提前通知您这类事件。

恢复发生故障的主机上运行的 Amazon EBS 支持的实例

1. 将您实例存储卷上的所有关键数据 Amazon EBS 或 Amazon S3。
2. 停止实例。
3. 启动实例。
4. 恢复所有重要数据。

有关更多信息，请参阅 启动和停止 Amazon EC2 实例。

恢复具有发生故障的主机上运行的实例存储根卷的实例

1. 从该实例创建 AMI。
2. 将映像上传到 Amazon S3。
3. 将重要数据备份到 Amazon EBS 或 Amazon S3。
4. 终止实例。
5. 从 AMI 启动新实例。
6. 将所有重要数据恢复到新实例。

实例出现脱机后又意外重启

如果实例出现脱机后又意外重启，则可能是已经进行了实例自动恢复。当 Amazon 检测到实例因底层硬件或软件问题而无法使用，且在该实例上启用了简化的自动恢复或基于 CloudWatch 操作的恢复时，就会出现这种情况。

在恢复过程中，Amazon 尝试将实例迁移到其他硬件来恢复实例的可用性。要确认实例是否进行了实例自动恢复，请参阅验证是否已进行实例自动恢复。

> **Note**
>
> 如果工作负载或应用程序没有响应，请检查工作负载或应用程序是否是在实例上运行。如果不是，则手动启动工作负载或应用程序。为防将来出现此问题，请实施恢复计划，确保工作负载或应用程序在实例恢复后能正常运行。

排查 Amazon EC2 Linux 实例的连接问题

以下信息和常见错误可以帮助您排查连接到实例时出现的问题。

连接问题

- 连接问题的常见原因
- 连接到您的实例时出错：连接超时
- Error: unable to load key ...(错误：无法加载密钥...) Expecting: ANY PRIVATE KEY(需要：任何私有密钥)
- 错误：服务器无法识别用户密钥
- 错误：权限被拒绝或 [实例] 端口 22 关闭了连接
- 错误：未保护的私有密钥文件
- 错误：私有密钥的格式必须以“-----BEGIN RSA PRIVATE KEY-----”开头，以“-----END RSA PRIVATE KEY-----”结尾
- 错误：主机密钥验证失败
- 错误：服务器拒绝我们的密钥或 没有支持的身份验证方法
- 无法对实例执行 Ping 操作
- 错误：服务器意外关闭了网络连接
- 错误：EC2 Instance Connect 的主机密钥验证失败
- 无法使用 EC2 Instance Connect 连接到 Ubuntu 实例
- 我丢失了私有密钥。如何连接到我的实例？

连接问题的常见原因

建议您通过验证是否正确执行了以下任务，开始排查实例连接问题。

验证实例的用户名

您可以使用用户账户的用户名或用于启动实例的 AMI 的默认用户名连接到实例。

- 获取用户账户的用户名。

  有关如何创建用户账户的更多信息，请参阅管理 Amazon EC2 Linux 实例上的系统用户。

- 获取用于启动实例的 AMI 的默认用户名。

用于启动实例的 AMI	默认用户名
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
其他	检查 AMI 提供程序

验证安全组规则是否允许流量

确保与您的实例关联的安全组允许来自您的 IP 地址的入站 SSH 流量。默认情况下，VPC 的默认安全组不允许传入 SSH 流量。默认情况下，由启动实例向导创建的安全组允许传入的 SSH 流量。有关向 Linux 实例添加入站 SSH 流量规则的步骤，请参阅 用于从您的计算机连接到实例的规则。有关验证步骤，请参阅 连接到您的实例时出错：连接超时。

验证实例是否准备就绪

启动实例后，可能需要花几分钟时间，实例才能准备好接受连接请求。检查实例以确保它正在运行并通过了状态检查。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择您的实例。
3. 请验证以下内容：
   a. 在Instance state (实例状态) 列中，验证您的实例是否处于 `running` 状态。
   b. 在 Status check (状态检查) 列中，验证您的实例是否已通过所有状态检查。

确认您已满足连接的所有先决条件

确保您拥有连接所需的所有信息。有关更多信息，请参阅 使用 SSH 连接到 Linux 实例。

从 Linux 或 macOS 进行连接

如果本地计算机操作系统是 Linux 或 macOS X，请检查以下连接到 Linux 实例的特定先决条件：

- SSH 客户端
- EC2 Instance Connect
- Amazon Systems Manager 会话管理器

从 Windows 进行连接

如果本地计算机操作系统是 Windows，请检查以下连接到 Linux 实例的特定先决条件：

- OpenSSH
- PuTTY
- Amazon Systems Manager 会话管理器
- 适用于 Linux 的 Windows 子系统

检查实例是否为托管实例

不允许用户向托管实例发起连接。要确定实例是否为托管实例，找到该实例的托管字段。如果此值为 true，则该实例为托管实例。有关更多信息，请参阅 Amazon EC2 托管式实例。

连接到您的实例时出错：连接超时

如果连接到实例时显示错误消息 `Network error: Connection timed out` 或 `Error connecting to [instance], reason: -> Connection timed out: connect`，请尝试以下选项：

检查您的安全组规则。

您需要一条安全组规则，它允许在适当端口上传输来自您的本地计算机公有 IPv4 地址的入站流量。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择您的实例。
3. 在控制台页面底部的 Security (安全) 选项卡上的 Inbound rules (入站规则) 下，检查对所选实例生效的规则列表。验证是否有允许流量从本地计算机到端口 22（SSH）的规则。

   如果您的安全组没有允许来自您的本地计算机的入站流量的规则，请向您的安全组添加一条规则。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。
4. 对于允许入站流量的规则，请检查 Source（源）字段。如果该值是单个 IP 地址，并且如果该 IP 地址不是静态的，则每次重新启动计算机时都会分配一个新的 IP 地址。这将导致该规则不包括您计算机的 IP 地址流量。如果您的计算机位于企业网络上，或者当您通过互联网服务提供商 (ISP) 进行连接时，抑或您的计算机的 IP 地址是动态的，则该 IP 地址可能不是静态的，并且每次重新启动计算机时都会更改。为了确保您的安全组规则允许来自您的本地计算机的入站流量，应该指定您的客户端计算机使用的 IP 地址的范围，而不是为 Source（源）指定单个 IP 地址。

   有关安全组规则的更多信息，请参阅Amazon VPC 用户指南中的安全组规则。

查看子网的路由表。

您需要使用某个路由，以将发往 VPC 外部的所有流量发送到 VPC 的 Internet 网关。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择您的实例。
3. 在 Networking (联网) 选项卡上，记下 VPC ID 和 Subnet ID (子网 ID) 的值。
4. 通过 https://console.aws.amazon.com/vpc/ 打开 Amazon VPC 控制台。
5. 在导航窗格中，选择 Internet Gateways（互联网网关）。验证是否有 Internet 网关附加到您的 VPC。否则，请选择 Create internet gateway (创建 Internet 网关)，为 Internet 网关输入名称，然后选择 Create internet gateway (创建 Internet 网关)。然后，对于您创建的 Internet 网关，依

次选择 Actions (操作)、Attach to VPC (附加到 VPC)，选择您的 VPC，然后选择 Attach internet gateway (附加 Internet 网关) 将其附加到您的 VPC。

6. 在导航窗格中，选择 Subnets，然后选择您的子网。
7. 在 Route Table (路由表) 选项卡上，验证 `0.0.0.0/0` 的路由是否为目的地以及您的 VPC 的 Internet 网关是否为目标。如果您使用实例的 IPv6 地址连接到实例，请检查是否有一个路由可以将所有 IPv6 流量 (`::/0`) 指向 Internet 网关。否则请执行以下操作：

   a. 选择路由表的 ID (rtb-xxxxxxxx) 以导航到路由表。
   b. 在 Routes（路由）选项卡上，选择 Edit routes（编辑路由）。选择 Add route (添加路由)，将 `0.0.0.0/0` 用作目的地并将 Internet 网关用作目标。对于 IPv6，选择 Add route (添加路由)，将 `::/0` 用作目的地并将 Internet 网关用作目标。
   c. 选择 Save routes（保存路由）。

检查子网的网络访问控制列表 (ACL)。

该网络 ACL 必须允许端口 22 上来自本地 IP 地址的入站 SSH 流量。还须允许到临时端口 (1024-65535) 的出站流量。

1. 通过 https://console.aws.amazon.com/vpc/ 打开 Amazon VPC 控制台。
2. 在导航窗格中，选择 Subnets(子网)。
3. 选择您的子网。
4. 对于 Inbound rules (入站规则)，请在 Network ACL (网络 ACL) 选项卡上验证这些规则是否允许来自所需端口的计算机的入站流量。如果不允许，请删除或修改阻止该流量的规则。
5. 对于 Outbound rules (出站规则)，验证规则是否允许到您的计算机临时端口的出站流量。如果不允许，请删除或修改阻止该流量的规则。

如果您的计算机在企业网络上

请询问网络管理员内部防火墙是否允许端口 22 上来自您计算机的入站和出站流量。

如果计算机有防火墙，请验证其是否允许端口 22 上来自您计算机的入站和出站流量。

检查您的实例是否具有公有 IPv4 地址。

如果没有，您可以将弹性 IP 地址与您的实例关联。有关更多信息，请参阅弹性 IP 地址。

检查实例上的 CPU 负载；服务器可能已超过负载。

Amazon 自动提供数据，例如 Amazon CloudWatch 指标和实例状态，您可以使用这些数据查看实例上 CPU 的负载情况；如有必要，还可以调整负载的处理方式。有关更多信息，请参阅使用 CloudWatch 监控您的实例。

- 如果您的负载是可变的，您可以使用 Auto Scaling 和 Elastic Load Balancing 自动增加或减少实例。
- 如果您的负载呈稳定增长的态势，您可以迁移到更大的实例类型。有关更多信息，请参阅Amazon EC2 实例类型更改。

要使用 IPv6 地址连接实例，请检查以下各项：

- 您的子网必须与一个路由表关联，此表中具有一个将 IPv6 流量 (`::/0`) 指向 Internet 网关的路由。
- 安全组规则必须允许端口 22 上来自本地 IPv6 地址的入站流量。
- 您的网络 ACL 规则必须允许入站和出站 IPv6 流量。
- 如果您从旧版 AMI 启动实例，则其可能未针对 DHCPv6 进行配置（IPv6 地址不会在网络接口上自动识别）。有关更多信息，请参阅《Amazon VPC 用户指南》中的 在实例中配置 IPv6。
- 您的本地计算机必须拥有 IPv6 地址，且必须配置为使用 IPv6。

Error: unable to load key ...(错误：无法加载密钥...) Expecting: ANY PRIVATE KEY(需要：任何私有密钥)

如果您尝试连接到您的实例并收到错误消息 `unable to load key ... Expecting: ANY PRIVATE KEY`，则说明未正确配置用于存储私有密钥的文件。如果私有密钥文件以 `.pem` 为结尾，则它可能仍未正确配置。未正确配置私有密钥文件的一个可能原因是缺少证书。

如果未正确配置私有密钥文件，请按照下列步骤解决该错误

1. 创建新的密钥对。有关更多信息，请参阅 使用 Amazon EC2 创建密钥对。

   > **Note**
   >
   > 另外，您也可以使用第三方工具创建新的密钥对。有关更多信息，请参阅 使用第三方工具创建密钥对，并将公有密钥导入 Amazon EC2。

2. 将新密钥对添加到您的实例。有关更多信息，请参阅我丢失了私有密钥。如何连接到我的实例？。
3. 使用新的密钥对连接到实例。

错误：服务器无法识别用户密钥

如果您使用 SSH 连接到实例

- 请在连接时使用 `ssh -vvv` 获得三倍的详细调试信息：

```
ssh -vvv -i path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

下列样本输出演示了如果您尝试使用服务器无法识别的密钥连接实例时您可能会看到的信息：

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-
interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

如果您使用 PuTTY 连接到实例

- 验证私有密钥 (.pem) 文件是否已转换为 PuTTY 可识别的格式 (.ppk)。有关转换您的私有密钥的更多信息，请参阅 使用 PuTTY 连接到 Linux 实例。

> **Note**
>
> 在 PuTTYgen 中，加载私有密钥文件并选择 Save Private Key (保存私有密钥) 而不是 Generate (生成)。

- 验证您在连接时是否对为 AMI 使用了正确的用户名。在 PuTTY 配置窗口的主机名框中输入用户名。

用于启动实例的 AMI	默认用户名
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
其他	检查 AMI 提供程序

- 验证您的入站安全组规则允许入站流量进入合适的端口。有关更多信息，请参阅 用于从您的计算机连接到实例的规则。

错误：权限被拒绝或 [实例] 端口 22 关闭了连接

如果您使用 SSH 连接到实例并出现任何以下错误：`Host key not found in [directory]`、`Permission denied (publickey)`、`Authentication failed, permission denied` 或 `Connection closed by [instance] port 22`，则请确认您使用的是 AMI 对应的用户名进行连接并且为实例指定了正确的私有密钥 (`.pem`) 文件。

正确的用户名如下所示：

用于启动实例的 AMI	默认用户名
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
其他	检查 AMI 提供程序

例如，要使用 SSH 客户端连接到从 Amazon Linux 实例，请使用以下命令：

```
ssh -i /path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

请确认您使用的私有密钥文件对应于您启动实例时选择的密钥对。

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择您的实例。
3. 在 Details （详细信息) 选项卡的 Instance details (实例详细信息) 下，验证 Key pair name (密钥对名称) 的值。
4. 如果您启动实例时没有指定密钥对，则可以终止实例并启动新实例，从而确保指定密钥对。如果这是您一直使用的实例，但您不再有密钥对的 .pem 文件，则可以使用新的密钥对取代该密钥对。有关更多信息，请参阅我丢失了私有密钥。如何连接到我的实例？。

如果您已经生成了您自己的密钥对，请确保您的密钥生成器被设置为创建 RSA 密钥。不接受 DSA 密钥。

如果您遇到 `Permission denied (publickey)` 错误但以上情况都不适用 (例如，您之前能够连接)，则可能是实例主目录的权限发生了更改。/home/*instance-user-name*/.ssh/authorized_keys 的权限必须限制为仅限所有者。

在您的实例上验证权限

1. 停止您的实例并分离根卷。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。
2. 在当前实例所在的可用区中启动一个临时实例 (使用与您用于当前实例的 AMI 类似或相同的 AMI)，并将根卷附加到此临时实例。
3. 连接临时实例，创建一个挂载点并挂载您附加的卷。
4. 在临时实例中，检查附加的卷的 /home/*instance-user-name*/ 目录的权限。如有必要，按如下方式调整权限：

```
[ec2-user ~]$ chmod 600 mount_point/home/instance-user-name/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name
```

5. 卸载该卷，将其与临时实例分离，然后将其重新附加到原来的实例。确保为根卷指定正确的设备名称；例如，/dev/xvda。
6. 启动您的实例。如果不再需要临时实例，可以终止它。

错误：未保护的私有密钥文件

必须保护您的私钥文件，防止其他任何用户对其进行读写操作。如果除您外其他任何人都能够读取或写入您的私钥，则 SSH 会忽略您的密钥，并且您会看到以下警告消息。

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem
Permission denied (publickey).
```

如果在尝试登录到您的实例时看到类似的消息，请检查此错误消息的第一行，验证您为实例使用的公钥是否正确。上述示例利用 `.ssh/my_private_key.pem` 文件权限使用私钥 `0777`，这可使任何人都能读取或写入此文件。此权限级别非常不安全，因此 SSH 会忽略此密钥。

如果从 macOS 或 Linux 连接，请运行以下命令以修复此错误，替入您的私有密钥文件的路径。

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

如果要从 Windows 连接到 Linux 实例，请在本地计算机上执行以下步骤。

1. 导航到您的 .pem 文件。
2. 右键单击 .pem 文件并选择 Properties（属性）。
3. 选择安全性选项卡。
4. 选择 Advanced (高级)。
5. 验证您是否是文件的所有者。如果不是，请将所有者更改为您的用户名。
6. 选择 Disable inheritance (禁用继承) 和 Remove all inherited permissions from this object (从此对象中删除所有继承的权限)。
7. 选择 Add (添加)、Select a principal (选择委托人)，输入您的用户名，然后选择 OK (确定)。
8. 从 Permission Entry (权限条目) 窗口，授予 Read (读取) 权限，然后选择 OK (确定)。
9. 单击 Apply（应用）以确保所有设置都已保存。
10. 选择 OK (确定) 关闭 Advanced Security Settings (高级安全设置) 窗口。
11. 选择 OK (确定) 关闭 Properties (属性) 窗口。

12. 您应该能够使用 SSH 从 Windows 连接到 Linux 实例。

从 Windows 命令提示符处运行以下命令。

1. 在命令提示符下，导航到 .pem 文件的文件路径位置。
2. 运行以下命令以重置和删除显式权限：

```
icacls.exe $path /reset
```

3. 运行以下命令以向当前用户授予读取权限：

```
icacls.exe $path /GRANT:R "$($env:USERNAME):(R)"
```

4. 运行以下命令以禁用继承并删除继承的权限。

```
icacls.exe $path /inheritance:r
```

5. 您应该能够使用 SSH 从 Windows 连接到 Linux 实例。

错误：私有密钥的格式必须以“-----BEGIN RSA PRIVATE KEY-----”开头，以“-----END RSA PRIVATE KEY-----”结尾

如果您使用第三方工具（如 ssh-keygen）创建 RSA 密钥对，则它会生成 OpenSSH 密钥格式的私有密钥。当您连接到实例时，如果使用 OpenSSH 格式的私有密钥来解密密码，您将收到错误 `Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"`。

要解决该错误，私有密钥必须采用 PEM 格式。使用以下命令创建 PEM 格式的私有密钥：

```
ssh-keygen -m PEM
```

错误：主机密钥验证失败

如果实例上存储在 `known_hosts` 文件中的主机密钥与客户端上的主机密钥不匹配，则会出现此错误。例如，假设使用一个公有 IP 地址连接到某个实例，然后尝试使用其他公有 IP 地址重新连接到该实例，则可能会出现不匹配的情况。这一问题可能在添加或移除弹性 IP 地址之后出现，因为此操作会更改实例的公有 IP 地址。

要解决此错误，首先要确认实例的主机密钥或网络配置更改符合预期。连接到实例之前，您可能还需要验证主机指纹。连接到实例后，您可以从 known_hosts 文件中移除旧主机密钥。有关说明，请参阅有关实例上所用 Linux 发行版的文档。

错误：服务器拒绝我们的密钥或 没有支持的身份验证方法

如果您使用 PuTTY 连接到实例并出现以下任一错误：错误：服务器拒绝了我们的密钥或错误：没有支持的身份验证方法，请确认您使用 AMI 的相应用户名进行连接。在 PuTTY 配置窗口的用户名中键入用户名。

正确的用户名如下所示：

用于启动实例的 AMI	默认用户名
Amazon Linux	`ec2-user`
CentOS	`centos` 或 `ec2-user`
Debian	`admin`
Fedora	`fedora` 或 `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` 或 `root`
SUSE	`ec2-user` 或 `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
其他	检查 AMI 提供程序

您还应该验证：

- 您使用的是否是最新版本的 PuTTY？ 有关更多信息，请参阅 PuTTY 网页。

- 已将您的私有密钥 (.pem) 文件正确转换为 PuTTY 可识别的格式 (.ppk)。有关转换您的私有密钥的更多信息，请参阅 使用 PuTTY 连接到 Linux 实例。

无法对实例执行 Ping 操作

ping 命令是一种 ICMP 流量 — 如果您无法对实例执行 ping 操作，请确保您的入站安全组规则允许的 Echo Request 消息的 ICMP 流量来自所有资源，或来自从中发出命令的计算机或实例。

如果您无法从实例发出 ping 命令，请确保您的出站安全组规则允许的 Echo Request 消息的 ICMP 流量发送到所有目标，或发送到您正在尝试对其执行 ping 操作的主机。

Ping由于网络延迟或硬件问题， 命令还可能被防火墙阻止或超时。您应咨询本地网络或系统管理员，以帮助进行进一步的故障排除。

错误：服务器意外关闭了网络连接

如果您使用 PuTTY 连接到实例并出现“服务器意外关闭了网络连接”错误，请确认您已在 PuTTY Configuration (PuTTY 配置) 的 Connection (连接) 页面上启用 keepalives 以避免断开连接。有些服务器如果在指定的时间内未接收到任何数据，将会断开与客户端的连接。将“Seconds between keepalives”(keepalives 之间的秒数) 设置为 59 秒。

如果在启用 keepalives 后仍出现问题，请尝试在 PuTTY Configuration (PuTTY 配置) 的 Connection (连接) 页面上禁用 Nagle 的算法。

错误：EC2 Instance Connect 的主机密钥验证失败

如果您轮换实例主机密钥，则新主机密钥不会自动上传到 Amazon 可信主机密钥数据库。当您尝试使用 EC2 Instance Connect 基于浏览器的客户端连接到实例时，会导致主机密钥验证失败，并且无法连接到实例。

要解决此错误，您必须在实例上运行 eic_harvest_hostkeys 脚本，该脚本会将新主机密钥上传到 EC2 Instance Connect。脚本位于 Amazon Linux 2 实例上的 /opt/aws/bin/ 和 Ubuntu 实例上的 /usr/share/ec2-instance-connect/。

Amazon Linux 2

解决 Amazon Linux 2 实例上的主机密钥验证失败错误

1. 使用 SSH 连接到实例。

您可以使用 EC2 Instance Connect CLI 或使用在启动实例时为其分配的 SSH 密钥对以及用于启动实例的 AMI 的默认用户名进行连接。对于 Amazon Linux 2，默认用户名为 ec2-user。

例如，如果实例是使用 Amazon Linux 2 启动的，实例的公有 DNS 名称是 ec2-a-b-c-d.us-west-2.compute.amazonaws.com，并且密钥对是 my_ec2_private_key.pem，请使用以下命令通过 SSH 连接到实例：

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

有关连接到实例的更多信息，请参阅使用 SSH 客户端连接到 Linux 实例。

2. 导航到以下文件夹。

```
[ec2-user ~]$ cd /opt/aws/bin/
```

3. 在您的实例上运行以下命令。

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

请注意，调用成功导致没有输出。

现在，您可以使用 EC2 Instance Connect 基于浏览器的客户端连接到您的实例。

Ubuntu

解决 Ubuntu 实例上的主机密钥验证失败错误

1. 使用 SSH 连接到实例。

您可以使用 EC2 Instance Connect CLI 或使用在启动实例时为其分配的 SSH 密钥对以及用于启动实例的 AMI 的默认用户名进行连接。对于 Ubuntu，默认用户名是 ubuntu。

例如，如果实例是使用 Ubuntu 启动的，实例的公有 DNS 名称是 ec2-a-b-c-d.us-west-2.compute.amazonaws.com，并且密钥对是 my_ec2_private_key.pem，请使用以下命令通过 SSH 连接到实例：

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

有关连接到实例的更多信息，请参阅使用 SSH 客户端连接到 Linux 实例。

2. 导航到以下文件夹。

```
[ec2-user ~]$ cd /usr/share/ec2-instance-connect/
```

3. 在您的实例上运行以下命令。

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

请注意，调用成功导致没有输出。

现在，您可以使用 EC2 Instance Connect 基于浏览器的客户端连接到您的实例。

无法使用 EC2 Instance Connect 连接到 Ubuntu 实例

如果您使用 EC2 Instance Connect 连接到 Ubuntu 实例，并且尝试连接时出错，则可以使用以下信息尝试修复此问题。

可能的原因

实例上的 `ec2-instance-connect` 软件包不是最新版本。

解决方案

将实例上的 `ec2-instance-connect` 软件包更新到最新版本，如下所示：

1. 使用 EC2 Instance Connect 以外的方法连接到实例。
2. 在您的实例上使用以下命令将 `ec2-instance-connect` 软件包更新到最新版本。

```
apt update && apt upgrade
```

我丢失了私有密钥。如何连接到我的实例？

如果丢失由 EBS 支持的实例的私有密钥，您可以重新获取对您的实例的访问权限。您必须停止实例，分离卷并将其作为数据卷附加到另一个实例，然后使用新的公有密钥修改 `authorized_keys` 文件，将卷移回原始实例，并重启实例。有关启动、连接和停止实例的更多信息，请参阅 Amazon EC2 实例状态更改。

此过程仅支持具有 EBS 根卷的实例。如果实例具有实例存储根卷，则无法使用此过程重新获得对实例的访问权限；您必须拥有私有密钥才能连接到实例。要确定实例的根卷类型，请打开 Amazon EC2 控制台，选择实例，选择相应实例，然后选择存储选项卡，并在根设备详细信息部分中，检查根设备类型的值。

该值为 EBS 或 INSTANCE-STORE。

如果丢失私有密钥，除以下步骤外，还有其他方法可以连接到 Linux 实例。有关更多信息，请参阅如果我在 SSH 密钥对初始启动后丢失，该如何连接到 Amazon EC2 实例？

使用不同的密钥对连接由 EBS 支持的实例的步骤

- 步骤 1：创建新的密钥对
- 步骤 2：获取有关原始实例及其根卷的信息
- 步骤 3：停止原始实例
- 步骤 4：启动临时实例
- 步骤 5：从原始实例中分离根卷并将其附加到临时实例
- 步骤 6：将新的公钥添加到已安装到临时实例的原始卷上的 authorized_keys
- 步骤 7：从临时实例中卸载并分离原始卷，然后将其重新附加到原始实例
- 步骤 8：使用新密钥对连接到原始实例
- 步骤 9：清除

步骤 1：创建新的密钥对

使用 Amazon EC2 控制台或第三方工具创建新的密钥对。如果您要将新密钥对的名称设置为与丢失的私有密钥相同的名称，则必须先删除现有密钥对。有关创建新密钥对的信息，请参阅使用 Amazon EC2 创建密钥对或使用第三方工具创建密钥对，并将公有密钥导入 Amazon EC2。

步骤 2：获取有关原始实例及其根卷的信息

请记下以下信息，因为您需要它来完成此过程。

获取有关原始实例的信息

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中选择 Instances (实例)，然后选择要连接到的实例。（我们将此称为原始实例。）
3. 在 Details（详细信息）选项卡上，记下实例 ID 和 AMI ID。
4. 在 Networking（网络）选项卡上，记下可用区。

5. 在 Storage（存储）选项卡的 Root device name（根设备名称）下，记下根卷的设备名称（例如 `/dev/xvda`）。然后，在 Block devices（块存储设备）下，查找此设备的名称并记下卷 ID（例如 vol-0a1234b5678c910de）。

步骤 3：停止原始实例

依次选择实例状态、停止实例。如果此选项处于禁用状态，则表示实例已停止，或者其根卷是实例存储卷。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

步骤 4：启动临时实例

启动临时实例

1. 在导航窗格中，选择 Instances（实例），然后选择 Launch instances（启动实例）。
2. 在 Name and tags（名称和标签）部分，对于 Name（名称），输入 Temporary（临时）。
3. 在 Application and OS Images（应用程序和操作系统映像）部分中，选择您启动原始实例时所用的 AMI。如果此 AMI 不可用，您可以创建一个可在已停止的实例中使用的 AMI。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。
4. 在 Instance type（实例类型）部分中，保留默认的实例类型。
5. 在 Key pair（密钥对）部分中，对于 Key pair name（密钥对名称），选择现有密钥对进行使用或创建新密钥对。
6. 在 Network settings（网络设置）部分，选择 Edit（编辑），然后为 Subnet（子网），选择与原始实例位于同一可用区的子网。
7. 在 Summary（摘要）面板中，选择 Launch（启动）。

步骤 5：从原始实例中分离根卷并将其附加到临时实例

1. 在导航窗格中，选择卷，并选择原始实例的根卷（您已在上一步骤中记下它的卷 ID）。依次选择 Actions（操作）、Detach Volume（分离卷），然后选择 Detach（分离）。等待卷的状态变为 `available`。（您可能需要选择“刷新”图标。）

2. 如果卷仍保持选中状态，则选择 Actions（操作），然后选择 Attach Volume（附加卷）。选择临时实例的实例 ID，记下在 Device name（设备名称）下指定的设备名称（例如 /dev/sdf），然后选择 Attach volume（附加卷）。

> **Note**
>
> 如果已从 Amazon Web Services Marketplace AMI 启动原始实例，并且卷包含 Amazon Web Services Marketplace 代码，则必须先停止临时实例，然后才能附加卷。

步骤 6：将新的公钥添加到已安装到临时实例的原始卷上的 authorized_keys

1. 连接到临时实例。
2. 在临时实例中，挂载附加到实例的卷以访问其文件系统。例如，如果设备名称为 /dev/sdf，请使用以下命令将卷挂载为 /mnt/tempvol。

> **Note**
>
> 您的实例上显示的设备名称可能不同。例如，作为 /dev/sdf 挂载的设备可能在实例上显示为 /dev/xvdf。某些版本的 Red Hat（或其变体，如 CentOS）甚至可能将尾部字母增加 4 个字符，其中 /dev/sd*f* 成为 /dev/xvd*k*。

a. 使用 lsblk 命令确定卷是否已分区。

```
[ec2-user ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
##xvda1 202:1    0    8G  0 part /
xvdf    202:80   0  101G  0 disk
##xvdf1 202:81   0  101G  0 part
xvdg    202:96   0   30G  0 disk
```

在前一个示例中，/dev/xvda 和 /dev/xvdf 是分区卷，而 /dev/xvdg 不是。如果您的卷已分区，则应在后续步骤中挂载分区 (/dev/xvdf1)，而不是原始设备 (/dev/xvdf)。

b. 创建临时目录以挂载卷。

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

c. 使用之前确定的卷名称或设备名称在临时挂载点挂载卷（或分区）。所需命令取决于操作系统的文件系统。请注意，您的实例上显示的设备名称可能不同。有关更多信息，请参阅步骤 6 中的 note。

- Amazon Linux、Ubuntu 和 Debian

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
```

- Amazon Linux 2、CentOS、SUSE Linux 12 和 RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

> **Note**
>
> 如果您收到说明文件系统受损的错误，请运行以下命令以使用 fsck 实用程序检查文件系统并修复任何问题：
>
> ```
> [ec2-user ~]$ sudo fsck /dev/xvdf1
> ```

3. 在临时实例中，借助临时实例 `authorized_keys` 中的新公有密钥，在已挂载卷上使用以下命令更新 `authorized_keys`。

> **Important**
>
> 以下示例使用 Amazon Linux 用户名 `ec2-user`。您可能需要使用其他用户名来替换，例如对于 Ubuntu 实例为 `ubuntu`。

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

如果复制成功，则可以转到下一步骤。

（可选）如果您没有权限编辑 `/mnt/tempvol` 中的文件，您必须使用 sudo 更新文件，然后检查文件的权限，以验证您是否能够登录原始实例。请使用以下命令检查文件权限。

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

```
total 4
-rw------- 1 222 500 398 Sep 13 22:54 authorized_keys
```

在这个输出示例中，*222* 是用户 ID；*500* 是组 ID。接下来，请使用 sudo 重新运行失败的复制命令。

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

再次运行以下命令以确定权限是否已更改。

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

如果用户 ID 和组 ID 已经更改，请使用以下命令进行恢复。

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

步骤 7：从临时实例中卸载并分离原始卷，然后将其重新附加到原始实例

1. 在临时实例中，卸载已附加的卷，以将其重新附加到原始实例。例如，使用以下命令卸载 /mnt/tempvol 处的卷。

   ```
   [ec2-user ~]$ sudo umount /mnt/tempvol
   ```

2. 从临时实例中分离卷（您在上一步中卸载了该卷）：从 Amazon EC2 控制台，在导航窗格中选择卷，为原始实例选择根卷（您在上一步中记下了卷 ID），依次选择操作、分离卷，然后选择分离。等待卷的状态变为 available。（您可能需要选择“刷新”图标。）

3. 将卷重新附加到原始实例：在卷仍保持选中状态时，选择 Actions（操作），然后选择 Attach Volume（附加卷）。选择原始实例的实例 ID，请指定您以前在步骤 2 中记下的原始根卷附加的设备名称（/dev/sda1 或 /dev/xvda），然后选择附加卷。

   > **Important**
   >
   > 如果您不指定与原始附加相同的设备名称，则无法启动原始实例。Amazon EC2 要求根卷位于 sda1 或者 /dev/xvda。

步骤 8：使用新密钥对连接到原始实例

选择原始实例，然后依次选择实例状态、启动实例。在实例进入 `running` 状态后，您可以使用新密钥对的私有密钥文件连接到该实例。

> **Note**
>
> 如果您的新密钥对和相应私有密钥文件的名称不同于原始密钥对的名称，请确保在连接到实例时指定新私有密钥文件的名称。

步骤 9：清除

(可选) 如果您将不再使用临时实例，可以将其终止。选择临时实例，然后依次选择实例状态、终止（删除）实例。

通过失败状态检查来排查 Amazon EC2 Linux 实例问题

如果您的 Linux 实例未能通过状态检查，则以下信息可帮助您排查问题。请首先确定您的应用程序是否存在任何问题。如果您验证的结果是实例没有按照预期运行应用程序，请查看状态检查信息和系统日志。

有关导致状态检查出现故障的问题示例，请参阅 Amazon EC2 实例的状态检查。

目录

- 查看状态检查信息
- 检索系统日志
- 排查 Linux 的实例的系统日志错误
- 内存不足：终止进程
- 错误：mmu_update 失败 (内存管理更新失败)
- I/O 错误（块储存设备故障）
- I/O 错误：既不是本地磁盘也不是远程磁盘（破损的分布式块储存设备）
- request_module：runaway loop modprobe (在较旧的 Linux 版本上循环旧内核 modprobe)
- “严重错误：内核太旧”和“fsck：在尝试打开 /dev 时没有此文件或目录”(内核与 AMI 不匹配)
- “FATAL: Could not load /lib/modules”或者“BusyBox”(内核模块缺失)
- ERROR：无效内核 (EC2 不兼容内核)

- fsck：尝试打开时没有找到此文件或目录... (未找到文件系统)
- 挂载文件系统时出现一般性错误（挂载失败）
- VFS：无法在未知块上挂载根 fs (根文件系统不匹配)
- 错误：无法确定根设备的主/次编号...（根文件系统/设备不匹配）
- XENBUS：设备没有驱动程序...
- ... 没有检查时，已强制执行检查的工作日 (文件系统检查要求)
- fsck 卡在退出状态...（设备缺失）
- GRUB 提示 (grubdom>)
- 提起接口 eth0：设备 eth0 的 MAC 地址与预期不同，驳回。(硬编码的 MAC 地址)。
- 无法加载 SELinux 策略。计算机处于强制执行模式。正在中断。(SELinux 配置错误)
- XENBUS：连接设备时超时 (Xenbus 超时)

查看状态检查信息

使用 Amazon EC2 控制台调查受损实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择您的实例。
3. 选择状态和警报选项卡，查看所有系统状态检查、实例状态检查和附加 EBS 状态检查的各项结果。

如果某种状态检查失败，可以尝试以下一种选项：

- 创建警报来恢复实例，作为对失败状态检查的响应。有关更多信息，请参阅 创建停止、终止、重启或恢复实例的警报。
- （实例状态检查）如果将实例类型更改为基于 Nitro 的实例，则在从没有所需 ENA 和 NVMe 驱动程序的实例迁移时，状态检查将会失败。有关更多信息，请参阅 更改实例类型的兼容性。
- 对于具有 EBS 根卷的实例，请停止并重新启动该实例。有关更多信息，请参阅 启动和停止 Amazon EC2 实例。
- 对于具有实例存储根卷的实例，请终止该实例并启动替换实例。有关更多信息，请参阅 终止 Amazon EC2 实例。
- 等待 Amazon EC2 解决问题。
- 联系 Amazon Web Services 支持 或将问题发布到 Amazon re:Post。

- 如果实例位于自动扩缩组中：
  - （系统状态检查和实例状态检查）默认情况下，Amazon EC2 Auto Scaling 会自动启动替换实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的自动扩缩组中实例的运行状况检查。
  - （附加 EBS 状态检查）必须配置 Amazon EC2 Auto Scaling 来自动启动替换实例。有关更多信息，请参阅《Amazon EC2 Auto Scaling User Guide》中的 Monitor and replace Auto Scaling instances with impaired Amazon EBS volumes。
- 检索系统日志并查找错误。有关更多信息，请参阅 检索系统日志。

检索系统日志

如果实例状态检查失败，则您可以重启实例并检索系统日志。日志能够显示错误之处，从而帮助您诊断问题。重启可清除日志中不必要的信息。

重启实例并检索系统日志

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances，然后选择您的实例。
3. 依次选择 Instance state (实例状态)、Reboot instance (启动实例)。实例重启可能需要几分钟时间。
4. 验证问题是否依然存在；在一些情况下，重启可以解决此问题。
5. 如果实例位于 `running` 状态中，选择实例，依次选择 Actions (操作)、Monitor and troubleshoot (监控和故障排除)、Get system log (获取系统日志)。
6. 查看屏幕上显示的日志，使用下面的已知系统日志错误语句列表来诊断问题。
7. 如果您的问题没有得到解决，您可以将问题发布到 Amazon re:Post。

排查 Linux 的实例的系统日志错误

对于无法通过实例状态检查（例如实例可达性检查）的 Linux 实例，请验证您是否按照上述步骤检索系统日志。以下列表中包含一些常见的系统日志错误，还有一些建议您采取以解决此问题的针对性操作。

内存错误

- 内存不足：终止进程
- 错误：mmu_update 失败 (内存管理更新失败)

设备错误

- I/O 错误（块储存设备故障）
- I/O 错误：既不是本地磁盘也不是远程磁盘（破损的分布式块储存设备）

内核错误

- request_module：runaway loop modprobe (在较旧的 Linux 版本上循环旧内核 modprobe)
- “严重错误：内核太旧”和“fsck：在尝试打开 /dev 时没有此文件或目录”(内核与 AMI 不匹配)
- “FATAL: Could not load /lib/modules”或者“BusyBox”(内核模块缺失)
- ERROR：无效内核 (EC2 不兼容内核)

文件系统错误

- fsck：尝试打开时没有找到此文件或目录... (未找到文件系统)
- 挂载文件系统时出现一般性错误（挂载失败）
- VFS：无法在未知块上挂载根 fs (根文件系统不匹配)
- 错误：无法确定根设备的主/次编号...（根文件系统/设备不匹配）
- XENBUS：设备没有驱动程序...
- ... 没有检查时，已强制执行检查的工作日 (文件系统检查要求)
- fsck 卡在退出状态...（设备缺失）

操作系统错误

- GRUB 提示 (grubdom>)
- 提起接口 eth0：设备 eth0 的 MAC 地址与预期不同，驳回。(硬编码的 MAC 地址)。
- 无法加载 SELinux 策略。计算机处于强制执行模式。正在中断。(SELinux 配置错误)
- XENBUS：连接设备时超时 (Xenbus 超时)

内存不足：终止进程

指示内存不足错误的系统日志条目与下方显示的内容类似。

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
```

```
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

潜在原因

内存耗尽

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	请执行下列操作之一： • 停止并修改实例以使用不同的实例类型，然后再次启动实例。例如，一个更大或内存优化型实例类型。 • 重启实例以使其恢复未受损状态。除非您更改实例类型，否则该问题可能还会出现。
实例存储支持的	请执行下列操作之一： • 终止实例并启动新实例，指定一个不同的实例类型。例如，一个更大或内存优化型实例类型。 • 重启实例以使其恢复未受损状态。除非您更改实例类型，否则该问题可能还会出现。

错误：mmu_update 失败 (内存管理更新失败)

表示内存管理更新故障的系统日志条目与以下示例类似：

```
...
Press `ESC' to enter the menu... 0   [H[J  Booting 'Amazon Linux 2011.09
 (2.6.35.14-95.38.amzn1.i686)'


root (hd0)
```

```
 Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=

en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22
```

潜在原因

Amazon Linux 的问题

建议采取的措施

将问题发布到 Amazon re:Post 或联系 Amazon Web Services 支持。

I/O 错误（块储存设备故障）

表示输入/输出错误的系统日志条目类似于以下示例：

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

潜在原因

实例类型	潜在原因
由 Amazon EBS 支持	发生故障的 Amazon EBS 卷
实例存储支持的	发生故障的物理驱动器

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止实例。 2. 分离该卷。 3. 尝试恢复该卷。 **Note** 最好的做法是经常拍摄 Amazon EBS 卷的快照。这样能大幅降低因故障而导致数据丢失的风险。 4. 重新将卷附加到实例。 5. 启动实例。
实例存储支持的	终止实例并启动新的实例。 **Note** 无法恢复数据。从备份恢复。

对于此实例类型	请执行该操作
	Note 比较好的做法是使用 Amazon S3 或 Amazon EBS 进行备份。实例存储卷是直接与单个主机和磁盘故障相关的。

I/O 错误：既不是本地磁盘也不是远程磁盘（破损的分布式块储存设备）

表示设备的输入/输出错误的系统日志条目类似于以下示例：

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...

Aborting journal on device drbd1-8.

block drbd1: IO ERROR: neither local nor remote disk

Buffer I/O error on device drbd1, logical block 557056

lost page write due to I/O error on drbd1

JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

潜在原因

实例类型	潜在原因
由 Amazon EBS 支持	发生故障的 Amazon EBS 卷
实例存储支持的	发生故障的物理驱动器

建议采取的措施

终止实例并启动新的实例。

对于由 Amazon EBS 支持的实例，您可以从最近拍摄的快照恢复数据，方法是从该快照创建映像。快照之后添加的任何数据都无法恢复。

request_module：runaway loop modprobe (在较旧的 Linux 版本上循环旧内核 modprobe)

表示此条件的系统日志类似于下方显示的示例。使用不稳定或陈旧的 Linux 内核 (如 2.6.16-xenU) 可能会在启动时导致无法终止的循环环境。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

 Xen: 0000000000000000 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	使用以下其中一个选项可使用较新的内核 (基于 GRUB 的内核或静态内核)。 选项 1：终止实例并启动新实例，指定 `-kernel` 和 `-ramdisk` 参数。 选项 2：

对于此实例类型	请执行该操作
	1. 停止实例。 2. 修改内核和虚拟磁盘的属性以使用较新的内核。 3. 启动实例。
实例存储支持的	终止实例并启动新实例，指定 `-kernel` 和 `-ramdisk` 参数。

“严重错误：内核太旧”和“fsck：在尝试打开 /dev 时没有此文件或目录”(内核与 AMI 不匹配)

表示此条件的系统日志类似于下方显示的示例。

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

潜在原因

不可兼容的内核和用户空间

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止实例。 2. 修改配置以使用较新的内核。 3. 启动实例。
实例存储支持的	执行以下步骤： 1. 创建使用较新内核的 AMI。

对于此实例类型	请执行该操作
	2. 终止实例。 3. 从您创建的 AMI 中启动新实例。

“FATAL: Could not load /lib/modules”或者“BusyBox”(内核模块缺失)

表示此条件的系统日志类似于下方显示的示例。

```
[    0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or
 directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No
 such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device.  Common problems:
 - Boot args (cat /proc/cmdline)
   - Check rootdelay= (did the system wait long enough?)
   - Check root= (did the system wait for the right device?)
 - Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
ALERT! /dev/sda1 does not exist. Dropping to a shell!
```

```
BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

潜在原因

以下一个或多个条件可能会导致此问题：

- 虚拟磁盘缺失
- 缺少正确的虚拟磁盘模块
- Amazon EBS 根卷没有正确附加为 `/dev/sda1`

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 为 Amazon EBS 卷选择经过纠正的虚拟磁盘。 2. 停止实例。 3. 分离并修复该卷。 4. 将卷附加到实例。 5. 启动实例。 6. 修改 AMI 以使用经过纠正的虚拟磁盘。
实例存储支持的	执行以下步骤： 1. 终止此实例，然后启动包含正确虚拟磁盘的新实例。 2. 创建包含正确虚拟磁盘的新 AMI。

ERROR：无效内核 (EC2 不兼容内核)

表示此条件的系统日志类似于下方显示的示例。

```
...
root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

潜在原因

以下一个或两个条件都可能会导致此问题：

- GRUB 不支持所提供的内核
- 后备内核不存在

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤：

对于此实例类型	请执行该操作
	1. 停止实例。 2. 替换为正在工作的内核。 3. 安装后备内核。 4. 通过纠正内核修改 AMI。
实例存储支持的	执行以下步骤： 1. 终止此实例，然后启动包含正确内核的新实例。 2. 创建包含正确内核的 AMI。 3. （可选）通过 Amazon Web Services 支持 寻求技术支持以便恢复数据。

fsck：尝试打开时没有找到此文件或目录... (未找到文件系统)

表示此条件的系统日志类似于下方显示的示例。

```
  Welcome to Fedora
  Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [  OK  ]

Starting udev: [  OK  ]

Setting hostname localhost:  [  OK  ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[  OK  ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh
```

```
/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]


*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

潜在原因

- 虚拟磁盘文件系统定义 /etc/fstab 中存在错误
- /etc/fstab 中存在配置错误的文件系统定义
- 硬盘丢失/故障

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止实例，分离根卷，修补/修改 /etc/fstab 该卷，将其附加到实例，然后启动该实例。 2. 修改虚拟磁盘以使其包含经过修改的 /etc/fstab（如果适用）。 3. 修改 AMI 以使用较新的虚拟磁盘。 fstab 中的第 6 个字段定义此安装的可用性要求，非零值暗示将在该卷上执行文件系统检查并且必须 成功完成。能否在 Amazon EC2 中使用此字段还不确定，因为故障一般会导致交互性控

对于此实例类型	请执行该操作
	制台提示信息，但是目前此功能在 Amazon EC2 中尚不可用。请谨慎使用此功能，并阅读 Linux man 页面了解有关 fstab 的信息。
实例存储支持的	执行以下步骤： 1. 终止实例并启动新的实例。 2. 将所有不正确 Amazon EBS 卷与重启的实例分离。 3. （可选）通过 Amazon Web Services 支持 寻求技术支持以便恢复数据。

挂载文件系统时出现一般性错误（挂载失败）

表示此条件的系统日志类似于下方显示的示例。

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting.  Commit interval 5 seconds

EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
```

```
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

潜在原因

实例类型	潜在原因
由 Amazon EBS 支持	• 分离或出故障的 Amazon EBS 卷。 • 文件系统损坏。 • 匹配错误的内存虚拟磁盘与 AMI 组合 (如 Debian 内存虚拟磁盘和 SUSE AMI)。
实例存储支持的	• 发生故障的驱动器。 • 损坏的文件系统。 • 匹配错误的虚拟磁盘和组合 (例如，Debian 虚拟磁盘和 SUSE AMI)。

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止实例。 2. 分离根卷。 3. 将根卷附加到已知正在工作的实例。

对于此实例类型	请执行该操作
	4. 执行文件系统检查（fsck -a /dev/...）。 5. 修正所有错误。 6. 从已知正在工作的实例分离卷。 7. 将卷附加到已停止的实例。 8. 启动实例。 9. 重新检查实例的状态。
实例存储支持的	请尝试以下任一操作： • 启动新实例。 • （可选）通过 Amazon Web Services 支持 寻求技术支持以便恢复数据。

VFS：无法在未知块上挂载根 fs (根文件系统不匹配)

表示此条件的系统日志类似于下方显示的示例。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
 20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line:  root=/dev/sda1 ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

潜在原因

实例类型	潜在原因
由 Amazon EBS 支持	• 设备附加错误。 • 根设备没有附加到正确的设备点。 • 文件系统不是预期的格式。 • 使用旧内核 (如 2.6.16-XenU)。

实例类型	潜在原因
	• 实例上的近期更新内核 (错误更新或更新错误)
实例存储支持的	硬件设备故障。

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	请执行下列操作之一： • 停止实例，然后再重启。 • 修改根卷以附加到正确的设备点，可能是 /dev/sda1，而不是 /dev/sda。 • 停止并修改新内核。 • 请参阅您的 Linux 发行版的文档以检查是否有已知更新错误。更改或重新安装内核。
实例存储支持的	终止实例并使用新内核启动新实例。

错误：无法确定根设备的主/次编号...（根文件系统/设备不匹配）

表示此条件的系统日志类似于下方显示的示例。

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
```

```
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

潜在原因

- 虚拟块储存设备驱动程序缺失或配置错误
- 设备枚举冲突 (sda 与 xvda，或是 sda 而不是 sda1)
- 实例内核选择错误

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止实例。 2. 分离该卷。 3. 修正设备映射问题。 4. 启动实例。 5. 修改 AMI 以解决设备映射问题。
实例存储支持的	执行以下步骤： 1. 创建附有适当补丁程序的新 AMI (正确映射块储存设备)。 2. 终止实例并从您创建的 AMI 中启动新实例。

XENBUS：设备没有驱动程序...

表示此条件的系统日志类似于下方显示的示例。

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
```

```
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

潜在原因

- 虚拟块储存设备驱动程序缺失或配置错误
- 设备枚举冲突 (sda 与 xvda)
- 实例内核选择错误

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止实例。 2. 分离该卷。 3. 修正设备映射问题。 4. 启动实例。 5. 修改 AMI 以解决设备映射问题。
实例存储支持的	执行以下步骤： 1. 创建附有适当补丁程序的 AMI (正确映射块储存设备)。 2. 终止实例并使用您创建的 AMI 启动新实例。

... 没有检查时，已强制执行检查的工作日 (文件系统检查要求)

表示此条件的系统日志类似于下方显示的示例。

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

潜在原因

文件系统检查时间已过；正在强制执行文件系统检查。

建议采取的措施

- 耐心等候文件系统检查的完成。文件系统检查可能需要很长一段时间，具体取决于根文件系统的大小。
- 使用 tune2fs 或适合您的文件系统的工具修改文件系统，以去除强制执行文件系统检查 (fsck) 的功能。

fsck 卡在退出状态...（设备缺失）

表示此条件的系统日志类似于下方显示的示例。

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

潜在原因

- 为缺失的磁盘查找虚拟磁盘
- 强制执行文件系统一致性检查

- 磁盘故障或者已分离

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	尝试以下一个或多个措施以解决此问题： • 停止实例，将该卷附加到正在运行的实例。 • 手动运行一致性检查。 • 修正虚拟磁盘以使其包含相关实用程序。 • 修改文件系统调整参数以删除一致性要求 (不推荐)。
实例存储支持的	尝试以下一个或多个措施以解决此问题： • 通过正确的工具作业重新绑定虚拟磁盘。 • 修改文件系统调整参数以删除一致性要求 (不推荐)。 • 终止实例并启动新的实例。 • （可选）通过 Amazon Web Services 支持 寻求技术支持以便恢复数据。

GRUB 提示 (grubdom>)

表示此条件的系统日志类似于下方显示的示例。

```
    GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

       [ Minimal BASH-like line editing is supported.   For

         the   first   word,  TAB  lists  possible  command

         completions.  Anywhere else TAB lists the possible

         completions of a device/filename. ]
```

```
grubdom>
```

潜在原因

实例类型	潜在原因
由 Amazon EBS 支持	• 缺少 GRUB 配置文件。 • 使用了错误的 GRUB 映像，应使用不同位置的 GRUB 配置文件。 • 使用了不受支持的文件系统存储您的 GRUB 配置文件 (例如，将您的根文件系统转换为 GRUB 早期版本不支持的类型)。
实例存储支持的	• 缺少 GRUB 配置文件。 • 使用了错误的 GRUB 映像，应使用不同位置的 GRUB 配置文件。 • 使用了不受支持的文件系统存储您的 GRUB 配置文件 (例如，将您的根文件系统转换为 GRUB 早期版本不支持的类型)。

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	选项 1：修改 AMI 并重启实例： 1. 修改源 AMI 以便在标准位置 (/boot/grub/menu.lst) 创建 GRUB 配置文件。 2. 验证您的 GRUB 版本支持基础文件系统类型，并根据需要升级 GRUB。 3. 选择合适的 GRUB 映像 (hd0 – 第一个磁盘或 hd00 – 第一个磁盘，第一个分区)。 4. 终止实例并使用您创建的 AMI 启动一个新实例。

对于此实例类型	请执行该操作
	选项 2：修复现有实例： 1. 停止实例。 2. 分离根卷文件系统。 3. 将根卷文件系统附加到已知正在工作的实例。 4. 挂载文件系统。 5. 创建 GRUB 配置文件。 6. 验证您的 GRUB 版本支持基础文件系统类型，并根据需要升级 GRUB。 7. 分离文件系统。 8. 附加到原始实例。 9. 修改内核属性以便使用正确的 GRUB 映像 (第 1 个磁盘或其上的第 1 个分区)。 10启动实例。
实例存储支持的	选项 1：修改 AMI 并重启实例： 1. 使用位于标准位置 (/boot/grub/menu.lst) 的 GRUB 配置文件创建新 AMI。 2. 选择合适的 GRUB 映像 (hd0 – 第一个磁盘或 hd00 – 第一个磁盘，第一个分区)。 3. 验证您的 GRUB 版本支持基础文件系统类型，并根据需要升级 GRUB。 4. 终止实例并使用您创建的 AMI 启动新实例。 选项 2：终止此实例并启动新实例，指定正确的内核。 **Note** 要从现有实例恢复数据，请联系 Amazon Web Services 支持。

提起接口 eth0：设备 eth0 的 MAC 地址与预期不同，驳回。(硬编码的 MAC 地址)。

表示此条件的系统日志类似于下方显示的示例。

```
...
Bringing up loopback interface:  [  OK  ]

Bringing up interface eth0:  Device eth0 has different MAC address than expected,
 ignoring.
[FAILED]

Starting auditd: [  OK  ]
```

潜在原因

AMI 配置中存在硬编码接口 MAC

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	请执行下列操作之一： • 修改 AMI 以删除硬编码并重启实例。 • 修改实例以删除硬编码 MAC 地址。 OR 执行以下步骤： 1. 停止实例。 2. 分离根卷。 3. 将卷附加到另一个实例并修改卷以删除硬编码 MAC 地址。 4. 将卷附加到原始实例。 5. 启动实例。

对于此实例类型	请执行该操作
实例存储支持的	请执行下列操作之一： • 修改实例以删除硬编码 MAC 地址。 • 终止实例并启动新的实例。

无法加载 SELinux 策略。计算机处于强制执行模式。正在中断。(SELinux 配置错误)

表示此条件的系统日志类似于下方显示的示例。

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

潜在原因

SELinux 已在错误的情况下启动：

- GRUB 不支持所提供的内核
- 后备内核不存在

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	执行以下步骤： 1. 停止失败的实例。 2. 分离失败的实例的根卷。 3. 将根卷附加到另一个 Linux 的运行实例 (之后称为“恢复实例”)。 4. 连接到恢复实例并挂载失败的实例的根卷。

对于此实例类型	请执行该操作
	5. 在挂载的根卷上禁用 SELinux。此过程因 Linux 分配而异；有关更多信息，请参阅特定于操作系统的文档。 **Note** 在某些系统上，可通过在 `SELINUX=disabled` 文件中设置 `/mount_point /etc/sysconfig/selinux` 来禁用 SELinux (其中，*mount_point* 是您在恢复实例上安装卷的位置)。 6. 从恢复实例卸载和分离根卷并将该根卷重新附加到原始实例。 7. 启动实例。
实例存储支持的	执行以下步骤： 1. 终止实例并启动新的实例。 2. （可选）通过 Amazon Web Services 支持 寻求技术支持以便恢复数据。

XENBUS：连接设备时超时 (Xenbus 超时)

表示此条件的系统日志类似于下方显示的示例。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found.  Try passing init= option to kernel.
```

潜在原因

- 块储存设备未连接到实例
- 此实例使用的是旧实例内核

建议采取的措施

对于此实例类型	请执行该操作
由 Amazon EBS 支持	请执行下列操作之一： • 修改 AMI 和实例以便使用新内核并重启实例。 • 重启实例。
实例存储支持的	请执行下列操作之一： • 终止实例。 • 修改 AMI 以使用新实例，使用此 AMI 启动新实例。

排查从错误的卷启动 Amazon EC2 Linux 实例的问题

在某些情况下，某个并非附加到 /dev/xvda 或 /dev/sda 的卷成为了 Linux 实例的根卷。当您将另一个实例的根卷或从某个根卷的快照中创建的卷附加到带有现有根卷的实例时，可能会发生这种情况。

这是由于 Linux 中的初始虚拟磁盘的工作方式导致的。它将选择在 / 中定义为 /etc/fstab 的卷，在某些发行版中，这是由附加到卷分区的标签决定的。具体来说，您将发现您的 /etc/fstab 与下面类似：

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

如果您检查两个卷的标签，您将看到它们都包含 / 标签：

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

在此示例中，您最终可以让 /dev/xvdf1（而不是您打算从中启动实例的 /dev/xvda1 卷）成为您的实例在初始虚拟磁盘运行后从中启动的根卷。要解决此问题，请使用相同的 e2label 命令更改您不想从中启动实例的附加卷的标签。

某些情况下，在 /etc/fstab 中指定 UUID 可以解决此问题。但是，如果两个卷来自同一快照，或者辅助卷是通过主卷的快照创建的，则它们将共享 UUID。

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

更改已附加 ext4 卷的标签

1. 使用 e2label 命令将卷的标签更改为 / 之外的其他标签。

   ```
   [ec2-user ~]$ sudo e2label /dev/xvdf1 old/
   ```

2. 验证卷是否有新标签。

   ```
   [ec2-user ~]$ sudo e2label /dev/xvdf1
   old/
   ```

更改已附加 xfs 卷的标签

- 使用 xfs_admin 命令将卷的标签更改为 / 之外的其他标签。

  ```
  [ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
  writing all SBs
  new label = "old/"
  ```

如上所示更改卷标签后，您应该能够重新引导实例并在实例引导时让初始虚拟磁盘选择适当的卷。

> **Important**
>
> 如果您要分离使用新标签的卷并将它返回到另一实例以用作根卷，则必须再次执行上述过程并将卷标签更改回其原始值。否则，其他实例将不会启动，因为内存虚拟磁盘无法找到标签为 / 的卷。

排查连接到 Amazon EC2 Windows 实例时遇到的问题

以下信息和常见错误可以帮助您排查连接到 Windows 实例时出现的问题。

连接问题

- 远程桌面无法连接到远程计算机
- 使用 macOS RDP 客户端时出错
- RDP 显示黑屏而不是桌面
- 无法使用非管理员用户远程登录到实例
- 使用 Amazon Systems Manager 解决远程桌面问题
- 在具有远程注册表的 EC2 实例上启用远程桌面
- 我丢失了私有密钥。我怎样才能连接到我的 Windows 实例？

远程桌面无法连接到远程计算机

尝试以下操作以解决与连接实例有关的问题：

- 确认您使用的是正确的公有 DNS 主机名。(在 Amazon EC2 控制台中，请选择实例，并在详细信息窗格中查看 Public DNS (IPv4) (公有 DNS (IPv4))。) 如果您的实例是在 VPC 中，并且您没有看到公有 DNS 名称，则必须启用 DNS 主机名。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。
- 验证您的实例是否具有公有 IPv4 地址。如果没有，您可以将弹性 IP 地址与您的实例关联。有关更多信息，请参阅弹性 IP 地址。
- 要使用 IPv6 地址连接到您的实例，请检查本地计算机是否具有 IPv6 地址以及是否已配置为可使用 IPv6。有关更多信息，请参阅《Amazon VPC 用户指南》中的 在实例中配置 IPv6。
- 确认安全组具有允许端口 3389 RDP 访问的规则。
- 如果您输入复制的密码，然后收到错误消息 `Your credentials did not work`，请尝试根据系统提示手动键入密码。可能是复制密码时丢失了字符或添加了额外空格。

- 确认实例已通过状态检查。有关更多信息，请参阅Amazon EC2 实例的状态检查和the section called “Linux 实例失败状态检查”。
- 确认子网的路由表具有这样一个路由，该路由将发往 VPC 外部的所有流量发送到该 VPC 的 Internet 网关。有关更多信息，请参阅 Amazon VPC 用户指南 中的创建自定义路由表（Internet 网关）。
- 确认 Windows 防火墙或其他防火墙软件没有阻止到实例的 RDP 流量。我们建议您禁用 Windows 防火墙，使用安全组规则来控制对实例的访问。尝试以下选项之一：
  - 可使用 AWSSupport-TroubleshootRDP 来disable the Windows Firewall profiles using SSM Agent。此选项要求为 Amazon Systems Manager 配置实例。
  - 使用 AWSSupport-ExecuteEC2Rescue。
  - 停止实例，分离根卷，然后将该卷作为数据卷附加到临时实例。连接到临时实例，将该卷联机。从数据卷加载注册表配置单元。在 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicyStandardProfile` 下方，将 `EnableFirewall` 设置为 0。卸载注册表配置单元，然后将卷脱机。从临时实例中分离卷并将其作为根卷附加到原始实例。启动原始实例。
- 验证是否在不属于 Active Directory 域的实例上禁用网络级身份验证（使用 AWSSupport-TroubleshootRDP 来 disable NLA）。
- 验证远程桌面服务（TermService）启动类型是否为“自动”，服务是否启动（使用 AWSSupport-TroubleshootRDP 来 enable and start the RDP service）。
- 验证是否连接到正确的远程桌面协议端口，默认情况下为 3389（使用 AWSSupport-TroubleshootRDP 来 read the current RDP port 和 change it back to 3389）。
- 验证实例是否允许远程桌面连接（使用 AWSSupport-TroubleshootRDP 来 enable Remote Desktop connections）。
- 确认密码没有过期。如果密码已过期，可以重置密码。有关更多信息，请参阅 重置 Amazon EC2 Windows 实例的 Windows 管理员密码。
- 如果您尝试使用在实例上创建的用户进行连接并收到错误消息 `The user cannot connect to the server due to insufficient access privileges`，请确认您已为该用户授予本地登录权限。有关更多信息，请参阅向成员授予本地登录权限。
- 如果尝试次数超过所允许的并发 RDP 会话的最大数目，则会终止会话，并显示消息 `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.`。默认情况下，实例可以有两个并发 RDP 会话。

使用 macOS RDP 客户端时出错

如果使用 Microsoft 网站上的远程桌面连接客户端连接到 Windows Server 实例，则可能会出现以下错误：

```
Remote Desktop Connection cannot verify the identity of the computer that you want to
 connect to.
```

从 Mac 应用商店下载 Microsoft 远程桌面应用程序，然后使用该应用程序连接到实例。

RDP 显示黑屏而不是桌面

请尝试以下操作解决该问题：

- 检查控制台输出有无其他信息。要使用 Amazon EC2 控制台获取您的实例的控制台输出，请选择实例，然后依次选择 Actions (操作)、Monitor and troubleshoot (显示屏和问题排查)、Get System Log (获取系统日志)。
- 确认您正在运行最新版本的 RDP 客户端。
- 尝试使用 RDP 客户端的默认设置。
- 如果您要使用远程桌面连接，请尝试使用 `/admin` 选项，通过以下方式将其启动。

  ```
  mstsc /v:instance /admin
  ```

- 如果服务器正在运行全屏应用程序，则它可能会停止响应。使用 Ctrl+Shift+Esc 以启动 Windows 任务管理器，然后关闭应用程序。
- 如果过度使用服务器，则服务器可能已停止响应。要使用 Amazon EC2 控制台来监控实例，请选择实例，然后选择 Monitoring (监控) 选项卡。如果需要将实例类型更改为更大大小，请参阅Amazon EC2 实例类型更改。

无法使用非管理员用户远程登录到实例

如果无法通过非管理员用户远程登录到 Windows 实例，请确保您已为该用户授予本地登录权限。请参阅为用户或组授予本地登录到域中的域控制器的权限。

使用 Amazon Systems Manager 解决远程桌面问题

您可以使用 Amazon Systems Manager 解决使用 RDP 连接到 Windows 实例的问题。

Amazon Web Services Support-TroubleshootRDP

利用 Amazon Web Services Support-TroubleshootRDP 自动化文档，用户可以检查或修改目标实例上可影响远程桌面协议 (RDP) 连接的常规设置，如 RDP 端口、网络层身份验证 (NLA) 和 Windows 防火墙配置文件。默认情况下，此文档读取和输出这些设置的值。

Amazon Web Services Support-TroubleshootRDP 自动化文档可与 EC2 实例、本地实例和虚拟机 (VM)（已允许它们与 Amazon Systems Manager（托管实例）一起使用）一起使用。此外，它还可以不 允许与 Systems Manager 一起使用的适用于 Windows 服务器的 EC2 实例一起使用。有关启用实例以与 Amazon Systems Manager 一起使用的信息，请参阅《Amazon Systems Manager 用户指南》中的 托管式节点。

使用 Amazon Web Services Support-TroubleshootRDP 文档进行故障排除

1. 登录 Systems Manager 控制台。
2. 确认您与受损 实例位于同一区域。
3. 从左侧导航窗格中选择 Documents（文档）。
4. 在 Owned by Amazon（由 Amazon 拥有）选项卡上，在搜索字段中输入 `AWSSupport-TroubleshootRDP`。`AWSSupport-TroubleshootRDP` 文档显示时，选择该文档。
5. 选择执行自动化。
6. 对于 Execution Mode (执行模式)，选择 Simple execution (简单执行)。
7. 对于 Input parameters (输入参数) InstanceId，启用Show interactive instance picker (显示交互式实例选取器)。
8. 选择您的 Amazon EC2 实例。
9. 检查示例，然后选择 Execute (执行)。
10. 要监控执行进度，对于 Execution status (执行状态)，等待状态从 Pending (待处理) 变为 Success (成功)。展开 Outputs (输出) 以查看结果。要查看各个步骤的输出，请在 Executed Steps (执行步骤) 中，选择 Step ID (步骤 ID)。

Amazon Web Services Support-TroubleshootRDP 示例

以下示例介绍如何使用 Amazon Web Services Support-TroubleshootRDP 完成常见的故障排除任务。您可以使用示例 Amazon CLI start-automation-execution 命令或提供的 Amazon Web Services 管理控制台 链接。

Example示例：检查当前的 RDP 状态

Amazon CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom" --region region_code
```

Amazon Systems Manager 控制台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region#documentVersion=$LATEST
```

Example示例：禁用 Windows 防火墙

Amazon CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=i-1234567890abcdef0, Action=Custom, Firewall=Disable" --
region region_code
```

Amazon Systems Manager 控制台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&Firewall=Disable
```

Example示例：禁用网络级别身份验证

Amazon CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, NLASettingAction=Disable" --
region region_code
```

Amazon Systems Manager 控制台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion
```

Example示例：将 RDP 服务启动类型设置为自动并启动 RDP 服务

Amazon CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPServiceStartupType=Auto,
 RDPServiceAction=Start" --region region_code
```

Amazon Systems Manager 控制台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/
AWSSupport-TroubleshootRDP?region=region_code#documentVersion=
$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example示例：恢复默认 RDP 端口 (3389)

Amazon CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPPortAction=Modify" --
region region_code
```

Amazon Systems Manager 控制台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RDPPortAction=Modify
```

Example示例：允许远程连接

Amazon CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RemoteConnections=Enable" --
region region_code
```

Amazon Systems Manager 控制台：

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RemoteConnections=Enable
```

Amazon Web Services Support-ExecuteEC2Rescue

AWSSupport-ExecuteEC2Rescue 自动化文档使用 EC2Rescue for Windows Server 自动排查并修复 EC2 实例连接和 RDP 问题。有关更多信息，请参阅 在无法访问的实例上运行 EC2Rescue 工具。

Amazon Web Services Support-ExecuteEC2Rescue 自动化文档需要停止并重新启动实例。Systems Manager Automation 将停止实例并创建 Amazon Machine Image (AMI) (Amazon 系统映像)。实例存储卷中存储的数据将会丢失。如果未使用弹性 IP 地址，公有 IP 地址将发生变化。有关更多信息，请参阅《Amazon Systems Manager 用户指南》中的 在无法访问的实例上运行 EC2Rescue 工具。

使用 Amazon Web Services Support-ExecuteEC2Rescue 文档进行故障排除

1. 打开 Systems Manager 控制台。
2. 确认您与受损 Amazon EC2 实例位于同一区域。
3. 在导航面板中，选择文档。
4. 搜索并选择 `AWSSupport-ExecuteEC2Rescue` 文档，然后选择执行自动化。
5. 在 Execution Mode (执行模式) 中，选择 Simple execution (简单执行)。
6. 在 Input parameters (输入参数) 部分的 UnreachableInstanceId 中，输入无法访问的实例的 Amazon EC2 实例 ID。
7. （可选）对于 LogDestination，请输入 Amazon Simple Storage Service (Amazon S3) 存储桶名称（如果需要收集操作系统日志以便排查 Amazon EC2 实例问题）。日志会自动上传到此指定存储桶。
8. 选择 Execute (执行)。
9. 要监控执行进度，请在 Execution (执行) 状态中，等待状态从 Pending (待处理) 变为 Success (成功)。展开 Outputs (输出) 以查看结果。要查看各个步骤的输出，请在 Executed Steps (执行步骤) 中，选择 Step ID (步骤 ID)。

在具有远程注册表的 EC2 实例上启用远程桌面

如果无法访问的实例未由 Amazon Systems Manager 会话管理器管理，则可以使用远程注册表启用远程桌面。

1. 从 EC2 控制台停止无法访问的实例。
2. 分离无法访问实例的根卷并将其附加到与存储卷位于同一可用区中的可访问实例。如果您在同一可用区中没有可访问的实例，请启动一个实例。记下无法访问的实例上根卷的设备名称。
3. 在可访问的实例上，打开磁盘管理。您可以通过在“命令提示符”窗口中运行以下命令来执行此操作。

```
diskmgmt.msc
```

4. 右键单击来自无法访问的实例的新附加卷，然后选择在线。
5. 打开 Windows 注册表编辑器。您可以通过在“命令提示符”窗口中运行以下命令来执行此操作。

```
regedit
```

6. 在注册表编辑器中，选择 HKEY_LOCAL_MACHINE，然后选择文件、加载配置单元。
7. 选择附加卷的驱动器，导航到 `\Windows\System32\config\`，选择 `SYSTEM`，然后选择 Open (打开)。
8. 对于 Key Name (密钥名称)，输入配置单元的唯一名称，然后选择 OK (确定)。
9. 在对注册表进行任何更改之前，备份注册表配置单元。
   a. 在注册表编辑器控制台树中，选择您加载的配置单元：HKEY_LOCAL_MACHINE*your-key-name*。
   b. 选择文件、导出。
   c. 在 Export Registry File (导出注册表文件) 对话框中，选择要保存备份副本的位置，然后在 File name (文件名) 字段中键入备份文件的名称 。
   d. 选择保存。
10. 在注册表编辑器中，导航到 `HKEY_LOCAL_MACHINE\`*`your key name`*`\ControlSet001\Control\Terminal Server`，然后在详细信息窗格中双击 fDenyTSConnections。
11. 在 Edit DWORD (编辑 DWORD) 值框中，在 Value data (值数据) 字段中输入 `0`。
12. 选择 OK。

> **Note**
>
> 如果 Value data (值数据) 字段中的值为 `1`，则实例将拒绝远程桌面连接。值为 `0` 允许远程桌面连接。

13. 在注册表编辑器中，选择 HKEY_LOCAL_MACHINE*your-key-name*，然后选择文件、卸载配置单元。
14. 关闭注册表编辑器和磁盘管理。
15. 在 EC2 控制台中，将根卷与可访问的实例分离，然后将其重新附加到无法访问的实例。将卷附加到无法访问的实例时，请在设备字段中输入您之前保存的设备名称。
16. 重新启动无法访问的实例。

我丢失了私有密钥。我怎样才能连接到我的 Windows 实例？

当您连接到新启动的 Windows 实例时，您需要使用在启动实例时指定的密钥对的私有密钥对管理员账户的密码进行解密。

如果丢失了管理员密码并且不再具有私有密钥，您必须重置密码或创建新的实例。有关更多信息，请参阅 重置 Amazon EC2 Windows 实例的 Windows 管理员密码。有关使用 Systems Manager 文档重置密码的步骤，请参阅《Amazon Systems Manager 用户指南》中的重置 Amazon EC2 实例上的密码和 SSH 密钥。

排查 Amazon EC2 Windows 实例的启动问题

以下问题排查技巧有助于解决与 Amazon EC2 Windows 实例相关的密码和激活问题。

问题

- "密码不可用"
- "密码尚不可用"
- "无法检索 Windows 密码"
- "等待元数据服务"
- "无法激活 Windows"
- "Windows 不是正版 (0x80070005)"
- "没有终端服务器许可服务器可提供许可证"
- “某些设置由您的组织管理”

"密码不可用"

要使用远程桌面连接到 Windows 实例，必须指定账户和密码。提供的账户和密码基于用于启动实例的 AMI。您可以为管理员账户检索自动生成的密码，也可以使用在创建该 AMI 的原始实例中所用的账户和密码。

您可以为使用自定义 Windows AMI 启动的实例生成管理员账户密码。要生成密码，您需要在创建 AMI 之前在操作系统中配置一些设置。有关更多信息，请参阅 创建 Amazon EBS-backed AMI。

如果未将 Windows 实例配置为生成随机密码，则在使用控制台检索自动生成的密码时会收到以下消息：

```
Password is not available.
The instance was launched from a custom AMI, or the default password has changed. A
password cannot be retrieved for this instance. If you have forgotten your password,
 you can
reset it using the Amazon EC2 configuration service. For more information, see
 Passwords for a
Windows Server instance.
```

检查实例的控制台输出，查看用于启动实例的 AMI 是不是在禁用了密码生成的情况下创建的。如果禁用了密码生成，则控制台输出包含以下内容：

```
Ec2SetPassword: Disabled
```

如果禁用了密码生成并且您未记住原始实例的密码，则可以重置该实例的密码。有关更多信息，请参阅重置 Amazon EC2 Windows 实例的 Windows 管理员密码。

"密码尚不可用"

要使用远程桌面连接到 Windows 实例，必须指定账户和密码。提供的账户和密码基于用于启动实例的 AMI。您可以为管理员账户检索自动生成的密码，也可以使用在创建该 AMI 的原始实例中所用的账户和密码。

您的密码应在几分钟内可用。如果密码不可用，则在使用控制台检索自动生成的密码时会收到以下消息：

```
Password not available yet.
Please wait at least 4 minutes after launching an instance before trying to retrieve
 the
auto-generated password.
```

如果超过四分钟后您仍然无法获取密码，则您的实例的启动代理可能未配置为生成密码。请根据控制台输出是否为空，对此进行验证。有关更多信息，请参阅无法获取控制台输出。

另请验证用于访问管理门户的 Amazon Identity and Access Management（IAM）账户是否允许执行 `ec2:GetPasswordData` 操作。有关 IAM 权限的更多信息，请参阅什么是 IAM？。

"无法检索 Windows 密码"

要为管理员账户检索自动生成的密码，您必须针对启动实例时指定的密钥对使用私有密钥。如果在启动实例时未指定密钥对，您将会收到以下消息。

```
Cannot retrieve Windows password
```

您可以终止该实例并使用相同的 AMI 启动新实例，并确保指定密钥对。

"等待元数据服务"

Windows 实例必须包含其实例元数据中的信息才能将自身激活。默认情况下，`WaitForMetaDataAvailable` 设置可确保 EC2Config 服务等待实例元数据可以访问，然后继续启动过程。有关更多信息，请参阅使用实例元数据管理 EC2 实例。

如果实例未通过实例可到达性测试，请尝试以下操作来解决此问题。

- 检查您的 VPC 的 CIDR 块。如果 Windows 实例启动至 IP 地址范围为 `224.0.0.0 - 255.255.255.255` (D 类和 E 类 IP 地址范围) 的 VPC，则其无法正确启动。这些 IP 地址范围是预留的，不应分配给主机设备。我们建议您创建一台 CIDR 块来自私有（非公共可路由）IP 地址范围（如 RFC 1918 所指）的 VPC。
- 系统有可能配置了静态 IP 地址。尝试创建网络接口并将其连接到实例。
- 在无法连接到的 Windows 实例上启用 DHCP

1. 停止受影响的实例并分离其根卷。
2. 在受影响的实例所在的可用区中启动临时实例。

> **Warning**
>
> 如果您的临时实例与原始实例基于相同的 AMI，则您必须完成额外的步骤，否则在您恢复原始实例的根卷之后，由于磁盘签名冲突，您将无法启动原始实例。或者，可以为临时实例选择不同的 AMI。例如，如果原始实例使用适用于 Windows Server 2016 的 Amazon Windows AMI，则使用适用于 Windows Server 2019 的 Amazon Windows AMI 来启动临时实例。

3. 将根卷从受影响的实例附加到此临时实例。连接到临时实例，打开 Disk Management (磁盘管理) 实用工具，将驱动器联机。
4. 从临时实例，打开 Regedit 并选择 HKEY_LOCAL_MACHINE。从 File 菜单中，选择 Load Hive。选择驱动器，打开文件 `Windows\System32\config\SYSTEM`，在出现提示时指定键名 (您可以使用任何名称)。
5. 选择刚加载的键并导航至 `ControlSet001\Services\Tcpip\Parameters\Interfaces`。每个网络接口均按 GUID 列出。选择正确的网络接口。如果禁用了 DHCP

且分配了静态 IP 地址，则 EnableDHCP 设置为 0。要启用 DHCP，请将 EnableDHCP 设置为 1，并且删除以下键 (如果存在)：NameServer、SubnetMask、IPAddress 和 DefaultGateway。再次选择该键，在 File 菜单中，选择 Unload Hive。

> **Note**
>
> 如果您有多个网络接口，您将需要确定适合的 DHCP 来启用接口。要确定适合的网络接口，请查看以下键值 NameServer、SubnetMask、IPAddress 和 DefaultGateway。这些值显示前一实例的静态配置。

6. （可选）如果已启用 DHCP，则可能是您没有通向该元数据服务的路由。更新 EC2Config 可以解决此问题。

   a. 下载并安装最新版本的 EC2Config 服务。有关安装此服务的更多信息，请参阅the section called “安装 EC2Config”。

   b. 将文件从 .zip 文件提取到附加的驱动器上的 Temp 目录。

   c. 打开 Regedit，选择 HKEY_LOCAL_MACHINE。从 File 菜单中，选择 Load Hive。选择驱动器，打开文件 Windows\System32\config\SOFTWARE，在出现提示时指定键名 (您可以使用任何名称)。

   d. 选择刚加载的键并导航至 Microsoft\Windows\CurrentVersion。选择 RunOnce 键。(如果此键不存在，请右键单击 CurrentVersion，指向 New，选择 Key，然后将该键命名为 RunOnce。) 右键单击，指向 New (新建)，选择 String Value (字符串值)。输入 Ec2Install 作为名称并输入 C:\Temp\Ec2Install.exe -q 作为数据。

   e. 再次选择该键，在 File 菜单中，选择 Unload Hive。

7. （可选）如果您的临时实例与原始实例基于相同的 AMI，则您必须完成以下步骤，否则在您恢复原始实例的根卷之后，由于磁盘签名冲突，您将无法启动原始实例。

> **Warning**
>
> 以下过程介绍了如何使用注册表编辑器编辑 Windows 注册表。如果您不熟悉 Windows 注册表或如何安全地使用注册表编辑器进行更改，请参阅配置注册表。

   a. 打开命令提示符，键入 regedit.exe，然后按 Enter。

   b. 在 Registry Editor (注册表编辑器) 中，从上下文菜单（右键单击）中选择 HKEY_LOCAL_MACHINE，然后选择 Find (查找)。

c. 键入 Windows Boot Manager，然后选择 Find Next (查找下一个)。

d. 选择名为 11000001 的密钥。此密钥是您在上一步中找到的密钥的同级。

e. 在右侧窗格中，选择 Element，然后从上下文菜单（右键单击）中选择 Modify (修改)。

f. 找到数据中偏移 0x38 处的四字节磁盘签名。反转字节以创建磁盘签名，然后将其记录下来。例如，由以下数据表示的磁盘签名是 E9EB3AA5：

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

g. 在命令提示符窗口运行以下命令以启动 Microsoft DiskPart。

```
diskpart
```

h. 运行以下 DiskPart 命令以选择卷。（您可以使用磁盘管理实用程序验证磁盘号是否为 1。）

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. 运行以下 DiskPart 命令以获取磁盘签名。

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. 如果上一步中显示的磁盘签名与之前写下的 BCD 中的磁盘签名不匹配，请使用以下 DiskPart 命令更改磁盘签名，使其匹配：

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. 使用 Disk Management (磁盘管理) 实用工具，将驱动器脱机。

> **Note**
>
> 如果临时实例与受影响的实例运行相同的操作系统，则驱动程序将自动离线，因此您无需手动使其离线。

9. 将该卷从临时实例分离。如果您不再使用临时实例，则可以将其终止。
10. 将受影响实例的根卷作为 `/dev/sda1` 附加，从而将其还原。
11. 启动受影响的实例。

如果您已连接到实例，请从该实例打开 Internet 浏览器，然后输入元数据服务器的以下 URL：

```
http://169.254.169.254/latest/meta-data/
```

如果您无法连接到元数据服务器，请尝试以下操作解决问题：

- 下载并安装最新版本的 EC2Config 服务。有关安装此服务的更多信息，请参阅the section called “安装 EC2Config”。
- 检查 Windows 实例是否正在运行 Red Hat PV 驱动程序。如果正在运行这种驱动程序，请更新至 Citrix PV 驱动程序。有关更多信息，请参阅 the section called “升级半虚拟化驱动程序”。
- 验证防火墙、IPSec 和代理设置没有阻止到元数据服务 (`169.254.169.254`) 或 Amazon KMS 服务器（地址在 `TargetKMSServer` 中的 `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml` 元素中指定）的传出流量。
- 使用以下命令验证您拥有到元数据服务 (`169.254.169.254`) 的路由。

  ```
  route print
  ```

- 检查可能影响您的实例的可用区的网络问题。请访问 http://status.aws.amazon.com/。

"无法激活 Windows"

Windows 实例使用 Windows Amazon KMS 激活。如果您的实例无法访问 `A problem occurred when Windows tried to activate. Error Code 0xC004F074` 服务器，您可能会收到此消息：Amazon KMS。必须每隔 180 天激活一次 Windows。在激活期限到期之前，EC2Config 会尝试连接 Amazon KMS 服务器以确保 Windows 仍处于激活状态。

如果您遇到 Windows 激活问题，请使用以下过程来解决此问题。

对于 EC2Config（Windows Server 2012 R2 AMI 及更早版本）

1. 下载并安装最新版本的 EC2Config 服务。有关安装此服务的更多信息，请参阅the section called “安装 EC2Config”。
2. 登录实例并打开以下文件：C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml。
3. 在 文件中找到 Ec2WindowsActivate 插件。config.xml将状态更改为 Enabled 并保存您的更改。
4. 在 Windows 服务管理单元中，重新启动 EC2Config 服务或者重启实例。

如果这没有解决激活问题，请按照下面这些额外步骤操作。

1. 设定 Amazon KMS 目标：C:\> slmgr.vbs /skms 169.254.169.250:1688
2. 激活 Windows：C:\> slmgr.vbs /ato

对于 EC2Launch（Windows Server 2016 AMI 及更高版本）

1. 从具有管理权限的 PowerShell 提示符中，导入 EC2Launch 模块：

```
PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1"
```

2. 调用 Add Routes 函数查看新路由列表：

```
PS C:\> Add-Routes
```

3. 调用 Set-ActivationSettings 函数：

```
PS C:\> Set-Activationsettings
```

4. 然后，运行以下脚本以激活 Windows：

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

对于 EC2Config 和 EC2Launch，如果您仍收到激活错误，请验证以下信息。

- 确认您有到 Amazon KMS 服务器的路由。打开 `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`，找到 `TargetKMSServer` 元素。运行以下命令，检查是否列出了这些 Amazon KMS 服务器的地址。

```
route print
```

- 确认已设置 Amazon KMS 客户端密钥。运行以下命令并检查输出。

```
C:\Windows\System32\slmgr.vbs /dlv
```

  如果输出包含 Error: product key not found，则说明未设置 Amazon KMS 客户端密钥。如果未设置 Amazon KMS 客户端密钥，请按 Microsoft 文章（Amazon KMS 客户端激活和产品密钥）中的说明查找客户端密钥，然后运行以下命令来设置 Amazon KMS 客户端密钥。

```
C:\Windows\System32\slmgr.vbs /ipk client_key
```

- 确认系统的时间和时区是正确的。如果您使用的是 UTC 之外的其他时区，请添加以下注册表项并将其设置为 1 以确保时间正确：`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal`。
- 如果启用了 Windows 防火墙，请使用以下命令临时将其禁用。

```
netsh advfirewall set allprofiles state off
```

"Windows 不是正版 (0x80070005)"

Windows 实例使用 Windows Amazon KMS 激活。如果实例无法完成激活过程，则会报告该 Windows 副本不是正版。

请尝试有关"无法激活 Windows" 的建议。

"没有终端服务器许可服务器可提供许可证"

默认情况下，Windows Server 已获得授权，允许两个用户通过远程桌面同时使用。如果需要允许两个以上用户通过远程桌面同时访问您的 Windows 实例，您可以购买远程桌面服务客户端访问许可证 (CAL)，并安装“Remote Desktop Session Host (远程桌面会话主机)”和“Remote Desktop Licensing Server (远程桌面许可服务器)”角色。

检查有无以下问题：

- 已超过并发 RDP 会话的最大数目。
- 已安装 Windows 远程桌面服务角色。
- 许可已过期。如果许可已过期，那么您无法作为用户连接到您的 Windows 实例。您可以尝试以下操作：
  - 使用 `/admin` 参数从命令行连接到实例，例如：

    ```
    mstsc /v:instance /admin
    ```

    有关更多信息，请参阅以下 Microsoft 文章：通过命令行访问远程桌面。
  - 停止该实例，分离其 Amazon EBS 卷，然后将这些卷附加到同一可用区中的另一个实例以恢复数据。

“某些设置由您的组织管理”

从最新的 Windows Server AMI 启动的实例可能会显示 Windows Update 对话框消息，说明“某些设置由您的组织管理”。显示此消息是由于 Windows Server 中的一些变化导致的，不会影响 Windows Update 的行为或您管理更新设置的能力。

消除警告

1. 打开 `gpedit.msc`，并依次导航到计算机配置、管理模板、Windows 组件、Windows 更新。编辑配置自动更新，并将其设置为已启用。
2. 在命令提示符下，使用 gpupdate /force 更新组策略。
3. 关闭并重新打开“Windows 更新设置”。此时将会看到有关您的组织正在管理的设置的上述消息，然后显示“我们将自动下载更新，但通过按流量计费的连接除外（在此情况下会收费）。在这种情况下，我们将自动下载保持 Windows 平稳运行所需的更新”。
4. 返回到 `gpedit.msc`，并将组策略重新设置为未配置。再次运行 gpupdate /force。
5. 关闭命令提示符并等待几分钟。
6. 重新打开“Windows 更新设置”。您将不会再看到“某些设置由您的组织管理”消息。

排查 Amazon EC2 Windows 实例问题

以下问题排查技巧有助于解决与 Amazon EC2 Windows 实例相关的问题。

问题

- 无法将 Amazon Systems Manager 会话管理器连接到 Windows Server 2025 实例
- EBS 卷在 Windows Server 2016 和 2019 上不初始化
- 将 EC2 Windows 实例启动至目录服务还原模式 (DSRM)
- 实例失去网络连接或计划的任务不按预期方式运行
- 无法获取控制台输出
- 网络上不可用的 Windows Server 2012 R2
- 磁盘签名冲突

无法将 Amazon Systems Manager 会话管理器连接到 Windows Server 2025 实例

将 Amazon Systems Manager 会话管理器连接到 Windows Server 2025 实例可能会遇到问题。要解决此问题，请登录实例，然后导航到`Settings > Apps > Optional Features`并添加 `WMIC`。重启 SSM 代理服务或重启实例，并且Sessions Manager应该可以连接了。

您也可以使用以下 PowerShell 命令执行相同的操作：

```
Start-Process -FilePath "$env:SystemRoot\system32\Dism.exe" -ArgumentList @('/
Online', '/Add-Capability', '/CapabilityName:WMIC~~~~') -Wait; Restart-Service -Name
 AmazonSSMAgent
```

EBS 卷在 Windows Server 2016 和 2019 上不初始化

从 Windows Server 2016 和 2019 的 Amazon 系统映像（AMI）创建的实例使用 EC2Launch v1 代理来执行各种启动任务，包括初始化 EBS 卷。默认情况下，EC2Launch v1 不初始化辅助卷。不过，您可以将 EC2Launch v1 配置为自动初始化这些磁盘，如下所示。

将盘符映射到卷

1. 连接到要配置的实例并在文本编辑器中打开 `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` 文件。
2. 指定卷设置如下：

```
{
"driveLetterMapping": [
```

```
 {
   "volumeName": "sample volume",
   "driveLetter": "H"
 }]
}
```

3. 保存更改并关闭文件。
4. 打开 Windows PowerShell 并使用以下命令来运行初始化磁盘的 EC2Launch v1 脚本：

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

要在每次实例启动时初始化磁盘，请添加 `-Schedule` 标记，如下所示：

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -
Schedule
```

EC2Launch v1 代理可以运行实例初始化脚本，例如 `initializeDisks.ps1` 与 `InitializeInstance.ps1` 脚本并行使用。如果 `InitializeInstance.ps1` 脚本重启实例，它可能会中断实例启动时运行的其他计划任务。为避免任何潜在的冲突，建议向 `initializeDisks.ps1` 脚本添加逻辑，以确保首先完成实例初始化。

> **Note**
>
> 如果 EC2Launch 脚本未初始化卷，请确保卷处于联机状态。如果卷处于脱机状态，请运行以下命令，使所有磁盘联机。
>
> ```
> PS C:\> Get-Disk | Where-Object IsOffline -Eq $True | Set-Disk -IsOffline
>  $False
> ```

将 EC2 Windows 实例启动至目录服务还原模式 (DSRM)

如果运行 Microsoft Active Directory 的实例遇到系统故障或其他关键问题，您可以通过启动至特殊版本的安全模式（称为目录服务还原模式 (DSRM)）对该实例进行故障排除。在 DSRM 中，您可以修复或恢复 Active Directory。

DSRM 的驱动程序支持

您启用 DSRM 和启动至实例的方式取决于实例正在运行的驱动程序。在 EC2 控制台中，您可以从系统日志查看实例的驱动程序版本详细信息。下表显示了 DSRM 支持的驱动程序。

驱动程序版本	DSRM 是否支持？	后续步骤
Citrix PV 驱动程序 5.9	否	从备份还原实例。您无法启用 DSRM。
Amazon PV 7.2.0	否	尽管该驱动程序不支持 DSRM，但您仍可以将根卷从实例分离并为其创建快照或 AMI，然后将其作为辅助卷附加到同一个可用区中的其他实例。然后，您可以按照本部分的说明启用 DSRM。
Amazon PV 7.2.2 和更高版本	是	分离根卷，将其附加到其他实例，然后启用 DSRM (如本节中所述)。
增强联网	是	分离根卷，将其附加到其他实例，然后启用 DSRM (如本节中所述)。

有关如何启用增强联网的信息，请参阅 the section called “弹性网络适配器（ENA）”。有关升级 Amazon PV 驱动程序的信息，请参阅在 Windows 实例上升级半虚拟化驱动程序。

将实例配置为启动至 DSRM

在操作系统运行前，EC2 Windows 实例没有网络连接。因此，您无法通过按键盘上的 F8 键来选择启动选项。您必须使用以下过程之一将 EC2 Windows Server 实例启动至 DSRM。

如果您怀疑 Active Directory 已损坏，但该实例仍在运行，则可以使用 System Configuration 对话框或命令提示符将该实例配置为启动至 DSRM。

使用 System Configuration 对话框将在线实例启动至 DSRM

1. 在 Run 对话框中，键入 `msconfig` 并按下 Enter。
2. 选择 Boot 选项卡。
3. 在 Boot options 下，选择 Safe boot。

4. 选择 Active Directory repair，然后选择 OK。系统将提示您重新启动服务器。

使用命令行将在线实例启动至 DSRM

从命令提示符窗口运行以下命令：

```
bcdedit /set safeboot dsrepair
```

如果某个实例处于离线状态并且无法访问，您必须先分离根卷并将其附加到其他实例才能启用 DSRM 模式。

将离线实例启动至 DSRM

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 找到并选择受影响的实例。依次选择实例状态、停止实例。
4. 选择启动实例并在受影响实例所在的同一可用区中创建一个临时实例。选择使用其他 Windows 版本的实例类型。例如，如果实例是 Windows Server 2016，请选择 Windows Server 2019 实例。

   > **Important**
   >
   > 如果您未在与受影响实例相同的可用区中创建该实例，则无法将受影响的实例的根卷附加到新实例。

5. 在导航窗格中，选择 Volumes。
6. 找到受影响的实例的根卷。分离该卷并将其附加到早先创建的临时实例。使用默认设备名称 (xvdf) 附加该卷。
7. 使用远程桌面连接临时实例，然后通过磁盘管理实用工具使该卷可供使用。
8. 打开命令提示符窗口并运行以下命令。将 D 替换为您刚刚附加的辅助卷的实际驱动器号：

   ```
   bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
   ```

9. 在磁盘管理实用工具中，选择您之前附加的驱动器，打开上下文（右键单击）菜单，然后选择脱机。
10. 在 EC2 控制台中，将受影响的卷从临时实例分离，然后将其重新附加到设备名称为 /dev/sda1 的原始实例。您必须指定此设备名称才能将卷指派为根卷。

11. 启动实例。
12. 在实例将健康检查传入 EC2 控制台后，使用远程桌面连接到实例，然后验证实例是否启动至 DSRM 模式。
13. (可选) 删除或停止您在本过程中创建的临时实例。

实例失去网络连接或计划的任务不按预期方式运行

如果您重新启动实例而该实例失去网络连接，则可能该实例的时间不正确。

默认情况下，Windows 实例使用协调世界时 (UTC)。如果将实例的时间设置为不同的时区，然后再重新启动实例，则时间会发生偏移，该实例暂时失去其 IP 地址。实例最终会重新获取网络连接，但这可能要用数小时的时间。实例重新获取网络连接所用的时间取决于 UTC 与其他时区之间的差异。

这种时间问题也可能导致计划的任务不如期运行。在这种情况下，计划的任务会因实例的时间不正确而不如期运行。

要持久使用 UTC 以外的时区，您必须设置 RealTimeIsUniversal 注册表项。若不设置此项，则实例会在重新启动后使用 UTC。

解决造成网络连接丢失的时间问题

1. 确保所运行的是建议的半虚拟化驱动程序。有关更多信息，请参阅 the section called “升级半虚拟化驱动程序”。
2. 验证以下注册表项存在并且已设置为 1：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal

无法获取控制台输出

对于 Windows 实例，实例控制台显示在 Windows 引导过程中执行的任务的输出。如果 Windows 成功启动，则最后记录的消息是 `Windows is Ready to use`。您也可以在控制台中显示事件日志消息，但根据您的 Windows 版本，此功能可能默认未启用。有关更多信息，请参阅 the section called “Windows 启动代理”。

要使用 Amazon EC2 控制台获取您的实例的控制台输出，请选择实例，然后依次选择 Actions (操作)、Monitor and troubleshoot (显示屏和问题排查)、Get System Log (获取系统日志)。要使用命令行获取控制台输出，请使用以下命令之一：get-console-output (Amazon CLI) 或 Get-EC2ConsoleOutput (Amazon Tools for Windows PowerShell)。

对于运行 Windows Server 2012 R2 和更早版本的实例，如果控制台输出为空，则可能表示 EC2Config 服务出现问题（如未正确配置的配置文件），或者 Windows 无法正确引导。要修复该问题，请下载并安装最新版本的 EC2Config。有关更多信息，请参阅 the section called “安装 EC2Config”。

网络上不可用的 Windows Server 2012 R2

有关对网络上不可用的 Windows Server 2012 R2 实例进行故障排除的信息，请参阅 Windows Server 2012 R2 在实例重启后丢失网络和存储连接。

磁盘签名冲突

您可以使用 EC2Rescue for Windows Server 检查和解决磁盘签名冲突。或者，您可以通过执行以下步骤手动解决磁盘签名问题。

> **Warning**
>
> 以下过程介绍了如何使用注册表编辑器编辑 Windows 注册表。如果您不熟悉 Windows 注册表或如何安全地使用注册表编辑器进行更改，请参阅配置注册表。

1. 打开命令提示符，键入 regedit.exe，然后按 Enter。
2. 在 Registry Editor (注册表编辑器) 中，从上下文菜单（右键单击）中选择 HKEY_LOCAL_MACHINE，然后选择 Find (查找)。
3. 键入 Windows Boot Manager，然后选择 Find Next (查找下一个)。
4. 选择名为 `11000001` 的密钥。此密钥是您在上一步中找到的密钥的同级。
5. 在右侧窗格中，选择 `Element`，然后从上下文菜单（右键单击）中选择 Modify (修改)。
6. 找到数据中偏移 0x38 处的四字节磁盘签名。这是启动配置数据库签名 (BCD)。反转字节以创建磁盘签名，然后将其记录下来。例如，由以下数据表示的磁盘签名是 `E9EB3AA5`：

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

7. 在命令提示符窗口运行以下命令以启动 Microsoft DiskPart。

```
diskpart
```

8. 运行 `select disk` DiskPart 命令，并为存在磁盘签名冲突的卷指定磁盘号。

> **Tip**
>
> 要检查存在磁盘签名冲突的卷的磁盘号，请使用磁盘管理实用程序。打开命令提示符，键入 `compmgmt.msc`，然后按 Enter 键。在左侧导航面板中，双击磁盘管理。在磁盘管理实用程序中，检查存在磁盘签名冲突的脱机卷的磁盘号。

```
DISKPART> select disk 1
Disk 1 is now the selected disk.
```

9. 运行以下 DiskPart 命令以获取磁盘签名。

```
DISKPART>  uniqueid disk
Disk ID: 0C764FA8
```

10. 如果上一步中显示的磁盘签名与之前写下的磁盘签名不匹配，请使用以下 DiskPart 命令更改磁盘签名，使其匹配：

```
DISKPART> uniqueid disk id=E9EB3AA5
```

通过网络对 Windows 实例进行内核调试

适用于弹性网络适配器（ENA）的 KDNET 可扩展性模块是一个硬件驱动程序支持层，它借助 Amazon Elastic Compute Cloud 实例上的 ENA，支持通过网络对 Windows 进行内核调试。您可以将可扩展性模块与 Windows 调试程序（WinDbg）结合使用，对运行 Windows 的 EC2 实例进行内核级调试。

内核调试可帮助您诊断并解决 EC2 Windows 实例上的低级操作系统问题，例如蓝屏错误（BSOD）、驱动程序故障和启动问题。

内容

- 前提条件
- 步骤 1：在调试主机上安装 Windows 调试工具
- 步骤 2：设置调试目标
- 步骤 3：在调试主机上启动调试会话
- 步骤 4：重新启动调试目标

- 调试后清理调试设置
- 限制
- 附加说明

前提条件

在开始之前，请确保您具有以下各项：

同一子网中的两个 EC2 Windows 实例：

- 一个调试主机实例：运行 Windows 调试程序（WinDbg）的实例。
- 一个调试目标实例：要调试的实例。

有关启动实例的更多信息，请参阅 入门教程。

主机和目标实例的安全组必须允许用于调试 KDNET 的端口上的入站和出站 UDP 流量（建议范围：50000 – 50039）。设置此规则的最简单方法是创建一个安全组，添加入站规则允许来自自身的 UDP 流量作为来源，然后将该安全组附加到两个实例。有关更多信息，请参阅 Amazon VPC 用户指南中的安全组规则。

调试目标实例必须运行以下 Windows 版本之一（或更高版本）：

- Windows Server 2025，内部版本号 26100.7462（2025 年 12 月补丁）
- Windows 11 24H2，内部版本号 26100.7309
- Windows 11 25H2，内部版本号 26200.7309

> **Note**
>
> 适用于 ENA 的 KDNET 可扩展性模块作为 Windows 的一部分分发，只能通过 Windows 每月累积更新进行更新。建议在调试目标上安装最新的 Windows 知识库，以确保您拥有最新的模块版本。
> 要验证调试目标上是否存在模块，请在提升权限的 PowerShell 会话中运行以下命令：
>
> ```
> Test-Path C:\Windows\system32\kd_02_1d0f.dll
> ```
>
> 如果命令返回 `True`，则模块可用。

步骤 1：在调试主机上安装 Windows 调试工具

在调试主机实例上，通过在 PowerShell 会话中运行以下命令来安装 Windows 调试工具：

```
winget install microsoft.windbg
```

有关详细的安装说明，请参阅 Microsoft 文档中的 Install the Windows Debugger。

安装完成后，通过在 PowerShell 会话中运行以下命令来验证调试程序是否正常工作：

```
windbgx
```

WinDbg 窗口应该会打开。如果打开，则说明安装成功，您可以关闭该窗口。

步骤 2：设置调试目标

> **Note**
>
> 当内核调试处于活动状态时，用于调试会话的 ENA 设备将仅用于调试程序流量。如果需要在调试期间保持调试目标实例的互联网访问，请在开始之前将第二个 ENA 附加到实例。

在调试目标上，打开提升权限的 PowerShell 会话，按照以下步骤配置内核调试。

运行以下命令，列出附加到实例的 ENA 适配器的总线、设备、功能编号：

```
Get-NetAdapter -Physical |
    Where-Object -Property PnPDeviceID -Match -Value '^PCI\\VEN_1D0F&DEV_EC2[01]&' |
    Get-NetAdapterHardwareInfo |
    Select-Object InterfaceDescription, BusNumber, DeviceNumber, FunctionNumber |
    Format-List
```

如果多个 ENA 适配器附加到实例

如果多个 ENA 适配器附加到实例，请使用以下命令将每个物理适配器映射到其私有 IP 地址。您可以在 Amazon Web Services 管理控制台中的 EC2 → 实例 → [实例 ID] → 联网 → 网络接口下交叉引用这些详细信息。这有助于您将特定的网络接口 ID、私有 IP 和安全组与操作系统级适配器关联起来，以便进行有针对性的调试。

```
Get-NetAdapter -Physical |
    Where-Object PnPDeviceID -Match '^PCI\\VEN_1D0F&DEV_EC2[01]&' |
    ForEach-Object {
        $adapter = $_
        $hwInfo = $adapter | Get-NetAdapterHardwareInfo
        $ipInfo = Get-NetIPAddress -InterfaceIndex $adapter.InterfaceIndex -
AddressFamily IPv4
        [PSCustomObject]@{
            InterfaceDescription = $adapter.InterfaceDescription
            IPAddress            = $ipInfo.IPAddress
            BusNumber            = $hwInfo.BusNumber
            DeviceNumber         = $hwInfo.DeviceNumber
            FunctionNumber       = $hwInfo.FunctionNumber
        }
    } | Format-List
```

记下输出中用于调试的 ENA 适配器的 BusNumber、DeviceNumber 和 FunctionNumber 值。

运行以下命令来配置内核调试。将占位符值替换为具体配置：

```
bcdedit /debug on
bcdedit /set loadoptions FORCEHVTONOTSHAREDEBUGDEVICE
bcdedit /dbgsettings net hostip:host-private-ip port:port-number key:encryption-key
 busparams:b.d.f
```

> **Note**
>
> 运行以下命令来检查现有的 loadoptions。如果返回值，请复制该字符串并为其附加 ;FORCEHVTONOTSHAREDEBUGDEVICE。
>
> ```
> (bcdedit /enum) -match "loadoptions"
> ```

其中：

- *host-private-ip*：调试主机实例的私有 IPv4 地址。如果实例是在仅使用 IPv6 的子网中启动的，请参阅 Microsoft 文档中的 Setting up KDNET with IPv6，以了解有关将 IPv6 与 KDNET 结合使用的更多信息。
- *port-number*：用于调试会话的端口。建议范围为 50000 – 50039（例如 50000）。

- *encryption-key*：用于加密调试连接的 256 位密钥，以四个 64 位值表示，用英文句点分隔。每个 64 位值最长可为 13 个字符，仅可使用小写字母 a – z 和数字 0 – 9。不允许使用特殊字符。加密密钥示例：1kdnet2keys3.4kdnet5keys6.7kdnet8keys9.10kdnet11ke。
- *b.d.f*：ENA 设备的总线、设备和功能编号，格式为 bus.device.function（例如 0.3.0），用于调试。

> **Tip**
>
> 要调试 Windows 启动流程，另请运行以下命令：
>
> ```
> bcdedit /bootdebug on
> ```

步骤 3：在调试主机上启动调试会话

要允许主机上的调试流量，可以为 WinDbg 应用程序或特定的 UDP 端口创建防火墙规则。

选项 1：允许 WinDbg 应用程序

运行以下命令以授权 WinDbg 可执行文件：

```
$WinDbgxPath = "$env:LocalAppData\Microsoft\WindowsApps\WinDbgX.exe"
New-NetFirewallRule -DisplayName "Allow Inbound KDNET Connection" -Action Allow -
Program $WinDbgxPath
```

选项 2：允许特定的 UDP 端口

或者，您可以允许入站 UDP 流量到达为内核调试配置的端口。将###替换为所选择的 KDNET 端口：

```
$DebugPort = port-number
New-NetFirewallRule -DisplayName "Allow Inbound KDNET Connection" -Direction Inbound -
LocalPort $DebugPort -Protocol UDP -Action Allow
```

> **Note**
>
> 防火墙配置可能受域组策略（GPO）限制，或者需要提升管理员权限。如果命令失败，请联系网络管理员。

使用与调试目标配置匹配的端口和密钥启动 WinDbg。您可以根据自己的使用案例指定 Microsoft WinDbg command line reference 中记录的其他内核调试选项。

```
windbgx -k net:port=port-number,key=encryption-key
```

WinDbg 打开并等待调试目标连接。

步骤 4：重新启动调试目标

在调试目标上，重新启动实例以启动调试连接：

```
shutdown -r -t 0
```

调试目标重新启动后，调试主机上的 WinDbg 会自动连接到目标。现在，您可以使用 WinDbg 来检查内核状态、设置断点并诊断问题。

调试后清理调试设置

在调试目标上

完成调试后，从调试目标中删除内核调试配置以恢复正常启动行为。在调试目标上，打开提升权限的 PowerShell 会话并运行以下命令：

```
bcdedit /debug off
bcdedit /dbgsettings LOCAL
bcdedit /deletevalue loadoptions
```

如果您除 FORCEHVTONOTSHAREDEBUGDEVICE 之外还有其他现有的 `loadoptions`，则应通过运行带有原始 `loadoptions` 的 `bcdedit /set loadoptions` 来恢复该设置。

如果您启用了启动调试，请另运行：

```
bcdedit /bootdebug off
```

重新启动实例以使更改生效。

在调试主机上

通过运行以下命令删除防火墙规则：

```
Remove-NetFirewallRule -DisplayName "Allow Inbound KDNET Connection"
```

限制

适用于 ENA 的 KDNET 可扩展性模块目前不支持：

- 第 8 代非裸机 x86_64 实例类型（例如 `m8a.xlarge`）
- 第 7 代 48xlarge 非裸机 x86_64 实例类型（例如 `m7a.48xlarge`）
- u7i 实例类型

该模块目前不支持启用了安全启动的实例。您可以通过在提升权限的 PowerShell 会话中运行 `Confirm-SecureBootUEFI` 来验证状态。如果输出为 `True`，则安全启动处于活动状态。请注意，所有带有“TPM”前缀的 Amazon 提供的映像默认情况下启用安全启动。

附加说明

如果在将调试程序连接到目标实例时遇到问题，请验证以下各项：

- 满足本指南中列出的所有先决条件，包括所需的 Windows Server 内部版本号和可扩展性模块的存在。
- 附加到两个实例的安全组配置正确，允许它们之间在配置的调试端口上进行流量通信。
- 主机实例上的 Windows 防火墙规则不会阻止两个实例之间在配置端口上的网络流量。

有关其他指导，请参阅 Microsoft 文档中的 Set up KDNET network kernel debugging manually。

重置 Amazon EC2 Windows 实例的 Windows 管理员密码

如果因为 Windows 管理员密码丢失或过期而无法连接到 Amazon EC2 Windows 实例，则可以重置密码。

> **Note**
>
> 有一个自动应用重置本地管理员密码所需的手动步骤的 Amazon Systems Manager 自动化文档。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的在 Amazon EC2 实例上重置密码和 SSH 密钥。

重置管理员密码的手动方法使用 EC2Launch v2、EC2Config 或 EC2Launch。

- 对于包含 EC2Launch v2 代理的所有受支持的 Windows AMI，请使用 EC2Launch v2。
- 对于 Windows Server 2016 之前的 Windows AMI，请使用 EC2Config 服务。
- 对于 Windows Server 2016 及更高版本 AMI，请使用 EC2Launch 服务。

这些过程还描述了在丢失用于创建实例的密钥对时如何连接实例。Amazon EC2 使用公有密钥加密某个数据（如一个密码），并使用私有密钥解密数据。公有和私有密钥被称为密钥对。对于 Windows 实例，您可以使用密钥对获得管理员密码，然后使用 RDP 登录实例。

> **Note**
>
> 如果您在实例上禁用了本地管理员账户，并且您的实例已配置为 Systems Manager，则还可以使用 EC2Rescue 和 Run Command 重新启用和重置本地管理员密码。有关更多信息，请参阅将适用于 Windows Server 的 EC2Rescue 与 Systems Manager Run Command 配合使用。

内容

- 使用 EC2Launch v2 重置 EC2 实例的 Windows 管理员密码
- 使用 EC2Launch 重置 EC2 实例的 Windows 管理员密码
- 使用 EC2Config 重置 EC2 实例的 Windows 管理员密码

使用 EC2Launch v2 重置 EC2 实例的 Windows 管理员密码

如果您丢失了 Windows 管理员密码，并且正在使用包含 EC2Launch v2 代理的受支持的 Windows AMI，则可以使用 EC2Launch v2 生成新密码。

如果您使用的是不包含 EC2Launch v2 代理的 Windows Server 2016 或更高版本的 AMI，请参阅 使用 EC2Launch 重置 EC2 实例的 Windows 管理员密码。

如果您使用的是不包含 EC2Launch v2 代理的低于 Windows Server 2016 版本的 Windows Server AMI，请参阅 使用 EC2Config 重置 EC2 实例的 Windows 管理员密码。

> **Note**
>
> 如果您在实例上禁用了本地管理员账户，并且您的实例已配置为 Systems Manager，则还可以使用 EC2Rescue 和 Run Command 重新启用和重置本地管理员密码。有关更多信息，请参阅将适用于 Windows Server 的 EC2Rescue 与 Systems Manager Run Command 配合使用。

> **Note**
>
> 有一个自动应用重置本地管理员密码所需的手动步骤的 Amazon Systems Manager 自动化文档。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的在 Amazon EC2 实例上重置密码和 SSH 密钥。

要使用 EC2Launch v2 重置 Windows 管理员密码，需要执行以下操作：

- 第 1 步：验证 EC2Launch v2 代理是否正在运行
- 步骤 2：从实例分离根卷
- 步骤 3：将卷附加到临时实例
- 步骤 4：删除 .run-once 文件
- 步骤 5：重启原始实例

第 1 步：验证 EC2Launch v2 代理是否正在运行

在尝试重置管理员密码之前，您需要首先验证 EC2Launch v2 代理已经安装并且正在运行。在本节后面的内容中，您将使用 EC2Launch v2 代理重置管理员密码。

验证 EC2Launch v2 代理是否正在运行

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择要重置密码的实例。在本过程中，该实例称为原始 实例。
3. 依次选择 Actions (操作)、Monitor and troubleshoot (监控和问题排查)、Get system log (获取系统日志)。
4. 找到 EC2 Launch 条目，例如 Launch: EC2Launch v2 service v2.0.124。如果您看到此条目，则表示 EC2Launch v2 服务正在运行。

   如果系统日志输出为空，或者 EC2Launch v2 代理未运行，则使用实例控制台屏幕截图服务对实例进行问题排查。有关更多信息，请参阅 捕获无法访问的实例的屏幕截图。

步骤 2：从实例分离根卷

如果存储密码的卷作为根卷附加到实例，则无法使用 EC2Launch v2 重置管理员密码。必须先从原始实例分离该卷，然后才能将其作为辅助卷附加到临时实例。

从实例分离根卷

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择需要重置密码的实例，然后依次选择实例状态、停止实例。在实例的状态更改为 Stopped (已停止) 后，继续下一步。
4. （可选）如果您拥有在启动此实例时指定的私有密钥，请继续执行下一步。否则，请使用以下步骤将实例替换为使用新密钥对启动的新实例。
   a. 使用 Amazon EC2 控制台创建新的密钥对。要将新密钥对的名称设置为与丢失的私有密钥相同的名称，则必须先删除现有密钥对。
   b. 选择要替换的实例。请注意实例的实例类型、VPC、子网、安全组和 IAM 角色。
   c. 选择实例后，再依次选择操作、映像和模板、创建映像。键入映像名称和描述，然后选择 Create Image (创建映像)。
   d. 在导航窗格中，选择 AMI。等待映像状态变为可用。然后，选择映像并选择从映像启动实例。
   e. 完成字段以启动实例，并确保选择与要替换的实例相同的实例类型、VPC、子网、安全组和 IAM 角色，然后选择启动。
   f. 出现提示后，选择您为新实例创建的密钥对，然后选择启动实例。
   g. （可选）如果原始实例有关联的弹性 IP 地址，则请将其传输到新实例。如果原始实例除了根卷之外还有 EBS 卷，请将它们传输到新实例。
5. 按如下操作从原始实例分离根卷：
   a. 选择原始实例，然后选择存储选项卡。记下根设备名称下根设备的名称。在块设备下找到带有此设备名称的卷，并记下卷 ID。
   b. 在导航窗格中，选择 Volumes。
   c. 在卷列表中，选择记为根设备的卷，然后依次选择操作、断开卷。在卷状态更改为可用后，继续进行下一步操作。
6. 如果您创建了新实例来替换原始实例，则现在可立即终止原始实例。该原始实例已不再需要。对于本过程的其余部分，对原始实例的所有引用都将应用于您创建的新实例。

步骤 3：将卷附加到临时实例

接下来，启动一个临时实例，将卷作为辅助卷附加到该临时实例。这是用来修改配置文件的实例。

启动临时实例并附加卷

1. 启动临时实例，如下所示：

   a. 在导航窗格中，选择 Instances (实例)，然后选择 Launch instances (启动实例) 和 AMI。

   > **Important**
   > 要避免磁盘签名冲突，必须为不同版本的 Windows 选择 AMI。例如，如果原始实例运行 Windows Server 2019，则使用 Windows Server 2016 的基础 AMI 启动临时实例。

   b. 保留默认实例类型，然后选择 Next: Configure Instance Details (下一步: 配置实例详细信息)。

   c. 在 Configure Instance Details (配置实例详细信息) 页面上，为 Subnet (子网) 选择与原始实例相同的可用区，然后选择 Review and Launch (审核和启动)。

   > **Important**
   > 临时实例必须与原始实例位于同一可用区。如果您的临时实例位于不同的可用区中，则无法将原始实例的根卷附加到该实例。

   d. 在 Review Instance Launch 页面上，选择 Launch。

   e. 出现提示后，创建新的密钥对，将其下载到您计算机中的安全位置，然后选择 Launch Instances (启动实例)。

2. 将卷作为辅助卷附加到临时实例，如下所示：

   a. 在导航窗格中，选择 Volumes (卷)，选择从原始实例中分离的卷，然后依次选择 Actions (操作)、Attach Volume (附加卷)。

   b. 在 Attach Volume (附加卷) 对话框中，对于 Instances (实例)，开始键入临时实例的名称或 ID，然后从列表中选择实例。

   c. 对于 Device (设备)，键入 **xvdf**（如果尚未打开），然后选择 Attach (附加)。

步骤 4：删除 .run-once 文件

现在，您必须从附加到实例的脱机卷中删除 `.run-once` 文件。这将指示 EC2Launch v2 运行以 once 频率执行所有任务，其中包括设置管理员密码。您附加的辅助卷中的文件路径将类似于 `D:\ProgramData\Amazon\EC2Launch\state\.run-once`。

删除 .run-once 文件

1. 打开磁盘管理实用程序，按照以下说明使驱动器联机：使 Amazon EBS 卷可供使用。
2. 在您联机的磁盘中找到 `.run-once` 文件。
3. 删除 .run-once 文件。

> **Important**
>
> 任何设置为运行一次的脚本都将由此操作触发。

步骤 5：重启原始实例

删除 `.run-once` 文件后，将卷作为根卷重新附加到原始实例，并使用其密钥对连接到该实例以检索管理员密码。

1. 按以下操作将卷重新附加到原始实例：
   a. 在导航窗格中，选择 Volumes (卷)，选择从临时实例中分离的卷，然后依次选择 Actions (操作)、Attach Volume (附加卷)。
   b. 在 Attach Volume (附加卷) 对话框中，对于 Instances (实例)，开始键入原始实例的名称或 ID，然后选择实例。
   c. 对于 Device (设备)，键入 **`/dev/sda1`**。
   d. 选择 附加。在卷状态更改为 `in-use` 后，继续进行下一步操作。
2. 在导航窗格中，选择 Instances (实例)。选择原始实例，然后依次选择 Instance state (实例状态)、Start instance (启动实例)。在实例状态更改为 `Running` 后，继续进行下一步操作。
3. 使用新密钥对的私有密钥会检索您的新 Windows 管理员密码，然后连接到实例。有关更多信息，请参阅 使用 RDP 连接到 Windows 实例。

> **Important**
>
> 实例在您停止并启动它后将获取一个新的公共 IP 地址。确保使用实例的当前公有 DNS 名称连接到实例。有关更多信息，请参阅 Amazon EC2 实例状态更改。

4. （可选）如果您不再使用临时实例，可以将其终止。选择临时实例，然后依次选择 Instance State (实例状态) 和 Terminate instance (终止实例)。

使用 EC2Launch 重置 EC2 实例的 Windows 管理员密码

如果丢失了 Windows 管理员密码并且使用的是 Windows Server 2016 或更高版本 AMI，可以使用 EC2Rescue 工具（使用 EC2Launch 服务）生成新密码。

如果您使用的是不包含 EC2Launch v2 代理的 Windows Server 2016 或更高版本的 AMI，则可以使用 EC2Launch v2 生成新密码。

如果您使用的是 Windows Server 2016 之前的 Windows Server AMI，请参阅使用 EC2Config 重置 EC2 实例的 Windows 管理员密码。

> **Warning**
>
> 停止某个实例时，实例存储卷上的数据将会丢失。要保留这些数据，请将其备份到持久性存储中。

> **Note**
>
> 如果您在实例上禁用了本地管理员账户，并且您的实例已配置为 Systems Manager，则还可以使用 EC2Rescue 和 Run Command 重新启用和重置本地管理员密码。有关更多信息，请参阅将适用于 Windows Server 的 EC2Rescue 与 Systems Manager Run Command 配合使用。

> **Note**
>
> 有一个自动应用重置本地管理员密码所需的手动步骤的 Amazon Systems Manager 自动化文档。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的在 Amazon EC2 实例上重置密码和 SSH 密钥。

要使用 EC2Launch 重置 Windows 管理员密码，需要执行以下操作：

- 步骤 1：从实例分离根卷
- 步骤 2：将卷附加到临时实例
- 步骤 3：重置管理员密码
- 步骤 4：重启原始实例

步骤 1：从实例分离根卷

如果存储密码的卷作为根卷附加到实例，则无法使用 EC2Launch 重置管理员密码。必须先从原始实例分离该卷，然后才能将其作为辅助卷附加到临时实例。

从实例分离根卷

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择需要重置密码的实例，然后依次选择实例状态、停止实例。在实例的状态更改为 Stopped (已停止) 后，继续下一步。
4. （可选）如果您拥有在启动此实例时指定的私有密钥，请继续执行下一步。否则，请使用以下步骤将实例替换为使用新密钥对启动的新实例。
   a. 使用 Amazon EC2 控制台创建新的密钥对。要将新密钥对的名称设置为与丢失的私有密钥相同的名称，则必须先删除现有密钥对。
   b. 选择要替换的实例。请注意实例的实例类型、VPC、子网、安全组和 IAM 角色。
   c. 选择实例后，再依次选择操作、映像和模板、创建映像。键入映像名称和描述，然后选择 Create Image (创建映像)。
   d. 在导航窗格中，选择 AMI。等待映像状态变为可用。然后，选择映像并选择从映像启动实例。
   e. 完成字段以启动实例，并确保选择与要替换的实例相同的实例类型、VPC、子网、安全组和 IAM 角色，然后选择启动。
   f. 出现提示后，选择您为新实例创建的密钥对，然后选择启动实例。
   g. （可选）如果原始实例有关联的弹性 IP 地址，则请将其传输到新实例。如果原始实例除了根卷之外还有 EBS 卷，请将它们传输到新实例。
5. 按如下操作从原始实例分离根卷：

a. 选择原始实例，然后选择存储选项卡。记下根设备名称下根设备的名称。在块设备下找到带有此设备名称的卷，并记下卷 ID。

b. 在导航窗格中，选择 Volumes。

c. 在卷列表中，选择记为根设备的卷，然后依次选择操作、断开卷。在卷状态更改为可用后，继续进行下一步操作。

6. 如果您创建了新实例来替换原始实例，则现在可立即终止原始实例。该原始实例已不再需要。对于本过程的其余部分，对原始实例的所有引用都将应用于您创建的新实例。

步骤 2：将卷附加到临时实例

接下来，启动一个临时实例，将卷作为辅助卷附加到该临时实例。这是用来运行 EC2Launch 的实例。

启动临时实例并附加卷

1. 启动临时实例，如下所示：

a. 在导航窗格中，选择 Instances (实例)，然后选择 Launch instances (启动实例) 和 AMI。

> **Important**
>
> 要避免磁盘签名冲突，必须为不同版本的 Windows 选择 AMI。例如，如果原始实例运行 Windows Server 2019，则使用 Windows Server 2016 的基础 AMI 启动临时实例。

b. 保留默认实例类型，然后选择 Next: Configure Instance Details (下一步: 配置实例详细信息)。

c. 在 Configure Instance Details (配置实例详细信息) 页面上，为 Subnet (子网) 选择与原始实例相同的可用区，然后选择 Review and Launch (审核和启动)。

> **Important**
>
> 临时实例必须与原始实例位于同一可用区。如果您的临时实例位于不同的可用区中，则无法将原始实例的根卷附加到该实例。

d. 在 Review Instance Launch 页面上，选择 Launch。

e. 出现提示后，创建新的密钥对，将其下载到您计算机中的安全位置，然后选择 Launch Instances (启动实例)。

2. 将卷作为辅助卷附加到临时实例，如下所示：

   a. 在导航窗格中，选择 Volumes (卷)，选择从原始实例中分离的卷，然后依次选择 Actions (操作)、Attach Volume (附加卷)。

   b. 在 Attach Volume (附加卷) 对话框中，对于 Instances (实例)，开始键入临时实例的名称或 ID，然后从列表中选择实例。

   c. 对于 Device (设备)，键入 **xvdf**（如果尚未打开），然后选择 Attach (附加)。

步骤 3：重置管理员密码

接下来，连接到临时实例并使用 EC2Launch 重置管理员密码。

重置管理员密码

1. 连接到该临时实例并在该实例上使用 EC2Rescue for Windows Server 工具重置管理员密码，如下所示：

   a. 下载 EC2Rescue for Windows Server zip 文件，提取内容，然后运行 EC2Rescue.exe。

   b. 在 License Agreement (许可协议) 屏幕上，阅读许可协议，如果您接受条款，请选择 I Agree (我同意)。

   c. 在 Welcome to EC2Rescue for Windows Server (欢迎使用 EC2Rescue for Windows Server) 屏幕上，选择 Next (下一步)。

   d. 在 Select mode (选择模式) 屏幕上，选择 Offline instance (离线实例)。

   e. 在 Select a disk (选择磁盘) 屏幕上，选择 xvdf 设备，然后选择 Next (下一步)。

   f. 确认所选磁盘并选择 Yes。

   g. 卷加载完成后，选择 OK (确定)。

   h. 在 Select Offline Instance Option (选择离线实例选项) 屏幕上，选择 Diagnose and Rescue (诊断和抢救)。

   i. 在 Summary (摘要) 屏幕上，查看信息，然后选择 Next (下一步)。

   j. 在 Detected possible issues (检测到可能存在的问题) 屏幕上，选择 Reset Administrator Password (重置管理员密码)，然后选择 Next (下一步)。

   k. 在 Confirm (确认) 屏幕上，选择 Rescue (抢救)，然后选择 OK (确定)。

   l. 在 Done (完成) 屏幕上，选择 Finish (完成)。

   m. 关闭 EC2Rescue for Windows Server 工具，断开与临时实例的连接，然后返回到 Amazon EC2 控制台。

2. 通过以下操作从临时实例分离辅助 (xvdf) 卷：

   a. 在导航窗格中，选择实例，然后选择临时实例。

   b. 在临时实例的 Storage (存储) 选项卡上，记下列为 xvdf 的 EBS 卷的 ID。

   c. 在导航窗格中，选择 Volumes。

   d. 在卷列表中，选择在上一步中记下的卷，然后依次选择操作、断开卷。在卷状态更改为可用后，继续进行下一步操作。

步骤 4：重启原始实例

使用 EC2Launch 重置管理员密码后，将卷作为根卷重新附加到原始实例，并使用其密钥对连接到实例以检索管理员密码。

重启原始实例

1. 按以下操作将卷重新附加到原始实例：

   a. 在导航窗格中，选择 Volumes (卷)，选择从临时实例中分离的卷，然后依次选择 Actions (操作)、Attach Volume (附加卷)。

   b. 在 Attach Volume (附加卷) 对话框中，对于 Instances (实例)，开始键入原始实例的名称或 ID，然后选择实例。

   c. 对于 Device (设备)，键入 **/dev/sda1**。

   d. 选择 附加。在卷状态更改为 in-use 后，继续进行下一步操作。

2. 在导航窗格中，选择 Instances (实例)。选择原始实例，然后依次选择 Instance state (实例状态)、Start instance (启动实例)。在实例状态更改为 Running 后，继续进行下一步操作。

3. 使用新密钥对的私有密钥会检索您的新 Windows 管理员密码，然后连接到实例。有关更多信息，请参阅 使用 RDP 连接到 Windows 实例。

4. （可选）如果您不再使用临时实例，可以将其终止。选择临时实例，然后依次选择 Instance State (实例状态) 和 Terminate instance (终止实例)。

使用 EC2Config 重置 EC2 实例的 Windows 管理员密码

如果您丢失了 Windows 管理员密码并且使用的是 Windows Server 2016 之前的 Windows AMI，则可以使用 EC2Config 代理生成新密码。

如果使用的是 Windows Server 2016 或更高版本 AMI，请参阅 使用 EC2Launch 重置 EC2 实例的 Windows 管理员密码，或者可以使用 EC2Rescue 工具（使用 EC2Launch 服务）生成新密码。

> **Note**
>
> 如果您在实例上禁用了本地管理员账户，并且您的实例已配置为 Systems Manager，则还可以使用 EC2Rescue 和 Run Command 重新启用和重置本地管理员密码。有关更多信息，请参阅将适用于 Windows Server 的 EC2Rescue 与 Systems Manager Run Command 配合使用。

> **Note**
>
> 有一个自动应用重置本地管理员密码所需的手动步骤的 Amazon Systems Manager 自动化文档。有关更多信息，请参阅 Amazon Systems Manager 用户指南中的在 Amazon EC2 实例上重置密码和 SSH 密钥。

要使用 EC2Config 重置 Windows 管理员密码，需要执行以下操作：

- 步骤 1：确认 EC2Config 服务正在运行
- 步骤 2：从实例分离根卷
- 步骤 3：将卷附加到临时实例
- 步骤 4：修改配置文件
- 步骤 5：重启原始实例

步骤 1：确认 EC2Config 服务正在运行

在尝试重置管理员密码之前，请确认已安装并运行 EC2Config 服务。在本节后面的内容中，您将使用 EC2Config 服务重置管理员密码。

确认 EC2Config 服务正在运行

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)，然后选择要重置密码的实例。在本过程中，该实例称为原始 实例。
3. 依次选择 Actions (操作)、Monitor and troubleshoot (监控和问题排查)、Get system log (获取系统日志)。

4. 找到 EC2 代理条目，例如，EC2 Agent: Ec2Config service v3.18.1118。如果您看到此条目，则 EC2Config 服务正在运行。

   如果系统日志输出为空，或者 EC2Config 服务未运行，则使用实例控制台屏幕快照截图服务对实例进行故障排除。有关更多信息，请参阅 捕获无法访问的实例的屏幕截图。

步骤 2：从实例分离根卷

如果存储密码的卷作为根卷附加到实例，则无法使用 EC2Config 重置管理员密码。必须先从原始实例分离该卷，然后才能将其作为辅助卷附加到临时实例。

从实例分离根卷

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择需要重置密码的实例，然后依次选择实例状态、停止实例。在实例的状态更改为 Stopped (已停止) 后，继续下一步。
4. （可选）如果您拥有在启动此实例时指定的私有密钥，请继续执行下一步。否则，请使用以下步骤将实例替换为使用新密钥对启动的新实例。

   a. 使用 Amazon EC2 控制台创建新的密钥对。要将新密钥对的名称设置为与丢失的私有密钥相同的名称，则必须先删除现有密钥对。

   b. 选择要替换的实例。请注意实例的实例类型、VPC、子网、安全组和 IAM 角色。

   c. 选择实例后，再依次选择操作、映像和模板、创建映像。键入映像名称和描述，然后选择 Create Image (创建映像)。

   d. 在导航窗格中，选择 AMI。等待映像状态变为可用。然后，选择映像并选择从映像启动实例。

   e. 完成字段以启动实例，并确保选择与要替换的实例相同的实例类型、VPC、子网、安全组和 IAM 角色，然后选择启动。

   f. 出现提示后，选择您为新实例创建的密钥对，然后选择启动实例。

   g. （可选）如果原始实例有关联的弹性 IP 地址，则请将其传输到新实例。如果原始实例除了根卷之外还有 EBS 卷，请将它们传输到新实例。

5. 按如下操作从原始实例分离根卷：

   a. 选择原始实例，然后选择存储选项卡。记下根设备名称下根设备的名称。在块设备下找到带有此设备名称的卷，并记下卷 ID。

b. 在导航窗格中，选择 Volumes。

c. 在卷列表中，选择记为根设备的卷，然后依次选择操作、断开卷。在卷状态更改为可用后，继续进行下一步操作。

6. 如果您创建了新实例来替换原始实例，则现在可立即终止原始实例。该原始实例已不再需要。对于本过程的其余部分，对原始实例的所有引用都将应用于您创建的新实例。

步骤 3：将卷附加到临时实例

接下来，启动一个临时实例，将卷作为辅助卷附加到该临时实例。这是用来修改配置文件的实例。

启动临时实例并附加卷

1. 启动临时实例，如下所示：

   a. 在导航窗格中，选择 Instances (实例)，然后选择 Launch instances (启动实例) 和 AMI。

   > **Important**
   >
   > 要避免磁盘签名冲突，必须为不同版本的 Windows 选择 AMI。例如，如果原始实例运行 Windows Server 2019，则使用 Windows Server 2016 的基础 AMI 启动临时实例。

   b. 保留默认实例类型，然后选择 Next: Configure Instance Details (下一步: 配置实例详细信息)。

   c. 在 Configure Instance Details (配置实例详细信息) 页面上，为 Subnet (子网) 选择与原始实例相同的可用区，然后选择 Review and Launch (审核和启动)。

   > **Important**
   >
   > 临时实例必须与原始实例位于同一可用区。如果您的临时实例位于不同的可用区中，则无法将原始实例的根卷附加到该实例。

   d. 在 Review Instance Launch 页面上，选择 Launch。

   e. 出现提示后，创建新的密钥对，将其下载到您计算机中的安全位置，然后选择 Launch Instances (启动实例)。

2. 将卷作为辅助卷附加到临时实例，如下所示：

a. 在导航窗格中，选择 Volumes (卷)，选择从原始实例中分离的卷，然后依次选择 Actions (操作)、Attach Volume (附加卷)。

b. 在 Attach Volume (附加卷) 对话框中，对于 Instances (实例)，开始键入临时实例的名称或 ID，然后从列表中选择实例。

c. 对于 Device (设备)，键入 **xvdf**（如果尚未打开），然后选择 Attach (附加)。

步骤 4：修改配置文件

将卷作为辅助卷附加到临时实例后，修改配置文件中的 `Ec2SetPassword` 插件。

修改配置文件

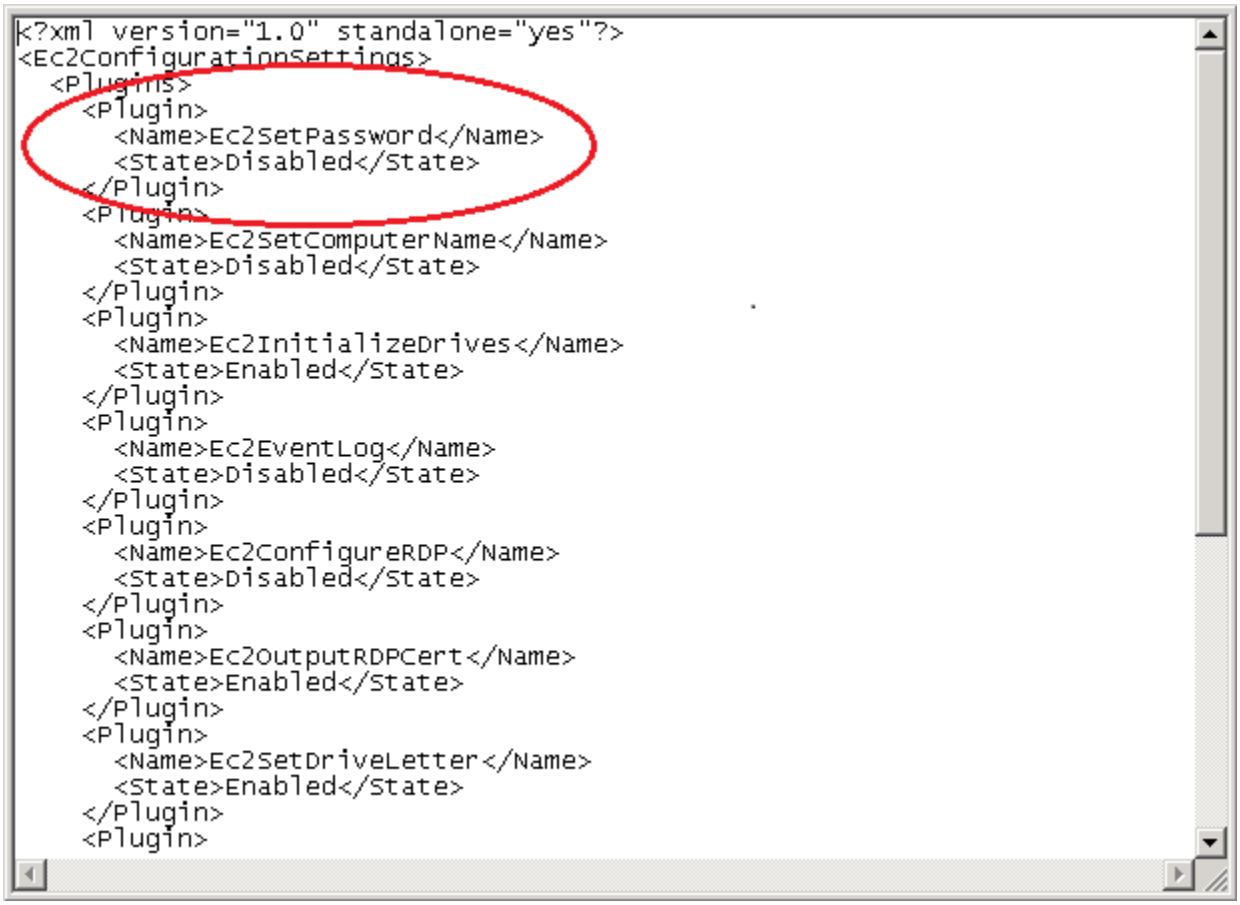
1. 在临时实例中，修改辅助卷上的配置文件，如下所示：

   a. 启动并连接到临时实例。

   b. 按照以下说明使驱动器联机：使 Amazon EBS 卷可用。

   c. 导航到辅助卷，使用文本编辑器 (如 Notepad) 打开 `\Program Files\Amazon \Ec2ConfigService\Settings\config.xml`。

   d. 在文件顶部，查找名为 `Ec2SetPassword` 的插件，如屏幕截图所示。将状态从 `Disabled` 更改为 `Enabled`，然后保存文件。



2. 修改配置文件后，按如下方式从临时实例分离辅助卷：

   a. 使用 Disk Management (磁盘管理) 实用工具，将该卷脱机。

   b. 断开与临时实例的连接，并返回 Amazon EC2 控制台。

   c. 在导航窗格中，选择卷，选择该卷，然后依次选择操作、断开卷。当卷的状态变为 available 之后，继续下一步操作。

步骤 5：重启原始实例

修改配置文件后，将卷作为根卷重新附加到原始实例，并使用其密钥对连接到该实例以检索管理员密码。

1. 按以下操作将卷重新附加到原始实例：

   a. 在导航窗格中，选择 Volumes (卷)，选择从临时实例中分离的卷，然后依次选择 Actions (操作)、Attach Volume (附加卷)。

b. 在 Attach Volume (附加卷) 对话框中，对于 Instances (实例)，开始键入原始实例的名称或 ID，然后选择实例。

c. 对于 Device (设备)，键入 **/dev/sda1**。

d. 选择 附加。在卷状态更改为 `in-use` 后，继续进行下一步操作。

2. 在导航窗格中，选择 Instances (实例)。选择原始实例，然后依次选择 Instance state (实例状态)、Start instance (启动实例)。在实例状态更改为 `Running` 后，继续进行下一步操作。

3. 使用新密钥对的私有密钥会检索您的新 Windows 管理员密码，然后连接到实例。有关更多信息，请参阅 使用 RDP 连接到 Windows 实例。

> **Important**
>
> 实例在您停止并启动它后将获取一个新的公共 IP 地址。确保使用实例的当前公有 DNS 名称连接到实例。有关更多信息，请参阅 Amazon EC2 实例状态更改。

4. （可选）如果您不再使用临时实例，可以将其终止。选择临时实例，然后依次选择 Instance State (实例状态) 和 Terminate instance (终止实例)。

排查 Amazon EC2 Windows 实例的 Sysprep 问题

如果在映像准备过程中遇到问题或收到错误消息，请查看以下日志。日志位置取决于您是使用 Sysprep 运行 EC2Config、EC2Launch v1 还是 EC2Launch v2。

- `%WINDIR%\Panther\Unattendgc`（EC2Config、EC2Launch v1 和 EC2Launch v2）
- `%WINDIR%\System32\Sysprep\Panther`（EC2Config、EC2Launch v1 和 EC2Launch v2）
- `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt`（仅限 EC2Config）
- `C:\ProgramData\Amazon\Ec2Config\Logs`（仅限 EC2Config）
- `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\EC2Launch.log`（仅限 EC2Launch v1）
- `%ProgramData%\Amazon\EC2Launch\log\agent.log`（仅限 EC2Launch v2）

如果在使用 Sysprep 进行映像准备的过程中收到错误消息，则 OS 可能无法访问。要查看日志文件，您必须停止实例，将其根卷作为辅助卷附加到另一个运行状况良好的实例，然后在辅助卷上查看前述日

志。如需详细了解如何通过名称辨别日志文件的用途，请参阅 Microsoft 文档中的 Windows 设置相关日志文件。

如果在 Unattendgc 日志文件中找到错误，请使用 Microsoft 错误查找工具获取有关错误的更多详细信息。以下在 Unattendgc 日志文件中报告的问题通常是实例上一个或多个损坏的用户配置文件造成的：

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003) [gle=0x00000003]
Error [Shell Unattend] CopyProfile failed (0x80070003) [gle=0x00000003]
```

有两个选项可用于解决此问题：

选项 1

在实例上使用 Regedit 搜索以下项。验证已删除用户是否没有配置文件注册表项。

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion
\ProfileList\
```

选项 2

1. 按如下方式编辑相关文件：
   - Windows Server 2012 R2 及更早版本 — 编辑 EC2Config 应答文件 (`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`)。
   - Windows Server 2016 和 2019 – 编辑 unattend.xml 应答文件 (`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml`)。
   - Windows Server 2022 – 编辑 unattend.xml 应答文件 (`C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml`)。
2. 将 `<CopyProfile>true</CopyProfile>` 更改为 `<CopyProfile>false</CopyProfile>`。
3. 再次运行 Sysprep。请注意，此配置更改会在 Sysprep 完成之后删除内置管理员用户配置文件。

使用 EC2Rescue 对受损的 Amazon EC2 Linux 实例进行问题排查

EC2Rescue for Linux 是一种易于使用的开源工具，可在 Amazon EC2 Linux 实例上运行此工具以通过其包含 100 多个模块的库来诊断、排查和修复常见问题。模块是包含 BASH 或 Python 脚本以及必要元数据的 YAML 文件。

EC2Rescue for Linux 实例的一些通用使用案例包括：

- 收集系统日志和软件包管理器日志
- 收集资源利用率数据
- 诊断和修复已知有问题的内核参数和常见 OpenSSH 问题

> **Note**
>
> `AWSSupport-TroubleshootSSH` Amazon Systems Manager Automation 运行手册安装了 EC2Rescue for Linux，然后使用该工具检查或尝试修复阻止 SSH 连接到 Linux 实例的常见问题。有关更多信息，请参阅 AWSSupport-TroubleshootSSH。

如果您使用的是 Windows 实例，则请参阅 the section called “EC2Rescue for Windows 实例”。

主题

- 在 Amazon EC2 Linux 实例上安装 EC2Rescue
- 在 Amazon EC2 Linux 实例上运行 EC2Rescue 命令
- 为 Amazon EC2 Linux 实例开发 EC2Rescue 模块

在 Amazon EC2 Linux 实例上安装 EC2Rescue

EC2Rescue for Linux 工具可以安装在满足以下先决条件的 Amazon EC2 Linux 实例上。

先决条件

- 支持的操作系统：
  - Amazon Linux 2
  - Amazon Linux 2016.09+
  - SUSE Linux Enterprise Server 12+
  - RHEL 7+
  - Ubuntu 16.04+
- 软件要求：
  - Python 2.7.9+ 或 3.2+

安装 EC2Rescue

`AWSSupport-TroubleshootSSH` 运行手册安装了 EC2Rescue for Linux，使用该工具检查或尝试修复阻止通过 SSH 远程连接到 Linux 计算机的常见问题。有关详细信息以及要运行此自动化，请参阅 Amazon Web Services 支持-TroubleshootSSH。

如果您的系统具有所需的 Python 版本，可以安装标准构建过程。否则，您可以安装捆绑构建过程，其中包括 Python 的最小副本。

安装标准构建过程

1. 从正常工作的 Linux 实例下载 EC2Rescue for Linux 工具：
2. （可选）验证 EC2Rescue for Linux 安装文件的签名。有关更多信息，请参阅 （可选）验证 EC2Rescue for Linux 的签名。
3. 下载 sha256 哈希文件：
4. 验证 tarball 的完整性：

```
sha256sum -c ec2rl.tgz.sha256
```

5. 解压缩 tarball：

```
tar -xzvf ec2rl.tgz
```

6. 通过列出帮助文件来验证安装：

```
cd ec2rl-<version_number>
./ec2rl help
```

安装捆绑构建过程

有关指向下载和限制列表的链接，请参阅 github 上的 EC2Rescue for Linux。

（可选）验证 EC2Rescue for Linux 的签名

下面是验证基于 Linux 的操作系统的 EC2Rescue for Linux 软件包是否有效的推荐过程。

当您从 Internet 下载应用程序时，我们建议您验证软件发行商的身份，并检查应用程序在发行后是否已遭更改或损坏。这会保护您免于安装含有病毒或其他恶意代码的应用程序版本。

如果您在执行本主题中的步骤后确定适用于 EC2Rescue for Linux 的软件已遭更改或损坏，请不要运行安装文件。否则，可联系 Amazon Web Services。

适用于基于 Linux 的操作系统的 EC2Rescue for Linux 文件是使用 GnuPG（安全数字签名的 Pretty Good Privacy 的开源式执行 (OpenPGP) 标准）进行签名的。GnuPG（也称为 GPG）通过数字签名进行身份验证和完整性检查。Amazon 发布了公有密钥和签名，可供您用于验证下载的 EC2Rescue for Linux 程序包。有关 PGP 和 GnuPG (GPG) 的更多信息，请访问 https://www.gnupg.org/。

第一步是与软件发行商建立信任。下载软件发行商的公有密钥，检查公有密钥的所有人是否真为其人，然后将该公有密钥添加到您的密钥环。密钥环是已知公有密钥的集合。验证公有密钥的真实性后，您可以使用它来验证应用程序的签名。

任务

- 验证并导入公有密钥
- 验证软件包的签名

验证并导入公有密钥

本流程的下一步是验证 EC2Rescue for Linux 公有密钥，并在 GPG 密钥环中将其添加为可信任密钥。

验证并导入 EC2Rescue for Linux 公有密钥

1. 在命令提示符处，使用以下命令获取我们的公共 GPG 生成密钥的副本：
2. 在保存 `ec2rl.key` 的目录中的命令提示符处，使用以下命令将 EC2Rescue for Linux 公有密钥导入密钥环：

   ```
   gpg2 --import ec2rl.key
   ```

   该命令返回的结果类似于下方内容：

   ```
   gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
   gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
    imported
   gpg: Total number processed: 1
   gpg:               imported: 1  (RSA: 1)
   ```

> **Tip**
>
> 如果您看到指示找不到该命令的错误，请使用 `apt-get install gnupg2`（基于 Debian 的 Linux）或 `yum install gnupg2`（基于 Red Hat 的 Linux）安装 GnuPG 实用程序。

验证软件包的签名

在安装 GPG 工具、验证并导入 EC2Rescue for Linux 公有密钥以及确认 EC2Rescue for Linux 公有密钥可信后，便可以验证 EC2Rescue for Linux 安装脚本的签名。

验证 EC2Rescue for Linux 安装脚本签名

1. 在命令提示符处，运行以下命令以下载安装脚本的签名文件：
2. 通过在保存 `ec2rl.tgz.sig` 和 EC2Rescue for Linux 安装文件的目录中的命令提示符处运行以下命令来验证签名。这两个文件都必须存在。

   ```
   gpg2 --verify ./ec2rl.tgz.sig
   ```

   输出应与以下内容类似：

   ```
   gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
   gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
   gpg: WARNING: This key is not certified with a trusted signature!
   gpg:          There is no indication that the signature belongs to the owner.
   Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C  C36A F780 4843 2FAE 2A1C
        Subkey fingerprint: 966B 0D27 85E9 AEEC 1146  7A9D 8851 1153 6991 ED45
   ```

   如果输出包含短语 `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"`，则意味着已成功验证签名，您可以继续运行 EC2Rescue for Linux 安装脚本。

   如果输出包含短语 `BAD signature`，则检查是否正确执行了此过程。如果您持续获得此响应，请联系 Amazon Web Services，而不要运行之前下载的安装文件。

下面是有关您可能看到的警告的详细信息：

- WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.这表示您坚信自己拥有 EC2Rescue for Linux 的可信公有密钥的个人信任级别。理想情况下，您将前往 Amazon Web Services 办公室并亲自接收此密钥。但更常见的情况是，从网站下载此密钥。在这种情况下，该网站是 Amazon Web Services 网站。
- gpg2: no ultimately trusted keys found. 这意味着您 (或您信任的其他人) 对特定密钥不是“绝对信任”。

有关更多信息，请参阅 https://www.gnupg.org/。

在 Amazon EC2 Linux 实例上运行 EC2Rescue 命令

EC2Rescue 是一个命令行工具。在 Linux 实例上安装 EC2Rescue 后，您可以通过运行 `./ec2rl help` 获得有关如何使用该工具的一般帮助。您可以通过运行 `./ec2rl list` 查看可用模块，也可以通过运行 `./ec2rl help module_name` 获得有关特定模块的帮助。

下面是您可以执行以便开始使用此工具的常见任务。

任务

- 运行 EC2Rescue 模块
- 上传 EC2Rescue 模块结果
- 创建 Amazon EC2 Linux 实例的备份

运行 EC2Rescue 模块

运行所有 EC2Rescue 模块

使用 ./ec2rl run 命令而不指定任何附加参数。有些模块需要根访问权限。如果您不是根用户，请在运行命令时使用 sudo。

```
./ec2rl run
```

运行特定的 EC2Rescue 模块

使用 ./ec2rl run 命令并为 `--only-modules` 指定要运行的模块的名称。有些模块需要参数才能使用。

```
./ec2rl run --only-modules=module_name --arguments
```

例如，要运行 dig 模块查询 `amazon.com` 域，请使用以下命令。

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

查看 EC2Rescue 模块的结果

运行该模块，然后查看 `cat /var/tmp/ec2rl/logfile_location` 中的日志文件。例如，可在以下位置找到 dig 模块的日志文件：

```
cat /var/tmp/ec2rl/timestamp/mod_out/run/dig.log
```

上传 EC2Rescue 模块结果

如果 Amazon Web Services 支持 请求 EC2Rescue 模块的结果，则可以使用 EC2Rescue 工具上传日志文件。您可以将结果上传到 Amazon Web Services 支持 提供的位置或您自己的 Amazon S3 存储桶。

将结果上传到 Amazon Web Services 支持 提供的位置

使用 ./ec2rl upload 命令。对于 `--upload-directory`，指定日志文件的位置。对于 `--support-url`，请指定 Amazon Web Services 支持 提供的 URL。

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --support-
url="url_provided_by_aws_support"
```

将结果上传到 Amazon S3 存储桶

使用 ./ec2rl upload 命令。对于 `--upload-directory`，指定日志文件的位置。对于 `--presigned-url`，为 S3 存储桶指定预签名 URL。有关为 Amazon S3 生成预签名 URL 的更多信息，请参阅使用预签名 URL 上传对象。

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --presigned-
url="presigned_s3_url"
```

创建 Amazon EC2 Linux 实例的备份

您可以使用 EC2Rescue 通过创建 AMI 或创建其附加卷的快照来备份您的 Linux 实例。

创建 AMI

使用 `./ec2rl run` 命令，并为 `--backup` 指定 `ami`。

```
./ec2rl run --backup=ami
```

创建所有附加卷的多卷快照

使用 `./ec2rl run` 命令，并为 `--backup` 指定 `allvolumes`。

```
./ec2rl run --backup=allvolumes
```

创建特定附加卷的快照

使用 `./ec2rl run` 命令，并为 `--backup` 指定要备份的卷的 ID。

```
./ec2rl run --backup=vol-01234567890abcdef
```

为 Amazon EC2 Linux 实例开发 EC2Rescue 模块

模块使用 YAML 编写，这是一种数据序列化标准。模块的 YAML 文件包括一个文档，用于表示模块及其属性。

添加模块属性

下表列出了可用的模块属性。

属性	说明
name	模块的名称。该名称长度应少于或等于 18 个字符。
版本	模块的版本号。
删除实例快照	模块的简短说明性标题。此值的长度应少于或等于 50 个字符。
helptext	模块的详细说明。每一行的长度应少于或等于 75 个字符。如果模块使用必需或可选参数，请在 helptext 值中包括这些参数。 例如： `helptext: !!str |`

属性	说明
	Collect output from ps for system analysis Consumes --times= for number of times to repeat Consumes --period= for time period between repetition
placement	运行模块的阶段。支持的值： • prediagnostic • run • postdiagnostic
language	编写模块代码使用的语言。支持的值： • bash • python Note Python 代码必须同时兼容 Python 2.7.9+ 和 Python 3.2+。
修复	指示模块是否支持修正。支持的值为 `True` 或 `False`。 模块默认为 `False`（如果不存在），这使其成为那些不支持修正的模块的可选属性。
content	整个脚本代码。
constraint	包含限制值的对象的名称。

属性	说明
域	说明如何分组或分类模块的描述符。所包括的模块组使用以下域： • application • net • os • performance
class	由模块执行的任务类型的描述符。所包括的模块组使用以下类： • collect (从程序收集输出) • diagnose (基于一组标准确定通过/失败) • gather (复制文件并写入特定文件)
distro	此模块支持的 Linux 发行版的列表。所包含的模块组使用以下发行版： • alami (Amazon Linux) • rhel • ubuntu • suse
必需	模块从 CLI 选项使用的必需参数。
optional	模块可使用的可选参数。
软件	模块中使用的软件可执行文件。此属性用于指定默认情况下未安装的软件。EC2Rescue for Linux 逻辑在运行模块之前确保这些程序存在并且可执行。
程序包	可执行文件的源软件包。此属性用于随软件提供软件包的详细信息，包括用于下载或者获取更多信息的 URL。

属性	说明
sudo	指示运行模块是否需要根访问权限。 您无需在模块脚本中实施 sudo 检查。如果值为 true，则 EC2Rescue for Linux 逻辑仅在执行用户具有根访问权限时才运行模块。
perfimpact	指示模块对其运行环境是否会产生重大性能影响。如果值为 true 并且没有 `--perfimpact=true` 参数，则跳过模块。
parallelexclusive	指定需要互斥的程序。例如，所有指定“bpf”的模块以串行方式运行。

设置环境变量

下表列出了可用的环境变量。

环境变量	说明
`EC2RL_CALLPATH`	`ec2rl.py` 的路径。此路径可用于定位 lib 目录和使用分发的 Python 模块。
`EC2RL_WORKDIR`	诊断工具的主 tmp 目录。 默认值：`/var/tmp/ec2rl` 。
`EC2RL_RUNDIR`	用于存储所有输出的目录。 默认值：`/var/tmp/ec2rl/<date&timestamp>` 。
`EC2RL_GATHEREDDIR`	用于放置收集的模块数据的根目录。 默认值:`/var/tmp/ec2rl/<date&timestamp>/mod_out/gathered/` 。

环境变量	说明
EC2RL_NET_DRIVER	为实例上第一个 (按照字母顺序排序) 非虚拟网络接口使用的驱动程序。 示例： • xen_netfront • ixgbevf • ena
EC2RL_SUDO	如果 EC2Rescue for Linux 以根身份运行，则为 true；否则为 false。
EC2RL_VIRT_TYPE	由实例元数据提供的虚拟化类型。 示例： • default-hvm • default-paravirtual
EC2RL_INTERFACES	系统上的接口枚举列表。该值为包含名称的字符串，例如 eth0、eth1 等。这通过 functions.bash 生成，仅对其来源模块可用。

使用 YAML 语法

在您构建模块 YAML 文件时，应注意以下事项：

- 三个连字符 (---) 表示文档的明确开始位置。
- !ec2rlcore.module.Module 标签指示 YAML 分析器在从数据流创建对象时调用哪个构造函数。您可在 module.py 文件内部查找构造函数。
- !!str 标签告知 YAML 解析器不尝试确定数据的类型，而是将内容解释为字符串文本。
- 竖线字符 (|) 告知 YAML 解析器该值为文字类型的标量。在这种情况下，解析器包括所有空格。对于模块而言这非常重要，因为保留了缩进和换行字符。
- YAML 标准缩进为两个空格，在下例中可以看到。请确保您为脚本保留了标准缩进 (例如，对于 Python 为四个空格)，然后在模块文件中将全部内容缩进两个空格。

模块示例

示例 1 (mod.d/ps.yaml) :

```
--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
  Collect output from ps for system analysis
  Requires --times= for number of times to repeat
  Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
      printf "%0.s=" {1..80}
      echo -e "\nERROR: "$BASH_COMMAND" exited with an error on line ${BASH_LINENO[0]}"
      exit 0
  }
  trap error_trap ERR

  # read-in shared function
  source functions.bash
  echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every
 $period seconds."
  for i in $(seq 1 $times); do
      ps auxww
      sleep $period
  done
constraint:
  requires_ec2: !!str False
  domain: !!str performance
  class: !!str collect
  distro: !!str alami ubuntu rhel suse
  required: !!str period times
  optional: !!str
  software: !!str
```

```
  sudo: !!str False
  perfimpact: !!str False
  parallelexclusive: !!str
```

使用 EC2Rescue 对受损的 Amazon EC2 Windows 实例进行问题排查

EC2Rescue for Windows Server 是一个易于使用的工具，可以在 Amazon EC2 Windows Server 实例上运行，以诊断和排查可能存在的问题。它不仅可以收集日志文件以及进行故障排除，还会主动搜索可能存在问题的区域，具有很高的价值。它甚至可以检查其他实例中的 Amazon EBS 根卷，并收集相关日志以对使用该卷的 Windows Server 实例进行故障排除。以下是 EC2Rescue 可以解决的一些常见问题：

- 防火墙、远程桌面协议（RDP）或网络接口配置导致的实例连接问题
- 停止错误、启动循环或注册表损坏导致的操作系统启动问题
- 可能需要进行高级日志分析和问题排查的问题

EC2Rescue for Windows Server 有两个不同模块：

- 一个是数据收集器模块，用于收集所有不同来源的数据
- 一个是分析器模块，用于解析根据一系列预定义规则收集的数据，从而识别问题并提供建议

EC2Rescue for Windows Server 工具只能在运行 Windows Server 2012 及更高版本的 Amazon EC2 实例上运行。该工具启动后，将检查它是否正在 Amazon EC2 实例上运行。

> **Note**
>
> `AWSSupport-ExecuteEC2Rescue` Amazon Systems Manager 自动化运行手册使用 EC2Rescue 工具进行问题排查，并尽可能修复指定 EC2 实例的常见连接问题。有关详细信息以及要运行此自动化，请参阅 >AWSSupport-ExecuteEC2Rescue。

如果您使用的是 Linux 实例，则请参阅 the section called “适用于 Linux 实例的 EC2Rescue”。

主题

- 使用 EC2Rescue GUI 对受损的 Windows 实例进行问题排查

- 使用 EC2Rescue CLI 对受损的 Windows 实例进行问题排查
- 使用 EC2Rescue 和 Systems Manager 对受损的 Windows 实例进行问题排查

使用 EC2Rescue GUI 对受损的 Windows 实例进行问题排查

EC2Rescue for Windows Server 可以对离线实例执行以下分析：

选项	描述

诊断和抢救	EC2Rescue for Windows Server 可以使用以下服务设置检测和解决问题：
	• 系统时间
	  • RealTimeisUniversal – 检测是否已启用 `RealTimeisUniversal` 注册表项。如果禁用，当时区设置为非 UTC 的值时，Windows 系统时间将有偏差。
	• Windows 防火墙
	  • 域网络 – 检测是否已启用或禁用此 Windows 防火墙配置文件。
	  • 私有网络 – 检测是否已启用或禁用此 Windows 防火墙配置文件。
	  • 访客或公有网络 – 检测是否已启用或禁用此 Windows 防火墙配置文件。
	• 远程桌面
	  • 服务启动 – 检测是否已启用远程桌面服务。
	  • 远程桌面连接 – 检测是否已启用。
	  • TCP 端口 – 检测远程桌面服务正在侦听的端口。
	• EC2Config（Windows Server 2012 R2 及更早版本）

选项	描述
	• 安装 – 检测所安装的 EC2Config 版本。 • 服务启动 – 检测是否已启用 EC2Config 服务。 • Ec2SetPassword – 生成新的管理员密码。 • Ec2HandleUserData – 在实例下一次启动时允许您运行用户数据脚本。 • EC2Launch（Windows Server 2016 及更高版本） • 安装 – 检测所安装的 EC2Launch 版本。 • Ec2SetPassword – 生成新的管理员密码。 • 网络接口 • DHCP 服务启动 – 检测是否已启用 DHCP 服务。 • 以太网详细信息 – 显示检测到的网络驱动器的版本信息。 • 以太网上的 DHCP – 检测是否已启用 DHCP。 • 磁盘签名状态 • Signature on disk（磁盘上的签名）和 Signature on Boot Configuration Database (BCD)（启动配置数据库 (BCD) 上的签名）– 检测磁盘签名和 BCD 签名是否相同。如果值不同，EC2Rescue 会尝试使用 BCD 上的签名覆盖磁盘签名。

选项	描述
Restore	执行以下操作之一： • 上次已知的正常配置 – 尝试以上次已知的可启动状态启动实例。 • 从备份还原注册表 – 从 `\Windows\System32\config\RegBack` 还原注册表。
捕获日志	允许您捕获实例的日志，用于分析。

EC2Rescue for Windows Server 可以从活动实例和离线实例收集以下数据：

Item	说明
Event Log (事件日志)	收集应用程序、系统和 EC2Config 事件日志。
注册表	收集 SYSTEM 和 SOFTWARE Hive。
Windows 更新日志	收集 Windows 更新生成的日志文件。 **Note** 在 Windows Server 2016 及更高版本中，日志以 Windows 事件跟踪 (ETW) 格式收集。
Sysprep 日志	收集 Windows 系统准备工具生成的日志文件。
驱动程序安装日志	收集 Windows SetupAPI 日志 (`setupapi.dev.log` 和 `setupapi.setup.log`)。
启动配置	收集 `HKEY_LOCAL_MACHINE\BCD00000000` Hive。
内存转储	收集实例上的所有内存转储文件。

Item	说明
EC2Config 文件	收集 EC2Config 服务生成的日志文件。
EC2Launch 文件	收集 EC2Launch 脚本生成的日志文件。
SSM Agent 文件	收集 SSM Agent 生成的日志文件和 Patch Manager 日志。
EC2 ElasticGPU 文件	收集与 Elastic GPU 相关的事件日志。
ECS	收集与 Amazon ECS 相关的日志。
CloudEndure	收集与 CloudEndure 代理相关的日志文件。
适用于 MGN 或 DRS 日志文件的 Amazon Replication Agent	收集与 Amazon Transform MGN 或 Amazon 弹性灾难恢复相关的日志文件。

EC2Rescue for Windows Server 可以从活动实例收集以下额外数据：

Item	说明
信息系统	收集 MSInfo32。
组策略结果	收集组策略报告。

分析离线实例

Offline Instance 选项在调试 Windows 实例的启动问题时很有用。

在离线实例上执行操作

1. 在正在运行的 Windows Server 实例上，下载 EC2Rescue for Windows Server 工具并提取文件。

   您可以运行以下 PowerShell 命令以下载 EC2Rescue，而无需更改 Internet Explorer 增强安全性配置 (ESC)：

```
Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
```

该命令将 EC2Rescue .zip 文件下载到当前登录用户的桌面中。

> **Note**
>
> 如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

2. 停止出现故障的实例 (如果目前尚未停止)。
3. 从出现故障的实例中分离 EBS 根卷，并将该卷附加到安装了 EC2Rescue for Windows Server 的正在运行的 Windows 实例。
4. 在正在运行的实例上运行 EC2Rescue for Windows Server 工具并选择离线实例。
5. 选择新装载卷的磁盘，并选择 Next。
6. 确认所选磁盘并选择 Yes。
7. 选择要执行的离线实例选项，并选择 Next。

EC2Rescue for Windows Server 工具会扫描该卷，并根据所选的日志文件收集故障排除信息。

收集来自活动实例的数据

您可以收集活动实例的日志和其他数据。

收集来自活动实例的数据

1. 连接到您的 Windows 实例。
2. 将 EC2Rescue for Windows Server 工具下载到您的 Windows 实例并提取文件。

   您可以运行以下 PowerShell 命令以下载 EC2Rescue，而无需更改 Internet Explorer 增强安全性配置 (ESC)：

```
Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
```

该命令将 EC2Rescue .zip 文件下载到当前登录用户的桌面中。

> **Note**
>
> 如果您在下载文件时收到错误，并且正在使用 Windows Server 2016 或更早的版本，则可能需要为 PowerShell 终端启用 TLS 1.2。可以使用以下命令为当前 PowerShell 会话启用 TLS 1.2，然后重试：
>
> ```
> [Net.ServicePointManager]::SecurityProtocol =
>  [Net.SecurityProtocolType]::Tls12
> ```

3. 打开 EC2Rescue for Windows Server 应用程序并接受许可协议。
4. 选择 Next、Current instance、Capture logs。
5. 选择要收集的数据项，然后选择 Collect...。阅读警告并选择 Yes 以继续操作。
6. 选择 ZIP 文件的文件名和位置，然后选择 Save。
7. EC2Rescue for Windows Server 完成之后，选择打开包含文件夹查看 ZIP 文件。
8. 选择结束。

使用 EC2Rescue CLI 对受损的 Windows 实例进行问题排查

利用 EC2Rescue for Windows Server 命令行界面 (CLI)，您可以编程方式运行 EC2Rescue for Windows Server 插件（称为“操作”）。

EC2Rescue for Windows Server 工具有两种执行模式：

- /online — 这允许您对安装了 EC2Rescue for Windows Server 的实例执行操作（如收集日志文件）。
- /offline:<device_id> — 这允许您对附加到单独的 Amazon EC2 Windows 实例（已安装 EC2Rescue for Windows Server）的脱机根卷执行操作。

将 EC2Rescue for Windows Server 工具下载到您的 Windows 实例并提取文件。您可以使用以下命令查看帮助文件：

```
EC2RescueCmd.exe /help
```

EC2Rescue for Windows Server 可对 Amazon EC2 Windows 实例执行下列操作：

- 收集操作
- 救援行动
- 还原操作

收集操作

> **Note**
>
> 您可以收集所有日志、整个日志组或一个组中的单个日志。

EC2Rescue for Windows Server 可以从活动实例和离线实例收集以下数据。

日志组	可用日志	说明
`all`		收集所有可用日志。
`eventlog`	• `'Application'` • `'System'` • `'EC2ConfigService'`	收集应用程序、系统和 EC2Config 事件日志。
`memory-dump`	• `'Memory Dump File'` • `'Mini Dump Files'`	收集实例上的所有内存转储文件。
`ec2config`	• `'Log Files'` • `'Configuration Files'`	收集 EC2Config 服务生成的日志文件。
`ec2launch`	• `'Logs'` • `'Config'`	收集 EC2Launch 脚本生成的日志文件。
`ssm-agent`	• `'Log Files'`	收集 SSM Agent 生成的日志文件和 Patch Manager 日志。

日志组	可用日志	说明
	• 'Patch Baseline Logs' • 'InstanceData'	
sysprep	'Log Files'	收集 Windows 系统准备工具生成的日志文件。
driver-setup	• 'SetupAPI Log Files' • 'DPInst Log File' • 'Amazon PV Setup Log File'	收集 Windows SetupAPI 日志 (setupapi.dev.log 和 setupapi.setup.log)。
registry	• 'SYSTEM' • 'SOFTWARE' • 'BCD'	收集 SYSTEM 和 SOFTWARE Hive。
egpu	• 'Event Log' • 'System Files'	收集与 Elastic GPU 相关的事件日志。
boot-config	'BCDEDIT Output'	收集 HKEY_LOCAL_MACHINE\BCD00000000 Hive。
windows-update	'Log Files'	收集 Windows 更新生成的日志文件。 **Note** 在 Windows Server 2016 及更高版本中，日志以 Windows 事件跟踪 (ETW) 格式收集。

日志组	可用日志	说明
cloudendure	• 'Migrate Script Logs' • 'Driver Logs' • 'CloudEndure File List'	收集与 CloudEndure 代理相关的日志文件。

EC2Rescue for Windows Server 可以从活动实例收集以下额外数据。

日志组	可用日志	说明
system-info	'MSInfo32 Output'	收集 MSInfo32。
gpresult	'GPResult Output'	收集组策略报告。

可用选项如下：

- /output:<outputFilePath> – 使用 zip 格式保存收集的日志文件所需的目标文件路径位置。
- /no-offline – 离线模式下使用的可选属性。不会在完成操作后将卷设置为脱机。
- /no-fix-signature – 离线模式下使用的可选属性。不会在完成操作后修复可能的磁盘签名冲突。

示例

以下示例使用的是 EC2Rescue for Windows Server CLI。

在线模式示例

收集所有可用日志：

```
EC2RescueCmd /accepteula /online /collect:all /output:<outputFilePath>
```

仅收集特定日志组：

```
EC2RescueCmd /accepteula /online /collect:ec2config /output:<outputFilePath>
```

收集日志组中的单个日志：

```
EC2RescueCmd /accepteula /online /collect:'ec2config.Log Files,driver-setup.SetupAPI
 Log Files' /output:<outputFilePath>
```

离线模式示例

从 EBS 卷收集所有可用日志。该卷按 device_id 值指定。

```
EC2RescueCmd /accepteula /offline:xvdf /collect:all /output:<outputFilePath>
```

仅收集特定日志组：

```
EC2RescueCmd /accepteula /offline:xvdf /collect:ec2config /output:<outputFilePath>
```

救援行动

EC2Rescue for Windows Server 可以使用以下服务设置检测和解决问题：

服务组	可用操作	说明
all		
system-time	'RealTimeIsUniversal'	系统时间 • RealTimeisUniversal – 检测是否已启用 RealTimeisUniversal 注册表项。如果禁用，当时区设置为非 UTC 的值时，Windows 系统时间将有偏差。
firewall	• 'Domain networks' • 'Private networks' • 'Guest or public networks'	Windows 防火墙 • 域网络 – 检测是否已启用或禁用此 Windows 防火墙配置文件。 • 私有网络 – 检测是否已启用或禁用此 Windows 防火墙配置文件。

服务组	可用操作	说明
		• 访客或公有网络 – 检测是否已启用或禁用此 Windows 防火墙配置文件。
rdp	• 'Service Start' • 'Remote Desktop Connections' • 'TCP Port'	远程桌面 • 服务启动 – 检测是否已启用远程桌面服务。 • 远程桌面连接 – 检测是否已启用。 • TCP 端口 – 检测远程桌面服务正在侦听的端口。
ec2config	• 'Service Start' • 'Ec2SetPassword' • 'Ec2HandleUserData'	EC2Config • 服务启动 – 检测是否已启用 EC2Config 服务。 • Ec2SetPassword – 生成新的管理员密码。 • Ec2HandleUserData – 在实例下一次启动时允许您运行用户数据脚本。
ec2launch	'Reset Administrator Password'	生成新的 Windows 管理员密码。
network	'DHCP Service Startup'	网络接口 • DHCP 服务启动 – 检测是否已启用 DHCP 服务。

可用选项如下：

- /level:<level> – 操作应触发的检查级别的可选属性。允许的值包括：information、warning、error、all。默认情况下，将它设置为 error。
- /check-only – 可生成报告但不修改离线卷的可选属性。

> **Note**
>
> 如果 EC2Rescue for Windows Server 检测到可能存在磁盘签名冲突，则默认情况下，即使您使用 `/check-only` 选项，它也会在此离线过程完成后更正签名。必须使用 `/no-fix-signature` 选项才能阻止更正。

- /no-offline – 可防止在完成操作之后将卷设置为离线的可选属性。
- /no-fix-signature – 不会在完成操作后修复可能的磁盘签名冲突的可选属性。

救援示例

以下示例使用的是 EC2Rescue for Windows Server CLI。该卷是使用 device_id 值指定的。

尝试修复在卷上发现的所有问题：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:all
```

尝试修复卷上的一个服务组内的所有问题：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:firewall
```

尝试修复卷上的一个服务组内的特定项目：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:rdp.'Service Start'
```

指定在一个卷上要尝试修复的多个问题：

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:'system-
time.RealTimeIsUniversal,ec2config.Service Start'
```

还原操作

EC2Rescue for Windows Server 可以使用以下服务设置检测和解决问题：

服务组	可用操作	说明
还原上次已知的正常配置	`lkgc`	上次已知的正常配置 – 尝试以上次已知的可启动状态启动实例。
从最新备份还原 Windows 注册表	`regback`	从备份还原注册表 – 从 `\Windows\System32\config\RegBack` 还原注册表。

可用选项如下：

- /no-offline — 可防止在完成操作之后将卷设置为脱机的可选属性。
- /no-fix-signature — 不会在完成操作后修复可能的磁盘签名冲突的可选属性。

还原示例

以下示例使用的是 EC2Rescue for Windows Server CLI。该卷是使用 device_id 值指定的。

在卷上还原上次已知的正常配置：

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

在卷上还原上一个 Windows 注册表备份：

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

使用 EC2Rescue 和 Systems Manager 对受损的 Windows 实例进行问题排查

Amazon Web Services 支持 为您提供了 Systems Manager Run Command 文档，目的是与您启用了 Systems Manager 的实例交互来运行 EC2Rescue for Windows Server。Run Command 文档称为 `AWSSupport-RunEC2RescueForWindowsTool`。

此 Systems Manager Run Command 文档执行下列任务：

- 下载并验证 EC2Rescue for Windows Server。
- 导入 PowerShell 模块以使您与此工具的交互变得简单。
- 使用提供的命令和参数运行 EC2RescueCmd。

Systems Manager Run Command 文档接受三个参数：

- 命令 — EC2Rescue for Windows Server 操作。当前允许的值如下：
  - ResetAccess — 重置本地管理员密码。将会重置当前实例的本地管理员密码，并且随机生成的密码将会作为 `/EC2Rescue/Password/<INSTANCE_ID>` 安全地存储在 Parameter Store 中。如果您选择此操作并且不提供任何参数，则密码将自动使用默认的 KMS 密钥 密钥加密。（可选）可以在参数中指定 KMS 密钥 ID，以使用您自己的密钥来加密密码。
  - CollectLogs — 使用 `/collect:all` 操作运行 EC2Rescue for Windows Server。如果选择此操作，`Parameters` 必须包含将日志上传到的 Amazon S3 存储桶名称。
  - FixAll — 使用 `/rescue:all` 操作运行 EC2Rescue for Windows Server。如果您选择此操作，`Parameters` 必须包含要抢救的块储存设备名称。
- 参数 — 要为指定命令传递的 PowerShell 参数。

要求

要运行 ResetAccess 操作，您的 Amazon EC2 实例必须附加策略，以授予将加密的密码写入 Parameter Store 的权限。附加策略后，请等待几分钟，然后再将此策略附加到相关 IAM 角色后尝试重置实例的密码。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
```

```
    }
  ]
}
```

如果您使用的是自定义 KMS 密钥而不是默认 KMS 密钥，请使用此策略。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<kmskeyid>"
      ]
    }
  ]
}
```

查看文档的 JSON

以下过程介绍了如何查看此文档的 JSON。

查看 Systems Manager Run Command 文档的 JSON

1. 访问 https://console.aws.amazon.com/systems-manager/，打开 Amazon Systems Manager 控制台。
2. 在导航窗格中，展开变更管理工具，然后选择文档。
3. 在搜索栏中，输入 AWSSupport-RunEC2RescueForWindowsTool，然后选择 AWSSupport-RunEC2RescueForWindowsTool 文档。
4. 选择内容选项卡。

示例

下面是一些有关如何使用 Systems Manager Run Command 文档通过 Amazon CLI 运行 EC2Rescue for Windows Server 的示例。有关使用 Amazon CLI 发送命令的更多信息，请参阅 send-command。

示例

- 尝试修复脱机根卷上的所有已识别问题
- 从当前 Amazon EC2 Windows 实例收集日志
- 重置本地管理员密码

尝试修复脱机根卷上的所有已识别问题

尝试修复在附加到 Amazon EC2 Windows 实例的脱机根卷上发现的所有问题：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=FixAll, Parameters='xvdf'" --output
 text
```

从当前 Amazon EC2 Windows 实例收集日志

从当前的在线 Amazon EC2 Windows 实例收集所有日志并将日志上传到 Amazon S3 存储桶：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=CollectLogs, Parameters='amzn-s3-
demo-bucket'" --output text
```

重置本地管理员密码

下面的示例显示了可用于重置本地管理员密码的方法。输出提供了一个指向 Parameter Store 的链接，在那里您可以找到随机生成的安全密码，然后您可以使用它以本地管理员身份来 RDP 到您的 Amazon EC2 Windows 实例。

使用默认的 Amazon KMS key 密钥 alias/aws/ssm 重置联机实例的本地管理员密码：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess" --output text
```

使用 KMS 密钥 重置联机实例的本地管理员密码：

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess, Parameters=a133dc3c-
a2g4-4fc6-a873-6c0720104bf0" --output text
```

> **Note**
>
> 在此示例中，KMS 密钥 为 a133dc3c-a2g4-4fc6-a873-6c0720104bf0。

适用于实例的 EC2 Serial Console

使用 EC2 Serial Console，您可以访问 Amazon EC2 实例的串行端口，通过该端口可以排查启动、网络配置和其他方面的问题。串行控制台不要求您的实例拥有任何联网功能。使用串行控制台，您可以向实例输入命令，就像键盘和显示器直接连接到实例的串行端口一样。串行控制台会话在实例重启和停止期间持续存在。在重新启动过程中，您可以从一开始就查看所有引导消息。

默认情况下，对串行控制台的访问权限不可用。您的组织必须授予账户对串行控制台的访问权限，并配置 IAM policy 以授予用户对串行控制台的访问权限。通过使用实例 ID、资源标签和其他 IAM 操作，可以精细地控制串行控制台访问。有关更多信息，请参阅配置对 EC2 Serial Console 的访问。

串行控制台可使用 EC2 控制台或 Amazon CLI 访问。

串行控制台不会产生额外的成本。

主题

- EC2 Serial Console 的先决条件
- 配置对 EC2 Serial Console 的访问
- 连接到 EC2 Serial Console
- 断开与 EC2 Serial Console 的连接
- 使用 EC2 Serial Console 对 Amazon EC2 实例进行故障排查

EC2 Serial Console 的先决条件

要连接到 EC2 Serial Console 并使用选定的工具进行故障排除，必须满足以下先决条件：

- Amazon Web Services 区域
- Wavelength 区域和 Amazon Outposts
- Local Zones
- 实例类型
- 授予访问权限
- 支持基于浏览器的客户端
- 实例状态
- Amazon EC2 Systems Manager
- 配置您选择的故障排除工具

Amazon Web Services 区域

在所有 Amazon Web Services 区域 中支持。

Wavelength 区域和 Amazon Outposts

不支持。

Local Zones

在所有 Local Zones 中支持。

实例类型

支持的实例类型：

- Linux
  - 基于 Nitro 系统构建的所有虚拟化实例。
  - 所有裸机实例，以下项除外：
    - 通用型：a1.metal、mac1.metal、mac2.metal、mac2-m1ultra.metal、mac2-m2.metal、mac2-m2pro.metal、mac-m3ultra.metal、mac-m4.metal、mac-m4max.metal 和 mac-m4pro.metal
    - 加速计算：g5g.metal
    - 内存优化型：u-6tb1.metal、u-9tb1.metal、u-12tb1.metal、u-18tb1.metal 和 u-24tb1.metal
- Windows
  - 基于 Nitro 系统构建的所有虚拟化实例。
  - 所有裸机实例，以下项除外：
    - 通用型：a1.metal、mac1.metal、mac2.metal、mac2-m1ultra.metal、mac2-m2.metal、mac2-m2pro.metal、mac-m3ultra.metal、mac-m4.metal、mac-m4max.metal 和 mac-m4pro.metal
    - 加速计算：g5g.metal
    - 内存优化型：u-6tb1.metal、u-9tb1.metal、u-12tb1.metal、u-18tb1.metal 和 u-24tb1.metal

授予访问权限

您必须完成配置任务才能授予对 EC2 Serial Console 的访问权限。有关更多信息，请参阅 配置对 EC2 Serial Console 的访问。

支持基于浏览器的客户端

如需使用基于浏览器的客户端连接至串行控制台，您的浏览器必须支持 WebSocket。如果您的浏览器不支持 WebSocket，请使用您自己的密钥和 SSH 客户端连接至串行控制台。

实例状态

必须是 running。

如果实例处于 pending、stopping、stopped、shutting-down 或 terminated 状态，则无法连接到串行控制台。

有关实例状态的更多信息，请参阅 Amazon EC2 实例状态更改。

Amazon EC2 Systems Manager

如果实例使用 Amazon EC2 Systems Manager，则必须在实例上安装 SSM Agent 3.0.854.0 版或更高版本。有关 SSM Agent 的更多信息，请参阅《Amazon Systems Manager 用户指南》中的使用 SSM Agent。

配置您选择的故障排除工具

要使用串行控制台对 Windows 实例进行故障排除，在 Linux 实例上可以使用 GRUB 或 SysRq，在 Windows 实例上可使用特殊管理控制台（SAC）。您必须先在要使用这些工具的每个实例上执行配置步骤，然后才可以使用这些工具。

请使用适用于实例操作系统的说明，配置您选择的故障排除工具。

（Linux 实例）配置 GRUB

要配置 GRUB，请根据用于启动实例的 AMI 选择以下程序之一。

Amazon Linux 2

在 Amazon Linux 2 实例上配置 GRUB

1. 使用 SSH 连接到 Linux 实例
2. 在 `/etc/default/grub` 中添加或更改以下选项：
   - 设置 `GRUB_TIMEOUT=1`。
   - 添加 `GRUB_TERMINAL="console serial"`。
   - 添加 `GRUB_SERIAL_COMMAND="serial --speed=115200"`。

   以下是 `/etc/default/grub` 的示例。您可能需要根据系统设置更改配置。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.shell=0"
GRUB_TIMEOUT=1
GRUB_DISABLE_RECOVERY="true"
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

3. 运行以下命令以应用更新后的配置。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Ubuntu

在 Ubuntu 实例上配置 GRUB

1. 连接到您的实例。
2. 在 /etc/default/grub.d/50-cloudimg-settings.cfg 中添加或更改以下选项：
   - 设置 GRUB_TIMEOUT=1。
   - 添加 GRUB_TIMEOUT_STYLE=menu。
   - 添加 GRUB_TERMINAL="console serial"。
   - 删除 GRUB_HIDDEN_TIMEOUT。
   - 添加 GRUB_SERIAL_COMMAND="serial --speed=115200"。

   以下是 /etc/default/grub.d/50-cloudimg-settings.cfg 的示例。您可能需要根据系统设置更改配置。

```
# Cloud Image specific Grub settings for Generic Cloud Images
# CLOUD_IMG: This file was created/modified by the Cloud Image build process

# Set the recordfail timeout
GRUB_RECORDFAIL_TIMEOUT=0

# Do not wait on grub prompt
GRUB_TIMEOUT=1
GRUB_TIMEOUT_STYLE=menu

# Set the default commandline
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
 nvme_core.io_timeout=4294967295"

# Set the grub console type
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed 115200"
```

3. 运行以下命令以应用更新后的配置。

```
[ec2-user ~]$ sudo update-grub
```

RHEL

在 RHEL 实例上配置 GRUB

1. 连接到您的实例。
2. 在 /etc/default/grub 中添加或更改以下选项：
   - 删除 GRUB_TERMINAL_OUTPUT。
   - 添加 GRUB_TERMINAL="console serial"。
   - 添加 GRUB_SERIAL_COMMAND="serial --speed=115200"。

   以下是 /etc/default/grub 的示例。您可能需要根据系统设置更改配置。

   ```
   GRUB_TIMEOUT=1
   GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    rd.blacklist=nouveau nvme_core.io_timeout=4294967295 crashkernel=auto"
   GRUB_DISABLE_RECOVERY="true"
   GRUB_ENABLE_BLSCFG=true
   GRUB_TERMINAL="console serial"
   GRUB_SERIAL_COMMAND="serial --speed=115200"
   ```
3. 运行以下命令以应用更新后的配置。

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
   ```

   对于 RHEL 9.2 及更早版本，请使用以下命令。

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

CentOS

对于使用 CentOS AMI 启动的实例，默认情况下为串行控制台配置了 GRUB。

以下是 /etc/default/grub 的示例。根据您的系统设置，您的配置可能会有所不同。

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
GRUB_CMDLINE_LINUX="console=tty0 crashkernel=auto console=ttyS0,115200"
GRUB_DISABLE_RECOVERY="true"
```

（Linux 实例）配置 SysRq

要配置 SysRq，请在当前引导周期中启用 SysRq 命令。要使配置永久化，您还可以为后续引导启用 SysRq 命令。

为当前引导周期启用所有 SysRq 命令

1. 连接到您的实例。
2. 运行以下命令。

   ```
   [ec2-user ~]$ sudo sysctl -w kernel.sysrq=1
   ```

   下次重新启动时将会清除此设置。

为后续引导启用所有 SysRq 命令

1. 创建文件 /etc/sysctl.d/99-sysrq.conf 并在您收藏的编辑器中打开。

   ```
   [ec2-user ~]$ sudo vi /etc/sysctl.d/99-sysrq.conf
   ```

2. 添加以下行。

   ```
   kernel.sysrq=1
   ```

3. 重启实例以应用更改。

   ```
   [ec2-user ~]$ sudo reboot
   ```

4. 在 login 提示符下，输入您之前设置的、基于密码的用户的用户名，然后按 Enter 键。

5. 在 Password 提示符下，输入密码，然后按 Enter 键。

（Windows 实例）启用 SAC 和启动菜单

> **Note**
>
> 如果您在实例上启用 SAC，则依赖密码检索的 EC2 服务将无法通过 Amazon EC2 控制台运行。Amazon EC2 启动代理（EC2Config、EC2Launch v1 和 EC2Launch v2）上的 Windows 依靠串行控制台来执行各种任务。当您在实例上启用 SAC 时，这些任务不会成功执行。有关 Amazon EC2 启动代理上的 Windows 的更多信息，请参阅 the section called “配置 Windows 实例”。如果您启用 SAC，则可在之后将其禁用。有关更多信息，请参阅 禁用 SAC 和启动菜单。

使用以下方法之一在实例上启用 SAC 和启动菜单。

PowerShell

在 Windows 实例上启用 SAC 和启动菜单

1. 连接到您的实例并从提升的 PowerShell 命令行执行以下步骤。
2. 启用 SAC。

```
bcdedit /ems '{current}' on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. 启用启动菜单。

```
bcdedit /set '{bootmgr}' displaybootmenu yes
bcdedit /set '{bootmgr}' timeout 15
bcdedit /set '{bootmgr}' bootems yes
```

4. 重新启动实例以应用更新后的配置。

```
shutdown -r -t 0
```

Command prompt

在 Windows 实例上启用 SAC 和启动菜单

1. 连接到您的实例并从命令提示符执行以下步骤。
2. 启用 SAC。

```
bcdedit /ems {current} on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. 启用启动菜单。

```
bcdedit /set {bootmgr} displaybootmenu yes
bcdedit /set {bootmgr} timeout 15
bcdedit /set {bootmgr} bootems yes
```

4. 重新启动实例以应用更新后的配置。

```
shutdown -r -t 0
```

配置对 EC2 Serial Console 的访问

要配置对串行控制台的访问权限，您必须在账户级别授予串行控制台访问权限，然后配置 IAM policy 以向用户授予访问权限。对于 Linux 实例，您还必须在每个实例上配置基于密码的用户，以使您的用户能够使用串行控制台进行问题排查。

EC2 Serial Console 使用虚拟串行端口连接与实例进行交互。此连接独立于实例 VPC，因此即使出现引导故障或网络配置问题，您仍可以使用实例操作系统和运行故障排除工具。由于此连接在 VPC 网络之外，因此 EC2 Serial Console 不使用实例安全组或子网网络 ACL 来授权实例的流量。

开始前的准备工作

验证是否满足先决条件。

内容

- EC2 Serial Console 的访问级别
- 管理账户对 EC2 Serial Console 的访问权限
- 为 EC2 Serial Console 访问配置 IAM policy

- 在 Linux 实例上设置操作系统用户密码

EC2 Serial Console 的访问级别

默认情况下，在账户级别无法访问串行控制台。您需要在账户级别明确授予对串行控制台的访问权限。有关更多信息，请参阅管理账户对 EC2 Serial Console 的访问权限。

您可以使用服务控制策略 (SCP) 来允许在组织内访问串行控制台。然后，您可以使用 IAM policy 控制访问，从而在用户级别实现精确的访问控制。通过使用 SCP 和 IAM policy 的组合，您可以对串行控制台进行不同级别的访问控制。

组织级别

您可以使用服务控制策略 (SCP) 来允许组织内的成员账户访问串行控制台。有关 SCP 的更多信息，请参阅 Amazon Organizations 用户指南中的服务控制策略。

实例级别

您可以通过使用 IAM PrincipalTag 和 ResourceTag 构造以及通过其 ID 指定实例，来配置串行控制台访问策略。有关更多信息，请参阅 为 EC2 Serial Console 访问配置 IAM policy。

用户级别

您可以通过配置 IAM policy 以允许或拒绝指定用户将 SSH 公钥推送到特定实例的串行控制台服务的权限，来配置用户级别的访问权限。有关更多信息，请参阅 为 EC2 Serial Console 访问配置 IAM policy。

操作系统级别（仅限 Linux 实例）

您可以在访客操作系统级别设置用户密码。这为某些使用案例提供了对串行控制台的访问权限。但是，要监控日志，您不需要基于密码的用户。有关更多信息，请参阅在 Linux 实例上设置操作系统用户密码。

管理账户对 EC2 Serial Console 的访问权限

默认情况下，在账户级别无法访问串行控制台。您需要在账户级别明确授予对串行控制台的访问权限。

此设置是在账户级别配置，可以直接在账户中配置，也可以使用声明性策略进行配置。必须在要授予串行控制台访问权限的每个 Amazon Web Services 区域中配置该设置。使用声明性策略可同时将设置应用于多个区域，也可以同时应用于多个账户。当使用声明性策略时，您无法直接在账户中修改设置。本主题介绍如何直接在账户中配置设置。有关使用声明性策略的信息，请参阅《Amazon Organizations 用户指南》中的声明性策略。

目录

- 授予用户账户访问管理权限
- 查看账户对串行控制台的访问权限状态
- 授予账户访问串行控制台的权限。
- 拒绝账户访问串行控制台

授予用户账户访问管理权限

要允许用户管理对 EC2 Serial Console 的账户访问权限，您需要授予其所需的 IAM 权限。

以下策略授予查看账户状态以及允许和阻止账户访问 EC2 Serial Console 的权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:EnableSerialConsoleAccess",
                "ec2:DisableSerialConsoleAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

有关更多信息，请参阅 IAM 用户指南 中的创建 IAM 策略。

查看账户对串行控制台的访问权限状态

Console

查看账户对串行控制台的访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。

3. 在账户属性卡的设置下，选择 EC2 Serial Console。
4. 在 EC2 Serial Console 选项卡上，EC2 Serial Console 访问权限的值为已允许或已阻止。

Amazon CLI

查看账户对串行控制台的访问权限

使用 get-serial-console-access-status 命令。

```
aws ec2 get-serial-console-access-status
```

下面是示例输出。值为 `true` 表示允许该账户访问串行控制台。

```
{
    "SerialConsoleAccessEnabled": true,
    "ManagedBy": "account"
}
```

`ManagedBy` 字段表示配置了该设置的实体。可能的值是 `account`（直接配置）或 `declarative-policy`。有关更多信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

PowerShell

查看账户对串行控制台的访问权限

使用 Get-EC2SerialConsoleAccessStatus cmdlet。

```
Get-EC2SerialConsoleAccessStatus -Select *
```

下面是示例输出。值为 `True` 表示允许该账户访问串行控制台。

```
ManagedBy SerialConsoleAccessEnabled
--------- --------------------------
account   True
```

`ManagedBy` 字段表示配置了该设置的实体。可能的值是 `account`（直接配置）或 `declarative-policy`。有关更多信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies。

授予账户访问串行控制台的权限。

Console

授予账户对串行控制台的访问权限

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡的设置下，选择 EC2 Serial Console。
4. 选择管理。
5. 要允许访问账户中所有实例的 EC2 Serial Console，请选中允许复选框。
6. 选择更新。

Amazon CLI

授予账户对串行控制台的访问权限

使用 enable-serial-console-access 命令。

```
aws ec2 enable-serial-console-access
```

下面是示例输出。

```
{
    "SerialConsoleAccessEnabled": true
}
```

PowerShell

授予账户对串行控制台的访问权限

使用 Enable-EC2SerialConsoleAccess cmdlet。

```
Enable-EC2SerialConsoleAccess
```

下面是示例输出。

```
True
```

拒绝账户访问串行控制台

Console

拒绝账户访问串行控制台

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择控制面板。
3. 在账户属性卡的设置下，选择 EC2 Serial Console。
4. 选择管理。
5. 要阻止访问账户中所有实例的 EC2 Serial Console，请清除允许复选框。
6. 选择更新。

Amazon CLI

拒绝账户访问串行控制台

使用 disable-serial-console-access 命令。

```
aws ec2 disable-serial-console-access
```

下面是示例输出。

```
{
    "SerialConsoleAccessEnabled": false
}
```

PowerShell

拒绝账户访问串行控制台

使用 Disable-EC2SerialConsoleAccess cmdlet。

```
Disable-EC2SerialConsoleAccess
```

下面是示例输出。

```
False
```

为 EC2 Serial Console 访问配置 IAM policy

默认情况下，用户无权访问串行控制台。您的组织必须配置 IAM policy 以授予用户所需的访问权限。有关更多信息，请参阅 IAM 用户指南 中的创建 IAM 策略。

对于串行控制台访问，请创建包含 `ec2-instance-connect:SendSerialConsoleSSHPublicKey` 操作的 JSON 策略文档。此操作会授予用户将公钥推送到串行控制台服务的权限，该服务将启动串行控制台会话。我们建议限制对特定 EC2 实例的访问。否则，具有此权限的所有用户都可以连接到所有 EC2 实例的串行控制台。

示例 IAM policy

- 明确允许访问串行控制台
- 明确拒绝访问串行控制台
- 使用资源标签控制对串行控制台的访问

明确允许访问串行控制台

默认情况下，没有人可以访问串行控制台。要授予对串行控制台的访问权限，您需要将策略配置为明确允许访问。我们建议配置一个策略以限制对特定实例的访问。

以下策略允许访问由实例 ID 标识的特定实例的串行控制台。

请注意，`DescribeInstances`、`DescribeInstanceTypes` 和 `GetSerialConsoleAccessStatus` 操作不支持资源级权限，因此必须为这些操作指定以 *（星号）表示的所有资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
```

```
            ],
             "Resource": "*"
        },
        {
            "Sid": "AllowinstanceBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

明确拒绝访问串行控制台

以下 IAM policy 允许访问标有 *（星号）的所有实例的串行控制台，并明确拒绝访问由 ID 标识的特定实例的串行控制台。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenySerialConsoleAccess",
            "Effect": "Deny",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
```

```
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

使用资源标签控制对串行控制台的访问

您可以使用资源标签来控制对实例的串行控制台的访问。

基于属性的访问控制是一种授权策略，它根据可附加到用户和 Amazon 资源的标签来定义权限。例如，只有当实例的资源标签和主体标签具有相同的标签键 SerialConsole 值时，以下策略才会允许用户启动该实例的串行控制台连接。

有关使用标签控制对Amazon资源的访问的更多信息，请参阅 IAM 用户指南中的控制对Amazon资源的访问。

请注意，DescribeInstances、DescribeInstanceTypes 和 GetSerialConsoleAccessStatus 操作不支持资源级权限，因此必须为这些操作指定以 *（星号）表示的所有资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowTagBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
```

```
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SerialConsole":
 "${aws:PrincipalTag/SerialConsole}"
                }
            }
        }
    ]
}
```

在 Linux 实例上设置操作系统用户密码

您可以在没有密码的情况下连接到串行控制台。但是，要使用串行控制台对 Linux 实例进行故障排查，实例必须拥有基于密码的操作系统用户。

您可以为任何 OS 用户（包括根用户）设置密码。请注意，根用户可以修改所有文件，而每个 OS 用户可能只有有限权限。

您必须为要使用串行控制台的每个实例设置用户密码。这是每个实例的一次性要求。

> **Note**
>
> 默认情况下，Amazon 提供的 AMI 未配置基于密码的用户。如果您使用已配置了根用户密码的 AMI 启动实例，则可以跳过这些说明。

在 Linux 实例上设置操作系统用户密码

1. 连接到您的 实例。除 EC2 Serial Console 连接方法外，您还可以使用任何方法连接到自己的实例。
2. 要为用户设置密码，请使用 passwd 命令。在以下示例中，用户是 `root`。

   ```
   [ec2-user ~]$ sudo passwd root
   ```

   下面是示例输出。

   ```
   Changing password for user root.
   ```

```
New password:
```

3. 出现提 New password 示时，输入新密码。
4. 出现提示时，重新输入密码。

连接到 EC2 Serial Console

您可以使用 Amazon EC2 控制台或通过 SSH 连接到 EC2 实例的串行控制台。连接到串行控制台后，您可以使用它来排除引导、网络配置和其他方面的问题。有关排查故障的更多信息，请参阅 使用 EC2 Serial Console 对 Amazon EC2 实例进行故障排查。

注意事项

- 每个实例仅支持一个活动串行控制台连接。
- 串行控制台连接通常会持续一小时，除非您断开连接。但是，在系统维护期间，Amazon EC2 将断开串行控制台会话。

  连接的持续时间并不是由 IAM 凭证的持续时间决定的。如果您的 IAM 凭证过期，连接将继续保持，直到达到串行控制台连接的最长持续时间。使用 EC2 Serial Console 控制台体验时，如果 IAM 凭证过期，请关闭浏览器页面终止连接。
- 在断开与串行控制台的连接后，需要 30 秒钟来停用会话，然后才能允许新会话。
- 支持的串行控制台端口：ttyS0（Linux 实例）和 COM1（Windows 实例）
- 连接到串行控制台时，您可能会观察到实例的吞吐量略有下降。

主题

- 使用基于浏览器的客户端进行连接
- 使用您自己的密钥和 SSH 客户端进行连接
- EC2 Serial Console 端点和指纹

使用基于浏览器的客户端进行连接

您可以使用基于浏览器的客户端连接到 EC2 实例的串行控制台。您可以通过在 Amazon EC2 控制台中选择实例并选择连接到串行控制台来完成此操作。基于浏览器的客户端处理权限并可提供成功连接。

EC2 Serial Console 适用于大多数浏览器，并支持键盘和鼠标输入。

在连接之前，请确保您已完成先决条件。

使用基于浏览器的客户端（Amazon EC2 控制台）连接到实例的串行端口

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 您可以选择相应实例，然后选择 Actions（操作）、Monitor and troubleshoot（监控和问题排查）、EC2 Serial Console、Connect（连接）。

   您也可以选择相应实例，然后选择 Connect（连接）、EC2 Serial Console、Connect（连接）。

   浏览器内终端窗口随即打开。
4. 按 Enter。如果返回登录提示，则表示您已连接到串行控制台。

   如果保持黑屏，则可以使用以下信息来帮助解决连接到串行控制台的问题：

   - 检查您是否已配置了串行控制台的访问权限。有关更多信息，请参阅 配置对 EC2 Serial Console 的访问。
   - （仅限 Linux 实例）使用 SysRq 连接到串行控制台。SysRq 不要求您使用基于浏览器的客户端进行连接。有关更多信息，请参阅（Linux 实例）使用 SysRq 对您的实例进行故障排除。
   - （仅限 Linux 实例）重新启动 getty。如果您对实例拥有 SSH 访问权限，则使用 SSH 连接到实例，然后使用以下命令重新启动 getty。

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```
   - 重启实例。您可以使用 SysRq（Linux 实例）、EC2 控制台或 Amazon CLI 重启实例 。有关更多信息，请参阅（Linux 实例）使用 SysRq 对您的实例进行故障排除（Linux 实例）或 重启 Amazon EC2 实例。
5. （仅限 Linux 实例）在 `login` 提示符下，输入您之前设置的、基于密码的用户的用户名，然后按 Enter 键。
6. （仅限 Linux 实例）在 `Password` 提示符下，输入密码，然后按 Enter 键。

使用您自己的密钥和 SSH 客户端进行连接

您可以使用自己的 SSH 密钥，并在使用 串行控制台 API 时从您选择的 SSH 客户端连接到您的实例。这使您能够从将公有密钥推送到实例的串行控制台功能中受益。

将 SSH 密钥推送到实例后，SSH 连接不受您配置用于授予用户 EC2 Serial Console 访问权限的 IAM 策略约束。

开始前的准备工作

验证是否满足先决条件。

使用 SSH 连接到实例的串行控制台

1. 将 SSH 公钥推送到实例以启动串行控制台会话

   使用 send-serial-console-ssh-public-key 命令可将 SSH 公有密钥推送到实例。这将启动串行控制台会话。

   如果已为此实例启动串行控制台会话，则该命令将失败，因为您每次只能打开一个会话。在断开与串行控制台的连接后，需要 30 秒钟来停用会话，然后才能允许新会话。

   ```
   aws ec2-instance-connect send-serial-console-ssh-public-key \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --serial-port 0 \
       --ssh-public-key file://my_key.pub \
       --region us-east-1
   ```

2. 使用私钥连接到串行控制台

   在从串行控制台服务删除公钥之前，使用 ssh 命令连接到串行控制台。在移除之前，您有 60 秒钟的时间。

   使用与公钥对应的私钥。

   用户名格式为 `instance-id.port0`，其中包括实例 ID 和端口 0。在以下示例中，用户名为 `i-001234a4bf70dec41EXAMPLE.port0`。

   串行控制台服务的端点因区域而异。有关每个区域的端点，请参阅 EC2 Serial Console 端点和指纹 表。在以下示例中，串行控制台服务位于 us-east-1 区域。

   ```
   ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-
   connect.us-east-1.aws
   ```

   以下示例使用 `timeout 3600` 将 SSH 会话设置为 1 小时后终止。会话终止后，会话期间启动的进程可能会继续在您的实例上运行。

   ```
   timeout 3600 ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-
   instance-connect.us-east-1.aws
   ```

3. （可选）验证指纹

   首次连接到串行控制台时，系统会提示您验证指纹。您可以比较串行控制台指纹与为了进行验证而显示的指纹。如果这些指纹不匹配，则表示有人可能在试图实施“中间人”攻击。如果它们匹配，您可以放心地连接到串行控制台。

   以下指纹用于 us-east-1 区域中的串行控制台服务。有关每个区域的指纹，请参阅 EC2 Serial Console 端点和指纹。

   ```
   SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
   ```

   该指纹仅在您第一次连接到串行控制台时出现。

4. 按 Enter。如果返回提示，则表示您已连接到串行控制台。

   如果保持黑屏，则可以使用以下信息来帮助解决连接到串行控制台的问题：

   - 检查您是否已配置了串行控制台的访问权限。有关更多信息，请参阅 配置对 EC2 Serial Console 的访问。
   - （仅限 Linux 实例）使用 SysRq 连接到串行控制台。SysRq 不要求您使用 SSH 连接。有关更多信息，请参阅 （Linux 实例）使用 SysRq 对您的实例进行故障排除。
   - （仅限 Linux 实例）重新启动 getty。如果您对实例拥有 SSH 访问权限，则使用 SSH 连接到实例，然后使用以下命令重新启动 getty。

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```

   - 重启实例。您可以使用 SysRq（仅限 Linux 实例）、EC2 控制台或 Amazon CLI 重启实例 。有关更多信息，请参阅 （Linux 实例）使用 SysRq 对您的实例进行故障排除（仅限 Linux 实例）或 重启 Amazon EC2 实例。

5. （仅限 Linux 实例）在 `login` 提示符下，输入您之前设置的、基于密码的用户的用户名，然后按 Enter 键。

6. （仅限 Linux 实例）在 `Password` 提示符下，输入密码，然后按 Enter 键。

EC2 Serial Console 端点和指纹

以下是 EC2 Serial Console 的服务端点和指纹。要以编程方式连接到实例的串行控制台，请使用 EC2 Serial Console 端点。EC2 Serial Console 端点和指纹对每个 Amazon 区域都是唯一的。

区域名称	区域	端点	指纹
美国东部（俄亥俄州）	us-east-2	serial-console.ec2-instance-connect.us-east-2.aws	SHA256:EhwPkTzRtTY7TRSzz26XbB0/HvV9jRM7mCZN0xw/d/0
美国东部（弗吉尼亚州北部）	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
美国西部（北加利福尼亚）	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OHldlcMET8u7QLSX3jmRTRAPFHVtqbyoLZBMUCqiH3Y
美国西部（俄勒冈州）	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EMCIe23TqKaBI6yGHainqZcMwqNkDhhAVHa1O2JxVUc
非洲（开普敦）	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RMWWZ2fVePeJUqzjO5jL2KIgXsczoHlz21Ed00biiWI
亚太地区（香港）	ap-east-1	ec2-serial-console.ap-east-1.api.aws	SHA256:T0Q1lpiXxChoZHplnAkjbP7tkm2xXViC9bJFsjYnifk
亚太地区（台北）	ap-east-2	ec2-serial-console.ap-east-2.api.aws	SHA256:z1rsF5DE8aQqsHwBr/D40tR2GnyMqnScCUa1z1eFwDQ

区域名称	区域	端点	指纹
亚太地区（海得拉巴）	ap-south-2	ec2-serial-console.ap-south-2.api.aws	SHA256:WJgPBSwV4/shN+OPITValoewAuYj15DVW845JEhDKRs
亚太地区（雅加达）	ap-southeast-3	ec2-serial-console.ap-southeast-3.api.aws	SHA256:5ZwgrCh+lfns32XITqL/4O0zIfbx4bZgsYFqy3o8mIk
亚太地区（马来西亚）	ap-southeast-5	ec2-serial-console.ap-southeast-5.api.aws	SHA256:cQXTHQMRcqRdIjmAGoAMBSExeoRobYyRwT67yTjnEiA
亚太地区（新西兰）	ap-southeast-6	ec2-serial-console.ap-southeast-6.api.aws	SHA256:wNltdH0gVfM5uHJKjrw+ElFuKoJgalcSKqjCS+lLkv4
亚太地区（墨尔本）	ap-southeast-4	ec2-serial-console.ap-southeast-4.api.aws	SHA256:Avaq27hFgLvjn5gTSShZ0oV7h90p0GG46wfOeT6ZJvM
亚太地区（孟买）	ap-south-1	serial-console.ec2-instance-connect.ap-south-1.aws	SHA256:oBLXcYmklqHHEbliARxEgH8IsO51rezTPiSM35BsU40
亚太地区（大阪）	ap-northeast-3	ec2-serial-console.ap-northeast-3.api.aws	SHA256:Am0/jiBKBnBuFnHr9aXsgEV3G8Tu/vVHFXE/3UcyjsQ

区域名称	区域	端点	指纹
亚太地区（首尔）	ap-northeast-2	serial-console.ec2-instance-connect.ap-northeast-2.aws	SHA256:FoqWXNX+DZ++GuNTztg9PK49WYMqBX+FrcZM2dSrqrI
亚太地区（新加坡）	ap-southeast-1	serial-console.ec2-instance-connect.ap-southeast-1.aws	SHA256:PLFNn7WnCQDHx3qmwLu1Gy/O8TUX7LQgZuaC6L45CoY
亚太地区（悉尼）	ap-southeast-2	serial-console.ec2-instance-connect.ap-southeast-2.aws	SHA256:yFvMwUK9lEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4
亚太地区（泰国）	ap-southeast-7	ec2-serial-console.ap-southeast-7.api.aws	SHA256:KCAZiRYrR1Q2lqsg7vTwixWmvc2wmjVT31XRgSdEfDY
亚太地区（东京）	ap-northeast-1	serial-console.ec2-instance-connect.ap-northeast-1.aws	SHA256:RQfsDCZTOfQawewTRDV1t9Em/HMrFQe+CRlIOT5um4k
加拿大（中部）	ca-central-1	serial-console.ec2-instance-connect.ca-central-1.aws	SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gczYMMO7s4

区域名称	区域	端点	指纹
加拿大西部（卡尔加里）	ca-west-1	ec2-serial-console.ca-west-1.api.aws	SHA256:s3rc8Il2xhbhr3iedjJNxGAFLPGOLjx7IxxXrGckk6Q
中国（北京）	cn-north-1	ec2-serial-console.cn-north-1.api.amazonwebservices.com.cn	SHA256:2gHVFy4H7uU3+WaFUxD28v/ggMeqjvSlgngpgLgGT+Y
中国（宁夏）	cn-northwest-1	ec2-serial-console.cn-northwest-1.api.amazonwebservices.com.cn	SHA256:TdgrNZkiQOdVfYEBUhO4SzUA09VWI5rYOZGTogpwmiM
欧洲地区（法兰克福）	eu-central-1	serial-console.ec2-instance-connect.eu-central-1.aws	SHA256:aCMFS/yIcOdOlkXvOl8AmZ1Toe+bBnrJJ3Fy0k0De2c
欧洲地区（爱尔兰）	eu-west-1	serial-console.ec2-instance-connect.eu-west-1.aws	SHA256:h2AaGAWO4Hathhtm6ezs3Bj7udgUxi2qTrHjZAwCW6E
欧洲地区（伦敦）	eu-west-2	serial-console.ec2-instance-connect.eu-west-2.aws	SHA256:a69rd5CE/AEG4Amm53I6lkD1ZPvS/BCV3tTPW2RnJg8

区域名称	区域	端点	指纹
欧洲地区（米兰）	eu-south-1	ec2-serial-console.eu-south-1.api.aws	SHA256:IC 0kOVJnpgF yBVrxn0A7 n99ecLbXS X95cuuS7X7QK30
欧洲地区（巴黎）	eu-west-3	serial-console.ec2-instance-connect.eu-west-3.aws	SHA256:q8ldnAf9pym eNe8BnFVngY3RPAr/ kxswJUzfrlxeEWs
欧洲（西班牙）	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:Go CW2DFRlu6 69QNxqFxE csR6fZUz/4F4n7T45Z cwoEc
欧洲地区（斯德哥尔摩）	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tk GFFUVUDvo cDiGSS3Cu 8Gdl6w2uI 32EPNpKFKLwX84
欧洲（苏黎世）	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8P px2mBMf6W dCw0NUlzKfwM4/IfRz 4OaXFutQXWp6mk
以色列（特拉维夫）	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR 6q8v6kNNP i8+QSFQ4d j5dimNmZP TgwgsM1SNvtYyU

区域名称	区域	端点	指纹
墨西哥（中部）	mx-central-1	ec2-serial-console.mx-central-1.api.aws	SHA256:BCuVl13iQNk+CcVnt18Ef4p2ZHUrBBAOxlFetB32GS0
中东（巴林）	me-south-1	ec2-serial-console.me-south-1.api.aws	SHA256:nPjLLKHu2QnLdUq2kVArsoK5xvPJOMRJKCBzCDqC3k8
中东（阿联酋）	me-central-1	ec2-serial-console.me-central-1.api.aws	SHA256:zpb5duKiBZ+l0dFwPeyykB4MPBYhl/XzXNeFSDKBvLE
南美洲（圣保罗）	sa-east-1	serial-console.ec2-instance-connect.sa-east-1.aws	SHA256:rd2+/32Ognjew1yVIemENaQzC+Botbih62OqAPDq1dI
Amazon GovCloud（美国东部）	us-gov-east-1	serial-console.ec2-instance-connect.us-gov-east-1.amazonaws.com	SHA256:tIwe19GWsoyLClrtvu38YEEh+DHIkqnDcZnmtebvF28
Amazon GovCloud（美国西部）	us-gov-west-1	serial-console.ec2-instance-connect.us-gov-west-1.amazonaws.com	SHA256:kfOFRWLaOZfB+utbd3bRf8OlPf8nGO2YZLqXZiIw5DQ

断开与 EC2 Serial Console 的连接

如果您不再需要连接到实例的 EC2 Serial Console，则可以断开连接。当您断开与串行控制台的连接时，在该实例上运行的所有 shell 会话都将继续运行。如果要结束 shell 会话，则需要先结束会话，然后再断开与串行控制台的连接。

注意事项

- 串行控制台连接通常会持续一小时，除非您断开连接。但是，在系统维护期间，Amazon EC2 将断开串行控制台会话。
- 在断开与串行控制台的连接后，需要 30 秒钟来停用会话，然后才能允许新会话。

从串行控制台断开连接的方法取决于客户端。

基于浏览器的客户端

要断开串行控制台，请关闭串行控制台浏览器内终端窗口。

标准 OpenSSH 客户端

要断开串行控制台，请使用以下命令关闭 SSH 连接。此命令必须在换行之后立即运行。

```
~.
```

您用于关闭 SSH 连接的命令可能会有所不同，取决于您正在使用的 SSH 客户端。

使用 EC2 Serial Console 对 Amazon EC2 实例进行故障排查

通过使用 EC2 Serial Console，您可以通过连接到实例的串行端口来排查启动、网络配置和其他方面的问题。

请使用适用于实例操作系统和在实例上配置的工具的说明。

工具

- （Linux 实例）使用 GRUB 对您的实例进行故障排除
- （Linux 实例）使用 SysRq 对您的实例进行故障排除
- （Windows 实例）使用 SAC 对您的实例进行故障排除

先决条件

开始之前，请确保您已完成先决条件，包括配置您选择的故障排除工具。

（Linux 实例）使用 GRUB 对您的实例进行故障排除

GNU GRUB（GNU GRand Unified Bootloader [统一引导加载程序] 的缩写，通常称为 GRUB）是大多数 Linux 操作系统的默认引导加载程序。从 GRUB 菜单中，您可以选择要引导到哪个内核，或修改菜单项以更改内核的启动方式。这在对失败的实例进行故障排查时非常有用。

在引导流程中将显示 GRUB 菜单。该菜单无法通过普通 SSH 访问，但您可以通过 EC2 Serial Console 访问。

启动到单用户模式或紧急模式。单用户模式将在较低的运行级别启动内核。例如，它可能会挂载文件系统，但不会激活网络，从而使您有机会执行修复实例所需的维护。紧急模式与单用户模式类似，只是内核在可达到的最低运行级别运行。

启动到单用户模式

1. 连接到实例的串行控制台。
2. 使用以下命令重新引导实例。

   ```
   [ec2-user ~]$ sudo reboot
   ```

3. 在重新启动期间，当 GRUB 菜单出现时，按任意键停止引导流程。
4. 在 GRUB 菜单中，使用箭头键选择要引导的内核，然后按键盘上的 e。
5. 使用箭头键将光标定位在包含内核的行上。该行以 `linux` 或 `linux16` 开头，取决于启动实例所使用的 AMI。对于 Ubuntu，有两行开头为 `linux`，必须在下一步中修改。
6. 在行末添加 `single` 一词。

   以下是 Amazon Linux 2 的示例。

   ```
   linux /boot/vmlinuz-4.14.193-149.317.amzn2.aarch64 root=UUID=d33f9c9a-\
   dadd-4499-938d-ebbf42c3e499 ro  console=tty0 console=ttyS0,115200n8 net.ifname\
   s=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.she\
   ll=0 single
   ```

7. 按 Ctrl+X 可启动到单用户模式。
8. 在 `login` 提示符下，输入您之前设置的、基于密码的用户的用户名，然后按 Enter 键。
9. 在 `Password` 提示符下，输入密码，然后按 Enter 键。

启动到紧急模式

请按照与单用户模式相同的步骤操作，但在第 6 步中添加单词 `emergency` 而不是 `single`。

（Linux 实例）使用 SysRq 对您的实例进行故障排除

系统请求 (SysRq) 键（有时称为“magic SysRq”）可用于在 Shell 外直接向内核发送命令，无论内核正在执行什么操作，内核都会响应。例如，如果实例已停止响应，则可以使用 SysRq 密钥告诉内核崩溃或重新启动。有关更多信息，请参阅 Wikipedia 中的 Magic SysRq 键。

您可以在基于 EC2 Serial Console 浏览器的客户端中或 SSH 客户端中使用 SysRq 命令。每个客户端发送中断请求的命令都不同。

要使用 SysRq，请根据您正在使用的客户端选择以下程序之一。

Browser-based client

在串行控制台基于浏览器的客户端中使用 SysRq

1. 连接到实例的串行控制台。
2. 要发送中断请求，请按 `CTRL+0`（零）。如果您的键盘支持，还可以使用暂停或中断键发送中断请求。

```
[ec2-user ~]$ CTRL+0
```

3. 要发出 SysRq 命令，请按键盘上与所需命令对应的键。例如，要显示 SysRq 命令列表，请按 h。

```
[ec2-user ~]$ h
```

h 命令的输出类似于以下内容。

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

SSH client

在 SSH 客户端中使用 SysRq

1. 连接到实例的串行控制台。
2. 要发送中断请求，请按 ~B（波浪号后跟大写 B）。

```
[ec2-user ~]$ ~B
```

3. 要发出 SysRq 命令，请按键盘上与所需命令对应的键。例如，要显示 SysRq 命令列表，请按 h。

```
[ec2-user ~]$ h
```

h 命令的输出类似于以下内容。

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

> **Note**
>
> 您用于发送中断请求的命令可能会有所不同，取决于您正在使用的 SSH 客户端。

（Windows 实例）使用 SAC 对您的实例进行故障排除

Windows 的特殊管理控制台 (SAC) 功能可用于对 Windows 实例进行故障排查。通过连接到实例的串行控制台并使用 SAC，您可以中断引导流程并在安全模式下启动 Windows。

> **Note**
>
> 如果您在实例上启用 SAC，则依赖密码检索的 EC2 服务将无法通过 Amazon EC2 控制台运行。Amazon EC2 启动代理（EC2Config、EC2Launch v1 和 EC2Launch v2）上的 Windows 依靠串行控制台来执行各种任务。当您在实例上启用 SAC 时，这些任务不会成功执行。有关 Amazon EC2 启动代理上的 Windows 的更多信息，请参阅 the section called “配置 Windows

实例”。如果您启用 SAC，则可在之后将其禁用。有关更多信息，请参阅 禁用 SAC 和启动菜单。

任务

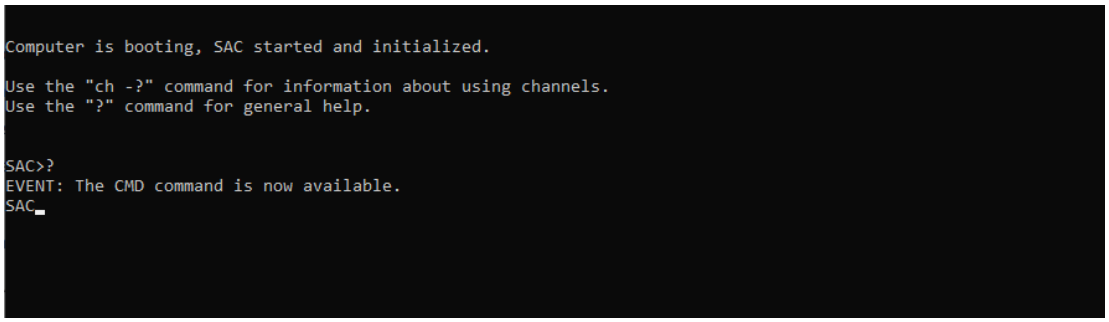
- 使用 SAC
- 使用启动菜单
- 禁用 SAC 和启动菜单

使用 SAC

使用 SAC

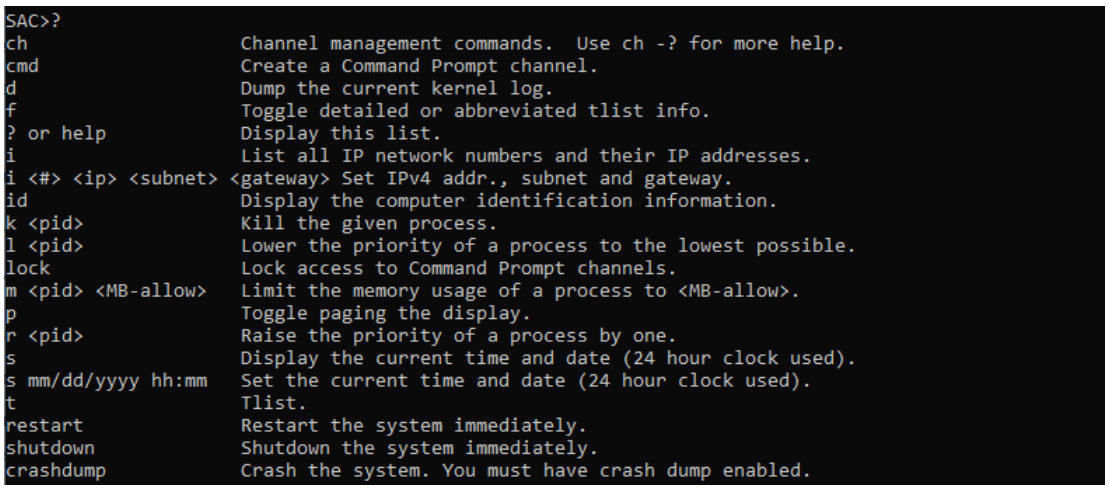
1. 连接到串行控制台。

   如果在实例上启用了 SAC，串行控制台将显示 SAC> 提示。



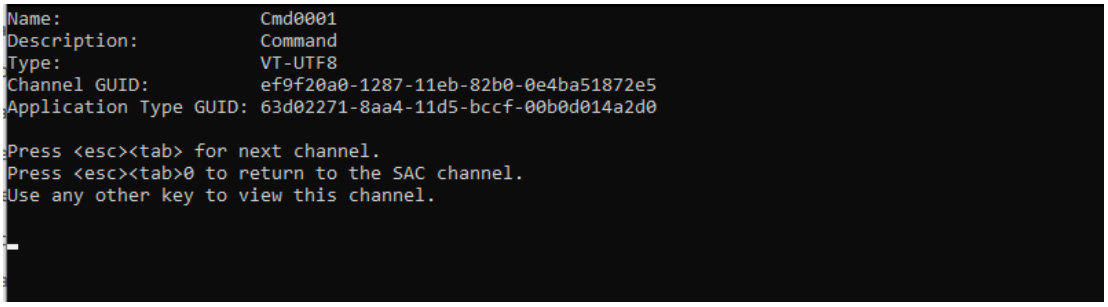
2. 要显示 SAC 命令，请输入 ?，然后按 Enter。

   预期输出



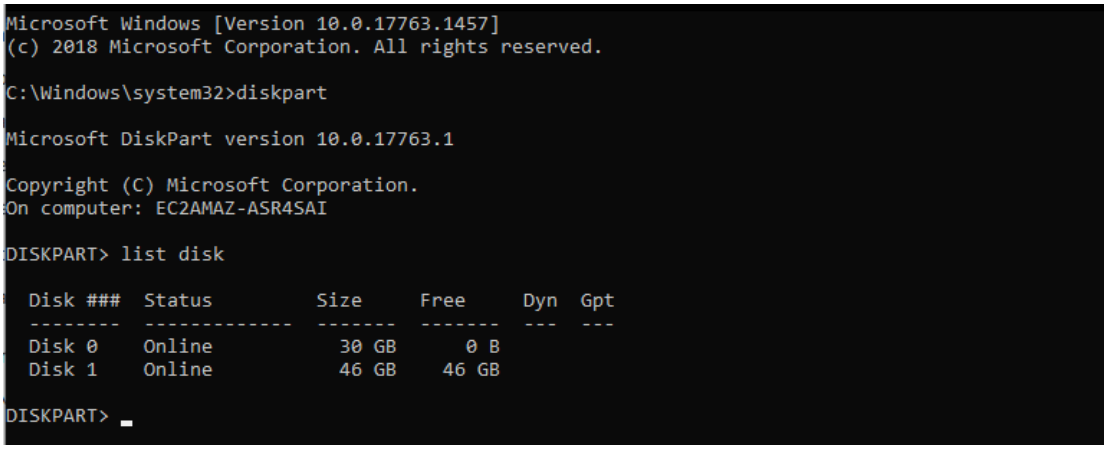
3. 要创建命令提示符通道（例如 cmd0001 或 cmd0002），请输入 cmd，然后按 Enter。
4. 要查看命令提示符通道，请按 ESC，然后按 TAB。

   预期输出



5. 要切换通道，请同时按 ESC+TAB+通道编号。例如，要切换至 cmd0002 通道（如果已创建），请按 ESC+TAB+2。
6. 输入命令提示符通道所需的凭证。

   命令提示符与您在桌面上获得的功能全面的命令 Shell 相同，例外之处是它不允许读取已输出的字符。



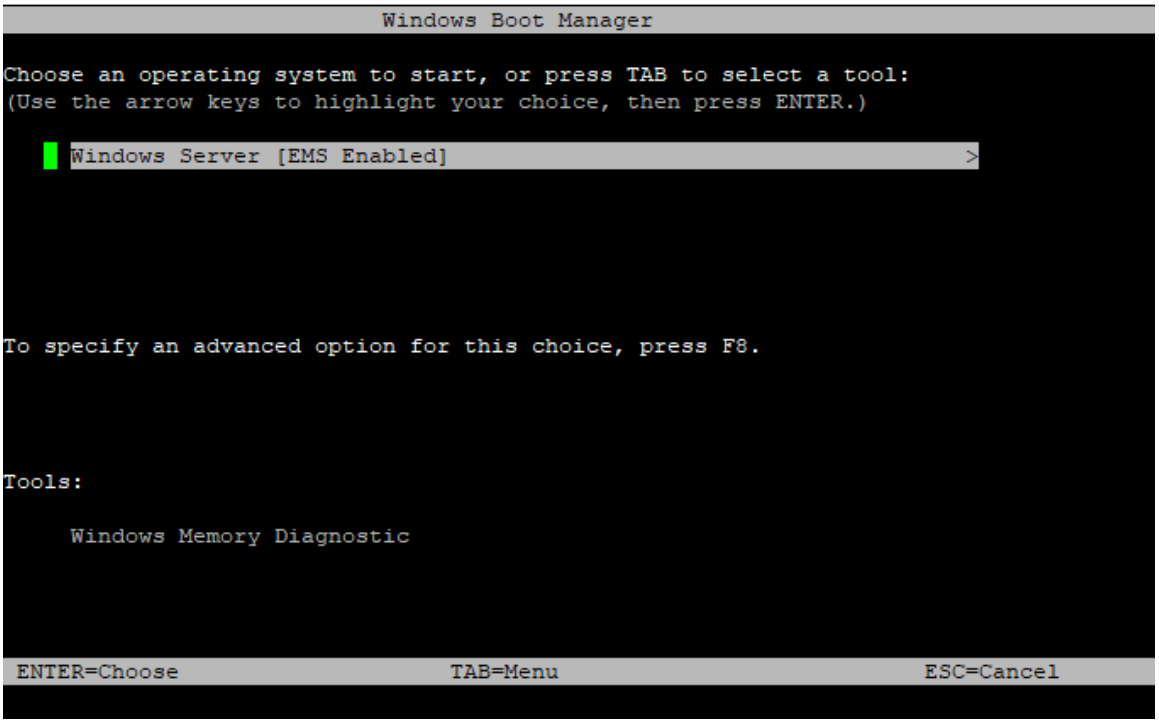
还可以从命令提示符使用 PowerShell。

请注意，您可能需要将进度首选项设置为静默模式。

```
PS C:\Windows\system32> $ProgressPreference="SilentlyContinue"
PS C:\Windows\system32> $computerInfo = Get-ComputerInfo
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Name
Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Description
Intel64 Family 6 Model 85 Stepping 4
PS C:\Windows\system32>
```

使用启动菜单

如果实例已启用了启动菜单并在通过 SSH 连接后重新启动，则您应看到启动菜单，如下所示。



启动菜单命令

ENTER

启动操作系统的选定条目。

TAB

切换到 Tools（工具）菜单。

ESC

取消并重新启动实例。

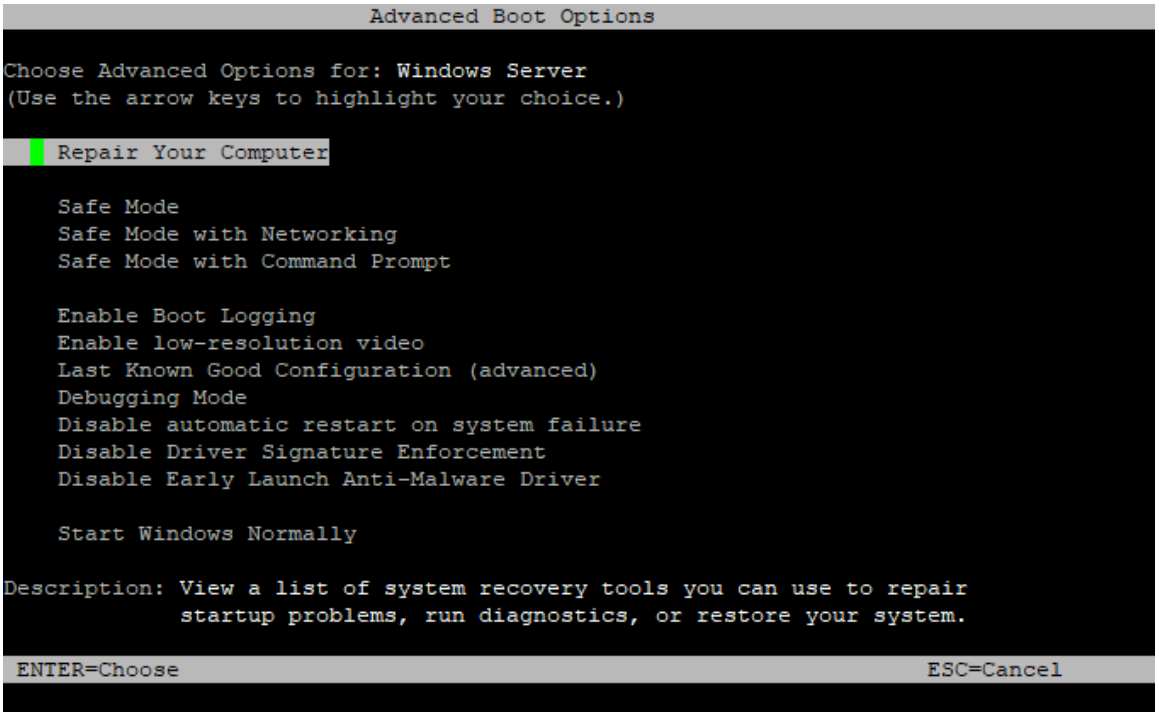
按 ESC 后再按 8

相当于按下 F8。显示所选项目的高级选项。

ESC 键 + 向左箭头

返回到初始启动菜单。

单靠 ESC 密钥不会让您返回主菜单，因为 Windows 正在等待查看转义序列是否正在进行中。



禁用 SAC 和启动菜单

如果您启用 SAC 和启动菜单，则可在之后将这些功能禁用。

使用以下方法之一在实例上禁用 SAC 和启动菜单。

PowerShell

在 Windows 实例上禁用 SAC 和启动菜单

1. 连接到您的实例并从提升的 PowerShell 命令行执行以下步骤。
2. 首先，将值更改为 no 以禁用启动菜单。

```
bcdedit /set '{bootmgr}' displaybootmenu no
```

3. 然后，将值更改为 `off` 以禁用 SAC。

```
bcdedit /ems '{current}' off
```

4. 重新启动实例以应用更新后的配置。

```
shutdown -r -t 0
```

Command prompt

在 Windows 实例上禁用 SAC 和启动菜单

1. 连接到您的实例并从命令提示符执行以下步骤。
2. 首先，将值更改为 no 以禁用启动菜单。

```
bcdedit /set {bootmgr} displaybootmenu no
```

3. 然后，将值更改为 off 以禁用 SAC。

```
bcdedit /ems {current} off
```

4. 重新启动实例以应用更新后的配置。

```
shutdown -r -t 0
```

发送诊断中断来调试无法访问的 Amazon EC2 实例

> **Warning**
> 诊断中断供高级用户使用。使用不当会对您的实例产生负面影响。向实例发送诊断中断可能会触发实例崩溃和重新启动，从而导致数据丢失。

您可以向无法访问或无响应的实例发送诊断中断，以手动触发 Linux 实例的内核错误或 Windows 实例的停止错误（通常称为蓝屏错误）。

Linux 实例

在出现内核错误时，Linux 操作系统通常会发生崩溃并重启。操作系统的具体行为取决于其配置。内核错误也可用于使实例的操作系统内核执行任务，例如生成崩溃转储文件。然后，您可以使用崩溃转储文件中的信息进行根本原因分析并调试实例。崩溃转储数据由操作系统在实例本身上本地生成。

Windows 实例

通常，Windows 操作系统在发生停止错误时发生崩溃并重启，但具体行为取决于其配置。停止错误还可能导致操作系统将调试信息（例如内核内存转储）写入文件。然后，您可以使用此信息进行根本原因分析以调试实例。内存转储数据由操作系统在实例本身上本地生成。

在向您的实例发送诊断中断之前，我们建议您查阅适合您操作系统的文档，然后进行必要的配置更改。

目录

- 支持的实例类型
- 先决条件
- 发送诊断中断

支持的实例类型

除 Amazon Graviton 处理器支持的实例类型外的所有基于 Nitro 的实例类型均支持诊断中断。有关更多信息，请参阅基于 Amazon Nitro System 构建的实例和 Amazon Graviton。

先决条件

在使用诊断中断之前，必须配置实例的操作系统。这可确保在发生内核错误（Linux 实例）或停止错误（Windows 实例）时执行所需的操作。

Linux 实例

将 Amazon Linux 2 或 Amazon Linux 2023 配置为在发生内核错误时生成崩溃转储

1. 连接到您的 实例。
2. 安装 kexec 和 kdump。

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. 配置内核以便为辅助内核预留适当的内存量。要预留的内存量取决于实例的总可用内存。使用首选文本编辑器打开 `/etc/default/grub` 文件，找到以 `GRUB_CMDLINE_LINUX_DEFAULT` 开始的

行，然后按以下格式添加 crashkernel 参数：crashkernel=*memory_to_reserve*。例如，要预留 256MB，请修改 grub 文件，如下所示：

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. 保存更改并关闭 grub 文件。
5. 重新构建 GRUB2 配置文件。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. 在基于 Intel 和 AMD 处理器的实例上，send-diagnostic-interrupt 命令将未知的非屏蔽中断 (NMI) 发送到实例。您必须将内核配置为在收到未知 NMI 时发生崩溃。使用首选文本编辑器打开 /etc/sysctl.conf 文件，并添加以下内容。

```
kernel.unknown_nmi_panic=1
```

7. 重启实例并重新连接到它。
8. 使用正确的 crashkernel 参数验证是否已启动内核。

```
$ grep crashkernel /proc/cmdline
```

以下示例输出指示成功的配置。

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0
```

9. 确认 kdump 服务正在运行。

```
[ec2-user ~]$ systemctl status kdump.service
```

以下示例输出显示在 kdump 服务正在运行的情况下的结果。

```
kdump.service - Crash recovery kernel arming
```

```
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
enabled)
   Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
  Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
Main PID: 2503 (code=exited, status=0/SUCCESS)
```

> **Note**
>
> 默认情况下，崩溃转储文件将保存到 /var/crash/。要更改位置，请使用首选文本编辑器修改 /etc/kdump.conf 文件。

配置 SUSE Linux Enterprise、Ubuntu 或 Red Hat Enterprise Linux

在基于 Intel 和 AMD 处理器的实例上，send-diagnostic-interrupt 命令将未知的非屏蔽中断 (NMI) 发送到实例。必须通过调整操作系统的配置文件，将内核配置为在收到未知 NMI 时发生崩溃。有关如何将内核配置为发生崩溃的信息，请参阅适用于操作系统的文档：

- SUSE Linux Enterprise
- Ubuntu
- Red Hat Enterprise Linux (RHEL)

Windows 实例

配置 Windows 以在发生停止错误时生成内存转储

1. 连接到您的 实例。
2. 打开控制面板，然后选择系统、高级系统设置。
3. 在系统属性对话框中，选择高级选项卡。
4. 在启动和恢复部分中，选择设置...。
5. 在系统故障部分中，根据需要配置设置，然后选择确定。

有关配置 Windows 停止错误的更多信息，请参阅 Windows 的内存转储文件概述。

发送诊断中断

在完成必要的配置更改后，您可以使用 Amazon CLI 或 Amazon EC2 API 将诊断中断发送到实例。

Amazon CLI

将诊断中断发送到实例

使用 send-diagnostic-interrupt 命令。

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

PowerShell

将诊断中断发送到实例

使用 Send-EC2DiagnosticInterrupt cmdlet。

```
Send-EC2DiagnosticInterrupt -InstanceId i-1234567890abcdef0
```

《Amazon EC2 用户指南》的文档历史记录

下表描述《Amazon EC2 用户指南》的重要补充部分（2019 年开始）。我们还经常更新该指南来处理发送给我们的反馈意见。

变更	说明	日期
IMDSv2 强制使用	可以在账户级别为每个 Amazon Web Services 区域强制要求使用 IMDSv2。强制使用后，实例只有配置为必须使用 IMDSv2 时才能启动。	2026 年 2 月 24 日
嵌套虚拟化	您可以在虚拟化的 Amazon EC2 实例内运行虚拟机管理程序，例如 Hyper-V 和 KVM。	2026 年 2 月 16 日
P6-B300 实例的实例拓扑结构	您可以获取 `p6-b300.48xlarge` 实例的实例拓扑结构信息。	2025 年 11 月 18 日
AMI 世系	现在，您可以通过获取所有原级 AMI 的列表来跟踪 AMI 的来源。	2025 年 11 月 12 日
容量预留拓扑	在启动实例之前，您可以使用 DescribeCapacityReservationTopology API 来描述您的预留容量在网络层次结构中相对于彼此的位置。	2025 年 10 月 28 日
优化 CPU 以节省许可证费用	现在，您可以配置活动 CPU，以节省包含许可证的 Windows 和 SQL Server 实例的许可费用。	2025 年 10 月 15 日

AWSEC2CapacityManagerServiceRolePolicy：新托管策略	Amazon EC2 新增了 AWSEC2CapacityManagerServiceRolePolicy 托管策略，以便可以代表您管理容量资源并与 Amazon Organizations 集成。	2025 年 10 月 15 日
GPUPowerUtilization 指标	加速型计算实例的新 GPUPowerUtilization 指标。	2025 年 10 月 13 日
EC2 Instance Connect 端点 IPv6 支持	EC2 Instance Connect 端点现已支持 IPv6 连接。现在，您可以将 EC2 Instance Connect 端点配置为支持双堆栈或仅 IPv6，以连接到具有 IPv6 地址的实例。	2025 年 10 月 2 日
允许的 AMI：新参数	允许的 AMI 是一项允许您限制在自己 Amazon Web Services 账户内发现和使用 AMI 的功能，现在支持四个新参数：Amazon Web Services Marketplace 产品代码、AMI 名称、AMI 弃用后的最长期限以及 AMI 的最长使用期限。	2025 年 9 月 25 日
实例存储卷的详细性能统计数据	Amazon EC2 为连接到基于 Nitro 的实例的 NVMe 实例存储卷提供实时、高分辨率的性能统计数据。	2025 年 9 月 16 日
Amazon EC2 Mac 专属主机更新	Amazon EC2 Mac 专属主机支持专属主机自动恢复和基于重启的主机维护。	2025 年 8 月 28 日

Local Zones 的 AMI 副本	您现在可以将 Amazon EBS-backed AMI 复制到 Local Zones 或从 Local Zones 复制 Amazon EBS-backed AMI。	2025 年 8 月 28 日
AmazonEC2ImageReferencesAccessPolicy – 新的托管策略	Amazon EC2 添加了 AmazonEC2ImageReferencesAccessPolicy 托管策略，以提供扫描 EC2 DescribeImageReferences API 支持的所有资源类型的权限。	2025 年 8 月 26 日
AMI 使用情况报告	如果与其他 Amazon Web Services 账户共享亚马逊机器映像（AMI）（无论是与特定 Amazon Web Services 账户、组织、组织单元（OU）还是公开），可以通过创建 AMI 使用情况报告来了解这些 AMI 的使用情况。通过报告，您可以了解哪些 Amazon Web Services 账户正在 EC2 实例或启动模板中使用 AMI，以及有多少个实例或启动模板正在引用每个 AMI。	2025 年 8 月 26 日
AMI 引用	您可以识别引用指定亚马逊机器映像（AMI）的 Amazon 资源，无论 AMI 是公共的还是私有的，也无论它们的所有者是谁。这种可见性可帮助您确保资源使用最新的合规 AMI。	2025 年 8 月 26 日
修改 EC2 Instance Connect Endpoint	您现在可使用 Amazon CLI 或 SDK 修改现有 EC2 Instance Connect Endpoint。	2025 年 8 月 14 日

容量预留共享增强功	您现在可以共享在集群置放群组中创建的容量预留。	2025 年 8 月 12 日
EC2 Instance Connect Endpoint 的更新托管式策略	为了支持修改现有 EC2 Instance Connect Endpoint，Amazon EC2 更新了 `Ec2InstanceConnectEndpoint` 策略，以添加分配和取消分配 IPv6 地址的权限，并修改 EC2 实例连接端点创建的网络接口上的安全组。Amazon EC2 还更新了此策略，用 `StringLike` 条件运算符替换了 `Null` 条件运算符。	2025 年 7 月 31 日
实例拓扑 - 增加的区域支持	DescribeInstanceTopology API 现已在亚太地区（墨尔本）推出。	2025 年 7 月 31 日
强制终止	现在，您可以强制终止实例。	2025 年 7 月 31 日
实例拓扑 - 增加的区域支持	现已在 Amazon GovCloud（美国西部）中提供 DescribeInstanceTopology API。	2025 年 7 月 24 日
在实例终止期间跳过操作系统关闭	终止实例时，您可以跳过操作系统 (OS) 的正常关闭。	2025 年 7 月 23 日
在实例停止期间跳过操作系统关闭	停止实例时，您可以跳过操作系统 (OS) 的正常关闭。	2025 年 7 月 23 日
us-east-1 中的 P6-B200 容量块	现已在美国东部（弗吉尼亚州北部）- `us-east-1` 中提供 P6-B200 容量块。	2025 年 7 月 23 日

更新了 EC2 Fast Launch 服务相关角色的托管式策略	为了防止出现孤立资源，Amazon EC2 对此策略进行了更新，增加了描述卷、卷属性和网络接口的权限，并增加了删除 EC2 Fast Launch 创建的卷和网络接口的权限。	2025 年 7 月 17 日
EC2 Serial Console - 增加的区域支持	目前在亚太地区（泰国）和墨西哥（中部）也提供了 EC2 Serial Console。	2025 年 7 月 15 日
EC2 Instance Connect - 增加的区域支持	目前在亚太地区（泰国）和墨西哥（中部）也提供了 EC2 Instance Connect。	2025 年 7 月 15 日
硬件数据包时间戳	您可以在 EC2 实例上启用硬件数据包时间戳，为每个传入的网络数据包添加 64 位纳秒精度的时间戳。	2025 年 6 月 23 日
在 AMI 注销过程中删除快照	注销 AMI 时，您可以同时自动删除其关联的快照。	2025 年 5 月 28 日
用户发起的重启迁移	如果实例收到 EC2 计划重启事件通知，则系统会在您于该事件到来前重启实例时，默认启用实例迁移。	2025 年 5 月 20 日
Amazon EC2 Mac 实例的 SIP 配置	您可以为采用 x86 架构的 Mac 实例和搭载 Apple 芯片的 Mac 实例配置系统完整性保护（SIP）设置。	2025 年 5 月 20 日
根卷所有权委托	您现在可以在搭载 Apple 芯片的 Mac 实例上更新操作系统时，自动委托 Amazon EBS 根卷的根卷所有权。	2025 年 5 月 20 日

P6-B200 实例的实例拓扑结构	您可以获取 p6-b200.48xlarge 实例的实例拓扑结构信息。	2025 年 5 月 15 日
更新了 EC2 Fast Launch 服务相关角色的托管式策略	更新了 EC2 Fast Launch 服务相关角色策略，包含了创建和维护 Amazon EventBridge 事件规则以及其他操作的访问权限，以增强 EC2 Fast Launch 体验。有关服务相关角色所授予权限的更多信息，请参阅 EC2FastLaunchServiceRolePolicy 授予的权限。	2025 年 5 月 14 日
更新了 EC2 Fast Launch 完全访问策略的托管式策略	更新了 EC2 Fast Launch 完全访问策略，包含了启动模板创建和修改操作的权限，以及创建网络和安全组资源、授权入口规则以及删除 EC2 Fast Launch 所创建资源的权限。此外，此策略还可以描述和创建 CloudFormation 堆栈，以及删除 EC2 Fast Launch 创建的堆栈。	2025 年 5 月 14 日
简化了 EC2 Fast Launch 的启用过程	更新了 EC2 Fast Launch 以简化启用过程。	2025 年 5 月 14 日
Amazon Config 与允许的 AMI 集成	使用 ec2-instance-launched-with-allowed-ami Amazon Config 规则来检查正在运行或已停止的实例是否使用符合“允许的 AMI”标准的 AMI 启动。	2025 年 3 月 11 日

更新了 AWSEC2CapacityReservationFleetRolePolicy 托管式策略	Amazon EC2 更新了 `AWSEC2CapacityReservationFleetRolePolicy` 托管式策略，使用 `ArnLike` 条件运算符，而不是 `StringLike` 条件运算符。	2025 年 3 月 3 日
基于时间的 AMI 复制	现在，您可以请求为 EBS 支持的 AMI 复制操作设置完成期限，以确保在特定的时间期限内完成 AMI 复制。	2025 年 2 月 25 日
更新了 AmazonEC2ReadOnlyAccess 策略	Amazon EC2 将 `GetSecurityGroupsForVpc` 操作添加到现有的 AmazonEC2ReadOnlyAccess 策略中。	2024 年 12 月 27 日
托管式实例	现在，您可以在 EC2 控制台中查看 Amazon EC2 托管实例。	2024 年 12 月 1 日
声明式策略	现在，您可以使用声明式策略将账户级别设置同时应用于多个区域和账户。支持使用声明式策略配置 EC2 Serial Console 访问权限、允许的 AMI、IMDS 默认值以及 VPC、AMI 和快照的屏蔽公共访问权限设置。有关更多信息，请参阅《Amazon Organizations User Guide》中的 Declarative policies 以及每项支持功能的具体文档。	2024 年 12 月 1 日
允许的 AMI	现在，您可以通过指定 AMI 必须满足的标准来控制 Amazon EC2 中对 AMI 的发现和使用。	2024 年 12 月 1 日

即时容量块	您现在可以为 ML 预留容量块，可在 30 分钟内开始。	2024 年 11 月 21 日
未来日期的容量预留	您现在可以请求未来某个日期的容量预留。	2024 年 11 月 21 日
延长容量块期限	现在，您可以延长现有容量块的持续时间。	2024 年 11 月 21 日
6 个月容量块	您现在可以为 ML 预留容量块长达 6 个月（182 天）。	2024 年 11 月 21 日
已保存的筛选条件集	您现在可以创建自定义的筛选条件组，并重复使用它们来有效地查看 EC2 资源。	2024 年 11 月 20 日
性能保护	当您对 EC2 实例集或竞价型实例集使用基于属性的实例类型选择时，现在可以启用性能保护，以确保所选实例类型类似于或超过指定的性能基准。	2024 年 11 月 20 日
仅容量预留	您现在可以指定实例仅在容量预留或容量预留资源组中运行。	2024 年 11 月 20 日
识别源 AMI	您现在可以识别用于创建 AMI 的源 AMI。	2024 年 11 月 13 日
拆分容量	您可以从现有容量预留中拆分容量来创建新的预留。	2024 年 10 月 30 日
移动容量	您现在可以将容量从一个容量预留移动到另一个容量预留。	2024 年 10 月 30 日
初学者教程	新推出两个面向初学者的教程：“启动我的第一个 EC2 实例”和“启动测试 EC2 实例并连接到实例”。	2024 年 10 月 21 日

Windows Server 2025 支持	添加了对 Windows Server 2025 的支持。	2024 年 10 月 16 日
EC2 全局视图	现在，您可以通过 EC2 全局视图来查看账户中所有区域的容量预留和容量块。	2024 年 10 月 16 日
实时迁移主机维护	Amazon EC2 专属主机现在支持实时迁移主机维护。此功能会自动将受支持的实例从受损的专属主机迁移到替代专属主机，而无需停止和重新启动实例。	2024 年 10 月 15 日
共享容量预留的账单分配	现在，您可以将共享容量预留中任何可用容量的账单分配给属于同一 Amazon 组织的使用者账户。	2024 年 10 月 14 日
PTP 硬件时钟 – 新增区域支持	PTP 硬件时钟现已在美国东部（俄亥俄州）和亚太地区（马来西亚）区域推出。	2024 年 9 月 23 日
EC2 Instance Connect 支持 IPv6	您现在可以使用 EC2 Instance Connect 连接到实例的gong'you公有 IPv6 地址。	2024 年 9 月 23 日
EC2 Instance Connect 前缀列表	您现在可以在安全组中创建规则时选择 IPv4 或 IPv6 地址的托管前缀列表，从而允许来自 EC2 Instance Connect 服务的 SSH 流量。	2024 年 9 月 23 日
管理按需容量预留的新功能	现在，您可以拆分容量预留，在容量预留之间移动容量，以及修改容量预留的实例资格属性。	2024 年 8 月 14 日

适用于 C6g、C6gn、C6gd、C7g、C7gd、M6g、M6gd、M7g、M7gd、R6g 和 R6gd 的休眠支持	休眠在 C6g、C6gn、C6gd、C7g、C7gd、M6g、M6gd、M7g、M7gd、R6g 和 R6gd 实例类型上运行的新启动实例。	2024 年 7 月 30 日
为支持 Graviton 实例类型的 AMI 提供休眠支持	休眠从支持 Graviton 实例类型的 Amazon Linux 或 Ubuntu AMI 启动的新启动实例。	2024 年 7 月 30 日
凭证保护支持的其他实例类型	您现在可以为 C7i、C7-flex、M7i、M7i-flex、R7i、R7i-flex 和 T3 实例启用凭证保护。	2024 年 6 月 26 日
EC2 实例类型查找器 - 其他参数	EC2 实例类型查找器现在提供了其他参数供您为工作负载指定更详细的要求。	2024 年 6 月 5 日
适用于 EC2 Fast Launch 的新托管策略	添加了 EC2FastLaunchFullAccess 策略，以便在实例中执行与 EC2 Fast Launch 功能相关的 API 操作。	2024 年 5 月 14 日
AMI 注销保护	您可以在 AMI 上开启注销保护，以防止意外或恶意删除。	2024 年 4 月 23 日
PTP 硬件时钟 – 实例类型支持	PTP 硬件时钟现已在 M7a、M7g、M7i、R7a 和 R7i 实例类型上推出。	2024 年 4 月 22 日
添加了用于增强联网的 Nitro 性能注意事项	本页重点介绍网络注意事项，以帮助调整基于 Nitro 的 Amazon EC2 实例的性能。	2024 年 4 月 4 日

基于 VSS 的 EBS 快照的新托管策略	Amazon EC2 VSS 提供了一种新的 IAM 托管策略，您可以将其添加到您的实例配置文件角色中，以确保您的权限保持最新并遵循最佳实践。	2024 年 3 月 28 日
PTP 硬件时钟 – 美国东部（弗吉尼亚州北部）	PTP 硬件时钟现已在美国东部（弗吉尼亚州北部）区域推出。	2024 年 3 月 26 日
将 IMDSv2 设置为账户默认设置	可将账户中启动的所有新 EC2 实例设置为默认使用实例元数据服务版本 2（IMDSv2）。	2024 年 3 月 25 日
标记从快照创建的新的 Linux AMI	从快照创建 Linux AMI 时，您可以标记新的 AMI。	2024 年 3 月 7 日
复制时标记新的 AMI 和快照	复制 AMI 时，您可以使用相同的标签来标记新的 AMI 和新的快照，也可以使用不同的标签来标记它们。	2024 年 3 月 7 日
移除 Amazon Management Pack 页面	Amazon Management Pack 主要搭配 Windows Server 2012 及更早版本使用。旧版操作系统平台版本已不再受支持。要管理在 Amazon 上和本地运行的服务器实例集并对其进行故障排除，请参阅 Amazon Systems Manager Fleet Manager。	2024 年 2 月 12 日
预装在 macOS AMI 上的 EC2 Instance Connect	EC2 Instance Connect 现在已预装在 macOS Sonoma 14.2.1 或更高版本、macOS Ventura 13.6.3 或更高版本，以及 macOS Monterey 12.7.2 或更高版本的 AMI 上。	2024 年 1 月 26 日

CentOS、macOS 和 RHEL 上的 EC2 Instance Connect 支持	现在可在受支持的 CentOS、macOS 和 RHEL AMI 上安装 EC2 Instance Connect。	2023 年 12 月 6 日
对 C7a、C7i、R7a、R7i 和 R7iz 的休眠支持	休眠新启动并在 C7a、C7i、R7a、R7i 和 R7iz 实例类型上运行的实例。	2023 年 12 月 1 日
Amazon Q EC2 实例类型选择器	Amazon Q EC2 实例类型选择器会考虑您的使用案例、工作负载类型和 CPU 制造商偏好，以及您优先考虑价格还是性能。随后，它利用这些数据提供有关最适合新工作负载的 Amazon EC2 实例类型的指导和建议。	2023 年 11 月 28 日
Console-to-Code	Console-to-Code 可以帮助您开始使用自动化代码。Console-to-Code 会记录您的控制台操作，然后利用生成式人工智能，以您首选的基础设施即代码格式提供代码建议。您可以将此代码用作一个起点，并对其进行自定义，以使它可用于您的特定使用案例的生产。	2023 年 11 月 26 日
可配置的空闲连接跟踪超时	保持空闲状态的安全组连接可能导致连接跟踪耗尽，并导致无法跟踪连接和丢弃数据包。现在，您可以在弹性网络接口上设置安全组连接跟踪的超时时间（以秒为单位）。	2023 年 11 月 17 日

PTP 硬件时钟	支持的实例现在具有精确时间协议（PTP）硬件时钟。PTP 硬件时钟支持 NTP 连接或直接 PTP 连接。	2023 年 11 月 16 日
更改已启用休眠的实例的实例类型	您现在可以更改处于 stopped 状态时已启用休眠的实例的实例类型。	2023 年 11 月 16 日
实例拓扑	您可以使用 DescribeInstanceTopology API 来检测您实例的位置，然后使用此信息通过在物理上彼此更靠近的实例上运行 HPC 和 ML 作业来优化 HPC 和 ML 作业。	2023 年 11 月 13 日
EC2 Fast Launch 共享 AMI 支持	现在，您可以在与您共享的 AMI 上启用 EC2 Fast Launch。在共享 AMI 上启用 EC2 Fast Launch 时，将在您的账户中创建用于加快启动速度的预置快照。	2023 年 11 月 6 日
适用于 ML 的容量块	您现在可以在未来某个日期预留 GPU 实例，从而支持您的短期机器学习（ML）工作负载。	2023 年 10 月 31 日
竞价型实例休眠	现在，您可以使用当前可用于按需型实例的相同休眠体验和实例系列来让竞价型实例休眠。	2023 年 10 月 24 日
阻止公开访问 AMI 的默认设置	现在，所有新账户和没有公有 AMI 的现有账户将默认启用阻止公开访问 AMI。	2023 年 10 月 20 日

Amazon EC2 全局视图	Amazon EC2 全局视图支持其他资源类型和可自定义的显示选项。	2023 年 10 月 18 日
对 Ubuntu 22.04.2 LTS（Jammy Jellyfish）的休眠支持	休眠从 Ubuntu 22.04.2 LTS（Jammy Jellyfish）AMI 启动的新启动实例。	2023 年 10 月 16 日
禁用 AMI	您可以禁用 AMI，以防止系统将其用于实例启动。	2023 年 10 月 12 日
附加的 EBS 状态检查	您可以使用附加的 EBS 状态检查来监控附加到实例的 Amazon EBS 卷是否可以访问。	2023 年 10 月 11 日
Red Hat Enterprise Linux 9 休眠支持	休眠从 Red Hat Enterprise Linux 9 AMI 启动的新启动实例。	2023 年 10 月 2 日
Microsoft Windows Server 2022 休眠支持	休眠从 Microsoft Windows Server 2022 AMI 启动的新启动实例。	2023 年 10 月 2 日
对 AL2023 的休眠支持	休眠从 AL2023 AMI 启动的新启动实例。	2023 年 10 月 2 日
启动竞价型实例集中竞价型实例的中断	您可以在 Amazon EC2 控制台中选择一个竞价型实例集并启动实例集中竞价型实例的中断，从而测试竞价型实例上的应用程序将如何处理中断。	2023 年 9 月 21 日
阻止公开访问 AMI	您可以在账户级别启用阻止公开访问 AMI，以阻止任何公开您的 AMI 的尝试。	2023 年 9 月 12 日
对 M7i 和 M7i-flex 的休眠支持	使 M7i 和 M7i-flex 实例类型上运行的新启动实例休眠。	2023 年 8 月 22 日

EC2-Classic 已弃用	通过使用 EC2-Classic，EC2 实例会在一个可与其他客户共享的扁平化网络中运行。Amazon VPC 将替代 EC2-Classic。通过使用 Amazon VPC，您的实例会在一个逻辑上与 Amazon 账户分离的 Virtual Private Cloud（VPC）中运行。	2023 年 8 月 8 日
专属主机	您可以在 Outpost 上的特定硬件资产上分配专属主机。	2023 年 6 月 20 日
EC2 Instance Connect Endpoint	现在，您可以通过 SSH 或 RDP 连接到实例，而无需实例具有公有 IPv4 地址。	2023 年 6 月 13 日
IMDS Package Analyzer	现在，您可以使用 IMDS Packet Analyzer 在 EC2 实例上识别 IMDSv1 调用的源。	2023 年 6 月 1 日
EC2 Serial Console 裸机实例	EC2 Serial Console 现在支持连接到选定裸机实例的串行端口。	2023 年 4 月 11 日
启动模板配额	现在，您可以在服务限额控制台中使用服务限额 CLI 查看启动模板和启动模板版本的配额。	2023 年 4 月 3 日
容量预留利用率通知	现在，当您的账户中容量预留的容量利用率降至 20% 以下时，Amazon Health 会发送通知。	2023 年 4 月 3 日
容量预留组	现在，您可以将与您共享的容量预留添加到您拥有的容量预留组。	2023 年 3 月 30 日

修改实例元数据选项	现在，您可以使用 Amazon EC2 控制台修改实例元数据选项。	2023 年 3 月 20 日
就地 macOS 操作系统更新	现在，您可以在 M1 Mac 实例上执行就地 Apple macOS 操作系统更新。	2023 年 3 月 14 日
UEFI Preferred	现在，您可以创建同时支持统一可扩展固件接口（UEFI）和传统 BIOS 启动模式的单个 AMI。	2023 年 3 月 3 日
修改 IMDSv2 的 AMI	修改现有 AMI，以确保从此 AMI 启动的实例默认要求使用 IMDSv2。	2023 年 2 月 28 日
EC2 Instance Connect Endpoint 的新托管式策略	添加了 `Ec2InstanceConnectEndpoint` 策略，该策略附加到 AWSServiceRoleForEC2InstanceConnect 服务相关角色，以允许 Amazon EC2 在您创建 EC2 Instance Connect Endpoint 时代表您执行操作。	2023 年 1 月 24 日
启动模板中的 AMI 别名	您可以在启动模板中指定 Amazon Systems Manager 参数而不是 AMI ID，从而无需每次 AMI ID 发生更改时都更新模板。	2023 年 1 月 19 日
对 C6i、I3en 和 M6i 的休眠支持	使 C6i、I3en 和 M6i 实例类型上运行的新启动实例休眠。	2022 年 12 月 19 日

撕裂写防护	借助撕裂写防护这一数据块存储功能，可提高 I/O 密集型关系数据库工作负载的性能并减少延迟，而不会影响数据弹性。	2022 年 11 月 29 日
ENA Express	使用 ENA Express 提高吞吐量并最大限度减少 EC2 实例之间网络流量的尾部延迟。	2022 年 11 月 28 日
复制 AMI 标签	复制 AMI 时，可以同时复制用户定义的 AMI 标签。	2022 年 11 月 18 日
用于存储和还原的 AMI 大小	可在 Amazon S3 存储桶之间存储和还原的 AMI 大小（压缩前）现可高达 5000 GB。	2022 年 11 月 16 日
竞价型实例的 priceCapacityOptimized 分配策略	使用 `priceCapacityOptimized` 分配策略的竞价型实例集需同时考虑价格和容量，以选择中断可能性最小、价格尽可能最低的竞价型实例池。	2022 年 11 月 10 日
竞价型实例的 price-capacity-optimized 分配策略	使用 `price-capacity-optimized` 分配策略的 EC2 实例集需同时考虑价格和容量，以选择中断可能性最小、价格尽可能最低的竞价型实例池。	2022 年 11 月 10 日
取消与您的账户共享 AMI	如果已与您的 Amazon Web Services 账户 共享 AMI，而您不想再与您的账户共享该 AMI，则可以将您的账户从 AMI 的启动权限中移除。	2022 年 11 月 4 日
转移弹性 IP 地址	您现在可以将弹性 IP 地址从一个 Amazon Web Services 账户转移到另一个。	2022 年 10 月 31 日

替换根卷	您可以使用 AMI 替换正在运行的实例的根 Amazon EBS 卷。	2022 年 10 月 27 日
自动将实例连接到数据库	使用自动连接功能将一个或多个 EC2 实例快速连接到一个 RDS 数据库，以允许它们之间的流量。	2022 年 10 月 10 日
AMI 限额	限额现在适用于创建和共享 AMI。	2022 年 10 月 10 日
为 AMI 配置 IMDSv2	配置您的 AMI，以确保从此 AMI 启动的实例默认要求使用 IMDSv2。	2022 年 10 月 3 日
启动竞价型实例中断	您可以在 Amazon EC2 控制台中选择一个竞价型实例并启动中断，从而测试竞价型实例上的应用程序将如何处理中断。	2022 年 9 月 26 日
经过验证的 AMI 提供商	在 Amazon EC2 控制台中，将 Amazon 或经过验证的 Amazon 合作伙伴拥有的公有 AMI 标记为经过验证的提供商。	2022 年 7 月 22 日
Amazon Outposts 上的置放群组	增加了适用于 Outpost 上的置放群组的主机分布策略。	2022 年 6 月 30 日
Amazon Outposts 上的专属主机	您可以在 Amazon Outposts 上分配专属主机。	2022 年 5 月 31 日
实例停止保护	要防止实例意外停止，可以为实例启用停止保护。	2022 年 5 月 24 日

UEFI 安全引导	UEFI 安全引导构建在 Amazon EC2 长期以来的安全启动过程之上，并提供了额外的深度防御功能，可以帮助客户保护软件免受重新引导过程中持续存在的威胁。	2022 年 5 月 10 日
NitroTPM	Nitro 可信平台模块 (NitroTPM) 是一种虚拟设备，由 Amazon Nitro 系统提供，并符合 TPM 2.0 规范。	2022 年 5 月 10 日
AMI 状态更改事件	现在，当 AMI 更改状态时，Amazon EC2 会生成一个事件。您可以使用 Amazon EventBridge 检测并响应这些事件。	2022 年 5 月 9 日
描述公有密钥	您可以查询 Amazon EC2 密钥对的公有密钥和创建日期。	2022 年 4 月 28 日
创建密钥对	您可以在创建新密钥对时指定密钥格式（PEM 或 PPK）。	2022 年 4 月 28 日
在启动时挂载 Amazon FSx 文件系统	您可以使用新的启动实例向导在启动时挂载新的或现有的 Amazon FSx for NetApp ONTAP 或 Amazon FSx for OpenZFS 文件系统。	2022 年 4 月 12 日
新启动实例向导	Amazon EC2 控制台中全新改进后的启动体验，提供了更快、更简单的 EC2 实例启动方法。	2022 年 4 月 5 日
自动弃用公用 AMI	默认情况下，所有公用 AMI 的弃用日期设置为自 AMI 创建日期起的两年。	2022 年 3 月 31 日

实例元数据类别：autoscaling/target-lifecycle-state	使用 Auto Scaling 组时，您可以从实例元数据访问实例的目标生命周期状态。	2022 年 3 月 24 日
竞价放置分数	根据您的 Spot 容量要求，获取Amazon区域或可用区的建议。	2022 年 3 月 16 日
AMI 的上次启动时间	`lastLaunchedTime` 指示您上次使用 AMI 启动实例时的时间戳。	2022 年 2 月 28 日
ED25519 密钥	EC2 Instance Connect 和 EC2 Serial Console 现在支持 ED25519 密钥。	2022 年 1 月 20 日
竞价型实例集的基于属性的实例类型选择	指定实例必须具有的属性，Amazon EC2 将使用这些属性标识所有实例类型。	2022 年 1 月 11 日
EC2 机群的基于属性的实例类型选择	指定实例必须具有的属性，Amazon EC2 将使用这些属性标识所有实例类型。	2022 年 1 月 11 日
用于容量预留的其他 RHEL 平台	用于按需容量预留的其他 Red Hat Enterprise Linux 平台。	2022 年 1 月 11 日
配置 Windows AMI 以加快启动速度	请使用预配置的快照，将 Windows AMI 配置为将启动实例的速度提高 65%。	2022 年 1 月 10 日
实例元数据中的实例标签	您可以从实例元数据访问实例的标签。	2022 年 1 月 6 日
集群置放群组中的容量预留	您可以在集群置放群组中创建容量预留。	2022 年 1 月 6 日
竞价型实例集 launch-before-terminate	竞价型实例集可以在新的替换竞价型实例启动后终止接收再平衡通知的竞价型实例。	2021 年 11 月 4 日

EC2 机群 launch-before-terminate	EC2 机群可以在新的替换竞价型实例启动后终止接收再平衡通知的竞价型实例。	2021 年 11 月 4 日
比较时间戳	您可以通过将 Amazon EC2 Linux 实例的时间戳与 ClockBound 进行比较来确定事件的真实时间。	2021 年 11 月 2 日
与企业和 OU 共享 AMI	您现在可以与以下 Amazon 资源共享 AMI：企业和企业部门 (OU)。	2021 年 10 月 29 日
按需容量预留机群	您可以使用容量预留机群启动容量预留的组或机群。	2021 年 10 月 5 日
Ubuntu 20.04 LTS - Focal 的休眠支持	休眠从 Ubuntu 20.04 LTS - Focal AMI 启动的新推出的实例。	2021 年 10 月 4 日
专属主机上的 T3 实例	Amazon EC2 专属主机上的 T3 实例支持。	2021 年 9 月 14 日
对 RHEL、Fedora 和 CentOS 的休眠支持	使从 RHEL、Fedora 和 CentOS AMI 启动的新启动实例休眠。	2021 年 9 月 9 日
Amazon EC2 全局视图	通过 Amazon EC2 全局视图，您可以在单个控制台中查看跨多个 Amazon 区域的 VPC、子网、实例、安全组和卷。	2021 年 9 月 1 日
C5d、M5d 和 M5d 和 M5d 的休眠支持	使 C5d、M5d 和 R5d 实例类型上运行的新启动实例休眠。	2021 年 8 月 19 日
Amazon EC2 密钥对	Amazon EC2 现在支持 Linux 和 Mac 实例上的 ED25519 密钥。	2021 年 8 月 17 日

网络接口前缀	您可以自动或手动为网络接口分配私有 IPv4 或 IPv6 CIDR 范围。	2021 年 7 月 22 日
事件窗口	您可以为重新启动、停止或终止 Amazon EC2 实例的计划事件定义每周重复运行的自定义事件窗口。	2021 年 7 月 15 日
针对安全组规则的资源 ID 和标记支持	您可以按资源 ID 引用安全组规则。您还可以为安全组规则添加标签。	2021 年 7 月 7 日
弃用 AMI	现在，您可以指定何时弃用 AMI。	2021 年 6 月 11 日
Windows 按秒计费	Amazon EC2 按秒对基于 Windows 和 SQL 服务器的使用收费，最低收取一分钟的费用。	2021 年 6 月 10 日
根卷替换	现在，您可以使用根卷替换任务来替换正在运行的实例的根 EBS 卷。	2021 年 4 月 22 日
使用 S3 存储和还原 AMI	将 EBS 支持的 AMI 存储在 S3 中，然后从 S3 还原它们，以启用 AMI 跨分区复制。	2021 年 4 月 6 日
EC2 Serial Console	通过建立面向实例串行端口的连接来排除引导和网络连接问题。	2021 年 3 月 30 日
启动模式	Amazon EC2 现在在选定的基于 AMD 和 Intel 的 EC2 实例上支持 UEFI 启动。	2021 年 3 月 22 日
创建反向 DNS 记录	您现在可以为弹性 IP 地址设置反向 DNS 查找。	2021 年 2 月 3 日

在创建 AMI 时标记 AMI 和快照	创建 AMI 时，您可以使用相同的标签来标记 AMI 和快照，也可以使用不同的标签来标记它们。	2020 年 12 月 4 日
使用 Amazon EventBridge 监控竞价型实例集事件	创建触发编程操作以响应竞价型实例集状态更改和错误的 EventBridge 规则。	2020 年 11 月 20 日
使用 Amazon EventBridge 监控 EC2 机群事件	创建触发编程操作以响应 EC2 队列 状态更改和错误的 EventBridge 规则。	2020 年 11 月 20 日
删除 instant 机群	在单个 API 调用中删除类型 `instant` 的 EC2 队列 并终止队列中的所有实例。	2020 年 11 月 18 日
对 T3 和 T3a 的休眠支持	使 T3 和 T3a 实例类型上运行的新启动实例休眠。	2020 年 11 月 17 日
Amazon EFS 快速创建	您可以使用 Amazon EFS 快速创建在启动时创建 Amazon EFS 文件系统并将其挂载到实例。	2020 年 11 月 9 日
Wavelength 区域中的容量预留	现在可以在Wavelength 区中创建和使用 容量预留。	2020 年 11 月 4 日
对 I3、M5ad 和 R5ad 的休眠支持	使 I3、M5ad 和 R5ad 实例类型上运行的新启动实例休眠。	2020 年 10 月 21 日
竞价型实例 vCPU 限制	竞价型实例限制现在根据您为完成未处理的请求而正在使用或将使用的运行中竞价型实例所用的 vCPU 数量进行管理。	2020 年 10 月 1 日
Local Zones 中的容量预留	现在可以在 Local Zones 中创建和使用容量预留。	2020 年 9 月 30 日

对 M5a 和 R5a 的休眠支持	使 M5a 和 R5a 实例类型上运行的新启动实例休眠。	2020 年 8 月 28 日
实例元数据提供实例位置和放置信息	`placement` 类别下的新实例元数据字段：区域、放置组名称、分区号、主机 ID 和可用区 ID。	2020 年 8 月 24 日
容量预留组	您可以使用 Amazon Resource Groups 创建容量预留的逻辑集合，然后将实例启动定位到这些组中。	2020 年 7 月 29 日
EC2Launch v2	您可以使用 EC2Launch v2 在实例启动期间执行任务，包括实例停止并稍后启动或者实例重新启动期间，它还可按需执行任务。EC2Launch v2 支持所有版本的 Windows Server，并替换 EC2Launch 和 EC2Config。	2020 年 6 月 30 日
自带 IPv6 地址	您可以将部分或全部 IPv6 地址范围从本地网络引入到您的 Amazon 账户中。	2020 年 5 月 21 日
使用 Systems Manager 参数启动实例	您可以在启动实例时指定 Amazon Systems Manager 参数而不是 AMI。	2020 年 5 月 5 日
停止和启动竞价型实例	停止由 Amazon EBS 支持的竞价型实例并根据需要随时启动它，而不是依赖于停止中断行为。	2020 年 1 月 13 日

为资源添加标签	您可以标记仅出口互联网网关、本地网关、本地网关路由表、本地网关虚拟接口、本地网关虚拟接口组、本地网关路由表 VPC 关联以及本地网关路由表虚拟接口组关联。	2020 年 1 月 10 日
使用会话管理器连接到您的实例	您可以从 Amazon EC2 控制台使用实例来启动会话管理器会话。	2019 年 12 月 18 日
账户级别的默认积分规范	您可以在每个 Amazon 区域的账户级别设置每个可突增性能实例系列的默认积分规范。	2019 年 11 月 25 日
实例类型查找	您可以找到满足您需求的实例类型。	2019 年 11 月 22 日
专属主机	现在，您可以配置专属主机以支持实例系列中的多种实例类型。	2019 年 11 月 21 日
实例元数据服务版本 2	您可以使用 实例元数据服务版本 2，这是一种面向会话的方法，用于请求实例元数据。	2019 年 11 月 19 日
对按需 Windows 实例的休眠支持	您可以使按需 Windows 实例休眠。	2019 年 10 月 14 日
排队购买预留实例	您最早可以提前三年排队购买 Reserved Instance。	2019 年 10 月 4 日
诊断中断	您可以将诊断中断发送到无法访问或无响应的实例以触发内核错误。	2019 年 8 月 14 日

容量优化的分配策略	使用 EC2 实例集或竞价型实例集，您可以通过为正在启动的实例数量提供最佳容量来从 Spot 池启动竞价型实例。	2019 年 8 月 12 日
Elastic Fabric Adapter	EFA 现在支持 Open MPI 3.1.4 和 Intel MPI 2019 Update 4。	2019 年 7 月 26 日
EC2 Instance Connect	EC2 Instance Connect 提供了一种简单且安全的方法以使用 Secure Shell（SSH）连接到实例。	2019 年 6 月 27 日
主机恢复	如果在专属主机上发生意外硬件故障，则在新主机上自动重新启动实例。	2019 年 6 月 5 日
VSS 应用程序一致性快照	通过使用 Amazon Systems Manager Run Command，您可以为附加到您的 Windows 实例的所有 Amazon EBS 卷拍摄应用程序一致性快照。	2019 年 5 月 13 日
Microsoft SQL Server 数据库从 Windows 到 Linux 的平台转换助手	将现有 Microsoft SQL Server 工作负载从 Windows 移动到 Linux 操作系统。更新后的链接指向《Amazon EC2 用户指南》上的 Microsoft SQL Server。	2019 年 5 月 8 日
Windows 自动升级	使用 Amazon Systems Manager 执行 EC2 Windows 实例的自动升级。	2019 年 5 月 6 日

有关 Amazon EC2 实例类型版本的信息，请参阅《Amazon EC2 Instance Types Guide》中的 Document history。

2018 年及更早的历史记录

下表描述《Amazon EC2 用户指南》的重要补充部分（2018 年以及之前年份）。

功能	API 版本	描述	发行日期
分区置放群组	2016-11-15	分区置放群组跨逻辑分区来分配实例，以确保一个分区中的实例不会与其他分区中的实例共享基础硬件。有关更多信息，请参阅分区置放群组。	2018 年 12 月 20 日
休眠 EC2 Linux 实例	2016-11-15	如果 Linux 实例已启用休眠并且满足休眠先决条件，则可以休眠该实例。有关更多信息，请参阅将您的 Amazon EC2 实例休眠。	2018 年 11 月 28 日
Amazon Elastic Inference 加速器	2016-11-15	您可以将 Amazon EI 加速器附加到实例上以添加 GPU 支持的加速，从而减少运行深度学习推理的成本。	2018 年 11 月 28 日
Spot 控制台推荐实例队列	2016-11-15	Spot 控制台根据 Spot 最佳实践（实例多元化）推荐实例队列，以满足您应用程序所需的最低硬件规格（vCPU、内存和存储）。有关更多信息，请参阅创建竞价型实例集。	2018 年 11 月 20 日
新 EC2 队列请求类型：instant	2016-11-15	EC2 队列现在支持新请求类型 `instant`，您可用它来跨实例类型和购买模式同步预配置容量。`instant` 请求在 API 响应中返回启动的实例，不采取更多操作，使您能够控制是否启动实例以及何时启动。有关更多信息，请参阅EC2 实例集和竞价型实例集请求类型。	2018 年 11 月 14 日
Spot 节省信息	2016-11-15	您可以查看使用单个竞价型实例集或所有竞价型实例的竞价型实例所节省的成本。有关更多信息，请参阅通过购买竞价型实例实现节省。	2018 年 11 月 5 日
用于优化 CPU 选项的控制台支持	2016-11-15	启动实例时，您可以使用 Amazon EC2 控制台优化 CPU 选项以适应特定的工作负载或业务需求。有关更多信息，请参阅Amazon EC2 实例适用的 CPU 选项。	2018 年 10 月 31 日

功能	API 版本	描述	发行日期
控制台支持从实例创建启动模板	2016-11-15	您可以通过 Amazon EC2 控制台，以实例为基础创建新的启动模板。有关更多信息，请参阅创建 Amazon EC2 启动模板。	2018 年 10 月 30 日
按需容量预留	2016-11-15	您可在特定可用区中为 Amazon EC2 实例预留容量任意持续时间。这使您能够独立于预留实例 (RI) 提供的账单折扣来创建和管理容量预留。有关更多信息，请参阅使用 EC2 按需容量预留来预留计算容量。	2018 年 10 月 25 日
自带 IP 地址 (BYOIP)	2016-11-15	您可将自己的全部或部分公有 IPv4 地址从本地网络引入到 Amazon 账户中。在将地址范围引入 Amazon 中之后，它会在您的账户中显示为地址池。您可从地址池创建弹性 IP 地址，并将其用于您的 Amazon 资源。有关更多信息，请参阅自带 IP 地址（BYOIP）到 Amazon EC2。	2018 年 10 月 23 日
在创建时标记专属主机和控制台支持	2016-11-15	您可在创建时标记专属主机，并可以使用 Amazon EC2 控制台管理专属主机标签。有关更多信息，请参阅分配一台 Amazon EC2 专属主机供您的账户使用。	2018 年 10 月 22 日
针对竞价型实例集计划扩展的控制台支持。	2016-11-15	根据日期和时间增加或减少队列的当前容量。有关更多信息，请参阅计划扩展：按计划扩展竞价型实例集。	2018 年 9 月 20 日
EC2 队列的分配策略	2016-11-15	您可以指定是按价格（最低价格优先）还是优先级（最高优先级优先）来满足按需容量。您可以指定在其中分配您的目标 Spot 容量的 Spot 池数量。有关更多信息，请参阅使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量。	2018 年 7 月 26 日

功能	API 版本	描述	发行日期
竞价型实例集的分配策略	2016-11-15	您可以指定是按价格（最低价格优先）还是优先级（最高优先级优先）来满足按需容量。您可以指定在其中分配您的目标 Spot 容量的 Spot 池数量。有关更多信息，请参阅使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量。	2018 年 7 月 26 日
自动化快照生命周期	2016-11-15	您可以使用 Amazon Data Lifecycle Manager 来自动创建和删除 EBS 卷的快照。有关更多信息，请参阅 Amazon Data Lifecycle Manager。	2018 年 7 月 12 日
启动模板 CPU 选项	2016-11-15	在使用命令行工具创建启动模板时，您可以优化 CPU 选项以适合特定工作负载或业务需求。有关更多信息，请参阅创建 Amazon EC2 启动模板。	2018 年 7 月 11 日
标记专属主机	2016-11-15	您可以标记专属主机。	2018 年 7 月 5 日
获取最新的控制台输出	2016-11-15	使用 get-console-output Amazon CLI 命令时，您可以检索某些实例类型的最新控制台输出。	2018 年 5 月 9 日
优化 CPU 选项	2016-11-15	启动实例时，您可以优化 CPU 选项以适应特定的工作负载或业务需求。有关更多信息，请参阅Amazon EC2 实例适用的 CPU 选项。	2018 年 5 月 8 日
EC2 Fleet	2016-11-15	您可以使用 EC2 实例集在不同的 EC2 实例类型和可用区以及按需型实例、预留实例和竞价型实例采购模型之间启动一组实例。有关更多信息，请参阅 EC2 实例集和竞价型实例集。	2018 年 5 月 2 日
竞价型实例集 中的按需型实例	2016-11-15	您可以在竞价型实例集请求中包含按需容量请求，以确保始终拥有实例容量。有关更多信息，请参阅 EC2 实例集和竞价型实例集。	2018 年 5 月 2 日
创建时标记 EBS 快照	2016-11-15	在创建过程中，您可以将标签应用于快照。	2018 年 4 月 2 日

功能	API 版本	描述	发行日期
更改置放群组	2016-11-15	您可以将实例移入或移出置放群组，也可以更改实例的置放群组。有关更多信息，请参阅更改 Amazon EC2 实例的置放。	2018 年 3 月 1 日
较长的资源 ID	2016-11-15	您可以为更多资源类型启用较长 ID 格式。	2018 年 2 月 9 日
网络性能改进	2016-11-15	在其他实例或 Amazon S3 之间发送或接收网络通信时，集群置放群组之外的实例现在可以受益于增加的带宽。	2018 年 1 月 24 日
标记弹性 IP 地址	2016-11-15	您可以标记您的弹性 IP 地址。	2017 年 12 月 21 日
Amazon Time Sync Service	2016-11-15	您可以使用 Amazon Time Sync Service 在您的实例上保持准确的时间。有关更多信息，请参阅在 EC2 实例上实现精确时钟和时间同步。	2017 年 11 月 29 日
T2 无限	2016-11-15	T2 无限实例可以突增到基准以上所需的时间。有关更多信息，请参阅可突增性能实例。	2017 年 11 月 29 日
启动模板	2016-11-15	启动模板可以包含用于启动实例的全部或部分参数，而无需在每次启动实例时都指定这些参数。有关更多信息，请参阅在 Amazon EC2 启动模板中存储实例启动参数。	2017 年 11 月 29 日
分布置放	2016-11-15	建议在具有少量应单独放置的重要实例的应用程序中使用分布置放群组。有关更多信息，请参阅分布置放群组。	2017 年 11 月 29 日
竞价型实例休眠	2016-11-15	在发生中断时，Spot 服务可以将竞价型实例休眠。	2017 年 11 月 28 日

功能	API 版本	描述	发行日期
竞价型实例集目标跟踪	2016-11-15	您可以为竞价型实例集设置目标跟踪扩展策略。有关更多信息，请参阅目标跟踪扩展：通过确定特定指标的目标值来扩展竞价型实例集。	2017 年 11 月 17 日
竞价型实例集与 Elastic Load Balancing 集成	2016-11-15	您可以将一个或多个负载均衡器附加到竞价型实例集。	2017 年 11 月 10 日
合并和拆分可转换预留实例	2016-11-15	您可以合并两个或更多可转换预留实例，也可以将其与新可转换预留实例交换。您还可以使用修改过程将 可转换预留实例 拆分为较小的预留。有关更多信息，请参阅交换 可转换预留实例。	2017 年 11 月 6 日
修改 VPC 租赁	2016-11-15	可以将 VPC 的实例租赁属性从 `dedicated` 改为 `default`。有关更多信息，请参阅更改 VPC 的实例租赁。	2017 年 10 月 16 日
按秒计费	2016-11-15	Amazon EC2 按秒对基于 Linux 的使用收费，最低收取一分钟的费用。	2017 年 10 月 2 日
在中断时停止	2016-11-15	您可以指定在竞价型实例中断时 Amazon EC2 应将其停止还是终止。有关更多信息，请参阅竞价型实例中断行为。	2017 年 9 月 18 日
给 NAT 网关加标签	2016-11-15	您可以给自己的 NAT 网关加标签。有关更多信息，请参阅标记 资源。	2017 年 9 月 7 日
安全组规则说明	2016-11-15	您可以向安全组规则添加说明。	2017 年 8 月 31 日
Elastic Graphics	2016-11-15	将 Elastic Graphics 加速器附加到您的实例来提升应用程序的图形性能。	2017 年 8 月 29 日

功能	API 版本	描述	发行日期
恢复弹性 IP 地址	2016-11-15	如果您释放了一个在 VPC 中使用的弹性 IP 地址，则可能能够恢复它。	2017 年 8 月 11 日
标记竞价型实例集实例	2016-11-15	您可以将您的竞价型实例集配置为自动标记其启动的实例。	2017 年 7 月 24 日
在创建过程中，为资源添加标签	2016-11-15	在创建过程中，您可以将标签应用于实例和卷。有关更多信息，请参阅标记 资源。此外，您可以使用基于标签的资源级权限来控制应用的标签。有关更多信息，请参阅 在创建过程中授予标记 Amazon EC2 资源的权限。	2017 年 3 月 28 日
在附加的 EBS 卷上进行修改	2016-11-15	对于附加到大多数 EC2 实例的大多数 EBS 卷，您可以在不分离卷或不停止实例的情况下修改卷大小、类型和 IOPS。	2017 年 2 月 13 日
附加 IAM 角色	2016-11-15	您可以为现有实例附加、分离或替换 IAM 角色。有关更多信息，请参阅适用于 Amazon EC2 的 IAM 角色。	2017 年 2 月 9 日
专用竞价型实例	2016-11-15	您可在 Virtual Private Cloud (VPC) 中的单租户硬件上运行竞价型实例。有关更多信息，请参阅在单租户硬件上启动。	2017 年 1 月 19 日
IPv6 支持	2016-11-15	您可以将一个 IPv6 CIDR 与 VPC 和子网关联，并为 VPC 中的实例分配 IPv6 地址。有关更多信息，请参阅Amazon EC2 实例 IP 寻址。	2016 年 12 月 1 日
竞价型实例集的自动扩展		现在，您可以为竞价型实例集设置扩展策略。有关更多信息，请参阅了解竞价型实例集的自动扩展。	2016 年 9 月 1 日
Elastic Network Adapter (ENA)	2016-04-01	您现在可以将 ENA 用于增强网络。有关更多信息，请参阅 Amazon EC2 实例上的增强联网功能。	2016 年 6 月 28 日

功能	API 版本	描述	发行日期
增强了对查看和修改较长 ID 的支持	2016-04-01	您现在可以查看和修改其他 IAM 用户、IAM 角色或根用户的较长 ID 设置。	2016 年 6 月 23 日
在 Amazon 账户之间复制加密的 Amazon EBS 快照	2016-04-01	您现在可以在 Amazon 账户之间复制加密的 EBS 快照。	2016 年 6 月 21 日
捕获实例控制台的屏幕截图	2015-10-01	您现在可以在调试无法访问的实例时获取其他信息。有关更多信息，请参阅 捕获无法访问的实例的屏幕截图。	2016 年 24 月 5 日
两种新的 EBS 卷类型	2015-10-01	您现在可以创建经过吞吐量优化的 HDD (st1) 以及冷数据 HDD (sc1) 卷。	2016 年 4 月 19 日
添加了针对 Amazon EC2 的新的 NetworkPacketsIn 和 NetworkPacketsOut 指标		添加了针对 Amazon EC2 的新的 NetworkPacketsIn 和 NetworkPacketsOut 指标。有关更多信息，请参阅实例指标。	2016 年 3 月 23 日
竞价型实例集的 CloudWatch 指标		您现在可以获取竞价型实例集的 CloudWatch 指标。有关更多信息，请参阅 使用 CloudWatch 监控 EC2 实例集或竞价型实例集。	2016 年 3 月 21 日
计划实例	2015-10-01	利用计划的预留实例 (计划实例)，您可以购买具有指定的开始时间和持续时间，并且每日、每周或每月重复一次的容量预留。	2016 年 1 月 13 日
较长的资源 ID	2015-10-01	我们将逐步引入某些 Amazon EC2 和 Amazon EBS 资源类型的更长 ID。在选择周期内，您可以为支持的资源类型启用较长 ID 格式。	2016 年 1 月 13 日

功能	API 版本	描述	发行日期
ClassicLink DNS 支持	2015-10-01	您可以对您的 VPC 启用 ClassicLink DNS 支持，以使定位在链接的 EC2-Classic 实例与 VPC 中的实例之间的 DNS 主机名解析为私有 IP 地址而不是公有 IP 地址。	2016 年 1 月 11 日
专属主机	2015-10-01	Amazon EC2 专属主机是指实例容量供您专用的物理服务器。有关更多信息，请参阅Amazon EC2 专属主机。	2015 年 11 月 23 日
竞价型实例持续时间	2015-10-01	现在，您可以为竞价型实例指定持续时间。不支持竞价型限制（2023 年 1 月）。	2015 年 10 月 6 日
竞价型实例集修改请求	2015-10-01	您现在可以修改竞价型实例集请求的目标容量。有关更多信息，请参阅修改竞价型实例集请求。	2015 年 9 月 29 日
竞价型实例集多样化分配策略	2015-04-15	您现在可以使用单个竞价型实例集请求在多个 Spot 池中分配竞价型实例。有关更多信息，请参阅 使用分配策略确定 EC2 实例集或竞价型实例集如何满足竞价型和按需型容量。	2015 年 9 月 15 日
竞价型实例集实例权重	2015-04-15	您现在可以定义每个实例类型在应用程序性能中所占的容量单位，并相应地为每个 Spot 池的竞价型实例调整愿意支付的金额。有关更多信息，请参阅 使用实例权重来管理 EC2 实例集或竞价型实例集的成本和性能。	2015 年 8 月 31 日
新的重启警报操作和用于警报操作的新 IAM 角色		增加了重启警报操作和与警报操作一起使用的新 IAM 角色。有关更多信息，请参阅创建停止、终止、重启或恢复实例的警报。	2015 年 7 月 23 日
Spot Fleets	2015-04-15	您可以管理竞价型实例的集合或队列，而不必管理单独的竞价型实例请求。有关更多信息，请参阅 EC2 实例集和竞价型实例集。	2015 年 5 月 18 日

功能	API 版本	描述	发行日期
将弹性 IP 地址迁移至 EC2-Classic	2015-04-15	您无法将已分配为在 EC2-Classic 中使用的弹性 IP 地址迁移到在 VPC 中使用。	2015 年 5 月 15 日
将具有多个磁盘的虚拟机作为 AMI 导入	2015-03-01	VM Import 过程现在支持将具有多个磁盘的虚拟机作为 AMI 导入。有关更多信息，请参阅 VM Import/Export 用户指南 中的使用 VM Import/Export 将虚拟机作为映像导入。	2015 年 4 月 23 日
Systems Manager		Systems Manager 使您可以配置和管理您的 EC2 实例。	2015 年 2 月 17 日
Systems Manager for Microsoft SCVMM 1.5		您现在可以使用 Systems Manager for Microsoft SCVMM 启动实例，并将虚拟机从 SCVMM 导入到 Amazon EC2。	2015 年 1 月 21 日
EC2 实例的自动恢复		您可以创建 Amazon CloudWatch 警报用于监控 Amazon EC2 实例，并且在实例受损（由于发生底层硬件故障或需要 Amazon 参与才能修复的问题）时自动恢复实例。恢复的实例与原始实例相同，包括实例 ID、IP 地址以及所有实例元数据。 有关更多信息，请参阅实例自动恢复。	2015 年 1 月 12 日
ClassicLink	2014-10-01	您可使用 ClassicLink 将 EC2-Classic 实例链接到您账户中的 VPC。您可以将 VPC 安全组与 EC2-Classic 实例关联起来，从而允许 EC2-Classic 实例与 VPC 中使用私有 IP 地址的实例进行通信。	2015 年 1 月 7 日

功能	API 版本	描述	发行日期
竞价型实例终止通知		防范竞价型实例中断的最佳方法是为应用程序设计容错能力。此外，您还可以利用竞价型实例中断通知，该通知可在 Amazon EC2 必须停止或终止您的竞价型实例时，提前两分钟发出警告。 有关更多信息，请参阅竞价型实例中断通知。	2015 年 1 月 5 日
Systems Manager for Microsoft SCVMM		Systems Manager for Microsoft SCVMM 可为从 Microsoft SCVMM 管理 Amazon 资源（如 EC2 实例）提供简单易用的界面。	2014 年 10 月 29 日
`DescribeVolumes` 分页支持	2014-09-01	`DescribeVolumes` API 调用现在可使用 `MaxResults` 和 `NextToken` 参数支持结果分页。有关更多信息，请参阅 Amazon EC2 API Reference 中的 DescribeVolumes。	2014 年 10 月 23 日
增加了对 Amazon CloudWatch Logs 的支持		您可以使用 Amazon CloudWatch Logs 来监控、存储和访问您的系统、应用程序以及来自实例或其他来源的自定义日志文件。然后可以使用 Amazon CloudWatch 控制台、Amazon CLI 中的 CloudWatch Logs 命令或使用 CloudWatch Logs 开发工具包，从 CloudWatch Logs 检索关联的日志数据。	2014 年 7 月 10 日
新 EC2 Service Limits (EC2 服务限制) 页面		使用 Amazon EC2 控制台中的 EC2 Service Limits (EC2 服务限制) 页面可按区域查看 Amazon EC2 和 Amazon VPC 提供的资源的当前限制。	2014 年 6 月 19 日
Amazon EBS 通用型 SSD 卷	2014-05-01	通用型 SSD 卷提供经济实惠的存储，是广泛工作负载的理想选择。这些卷可提供不超过 10 毫秒的延迟，能突增至 3000 IOPS 很长时间，基准性能为 3 IOPS/GiB。通用型 SSD 卷的大小范围为 1 GiB 到 1 TiB。	2014 年 6 月 16 日

功能	API 版本	描述	发行日期
Amazon Management Pack		Amazon 管理包现在支持 System Center Operations Manager 2012 R2。	2014 年 5 月 22 日
Amazon EBS encryption	2014-05-01	Amazon EBS 加密 提供 EBS 数据卷和快照的无缝加密，无需构建和维护安全密钥管理基础设施。通过使用 Amazon 托管式密钥 加密数据，EBS 加密可保护静态数据的安全。加密还发生在托管 EC2 实例的服务器上，当数据在 EC2 实例和 EBS 存储之间移动时提供数据加密。	2014 年 5 月 21 日
导入 Linux 虚拟机	2013-10-15	VM Import 过程现在支持 Linux 实例的导入。有关更多信息，请参阅 VM Import/Export 用户指南。	2013 年 12 月 16 日
RunInstances 的资源级权限	2013-10-15	您现在可以在 Amazon Identity and Access Management 中创建策略以控制 Amazon EC2 RunInstances API 操作的资源级权限。有关更多信息以及示例策略，请参阅适用于 Amazon EC2 的 Identity and Access Management。	2013 年 11 月 20 日
从 Amazon Web Services Marketplace 启动实例		您现在可以使用 Amazon EC2 启动向导从 Amazon Web Services Marketplace 启动实例。有关更多信息，请参阅从 Amazon Web Services Marketplace AMI 中启动 Amazon EC2 实例。	2013 年 11 月 11 日
新启动向导		提供一个重新设计的新的 EC2 启动向导。有关更多信息，请参阅 使用控制台中的启动实例向导来启动 EC2 实例。	2013 年 10 月 10 日
修改预留实例的实例类型	2013-10-01	您现在可以修改同一系列 (例如 M1、M2、M3、C1) 中 Linux 预留实例的实例类型。有关更多信息，请参阅修改 预留实例。	2013 年 10 月 09 日

功能	API 版本	描述	发行日期
修改 Amazon EC2 预留实例	2013-08-15	您现在可以修改区域中的预留实例。有关更多信息，请参阅修改 预留实例。	2013 年 9 月 11 日
分配公有 IP 地址	2013-07-15	在 VPC 中启动实例时，现在可以分配公有 IP 地址。有关更多信息，请参阅在启动时分配公有 IPv4 地址。	2013 年 8 月 20 日
授予资源级权限	2013-06-15	Amazon EC2 支持新的亚马逊资源名称 (ARN) 和条件键。有关更多信息，请参阅Amazon EC2 基于身份的策略。	2013 年 7 月 8 日
增量快照副本	2013-02-01	您现在可以执行增量快照副本。	2013 年 6 月 11 日
Amazon Management Pack		Amazon Management Pack 会将 Amazon EC2 实例与其内部运行的 Windows 或 Linux 操作系统相关联。Amazon 管理包是 Microsoft System Center Operations Manager 的一种扩展程序。	2013 年 5 月 8 日
新 Tags (标签) 页面		Amazon EC2 控制台中提供一个新的 Tags (标签) 页面。有关更多信息，请参阅标记 Amazon EC2 资源。	2013 年 4 月 4 日
将 AMI 从一个区域复制到另一个区域	2013-02-01	您可以将 AMI 从一个区域复制到另一个区域，以便快速轻松地在多个 Amazon 区域中启动一致的实例。 有关更多信息，请参阅复制 Amazon EC2 AMI。	2013 年 3 月 11 日

功能	API 版本	描述	发行日期
在默认 VPC 中启动实例	2013-02-01	根据各区域的情况，您的 Amazon 账户可以在 EC2-Classic 或 VPC 中启动实例，或者仅在 VPC 中启动。如果您只能在 VPC 中启动实例，我们会为您创建一个默认 VPC。当您启动实例时，我们会将其启动为默认 VPC，除非您创建了非默认 VPC 并在启动实例时对其做出指定。	2013 年 3 月 11 日
EBS 快照副本	2012-12-01	您可以使用快照备份创建数据备份、创建新 Amazon EBS 卷，或创建亚马逊机器映像（AMI）。	2012 年 12 月 17 日
已更新 Provisioned IOPS SSD 卷的 EBS 指标和状态检查	2012 年 10 月 1 日	已更新 EBS 指标，以便包含 Provisioned IOPS SSD 卷的两项新指标。还添加了 Provisioned IOPS SSD 卷的新状态检查。	2012 年 11 月 20 日
竞价型实例请求状态	2012 年 10 月 1 日	竞价型实例请求状态简化了确定您的 Spot 请求状态的过程。	2012 年 10 月 14 日
Amazon EC2 预留实例 Marketplace	2012 年 8 月 15 日	预留实例 Marketplace 将想要出售不再需要的 Amazon EC2 预留实例的卖方与正在寻找购买额外容量的买方匹配起来。通过预留实例 Marketplace 购买和出售的预留实例与其他预留实例一样工作，不同的是他们受标准条款限制更少，并能以不同价格出售。	2012 年 9 月 11 日
适用于 Amazon EBS 的 Provisioned IOPS SSD	2012 年 7 月 20 日	Provisioned IOPS SSD 卷为 I/O 密集型工作负载，如依赖于稳定和快速响应时间的数据库应用程序，提供可预测、高性能的服务。	2012 年 7 月 31 日

功能	API 版本	描述	发行日期
IAM 对 Amazon EC2 实例的作用	2012-06-01	适用于 Amazon EC2 实例的 IAM 角色提供： • 在 Amazon EC2 实例上运行的应用程序的 Amazon 访问密钥。 • Amazon EC2 实例上的 Amazon 访问密钥的自动交替。 • 为 Amazon EC2 实例上请求 Amazon 服务的运行应用程序细调权限。	2012 年 6 月 11 日
让启动和处理中断可能性变得更加容易的竞价型实例功能。		现在您可以按照以下方式管理您的竞价型实例： • 使用 Auto Scaling 启动配置指定您愿意为竞价型实例支付的金额，并设置计划来指定您愿意为竞价型实例支付的金额。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的请求竞价型实例。 • 实例启动或终止时获得通知。 • 在 Amazon 资源堆栈中使用 Amazon CloudFormation 模板来启动竞价型实例。	2012 年 6 月 7 日
用于 Amazon EC2 状态检查的 EC2 实例导出和时间戳	2012 年 5 月 1 日	已添加将您原先导入到 EC2 的 Windows Server 实例导出的支持。 已添加对实例状态和系统状态时间戳的支持，该戳记显示状态检查失败的日期和时间。	2012 年 5 月 25 日
Amazon VPC 实例和系统状态检查中的 EC2 实例导出和时间戳	2012 年 5 月 1 日	已添加将 EC2 实例导出至 Citrix Xen、Microsoft Hyper-V 和 VMware vSphere 的支持。 已添加多实例和系统状态检查中的时间戳的支持。	2012 年 5 月 25 日

功能	API 版本	描述	发行日期
Amazon Web Services Marketplace AMI	2012 年 4 月 1 日	增加了对 Amazon Web Services Marketplace AMI 的支持。	2012 年 4 月 19 日
预留实例定价套餐	2011 年 12 月 15 日	增加了新的一节来讨论如何充分利用预留实例定价套餐自带的折扣价格。	2012 年 3 月 5 日
Amazon Virtual Private Cloud 中的适用于 EC2 实例的 Elastic Network Interfaces (ENIs)	2011 年 12 月 1 日	已添加有关 VPC 中适用于 EC2 实例的 Elastic Network Interfaces (ENIs) 的新章节。有关更多信息，请参阅弹性网络接口。	2011 年 12 月 21 日
Amazon EC2 预留实例的新产品类型	2011 年 11 月 1 日	您可以选择各种各样的预留实例产品，以满足您对实例的预期使用要求。	2011 年 12 月 1 日
Amazon EC2 实例状态	2011 年 11 月 1 日	您可以查看您的实例状态的其他详细信息，包括 Amazon 计划的可能会影响您的实例的事件。这些操作活动包括，执行软件更新和安全性补丁程序所要求的实例重启，和出现硬件问题时所需的实例停止。有关更多信息，请参阅监控 Amazon EC2 实例的状态。	2011 年 11 月 16 日
Amazon VPC 中的竞价型实例	2011-07-15	增加了 Amazon VPC 中的竞价型实例支持的有关信息。通过此更新，用户可以在 Virtual Private Cloud (VPC) 中启动竞价型实例。通过在 VPC 中启动竞价型实例，竞价型实例的用户可以享受到 Amazon VPC 带来的好处。	2011 年 10 月 11 日
适用于 CLI 工具用户的简化的 VM Import 过程	2011-07-15	`ImportInstance` 和 `ImportVolume` 的增强功能简化了 VM Import 过程；现在，创建导入任务后，系统会将映像上传到 Amazon EC2 上。此外，通过引入 `ResumeImport` ，用户可以从任务停止的时间点重新开始未完成的上传。	2011 年 9 月 15 日

功能	API 版本	描述	发行日期
导入 VHD 文件格式的支持		VM Import 现在可导入 VHD 格式的虚拟机图像文件。VHD 文件格式与 Citrix Xen 和 Microsoft Hyper-V 虚拟化平台兼容。通过这种新产品，VM Import 现在支持 RAW、VHD 和 VMDK (与 VMware ESX 兼容的) 图像格式。有关更多信息，请参阅 VM Import/Export 用户指南。	2011 年 8 月 24 日
更新至适用于 VMware vCenter 的 Amazon EC2 VM Import 连接器		适用于 VMware vCenter 虚拟设备 (连接器) 的 1.1 版本的 Amazon EC2 VM Import 连接器的有关补充信息。此次更新包括访问因特网的代理支持、更好的错误处理方法、任务进度栏准确性的提高以及一些 bug 修复。	2011 年 6 月 27 日
竞价型实例可用区定价更改	2011 年 5 月 15 日	增加了竞价型实例可用区定价功能的有关信息。在本版本中，我们增加了新的可用区定价选项，当您查询竞价型实例请求和 Spot 价格历史记录时，返回的信息中包括这些新的定价选项。通过这些新增功能，可以更方便地确定将竞价型实例启动到特定可用区所需的价格。	2011 年 5 月 26 日
Amazon Identity and Access Management		已添加 Amazon Identity and Access Management（IAM）的有关信息，使用户可以指定借助 Amazon EC2 资源一般能使用哪些 Amazon EC2 功能。有关更多信息，请参阅适用于 Amazon EC2 的 Identity and Access Management。	2011 年 4 月 26 日
专用实例		专用实例是在 Amazon Virtual Private Cloud (Amazon VPC) 中启动的，是在主机硬件层次上物理隔离的实例。专用实例让您能享用 Amazon VPC 和 Amazon 云的好处，包括按需弹性配置、仅为实际用量付费等，但同时也在硬件层次上隔离您的 Amazon EC2 计算实例。有关更多信息，请参阅Amazon EC2 专用实例。	2011 年 3 月 27 日

功能	API 版本	描述	发行日期
预留实例更新至 Amazon 管理控制台		更新至 Amazon 管理控制台让用户查看他们的预留实例以及购买额外预留实例，包括专用预留实例，变得更加简单。	2011 年 3 月 27 日
元数据信息	2011 年 1 月 1 日	已添加元数据的有关信息，将反映 2011-01-01 版本中的更改。有关更多信息，请参阅使用实例元数据管理 EC2 实例和实例元数据类别。	2011 年 3 月 11 日
适用于 VMware vCenter 的 Amazon EC2 VM Import 连接器。		已添加适用于 VMware vCenter 虚拟设备 (连接器) 的 Amazon EC2 VM Import 连接器的有关信息。这个连接器是一个用于 VMware vCenter 的插件，集成了 VMware vSphere 客户端，并提供了一个图形用户界面，您可以用它来将 VMware 虚拟机导入到 Amazon EC2 中。	2011 年 3 月 3 日
实施卷分离		您现在可以使用 Amazon Web Services 管理控制台 强制将 Amazon EBS 卷从实例分离。	2011 年 2 月 23 日
实例终止保护		您现在可以使用 Amazon 管理控制台来防止实例终止。有关更多信息，请参阅更改实例终止保护。	2011 年 2 月 23 日
VM Import	2010-11-15	已添加有关 VM Import 的信息，允许您将虚拟机或卷导入到 Amazon EC2 中。有关更多信息，请参阅 VM Import/Export 用户指南。	2010 年 12 月 15 日
实例的基本监控	2010-08-31	已添加有关 EC2 实例的基本监控的信息。	2010 年 12 月 12 日
筛选条件和标记	2010-08-31	已添加列举、筛选和标记资源的有关信息。有关更多信息，请参阅查找 Amazon EC2 资源和标记 Amazon EC2 资源。	2010 年 9 月 19 日

功能	API 版本	描述	发行日期
幂等实例启动	2010-08-31	已添加关于确保实例运行时的幂等性的信息。	2010 年 9 月 19 日
Amazon EC2 的 Amazon Identity and Access Management		Amazon EC2 现已集成（Amazon Identity and Access ManagementIAM）。有关更多信息，请参阅适用于 Amazon EC2 的 Identity and Access Management。	2010 年 9 月 2 日
Amazon VPC IP 地址指定	2010-06-15	Amazon VPC 用户现在可以指定用于分配在 VPC 中启动的实例的 IP 地址。	2010 年 7 月 12 日
对 Amazon EBS 卷进行 Amazon CloudWatch 监控		Amazon CloudWatch 监控现在可以自动用于 Amazon EBS 卷。	2010 年 6 月 14 日
Windows 预留实例		Amazon EC2 现在支持 Windows 预留实例。	2010 年 2 月 22 日