亚马逊云科技

管理指南

Amazon EMR

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon EMR: 管理指南

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

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

Table of Contents

什么是 Amazon EMR？ 1
概览 1
了解集群和节点 1
向集群提交工作 2
处理数据 3
了解集群的生命周期 4
优点 6
节省成本 6
Amazon集成 7
部署 7
可扩展性和灵活度 7
可靠性 8
安全性 8
监控 9
管理界面 10
架构 10
存储 11
集群资源管理 11
数据处理框架 12
应用程序和项目 12
设置 Amazon EMR 14
注册一个 Amazon Web Services 账户 14
保护 IAM 用户 14
为 SSH 创建 Amazon EC2 密钥对 15
后续步骤 15
入门教程 16
概述 16
第 1 步：计划和配置 17
为 Amazon EMR 准备存储 17
为 Amazon EMR 准备含有输入数据的应用程序 17
启动 Amazon EMR 集群 19
第 2 步：管理 23
将工作提交到 Amazon EMR 23
查看结果 28

第 3 步：清除 32
终止集群 32
删除 S3 资源 34
后续步骤 35
了解 Amazon EMR 的大数据应用程序 35
规划集群硬件、联网和安全 35
管理集群 35
使用不同的界面 35
浏览 EMR 技术博客 36
控制台中的新增功能 37
我使用的控制台 37
使用旧控制台 38
差异摘要 38
新旧控制台之间的集群兼容性 39
创建集群时的差异 39
列出和搜索集群时的差异 41
查看或编辑集群详细信息时的差异 41
使用安全配置时的差异 42
Amazon EMR Studio 44
主要特征 44
功能历史记录 45
工作原理 45
身份验证和用户登录 46
访问控制 49
Workspaces 49
笔记本存储 50
注意事项 50
注意事项 50
已知问题 51
功能限制 53
Service Limits 53
VPC 和子网最佳实践 53
集群要求 54
配置 EMR Studio 56
创建 EMR Studio 所需的管理员权限 56
设置 Amazon EMR Studio 61

管理 Studio 117
控制 EMR Studio 网络流量 124
创建集群模板 126
基于 Git 的存储库的访问和权限 132
优化 Spark 任务 135
使用 EMR Studio 136
WorkSpace 基础知识 137
Workspace 协作 143
使用运行时系统角色运行 Workspace 146
以编程方式运行 Workspace 笔记本 151
用 SQL Explorer 浏览数据 151
将计算资源附加到 Workspace 152
链接 Git 存储库 158
调试应用程序和任务 161
安装内核和库 164
Magic 命令 166
将多语言笔记本与 Spark 内核配合使用 174
EMR Notebooks 177
新控制台中的 Notebooks 178
关于过渡 178
您需要做什么？ 178
Workspace 优势 179
所需的权限 179
注意事项 180
集群要求 181
集群发行版的功能差异 181
并发附加的 EMR Notebooks 的限制 182
Jupyter notebook 和 Python 版本 183
安全注意事项 183
创建笔记本 183
使用 EMR Notebooks 186
了解笔记本状态 186
使用笔记本编辑器 187
更改集群 189
删除笔记本和笔记本文件 189
共享笔记本文件 190

编程执行 191
概述 191
权限 191
限制 193
示例 193
CLI 命令示例 193
Boto3 SDK 示例脚本 199
Ruby 示例脚本 202
适用于 Spark 的用户模拟 204
设置 Spark 用户模拟 204
使用 Spark 任务监控小组件 205
安全性 206
安装并使用内核和库 207
..... 207
在集群主节点上安装内核和 Python 库 208
笔记本范围的库注意事项和限制 210
使用笔记本范围的库 210
将基于 Git 的存储库与 EMR Notebooks 关联 211
先决条件和注意事项 213
将基于 Git 的存储库添加到 Amazon EMR 216
更新或删除基于 Git 的存储库。 218
链接或取消链接基于 Git 的存储库 219
使用关联的 Git 存储库创建新的笔记本 222
在笔记本中使用 Git 存储库 222
计划和配置集群 224
快速启动集群 224
配置集群位置和数据存储 226
选择一个Amazon区域 226
使用存储和文件系统 227
准备输入数据 230
配置输出位置 244
计划和配置主节点 249
支持的应用程序和功能 250
启动具有多个主节点的 Amazon EMR 集群 257
Amazon EMR 与 EC2 置放群组的集成 260
注意事项和最佳实践 265

Amazon Outposts 上的 EMR 集群 266
先决条件 267
限制 267
网络连接注意事项 268
在 Amazon Outposts 上创建 Amazon EMR 集群 268
Amazon Local Zones 上的 EMR 集群 270
支持的实例类型 270
在 Local Zones 上创建 Amazon EMR 集群 271
配置 Docker 272
Docker 注册表 273
配置 Docker 注册表 274
将 YARN 配置为访问 Amazon ECR on EMR 6.0.0 及更早版本 275
控制集群终止 277
将集群配置为在步骤执行后继续或终止 277
使用自动终止策略 280
使用终止保护 285
使用 AMI 292
概述 292
使用默认 AMI 292
使用自定义 AMI 324
创建集群时更改 Amazon Linux 发行版 336
指定 Amazon EBS 根设备卷大小 337
配置集群软件 339
创建引导操作 340
配置集群硬件和联网 345
了解节点类型 346
配置 Amazon EC2 实例 348
配置集群日志记录和调试 782
默认日志文件 782
将日志文件归档到 Amazon S3 783
日志位置 787
启用调试工具 789
调试选项信息 790
为集群添加标签 791
标签限制 792
针对账单为资源添加标签 792

向集群添加标签 793
查看集群上的标签 796
从集群中删除标签 797
驱动程序和第三方应用程序集成 798
将商业智能工具与 Amazon EMR 结合使用 799
安全性 800
安全配置 800
数据保护 800
Amazon Identity and Access Management 与 Amazon EMR 结合使用 800
Kerberos 801
Lake Formation 801
Secure Socket Shell (SSH) 801
Amazon EC2 安全组 801
默认 Amazon Linux AMI 的更新 802
使用安全配置设置集群安全性 802
创建安全配置 802
为集群指定安全配置 827
数据保护 829
加密静态数据和传输中的数据 829
IAM 与 Amazon EMR 结合使用 841
受众 841
使用身份进行身份验证 842
使用策略管理访问 844
Amazon EMR 如何与 IAM 配合使用 846
Amazon EMR 步骤的运行时角色 852
配置 Amazon EMR 的服务角色 859
基于身份的策略示例 907
对集群节点进行身份验证 940
对 SSH 凭证使用 EC2 密钥对 940
使用 Kerberos 身份验证 941
使用 LDAP 身份验证 974
将 Amazon EMR 与 Lake Formation 集成 983
Amazon EMR 如何与 Lake Formation 结合使用 984
先决条件 984
使用 Amazon EMR 启用 Lake Formation 985
Hudi 和 Lake Formation 988

注意事项 990
将 Amazon EMR 与 Apache Ranger 集成 990
Ranger 概览 991
应用程序支持和限制 993
为 Apache Ranger 设置 Amazon EMR 995
Apache Ranger 插件 1012
Apache Ranger 疑难解答 1037
使用安全组控制网络流量 1040
使用 Amazon EMR 管理的安全组 1042
使用其它安全组 1050
指定安全组 1050
EMR Notebooks 的安全组 1053
阻止公有访问 1055
合规性验证 1060
故障恢复能力 1061
基础设施安全性 1061
使用接口 VPC 终端节点连接到 Amazon EMR 1062
管理集群 1066
连接到集群 1066
连接之前 1067
使用 SSH 连接到主节点 1070
向集群提交工作 1092
使用控制台添加步骤 1093
使用 CLI 添加步骤 1096
运行多个步骤 1098
查看步骤 1099
取消步骤 1099
查看和监控集群 1101
查看集群状态和详细信息 1102
增强型步骤调试 1108
查看应用程序历史记录。 1110
查看日志文件 1120
查看 Amazon EC2 中的集群实例 1124
CloudWatch 事件和指标 1125
使用 Ganglia 查看集群应用程序指标 1175
在 Amazon CloudTrail 中记录 Amazon EMR API 调用 1175

使用集群扩展 1178
注意事项 1178
托管扩展 1179
使用自定义策略自动扩缩 1202
调整正在运行的集群的大小 1213
预调配超时 1220
集群缩减 1225
终止集群 1228
通过控制台终止 1229
通过 CLI 终止 1230
通过 API 终止 1231
克隆集群 1231
通过 Amazon Data Pipeline 自动处理定期集群 1233
集群问题排查 1234
故障排除工具 1234
查看集群详细信息 1235
查看错误详细信息 1235
运行脚本和配置流程 1235
查看日志文件 1236
监控集群的性能 1236
查看和重新启动进程 1237
查看正在运行的进程 1237
停止并重新启动进程 1239
常见错误 1241
错误代码 1242
资源错误 1254
输入与输出错误 1262
权限错误 1265
Hive 集群错误 1266
VPC 错误 1267
流式集群错误 1271
自定义 JAR 集群错误 1272
Amazon GovCloud (US-West) 错误 1273
查找缺少的集群 1273
排查失败集群的问题 1274
步骤 1：收集有关此问题的数据 1274

步骤 2：检查环境 1275
步骤 3：查看最后一次状态更改 1276
步骤 4：检查日志文件 1276
步骤 5：分步测试集群 1278
排查速度缓慢的集群问题 1278
步骤 1：收集有关此问题的数据 1279
步骤 2：检查环境 1279
步骤 3：检查日志文件 1280
步骤 4：检查集群和实例运行状况 1282
步骤 5：检查已暂停的组 1283
步骤 6：审查配置设置 1284
步骤 7：检查输入数据 1286
对 Lake Formation 集群进行故障排查 1286
不允许访问数据湖 1286
会话过期 1286
用户在所请求表上没有权限 1287
查询与 Lake Formation 共享的跨账户数据 1287
插入、创建和变更表 1288
编写用于启动和管理集群的应用程序 1289
端到端 Amazon EMR Java 源代码示例 1289
API 调用的常见概念 1293
Amazon EMR 的端点 1293
在 Amazon EMR 中指定集群参数 1294
Amazon EMR 中的可用区 1294
如何在 Amazon EMR 集群中使用其它文件和库 1294
使用 SDK 调用 Amazon EMR API 1295
使用 Amazon SDK for Java 创建 Amazon EMR 集群 1295
管理 Amazon EMR Service Quotas 1297
什么是 Amazon EMR Service Quotas 1298
如何管理 Amazon EMR Service Quotas 1298
何时在 CloudWatch 中设置 EMR 事件 1299
Amazon 术语表 1302

什么是 Amazon EMR？

Amazon EMR（以前称为 Amazon Elastic MapReduce）是一个托管集群平台，可简化在Amazon上运行大数据框架（如 Apache Hadoop 和 Apache Spark）的过程，以处理和分析海量数据。使用这些框架和相关的开源项目，您可以处理用于分析目的的数据和业务情报工作负载。Amazon EMR 还允许您转换大量数据并移出/移入到其它Amazon数据存储和数据库中，例如 Amazon Simple Storage Service（Amazon S3）和 Amazon DynamoDB。

如果您是首次接触 Amazon EMR 的用户，我们建议您除了本部分外还应先阅读以下各部分：

- Amazon EMR – 此服务页面提供 Amazon EMR 亮点、产品详细信息和定价信息。
- 教程：Amazon EMR 入门 – 本教程将帮助您快速开始使用 Amazon EMR。

在本章节中

- Amazon EMR 概览
- 使用 Amazon EMR 的优势
- Amazon EMR 架构概览

Amazon EMR 概览

本主题概览 Amazon EMR 集群，包括：如何向集群提供工作、数据的处理方式、集群在处理期间经历的各种状态。

本主题内容

- 了解集群和节点
- 向集群提交工作
- 处理数据
- 了解集群的生命周期

了解集群和节点

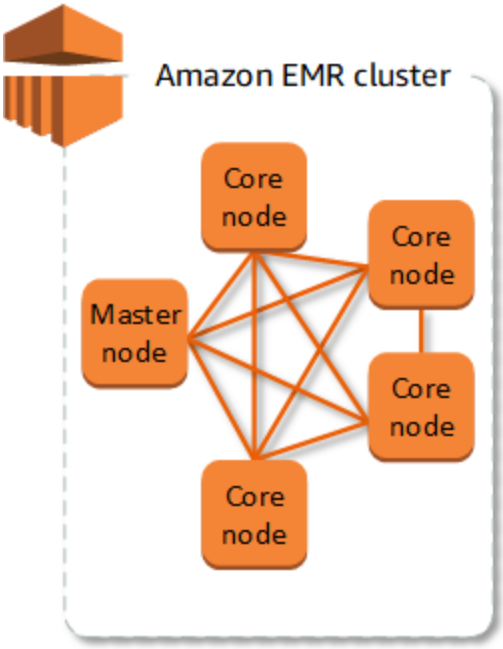
集群是 Amazon EMR 的核心组件。集群是 Amazon Elastic Compute Cloud（Amazon EC2）实例的集合。集群中的每个实例称作节点。集群中的每个节点都有一个角色，称作节点类型。Amazon EMR 还

在每个节点类型上安装不同的软件组件，在分布式应用程序（如 Apache Hadoop）中为每个节点赋予一个角色。

Amazon EMR 中的节点类型有：

- 主节点：该节点管理集群，它通过运行软件组件来协调在其他节点之间分配数据和任务的过程以便进行处理。主节点跟踪任务的状态并监控集群的运行状况。每个集群有一个主节点，并且可以创建仅包含主节点的单节点集群。
- 核心节点：该节点具有运行任务并在集群上的 Hadoop Distributed File System（HDFS）中存储数据的软件组件。多节点集群至少具有一个核心节点。
- 任务节点：该节点具有仅运行任务但不在 HDFS 中存储数据的软件组件。任务节点是可选的。

下图中的集群包含一个主节点和四个核心节点。



向集群提交工作

在 Amazon EMR 上运行集群时，可以通过几个选项指定需要完成的工作。

- 在函数中提供要完成的工作的完整定义，您可以在创建集群时将其指定为步骤。对于那些处理固定的数据量并在处理完成时终止的集群，通常会采取这种做法。
- 创建一个长时间运行的集群并使用 Amazon EMR 控制台、Amazon EMR API 或 Amazon CLI 提交包含一个或多个任务的步骤。有关更多信息，请参阅向集群提交工作。

- 创建一个集群，根据需要使用 SSH 连接到主节点和其他节点，并使用安装的应用程序提供的界面以脚本或交互方式执行任务和提交查询。有关更多信息，请参阅《Amazon EMR 版本指南》。

处理数据

启动集群时，您需要选择要安装的框架和应用程序，以满足您的数据处理需求。要处理 Amazon EMR 集群中的数据，您可以直接向已安装的应用程序提交任务或查询，或在集群中运行步骤。

直接向应用程序提交任务

您可以直接向安装在 Amazon EMR 集群中的应用程序提交任务和与之交互。为此，您通常需要通过安全连接与主节点连接，并访问可用于直接运行在集群上的软件的接口和工具。有关更多信息，请参阅连接到集群。

运行步骤以处理数据

您可以向 Amazon EMR 集群提交一个或多个有序的步骤。每个步骤都是一个工作单位，其中包含可由集群上安装的软件处理的数据操作指令。

下面是一个使用四个步骤的示例处理操作：

1. 提交要处理的输入数据集。
2. 使用 Pig 程序处理第一个步骤的输出。
3. 使用 Hive 程序处理第二个输入数据集。
4. 写入一个输出数据集。

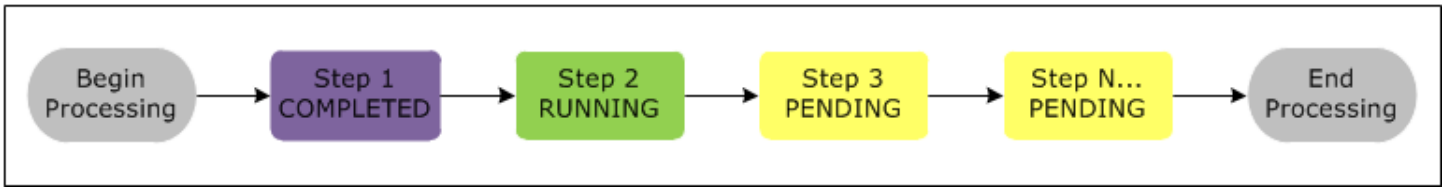
通常，在 Amazon EMR 中处理数据时，输入为以文件形式存储在您选择的底层文件系统（如 Amazon S3 或 HDFS）中的数据。数据从处理序列中的一个步骤传递到下一个。最后一步将输出数据写入指定位置，如 Amazon S3 存储桶。

步骤按下面的序列运行：

1. 提交请求以开始处理步骤。
2. 所有步骤的状态均设为 PENDING (待处理)。
3. 序列中的第一个步骤启动时，其状态更改为 RUNNING (正在运行)。其它步骤仍处于 PENDING (待处理) 状态。
4. 第一个步骤完成后，其状态更改为 COMPLETED (已完成)。

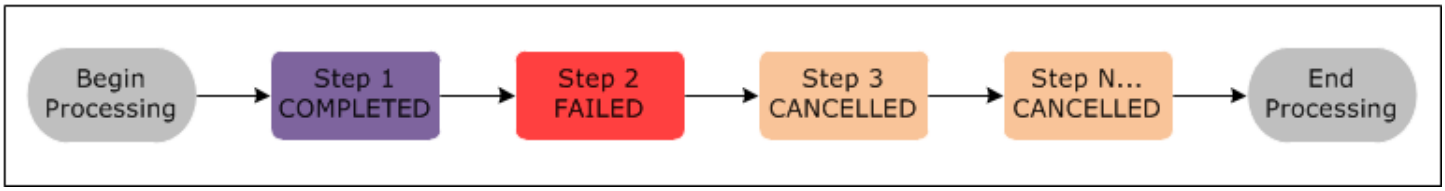
5. 序列中的下一个步骤启动，其状态更改为 RUNNING (正在运行)。完成时，其状态更改为 COMPLETED (已完成)。
6. 对每个步骤重复这一模式，直到所有步骤均完成，处理结束。

下图显示了此步骤序列及随着处理的进行各步骤的状态更改。



如果处理期间步骤失败，其状态会更改为 FAILED (失败)。您可以确定接下来如何处理每个步骤。默认情况下，序列中的任何其余步骤设置为 CANCELLED (已取消)，并且如果前面的步骤失败则不会运行。您也可以选择忽略失败并允许继续执行其余步骤，或者立即终止集群。

下图显示了此步骤序列和处理期间某个步骤失败时默认的状态变更。



了解集群的生命周期

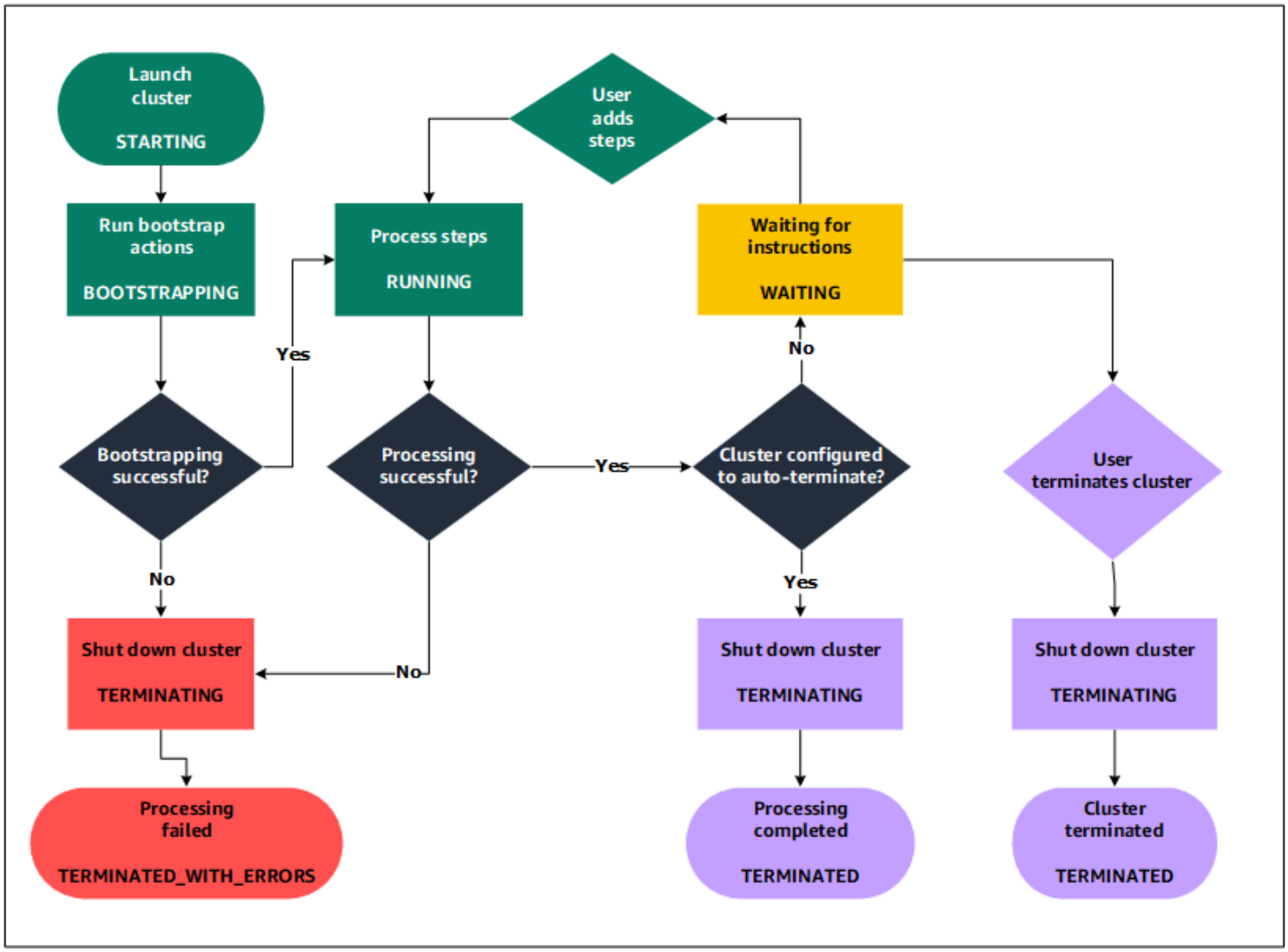
成功的 Amazon EMR 集群会遵循以下流程：

1. Amazon EMR 首先会根据您的规格，在集群中为每个实例预置 EC2 实例。有关更多信息，请参阅配置集群硬件和联网。对于所有实例，Amazon EMR 使用 Amazon EMR 的默认 AMI 或您指定的自定义 Amazon Linux AMI。有关更多信息，请参阅使用自定义 AMI。在此期间，集群的状态是 STARTING。
2. Amazon EMR 在每个实例上运行您指定的引导操作。您可以使用引导操作安装自定义应用程序并执行所需的自定义。有关更多信息，请参阅创建引导操作以安装其它软件。在此期间，集群的状态是 BOOTSTRAPPING。
3. Amazon EMR 安装在您创建集群时指定的本机应用程序，例如，Hive、Hadoop 和 Spark 等。
4. 在成功完成引导操作并安装本机应用程序后，集群状态为RUNNING。此时，您可以连接到集群实例，集群将按顺序运行在创建集群时指定的任何步骤。您可以提交额外的步骤，这些步骤在任何以前的步骤完成后运行。有关更多信息，请参阅向集群提交工作。

5. 在成功运行步骤后，集群将进入WAITING状态。如果集群配置为在完成最后一个步骤后自动终止，则会进入 TERMINATING 状态然后进入 TERMINATED 状态。如果集群配置为等待，您必须在不再需要它时手动将其关闭。手动关闭集群后，它先进入 TERMINATING 状态，再进入 TERMINATED 状态。

集群生命周期中的故障将导致 Amazon EMR 终止集群及其所有实例，除非您启用了终止保护。如果集群由于故障而终止，则会删除集群上存储的任何数据并将集群状态设置为TERMINATED_WITH_ERRORS。如果启用了终止保护，您可以从集群中检索数据，然后删除终止保护并终止集群。有关更多信息，请参阅使用终止保护。

下图显示的是集群的生命周期以及生命周期的每个阶段是如何映射到具体的集群状态的。



使用 Amazon EMR 的优势

使用 Amazon EMR 有诸多优势。本节概览这些优势并提供指向其它信息的链接以帮助您加深了解。

主题

- 节省成本
- Amazon集成
- 部署
- 可扩展性和灵活度
- 可靠性
- 安全性
- 监控
- 管理界面

节省成本

Amazon EMR 的定价取决于您部署的 Amazon EC2 实例的实例类型和数量及您启动集群的区域。按需定价提供很低的费率，但您可以通过购买预留实例或竞价型实例来进一步降低成本。Spot 实例可以显著节省成本，在某些情况下，低至按需定价的十分之一。

> **Note**
>
> 如果您为 EMR 集群使用 Amazon S3、Amazon Kinesis 或 DynamoDB，这些服务会产生额外费用 - 与您的 Amazon EMR 使用费分开计费。

> **Note**
>
> 当您在私有子网中设置 Amazon EMR 集群时，我们建议您也为 Amazon S3 设置 VPC 端点。如果您的 EMR集群处于没有适用于 Amazon S3 的 VPC 端点的私有子网中，则您需要为与 S3 流量关联的其他 NAT 网关付费，因为您的 EMR 集群与 S3 之间的流量不是位于您的 VPC 内。

有关定价选项和定价详情的更多信息，请参阅 Amazon EMR 定价。

Amazon集成

Amazon EMR 可与其它Amazon服务集成，为您的集群提供联网、存储、安全等功能。下面的列表提供了几个这种集成的示例：

- Amazon EC2 适用于包含集群中的节点的实例
- Amazon Virtual Private Cloud（Amazon VPC）用于配置在其中启动实例的虚拟网络
- Amazon S3 可存储输入和输出数据
- Amazon CloudWatch 可监控集群性能和配置警报
- Amazon Identity and Access Management（IAM）可配置权限
- Amazon CloudTrail 可审计服务请求
- Amazon Data Pipeline 可调度和启动您的集群
- Amazon Lake Formation 可在 Amazon S3 数据湖中发现、编目和保护数据

部署

您的 EMR 集群由 EC2 实例组成，这些实例执行您提交给集群的工作。当您启动集群时，Amazon EMR 使用您选择的应用程序（如 Apache Hadoop 或 Spark）配置实例。为集群选择最适合处理需求的实例大小和类型：批处理、低延迟查询、流数据或大数据存储。有关 Amazon EMR 实例类型的更多信息，请参阅配置集群硬件和联网。

Amazon EMR 提供了多种在集群上配置软件的方式。例如，您可以安装包含选定应用程序集（可以包含 Hadoop 等通用框架及 Hive、Pig、Spark 等应用程序）的 Amazon EMR 版本。您也可以安装多个 MapR 分发版之一。Amazon EMR 使用 Amazon Linux，因此，您还可以在集群上借助 yum 包管理器或从源代码手动安装软件。有关更多信息，请参阅配置集群软件。

可扩展性和灵活度

Amazon EMR 可根据您的计算需求变化灵活扩缩集群。您可以调整集群，在工作负载高峰时增加实例，在工作负载高峰过后移除实例，从而控制成本。有关更多信息，请参阅手动调整正在运行的集群的大小。

Amazon EMR 还提供了运行多个实例组的选项，因此，您可以在一个组中使用按需实例来保障处理能力，同时在另一个组中使用竞价型实例来加快任务完成速度并降低成本。您还可以混合多种实例类型以充分利用某种竞价型实例类型的定价优势。有关更多信息，请参阅什么时候应该使用竞价型实例？。

此外，Amazon EMR 还可以为您的输入、输出和中间数据灵活使用多种文件系统。例如，对于不需要在集群生命周期之外存储的处理数据，您可以选择在集群的主节点和核心节点上运行的 Hadoop Distributed File System（HDFS）。您可以选择 EMR 文件系统（EMRFS），将 Amazon S3 用作在集群上运行的应用程序的数据层，以便分离计算和存储，并在集群生命周期之外保留数据。EMRFS 具备更多优势，可供您独立扩展或收缩计算和存储需求。您可以通过调整集群来扩展计算需求，也可以通过使用 Amazon S3 来扩展存储需求。有关更多信息，请参阅使用存储和文件系统。

可靠性

Amazon EMR 能够监控集群中的节点并自动终止和替换出现故障的实例。

Amazon EMR 提供了控制集群终止方式（自动或手动）的配置选项。如果您将集群配置为自动终止，则集群会在所有步骤完成后终止。这称作暂时性集群。不过，您可以将集群配置为在处理完成后继续运行，这样，当您不再需要它时，可以选择手动终止集群。或者，您可以创建一个集群，与所安装的应用程序直接交互，然后在不再需要时手动将其终止。这些示例中的集群称作长时间运行的集群。

此外，您还可以配置终止保护，以防止集群中的实例由于处理期间出现的错误或问题而终止。如果启用终止保护，您可以在终止前从实例恢复数据。这些选项的默认设置根据您启动集群时是采用控制台、CLI 还是 API 方式而有所不同。有关更多信息，请参阅使用终止保护。

安全性

Amazon EMR 利用其它Amazon服务（如 IAM 和 Amazon VPC）和功能（如 Amazon EC2 密钥对）来帮助您保护集群和数据。

IAM

Amazon EMR 与 IAM 集成以管理权限。您可以使用 IAM policy 定义权限，并将其附加到 IAM 用户或 IAM 组。您在策略中定义的权限确定了这些用户或组成员能够执行的操作及其能够访问的资源。有关更多信息，请参阅Amazon EMR 如何与 IAM 配合使用。

此外，Amazon EMR 为 Amazon EMR 服务本身使用 IAM 角色，为实例使用 EC2 实例配置文件。这些角色授予服务和实例代表您访问其它Amazon服务的权限。Amazon EMR 服务及 EC2 实例配置文件都有默认角色。默认角色使用Amazon托管式策略，它在您首次从控制台启动 EMR 集群并选择默认权限时自动创建。您还可以从 Amazon CLI 创建默认的 IAM 角色。如果您想要代替Amazon管理权限，则可以为此服务及实例配置文件选择自定义角色。有关更多信息，请参阅为 Amazon EMR 配置 IAM 服务角色对Amazon服务和资源的权限。

安全组

Amazon EMR 使用安全组控制 EC2 实例的入站和出站流量。当您启动集群时，Amazon EMR 为您的主实例使用一个安全组，为核心/任务实例使用一个共享安全组。Amazon EMR 配置安全组规则以确保集群中实例间通信的安全。对于更高级的规则，您可以选择配置额外的安全组并将它们分配给主实例及核心/任务实例。有关更多信息，请参阅使用安全组控制网络流量。

加密

Amazon EMR 通过 EMRFS 支持可选的 Amazon S3 服务器端和客户端加密，以帮助保护您在 Amazon S3 中存储的数据的安全。采用服务器端加密时，Amazon S3 会在您上载数据后对数据进行加密。

对于客户端加密，加密和解密过程在您的 EMR 集群上的 EMRFS 客户端中进行。您使用 Amazon Key Management Service（Amazon KMS）或您自己的密钥管理系统来管理用于客户端加密的根密钥。

有关更多信息，请参阅使用 EMRFS 属性指定 Amazon S3 加密。

Amazon VPC

Amazon EMR 支持在 Amazon VPC 中的 Virtual Private Cloud（VPC）中启动集群。VPC 是Amazon 中一种隔离的虚拟网络，可提供控制高级网络配置和访问方面的功能。有关更多信息，请参阅配置联网。

Amazon CloudTrail

Amazon EMR 与 CloudTrail 集成可记录有关您的Amazon账户或代表您的账户发起的请求的信息。借助这些信息，您可以跟踪谁在何时访问了您的集群，以及他们发起请求的 IP 地址。有关更多信息，请参阅在 Amazon CloudTrail 中记录 Amazon EMR API 调用。

Amazon EC2 密钥对

您可以通过在远程电脑与主节点之间构建安全连接来监控集群并与之交互。您可以使用 Secure Shell（SSH）网络协议进行此连接或使用 Kerberos 进行身份验证。如果您使用 SSH，则 Amazon EC2 密钥对是必需的。有关更多信息，请参阅对 SSH 凭证使用 EC2 密钥对。

监控

您可以使用 Amazon EMR 管理界面和日志文件来对集群问题进行故障排查，如故障或错误。Amazon EMR 可以在 Amazon S3 中归档日志文件，因此，您可以存储日志并对问题进行问题排查（甚至在

集群终止后）。Amazon EMR 还在 Amazon EMR 控制台中提供了可选的调试工具，让您能够基于步骤、任务和任务浏览日志文件。有关更多信息，请参阅配置集群日志记录和调试。

Amazon EMR 与 CloudWatch 集成以跟踪集群及集群中任务的性能指标。您可以基于多种指标（如集群是否处于空闲状态、已使用的存储的百分比）配置警报。有关更多信息，请参阅使用 CloudWatch 指标监控 Amazon EMR 指标。

管理界面

有多种可以和 Amazon EMR 交互的方式：

- 控制台 – 图形用户界面，可用于启动和管理集群。借助这个界面，您可以填写各种 Web 窗体，指定待启动集群的详细信息、查看现有集群的详细信息、调试和终止集群。使用控制台是开始使用 Amazon EMR 的最简单的方式，您无需具备编程知识。您可以在线使用该控制台，网址为 https://console.aws.amazon.com/elasticmapreduce/home。
- Amazon Command Line Interface（Amazon CLI）– 可在本地计算机上运行的客户端应用程序，用于连接 Amazon EMR 以及创建和管理集群。Amazon CLI 包含特定于 Amazon EMR 的功能丰富的命令集。您可以使用它来编写脚本，以实现启动和管理集群的自动化。如果您更喜欢在命令行中工作，则最好的选择是使用 Amazon CLI。有关更多信息，请参阅 Amazon CLI 命令参考中的 Amazon EMR。
- 软件开发工具包 (SDK) – SDK 提供了调用 Amazon EMR 以创建和管理集群的函数。借助它们，您可以编写应用程序，用于自动处理集群的创建和管理流程。使用 SDK 是扩展或者自定义 Amazon EMR 的功能的理想选择。Amazon EMR 目前在可用于以下 SDK：Go、Java、.NET（C# 和 VB.NET）、Node.js、PHP、Python 和 Ruby。有关这些 SDK 的更多信息，请参阅适用于Amazon的工具和 Amazon EMR 示例代码和库。
- Web 服务 API – 一种低层接口，可以用来直接使用 JSON 调用 Web 服务。使用 API 是创建调用 Amazon EMR 的自定义 SDK 的最佳选择。有关更多信息，请参阅 Amazon EMR API 参考。

Amazon EMR 架构概览

Amazon EMR 服务架构包括多个层，每个层为集群提供特定的功能。本节概览这些层及每个层的组件。

本主题内容

- 存储
- 集群资源管理

- 数据处理框架
- 应用程序和项目

存储

存储层包含可用于集群的不同的文件系统。有多种不同类型的存储选项，如下所述。

Hadoop Distributed File System（HDFS）

Hadoop Distributed File System（HDFS）是一种分布式、可扩展的文件系统，供 Hadoop 使用。HDFS 将它所存储的数据在集群中的实例之间进行分配，从而在不同的实例上存储多份数据副本，确保在单个实例发生故障的情况下不会出现数据的丢失。HDFS 属于一种暂时性存储，会在集群终止时收回。HDFS 在缓存 MapReduce 处理期间的中间结果或具有大量随机 I/O 的工作负载时非常有用。

有关更多信息，请参阅本指南中的 实例存储 或访问 Apache Hadoop 网站上的 HDFS 用户指南。

EMR 文件系统 (EMRFS)

借助 EMR 文件系统（EMRFS），Amazon EMR 可使 Hadoop 具备直接访问存储在 Amazon S3 中的数据（就像使用 HDFS 文件系统时一样）的功能。在集群中，您可以将 HDFS 或 Amazon S3 用作文件系统。Amazon S3 最常用于存储 HDFS 中存储的输入和输出数据以及中间结果。

本地文件系统

本地文件系统指的是本地连接的磁盘。创建 Hadoop 集群时，会从 Amazon EC2 实例上创建各个节点，这些节点附带了预先配置的数据块，这些数据块属于称为实例存储的预先附加的磁盘存储。实例存储卷上的数据仅在 Amazon EC2 实例的生命周期内保留。

集群资源管理

资源管理层负责管理集群资源和调度作业，以进行数据处理。

默认情况下，Amazon EMR 使用 YARN（Yet Another Resource Negotiator，Apache Hadoop 2.0 中引入的一个组件）集中管理多个数据处理框架的集群资源。但是，Amazon EMR 中提供有不使用 YARN 作为资源管理器的其它框架和应用程序。Amazon EMR 还在每个节点上配备一个代理，用于管理 YARN 组件、确保集群正常运行以及与 Amazon EMR 服务进行通信。

由于竞价型实例通常用于运行任务节点，而 Amazon EMR 具有调度 YARN 任务的默认功能，因此在竞价型实例上运行的任务节点终止时，正在运行的任务不会失败。Amazon EMR 通过允许应用程序主

进程仅在核心节点上运行来实现这一目标。应用程序主进程控制正在运行的任务，并且需要在任务的整个生命周期内保持活动状态。

Amazon EMR 发行版 5.19.0 及更高版本使用内置 YARN 节点标签功能来实现这一目标。（早期版本使用代码补丁）。`yarn-site` 中的属性和 `capacity-scheduler` 配置分类是默认配置，以便 YARN capacity-scheduler 和 fair-scheduler 利用节点标注。Amazon EMR 自动为核心节点添加 `CORE` 标注，并设置属性，以便只在具有 CORE 标签的节点上安排应用程序主节点。手动修改 yarn-site 和 capacity-scheduler 配置分类中的相关属性，或直接在关联的 XML 文件中修改相关属性，可能会破坏此功能或调整此功能。

数据处理框架

数据处理框架层是用于分析和处理数据的引擎。可在 YARN 上运行并具有自己的资源管理功能的框架有很多。不同的框架适用于不同类型的处理需求，如批处理、交互式处理、内存中处理、流式处理等。框架的选择因使用案例而定。这影响到应用程序层 (用于与要处理的数据交互的层) 可用的语言和接口。Hadoop MapReduce 和 Spark 是可用于 Amazon EMR 的主处理框架。

Hadoop MapReduce

Hadoop MapReduce 是一种用于分布式计算的开源编程模型。它通过处理 (除 Map-Reduce 功能外的) 所有逻辑简化了编写平行分布式应用程序的过程。Map 函数将数据映射到一系列称为中间结果的键值对上。Reduce 函数则汇总这些中间结果、应用其它计算法并生成最终输出。有多个可用于 MapReduce 的框架，如可自动生成 Map 和 Reduce 程序的 Hive。

有关更多信息，请参阅 Apache Hadoop Wiki 网站上的 Map 和 Reduce 操作的执行原理。

Apache Spark

Spark 是一种用于处理大数据工作负载的集群框架和编程模型。与 Hadoop MapReduce 一样，Spark 是开源、分布式的处理系统，但它为执行计划使用有向无环图并为数据集使用内存缓存。在 Amazon EMR 上运行 Spark 时，您可以使用 EMRFS 直接访问 Amazon S3 中的数据。Spark 支持多种交互式查询模块，如 SparkSQL。

有关更多信息，请参阅《Amazon EMR 版本指南》中的Amazon EMR 集群上的 Apache Spark。

应用程序和项目

Amazon EMR 支持许多应用程序，如 Hive、Pig 和 Spark Streaming 库，以提供使用更高级的语言创建处理工作负载、运用机器学习算法、制作流处理应用程序、构建数据仓库等功能。此外，Amazon EMR 还支持拥有自己的集群管理功能而不使用 YARN 的开源项目。

您可以使用多种库和语言与在 Amazon EMR 中运行的应用程序交互。例如，您可以使用 Java、Hive、Pig with MapReduce 或 Spark Streaming、Spark SQL、MLlib 和 GraphX with Spark。

有关更多信息，请参阅《Amazon EMR 版本指南》。

设置 Amazon EMR

首次启动 Amazon EMR 集群之前，请完成本节中的任务：

首次使用 Amazon EMR 前，请完成以下任务：

注册一个 Amazon Web Services 账户

如果您还没有 Amazon Web Services 账户，请完成以下步骤来创建一个。

注册 Amazon Web Services 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

   当您注册 Amazon Web Services 账户 时，系统将会创建一个 Amazon Web Services 账户根用户。根用户有权访问该账户中的所有 Amazon Web Services和资源。作为安全最佳实践，请 为管理用户分配管理访问权限，并且只使用根用户执行 需要根用户访问权限的任务。

Amazon注册过程完成后，会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 https://aws.amazon.com/ 并选择 My Account (我的账户) 来查看当前的账户活动并管理您的账户。

保护 IAM 用户

注册 Amazon Web Services 账户 后，启用多重身份验证（MFA）保护您的管理用户。有关说明，请参阅 IAM 用户指南中的为 为 IAM 用户（控制台）启用虚拟 MFA 设备。

要授予其他用户访问您的 Amazon Web Services 账户 资源的权限，请创建 IAM 用户。为了保护您的 IAM 用户，请启用 MFA 并仅向 IAM 用户授予执行任务所需的权限。

有关创建和保护 IAM 用户的更多信息，请参阅 IAM 用户指南中的以下主题：

- 在您的 Amazon Web Services 账户 中创建 IAM 用户
- 适用于 Amazon 资源的访问管理
- IAM 基于身份的策略示例

为 SSH 创建 Amazon EC2 密钥对

> **Note**
>
> 使用 Amazon EMR 发行版本 5.10.0 或更高版本，可以配置 Kerberos 来对用户和到集群的 SSH 连接进行身份验证。有关更多信息，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。

要使用 Secure Shell（SSH）协议通过安全通道进行身份验证并连接到集群中的节点，请在启动集群之前创建 Amazon Elastic Compute Cloud（Amazon EC2）密钥对。您也可以在没有密钥对的情况下创建集群。这通常适用于瞬态集群：这种集群启动、运行步骤、然后自动终止。

如果...	则...
您已经拥有要使用的 Amazon EC2 密钥对，或者您不需要对集群进行身份验证。	跳过此步骤。
您需要创建密钥对。	请参阅使用 Amazon EC2 创建密钥对。

后续步骤

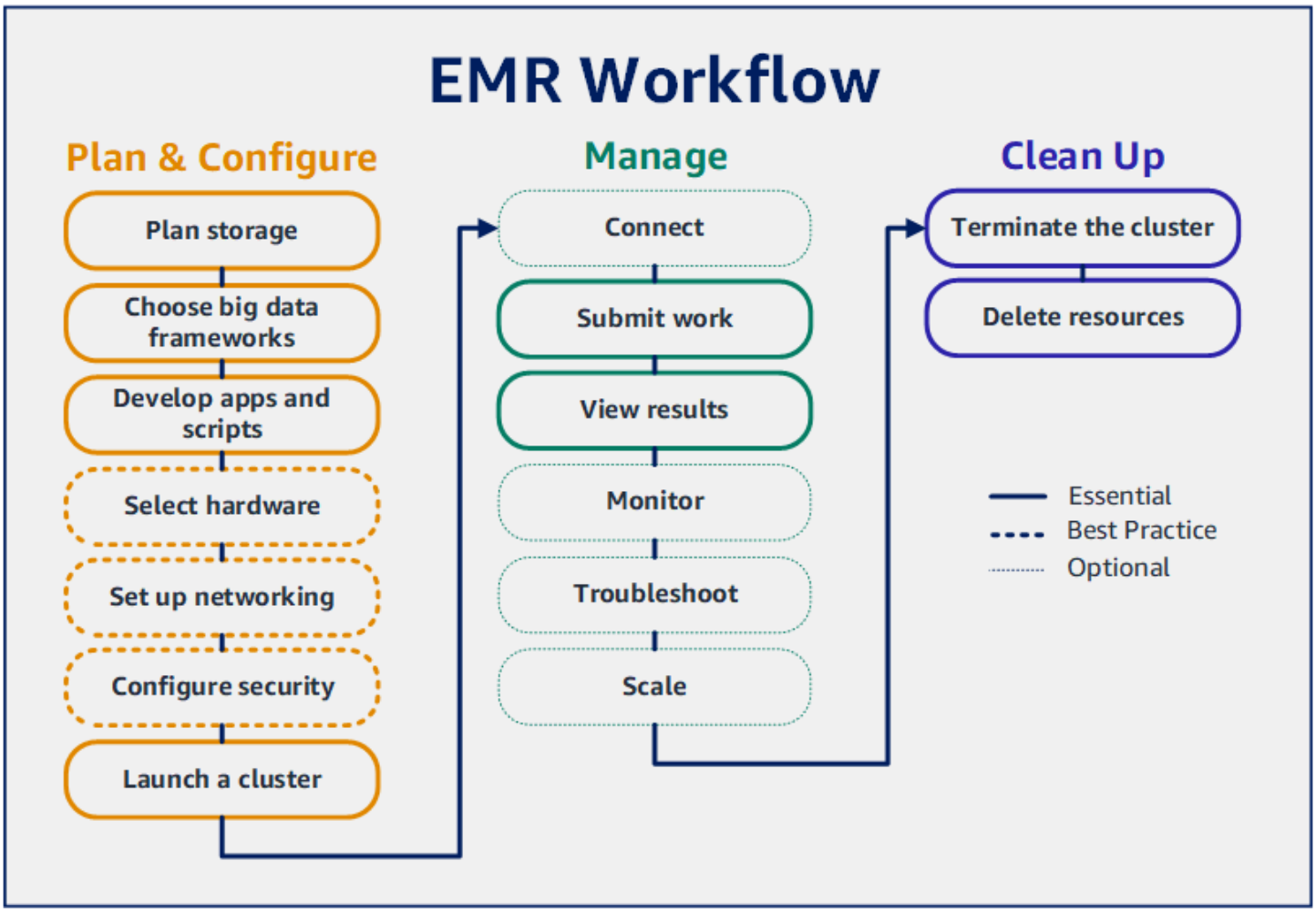
- 有关创建示例集群的指南，请参阅教程：Amazon EMR 入门。
- 有关如何配置自定义集群并控制其访问权限的更多信息，请参阅计划和配置集群和Amazon EMR 中的安全性。

教程：Amazon EMR 入门

概述

借助 Amazon EMR，您可以设置集群以便在几分钟内使用大数据框架处理和分析数据。本教程介绍如何使用 Spark 启动示例集群，以及如何运行存储在 Amazon S3 存储桶中的简单 PySpark 脚本。它涵盖了三个主要工作流类别中的基本 Amazon EMR 任务：计划和配置、管理以及清除。

在学习本教程时，您可以找到指向更详细主题的链接，以及 后续步骤 部分中其他步骤的想法。如果您有任何疑问或遇到困难，请在我们的论坛上联系 Amazon EMR 团队。



先决条件

- 在启动 Amazon EMR 集群之前，请确保您已完成 设置 Amazon EMR 中的任务。

成本

- 您创建的示例集群将在实际环境中运行。该集群产生最低费用。为避免额外费用，请确保您完成本教程最后一步中的清理任务。根据 Amazon EMR 定价，费用按秒计算。费用也因区域而有所不同。有关更多信息，请参阅 Amazon EMR 定价。
- 对于您在 Amazon S3 中存储的小文件，也可能会产生最低费用。如果您在 Amazon 免费套餐的使用限制内，则可能会免除 Amazon S3 所产生的部分或全部费用。有关更多信息，请参阅 Amazon S3 定价和Amazon免费套餐。

第 1 步：计划和配置 Amazon EMR 集群

为 Amazon EMR 准备存储

当您使用 Amazon EMR 时，您可以从各种文件系统中进行选择，以存储输入数据、输出数据和日志文件。在本教程，您可以使用 EMRFS 将数据存储在 S3 存储桶中。EMRFS 是 Hadoop 文件系统的一种实现方式，允许您读取常规文件并将其写入到 Amazon S3。有关更多信息，请参阅使用存储和文件系统。

要为本教程创建存储桶，请参照《Amazon Simple Storage Service 控制台用户指南》中的如何创建 S3 存储桶？。在您计划启动 Amazon EMR 集群的同一Amazon区域中创建存储桶。例如，美国西部（俄勒冈）us-west-2。

您用于 Amazon EMR 的存储桶和文件夹具有以下限制：

- 名称由小写字母、数字、句点 (.) 和连字符 (-) 组成。
- 名称不能以数字结尾。
- 对于所有Amazon账户，存储桶名称必须是唯一的。
- 输出文件夹必须为空。

为 Amazon EMR 准备含有输入数据的应用程序

为 Amazon EMR 准备应用程序的最常见方法是将应用程序及其输入数据上传至 Amazon S3。然后，您在向集群提交工作时，指定脚本和数据的 Amazon S3 位置。

在此步骤中，您将示例 PySpark 脚本上传到 Amazon S3 存储桶。我们提供了 PySpark 脚本供您使用。该脚本处理食品企业检查数据并在 S3 存储桶中返回结果文件。结果文件列出了红色类违规最多的十大企业。

您还可以将示例输入数据上载到 Amazon S3 以供 PySpark 脚本处理。输入数据是修正版，其中包含 2006 年至 2020 年华盛顿州金县卫生部门的检查结果。有关更多信息，请参阅 King County Open Data: Food Establishment Inspection Data。以下是数据集中的示例行。

```
name, inspection_result, inspection_closed_business, violation_type, violation_points
100 LB CLAM, Unsatisfactory, FALSE, BLUE, 5
100 PERCENT NUTRICION, Unsatisfactory, FALSE, BLUE, 5
7-ELEVEN #2361-39423A, Complete, FALSE, , 0
```

为 EMR 准备示例 PySpark 脚本

1. 将下面的示例代码复制到您选择的编辑器中的新文件中。

```
import argparse

from pyspark.sql import SparkSession

def calculate_red_violations(data_source, output_uri):
    """
    Processes sample food establishment inspection data and queries the data to
 find the top 10 establishments
    with the most Red violations from 2006 to 2020.

    :param data_source: The URI of your food establishment data CSV, such as 's3://
DOC-EXAMPLE-BUCKET/food-establishment-data.csv'.
    :param output_uri: The URI where output is written, such as 's3://DOC-EXAMPLE-
BUCKET/restaurant_violation_results'.
    """
    with SparkSession.builder.appName("Calculate Red Health
 Violations").getOrCreate() as spark:
        # Load the restaurant violation CSV data
        if data_source is not None:
            restaurants_df = spark.read.option("header", "true").csv(data_source)

        # Create an in-memory DataFrame to query
        restaurants_df.createOrReplaceTempView("restaurant_violations")

        # Create a DataFrame of the top 10 restaurants with the most Red violations
        top_red_violation_restaurants = spark.sql("""SELECT name, count(*) AS
 total_red_violations
          FROM restaurant_violations
          WHERE violation_type = 'RED'
```

```
            GROUP BY name
            ORDER BY total_red_violations DESC LIMIT 10""")

        # Write the results to the specified output URI
        top_red_violation_restaurants.write.option("header",
 "true").mode("overwrite").csv(output_uri)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '--data_source', help="The URI for you CSV restaurant data, like an S3
 bucket location.")
    parser.add_argument(
        '--output_uri', help="The URI where output is saved, like an S3 bucket
 location.")
    args = parser.parse_args()

    calculate_red_violations(args.data_source, args.output_uri)
```

2. 将该文件保存为 `health_violations.py`。
3. 将 `health_violations.py` 上传到您为本教程创建的 Amazon S3 存储桶中。有关说明，请参阅 Amazon Simple Storage Service 入门指南中的将对象上传到存储桶。

为 EMR 准备示例输入数据

1. 下载 zip 文件 food_establishment_data.zip。
2. 解压并保存 `food_establishment_data.zip` 至 计算机的 `food_establishment_data.csv`。
3. 将 CSV 文件上载到您为本教程创建的 S3 存储桶。有关说明，请参阅 Amazon Simple Storage Service 入门指南中的将对象上传到存储桶。

有关为 EMR 设置数据的更多信息，请参阅准备输入数据。

启动 Amazon EMR 集群

您在准备存储位置和应用程序之后，则可以启动示例 Amazon EMR 集群。在此步骤，您使用最新的 Amazon EMR 版本启动 Apache Spark 集群。

New console

使用新控制台启动安装有 Spark 的集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Create Cluster（创建集群）页面上，记下 Release（发行版）、Instance type（实例类型）、Number of instances（示例数量）和 Permissions（权限）的默认值。这些字段使用适用于通用型集群的值自动填充。
4. 在 Cluster name（集群名称）字段中，输入唯一的集群名称以帮助您识别集群，例如#######。
5. 在 Applications (应用程序) 下，选择 Spark 选项以在集群上安装 Spark。

   > **Note**
   >
   > 在启动集群之前，选择要在 Amazon EMR 集群上运行的应用程序。在启动集群之后，您无法在集群中添加或删除应用程序。

6. 在 Cluster logs（集群日志）下，选中 Publish cluster-specific logs to Amazon S3（将集群特定日志发布到 Amazon S3）复选框。将 Amazon S3 location（Amazon S3 位置）值替换为您创建的 Amazon S3 存储桶，后跟 **/logs**。例如，**s3://DOC-EXAMPLE-BUCKET/logs**。添加 **/logs** 会在存储桶中创建一个名为“logs”的新文件夹，Amazon EMR 可以从中复制集群的日志文件。
7. 在 Security configuration and permissions（安全配置和权限）下，选择您的 EC2 key pair（EC2 密钥对）。在同一部分中，选择 Service role for Amazon EMR（适用于 Amazon EMR 的服务角色）下拉菜单，然后选择 EMR_DefaultRole。然后，选择 IAM role for instance profile（实例配置文件的 IAM 角色）下拉菜单并选择 EMR_EC2_DefaultRole。
8. 选择 Create cluster（创建集群）以启动集群并打开集群详细信息页面。
9. 找到集群名称旁边的集群 Status (状态)。Amazon EMR 预制集群过程中，状态会从 Starting (正在启动) 变为 Running (正在运行) 再变为 Waiting (正在等待)。您可能需要选择右侧的刷新图标或刷新您的浏览器才能查看状态更新情况。

集群已启动且正在运行，并准备好接受工作时，集群状态会变更为 Waiting (正在等待)。有关读取集群摘要的更多信息，请参阅查看集群状态和详细信息。有关集群状态的信息，请参阅了解集群的生命周期。

Old console

使用旧控制台启动安装有 Spark 的集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Create cluster (创建集群) 以打开 Quick Options wizard (快速选项向导)。
3. 在 Create Cluster - Quick Options（创建集群 - 快速选项）页面上，记下 Release（发行版）、Instance type（实例类型）、Number of instances（实例数量）和 Permissions（权限）的原定设置值。这些字段使用为通用型集群操作的值自动填充。
4. 输入 Cluster name (集群名称) 以帮助您标识集群。例如#######。
5. 启用 Logging (日志记录)，但将 S3 folder (S3 文件夹) 的值替换为您创建的 Amazon S3 存储桶，后跟 **/logs**。例如，**s3://DOC-EXAMPLE-BUCKET/logs**。在存储桶中创建名为“logs”的**/logs**，EMR 可以从中复制集群的日志文件。
6. 在 Applications (应用程序) 下，选择 Spark 选项以在集群上安装 Spark。

   > **Note**
   >
   > 在启动集群之前，选择要在 Amazon EMR 集群上运行的应用程序。在启动集群之后，您无法在集群中添加或删除应用程序。

7. 在 Security and access (安全与访问) 选择 EC2 key pair (EC2 密钥对)。
8. 选择 Create cluster (创建集群)，启动集群并打开集群状态页面。
9. 找到集群名称旁边的集群 Status (状态)。Amazon EMR 预制集群过程中，状态会从 Starting (正在启动) 变为 Running (正在运行) 再变为 Waiting (正在等待)。您可能需要选择右侧的刷新图标或刷新您的浏览器才能查看状态更新情况。

集群已启动且正在运行，并准备好接受工作时，集群状态会变更为 Waiting (正在等待)。有关读取集群摘要的更多信息，请参阅查看集群状态和详细信息。有关集群状态的信息，请参阅了解集群的生命周期。

CLI

使用 Amazon CLI 启动安装有 Spark 的集群

1. 创建 IAM 原定设置角色，然后您可以使用该角色通过以下命令创建集群。

```
aws emr create-default-roles
```

有关 create-default-roles 的更多信息，请参阅 Amazon CLI 命令参考。

2. 使用以下命令创建 Spark 集群。使用 --name 选项为您的集群输入一个名称，并使用 --ec2-attributes 选项指定您的 EC2 密钥对的名称。

```
aws emr create-cluster \
--name "<My First EMR Cluster>" \
--release-label <emr-5.36.1> \
--applications Name=Spark \
--ec2-attributes KeyName=<myEMRKeyPairName> \
--instance-type m5.xlarge \
--instance-count 3 \
--use-default-roles
```

请注意 --instance-type、--instance-count 和 --use-default-roles 的其它必需值。这些值是为通用型集群选择的。有关 create-cluster 的更多信息，请参阅 Amazon CLI 命令参考。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字符 (^)。

将会看到类似下面的输出。输出将显示新集群的ClusterId和ClusterArn。记下您的ClusterId。您可以使用ClusterId检查集群状态并提交工作。

```
{
    "ClusterId": "myClusterId",
    "ClusterArn": "myClusterArn"
}
```

3. 使用以下命令检查集群状态。

```
aws emr describe-cluster --cluster-id <myClusterId>
```

您应看到类似下面的输出，可以使用Status对象适用于新集群。

```
{
    "Cluster": {
        "Id": "myClusterId",
        "Name": "My First EMR Cluster",
        "Status": {
            "State": "STARTING",
            "StateChangeReason": {
                "Message": "Configuring cluster software"
            }
        }
     }
}
```

在 Amazon EMR 预置集群时，State值从STARTING变为RUNNING，再变为WAITING。

集群已启动且正在运行，并准备好接受工作时，集群状态会变更为 **WAITING**。有关集群状态的信息，请参阅 了解集群的生命周期。

第 2 步：管理 Amazon EMR 集群

将工作提交到 Amazon EMR

启动集群后，可以向正在运行的集群提交工作，从而处理分析数据。提交工作至 Amazon EMR 集群，作为 step (步骤)。步骤是由一个或多个任务组成的工作单位。例如，您可以提交一个步骤来计算值，或传输和处理数据。可以在创建集群时提交步骤，也可以将步骤提交到正在运行的集群。教程这部分内容中，您向正在运行的集群提交health_violations.py步骤。要了解有关步骤的更多信息，请参阅向集群提交工作。

New console

使用新控制台以步骤的形式提交 Spark 应用程序

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要提交工作的集群。集群状态必须为 Waiting (正在等待)。
3. 选择 Steps（步骤）选项卡，然后选择 Add step（添加步骤）。
4. 根据以下准则配置步骤：

   - 对于 Type（类型），选择 Spark application（Spark 应用程序）。您应该会看到 Deploy Mode（部署模式）、Application location（应用程序位置）和 Spark-submit options（Spark-submit 选项）的其他字段。
   - 对于 Name（名称），输入新名称。如果您在集群中有很多步骤，命名每个步骤有助于您跟踪它们。
   - 对于 Deploy mode（部署模式），保留默认值 Cluster mode（集群模式）。有关 Spark 部署模式的更多信息，请参阅 Apache Spark 文档中的 Cluster mode overview（集群模式概览）。
   - 对于 Application location（应用程序位置），输入 Amazon S3 中 `health_violations.py` 脚本的位置，例如 *`s3://DOC-EXAMPLE-BUCKET/health_violations.py`*。
   - Spark-submit options（Spark-submit 选项）字段留空。有关 `spark-submit` 选项的更多信息，请参阅 Launching applications with spark-submit（使用 spark-submit 启动应用程序）。
   - 在 Arguments (参数) 字段中，输入以下参数和值：

     ```
     --data_source s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv
     --output_uri s3://DOC-EXAMPLE-BUCKET/myOutputFolder
     ```

     将 *`s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv`* 替换为您在 为 Amazon EMR 准备含有输入数据的应用程序 中准备的输入数据的 S3 存储桶 URI。

     将 *`DOC-EXAMPLE-BUCKET`* 替换为您为本教程创建的存储桶名称，并将 *`myOutputFolder`* 替换为您集群输出文件夹的名称。

- 对于 Action if step fails（步骤失败时的操作），接受默认选项 Continue（继续）。这样，如果步骤失败，集群将继续运行。

5. 选择 Add (添加) 以提交步骤。该步骤应出现在控制台中，状态为 Pending (待处理)。
6. 监控步骤状态。状态将从 Pending（待处理）变为 Running（正在运行）再变为 Completed（已完成）。要更新控制台中的状态，请选择 Filter（筛选条件）右侧的刷新图标。运行该脚本大约需要一分钟时间。步骤成功完成后，该状态将变为 Completed（已完成）。

Old console

使用旧控制台以步骤的形式提交 Spark 应用程序

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 在 Cluster List (集群列表) 中，选择集群名称。集群状态必须为 Waiting (正在等待)。
3. 选择 Steps (步骤)，然后选择 Add step (添加步骤)。
4. 根据以下准则配置步骤：

   - 对于 Step type (步骤类型)，选择 Spark application (Spark 应用程序)。您应该会看到 Deploy Mode (部署模式)、Spark-submit options (Spark-submit 选项) 和 Application location (应用程序位置) 的其它字段显示。
   - 对于 Name (名称)，可以保留默认值或键入新名称。如果您在集群中有很多步骤，命名每个步骤有助于您跟踪它们。
   - 对于 Deploy mode (部署模式)，保留默认值 Cluster (集群)。有关 Spark 部署模式的更多信息，请参阅 Apache Spark 文档中的集群模式概览。
   - 将 Spark-submit options (Spark-submit 选项) 字段留空。有关 `spark-submit` 选项的更多信息，请参阅使用 spark-submit 启动应用程序。
   - 对于 Application location (应用程序位置)，输入您的 `health_violations.py` 脚本在 Amazon S3 中的位置。例如，*`s3://DOC-EXAMPLE-BUCKET/health_violations.py`*。
   - 在 Arguments (参数) 字段中，输入以下参数和值：

     ```
     --data_source s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv
     --output_uri s3://DOC-EXAMPLE-BUCKET/myOutputFolder
     ```

将 *s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv* 替换为您在 为 Amazon EMR 准备含有输入数据的应用程序 中准备的输入数据的 S3 URI。

将 *DOC-EXAMPLE-BUCKET* 替换为您为本教程创建的存储桶的名称，并将 *myOutputFolder* 替换为您的集群输出文件夹的名称。

- 对于 Action on failure (出现故障时的操作)，保留默认选项 Continue (继续)，以使集群能够在步骤失败时继续运行。

5. 选择 Add (添加) 以提交步骤。该步骤应出现在控制台中，状态为 Pending (待处理)。
6. 检查步骤状态从 Pending (待处理) 变为Running (正在运行) 再变为Completed (已完成)。要更新控制台中的状态，请选择 Filter (筛选条件) 右侧的刷新图标。运行该脚本大约需要一分钟时间。

当状态更改为 Completed (已完成) 时，您将知道该步骤已成功完成。

CLI

使用 Amazon CLI 以步骤的形式提交 Spark 应用程序

1. 确保您拥有在 启动 Amazon EMR 集群 中启动的集群的 `ClusterId`。您还可以使用以下命令检索集群 ID。

```
aws emr list-clusters --cluster-states WAITING
```

2. 通过 `add-steps` 命令和 `ClusterId`，将 `health_violations.py` 以步骤的形式提交。

- 您可以通过替换 *"My Spark Application"* 为您的步骤指定一个名称。在 `Args` 数组中，将 *s3://DOC-EXAMPLE-BUCKET/health_violations.py* 替换为您的 `health_violations.py` 应用程序的位置。
- 将 *s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv* 替换为您的 `food_establishment_data.csv` 数据集的 S3 位置。
- 将 *s3://DOC-EXAMPLE-BUCKET/MyOutputFolder* 替换为您指定存储桶的 S3 路径和您的集群输出文件夹的名称。
- `ActionOnFailure=CONTINUE` 表示如果步骤失败，集群将继续运行。

```
aws emr add-steps \
--cluster-id <myClusterId> \
```

```
--steps Type=Spark,Name="<My Spark
 Application>",ActionOnFailure=CONTINUE,Args=[<s3://DOC-EXAMPLE-
BUCKET/health_violations.py>,--data_source,<s3://DOC-EXAMPLE-BUCKET/
food_establishment_data.csv>,--output_uri,<s3://DOC-EXAMPLE-BUCKET/
MyOutputFolder>]
```

有关使用 CLI 提交步骤的更多信息，请参阅 Amazon CLI 命令参考。

提交步骤后，您应该会看到包含 StepIds 的列表的输出，类似于下列内容。由于您提交了一个步骤，因此列表中只有一个 ID。复制步骤 ID。使用步骤 ID，检查步骤状态。

```
{
    "StepIds": [
        "s-1XXXXXXXXXXA"
    ]
}
```

3. 使用带有 describe-step 命令的步骤，查询步骤的状态。

```
aws emr describe-step --cluster-id <myClusterId> --step-id <s-1XXXXXXXXXXA>
```

您应该会看到包含步骤相关内容的输出，类似于以下内容。

```
{
    "Step": {
        "Id": "s-1XXXXXXXXXXA",
        "Name": "My Spark Application",
        "Config": {
            "Jar": "command-runner.jar",
            "Properties": {},
            "Args": [
                "spark-submit",
                "s3://DOC-EXAMPLE-BUCKET/health_violations.py",
                "--data_source",
                "s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv",
                "--output_uri",
                "s3://DOC-EXAMPLE-BUCKET/myOutputFolder"
            ]
        },
        "ActionOnFailure": "CONTINUE",
        "Status": {
```

```
            "State": "COMPLETED"
        }
    }
}
```

随着步骤的运行，步骤的 State 从 PENDING 变为 RUNNING，再变为 COMPLETED。运行该步骤大约需要一分钟，因此您可能需要多次检查状态。

当 State 更改为 **COMPLETED** 时，您将知道该步骤已成功。

有关步骤生命周期的更多信息，请参阅运行步骤以处理数据。

查看结果

步骤成功运行后，您可以在 Amazon S3 输出文件夹中查看其输出结果。

查看 health_violations.py 的结果

1. 通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 选择 Bucket name (存储桶名称)，然后选择您在提交步骤时指定的输出文件夹。例如，选择 *DOC-EXAMPLE-BUCKET* 然后选择 *myOutputFolder*。
3. 验证以下项目是否位于输出文件夹中：
   - 称为 _SUCCESS 的小格式对象。
   - 以前缀 part- 开头的 CSV 文件，包含结果。
4. 选择包含结果的对象，然后选择 Download (下载) 以将结果保存到本地文件系统。
5. 在选定编辑器中打开结果。输出文件列出了红色违规最多的十大食品企业。输出文件还显示每个机构的红色违规总数。

   以下是 health_violations.py 结果的示例。

```
name, total_red_violations
SUBWAY, 322
T-MOBILE PARK, 315
WHOLE FOODS MARKET, 299
PCC COMMUNITY MARKETS, 251
TACO TIME, 240
MCDONALD'S, 177
```

```
THAI GINGER, 153
SAFEWAY INC #1508, 143
TAQUERIA EL RINCONSITO, 134
HIMITSU TERIYAKI, 128
```

有关 Amazon EMR 集群输出的更多信息，请参阅配置输出位置。

（可选）连接到正在运行的 Amazon EMR 集群

当您使用 Amazon EMR 时，您可能希望连接到正在运行的集群，以读取日志文件、调试集群或使用类似于 Spark shell 的 CLI 工具。通过 Amazon EMR，您可以使用 Secure Shell（SSH）协议连接到集群。本部分介绍如何配置 SSH、集群连接以及查看 Spark 的日志文件。有关连接到集群的更多信息，请参阅 对 Amazon EMR 集群节点进行身份验证。

授权与集群的 SSH 连接

在连接到集群之前，您需要修改集群安全组以授权入站 SSH 连接。Amazon EC2 安全组充当虚拟防火墙以控制至您的集群的入站和出站流量。为本教程创建集群时，Amazon EMR 代表您创建了以下安全组：

ElasticMapReduce-master

与主节点关联的默认 Amazon EMR 托管安全组。在 Amazon EMR 集群中，主节点是管理集群的 Amazon EC2 实例。

ElasticMapReduce-slave

与核心和任务节点关联的默认安全组。

New console

使用新控制台允许主要安全组的可信来源进行 SSH 访问

要编辑安全组，您必须具有权限来管理集群所在 VPC 的安全组。有关更多信息，请参阅《IAM 用户指南》中的更改用户的权限和允许管理 EC2 安全组的示例策略。

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。集群详细信息页面随即打开。应预先选择此页面上的 Properties（属性）选项卡。

3. 在 Properties（属性）选项卡的 Networking（联网）下，选择 EC2 security groups (firewall) [EC2 安全组（防火墙）] 旁边的箭头以展开此部分。在 Primary node（主节点）下，选择安全组链接。完成以下步骤后，可以选择返回此步骤，选择 Core and task nodes（核心节点和任务节点），然后重复以下步骤以允许 SSH 客户端访问核心节点和任务节点。
4. 此操作将打开 EC2 控制台。选择 Inbound rules（入站规则）选项卡，然后选择 Edit inbound rules（编辑入站规则）。
5. 通过以下设置检查允许公有访问的入站规则。如果存在，请选择 Delete (删除) 以将其删除。

   - Type

     SSH
   - 端口

     22
   - 源

     自定义 0.0.0.0/0

   > **Warning**
   >
   > 2020 年 12 月之前，ElasticMapRemoe-Master 安全组有一个预先配置的规则，允许来自所有来源的端口 22 入站流量。此规则的创建可以简化与主节点 (master node) 的初始 SSH 客户端连接。我们强烈建议您移除此入站规则，限制流量进入可信来源。

6. 滚动到规则列表的底部并选择 Add Rule (添加规则)。
7. 对于Type (类型)，选择 SSH。选择 SSH，自动为 Protocol (协议) 输入 TCP，为 Port Range (端口范围) 输入 22。
8. 针对源，选择 My IP（我的 IP），自动添加您的 IP 地址作为源地址。您也可以添加一系列 Custom (自定义) 可信客户端 IP 地址，或创建针对其他客户端的其他规则。许多网络环境动态分配 IP 地址，因此您以后可能需要更新受信任客户端的 IP 地址。
9. 选择 Save（保存）。
10. 或者，从列表中选择 Core and task nodes（核心节点和任务节点）并重复上述步骤，以允许 SSH 客户端访问核心节点和任务节点。

Old console

使用旧控制台向可信源 SSH 授予主要安全组访问权限

要编辑安全组，您必须具有权限来管理集群所在 VPC 的安全组。有关更多信息，请参阅《IAM 用户指南》中的更改用户的权限和允许管理 EC2 安全组的示例策略。

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Clusters (集群)。选择您要修改的集群的 Name（名称）。
3. 在 Security and access (安全与访问) 下，选择 Security groups for Master (主节点的安全组) 链接。
4. 从列表中选择 ElasticMapReduce-master。
5. 选择 Inbound rules（入站规则）选项卡，然后选择 Edit inbound rules（编辑入站规则）。
6. 通过以下设置检查允许公有访问的入站规则。如果存在，请选择 Delete (删除) 以将其删除。

   - Type

     SSH
   - 端口

     22
   - 源

     自定义 0.0.0.0/0

   > **Warning**
   >
   > 2020 年 12 月之前，ElasticMapRemoe-Master 安全组有一个预先配置的规则，允许来自所有来源的端口 22 入站流量。创建此规则可简化与主节点的初始 SSH 客户端连接。我们强烈建议您移除此入站规则，限制流量进入可信来源。

7. 滚动到规则列表的底部并选择 Add Rule (添加规则)。
8. 对于Type (类型)，选择 SSH。

   选择 SSH，自动为 Protocol (协议) 输入 TCP，为 Port Range (端口范围) 输入 22。

9. 针对源，选择 My IP（我的 IP），自动添加您的 IP 地址作为源地址。您也可以添加一系列 Custom (自定义) 可信客户端 IP 地址，或创建针对其他客户端的其他规则。许多网络环境动态分配 IP 地址，因此您以后可能需要更新受信任客户端的 IP 地址。
10. 选择 Save（保存）。
11. 或者，从列表中选择 ElasticMapReduce-slave 并重复上述步骤，以允许对核心和任务节点执行 SSH 客户端访问。

使用 Amazon CLI 连接到您的集群。

无论您的操作系统如何，都可以使用 Amazon CLI 创建 SSH 连接到集群。

要使用 Amazon CLI 连接到集群并查看日志文件

1. 使用以下命令开启与集群的 SSH 连接。将 *<mykeypair.key>* 替换为密钥对的完整路径和文件名。例如，`C:\Users\<username>\.ssh\mykeypair.pem`。

```
aws emr ssh --cluster-id <j-2AL4XXXXXX5T9> --key-pair-file <~/mykeypair.key>
```

2. 导航到 `/mnt/var/log/spark` 访问集群主节点 (master node) 上的 Spark 日志。然后查看该位置的文件。有关主节点 (master node) 上其他日志文件的列表，请参阅 查看主节点上的日志文件。

```
cd /mnt/var/log/spark
ls
```

第 3 步：清除 Amazon EMR 资源

终止集群

现在，您已向集群提交工作并查看了 PySpark 应用程序的结果，因此可以终止集群。终止集群会停止所有集群关联的 Amazon EMR 费用和 Amazon EC2 实例。

在终止集群后，Amazon EMR 将有关集群的元数据免费保留两个月。归档元数据有助于为新任务克隆集群 (clone the cluster) 或为了参考的目的重新访问集群配置。元数据不包括集群写入 S3 的数据，或存储在集群上的 HDFS 中的数据。

> **Note**
>
> Amazon EMR 控制台不允许您在关闭集群后从列表视图中终止集群。当 Amazon EMR 清除其元数据时，终止的集群将从控制台消失。

New console

使用新控制台终止集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 选择 Clusters（集群），然后选择要终止的集群。
3. 在 Actions（操作）下拉菜单下，选择 Terminate cluster（终止集群）。
4. 在对话框中选择 Terminate（终止）。根据集群配置，终止可能需要 5 至 10 分钟。有关如何终止 Amazon EMR 集群的更多信息，请参阅 终止集群。

Old console

使用旧控制台终止集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Clusters (集群)，然后选择要终止的集群。例如，*##### EMR ##*。
3. 选择 Terminate (终止) 以打开 Terminate cluster (终止集群) 提示。
4. 在打开提示中选择Terminate (终止)。根据集群配置，终止可能需要 5 至 10 分钟。有关终止 Amazon EMR 集群的更多信息，请参阅终止集群。

> **Note**
>
> 如果您已严格遵循本教程，则终止保护应已关闭。集群终止保护防止意外终止。如果终止保护处于开启状态，则在终止集群之前，您将看到更改此设置的提示。选择 Change (更改)，然后选择 Off (关闭)。

CLI

使用 Amazon CLI 终止集群

1. 使用以下命令启动集群终止过程。将 *<myClusterId>* 替换为示例集群 ID。命令不会返回输出。

```
aws emr terminate-clusters --cluster-ids <myClusterId>
```

2. 要检查集群终止过程是否正在进行，请使用以下命令检查集群状态。

```
aws emr describe-cluster --cluster-id <myClusterId>
```

以下是 JSON 格式的示例输出。集群 Status 应从 **TERMINATING** 更改为 **TERMINATED**。终止可能需要 5 到 10 分钟，具体取决于集群配置。有关终止 Amazon EMR 集群的更多信息，请参阅终止集群。

```
{
    "Cluster": {
        "Id": "j-xxxxxxxxxxxxx",
        "Name": "My Cluster Name",
        "Status": {
            "State": "TERMINATED",
            "StateChangeReason": {
                "Code": "USER_REQUEST",
                "Message": "Terminated by user request"
            }
        }
    }
}
```

删除 S3 资源

为避免产生额外费用，您应删除 Amazon S3 存储桶。删除存储桶意味着将删除本教程中的所有 Amazon S3 资源。您的存储桶应包含：

- PySpark 脚本
- 输入数据集
- 您的输出结果文件夹

- 您的日志文件夹

如果您将 PySpark 脚本或输出保存在不同位置，您可能需要采取额外的步骤来删除存储的文件。

> **Note**
>
> 在删除存储桶之前，必须终止集群。否则，可能不允许您清空存储桶。

要删除存储桶，请参照《Amazon Simple Storage Service 用户指南》中如何删除 S3 存储桶？的说明。

后续步骤

您现在已经从头到尾启动了第一个 Amazon EMR 集群。您还完成了基本的 EMR 任务，例如：准备和提交大数据应用程序、查看结果以及终止集群。

可以使用以下主题了解如何自定义 Amazon EMR 工作流程的更多信息。

了解 Amazon EMR 的大数据应用程序

在《Amazon EMR 版本指南》中发现并比较您可以在集群上安装的大数据应用程序。《发布指南 (Release Guide)》详细介绍了每个 EMR 发布版本，并包括使用 Amazon EMR 上的 Spark 和 Hadoop 等框架的提示。

规划集群硬件、联网和安全

本教程中，您将创建一个简单的 EMR 集群，而无需配置高级选项。高级选项允许您指定 Amazon EC2 实例类型、集群联网和集群安全性。有关规划和启动满足您要求的集群的更多信息，请参阅计划和配置集群和Amazon EMR 中的安全性。

管理集群

深入了解如何在管理集群中运行的集群。要管理集群，您可以连接到集群、调试步骤以及跟踪集群活动和健康状况。您还可以通过 EMR 托管扩展，调整集群资源以响应工作负载需求。

使用不同的界面

除了 Amazon EMR 控制台之外，您还可以使用 Amazon Command Line Interface、Web 服务 API 或众多受支持的 Amazon SDK 之一来管理 Amazon EMR。有关更多信息，请参阅管理界面。

您可以通过多种方式与安装在 Amazon EMR 集群上的应用程序进行交互。某些应用程序（如 Apache Hadoop）会发布您可以查看的 Web 界面。有关更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

浏览 EMR 技术博客

有关新的 Amazon EMR 功能的示例演练和深入的技术讨论，请参阅Amazon大数据博客。

控制台中的新增功能

Amazon EMR 已迁移到全新控制台。新控制台更新了界面，您不仅能够以直观方式管理 Amazon EMR 环境，而且能轻松访问文档、产品信息和其他资源。本页介绍旧控制台与 Amazon EMR 新 Amazon Web Services Management Console 在体验方面的重要区别。

我使用的控制台

要确定您目前使用的 Amazon EMR 控制台，请在浏览器中查看控制台页面的 URL：

- 新控制台 URL：https://console.aws.amazon.com/emr
- 旧控制台 URL：https://console.aws.amazon.com/elasticmapreduce

> **Note**
>
> 如果您选择使用旧版控制台，Amazon EMR 会记住您的偏好 8 小时。超过时间限制后，即便是从其他来源访问旧控制台的直接链接，也会转到新控制台。点击新控制台侧面导航栏中的切换到旧控制台链接，您可以继续访问和使用旧控制台。此链接会一直保留到 2023 年 11 月 8 日，直到旧控制台不再可用。

Amazon EMR 控制台功能正在分阶段迁移到新控制台。下表列出了 Amazon EMR 控制台主要组件及其控制台迁移状态。

Amazon EMR 控制台组件	新控制台	旧控制台
EMR Studio[1]	✓	✓
创建和管理集群	✓	✓
阻止公有访问	✓	✓
监控 Amazon CloudWatch Events	✓	✓
安全配置	✓	✓

Amazon EMR 控制台组件	新控制台	旧控制台
虚拟集群（Amazon EMR on EKS）	✓	✓
查看并管理 Amazon Virtual Private Cloud 子网[2]	✓	✓
Notebooks[3]	✓	✓

[1] EMR Studio 在新旧控制台中都使用了新的界面体验。

[2] 在新控制台中，您可以创建集群时在 Networking（联网）部分查看和管理 Amazon VPC 子网。在旧控制台中，使用左侧导航栏中的链接访问 Amazon VPC 子网列表。

[3] EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

使用旧控制台

如果您选择使用旧版控制台，Amazon EMR 会记住您的偏好 8 小时。超过时间限制后，即便是从其他来源访问旧控制台的直接链接，也会转到新控制台。点击新控制台侧面导航栏中的切换到旧控制台链接，您可以继续访问和使用旧控制台。此链接会一直保留到 2023 年 11 月 8 日，直到旧控制台不再可用。

差异摘要

本部分概括介绍了 Amazon EMR 旧控制台和 Amazon EMR 新控制台之间的体验差异。差异分为以下几类：

- 新旧控制台之间的集群兼容性
- 创建集群时的差异
- 查看或编辑集群详细信息时的差异

- 列出和搜索集群时的差异
- 使用安全配置时的差异

新旧控制台之间的集群兼容性

在某些情况下，您在 Amazon EMR 旧控制台中创建的集群可能与新控制台不兼容。以下列表说明了 Amazon EMR 新控制台的兼容性要求。

- 新控制台支持在 Amazon EMR 发行版 5.20.1 及更高版本中创建的集群。
- 您可以在新控制台中克隆使用横向缩减的集群，但如果您想对其进行手动扩展或使用托管扩展，则只能创建新集群。

要创建和使用与新控制台不兼容的集群，您可以使用Amazon Command Line Interface（Amazon CLI）、Amazon SDK 或旧控制台。

创建集群时的差异

下表重点介绍了使用 Amazon EMR 新控制台与 Amazon EMR 旧控制台创建集群时可能会出现的差异。

能力	新控制台	旧控制台
术语：Amazon EMR 集群节点类型	主节点、核心节点、任务节点	主节点、核心节点、任务节点
Amazon EMR 支持的发行版[1]	Amazon EMR 发行版 5.20.1 及更高版本	所有 Amazon EMR 发行版
快速启动集群	使用 Summary（摘要）面板下的 Create cluster（创建集群）按钮	使用 Create cluster - Quick Options（创建集群 – 快速选项）页面
配置 Spot 预调配超时	定义为集群中的每个实例集预调配实例的超时期限。	创建集群时，您无法自定义预调配超时时间。

能力	新控制台	旧控制台
服务角色和 Amazon EC2 实例配置文件角色	新控制台不创建默认角色；您必须使用 IAM 控制台创建角色或选择一个已创建的 IAM 角色	支持使用 v1 和 v2 策略创建默认角色，或者您可以选择已创建的 IAM 角色
集群可见性	在 Amazon EMR 控制台中，您无法让集群对所有用户可见；IAM policy 决定了集群访问权限	在 Amazon EMR 控制台中，如果您使用已弃用的 v1 角色创建策略，则可以让集群对所有用户可见
联网 – 配置私有子网	您必须在各自的 Amazon S3 和 Amazon VPC 控制台中配置 Amazon S3 端点和 NAT 网关	您可以直接从旧控制台的 Create cluster（创建集群）工作流配置 Amazon S3 端点和 NAT 网关
EMR 文件系统一致视图（EMRFS CV）	Amazon S3 已于 2020 年 12 月 1 日发布强大的先写后读一致性，您无需对 EMR 集群使用 EMRFS CV	EMRFS CV 已启用，但您可以关闭 EMRFS CV 并删除其所使用的 Amazon DynamoDB 数据库；有关更多信息，请参阅一致视图
调试	您可以使用集群详细信息页面上的应用程序 UI 界面调试任务	您可以使用调试程序工具 [advanced options（高级选项）中的步骤 3]，为在 Amazon EMR 发行版 4.1.0 到 5.27.0 上运行的集群调试任务

[1] 您无法在新控制台中使用早于 Amazon EMR 5.20.1 的发行版创建或编辑集群，但使用早于 5.20.1 的发行版创建的任何现有集群将继续运行。要使用早于 5.20.1 的 Amazon EMR 发行版创建和编辑集群，请使用 API 或 CLI，或切换回旧控制台。

列出和搜索集群时的差异

下表重点介绍了使用 Amazon EMR 新控制台与 Amazon EMR 旧控制台在列表视图中查看和搜索集群详细信息时可能会出现的差异。

> **Note**
>
> 对于旧控制台和新控制台，当您对集群列表应用数据筛选条件时，它会查询整个数据库。但是，当您在搜索框中输入文本字符串时，搜索仅适用于列表已加载客户端的结果。

能力	新控制台	旧控制台
查看集群详细信息	您可以选择 Cluster ID（集群 ID）来查看详尽的集群详细信息，例如配置选项、持久性应用程序 UI 和日志。	您可以展开和折叠每个集群行以查看配置详细信息等信息，并访问集群监控和日志的链接。
搜索集群	使用单个搜索字段输入文本搜索查询，创建并应用数据筛选条件，如“Status = Any active status”（状态=任何活动状态）。	使用下拉列表细化集群的状态（活动、已终止、失败），并使用单独的字段输入文本搜索查询。
查找失败的集群	要搜索失败的集群，请应用筛选条件 Status（状态）= Terminated with errors（已终止但有错误）。	要搜索失败的集群，请应用筛选条件 Failed clusters（失败的集群）。

查看或编辑集群详细信息时的差异

下表重点介绍了使用 Amazon EMR 新控制台与 Amazon EMR 旧控制台查看或编辑现有集群详细信息时可能会出现的差异。

能力	新控制台	旧控制台
查看实例组和实例集中的实例，以及扩展、预配置、调整大小和终止选项	在实例选项卡中查看实例的选项和详细信息。在属性选项卡中查看终止选项。	在硬件选项卡中查看实例配置和终止选项。
查看应用程序 UI、日志和配置（Apache Spark UI、Spark 历史记录服务、Apache Tez UI、YARN 时间线服务器）	在配置选项卡中查看集群配置。您可以启动实时、持久性应用程序 UI，从应用程序选项卡查看应用程序的日志。	在配置选项卡中查看集群配置。您可以启动实时、持久性应用程序 UI，从应用程序用户界面选项卡查看应用程序的日志。自 2023 年 1 月 23 日起，高级应用程序历史记录不再可用。
将集群导出到 CL	集群详细信息和列表视图“操作”菜单中可用的选项为“查看适用于克隆集群的命令”	集群列表视图“操作”菜单中可用的选项为“Amazon CLI 导出”

使用安全配置时的差异

下表重点介绍了使用 Amazon EMR 新控制台与 Amazon EMR 旧控制台配置安全选项时可能出现的差异。

能力	新控制台	旧控制台
克隆安全配置	✓	
使用 Trino 和 Apache Ranger 进行联合治理	✓	
使用运行时角色将工作提交到集群[1]	✓	

能力	新控制台	旧控制台
授予对 EMR 文件系统（EMRFS）数据的访问权限	Amazon S3 接入点	Amazon Identity and Access Management（IAM）角色
Amazon Lake Formation 访问控制	运行时角色	SAML 联合身份验证

[1] 要在步骤提交期间传递角色，集群所使用的安全配置必须附加 IAM 权限策略，以便用户只传递批准的角色，并且您的任务可以访问 Amazon EMR 资源。有关更多信息，请参阅Amazon EMR 步骤的运行时角色。

Amazon EMR Studio

Amazon EMR Studio 是一个基于 Web 的集成开发环境 (IDE)，适用于依托 Amazon EMR 集群运行的完全托管式 Jupyter notebook。您可以为团队设置 EMR Studio，以开发、可视化和调试用 R、Python、Scala 和 PySpark 编写的应用程序。EMR Studio 已与 Amazon Identity and Access Management（IAM）和 IAM Identity Center 集成，以便用户使用其公司凭证登录。

您可以免费创建 EMR Studio。当您使用 EMR Studio 时，需为 Amazon S3 存储和 Amazon EMR 集群支付相应的费用。有关产品详细信息和亮点，请参阅 Amazon EMR Studio 服务页面。

EMR Studio 主要功能

Amazon EMR Studio 提供以下功能：

- 凭借 Amazon Identity and Access Management（IAM）或 Amazon IAM Identity Center（IAM Identity Center）与贵企业的身份供应商进行用户的身份验证。
- 按需访问并启动 Amazon EMR 集群以运行 Jupyter notebook 任务。
- 连接到 Amazon EMR on EKS 集群以在任务运行时提交工作。
- 浏览并保存示例笔记本。有关示例笔记本的更多信息，请参阅 EMR Studio 笔记本示例 GitHub 存储库。
- 使用 Python、PySpark、Spark Scala、Spark R 或 SparkSQL 分析数据，并安装自定义内核和库。
- 与同一 Workspace 中的其他用户实时协作。有关更多信息，请参阅配置 Workspace 协作。
- 在处理笔记本中的数据之前，使用 EMR Studio SQL Explorer 浏览数据目录、运行 SQL 查询和下载结果。
- 使用编排工具（例如 Apache Airflow 或 Amazon Managed Workflows for Apache Airflow）将参数化笔记本作为计划工作流的一部分运行。有关更多信息，请参阅Amazon大数据博客中的 Orchestrating analytics jobs on EMR Notebooks using MWAA。
- 链接代码存储库，例如 GitHub 和 BitBucket。
- 使用 Spark 历史记录服务器、Tez UI 或 YARN 时间线服务器跟踪和调试任务。

EMR Studio 也符合 HIPAA 资格，并通过 HITRUST CSF 和 SOC 2 认证。有关 Amazon 服务 HIPAA 合规性的更多信息，请参阅https://www.amazonaws.cn/compliance/hipaa-compliance/。要了解有关 Amazon 服务 HITRUST CSF 合规性的更多信息，请参阅 https://www.amazonaws.cn/compliance/hitrust/。有关 Amazon 服务其他合规性计划的更多信息，请参阅 Amazon 合规性计划范围内的服务。

Amazon EMR Studio 功能历史记录

此表列出了对 Amazon EMR 托管扩展功能的更新。

发行日期	能力
2023 年 10 月 26 日	增加了创建具有交互功能的 EMR Serverless 应用程序的功能。
2023 年 2 月 28 日	为 EMR Serverless 应用程序添加了对应用程序日志存储的 Amazon KMS 客户托管密钥支持。
2023 年 2 月 23 日	为 EMR 无服务器任务提交添加了一键创建 IAM 角色功能。为 EMR Serverless 应用程序选择自定义映像添加了 ECR 查询功能。
2023 年 1 月 27 日	无头执行笔记本可以用 `%execute_notebook` magic 跟踪每个单元格的执行进度。
2023 年 1 月 23 日	持久性应用程序已经过优化，可以缩短启动时间。

Amazon EMR Studio 工作原理

Amazon EMR Studio 是您为用户团队创建的 Amazon EMR 资源。EMR Studio 是一种基于 Web 的独立集成开发环境，适用于依托 Amazon EMR 集群运行的 Jupyter notebook。用户使用其公司凭证登录 Studio。

您创建的每个 EMR Studio 都使用以下 Amazon 资源：

- 含有子网的 Amazon Virtual Private Cloud (VPC) - 用户可在指定 VPC 中的 Amazon EMR 和 Amazon EMR on EKS 集群上运行 Studio 内核和应用程序。EMR Studio 可以连接到您在创建 Studio 时指定的子网中的任何集群。
- IAM 角色和权限策略 – 要管理用户权限，您可以创建附加到用户的 IAM 身份或用户角色的 IAM 权限策略。EMR Studio 还使用 IAM 服务角色和安全组与其他 Amazon 服务进行互操作。有关更多信息，请参阅 访问控制 和 定义安全组以控制 EMR Studio 网络流量。
- 安全组 - EMR Studio 使用安全组在 Studio 和 EMR 集群之间建立安全的网络通道。
- Amazon S3 备份位置 - EMR Studio 将笔记本作业保存在 Amazon S3 所在位置。

以下步骤概括了如何创建和管理 EMR Studio：

1. 在您的 Amazon Web Services 账户 中，使用 IAM 或 IAM Identity Center 身份验证创建 Studio。有关说明，请参阅 设置 Amazon EMR Studio。
2. 将用户和组分配给 Studio。使用权限策略为每个用户设置细化权限。有关更多信息，请参阅主题 分配和管理 EMR Studio 用户。
3. 借助 Amazon CloudTrail 事件开启监控 EMR Studio 操作。有关更多信息，请参阅 监控 Amazon EMR Studio 操作。
4. 通过集群模板和 Amazon EMR on EKS 托管式终端节点向 Studio 用户提供更多集群选项。

身份验证和用户登录

Amazon EMR Studio 支持两种身份验证模式：IAM 身份验证模式和 IAM Identity Center 身份验证模式。IAM 模式使用 Amazon Identity and Access Management（IAM），而 IAM Identity Center 模式使用 Amazon IAM Identity Center。创建 EMR Studio 时，您可以为 EMR Studio 的所有用户选择身份验证模式。

IAM 身份验证模式

通过 IAM 身份验证模式，您可以使用 IAM 身份验证或 IAM 联合身份验证。

IAM 身份验证允许您在 IAM 中管理其用户、组和角色等身份。您凭借 IAM 权限策略以及基于属性的访问控制 (ABAC)，授予用户访问 Studio 的权限。

IAM 联合身份验证允许您在第三方身份提供商 (IdP) 和 Amazon 之间建立信任，以便您可以通过自己的 IdP 管理用户身份。

IAM Identity Center 身份验证模式

IAM Identity Center 身份验证模式允许您给予用户联合访问 EMR Studio 的权限。您可以使用 IAM Identity Center 通过您的 IAM Identity Center 目录、现有公司目录或诸如 Azure Active Directory（AD）的外部 IdP 进行用户和组的身份验证。之后，您可以用身份提供商 (IdP) 管理用户。

EMR Studio 支持使用 IAM Identity Center 的以下身份提供商：

- Amazon Managed Microsoft AD 和自行管理的 Active Directory – 有关更多信息，请参阅连接到 Microsoft AD 目录。
- 基于 SAML 提供商 – 有关完整列表，请参阅支持的身份提供商。
- IAM Identity Center 目录 – 有关更多信息，请参阅管理 IAM Identity Center 中的身份。

身份验证如何影响登录和用户分配

您为 EMR Studio 选择的身份验证模式会影响用户登录 Studio 的方式、将用户分配给 Studio 的方式以及您授权（授予权限）用户执行诸如创建新的 Amazon EMR 集群之类操作的方式。

下表根据身份验证模式概述了 EMR Studio 的登录方法。

身份验证模式提供的 EMR Studio 登录选项

身份验证模式	登录方法	描述
• IAM（身份验证和联合身份验证） • IAM Identity Center	EMR Studio URL	用户使用 Studio 访问 URL 登录 Studio。例如，`https://xxxxxxxxxxxxxxxxxxxx xxxx.emrstudio-prod.us-east -1.amazonaws.com` 。 使用 IAM 身份验证时，用户输入 IAM 凭证。当您使用 IAM 联合身份验证时或 IAM Identity Center 时，EMR Studio 将用户重定向到身份提供商的登录 URL 以输入凭证。 在联合身份验证的条件下，此登录选项称为服务提供商 (SP) 启动的登录。
• IAM（联合身份验证） • IAM Identity Center	Identity provider (IdP)（身份提供商 (IdP)）门户	用户登录身份提供商的门户网站，例如 Azure 门户，然后启动 Amazon EMR 控制台。启动 Amazon EMR 控制台之后，用户根据 Studios 列表选择并打开 Studio。 您还可以将 EMR Studio 配置为 SAML 应用程序，以便用户可以从身份提供商的门户网站登录到指定的 Studio。有关说明，请参阅在身份提供商 (IdP) 门户网站将 EMR Studio 配置为 SAML 应用程序。 在联合身份验证条件下，此登录选项称为身份提供商 (IdP) 启动的登录。
• IAM (身份验证)	Amazon Web Services	用户使用 IAM 凭证登录 Amazon Web Services Management Console 并通过 Studios 列表 在 Amazon EMR 控制台打开 Studio。

身份验证模式	登录方法	描述
	Management Console	

下表概述了在身份验证模式下针对 EMR Studio 进行的用户分配和授权。

在身份验证模式下进行 EMR Studio 用户分配和授权

身份验证模式	用户分配	用户授权
IAM（身份验证和联合身份验证）	在附加到 IAM 身份（用户、组或角色）的 IAM 权限策略中，允许IAM 的 `CreateStudioPresignedUrl` 操作。 对于联合身份用户，在您为用于联合身份验证的 IAM 角色配置的权限策略中，允许 IAM 的 `CreateStudioPresignedUrl` 操作。 使用基于属性的访问控制 (ABAC) 来指定用户可以访问的 Studio 或 Studios。 有关说明，请参阅 将用户或组分配到 EMR Studio。	定义允许某些 EMR Studio 操作的 IAM 权限策略。 对于本地用户，请将 IAM 权限策略附加到 IAM 身份（用户、组或角色）。对于联合身份用户，在您为用于联合身份验证的 IAM 角色配置的权限策略中，允许执行 Studio 操作。 有关更多信息，请参阅为 Amazon EC2 或 Amazon EKS 配置 EMR Studio 用户权限。
IAM Identity Center	通过将用户映射到具有指定会话策略的 Studio，将用户分配给 Studio。 有关说明，请参阅 将用户或组分配到 EMR Studio。	定义允许某些 EMR Studio 操作的 IAM 会话策略。当您将用户分配给 Studio 时，请将会话策略映射到用户。 有关更多信息，请参阅IAM Identity Center 身份验证模式的用户权限。

访问控制

在 Amazon EMR Studio，您可以配置用户授权（权限）Amazon Identity and Access Management(IAM) 基于身份的策略。通过这些策略，您可以指定允许的操作和资源以及允许操作的条件。

IAM 身份验证模式的用户权限

在使用 EMR Studio 的 IAM 身份验证时，要设置用户权限，您必须允许 IAM 权限策略的`elasticmapreduce:RunJobFlow` 等操作。您可以创建一个或多个应用的权限策略。例如，您可以创建禁止用户创建新的 Amazon EMR 集群的基本策略，也可以创建允许创建集群的策略。有关 Studio 的所有操作列表，请参阅 EMR Studio 用户的 Amazon Identity and Access Management 权限。

IAM Identity Center 身份验证模式的用户权限

当您使用 IAM Identity Center 身份验证时，您可以创建单个 EMR Studio 用户角色。用户角色 是指 Studio 在用户登录时担任的专用 IAM 角色。

您可以将 IAM 会话策略附加到 EMR Studio 用户角色。会话策略是一种特殊的 IAM 权限策略，它限制了联合身份用户在 Studio 登录会话期间可以执行的操作。会话策略让您可以为用户或组设置特定的 EMR Studio 权限，而无需创建多个 IAM 角色。

当您将用户和组分配到 Studio 时，您可以将会话策略映射到该用户或组以应用精细的权限。您还可以随时更新用户或组的会话策略。Amazon EMR 存储您创建的每个会话策略映射。

有关会话策略的更多信息，请参阅《Amazon Identity and Access Management 用户指南》中的策略与权限。

Workspaces

Workspaces 是 Amazon EMR Studio 的主要构建块。要组织笔记本，用户必须在 Studio 创建一个或多个 Workspaces。有关更多信息，请参阅学习 WorkSpace 基础知识。

Workspace 类似于 JupyterLab 中的 Workspaces，因为它会保留笔记本作业的状态。然而，Workspace 用户界面通过其他工具扩展了开源 JupyterLab 界面，以帮助您创建和附加 EMR 集群、运行任务、探索示例笔记本和链接 Git 存储库。

下面的列表包括 EMR Studio Workspace 的主要功能：

- Workspace 可见性是基于 Studio 的。您在 Studio 中创建的 Workspaces 在其他 Studio 中不可见。
- 预设情况下，Workspace 是共享的，所有 Studio 用户都可以看到。但一次只能有一个用户打开一个 Workspace 并在其中工作。要与其他用户同时工作，您可以配置 Workspace 协作
- 启用 Workspace 协作后，您可以同时与 Workspace 中的其他用户协作。有关更多信息，请参阅配置 Workspace 协作。
- Workspace 中的笔记本共享同一 EMR 集群以运行命令。您可以将 Workspace 附加到 Amazon EC2 上运行的 Amazon EMR 集群或附加到 Amazon EMR on EKS 虚拟集群和托管式终端节点上。
- Workspace 可以切换到与 Studio 子网关联的另一个可用区。您可以停止并重启 Workspace 以提示故障转移过程。当您重启 Workspace 时，如果 Studio 被配置为拥有对多个可用区的访问权限，则 EMR Studio 将在 Studio VPC 的其他可用区中启动该 Workspace。如果 Studio 只有一个可用区，则 EMR Studio 将尝试在其他子网中启动该 Workspace。有关更多信息，请参阅解决 Workspace 连接问题。
- Workspace 可以连接到与 Studio 关联的任何子网中的集群。

有关创建和配置 EMR Studio Workspaces 的更多信息，请参阅学习 WorkSpace 基础知识。

Amazon EMR Studio 中的笔记本存储

当您使用 Workspace 时，EMR Studio 会定期将您的笔记本文件中的单元格自动保存到与您的 Studio 关联的 Amazon S3 位置。此备份过程可以保留会话之间的工作，以便您以后可以在不向 Git 存储库提交更改的情况下再返回。有关更多信息，请参阅 保存 Workspace 内容。

当您从 Workspace 中删除笔记本文件时，EMR Studio 会为您从 Amazon S3 中删除备份版本。但是，如果您删除 Workspace 而没有先删除其笔记本文件，则笔记本文件会保留在 Amazon S3 中并继续产生存储费用。要了解更多信息，请参阅 删除 Workspace 和笔记本文件。

EMR Studio 注意事项

注意事项

使用 EMR Studio 时请考虑以下事项：

- EMR Studio 已在下列Amazon区域开放：美国东部（弗吉尼亚北部、俄亥俄）、美国西部（加利福尼亚北部、俄勒冈）、亚太地区（孟买、首尔、新加坡、悉尼、东京）、加拿大（中部）、欧洲（法兰克福、爱尔兰、伦敦、巴黎、斯德哥尔摩）和南美洲（圣保罗）。

- 要让用户为 Workspace 预置在 Amazon EC2 上运行的新 EMR 集群，请将 EMR Studio 与一组集群模板关联起来。管理员可以使用 Service Catalog 定义集群模板，并且可以选择用户或组是可以访问 Studio 中的集群模板，还是不能访问集群模板。
- 当您定义对存储在 Amazon S3 中的笔记本文件的访问权限或从 Amazon Secrets Manager 读取密钥时，请使用 Amazon EMR 服务角色。这些权限不支持会话策略。
- 您可以创建多个 EMR Studio 来控制对不同 VPC 中的 EMR 集群的访问。
- 请使用 Amazon CLI 设置 Amazon EMR on EKS 集群。然后，您可以使用 Studio 界面通过托管式终端节点将集群附加到 Workspaces，以运行笔记本任务。
- EMR Studio 不支持以下 Python 魔术命令：
  - `%alias`
  - `%alias_magic`
  - `%automagic`
  - `%macro`
  - `%%js`
  - `%%javascript`
  - 使用 `%configure` 修改 `proxy_user`
  - 使用 `%env` 或 `%set_env` 修改 `KERNEL_USERNAME`
- Amazon EMR on EKS 集群不支持 EMR Studio 的 SparkMagic 命令。
- 要在笔记本单元格中编写多行 Scala 语句，请确保除最后一行以外的所有其他行均以句点结束。下面的示例对多行 Scala 语句使用了正确的句法。

```
val df = spark.sql("SELECT * from table_name).
        filter("col1=='value'").
        limit(50)
```

已知问题

- 在创建 Studio 之前，请确保在浏览器中禁用了 FoxyProxy 或 SwitchyOmega 等代理管理工具。当您选择 Create Studio (创建 Studio) 时，活动代理可能会导致错误，并导致出现 Network Failure (网络故障) 错误消息。
- 由于超时问题，在 Amazon EMR on EKS 集群上运行的内核会无法启动。如果您在启动内核时遇到错误或问题，请关闭笔记本文件、关闭内核，然后重新打开笔记本文件。

- 当您使用 Amazon EMR on EKS 集群时，Restart kernel (重新启动内核) 操作将无法按预期工作。在您选择 Restart kernel (重新启动内核) 后，请刷新 Workspace 以使重启生效。
- 如果 Workspace 未附加到集群，则当 Studio 用户打开笔记本文件并尝试选择内核时会显示一条错误消息。您可以通过选择 Ok (确定) 忽略此错误消息，但您必须先将 Workspace 附加到集群并选择内核，然后才能运行笔记本代码。
- 当您使用具有安全配置的 Amazon EMR 6.2.0 来设置集群安全时，Workspace 界面显示为空白并且无法按预期工作。如果您要为集群的 EMRFS 配置数据加密或 Amazon S3 授权，我们建议您使用其他受支持的 Amazon EMR 版本。EMR Studio 适用于 Amazon EMR 版本 5.32.0（Amazon EMR 5.x 系列）或 6.2.0（Amazon EMR 6.x 系列）及更高版本。
- 当您从笔记本文件启动集群上 Spark UI 时，您可以在运行笔记本代码后查看有关任务的信息。但是，当您从 Studio 中 Clusters (集群) 列表启动 Spark 历史记录服务器时，任务可能将有长达两分钟的时间不显示。
- 当您调试 Amazon EC2 任务上运行的 Amazon EMR时，指向集群上 Spark UI 的链接可能无法正常工作或无法显示。要重新生成链接，请创建一个新的笔记本单元格并运行`%%info`命令。
- Jupyter Enterprise Gateway 不会在以下 Amazon EMR 发行版本中清理集群主节点上的空闲内核：5.32.0、5.33.0、6.2.0 和 6.3.0。空闲内核消耗计算资源，这可能会导致长时间运行的集群出现故障。您可以使用以下示例脚本为 Jupyter Enterprise Gateway 配置空闲内核清理。您可以使用 SSH 连接到主节点，或以步骤的形式提交脚本。有关更多信息，请参阅在 Amazon EMR 集群上运行命令和脚本。

```
#!/bin/bash
sudo tee -a /emr/notebook-env/conf/jupyter_enterprise_gateway_config.py << EOF
c.MappingKernelManager.cull_connected = True
c.MappingKernelManager.cull_idle_timeout = 10800
c.MappingKernelManager.cull_interval = 300
EOF
sudo systemctl daemon-reload
sudo systemctl restart jupyter_enterprise_gateway
```

- 当您将自动终止策略与 Amazon EMR 版本 5.32.0、5.33.0、6.2.0 或 6.3.0 配合使用时，Amazon EMR 会将集群标记为空闲，即使您有活动的 Python3 内核，也有可能自动终止该集群。这是因为执行 Python3 内核不会在该集群上提交 Spark 作业。要将自动终止与 Python3 内核配合使用，我们建议您使用 Amazon EMR 版本 6.4.0 或更高版本。有关自动终止的更多信息，请参阅 使用自动终止策略。
- 当您使用 `%%display` 要在表格中显示 Spark DataFrame 时，非常宽的表格可能会被截断。您可以右键单击输出，然后选择 Create New View for Output（为输出创建新视图），以获取输出的可滚动视图。

- 启动基于 Spark 的内核（如 PySpark、Spark 或 SparkR）将启动 Spark 会话，在笔记本中运行单元格会将该会话中的 Spark 作业排队。当您中断正在运行的单元格时，Spark 作业将继续运行。要停止 Spark 作业，您应该使用集群上的 Spark UI。有关如何连接到 Spark UI 的说明，请参阅 使用 EMR Studio 调试应用程序和任务。

功能限制

Amazon EMR Studio 不支持以下 Amazon EMR 功能：

- 在使用指定 Kerberos 身份验证的安全配置创建的 EMR 集群上附加和运行任务
- 具有多个主节点的集群
- 对于低于 6.9.0 的 Amazon EMR 6.x 版本和低于 5.36.1 的 5.x 版本，使用基于 Amazon Graviton2 的 Amazon EC2 实例的集群

EMR Studio 的 Service Limits

下表呈现了 EMR Studio 的服务限制。

项目	限制
EMR Studio	每个 Amazon 账户最多 100 个
子网	每个 EMR Studio 最多关联 5 个
IAM Identity Center 组	每个 EMR Studio 最多分配 5 个
IAM Identity Center 用户	每个 EMR Studio 最多分配 100 个

VPC 和子网最佳实践

使用以下最佳实践为 EMR Studio 设置带有子网的 Amazon Virtual Private Cloud (Amazon VPC)：

- 您最多可以在 VPC 中指定五个子网与 Studio 关联。我们建议您在不同的可用区中提供多个子网，以便支持 Workspace 可用性，并为 Studio 用户提供跨不同可用区访问集群的权限。要了解有关使用 VPC、子网和可用区的更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的 VPC 和子网。

- 您指定的子网应能相互通信。
- 要让用户将 Workspace 链接到公共托管的 Git 存储库，您应仅指定有权通过网络地址转换 (NAT) 访问互联网的私有子网。有关为 Amazon EMR 设置私有子网的更多信息，请参阅 私有子网。
- 当您将 Amazon EMR on EKS 与 EMR Studio 配合使用时，您的 Studio 与用于注册虚拟集群的 Amazon EKS 集群之间必须有至少一个公有子网。否则，您的托管式终端节点将不会作为选项显示在 Studio Workspace 中。您可以创建 Amazon EKS 集群并将其与属于 Studio 的子网关联，或创建 Studio 并指定您的 EKS 集群子网。
- 如果计划通过 EMR Studio 使用 Amazon EMR on EKS，请选择同一 VPC 作为您的 Amazon EKS 集群 Worker 节点。

Amazon EMR Studio 的集群要求

在 Amazon EC2 上运行的 Amazon EMR 集群

您为 EMR Studio Workspace 创建的在 Amazon EC2 上运行的所有 Amazon EMR 集群都必须满足以下要求。您使用 EMR Studio 界面创建的集群会自动满足这些要求。

- 集群必须使用 Amazon EMR 5.32.0（Amazon EMR 5.x 系列）或 6.2.0（Amazon EMR 6.x 系列）及更高版本。您可以使用 Amazon EMR 控制台、Amazon Command Line Interface 或 SDK 创建集群，然后将其附加到 EMR Studio Workspace。Studio 用户也可以在创建 Amazon EMR Workspace 或在其中工作时预置和附加集群。有关更多信息，请参阅将计算资源附加到 EMR Studio Workspace。
- 集群必须在 Amazon Virtual Private Cloud 中启动。不支持 EC2-Classic 平台。
- 集群必须安装有 Spark、Livy 和 Jupyter Enterprise Gateway。如果您计划将集群用于 SQL Explorer，则应同时安装 Presto 和 Spark。
- 要使用 SQL Explorer，集群必须使用 Amazon EMR 版本 5.34.0 或更高版本或者 6.4.0 或更高版本并安装了 Presto。如果要将 Amazon Glue 数据目录指定为 Presto 的 Hive 元数据仓，则必须在集群上进行配置。有关更多信息，请参阅将 Presto 与 Amazon Glue 数据目录结合使用。
- 集群必须在网络地址转换 (NAT) 的私有子网中启动，公共托管的 Git 存储库与 EMR Studio 才能配合使用。

当您使用 EMR Studio 时，我们建议使用以下集群配置。

- 将 Spark 会话的部署模式设置为集群模式。集群模式将应用程序主进程放置在核心节点上，而不是集群的主节点上。这样做可以减轻主节点的潜在内存压力。有关更多信息，请参阅 Apache Spark 文档中的集群模式概述。

- 如下面的示例配置所示，将 Livy 超时从默认的 1 小时更改为 6 小时。

```
{
    "classification":"livy-conf",
        "Properties":{
            "livy.server.session.timeout":"6h",
            "livy.spark.deploy-mode":"cluster"
        }
}
```

- 创建包含多达 30 个实例的多样化实例队列，并在竞价型实例队列中选择多种实例类型。例如，您可以为 Spark 工作负载指定以下内存优化的实例类型：r5.2x、r5.4x、r5.8x、r5.12x、r5.16x、r4.2x、r4.4x、r4.8x、r4.12 等。有关更多信息，请参阅配置实例集。
- 将容量优化的分配策略用于竞价型实例，以帮助 Amazon EMR 根据来自 Amazon EC2 的实时容量洞察进行有效的实例选择。有关更多信息，请参阅实例集的分配策略。
- 在集群上启用托管式扩缩。将最大核心节点参数设置为您计划使用的最小持久性容量，并在竞价型实例上运行的充分多样化的任务队列上配置扩缩以节省成本。有关更多信息，请参阅在 Amazon EMR 中使用托管扩展。

我们还会敦促您始终启用 Amazon EMR Block Public Access（阻止公开访问），同时限制入站 SSH 流量进入可信来源。对集群的入站访问允许用户在集群上运行笔记本。有关更多信息，请参阅使用 Amazon EMR 阻止公有访问和使用安全组控制网络流量。

Amazon EMR on EKS 集群

除了在 Amazon EC2 上运行的 EMR 集群之外，您还可以使用 Amazon CLI 为 EMR Studio 设置和管理 Amazon EMR on EKS 集群。使用以下准则设置 Amazon EMR on EKS 集群：

- 针对 Amazon EMR on EKS 集群创建托管式 HTTPS 终端节点。用户将 Workspace 附加到托管式终端节点。您用于注册虚拟集群的 Amazon Elastic Kubernetes Service (EKS) 集群必须具有私有子网才能启用托管式终端节点。
- 如果您想使用公共托管的 Git 存储库，请使用具有至少一个私有子网和网络地址转换 (NAT) 的 Amazon EKS 集群。
- 避免使用 Amazon EKS 优化的 Arm Amazon Linux AMI，因为 Amazon EMR on EKS 托管式终端节点不支持。
- 避免使用仅限 Amazon Fargate 的 Amazon EKS 集群，因为不受支持。

配置 Amazon EMR Studio

本部分适用于 EMR Studio 管理员。它介绍了为您的团队设置 EMR Studio 的方法，并提供了为 EMR Studio 分配用户和组、设置集群模板以及优化 Apache Spark 等任务的说明。

主题

- 创建和管理 EMR Studio 所需的管理员权限
- 设置 Amazon EMR Studio
- 管理 Amazon EMR Studio
- 定义安全组以控制 EMR Studio 网络流量
- 为 Amazon EMR Studio 创建 Amazon CloudFormation 模板
- 为基于 Git 的存储库建立访问和权限
- 在 EMR Studio 优化 Spark 任务

创建和管理 EMR Studio 所需的管理员权限

本页所描述的 IAM 权限允许您创建和管理 EMR Studio。有关每个所需权限的详细信息，请参阅管理 EMR Studio 所需的权限。

管理 EMR Studio 所需的权限

下表列出了与创建和管理 EMR Studio 相关的运营。该表还显示了每个运营所需的权限。

> **Note**
>
> 您在使用 IAM Identity Center 身份验证模式时只需要 IAM Identity Center 和 Studio `SessionMapping` 操作。

创建和管理 EMR Studio 所需的管理员权限

操作	权限
创建 Studio	`"elasticmapreduce:CreateStudio", "sso:CreateManagedApplicationInstance", "iam:PassRole"`

操作	权限
描述 Studio	`"elasticmapreduce:DescribeStudio", "sso:GetManagedApplicationInstance"`
列出 Studios	`"elasticmapreduce:ListStudios"`
删除 Studio	`"elasticmapreduce:DeleteStudio", "sso:DeleteManagedApplicationInstance"`
Additional permissions required when you use IAM Identity Center mode	
将用户或组分配给 Studio	`"elasticmapreduce:CreateStudioSessionMapping", "sso:GetProfile", "sso:ListDirectoryAssociations", "sso:ListProfiles", "sso:AssociateProfile" "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup"`
请检索特定用户或组的 Studio 分配详细信息	`"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:GetManagedApplicationInstance", "elasticmapreduce:GetStudioSessionMapping"`
列出分配给 Studio 的所有用户和组	`"elasticmapreduce:ListStudioSessionMappings"`

操作	权限
更新附加到分配给 Studio 的用户或组的会话策略	`"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:GetManagedApplicationInstance", "elasticmapreduce:UpdateStudioSessionMapping"`
从 Studio 中删除用户或组	`"elasticmapreduce:DeleteStudioSessionMapping", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListDirectoryAssociations", "sso:GetProfile", "sso:GetManagedApplicationInstance" "sso:ListProfiles", "sso:DisassociateProfile"`

创建具有 EMR Studio 管理员权限的策略

1. 按照创建 IAM policy中的说明使用以下任一示例创建策略。您需要的权限取决于您的 EMR Studio 身份验证模式。

   为这些项插入您自己的值：

   - 替换 *<your-resource-ARN>* 以指定该语句针对您的使用案例涵盖的一个或多个对象的 Amazon Resource Name（ARN）。
   - 将 *<region>* 替换为您计划在其中创建 Studio 的 Amazon Web Services 区域 的代码。
   - 将 *<aws-account-id>* 替换为 Studio 的 Amazon 账户 ID。
   - 将 *<EMRStudio_Service_Role>* 和 *<EMRStudio_User_Role>* 替换为您的 EMR Studio 服务角色和 EMR Studio 用户角色的名称。

Example 示例策略：您使用 IAM 身份验证模式时的管理员权限

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/
*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "<your-resource-ARN>",
            "Action": [
                "elasticmapreduce:ListStudios"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>"
            ],
            "Action": "iam:PassRole"
        }
    ]
}
```

Example 示例策略：您使用 IAM Identity Center 身份验证模式时的管理员权限

> **Note**
>
> IAM Identity Center 和 IAM Identity Center Directory API 不支持在 IAM policy 语句的资源元素中指定 ARN。要允许访问 IAM Identity Center 和 IAM Identity Center 目录，以下权

限为 IAM Identity Center 操作指定所有资源 "Resource":"*"。有关更多信息，请参阅 IAM Identity Center 目录的操作、资源和条件键。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/
*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio",
                "elasticmapreduce:CreateStudioSessionMapping",
                "elasticmapreduce:GetStudioSessionMapping",
                "elasticmapreduce:UpdateStudioSessionMapping",
                "elasticmapreduce:DeleteStudioSessionMapping"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "<your-resource-ARN>",
            "Action": [
                "elasticmapreduce:ListStudios",
                "elasticmapreduce:ListStudioSessionMappings"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>",
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-User-Role>"
            ],
            "Action": "iam:PassRole"
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "sso:CreateManagedApplicationInstance",
```

```
                "sso:GetManagedApplicationInstance",
                "sso:DeleteManagedApplicationInstance",
                "sso:AssociateProfile",
                "sso:DisassociateProfile",
                "sso:GetProfile",
                "sso:ListDirectoryAssociations",
                "sso:ListProfiles",
                "sso-directory:SearchUsers",
                "sso-directory:SearchGroups",
                "sso-directory:DescribeUser",
                "sso-directory:DescribeGroup"
            ]
        }
    ]
}
```

2. 将该策略附加到您的 IAM 身份（用户、角色或组）。有关说明，请参阅添加和删除 IAM 身份权限。

设置 Amazon EMR Studio

完成以下步骤以设置 Amazon EMR Studio。

开始之前

> **Note**
>
> 如果您计划将 EMR Studio 与 Amazon EMR on EKS 结合使用，我们建议您在设置 Studio 之前，先设置适用于 EMR Studio 的 Amazon EMR on EKS。

在您设置 EMR Studio 之前，请确保您具有以下项目：

- 一个 Amazon Web Services 账户。有关说明，请参阅 设置 Amazon EMR。
- 创建和管理 EMR Studio 所需的权限。有关更多信息，请参阅the section called “创建 EMR Studio 所需的管理员权限”。
- 一个 Amazon S3 存储桶，EMR Studio 可以在其中备份 Studio 中的 Workspaces 和笔记本文件。有关说明，请参阅《Amazon Simple Storage Service (S3) 用户指南》中的创建存储桶。
- 如果要连接到 EC2 上的 Amazon EMR 或 Amazon EMR on EKS 集群，或者使用 Git 存储库，则需要使用适用于 Studio 的 Amazon Virtual Private Cloud（VPC），并且最多五个子网。将 EMR

Studio 与 EMR Serverless 结合使用时不需要 VPC。有关如何配置联网的提示，请参阅 VPC 和子网最佳实践。

设置 EMR Studio

1. 为 Amazon EMR Studio 选择身份验证模式
2. 创建以下 Studio 资源。
   - 创建 EMR Studio 服务角色
   - 为 Amazon EC2 或 Amazon EKS 配置 EMR Studio 用户权限
   - （可选）定义安全组以控制 EMR Studio 网络流量。
3. 创建 EMR Studio
4. 将用户或组分配到 EMR Studio

完成设置步骤后，您可以使用 Amazon EMR Studio。

为 Amazon EMR Studio 选择身份验证模式

EMR Studio 支持两种身份验证模式：IAM 身份验证模式和 IAM Identity Center 身份验证模式。IAM 模式使用 Amazon Identity and Access Management（IAM），而 IAM Identity Center 模式使用 Amazon IAM Identity Center。创建 EMR Studio 时，您可以为 EMR Studio 的所有用户选择身份验证模式。有关不同身份验证模式的更多信息，请参阅 身份验证和用户登录。

使用下表为 EMR Studio 选择身份验证模式。

如果您是…	我们建议…
已经熟悉或之前已经设置过 IAM 身份验证或联合	IAM 身份验证模式，具备下列优点： • 如果您已经管理 IAM 中的用户和组等身份，则能够快速设置 EMR Studio。 • 与 OpenID Connect (OIDC) 或安全断言标记语言 2.0 (SAML 2.0) 兼容的身份提供商一起工作。 • 支持使用具有同一 Amazon Web Services 账户 的多重身份提供商。

如果您是…	我们建议…
	• 有大量可用 Amazon Web Services 区域。 • 符合 SOC 2。
对 Amazon 或 Amazon EMR 不熟悉	IAM Identity Center 身份验证模式，提供以下功能： • 支持将简单的用户和组分配给 Amazon 资源。 • 与 Microsoft Active Directory 和 SAML 2.0 身份提供商一起使用。 • 简化多账户联合设置，以便您无需为企业的每个 Amazon Web Services 账户 单独配置联合身份。

为 Amazon EMR Studio 设置 IAM 身份验证模式

通过 IAM 身份验证模式，您可以使用 IAM 身份验证或 IAM 联合身份验证。IAM 身份验证 允许您在 IAM 中管理其用户、组和角色等身份。您凭借 IAM 权限策略以及基于属性的访问控制 (ABAC)，授予用户访问 Studio 的权限。IAM 联合身份验证 允许您在第三方身份提供商 (IdP) 和 Amazon 之间建立信任，以便您可以通过自己的 IdP 管理用户身份。

> **Note**
>
> 如果您已使用 IAM 来控制访问 Amazon 资源的权限，或者您已经为 IAM 配置了身份提供商 (IdP)，请参阅 IAM 身份验证模式的用户权限 以便您在使用 EMR Studio 的 IAM 身份验证模式时设置用户权限。

使用 Amazon EMR Studio 的 IAM 联合身份验证

要使用 EMR Studio 的 IAM 联合身份验证，您可以在您的 Amazon Web Services 账户 和身份提供者（IdP）之间创建信任关系，并允许联合身份用户访问 Amazon Web Services Management Console。创建此信任关系所采取的步骤取决于您身份提供商 (IdP) 的联合身份验证标准。

通常，您可以完成以下任务以使用外部身份供应商 (IdP) 配置联合身份验证。有关完整说明，请参阅《Amazon Identity and Access Management 用户指南》中的使 SAML 2.0 联合身份用户能够访问

Amazon Web Services Management Console和使自定义身份代理能够访问Amazon Web Services Management Console。

1. 通过您的身份提供商 (IdP) 收集信息。通常，这表示生成元数据文档以验证来自 IdP 的 SAML 身份验证请求。
2. 创建身份提供商 IAM 实体以存储有关 IdP 的信息。有关说明，请参阅创建 IAM 身份提供商。
3. 为您的 IdP 创建一个或多个 IAM 角色。在用户登录时，EMR Studio 将为联合身份用户分配角色。该角色允许您的 IdP 请求临时安全凭证以便访问 Amazon。有关说明，请参阅针对第三方身份提供商（联合）创建角色。您分配角色的权限策略决定了哪类联合身份用户可以在 Amazon 和 EMR Studio 操作。有关更多信息，请参阅 IAM 身份验证模式的用户权限。
4. （针对 SAML 提供商）借助与 Amazon 相关的信息以及您希望联合身份用户担任的角色，以完成 SAML IdP 信任配置。在配置过程中，在您的 IdP 和 Amazon 之间创建了信赖方信任。有关详细信息，请参阅《用户指南》中的使用信赖方信任配置您的 SAML 2.0 IdP 并添加声明。

在身份提供商 (IdP) 门户中将 EMR Studio 配置为 SAML 应用程序

您可以使用 EMR Studio 的深层链接将特定 EMR Studio 配置为 SAML 应用程序。这样，用户可以登录 IdP 门户并启动特定的 Studio，而不是通过 Amazon EMR 控制台进行导航。

- 使用以下格式将指向 EMR Studio 的深层链接配置为 SAML 断言验证后的着陆 URL。

```
https://console.aws.amazon.com/emr/home?region=<aws-region>#studio/<your-studio-id>/start
```

为 Amazon EMR Studio 设置 IAM Identity Center 身份验证模式

要准备 EMR Studio 的 Amazon IAM Identity Center，您必须配置身份源并预调配用户和组。预调配是使用户和组信息可供 IAM Identity Center 和使用 IAM Identity Center 的应用程序使用的过程。有关更多信息，请参阅用户和组预置。

EMR Studio 支持使用 IAM Identity Center 的以下身份提供商：

- Amazon Managed Microsoft AD 和自行管理的 Active Directory – 有关更多信息，请参阅连接到 Microsoft AD 目录。
- 基于 SAML 提供商 – 有关完整列表，请参阅支持的身份提供商。
- IAM Identity Center 目录 – 有关更多信息，请参阅管理 IAM Identity Center 中的身份。

为 EMR Studio 设置 IAM Identity Center

1. 要为 EMR Studio 设置 IAM Identity Center，您需要具备以下各项：

   - 您 Amazon 组织中的管理账户（如果您在组织中使用多个账户）。

     > **Note**
     >
     > 您只能使用管理账户启用 IAM Identity Center 并预调配用户和组。在设置 IAM Identity Center 之后，您可以使用成员账户创建 EMR Studio 并分配用户和组。要了解有关 Amazon术语的更多信息，请参阅 Amazon Organizations 术语和概念。

   - 如果您在 2019 年 11 月 25 日之前启用了 IAM Identity Center，则可能得为您 Amazon 企业中的账户启用使用 IAM Identity Center 的应用程序。有关更多信息，请参阅在 Amazon 账户中启用 IAM Identity Center 集成应用程序。
   - 确保您拥有 IAM Identity Center 先决条件页面上列出的先决条件。
2. 按照启用 IAM Identity Center 中的说明在要创建 EMR Studio 的 Amazon Web Services 区域 中启用 IAM Identity Center。
3. 将 IAM Identity Center 连接到您的身份提供商并预调配您要分配给 Studio 的用户和组。

如果您使用...	请执行此操作...
Microsoft AD 目录	1. 按照连接到您的 Microsoft AD 目录中的说明使用 Amazon Directory Service 连接您自行管理的 Active Directory 或 Amazon Managed Microsoft AD 目录。 2. 要为 IAM Identity Center 预置用户和组，您可以将身份数据从源 AD 同步到 IAM Identity Center。您可以通过多种方式同步源 AD 中的身份。一种方法是将 AD 用户或组分配给企业中的 Amazon 账户。有关说明，请参阅单点登录。 同步最多可能需要两个小时。完成此步骤后，同步的用户和组都显示在您的 Identity Store 中。

如果您使用...	请执行此操作...
	Note 在您同步用户和组信息或使用即时（JIT）用户预调配之前，用户和组不会出现在您的 Identity Store 中。有关更多信息，请参阅当用户来自 Active Directory 时进行预置。 3. （可选）同步 AD 用户和组后，您可以删除他们对您在上一步中配置的 Amazon 账户的访问权限。有关说明，请参阅删除用户访问权限。
外部身份提供商	按照连接到您的外部身份提供商中的说明进行操作。
IAM Identity Center 目录	当您在 IAM Identity Center 中创建用户和组时，则会进行自动预调配。有关更多信息，请参阅管理 IAM Identity Center 中的身份。

您现在可以将 Identity Store 中的用户和组分配到 EMR Studio。有关说明，请参阅 将用户或组分配到 EMR Studio。

创建 EMR Studio 服务角色

关于 EMR Studio 服务角色

每个 EMR Studio 都使用一个 IAM 角色，该角色具有允许 Studio 与其它 Amazon 服务进行交互的权限。此服务角色必须包括允许 EMR Studio 在 Workspaces 和集群之间建立安全网络通道、将笔记本文件存储在 Amazon S3 Control 中以及在将 Workspace 链接到 Git 存储库时访问 Amazon Secrets Manager 的权限。

使用 Studio 服务角色（而不是会话策略）定义所有 Amazon S3 访问权限以存储笔记本文件，并定义 Amazon Secrets Manager 访问权限。

如何在 Amazon EC2 或 Amazon EKS 上为 EMR Studio 创建服务角色

1. 按照创建角色以将权限委派给Amazon服务中的说明使用以下信任策略创建服务角色。

> **Important**
>
> 下面的信任策略包括 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件密钥，用于限制您为 EMR Studio 授予的、针对您账户中特定资源的权限。这样做可以防止混淆代理人问题。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
        }
      }
    }
  ]
}
```

2. 删除默认角色权限。然后，包括下面的示例 IAM 权限策略中的权限。或者，您也可以创建使用 EMR Studio 服务角色权限 的自定义策略。

   如果适用，请更改以下策略中的 `"Resource":"*"`，以便为您的使用案例指定该语句涵盖的一个或多个资源的 Amazon Resource Name (ARN)。

Important

- 要访问 `ModifyNetworkInterfaceAttribute` API，以下策略内容必须保持不变，因为受到 Amazon EC2 标记的访问控制以及 EMR Studio 使用 `ModifyNetworkInterfaceAttribute` 方式的技术限制。
- 为了让 EMR Studio 与服务角色 `AllowAddingEMRTagsDuringDefaultSecurityGroupCreation` 和 `AllowAddingTagsDuringEC2ENICreation` 配合使用，以下语句必须保持不变。
- 要您使用示例策略，必须使用密钥 "**`for-use-with-amazon-emr-managed-policies`**" 和值 "**`true`**" 标记以下资源。
  - 用于 EMR Studio 的 Amazon Virtual Private Cloud (VPC)。
  - 您要与 Studio 配合使用的每个子网。
  - 任何自定义 EMR Studio 安全组。如果您想继续使用在 EMR Studio 预览期间创建的安全组，则必须标记它们。
  - 在 Amazon Secrets Manager 中维护的 Studio 用户用于将 Git 存储库链接到 Workspace 的密钥。

  您可以在Amazon Web Services Management Console中，使用相关资源屏幕上的 Tags (标签) 选项卡将标签应用于资源。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEMRReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowEC2ENIActionsWithEMRTags",
      "Effect": "Allow",
```

```
      "Action": [
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowEC2ENIAttributeAction",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifyNetworkInterfaceAttribute"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid": "AllowEC2SecurityGroupActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:DeleteNetworkInterfacePermission"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationWithEMRTags",
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
          "ec2:CreateAction": "CreateSecurityGroup"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationWithEMRTags",
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationInSubnetAndSecurityGroupWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingTagsDuringEC2ENICreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        }
      }
    },
    {
      "Sid": "AllowEC2ReadOnlyActions",
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeTags",
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowWorkspaceCollaboration",
      "Effect": "Allow",
      "Action": [
        "iam:GetUser",
        "iam:GetRole",
        "iam:ListUsers",
        "iam:ListRoles",
        "sso:GetManagedApplicationInstance",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

3. 为您的服务角色授予对用于 EMR Studio 的 Amazon S3 位置的读写权限。使用以下最小权限集。有关更多信息，请参阅 Amazon S3：允许以编程方式和在控制台中对 S3 Bucket 中的对象进行读写访问示例。

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

如果您加密了 Amazon S3 存储桶，您必须为 Amazon Key Management Service包含以下权限。

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

EMR Serverless 最低需要的服务角色

如果要通过 EMR Studio 笔记本电脑实例将交互式工作负载与 EMR Serverless 结合使用，请使用与上一节“如何在 Amazon EC2 或 Amazon EKS 上为 EMR Studio 创建服务角色”中设置 EMR Studio 相同的信任策略。

对于 IAM policy，最低可行策略应具有以下权限。使用您在配置 EMR Studio 和 Workspace 时计划使用的存储桶名称更新 *bucket-name*。EMR Studio 使用存储桶来备份 Studio 中的 Workspaces 和笔记本电脑实例文件。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ObjectActions",
            "Effect": "Allow",
            "Action": [
               "s3:PutObject",
               "s3:GetObject",
               "s3:DeleteObject"
            ],
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        },
        {
            "Sid": "BucketActions",
            "Effect": "Allow",
```

```
            "Action": [
               "s3:ListBucket",
               "s3:GetEncryptionConfiguration"
            ],
            "Resource": ["arn:aws:s3:::bucket-name"]
        }
    ]
}
```

如果您计划使用加密的 Amazon S3 存储桶，则请在策略中添加以下权限：

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

EMR Studio 服务角色权限

下表列出了 EMR Studio 使用服务角色运行的运算，以及每项运算所需的 IAM 操作。

操作	操作

在 Workspace 和 EMR 集群之间建立安全的网络通道，并执行必要的清理操作。	`"ec2:CreateNetworkInterface",`
	`"ec2:CreateNetworkInterfacePermission",`
	`"ec2:DeleteNetworkInterface",`
	`"ec2:DeleteNetworkInterfacePermission",`
	`"ec2:DescribeNetworkInterfaces",`
	`"ec2:ModifyNetworkInterfaceAttribute",`
	`"ec2:AuthorizeSecurityGroupEgress",`
	`"ec2:AuthorizeSecurityGroupIngress",`
	`"ec2:CreateSecurityGroup",`
	`"ec2:DescribeSecurityGroups",`
	`"ec2:RevokeSecurityGroupEgress",`
	`"ec2:DescribeTags",`
	`"ec2:DescribeInstances",`
	`"ec2:DescribeSubnets",`
	`"ec2:DescribeVpcs",`
	`"elasticmapreduce:ListInstances",`
	`"elasticmapreduce:DescribeCluster",`
	`"elasticmapreduce:ListSteps"`

操作	操作
使用存储在 Amazon Secrets Manager 中的 Git 凭证将 Git 存储库链接到 Workspace。	`"secretsmanager:GetSecretValue"`
将Amazon标签应用于 EMR Studio 在设置安全网络通道时创建的网络接口和默认安全组。有关更多信息，请参阅标记Amazon资源。	`"ec2:CreateTags"`

访问笔记本文件和元数据或将它们上传到 Amazon S3。	`"s3:PutObject",`
	`"s3:GetObject",`
	`"s3:GetEncryptionConfiguration",`
	`"s3:ListBucket",`
	`"s3:DeleteObject"`
	
	如果您使用 加密的 Amazon S3 存储桶，则必须包含以下权限。
	
	`"kms:Decrypt",`
	`"kms:GenerateDataKey",`
	`"kms:ReEncryptFrom",`
	`"kms:ReEncryptTo",`
	`"kms:DescribeKey"`

启用和配置 Workspace 协作。	`"iam:GetUser",`
	`"iam:GetRole",`
	`"iam:ListUsers",`
	`"iam:ListRoles",`
	`"sso:GetManagedApplicationInstance",`
	`"sso-directory:SearchUsers"`

为 Amazon EC2 或 Amazon EKS 配置 EMR Studio 用户权限

您必须为 Amazon EMR Studio 配置用户权限策略，以便您能设置精细的用户和组权限。有关用户权限在 EMR Studio 中如何工作的信息，请参阅 Amazon EMR Studio 工作原理 中的 访问控制。

> **Note**
>
> 本部分中介绍的权限不会强制执行数据访问控制。要管理对输入数据集的访问，您应该为 Studio 使用的集群配置权限。有关更多信息，请参阅Amazon EMR 中的安全性。

创建 IAM Identity Center 身份验证模式的 EMR Studio 用户角色

当您使用 IAM Identity Center 身份验证模式时，必须创建 EMR Studio 用户角色。

创建 EMR Studio 用户角色

1. 按照《Amazon Identity and Access Management 用户指南》中的创建向Amazon服务委派权限的角色中的说明，创建用户角色。

   在创建角色时，使用以下信任关系策略。

   ```
   {
     "Version": "2008-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "elasticmapreduce.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

2. 删除默认角色权限和策略。
3. 在将用户和组分配给 Studio 之前，应将您的 EMR Studio 会话策略附加到用户角色。有关如何创建会话策略的说明，请参阅 为 EMR Studio 用户创建权限策略。

为 EMR Studio 用户创建权限策略

完成以下步骤，为 EMR Studio 创建权限策略。

> **Note**
>
> 要设置用于存储笔记本文件的 Amazon S3 访问权限，以及在将 Workspaces 链接到 Git 存储库时读取密钥的 Amazon Secrets Manager 访问权限，请使用 EMR Studio 服务角色。

1. 创建一个或多个指定用户可在您的 Studio 中执行哪些操作的 IAM 权限策略。例如，您可以使用此页面上的示例策略为基本、中间和高级 Studio 用户创建三个独立策略。

   EMR Studio 用户的 Amazon Identity and Access Management 权限表细分了用户可能执行的每个 Studio 运营，并列出了执行该运营所需的最低 IAM 操作。有关说明，请参阅创建 IAM policy。

   您的权限策略必须包含以下语句。

   ```
   {
               "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
               "Effect": "Allow",
               "Action": "secretsmanager:TagResource",
               "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
   },
   {
               "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
               "Action": "iam:PassRole",
               "Resource": [
                   "arn:aws:iam::*:role/<your-emr-studio-service-role>"
               ],
               "Effect": "Allow"
   }
   ```

   设置 Workspace 协作的所有权

   借助 Workspace 协作功能，多个用户可以在同一 Workspace 中同时工作，并且可以使用 Workspace UI 中的 Collaboration（协作）面板进行配置。要查看和使用 Collaboration（协作）面板，用户必须具有以下权限。任何拥有这些权限的用户都可以查看和使用 Collaboration（协作）面板。

   ```
   "elasticmapreduce:UpdateEditor",
   "elasticmapreduce:PutWorkspaceAccess",
   "elasticmapreduce:DeleteWorkspaceAccess",
   "elasticmapreduce:ListWorkspaceAccessIdentities"
   ```

要限制对 Collaboration（协作）面板的访问权限，您可以使用基于标签的访问控制。当用户创建某个 Workspace 时，EMR Studio 会应用一个原定设置标签，标签键为 `creatorUserId`，标签值为创建该 WorkSpace 的用户的 ID。

> **Note**
>
> 对于在 2021 年 11 月 16 日之前创建的 WorkSpace，EMR Studio 并未添加 `creatorUserId` 标签。要限制有权配置协作的用户，我们建议为您的 Workspace 手动添加 `creatorUserId` 标签，然后在用户权限策略中使用基于标签的访问控制。

以下示例语句允许用户为任何具有标签键 `creatorUserId` 并且标签值与该用户的 ID（由策略变量 `aws:userId` 指示）一致的 WorkSpace 配置协作。换言之，该语句允许用户为他们创建的 Workspace 配置协作。要详细了解策略变量，请参阅 IAM policy 元素：变量和标签。

```
{
    "Sid": "UserRolePermissionsForCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
        "elasticmapreduce:DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userid}"
        }
    }
}
```

2. 将权限策略附加到您的 IAM 身份。

下表总结了您将权限策略附加到的具体 IAM 身份，其取决于您的 EMR Studio 身份验证模式。有关如何附加策略的说明，请参阅添加和删除 IAM 身份权限。

如果您使用...	将策略附加到...
IAM 身份验证	您的 IAM 身份（用户、用户组或角色）。例如，您可以将权限策略附加到 Amazon Web Services 账户中的用户。
与外部身份提供商 (IdP) 进行 IAM 联合身份验证	您为外部 IdP 创建的一个或多个 IAM 角色。例如，用于 SAML 2.0 联合身份验证的 IAM。 对于具有 Studio 联合访问权限的用户，EMR Studio 使用您附加到 IAM 单个或多个角色的权限。
IAM Identity Center	您的 Amazon EMR Studio 用户角色。

示例用户策略

以下基本用户策略允许大多数 EMR Studio 操作，但不允许用户创建新的 Amazon EMR 集群。

基本策略

> **Important**
>
> 该示例策略不包括 `CreateStudioPresignedUrl` 权限，当您使用 IAM 身份验证模式时，必须允许用户使用该权限。有关更多信息，请参阅[将用户或组分配到 EMR Studio](#)。

该示例策略包括要强制执行基于标签的访问控制 (TBAC) 的 `Condition` 元素，以便您可以将该策略与 EMR Studio 的示例服务角色配合使用。有关更多信息，请参阅[创建 EMR Studio 服务角色](#)。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSecurityGroup"
         ],
```

```
            "Resource":[
               "arn:aws:ec2:*:*:vpc/*"
            ],
            "Condition":{
               "StringEquals":{
                  "aws:ResourceTag/for-use-with-amazon-emr-managed-policies":"true"
               }
            }
         },
         {
            "Sid":"AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
            "Effect":"Allow",
            "Action":[
               "ec2:CreateTags"
            ],
            "Resource":"arn:aws:ec2:*:*:security-group/*",
            "Condition":{
               "StringEquals":{
                  "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true",
                  "ec2:CreateAction":"CreateSecurityGroup"
               }
            }
         },
         {
            "Sid":"AllowSecretManagerListSecrets",
            "Action":[
               "secretsmanager:ListSecrets"
            ],
            "Resource":"*",
            "Effect":"Allow"
         },
         {
            "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:CreateSecret",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition":{
               "StringEquals":{
                  "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
               }
            }
         },
         {
            "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
```

```
            "Effect":"Allow",
            "Action":"secretsmanager:TagResource",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
        },
        {
            "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
            "Action":"iam:PassRole",
            "Resource":[
                "arn:aws:iam::*:role/<your-emr-studio-service-role>"
            ],
            "Effect":"Allow"
        },
        {
            "Sid":"AllowS3ListAndLocationPermissions",
            "Action":[
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource":"arn:aws:s3:::*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowS3ReadOnlyAccessToLogs",
            "Action":[
                "s3:GetObject"
            ],
            "Resource":[
                "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
            ],
            "Effect":"Allow"
        },
        {
            "Sid":"AllowConfigurationForWorkspaceCollaboration",
            "Action":[
                "elasticmapreduce:UpdateEditor",
                "elasticmapreduce:PutWorkspaceAccess",
                "elasticmapreduce:DeleteWorkspaceAccess",
                "elasticmapreduce:ListWorkspaceAccessIdentities"
            ],
            "Resource":"*",
            "Effect":"Allow",
            "Condition":{
                "StringEquals":{
```

```
                "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
            }
         }
      },
      {
         "Sid":"DescribeNetwork",
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      }
   ]
}
```

以下中间用户策略允许大多数 EMR Studio 操作，并允许用户使用集群模板创建新的 Amazon EMR 集群。

中间策略

> **Important**
>
> 该示例策略不包括 `CreateStudioPresignedUrl` 权限，当您使用 IAM 身份验证模式时，必须允许用户使用该权限。有关更多信息，请参阅将用户或组分配到 EMR Studio。

该示例策略包括要强制执行基于标签的访问控制 (TBAC) 的 `Condition` 元素，以便您可以将该策略与 EMR Studio 的示例服务角色配合使用。有关更多信息，请参阅创建 EMR Studio 服务角色。

```
{
   "Version":"2012-10-17",
   "Statement":[
```

```
    {
       "Sid":"AllowEMRBasicActions",
       "Action":[
          "elasticmapreduce:CreateEditor",
          "elasticmapreduce:DescribeEditor",
          "elasticmapreduce:ListEditors",
          "elasticmapreduce:StartEditor",
          "elasticmapreduce:StopEditor",
          "elasticmapreduce:DeleteEditor",
          "elasticmapreduce:OpenEditorInConsole",
          "elasticmapreduce:AttachEditor",
          "elasticmapreduce:DetachEditor",
          "elasticmapreduce:CreateRepository",
          "elasticmapreduce:DescribeRepository",
          "elasticmapreduce:DeleteRepository",
          "elasticmapreduce:ListRepositories",
          "elasticmapreduce:LinkRepository",
          "elasticmapreduce:UnlinkRepository",
          "elasticmapreduce:DescribeCluster",
          "elasticmapreduce:ListInstanceGroups",
          "elasticmapreduce:ListBootstrapActions",
          "elasticmapreduce:ListClusters",
          "elasticmapreduce:ListSteps",
          "elasticmapreduce:CreatePersistentAppUI",
          "elasticmapreduce:DescribePersistentAppUI",
          "elasticmapreduce:GetPersistentAppUIPresignedURL",
          "elasticmapreduce:GetOnClusterAppUIPresignedURL"
       ],
       "Resource":"*",
       "Effect":"Allow"
    },
    {
       "Sid":"AllowEMRContainersBasicActions",
       "Action":[
          "emr-containers:DescribeVirtualCluster",
          "emr-containers:ListVirtualClusters",
          "emr-containers:DescribeManagedEndpoint",
          "emr-containers:ListManagedEndpoints",
          "emr-containers:DescribeJobRun",
          "emr-containers:ListJobRuns"
       ],
       "Resource":"*",
       "Effect":"Allow"
    },
```

```
      {
          "Sid": "AllowRetrievingManagedEndpointCredentials",
          "Effect": "Allow",
          "Action": [
              "emr-containers:GetManagedEndpointSessionCredentials"
          ],
          "Resource": [
              "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
          ],
          "Condition": {
              "StringEquals": {
                  "emr-containers:ExecutionRoleArn": [
                      "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
                  ]
              }
          }
      },
      {
         "Sid":"AllowSecretManagerListSecrets",
         "Action":[
            "secretsmanager:ListSecrets"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
         "Effect":"Allow",
         "Action":"secretsmanager:CreateSecret",
         "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
            }
         }
      },
      {
         "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
         "Effect":"Allow",
         "Action":"secretsmanager:TagResource",
         "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
      },
      {
```

```
        "Sid":"AllowClusterTemplateRelatedIntermediateActions",
        "Action":[
           "servicecatalog:DescribeProduct",
           "servicecatalog:DescribeProductView",
           "servicecatalog:DescribeProvisioningParameters",
           "servicecatalog:ProvisionProduct",
           "servicecatalog:SearchProducts",
           "servicecatalog:UpdateProvisionedProduct",
           "servicecatalog:ListProvisioningArtifacts",
           "servicecatalog:ListLaunchPaths",
           "servicecatalog:DescribeRecord",
           "cloudformation:DescribeStackResources"
        ],
        "Resource":"*",
        "Effect":"Allow"
     },
     {
        "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
        "Action":"iam:PassRole",
        "Resource":[
           "arn:aws:iam::*:role/<your-emr-studio-service-role>"
        ],
        "Effect":"Allow"
     },
     {
        "Sid":"AllowS3ListAndLocationPermissions",
        "Action":[
           "s3:ListAllMyBuckets",
           "s3:ListBucket",
           "s3:GetBucketLocation"
        ],
        "Resource":"arn:aws:s3:::*",
        "Effect":"Allow"
     },
     {
        "Sid":"AllowS3ReadOnlyAccessToLogs",
        "Action":[
           "s3:GetObject"
        ],
        "Resource":[
           "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
        ],
        "Effect":"Allow"
     },
```

```
    {
       "Sid":"AllowConfigurationForWorkspaceCollaboration",
       "Action":[
          "elasticmapreduce:UpdateEditor",
          "elasticmapreduce:PutWorkspaceAccess",
          "elasticmapreduce:DeleteWorkspaceAccess",
          "elasticmapreduce:ListWorkspaceAccessIdentities"
       ],
       "Resource":"*",
       "Effect":"Allow",
       "Condition":{
          "StringEquals":{
             "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
          }
       }
    },
    {
       "Sid":"DescribeNetwork",
       "Effect":"Allow",
       "Action":[
          "ec2:DescribeVpcs",
          "ec2:DescribeSubnets",
          "ec2:DescribeSecurityGroups"
       ],
       "Resource":"*"
    },
    {
       "Sid":"ListIAMRoles",
       "Effect":"Allow",
       "Action":[
          "iam:ListRoles"
       ],
       "Resource":"*"
    },
    {
       "Sid": "AllowServerlessActions",
       "Action": [
          "emr-serverless:CreateApplication",
          "emr-serverless:UpdateApplication",
          "emr-serverless:DeleteApplication",
          "emr-serverless:ListApplications",
          "emr-serverless:GetApplication",
          "emr-serverless:StartApplication",
          "emr-serverless:StopApplication",
```

```
            "emr-serverless:StartJobRun",
            "emr-serverless:CancelJobRun",
            "emr-serverless:ListJobRuns",
            "emr-serverless:GetJobRun",
            "emr-serverless:GetDashboardForJobRun",
            "emr-serverless:AccessInteractiveEndpoints"
         ],
         "Resource": "*",
         "Effect": "Allow"
        },
        {
           "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
           "Action": "iam:PassRole",
           "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
           "Effect": "Allow"
        }
    ]
}
```

以下高级用户策略允许所有 EMR Studio 操作，并允许用户使用集群模板或通过提供集群配置创建新的 Amazon EMR 集群。

高级策略

> **Important**
>
> 该示例策略不包括 `CreateStudioPresignedUrl` 权限，当您使用 IAM 身份验证模式时，必须允许用户使用该权限。有关更多信息，请参阅将用户或组分配到 EMR Studio。

该示例策略包括要强制执行基于标签的访问控制 (TBAC) 的 `Condition` 元素，以便您可以将该策略与 EMR Studio 的示例服务角色配合使用。有关更多信息，请参阅创建 EMR Studio 服务角色。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowEMRBasicActions",
         "Action":[
            "elasticmapreduce:CreateEditor",
            "elasticmapreduce:DescribeEditor",
            "elasticmapreduce:ListEditors",
```

```
            "elasticmapreduce:StartEditor",
            "elasticmapreduce:StopEditor",
            "elasticmapreduce:DeleteEditor",
            "elasticmapreduce:OpenEditorInConsole",
            "elasticmapreduce:AttachEditor",
            "elasticmapreduce:DetachEditor",
            "elasticmapreduce:CreateRepository",
            "elasticmapreduce:DescribeRepository",
            "elasticmapreduce:DeleteRepository",
            "elasticmapreduce:ListRepositories",
            "elasticmapreduce:LinkRepository",
            "elasticmapreduce:UnlinkRepository",
            "elasticmapreduce:DescribeCluster",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListSteps",
            "elasticmapreduce:CreatePersistentAppUI",
            "elasticmapreduce:DescribePersistentAppUI",
            "elasticmapreduce:GetPersistentAppUIPresignedURL",
            "elasticmapreduce:GetOnClusterAppUIPresignedURL"
        ],
        "Resource":"*",
        "Effect":"Allow"
    },
    {
        "Sid":"AllowEMRContainersBasicActions",
        "Action":[
            "emr-containers:DescribeVirtualCluster",
            "emr-containers:ListVirtualClusters",
            "emr-containers:DescribeManagedEndpoint",
            "emr-containers:ListManagedEndpoints",
            "emr-containers:DescribeJobRun",
            "emr-containers:ListJobRuns"
        ],
        "Resource":"*",
        "Effect":"Allow"
    },
    {
        "Sid": "AllowRetrievingManagedEndpointCredentials",
        "Effect": "Allow",
        "Action": [
            "emr-containers:GetManagedEndpointSessionCredentials"
        ],
```

```
            "Resource": [
                "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
            ],
            "Condition": {
                "StringEquals": {
                    "emr-containers:ExecutionRoleArn": [
                        "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
                    ]
                }
            }
        },
        {
            "Sid":"AllowSecretManagerListSecrets",
            "Action":[
               "secretsmanager:ListSecrets"
            ],
            "Resource":"*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:CreateSecret",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition":{
               "StringEquals":{
                  "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
               }
            }
        },
        {
            "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:TagResource",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
        },
        {
            "Sid":"AllowClusterTemplateRelatedIntermediateActions",
            "Action":[
               "servicecatalog:DescribeProduct",
               "servicecatalog:DescribeProductView",
               "servicecatalog:DescribeProvisioningParameters",
               "servicecatalog:ProvisionProduct",
```

```
            "servicecatalog:SearchProducts",
            "servicecatalog:UpdateProvisionedProduct",
            "servicecatalog:ListProvisioningArtifacts",
            "servicecatalog:ListLaunchPaths",
            "servicecatalog:DescribeRecord",
            "cloudformation:DescribeStackResources"
        ],
        "Resource":"*",
        "Effect":"Allow"
    },
    {
        "Sid":"AllowEMRCreateClusterAdvancedActions",
        "Action":[
            "elasticmapreduce:RunJobFlow"
        ],
        "Resource":"*",
        "Effect":"Allow"
    },
    {
        "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
        "Action":"iam:PassRole",
        "Resource":[
            "arn:aws:iam::*:role/<your-emr-studio-service-role>",
            "arn:aws:iam::*:role/EMR_DefaultRole_V2",
            "arn:aws:iam::*:role/EMR_EC2_DefaultRole"
        ],
        "Effect":"Allow"
    },
    {
        "Sid":"AllowS3ListAndLocationPermissions",
        "Action":[
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Resource":"arn:aws:s3:::*",
        "Effect":"Allow"
    },
    {
        "Sid":"AllowS3ReadOnlyAccessToLogs",
        "Action":[
            "s3:GetObject"
        ],
        "Resource":[
```

```
            "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
         ],
         "Effect":"Allow"
      },
      {
         "Sid":"AllowConfigurationForWorkspaceCollaboration",
         "Action":[
            "elasticmapreduce:UpdateEditor",
            "elasticmapreduce:PutWorkspaceAccess",
            "elasticmapreduce:DeleteWorkspaceAccess",
            "elasticmapreduce:ListWorkspaceAccessIdentities"
         ],
         "Resource":"*",
         "Effect":"Allow",
         "Condition":{
            "StringEquals":{
               "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
            }
         }
      },
      {
         "Sid" : "SageMakerDataWranglerForEMRStudio",
         "Effect" : "Allow",
         "Action" : [
            "sagemaker:CreatePresignedDomainUrl",
            "sagemaker:DescribeDomain",
            "sagemaker:ListDomains",
            "sagemaker:ListUserProfiles"
         ],
         "Resource":"*"
      },
      {
         "Sid":"DescribeNetwork",
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
```

```
            "Action":[
               "iam:ListRoles"
            ],
            "Resource":"*"
         },
         {
            "Sid": "AllowServerlessActions",
            "Action": [
               "emr-serverless:CreateApplication",
               "emr-serverless:UpdateApplication",
               "emr-serverless:DeleteApplication",
               "emr-serverless:ListApplications",
               "emr-serverless:GetApplication",
               "emr-serverless:StartApplication",
               "emr-serverless:StopApplication",
               "emr-serverless:StartJobRun",
               "emr-serverless:CancelJobRun",
               "emr-serverless:ListJobRuns",
               "emr-serverless:GetJobRun",
               "emr-serverless:GetDashboardForJobRun",
               "emr-serverless:AccessInteractiveEndpoints"
            ],
            "Resource": "*",
            "Effect": "Allow"
         },
         {
            "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
            "Effect": "Allow"
         }
      ]
}
```

以下用户策略包含将 EMR Serverless 交互式应用程序与 EMR Studio Workspace 结合使用所需的最低用户权限。

EMR Serveless 交互式策略

此示例策略具有将 EMR Serverless 交互式应用程序与 EMR Studio 结合使用所需的用户权限，请将 *serverless-runtime-role* 和 *emr-studio-service-role* 的占位符分别替换为正确的 EMR Studio 服务角色 和 EMR Serverless 运行时系统角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowServerlessActions",
            "Action": [
                "emr-serverless:CreateApplication",
                "emr-serverless:UpdateApplication",
                "emr-serverless:DeleteApplication",
                "emr-serverless:ListApplications",
                "emr-serverless:GetApplication",
                "emr-serverless:StartApplication",
                "emr-serverless:StopApplication",
                "emr-serverless:StartJobRun",
                "emr-serverless:CancelJobRun",
                "emr-serverless:ListJobRuns",
                "emr-serverless:GetJobRun",
                "emr-serverless:GetDashboardForJobRun",
                "emr-serverless:AccessInteractiveEndpoints"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowEMRBasicActions",
            "Action": [
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:UpdateStudio",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:AttachEditor",
                "elasticmapreduce:DetachEditor",
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio",
                "elasticmapreduce:ListStudios",
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "*",
```

```
            "Effect": "Allow"
        },
        {
            "Sid": "AllowPassingRuntimeRoleForRunningEMRServerlessJob",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/emr-studio-service-role",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowS3ListAndGetPermissions",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
        {
            "Sid":"DescribeNetwork",
            "Effect":"Allow",
            "Action":[
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource":"*"
        },
        {
            "Sid": "ListIAMRoles",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

EMR Studio 用户的 Amazon Identity and Access Management 权限

下表包括用户可能执行的每个 Amazon EMR Studio 运营，并列出了执行该运营所需的最低 IAM 操作。您允许在 IAM 权限策略（当您使用 IAM 身份验证时）或会话策略（当您使用 IAM Identity Center 身份验证时）中执行这些针对 EMR Studio 的操作。

该表还显示了 EMR Studio 的每个示例权限策略中允许的运营。有关示例权限策略的更多信息，请参阅为 EMR Studio 用户创建权限策略。

操作	基本	中间	高级	关联操作
创建和删除 Workspaces	是	是	是	`"elasticmapreduce:CreateEditor", "elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:DeleteEditor"`
查看协作面板、启用 Workspace 协作并添加协作者。有关更多信息，请参阅设置 Workspace 协作的所有权。	是	是	是	`"elasticmapreduce:UpdateEditor", "elasticmapreduce:PutWorkspaceAccess", "elasticmapreduce:DeleteWorkspaceAccess", "elasticmapreduce:ListWorkspaceAccessIdentities"`
创建新 EMR 集群时查看与 Studio 相同账户中的 Amazon S3 Control 存储存储桶列表，并在使用 Web UI 调试应用程序时访问容器日志	是	是	是	`"s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetBucketLocation", "s3:GetObject"`

操作	基本	中间	高级	关联操作
访问 Workspaces	是	是	是	`"elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:StartEditor", "elasticmapreduce:StopEditor", "elasticmapreduce:OpenEditorInConsole"`
附加或分离与 Workspace 关联的现有 Amazon EMR 集群	是	是	是	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListInstanceGroups", "elasticmapreduce:ListBootstrapActions"`

操作	基本	中间	高级	关联操作
附加或分离 Amazon EMR on EKS 集群	是	是	是	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListManagedEndpoints", "emr-containers:DescribeManagedEndpoint", "emr-containers:GetManagedEndpointSessionCredentials"`
附加或分离与 Workspace 关联的 EMR Serverless 应用程序	否	是	是	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "emr-serverless:GetApplication", "emr-serverless:StartApplication", "emr-serverless:ListApplications", "emr-serverless:GetDashboardForJobRun", "emr-serverless:AccessInteractiveEndpoints", "iam:PassRole"` 需要具有 `PassRole` 权限才能传递 EMR Serverless 作业运行时系统角色。有关更多信息，请参阅《Amazon EMR Serverless 用户指南》中的 Job runtime roles。

操作	基本	中间	高级	关联操作
使用持久的应用程序用户界面调试 Amazon EMR on EC2 任务	是	是	是	`"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "elasticmapreduce:ListClusters", "elasticmapreduce:ListSteps", "elasticmapreduce:DescribeCluster", "s3:ListBucket", "s3:GetObject"`
使用集群上应用程序用户界面调试 Amazon EMR on EC2 任务	是	是	是	`"elasticmapreduce:GetOnClusterAppUIPresignedURL"`
使用 Spark 历史记录服务器调试 Amazon EMR on EKS 任务运行	是	是	是	`"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListJobRuns", "emr-containers:DescribeJobRun", "s3:ListBucket", "s3:GetObject"`

操作	基本	中间	高级	关联操作
创建和删除 Git 存储库	是	是	是	`"elasticmapreduce:CreateRepository", "elasticmapreduce:DeleteRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository", "secretsmanager:CreateSecret", "secretsmanager:ListSecrets", "secretsmanager:TagResource"`
链接和取消链接 Git 存储库	是	是	是	`"elasticmapreduce:LinkRepository", "elasticmapreduce:UnlinkRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository"`

操作	基本	中间	高级	关联操作
根据预定义的集群模板创建新集群	否	是	是	`"servicecatalog:SearchProducts", "servicecatalog:DescribeProduct", "servicecatalog:DescribeProductView", "servicecatalog:DescribeProvisioningParameters", "servicecatalog:ProvisionProduct", "servicecatalog:UpdateProvisionedProduct", "servicecatalog:ListProvisioningArtifacts", "servicecatalog:DescribeRecord", "servicecatalog:ListLaunchPaths", "cloudformation:DescribeStackResources", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"`
通过提供集群配置创建新的集群	否	否	是	`"elasticmapreduce:RunJobFlow", "iam:PassRole", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"`
当您使用 IAM 身份验证模式时，应将用户分配给 Studio。有关更多信息，请参阅[将用户或组分配到 EMR Studio](#)。	否	否	否	`"elasticmapreduce:CreateStudioPresignedUrl"`

操作	基本	中间	高级	关联操作

描述网络对象。	是	是	是	<pre>{
				    "Version": "2012-10-17",
				    "Statement": [
				        {
				            "Sid": "DescribeNetwork",
				            "Effect": "Allow",
				            "Action": [
				                "ec2:DescribeVpcs",
				                "ec2:DescribeSubnets",
				                "ec2:DescribeSecurityGroups"
				            ],
				            "Resource": "*"
				        }
				    ]
				}</pre>

操作	基本	中间	高级	关联操作
列出 IAM 角色。	是	是	是	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "ListIAMRoles", "Effect": "Allow", "Action": [ "iam:ListRoles" ], "Resource": "*" } ] }`
从 Amazon SageMaker Studio 连接到 EMR Studio，然后使用 Data Wrangler 可视化界面。	否	否	是	`"sagemaker:CreatePresignedDomainUrl", "sagemaker:DescribeDomain", "sagemaker:ListDomains", "sagemaker:ListUserProfiles"`

创建 EMR Studio

您可以使用 Amazon EMR 控制台或 Amazon CLI为您的团队创建 EMR Studio。创建 Studio 实例是设置 Amazon EMR Studio 的一部分。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

先决条件

在创建 Studio 之前，请确保您已在 设置 Amazon EMR Studio 中完成了之前的任务。

要使用 Amazon CLI 创建 Studio，您应该已经安装了最新版本。有关更多信息，请参阅安装或更新 Amazon CLI 的最新版本。

> **Important**
>
> 在创建 Studio 之前，请确保在浏览器中停用 FoxyProxy 或 SwitchyOmega 等代理管理工具。当您选择 Create Studio（创建 Studio）时，活动代理可能会导致出现 Network Failure（网络故障）错误消息。

New console

使用新 EMR 控制台创建 EMR Studio

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/emr。
2. 在左侧导航栏的 EMR Studio 下，选择 Getting started（入门）。您还可以从 Studios 页面创建新 Studios。
3. 选择创建 Studio，打开创建 Studio 页面。
4. 输入 Studio name (Studio 名称) 和可选 Description (描述)。
5. 如果您针对 Studio 使用 IAM 身份验证，则可以选择添加新标签将您选择的一个或多个键值标签添加到 Studio。您可以使用标签向指定用户授予访问 Studio 的权限。有关更多信息，请参阅将用户或组分配到 EMR Studio。

   您还可以添加标签以帮助您管理、识别、组织和筛选 Studios。有关更多信息，请参阅标记Amazon资源。
6. 在 Networking (联网) 下，从下拉列表中为 Studio 选择 Amazon Virtual Private Cloud (VPC)。
7. 在 Subnets (子网) 下，在您的 VPC 中选择最多五个子网以与 Studio 关联。创建 Studio 后，您可以选择添加更多子网。
8. 对于 Security groups (安全组)，选择默认安全组或自定义安全组。有关更多信息，请参阅定义安全组以控制 EMR Studio 网络流量。

如果选择...	请执行此操作...
默认 EMR Studio 安全组	要为 Studio 启用基于 Git 的存储库链接，请选择 Enable clusters/endpoints and Git repository (启用集群/终端节点和 Git 存储库)。否则，请选择 Enable clusters/endpoints (启用集群/终端节点)。
您 Studio 的自定义安全组	• 在 Cluster/endpoint security group (集群/端点安全组) 下，从下拉列表中选择您配置的引擎安全组。您的 Studio 使用此安全组来允许来自附加 Workspaces 的入站访问。 • 在 Workspace security group (Workspace 安全组) 下，从下拉列表中选择您配置的 Workspace 安全组。您的 Studio 使用此安全组和 Workspaces 来提供对附加 Amazon EMR 集群和公共托管 Git 存储库的出站访问。

9. 在身份验证中，为 Studio 选择身份验证模式并根据下表提供信息。要了解有关 EMR Studio 身份验证的更多信息，请参阅 为 Amazon EMR Studio 选择身份验证模式。

如果您使用...	请执行此操作...
IAM 身份验证或联合身份验证	选择 Studio 的登录方法。 如果您希望联合身份用户使用 Studio URL 和您的身份提供商 (IdP) 的凭据登录，请从下拉列表中选择您的 IdP，然后输入您的身份提供商 (IdP) 登录 URL以及RelayState参数名称。 有关 IdP 身份验证 URL 和 RelayState 名称的列表，请参阅 身份提供商 RelayState 参数和身份验证 URL。

如果您使用...	请执行此操作...
	然后，从下拉列表选择您的 EMR Studio 服务角色。有关更多信息，请参阅创建 EMR Studio 服务角色。
IAM Identity Center 身份验证	选择您的 EMR Studio服务角色和用户角色。有关更多信息，请参阅 创建 EMR Studio 服务角色 和 创建 IAM Identity Center 身份验证模式的 EMR Studio 用户角色。

10. 在 Workspace storage（Workspace 存储）下，选择 Browse S3（浏览 S3），以选择用于备份 Workspace 和笔记本文件的 Amazon S3 存储桶。

> **Note**
>
> 您的 EMR Studio 服务角色必须对您选择的存储桶具有读写访问权限。

11. 选择创建 Studio，完成并导航到 Studio 页面。您的新 Studio 在列表中，其中包含 Studio name (Studio 名称)、Creation date (日期创建) 和 Studio access URL (Studio 访问 URL) 等详细信息。

创建 Studio 后，请按照将用户或组分配到 EMR Studio中的说明进行操作。

Old console

使用旧控制台创建 EMR Studio

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/home。
2. 从左侧导航中选择 EMR Studio。
3. 选择创建 Studio，打开创建 Studio 页面。
4. 输入 Studio name (Studio 名称) 和可选 Description (描述)。
5. 如果您针对 Studio 使用 IAM 身份验证，则可以选择添加新标签将您选择的一个或多个键值标签添加到 Studio。您可以使用标签向指定用户授予访问 Studio 的权限。有关更多信息，请参阅将用户或组分配到 EMR Studio。

您还可以添加标签以帮助您管理、识别、组织和筛选 Studios。有关更多信息，请参阅标记 Amazon资源。

6. 在 Networking (联网) 下，从下拉列表中为 Studio 选择 Amazon Virtual Private Cloud (VPC)。
7. 在 Subnets (子网) 下，在您的 VPC 中选择最多五个子网以与 Studio 关联。创建 Studio 后，您可以选择添加更多子网。
8. 对于 Security groups (安全组)，选择默认安全组或自定义安全组。有关更多信息，请参阅定义安全组以控制 EMR Studio 网络流量。

如果选择...	请执行此操作...
默认 EMR Studio 安全组	要为 Studio 启用基于 Git 的存储库链接，请选择 Enable clusters/endpoints and Git repository (启用集群/终端节点和 Git 存储库)。否则，请选择 Enable clusters/endpoints (启用集群/终端节点)。
您 Studio 的自定义安全组	• 在 Cluster/endpoint security group (集群/端点安全组) 下，从下拉列表中选择您配置的引擎安全组。您的 Studio 使用此安全组来允许来自附加 Workspaces 的入站访问。 • 在 Workspace security group (Workspace 安全组) 下，从下拉列表中选择您配置的 Workspace 安全组。您的 Studio 使用此安全组和 Workspaces 来提供对附加 Amazon EMR 集群和公共托管 Git 存储库的出站访问。

9. 在身份验证中，为 Studio 选择身份验证模式并根据下表提供信息。要了解有关 EMR Studio 身份验证的更多信息，请参阅 为 Amazon EMR Studio 选择身份验证模式。

如果您使用...	请执行此操作...
IAM 身份验证或联合身份验证	选择 Studio 的登录方法。

如果您使用...	请执行此操作...
	如果您希望联合身份用户使用 Studio URL 和您的身份提供商 (IdP) 的凭据登录，请从下拉列表中选择您的 IdP，然后输入您的身份提供商 (IdP) 登录 URL以及RelayState参数名称。 有关 IdP 身份验证 URL 和 RelayState 名称的列表，请参阅 身份提供商 RelayState 参数和身份验证 URL。 然后，从下拉列表选择您的 EMR Studio 服务角色。有关更多信息，请参阅创建 EMR Studio 服务角色。
IAM Identity Center 身份验证	选择您的 EMR Studio服务角色和用户角色。有关更多信息，请参阅 创建 EMR Studio 服务角色 和 创建 IAM Identity Center 身份验证模式的 EMR Studio 用户角色。

10. 在 Workspace storage（Workspace 存储）下，选择 Browse S3（浏览 S3），以选择用于备份 Workspace 和笔记本文件的 Amazon S3 存储桶。

> **Note**
>
> 您的 EMR Studio 服务角色必须对您选择的存储桶具有读写访问权限。

11. 选择创建 Studio，完成并导航到 Studio 页面。您的新 Studio 在列表中，其中包含 Studio name (Studio 名称)、Creation date (日期创建) 和 Studio access URL (Studio 访问 URL) 等详细信息。

创建 Studio 后，请按照将用户或组分配到 EMR Studio中的说明进行操作。

CLI

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^)。

Example 用于创建具有 IAM 身份验证模式的 EMR Studio 的 CLI 命令

下面的示例 Amazon CLI 命令将创建使用 IAM 身份验证模式的 EMR Studio。当您为 Studio 使用 IAM 身份验证或联合身份验证时，您不需要指定 `--user-role`。

要允许联合身份用户使用 Studio URL 和身份提供商 (IdP) 的凭据登录，请指定您的 `--idp-auth-url` 和 `--idp-relay-state-parameter-name`。有关 IdP 身份验证 URL 和 RelayState 名称的列表，请参阅 身份提供商 RelayState 参数和身份验证 URL。

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode IAM \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5>  \
--service-role <example-studio-service-role-name> \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location> \
--idp-auth-url <https://EXAMPLE/login/> \
--idp-relay-state-parameter-name <example-RelayState>
```

Example 用于创建具有 IAM Identity Center 身份验证模式的 EMR Studio 的 CLI 命令

以下 Amazon CLI 示例命令创建使用 IAM Identity Center 身份验证模式的 EMR Studio。当您使用 IAM Identity Center 身份验证时，则必须指定 `--user-role`。

有关 IAM Identity Center 身份验证的更多信息，请参阅 为 Amazon EMR Studio 设置 IAM Identity Center 身份验证模式。

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode SSO \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5>  \
```

```
--service-role <example-studio-service-role-name> \
--user-role <example-studio-user-role-name> \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location>
```

Example 用于**aws emr create-studio**的 CLI 输出

以下是您创建 Studio 之后出现的输出示例。

```
{
    StudioId: "es-123XXXXXXXXX",
    Url: "https://es-123XXXXXXXXX.emrstudio-prod.us-east-1.amazonaws.com"
}
```

有关 `create-studio` 命令的更多信息，请参阅 Amazon CLI 命令参考。

身份提供商 RelayState 参数和身份验证 URL

当您使用 IAM 联合身份验证，并且希望用户使用您的 Studio URL 和身份提供商 (IdP) 的凭证登录时，您可以指定身份提供商 (IdP) 登录 URL 和当您创建 EMR Studio时的 RelayState 参数名称。

下表列出了一些常用身份提供商的标准身份验证 URL 和 RelayState 参数名称。

身份提供商	参数	身份验证 URL
Auth0	RelayState	https://*<sub_domain>* .auth0.com/samlp/*<app_id>*
Google 账户	RelayState	https://accounts.google.com/o/saml2/initsso?idpid= *<idp_id>*&spid=*<sp_id>*&forceauthn=false
Microsoft Azure	RelayState	https://myapps.microsoft.com/signin/ *<app_name>* /*<app_id>*?tenantId= *<tenant_id>*
Okta	RelayState	https://*<sub_domain>* .okta.com/app/*<app_name>* /*<app_id>*/sso/saml

身份提供商	参数	身份验证 URL
PingFederate	`TargetResource`	`https://<host>/idp/<idp_id>/startSSO.ping?PartnerSpId=<sp_id>`
PingOne	`TargetResource`	`https://sso.connect.pingidentity.com/sso/sp/initsso?saasid=<app_id>&idpid=<idp_id>`

分配和管理 EMR Studio 用户

创建 EMR Studio 后，您可以分配用户和组到 EMR Studio。分配、更新和删除用户的方法取决于 Studio 身份验证模式。

- 当您使用 IAM 身份验证模式时，可以在 IAM 中或使用 IAM 和身份提供商配置 EMR Studio 用户任务和权限。
- 通过 IAM Identity Center 身份验证模式，您可以使用 Amazon EMR 管理控制台或 Amazon CLI 管理用户。

要了解有关 Amazon EMR Studio 身份验证的更多信息，请参阅 为 Amazon EMR Studio 选择身份验证模式。

将用户或组分配到 EMR Studio

IAM

当您使用 为 Amazon EMR Studio 设置 IAM 身份验证模式 时，必须允许 `CreateStudioPresignedUrl` 在用户的 IAM 权限策略中执行操作，并将用户限制到某一特定 Studio。您可以将 `CreateStudioPresignedUrl` 包括在您的 IAM 身份验证模式的用户权限 中，或者使用单独的策略。

要将用户限制到某一 Studio（或某一组 Studio），您可以使用基于属性的访问控制 (ABAC)，或在权限策略的 `Resource` 元素中指定 Studio 的 Amazon Resource Name (ARN)。

Example 使用 Studio ARN 将用户分配到 Studio

以下示例策略允许用户访问特定 EMR Studio，方法是允许 `CreateStudioPresignedUrl` 操作并在 `Resource` 元素指定 Studio 的 Amazon Resource Name (ARN)。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-
id>:studio/<studio-id>"
        }
    ]
}
```

Example 使用 ABAC 将用户分配至 Studio 以进行 IAM 身份验证

为 Studio 配置基于属性的访问控制 (ABAC) 的方法有多种。例如，您可以将一个或多个标签附加到 EMR Studio，然后创建 IAM policy，该策略限制附有此类标签的特定或一组 Studio 进行 `CreateStudioPresignedUrl` 操作。

您可以在 Studio 创建期间或之后添加标签。要将标签添加到现有 Studio，您可以使用 Amazon CLI`emr add-tags` 命令。以下示例将带有键值对的标签 `Team = Data Analytics` 添加到 EMR Studio。

```
aws emr add-tags --resource-id <example-studio-id> --tags Team="Data Analytics"
```

以下示例权限策略允许 `CreateStudioPresignedUrl` 使用标签键值对 `Team = DataAnalytics` 为 EMR Studios 执行的操作。有关使用标签进行访问控制的更多信息，请参阅使用标签控制对 IAM 用户和角色的访问以及他们进行的访问或使用标签控制对 Amazon 资源的访问。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
```

```
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/Team": "Data Analytics"
                }
            }
        }
    ]
}
```

Example 使用 aws:SourceIdentity 全局条件密钥将用户分配给 Studio

在使用 IAM 联合身份验证时，您可以使用权限策略的全局条件密钥 (global condition key)`aws:SourceIdentity`，以便在用户担任 IAM 角色进行联合身份验证时向他们授予 Studio 访问权限。

您必须先配置身份提供商 (IdP)，以便在用户对联合身份进行身份验证并担任 IAM 角色时返回标识字符串，例如：电子邮件地址或用户名。IAM 将全局条件密钥 (global condition key)`aws:SourceIdentity` 设置为身份提供商 (IdP) 返回的标识字符串。

有关更多信息，请参阅 Amazon 安全博客中的如何将 IAM 角色活动与公司身份关联起来的博客文章，以及全局条件密钥参考中的 aws:SourceIdentity 条目。

以下示例策略允许 `CreateStudioPresignedUrl` 操作并为用户提供 `aws:SourceIdentity`，匹配*<example-source-identity>*访问由 *<example-studio-arn>*指定的 EMR Studio。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticmapreduce:CreateStudioPresignedUrl",
      "Resource": "<example-studio-arn>",
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": "<example-source-identity>"
        }
      }
    }
  ]
}
```

IAM Identity Center

将用户或组分配给 EMR Studio 时，请指定会话策略来定义精细权限，例如为该用户或组创建新 EMR 集群的能力。Amazon EMR 存储这些会话策略映射。您可以在分配后更新用户或组的会话策略。

> **Note**
>
> 用户或组的最终权限是 EMR Studio 用户角色中定义的权限与用户或组会话策略中定义的权限的交集。如果用户属于分配给 Studio 的多个组，EMR Studio 则将为该用户使用权限联合。

使用 Amazon EMR 控制台将用户或组分配到 EMR Studio

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 从左侧导航中选择 EMR Studio。
3. 从 Studios 列表选择您的 Studio 名称，或选择 Studio 后选择 View details (查看详细信息)，以打开 Studio 详细信息页面。
4. 选择 Add Users (添加用户) 以查看 Users (用户) 和 Groups (组) 搜索表。
5. 选择 Users (用户) 选项卡或 Groups (组) 选项卡，然后在搜索栏中输入搜索词以查找用户或组。
6. 从搜索结果列表中选择一个或多个用户或组。您可以在 Users (用户) 选项卡和 Groups (组) 选项卡之间来回切换。
7. 选择要添加到 Studio 的用户和组后，选择 Add (添加)。您应该会看到 Studio users (Studio 用户) 列表中显示用户和组。列表可能需要几秒钟才能刷新。
8. 按照更新分配给 Studio 的用户或组的权限中的说明优化用户或组的 Studio 权限。

使用Amazon CLI，将用户或组分配到 EMR Studio

为以下 `create-studio-session-mapping` 参数插入您自己的值。有关 `create-studio-session-mapping` 命令的更多信息，请参阅 Amazon CLI 命令参考。

- **`--studio-id`** – 要将用户或组分配到 Studio 的 ID。有关如何检索 Studio ID 的说明，请参阅 查看 Studio 详细信息。

- **--identity-name** – Identity Store 中的用户或组的名称。有关更多信息，请参阅《 Identity Store API 参考》中的用户的 UserName 和组的 DisplayName。
- **--identity-type** – 使用 USER 或 GROUP 指定身份类型。
- **--session-policy-arn** – 您要与用户或组关联的会话策略的 Amazon Resource Name（ ARN ）。例如，**arn:aws:iam::*<aws-account-id>*:policy/*EMRStudio_Advanced_User_Policy***。有关更多信息，请参阅 为 EMR Studio 用户创建权限策略。

```
aws emr create-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-name <example-identity-name> \
 --identity-type <USER-or-GROUP> \
 --session-policy-arn <example-session-policy-arn>
```

> **Note**
>
> 为了便于读取，包含 Linux 行继续符（ \ ）。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号（ ^ ）。

使用 `get-studio-session-mapping` 命令验证新分配。将 *<example-identity-name>* 替换为您更新的用户或组的 IAM Identity Center 名称。

```
aws emr get-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <user-or-group-name> \
```

更新分配给 Studio 的用户或组的权限

IAM

要在使用 IAM 身份验证模式时更新用户或组权限，请使用 IAM 更改附加到 IAM 身份（ 用户、组或角色 ）的 IAM 权限策略。

有关更多信息，请参阅 IAM 身份验证模式的用户权限。

IAM Identity Center

使用控制台更新用户或组的 EMR Studio 权限

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 从左侧导航中选择 EMR Studio。
3. 从 Studios 列表选择您的 Studio 名称，或选择 Studio 后选择 View details (查看详细信息)，以打开 Studio 详细信息页面。
4. 在 Studio 详细信息页面上的 Studio users (Studio 用户) 列表中，搜索要更新的用户或组。您可以按名称或身份类型进行搜索。
5. 选择要更新的用户或组，然后选择 Assign policy (分配策略) 以打开 Session policy (会话策略) 对话框。
6. 选择要应用于您在步骤 5 中选择的用户或组的策略，然后选择 Apply policy (应用策略)。Studio users (Studio 用户) 列表应在 Session policy (会话策略) 列中显示您更新的用户或组的策略名称。

使用 Amazon CLI更新用户或组的 EMR Studio 权限

为以下 `update-studio-session-mappings` 参数插入您自己的值。有关 `update-studio-session-mappings` 命令的更多信息，请参阅 Amazon CLI 命令参考。

```
aws emr update-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-name <name-of-user-or-group-to-update> \
 --session-policy-arn <new-session-policy-arn-to-apply> \
 --identity-type <USER-or-GROUP> \
```

使用 `get-studio-session-mapping` 命令验证新的会话策略分配。将 *<example-identity-name>* 替换为您更新的用户或组的 IAM Identity Center 名称。

```
aws emr get-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <user-or-group-name> \
```

从 Studio 中删除用户或组

IAM

要在使用 IAM 身份验证模式时从 EMR Studio 中删除用户或组，必须通过重新配置用户的 IAM 权限策略来撤销该用户对 Studio 的访问权限。

在以下示例策略中，假设您有附加标签键值对 `Team = Quality Assurance` 的 EMR Studio。根据该策略，用户可以访问标有 `Team` 键的 Studios，该键的值等于 `Data Analytics` 或者 `Quality Assurance`。要从标记为 `Team = Quality Assurance` 的 Studio 中删除用户，必须从标签值列表中删除 `Quality Assurance`。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
            "Condition": {
                "StringEquals": {
                    "emr:ResourceTag/Team": [
                        "Data Analytics",
                        "Quality Assurance"
                  ]
                }
            }
        }
    ]
}
```

IAM Identity Center

使用控制台从 EMR Studio 中删除用户或组

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 从左侧导航中选择 EMR Studio。

3. 从 Studios 列表选择您的 Studio 名称，或选择 Studio 后选择 View details (查看详细信息)，以打开 Studio 详细信息页面。
4. 在 Studio 详细信息页面上的 Studio users (Studio 用户) 列表中，查找要从 Studio 中删除的用户或组。您可以按名称或身份类型进行搜索。
5. 选择要删除的用户或组，选择 Delete (删除) 并确认。您删除的用户或组将从 Studio users (Studio 用户) 列表中消失。

使用Amazon CLI 从 EMR Studio 中删除用户或组

为以下 `delete-studio-session-mapping` 参数插入您自己的值。有关 `delete-studio-session-mapping` 命令的更多信息，请参阅 Amazon CLI 命令参考。

```
aws emr delete-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <name-of-user-or-group-to-delete> \
```

管理 Amazon EMR Studio

本部分描述了监控、更新或删除 EMR Studio 资源的说明。有关分配用户或更新用户权限的信息，请参阅 分配和管理 EMR Studio 用户。

查看 Studio 详细信息

New console

使用新控制台查看有关 EMR Studio 的详细信息

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/emr。
2. 在左侧导航栏的 EMR Studio 下，选择 Studios。
3. 从 Studios 列表中选择 Studio 以打开 Studio 详细信息页面。Studio 详细信息页面包括 Studio setting (Studio 设置) 信息，例如 Studio Description (描述)、VPC 和 Subnets (子网)。

Old console

使用旧控制台查看有关 EMR Studio 的详细信息

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/home。
2. 从左侧导航中选择 EMR Studio。
3. 从 Studios 列表中选择 Studio 以打开 Studio 详细信息页面。Studio 详细信息页面包括 Studio setting (Studio 设置) 信息，例如 Studio Description (描述)、VPC 和 Subnets (子网)。

CLI

使用 Amazon CLI 按 Studio ID 检索 EMR Studio 的详细信息

使用以下 `describe-studio` Amazon CLI 命令获取有关特定 EMR Studio 的详细信息。有关更多信息，请参阅 Amazon CLI 命令参考。

```
aws emr describe-studio \
 --studio-id <id-of-studio-to-describe> \
```

使用Amazon CLI检索 EMR Studio 列表

使用以下 `list-studios` Amazon CLI 命令。有关更多信息，请参阅 Amazon CLI 命令参考。

```
aws emr list-studios
```

以下是 `list-studios` 命令的示例返回值（JSON 格式）。

```
{
    "Studios": [
        {
            "AuthMode": "IAM",
            "VpcId": "vpc-b21XXXXX",
            "Name": "example-studio-name",
            "Url": "https://es-7HWP74SNGDXXXXXXXXXXXXXX.emrstudio-prod.us-
east-1.amazonaws.com",
            "CreationTime": 1605672582.781,
            "StudioId": "es-7HWP74SNGDXXXXXXXXXXXXXX",
            "Description": "example studio description"
        }
```

```
    ]
}
```

监控 Amazon EMR Studio 操作

查看 EMR Studio 和 API 活动

EMR Studio 与 Amazon CloudTrail 集成，该服务提供用户、IAM 角色或 EMR Studio 中其他Amazon 服务所采取的操作的记录。CloudTrail 将 EMR Studio 的 API 调用作为事件捕获。您可以使用 CloudTrail 控制台 (https://console.aws.amazon.com/cloudtrail/) 查看这些事件。

EMR Studio 事件提供诸如哪个 Studio 或 IAM 用户发出请求以及请求的类型等信息。

> **Note**
>
> 集群上的操作，例如：运行笔记本任务，不会发出 Amazon CloudTrail。

您还可以创建跟踪,以将 EMR Studio CloudTrail 事件持续传送到 Amazon S3 存储桶。有关更多信息，请参阅《Amazon CloudTrail 用户指南》。

CloudTrail 事件示例：用户调用 DescribeStudio API

以下是在用户 `admin` 调用 DescribeStudio API 时创建的示例 Amazon CloudTrail 事件。CloudTrail 将用户名记录为 `admin`。

> **Note**
>
> DescribeStudio 的 EMR Studio API 事件排除`responseElements`的值来保护 Studio 详细信息。

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDXXXXXXXXXXXXXXXXX",
      "arn":"arn:aws:iam::653XXXXXXXX:user/admin",
      "accountId":"653XXXXXXXX",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
```

```
      "userName":"admin"
   },
   "eventTime":"2021-01-07T19:13:58Z",
   "eventSource":"elasticmapreduce.amazonaws.com",
   "eventName":"DescribeStudio",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"72.XX.XXX.XX",
   "userAgent":"aws-cli/1.18.188 Python/3.8.5 Darwin/18.7.0 botocore/1.19.28",
   "requestParameters":{
      "studioId":"es-9O5XXXXXXXXXXXXXXXXXXXX"
   },
   "responseElements":null,
   "requestID":"0fxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
   "eventID":"b0xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
   "readOnly":true,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "eventCategory":"Management",
   "recipientAccountId":"653XXXXXXXXX"
}
```

查看 Spark 用户和任务活动

要查看 Amazon EMR Studio 用户的 Spark 任务活动，您可以在集群上配置用户模拟。借助用户模拟，从 Workspace 提交的每个 Spark 任务都与运行代码的 Studio 用户相关联。

启用用户模拟后，Amazon EMR 会在集群的主节点上为在 Workspace 中运行代码的每个用户创建一个 HDFS 用户目录。例如，如果用户 `studio-user-1@example.com` 运行代码，您可以连接主节点并可以看到 `hadoop fs -ls /user` 显示 `studio-user-1@example.com` 的目录。

要设置 Spark 用户模拟，请在以下配置分类中设置以下属性：

- `core-site`
- `livy-conf`

```
[
    {
        "Classification": "core-site",
        "Properties": {
          "hadoop.proxyuser.livy.groups": "*",
          "hadoop.proxyuser.livy.hosts": "*"
        }
```

```
    },
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.impersonation.enabled": "true"
        }
    }
]
```

要查看历史服务器页面，请参阅使用 EMR Studio 调试应用程序和任务。您还可以使用 SSH 连接到集群的主节点来查看应用程序的 Web 界面。有关更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

更新 Amazon EMR Studio

创建 EMR Studio 后，您可以使用 Amazon CLI 更新以下属性：

- 名称
- 描述
- 默认 S3 位置
- 子网

使用Amazon CLI更新 EMR Studio

使用 `update-studio` Amazon CLI 命令更新 EMR Studio。有关更多信息，请参阅 Amazon CLI 命令参考。

> **Note**
>
> 您可以将 Studio 与最多 5 个子网关联。这些子网必须与 Studio 属于同一 VPC。您提交给 `update-studio` 命令的子网 ID 列表可以包括新的子网 ID，但还必须包括您之前与 Studio 关联的所有子网 ID。您无法从 Studio 中删除子网。

```
aws emr update-studio \
 --studio-id <example-studio-id-to-update> \
 --name <example-new-studio-name> \
 --subnet-ids <old-subnet-id-1 old-subnet-id-2 old-subnet-id-3 new-subnet-id> \
```

要验证更改，请使用 `describe-studio` Amazon CLI 命令并指定您的 Studio ID。有关更多信息，请参阅 Amazon CLI 命令参考。

```
aws emr describe-studio \
 --studio-id <id-of-updated-studio> \
```

删除 Amazon EMR Studio 和 Workspaces

当您删除 Studio 时，EMR Studio 会删除与 Studio 关联的所有 IAM Identity Center 用户和组分配。

> **Note**
>
> 当您删除 Studio 时，Amazon EMR 不会删除与其关联的 Workspaces。您必须单独删除 Studio 中的 Workspaces。

删除 Workspaces

Console

由于每个 EMR Studio Workspace 都是 EMR Notebooks 实例，因此您可以使用 Amazon EMR 管理控制台删除 Workspaces。您可以在删除 Studio 前后使用 Amazon EMR 控制台删除 Workspaces

使用 Amazon EMR 控制台删除 Workspaces

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Notebooks (笔记本)。
3. 选择要删除的一个或多个 Workspace。
4. 选择 Delete (删除)，然后再一次选择 Delete (删除) 已确认删除。
5. 按照《Amazon Simple Storage Service 控制台用户指南》中的删除对象说明，从 Amazon S3 中删除与已删除 Workspace 关联的笔记本文件。

EMR Studio UI

From the Workspace UI

从 EMR Studio 中删除 Workspace 及其关联的备份文件

1. 使用您的 Studio 访问 URL 登录 EMR Studio，然后从左侧导航中选择 Workspaces。
2. 在列表中找到您的 Workspace，然后选中其名称旁边的复选框。您可以选择同时删除多个 Workspaces。
3. 选择 Workspaces 列表右上角的 Delete (删除) 并确认您要删除选定的 Workspaces。选择 Delete (删除) 以确认。
4. 按照《Amazon Simple Storage Service 控制台用户指南》中的删除对象说明，从 Amazon S3 中删除与已删除 Workspace 关联的笔记本文件。如果您没有创建 Studio，请咨询您的 Studio 管理员以确定已删除 Workspace 的 Amazon S3 备份位置。

From the Workspaces list

从 Workspace 列表中删除 Workspace 及其关联的备份文件

1. 在控制台中导航到 Workspace 列表。
2. 选择要从列表中删除的 Workspace，然后选择操作。
3. 选择 Delete (删除)。
4. 按照《Amazon Simple Storage Service 控制台用户指南》中的删除对象说明，从 Amazon S3 中删除与已删除 Workspace 关联的笔记本文件。如果您没有创建 Studio，请咨询您的 Studio 管理员以确定已删除 Workspace 的 Amazon S3 备份位置。

删除 EMR Studio

New console

使用新控制台删除 EMR Studio

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/emr。
2. 在左侧导航栏的 EMR Studio 下，选择 Studios。
3. 切换到 Studio 名称左侧，从 Studios 列表中选择 Studio。选择 Delete (删除)。

Old console

使用旧控制台删除 EMR Studio

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/home。
2. 从左侧导航中选择 EMR Studio。
3. 从 Studios 列表中选择 Studio，然后选择 Delete (删除)。

CLI

使用 Amazon CLI 删除 EMR Studio

使用 `delete-studio` Amazon CLI 命令删除 EMR Studio。有关更多信息，请参阅 Amazon CLI 命令参考。

```
aws emr delete-studio --studio-id <id-of-studio-to-delete>
```

定义安全组以控制 EMR Studio 网络流量

关于 EMR Studio 安全组

Amazon EMR Studio 使用两个安全组来控制 Studio 中的 Workspaces 与在 Amazon EC2 上运行的附加 Amazon EMR 集群之间的网络流量：

- 引擎安全组，它使用端口 18888 与在 Amazon EC2 上运行的附加 Amazon EMR 集群进行通信。
- 与 Studio 中的 Workspaces 关联的 Workspace 安全组。此安全组包含出站 HTTPS 规则，以允许 Workspace 将流量路由到互联网，并且必须允许端口 443 上到互联网的出站流量才能将 Git 存储库链接到 Workspace。

除了与附加到 Workspace 的 EMR 集群关联的任何安全组之外，EMR Studio 还使用这些安全组。

当您使用 Amazon CLI 创建 Studio 时，您必须创建这些安全组。

> **Note**
>
> 您可以使用为您的环境专门定制的规则为 EMR Studio 自定义安全组，但必须包含此页面上注明的规则。您的 Workspace 安全组不能允许任何入站流量，引擎安全组必须允许来自 Workspace 安全组的入站流量。

使用原定设置的 EMR Studio 安全组

您在使用 Amazon EMR 控制台时，可以选择以下默认安全组。默认安全组由 EMR Studio 代表您所创建，其中包括 EMR Studio Workspaces 所需的最低入站和出站规则。

- `DefaultEngineSecurityGroup`
- `DefaultWorkspaceSecurityGroupGit` 或 `DefaultWorkspaceSecurityGroupWithoutGit`

先决条件

要为 EMR Studio 创建安全组，您需要用于该 Studio 的 Amazon Virtual Private Cloud (VPC)。您可以在创建安全组时选择此 VPC。此 VPC 应与您在创建 Studio 时指定的 VPC 相同。如果计划通过 EMR Studio 使用 Amazon EMR on EKS，请为您的 Amazon EKS 集群 Worker 节点选择 VPC。

说明

按照适用于 Linux 实例的《Amazon EC2 用户指南》内创建安全组中的说明，在您的 VPC 中创建引擎安全组和 Workspace 安全组。安全组必须包括下表总结的规则。

当您为 EMR Studio 创建安全组时，请记下两者的 ID。创建 Studio 时，您可以按 ID 指定每个安全组。

引擎安全组

EMR Studio 使用端口 18888 与已附加的集群进行通信。

入站规则

类型	协议	端口	目标位置	描述
TCP	TCP	18888	您的 EMR Studio Workspace 安全组。	允许来自 EMR Studio 的 Workspace 安全组中的任何资源的流量。

WorkSpace 安全组

此安全组与 EMR Studio 中的 Workspaces 相关联。

出站规则

类型	协议	端口	目标位置	描述
TCP	TCP	18888	您的 EMR Studio 引擎安全组。	允许进入 EMR Studio 引擎安全组中的任何资源的流量。
HTTPS	TCP	443	0.0.0.0/0	允许进入互联网的流量以便将公共托管的 Git 存储库链接到 Workspaces。

为 Amazon EMR Studio 创建 Amazon CloudFormation 模板

关于 EMR Studio 集群模板

您可以创建 Amazon CloudFormation 模板以帮助 EMR Studio 用户在 Workspace 启动新的 Amazon EMR 集群。CloudFormation 模板是 JSON 或 YAML 格式的文本文件。在模板中，您描述了 Amazon 资源堆栈并告知 CloudFormation 如何为您预置这些资源。针对 EMR Studio，您可以创建一个或多个描述 Amazon EMR 集群的模板。

您在 Amazon Service Catalog 中组织模板，Amazon Service Catalog以便您可以创建并管理常用部署的 IT 服务，该服务名为 Amazon 的产品。您将模板作为产品收集到与 EMR Studio 用户共享的文件夹。创建集群模板后，Studio 用户可以使用其中一个模板为 Workspace 启动新集群。用户必须具有从模板创建新集群的权限。您可以在 EMR Studio 权限策略 设置用户权限。

要了解有关 CloudFormation 模板的更多信息，请参阅《Amazon CloudFormation 用户指南》中的模板。有关 Amazon Service Catalog 的更多信息，请参阅什么是 Amazon Service Catalog。

以下视频演示了如何为 EMR Studio 设置 Amazon Service Catalog 的集群模板。如果您想了解详情，还可参阅博文 Build a self-service environment for each line of business using Amazon EMR and Service Catalog（使用 Amazon EMR 和 Service Catalog 为每条业务线构建自助服务环境）。

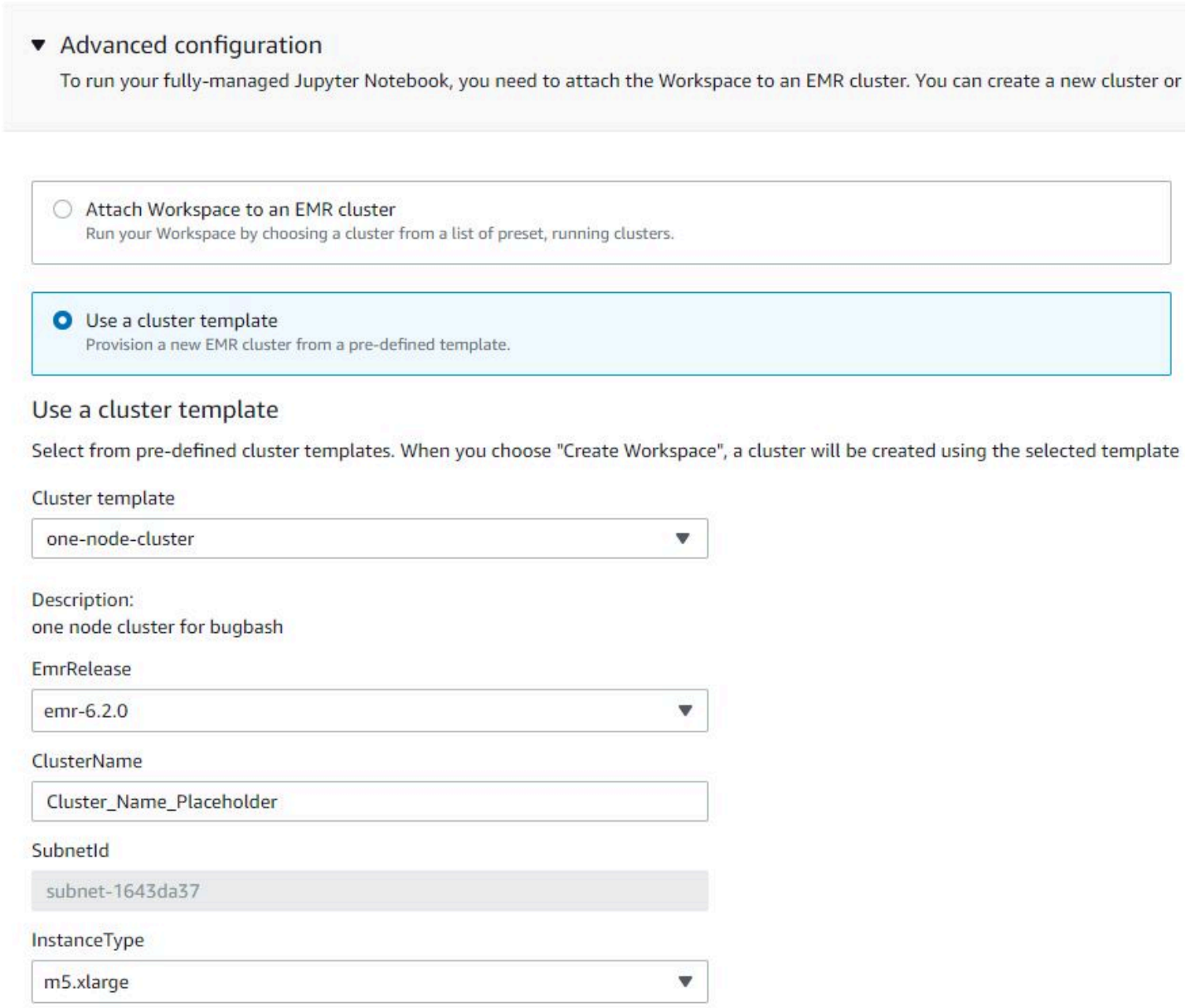
可选模板参数

您可以在模板的`Parameters`部分添加额外选项。参数允许 Studio 用户为集群输入或选择自定义值。例如，您可以添加允许用户选择特定 Amazon EMR 版本的参数。有关更多信息，请参阅《Amazon CloudFormation 用户指南》中的参数。

以下示例 `Parameters` 部分定义了其他输入参数，例如 `ClusterName`、`EmrRelease` 版本和 `ClusterInstanceType`。

```
Parameters:
  ClusterName:
    Type: "String"
    Default: "Cluster_Name_Placeholder"
  EmrRelease:
    Type: "String"
    Default: "emr-6.2.0"
    AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
    Default: "m5.xlarge"
    AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"
```

添加参数时，Studio 用户在选择集群模板后会看到其他表单选项。下图为 EmrRelease 版本、ClusterName 和 InstanceType 提供了以下的其他表单选项。



先决条件

在创建集群模板之前，请确保您具有访问 Service Catalog 管理员控制台视图的 IAM 权限。您还需要 IAM 权限才能执行 Service Catalog 管理任务。有关更多信息，请参阅向 Service Catalog 管理员授予权限。

说明

使用 Service Catalog 创建 EMR 集群模板

1. 创建一个或多个 CloudFormation 模板。您可以随意选择存储模板的位置。由于模板是格式化的文本文件，因此您可以将它们上传到 Amazon S3 或保存在本地文件系统。要了解有关 CloudFormation 模板的更多信息，请参阅《Amazon CloudFormation 用户指南》中的模板。

   使用以下规则来命名您的模板，或根据模式 `[a-zA-Z0-9][a-zA-Z0-9._-]*` 检查您的名称。

   - 模板名称必须以字母或数字开头。
   - 模板名称只能包含字母、数字、句点 (.)、下划线 (_) 和连字符 (-)。

   每个您创建的集群模板都必须包括以下选项：

   输入参数

   - ClusterName – 集群的名称，用于帮助用户在预置集群后识别集群。

   输出

   - `ClusterId` – 新预置的 EMR 集群的 ID。

   以下是 YAML 格式的示例 Amazon CloudFormation 模板，适用于有两个节点的集群。示例模板包括所需的模板选项，并定义了 `EmrRelease` 和 `ClusterInstanceType` 的其他输入参数。

```
awsTemplateFormatVersion: 2010-09-09

Parameters:
  ClusterName:
    Type: "String"
    Default: "Example_Two_Node_Cluster"
  EmrRelease:
    Type: "String"
    Default: "emr-6.2.0"
    AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
```

```
    Default: "m5.xlarge"
    AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"

Resources:
  EmrCluster:
    Type: AWS::EMR::Cluster
    Properties:
      Applications:
      - Name: Spark
      - Name: Livy
      - Name: JupyterEnterpriseGateway
      - Name: Hive
      EbsRootVolumeSize: '10'
      Name: !Ref ClusterName
      JobFlowRole: EMR_EC2_DefaultRole
      ServiceRole: EMR_DefaultRole_V2
      ReleaseLabel: !Ref EmrRelease
      VisibleToAllUsers: true
      LogUri:
        Fn::Sub: 's3://aws-logs-${AWS::AccountId}-${AWS::Region}/elasticmapreduce/'
      Instances:
        TerminationProtected: false
        Ec2SubnetId: 'subnet-ab12345c'
        MasterInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
        CoreInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
          Market: ON_DEMAND
          Name: Core

Outputs:
  ClusterId:
    Value:
      Ref: EmrCluster
    Description: The ID of the  EMR cluster
```

2. 在与您的 Studio 相同的 Amazon 账户中为您的集群模板创建一个产品组合。

   a. 打开 Amazon Service Catalog 控制台：https://console.aws.amazon.com/servicequotas/。

b. 从左侧导航菜单中，选择 Portfolios (产品组合)。

c. 在 Create portfolio (创建产品组合) 页面输入所需信息。

d. 选择 Create (创建)。Amazon Service Catalog 创建产品组合并显示产品组合详细信息。

3. 使用以下步骤将集群模板添加为 Amazon Service Catalog 产品。

a. 在 Amazon Service Catalog 管理控制台，导航到 Administration (管理) 下面的 Products (产品) 页面。

b. 选择 Upload new product (上传新产品)。

c. 输入 Product name (产品名称) 和 Owner (拥有者)。

d. 在 Version details (详细版本) 指定模板文件。

e. 选择 Review (审核)，查看您的产品设置，然后选择 Create product (创建产品)。

4. 完成以下步骤，将产品添加到产品组合。

a. 导航到 Amazon Service Catalog 管理控制台的 Products (产品) 页面。

b. 选择您的产品，选择 Actions (操作)，然后选择 Add product to portfolio (添加产品至产品组合)。

c. 选择您的产品组合，然后选择 Add product to portfolio (添加产品至产品组合)。

5. 为您的产品创建启动约束。启动约束是指定用户启动产品权限的 IAM 角色。您可以定制启动约束，但必须通过 CloudFormation、Amazon EMR 和Amazon Service Catalog授予权限。有关更多信息和说明，请参阅 Service Catalog 启动约束。

6. 将启动约束应用于产品组合中的每个产品。您必须将启动约束分别应用于每个产品。

a. 从 Amazon Service Catalog 管理控制台的 Portfolios (产品组合) 页面选择产品组合。

b. 选择 Constraints (约束) 选项卡并选择 Create constraint (创建约束)。

c. 在 Constraint type (约束类型) 下，选择产品，然后选择 Launch (启动)。选择 Continue (继续) 。

d. 选择 Launch constraint (启动约束) 部分中的启动约束角色，然后选择Create (创建)。

7. 授予对产品组合的访问权限。

a. 从 Amazon Service Catalog 管理控制台的 Portfolios (产品组合) 页面选择产品组合。

b. 展开 Groups, roles, and users (组、角色和用户) 选项卡然后选择 Add groups, roles, users (添加组、角色和用户)。

c. 在 Roles (角色)选 项卡搜索 EMR Studio IAM 角色，选择角色，然后选择 Add access (添加访问)。

如果您使用…	授予对…的访问权限
IAM 身份验证	您的本地用户
IAM 联合身份验证	联合身份验证的 IAM 角色
IAM Identity Center 联合身份验证	您的 EMR Studio 用户角色

为基于 Git 的存储库建立访问和权限

EMR Studio 支持以下基于 Git 的服务：

- Amazon CodeCommit
- GitHub
- Bitbucket
- GitLab

要让 EMR Studio 用户将 Git 存储库与 Workspace 相关联，请设置以下访问和权限要求。您还可以按照为 EMR Studio 配置私有托管的 Git 存储库中的说明配置在专用网络中托管的基于 Git 的存储库。

集群互联网访问

在 Amazon EC2 上运行的 Amazon EMR 集群和附加到 Studio Workspaces 的 Amazon EMR on EKS 集群都必须位于使用网络地址转换（NAT）网关的私有子网中，或者它们必须能够通过虚拟私有网关访问互联网。有关更多信息，请参阅 Amazon VPC 选项。

您与 EMR Studio 一起使用的安全组还必须包含一个出站规则，该规则允许 Workspaces 将流量从附加的 EMR 集群路由到互联网。有关更多信息，请参阅 定义安全组以控制 EMR Studio 网络流量。

> **Important**
>
> 如果网络接口在公有子网，它将无法通过互联网网关 (IGW) 与互联网通信。

Amazon Secrets Manager 权限

要让 EMR Studio 用户使用存储在 Amazon Secrets Manager 中的密钥访问 Git 存储库，请向 EMR Studio 的服务角色添加允许 `secretsmanager:GetSecretValue` 运营的权限策略。

有关如何将基于 Git 的存储库链接到 Workspaces 的信息，请参阅将基于 Git 的存储库链接到 EMR Studio Workspace。

为 EMR Studio 配置私有托管的 Git 存储库

按照以下说明为 Amazon EMR Studio 配置私有托管的存储库。请提供一个配置文件，其中包含有关您的 DNS 和 Git 服务器的信息。EMR Studio 使用此信息配置可将流量路由到您自行管理的存储库的 Workspaces。

> **Note**
>
> 如果您配置 `DnsServerIpV4`，EMR Studio 将使用您的 DNS 服务器来解析您的 `GitServerDnsName` 和您的 Amazon EMR 端点，例如 `elasticmapreduce.us-east-1.amazonaws.com`。要为 Amazon EMR 设置端点，请通过您在 Studio 中使用的 VPC 连接到您的端点。这可确保 Amazon EMR 端点会解析为私有 IP。有关更多信息，请参阅使用接口 VPC 终端节点连接到 Amazon EMR。

先决条件

在为 EMR Studio 配置私有托管的 Git 存储库之前，您需要一个 Amazon S3 存储位置，EMR Studio 可以在其中备份 Studio 中的工作区和笔记本文件。使用您在创建 Studio 时指定的相同 S3 存储桶。

为 EMR Studio 配置一个或多个私有托管的 Git 存储库

1. 使用以下模板创建配置文件。为要在配置中指定的每个 Git 服务器包括以下值：

   - **`DnsServerIpV4`** – 您的 DNS 服务器的 IPv4 地址。如果您同时提供 `DnsServerIpV4` 和 `GitServerIpV4List` 的值，则 `DnsServerIpV4` 的值优先，而且 EMR Studio 使用 `DnsServerIpV4` 解析您的 `GitServerDnsName`。

> **Note**
>
> 要使用私有托管的 Git 存储库，您的 DNS 服务器必须允许来自 EMR Studio 的入站访问。我们敦促您保护您的 DNS 服务器，防止其他未经授权的访问。

- **`GitServerDnsName`** – 您的 Git 服务器的 DNS 名称。例如 `"git.example.com"`。
- **`GitServerIpV4List`** – 属于您的 Git 服务器的 IPv4 地址列表。

```
[
    {
        "Type": "PrivatelyHostedGitConfig",
        "Value": [
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<enterprise.git.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            },
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<git.example.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            }
        ]
    }
]
```

2. 将配置文件另存为 `configuration.json`。
3. 将配置文件上载到名为 `life-cycle-configuration` 的文件夹中的 Amazon S3 存储位置。例如，如果您的默认 S3 位置是 `s3://DOC-EXAMPLE-BUCKET/studios`，则您的配置文件将位于 `s3://DOC-EXAMPLE-BUCKET/studios/`**`life-cycle-configuration/configuration.json`**。

> **Important**
>
> 我们敦促您将 `life-cycle-configuration` 文件夹的访问权限限制为仅限 Studio 管理员和您的 EMR Studio 服务角色，并保护 `configuration.json` 免受未经授权的访问。有关说明，请参阅使用用户策略控制对存储桶的访问或 Amazon S3 的安全最佳实践。

有关上载说明，请参阅《Amazon Simple Storage Service 用户指南》中的创建文件夹和上载对象。要将您的配置应用到现有 Workspace，您应该在将配置文件上传到 Amazon S3 后关闭并重新启动 Workspace。

在 EMR Studio 优化 Spark 任务

使用 EMR Studio 运行 Spark 任务时，您可以采取几个步骤以确保优化 Amazon EMR 集群资源。

延长您的 Livy 会话

如果您在 Amazon EMR 集群将 Apache Livy 和 Spark 一起使用，我们建议您选择以下一种操作来延长 Livy 会话时间：

- 当您创建 Amazon EMR 集群时，请在 Enter Configuration (输入配置) 字段设置配置类型。

```
[
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.server.session.timeout": "8h"
        }
    }
]
```

- 针对已经运行的 EMR 集群，请使用 `ssh` 连接集群，然后在 `/etc/livy/conf/livy.conf` 设置 `livy-conf` 配置分类。

```
[
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.server.session.timeout": "8h"
```

```
        }
    }
]
```

更改配置后，您可能需要重新启动 Livy。

- 如果您不想 Livy 会话超时，请在 `/etc/livy/conf/livy.conf` 将属性 `livy.server.session.timeout-check` 设置为 `false`。

在集群模式下运行 Spark

在集群模式下，Spark 驱动程序在核心节点上运行，而不是在主节点上运行，从而提高了主节点上的资源利用率。

要以集群模式而不是默认客户端模式运行 Spark 应用程序，在设置部署模式时请选择 Cluster (集群) 模式，同时在新的 Amazon EMR 集群中配置 Spark 步骤。有关更多信息，请参阅 Apache Spark 文档中的集群模式概览。

增加 Spark 驱动程序内存

要增加 Spark 驱动程序内存，请使用 EMR Notebooks 的 `%%configure` 魔术命令配置 Spark 会话，如以下示例所示。

```
%%configure -f
{"driverMemory": "6000M"}
```

使用 Amazon EMR Studio

本节中的主题有助于您配置 Amazon EMR Studio 并与之交互。

以下视频介绍了实用信息，例如如何创建新的 Workspace 以及如何使用集群模板启动新的 Amazon EMR 集群。该视频还演示了如何运行示例笔记本电脑实例。

本节包含以下主题，可帮助您使用 EMR Studio：

- 学习 WorkSpace 基础知识
- 配置 Workspace 协作
- 使用运行时系统角色运行 EMR Studio Workspace
- 以编程方式运行 Workspace 笔记本

- 用 SQL Explorer 浏览数据
- 将计算资源附加到 EMR Studio Workspace
- 将基于 Git 的存储库链接到 EMR Studio Workspace
- 使用 EMR Studio 调试应用程序和任务
- 在 EMR Studio Workspace 中安装内核和库
- 使用 magic 命令增强内核
- 将多语言笔记本与 Spark 内核配合使用

学习 WorkSpace 基础知识

使用 EMR Studio 后，您可以创建和配置不同的 Workspaces 来组织和运行笔记本。本部分介绍了创建与使用 Workspaces。有关概念性概述，请参阅第 Amazon EMR Studio 工作原理 页上的 Workspaces。

本节涵盖以下主题，可帮助您使用 EMR Studio Workspaces：

- 创建 EMR Studio Workspace
- 启动 WorkSpace
- 了解 Workspace 用户界面
- 探索笔记本示例
- 保存 Workspace 内容
- 删除 Workspace 和笔记本文件
- 了解 Workspace 状态
- 解决 Workspace 连接问题

创建 EMR Studio Workspace

您可以创建 EMR Studio Workspaces 以使用 EMR Studio 界面运行笔记本代码。

在 EMR Studio 中创建 Workspace

1. 登录 EMR Studio。
2. 选择创建 Workspace。
3. 输入 Workspace name (Workspace 名称) 和 Description (描述)。命名 Workspace 有助于您在 Workspaces 页面上识别它。

4. 要与此 Workspace 中的其他 Studio 用户实时协作，可启用 Workspace 协作功能。启动 Workspace 后，您可以配置协作者。
5. 如果要将集群连接到 Workspace，请展开高级配置部分。如果愿意，可在稍后连接集群。有关更多信息，请参阅将计算资源附加到 EMR Studio Workspace。

   > **Note**
   >
   > 要预置新集群，您需要从管理员处获取访问权限。

   为 Workspace 选择其中一个集群选项并附加集群。有关在创建 Workspace 时预置集群的更多信息，请参阅 创建新的 EMR 集群并将其附加到 EMR Studio Workspace。
6. 选择页面右下角的创建 Workspace。

创建 Workspace 后，EMR Studio 将打开 Workspaces 页面。您将在页面顶部看到一个绿色的成功横幅，并且可以在列表中找到新创建的 Workspace。

预设情况下，Workspace 是共享的，所有 Studio 用户都可以看到。但一次只能有一个用户打开一个 Workspace 并在其中工作。要与其他用户同时工作，您可以配置 Workspace 协作

启动 WorkSpace

要开始使用笔记本文件，请启动 Workspace 以访问笔记本编辑器。Studio 中的 Workspaces 页面列出了您有权访问的所有 Workspace，以及 Name（名称）、Status（状态）、Creation time（创建时间）和 Last modified（上次修改时间）等详细信息。

> **Note**
>
> 如果在 Amazon EMR 旧控制台上有 EMR Notebooks，您可以在新控制台中以 EMR Studio Workspaces 的形式找到它们。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。如果最近在旧控制台中创建了笔记本，则可能需要刷新 Workspace 列表才能在新控制台中看到它们。有关过渡的更多信息，请参阅 Amazon EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供 和 控制台中的新增功能。

启动 Workspace 以编辑和运行笔记本

1. 在 Studio 的 Workspaces 页面上，查找 Workspace。您可以按关键字或列值来筛选列表。

2. 选择 Workspace 名称，以在新的浏览器选项卡中启动 Workspace。如果 Workspace 处于 Idle (空闲) 状态，系统可能需要几分钟才能打开它。另外，可以选择 Workspace 的行，然后选择启动 Workspace。可从以下选项中进行选择：

   - 快速启动 – 使用默认选项快速启动 Workspace。如果要在 JupyterLab 中将集群连接到 Workspace，请选择快速启动。
   - 使用选项启动 – 使用自定义选项启动 Workspace。您可以选择在 Jupyter 或 JupyterLab 中启动，将 Workspace 连接到 EMR 集群，然后选择安全组。

> **Note**
>
> 一次只能有一个用户打开 Workspace 并在其中工作。如果您选择一个已在使用的 Workspace，当您尝试打开它时，EMR Studio 会显示一条通知。Workspaces 页面上的 User (用户) 列显示当前正在使用 Workspace 的用户。

了解 Workspace 用户界面

EMR Studio Workspace 用户界面基于 JupyterLab 界面，左侧边栏上有以图标表示的选项卡。当您在图标上暂停时，您会看到一个工具提示，其中显示了选项卡的名称。从左侧边栏中选择选项卡以访问以下面板。

- File Browser (文件浏览器) – 显示 Workspace 中的文件和目录，以及链接的 Git 存储库的文件和目录。
- Running Kernels and Terminals (正在运行的内核和终端) – 列出在 Workspace 中运行的所有内核和终端。有关更多信息，请参阅 JupyterLab 官方文档中的管理内核和终端。
- Git – 提供图形用户界面，用于在附加到 Workspace 的 Git 存储库中执行命令。该面板是一个名为 jupyterlab-git 的 JupyterLab 扩展。有关更多信息，请参阅 jupyterlab-git。
- EMR 集群 – 允许将集群附加到 Workspace 或从 Workspace 分离集群，以便运行笔记本代码。EMR 集群配置面板还提供高级配置选项，以帮助您创建新集群并将其附加到 Workspace。有关更多信息，请参阅创建新的 EMR 集群并将其附加到 EMR Studio Workspace。
- Amazon EMR Git 存储库 – 帮助您将 Workspace 与最多三个 Git 存储库链接起来。有关详细信息和说明，请参阅将基于 Git 的存储库链接到 EMR Studio Workspace。
- Notebook Examples (笔记本示例) – 提供可以保存到 Workspace 的笔记本示例列表。您还可以通过在 Workspace 的 Launcher (启动器) 页面上选择 Notebook Examples (笔记本示例) 来访问示例。

- Commands (命令) – 提供一种键盘驱动的方式来搜索和运行 JupyterLab 命令。有关更多信息，请参阅 JupyterLab 文档中的 Command palette (命令调色板) 页面。
- Notebook Tools (笔记本工具) – 允许您选择和设置单元格滑动类型和元数据等选项。Notebook Tools (笔记本工具) 选项在您打开笔记本文件后显示在左侧边栏中。
- Open Tabs（打开的选项卡）– 列出主工作区中打开的文档和活动，以便您可以跳转到打开的选项卡。有关更多信息，请参阅 JupyterLab 文档中的选项卡和单个文档模式页面。
- Collaboration（协作）– 允许您启用或禁用 Workspace 协作以及管理协作者。要查看 Collaboration（协作）面板，您必须具有必要的权限。有关更多信息，请参阅设置 Workspace 协作的所有权。

探索笔记本示例

每个 EMR Studio Workspace 都包含一组笔记本示例，可用于探索 EMR Studio 功能。要编辑或运行笔记本示例，您可以将其保存到 Workspace。

将笔记本示例保存到 Workspace

1. 从左侧边栏中，选择 Notebook Examples (笔记本示例) 选项卡以打开 Notebook Examples (笔记本示例) 面板。您还可以通过在 Workspace 的 Launcher (启动器) 页面上选择 Notebook Examples (笔记本示例) 来访问示例。
2. 选择一个笔记本示例以在主工作区中进行预览。示例是只读的。
3. 要将笔记本示例保存到 Workspace，请选择 Save to Workspace (保存到 Workspace)。EMR Studio 将示例保存在您的主目录中。将笔记本示例保存到 Workspace 后，您可以重命名、编辑和运行它。

有关笔记本示例的更多信息，请参阅 EMR Studio 笔记本示例 GitHub 存储库。

保存 Workspace 内容

当您在 Workspace 的笔记本编辑器中工作时，EMR Studio 会为您将笔记本单元格和输出的内容保存在与 Studio 关联的 Amazon S3 位置。此备份过程可在会话之间保留工作。

您还可以通过在打开的笔记本选项卡中按 CTRL+S 或使用 File (文件) 下的其中一个保存选项保存笔记本。

备份 Workspace 中笔记本文件的另一种方法是将 Workspace 与基于 Git 的存储库相关联，并将更改与远程存储库同步。这样做还可以让您与使用不同 Workspace 或 Studio 的团队成员保存和共享笔记本。有关说明，请参阅 将基于 Git 的存储库链接到 EMR Studio Workspace。

删除 Workspace 和笔记本文件

当您从 EMR Studio Workspace 中删除笔记本文件时，您会从 File browser (文件浏览器) 中删除该文件，并且 EMR Studio 会删除其在 Amazon S3 中的备份副本。当您从 Workspace 删除文件时，无需采取任何进一步措施来避免存储费用。

删除整个 Workspace 时，其笔记本文件和文件夹将保留在 Amazon S3 存储位置中。文件会继续产生存储费用。为避免产生存储费用，请从 Amazon S3 中删除与已删除 Workspace 关联的所有备份文件和文件夹。

从 EMR Studio Workspace 中删除笔记本文件

1. 从 Workspace 的左侧边栏中选择 File browser (文件浏览器) 面板。
2. 选择要删除的文件或文件夹。右键单击选定内容，然后选择 Delete (删除)。文件将从列表中消失。EMR Studio 会为您从 Amazon S3 中删除文件或文件夹。

From the Workspace UI

从 EMR Studio 中删除 Workspace 及其关联的备份文件

1. 使用您的 Studio 访问 URL 登录 EMR Studio，然后从左侧导航中选择 Workspaces。
2. 在列表中找到您的 Workspace，然后选中其名称旁边的复选框。您可以选择同时删除多个 Workspaces。
3. 选择 Workspaces 列表右上角的 Delete (删除) 并确认您要删除选定的 Workspaces。选择 Delete (删除) 以确认。
4. 按照《Amazon Simple Storage Service 控制台用户指南》中的删除对象说明，从 Amazon S3 中删除与已删除 Workspace 关联的笔记本文件。如果您没有创建 Studio，请咨询您的 Studio 管理员以确定已删除 Workspace 的 Amazon S3 备份位置。

From the Workspaces list

从 Workspace 列表中删除 Workspace 及其关联的备份文件

1. 在控制台中导航到 Workspace 列表。

2. 选择要从列表中删除的 Workspace，然后选择操作。
3. 选择 Delete (删除)。
4. 按照《Amazon Simple Storage Service 控制台用户指南》中的删除对象说明，从 Amazon S3 中删除与已删除 Workspace 关联的笔记本文件。如果您没有创建 Studio，请咨询您的 Studio 管理员以确定已删除 Workspace 的 Amazon S3 备份位置。

了解 Workspace 状态

创建 EMR Studio Workspace 后，它在 Workspaces 列表中显示为一行，其中包含其名称、状态、创建时间和上次修改时间戳。下表描述了 Workspace 的状态。

状态	描述
Starting (正在启动)	Workspace 正在准备中，但尚未准备好使用。当 Workspace 状态为“Starting (正在启动)”时，您无法打开它。
Ready (就绪)	您可以打开 Workspace 以使用笔记本编辑器，但必须先将 Workspace 附加到 EMR 集群，然后才能运行笔记本代码。
Attaching (正在附加)	正将 Workspace 附加到集群。
Attached (已附加)	Workspace 已附加到 EMR 集群，可供您编写和运行笔记本代码。如果 Workspace 的状态不是 Attached (已附加)，则必须先将其附加到集群，然后才能运行笔记本代码。
Idle (空闲)	Workspace 已停止。要重新激活空闲 Workspace，请从 Workspaces 列表中选择它。当您选择 Workspace 时，状态从 Idle (空闲) 变为 Starting (正在启动) 再变为 Ready (就绪)。
Stopping (正在停止)	Workspace 正在关闭，并将设置为空闲。停止 Workspace 时，会终止所有相应的笔记本内核。EMR Studio 会停止长时间处于非活动状态的笔记本。

状态	描述
Deleting (正在删除)	删除 Workspace 时，EMR Studio 会将其标记为删除并启动删除过程。删除过程完成后，Workspace 将从列表中消失。删除 Workspace 时，其笔记本文件将保留在 Amazon S3 存储位置中。

解决 Workspace 连接问题

要解决 Workspace 连接问题，您可以停止并重启 Workspace。当您重启 Workspace 时，EMR Studio 会在与您的 Studio 关联的其他可用区或其他子网中启动该 Workspace。

停止并重启 EMR Studio Workspace

1. 在浏览器中关闭该 Workspace。
2. 在控制台中导航到 Workspace 列表。
3. 从该列表中选择您的 Workspace，然后选择 Actions（操作）。
4. 选择 Stop（停止），然后等待 Workspace 状态从 Stopping（正在停止）变为 Idle（空闲）。
5. 再次选择 Actions（操作），然后选择 Start（启动）以重启该 Workspace。
6. 等待 Workspace 状态从 Starting（正在启动）变为 Ready（就绪），然后选择 Workspace 名称，以便在新的浏览器选项卡中重新打开它。

配置 Workspace 协作

借助 Workspace 协作功能，您可以与团队的其他成员同时编写和运行笔记本代码。当您在同一笔记本文件中工作时，您会看到协作者做出的更改。您可以在创建 Workspace 时启用协作，也可以在现有 Workspace 中打开和关闭协作。

> **Note**
>
> EMR Studio Workspace 协作不支持与 EMR Serverless 交互式应用程序 结合使用。

先决条件

您需要首先完成以下任务，然后才能为 Workspace 配置协作：

- 确保您的 EMR Studio 管理员已经为您提供了必要的权限。例如，以下示例语句允许用户为任何具有标签键 `creatorUserId` 并且标签值与该用户的 ID（由策略变量 `aws:userId` 指示）一致的 WorkSpace 配置协作。

```
{
    "Sid": "UserRolePermissionsForCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
        "elasticmapreduce:DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userid}"
        }
    }
}
```

- 确保与 EMR Studio 关联的服务角色具有启用和配置 Workspace 协作所需的权限，如下面的示例语句所示。

```
{
    "Sid": "AllowWorkspaceCollaboration",
    "Effect": "Allow",
    "Action": [
        "iam:GetUser",
        "iam:GetRole",
        "iam:ListUsers",
        "iam:ListRoles",
        "sso:GetManagedApplicationInstance",
        "sso-directory:SearchUsers"
    ],
    "Resource": "*"
}
```

有关更多信息，请参阅创建 EMR Studio 服务角色。

启用 Workspace 协作并添加协作者

1. 在您的 Workspace 中，选择 Launcher（启动程序）屏幕或左侧面板底部的 Collaboration（协作）图标。

   > **Note**
   >
   > 除非您的 Studio 管理员已经向您授予了配置 Workspace 协作的权限，否则您不会看到 Collaboration（协作）面板。有关更多信息，请参阅设置 Workspace 协作的所有权。

2. 确保 Allow Workspace collaboration（允许 Workspace 协作）切换按钮位于打开位置。启用协作后，只有您和您添加的协作者才会在 Studio Workspaces 页面上的列表中看到该 Workspace。

3. 输入一个 Collaborator name（协作者姓名）。您的 Workspace 最多可有五个协作者，包括您自己。协作者可以是有权访问 EMR Studio 的任何用户。如果您没有输入协作者，则该工作区是一个只有您可以访问的私有工作区。

   下表根据拥有者的身份类型指定了要输入的适用的协作者值。

   > **Note**
   >
   > 拥有者只能邀请具有相同身份类型的协作者。例如，用户只能添加其他用户，IAM Identity Center 用户只能添加其他 IAM Identity Center 用户。

身份验证模式	要为 Collaborator name（协作者姓名）输入的值
IAM 身份验证	用户名。这是用户在登录到Amazon Web Services Management Console后会看到的名称。
IAM 联合身份验证	IAM 角色的名称以及一个可选的会话名称。 要添加将代入同一 IAM 角色的所有联合身份用户，请指定用于联合身份的 IAM 角色的名称。

身份验证模式	要为 Collaborator name（协作者姓名）输入的值
	要将单个用户添加为协作者，请指定一个角色和一个会话名称。例如，`MyRoleName:MySessionName` 。
SSO	IAM Identity Center 用户名，例如 `user@example.com`.

4. 选择 Add（添加）。协作者现在可以在他们的 EMR Studio Workspaces 页面看到该 Workspace，并且可以启动该 Workspace 以实时与您协作使用。

> **Note**
>
> 如果您禁用了 Workspace 协作，Workspace 将返回其共享状态，所有 Studio 用户都可以看到。在共享状态下，一次只有一个 Studio 用户能够打开该 Workspace 并在其中工作。

使用运行时系统角色运行 EMR Studio Workspace

> **Note**
>
> 此页描述的运行时系统角色功能仅适用于在 Amazon EC2 上运行的 Amazon EMR，并非指 EMR Serverless 交互式应用程序中的运行时系统角色功能。要详细了解如何在 EMR Serverless 中使用运行时系统角色，请参阅《Amazon EMR Serverless 用户指南》中的 Job runtime roles。

运行时角色是一个 Amazon Identity and Access Management（IAM）角色，您可以在向 Amazon EMR 集群提交作业或查询时指定该角色。您提交给 EMR 集群的任务或查询使用运行时系统角色访问 Amazon 资源，例如 Amazon S3 中的对象。

在将 EMR Studio Workspace 附加到使用 Amazon EMR 6.11 或更高版本的 EMR 集群时，您可以为提交的任务或查询选择运行时系统角色，供其在访问 Amazon 资源时使用。但是，如果 EMR 集群不支持运行时系统角色，则 EMR 集群在访问 Amazon 资源时不会代入该角色。

在将运行时系统角色用于 Amazon EMR Studio Workspace 之前，管理员必须配置用户权限，以便 Studio 用户对运行时系统角色调用 `elasticmapreduce:GetClusterSessionCredentials` API。然后，启动一个具有运行时系统角色的新集群，您可以将其与 Amazon EMR Studio Workspace 配合使用。

本页内容

- 为运行时系统角色配置用户权限
- 使用运行时系统角色启动新集群
- 在 Workspace 中使用具有运行时系统角色的 EMR 集群
- 注意事项

为运行时系统角色配置用户权限

配置用户权限，以便 Studio 用户可以针对自己要使用的运行时系统角色调用 `elasticmapreduce:GetClusterSessionCredentials` API。您还必须先配置 the section called “Studio 用户权限（EC2、EKS）”，用户才能开始使用 Studio。

> **Warning**
>
> 要授予此权限，请在向调用者授予调用 `GetClusterSessionCredentials` API 的权限时，根据 `elasticmapreduce:ExecutionRoleArn` 上下文键创建条件。以下示例演示了如何执行此操作。

```
{
      "Sid": "AllowSpecificExecRoleArn",
      "Effect": "Allow",
      "Action": [
          "elasticmapreduce:GetClusterSessionCredentials"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": {
              "elasticmapreduce:ExecutionRoleArn": [
                  "arn:aws:iam::111122223333:role/test-emr-demo1",
                  "arn:aws:iam::111122223333:role/test-emr-demo2"
              ]
          }
```

```
      }
  }
```

以下示例演示如何允许 IAM 主体使用名为 test-emr-demo3 的 IAM 角色作为运行时系统角色。此外，策略持有人只能使用集群 ID j-123456789 访问 Amazon EMR 集群。

```
{
    "Sid":"AllowSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:GetClusterSessionCredentials"
    ],
    "Resource": [
          "arn:aws:elasticmapreduce:<region>:111122223333:cluster/j-123456789"
     ],
    "Condition":{
        "StringEquals":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::111122223333:role/test-emr-demo3"
            ]
        }
    }
}
```

以下示例允许 IAM 主体使用名称以字符串 test-emr-demo4 开头的任何 IAM 角色作为运行时系统角色。此外，策略持有人只能使用键值对 tagKey: tagValue 访问 Amazon EMR 集群。

```
{
    "Sid":"AllowSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:GetClusterSessionCredentials"
    ],
    "Resource": "*",
    "Condition":{
        "StringEquals":{
             "elasticmapreduce:ResourceTag/tagKey": "tagValue"
        },
        "StringLike":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::111122223333:role/test-emr-demo4*"
            ]
```

```
        }
    }
}
```

使用运行时系统角色启动新集群

现在有了所需权限，启动一个具有运行时系统角色的新集群，您可以将其与 Amazon EMR Studio Workspace 配合使用。

如果已经启动具有运行时系统角色的新集群，则可跳至 the section called “将集群与 Workspace 配合使用” 小节。

1. 首先，完成 Amazon EMR 步骤的运行时角色 小节中的先决条件。
2. 然后，启动具有以下设置的集群，以便在 Amazon EMR Studio Workspaces 中使用运行时系统角色。有关如何启动集群的说明，请参阅 为集群指定安全配置。

   - 对于发行版标签，选择 emr-6.11.0 或更高版本。
   - 选择 Spark、Livy 和 Jupyter Enterprise Gateway 作为集群应用程序。
   - 使用在上一步中创建的安全配置。
   - 您可以选择为 EMR 集群启用 Lake Formation。有关更多信息，请参阅使用 Amazon EMR 启用 Lake Formation。

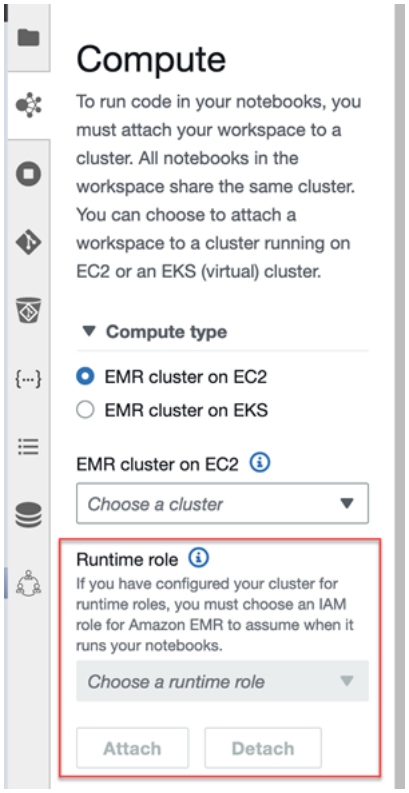
启动集群后，即可在 EMR Studio Workspace 中使用启用了运行时系统角色的集群。

> **Note**
>
> 当 `ExecutionEngineConfig.Type` 值为 `EMR` 时，StartNotebookExecution API 操作目前不支持 ExecutionRoleArn 值。

在 Workspace 中使用具有运行时系统角色的 EMR 集群

设置并启动集群后，即可将启用了运行时系统角色的集群与 EMR Studio Workspace 配合使用。

1. 创建新 Workspace 或启动现有 Workspace。有关更多信息，请参阅创建 EMR Studio Workspace。
2. 从打开的 Workspace 的左侧边栏中选择 EMR 集群选项卡，展开计算类型部分，然后从 EMR on EC2 集群菜单中选择自己的集群，从运行时系统角色菜单中选择运行时系统角色。



3. 选择附加将具有运行时系统角色的集群附加到 Workspace。

注意事项

在 Amazon EMR Studio Workspace 中使用支持运行时系统角色的集群时，请记住以下注意事项：

- 只有将 EMR Studio Workspace 附加到使用 Amazon EMR 6.11 或更高版本的 EMR 集群时，您才能选择运行时系统角色。
- 此页描述的运行时系统角色功能仅支持在 Amazon EC2 上运行的 Amazon EMR，不支持 EMR Serverless 交互式应用程序。要详细了解 EMR Serverless 的运行时系统角色，请参阅《Amazon EMR Serverless 用户指南》中的 Job runtime roles。
- 虽然在向集群提交任务时需要先配置其他权限，才能指定运行时系统角色，但无需其他权限即可访问 EMR Studio Workspace 生成的文件。此类文件的权限与没有运行时系统角色的集群生成的文件相同。
- 您不能在具有运行时系统角色的集群的 EMR Studio Workspace 中使用 SQL Explorer。当 Workspace 连接到启用了运行时系统角色的 EMR 集群时，Amazon EMR 会在用户界面中禁用 SQL Explorer。

- 您不能在 EMR Studio Workspace 中将协作模式与具有运行时系统角色的集群配合使用。当 Workspace 连接到启用了运行时系统角色的 EMR 集群时，Amazon EMR 会禁用 Workspace 协作功能。只有连接了 Workspace 的用户才能访问 Workspace。
- 您可能会遇到 Spark UI 发出的 Page may not be safe! 警告，如果集群启用了运行时系统角色。如果发生这种情况，请绕过警报继续查看 Spark UI。

以编程方式运行 Workspace 笔记本

> **Note**
>
> Amazon EMR Serverless 交互式应用程序不支持以编程方式执行笔记本电脑实例。

您可以使用脚本或在 Amazon CLI 上以编程方式运行 Amazon EMR Studio Workspace 笔记本。要了解如何以编程方式运行笔记本，请参阅 以编程方式执行 EMR Notebooks 的示例命令。

用 SQL Explorer 浏览数据

> **Note**
>
> Amazon EMR Serverless 交互式应用程序不支持适用于 EMR Studio 的 SQL Explorer。

本主题提供的信息有助于您开始在 Amazon EMR Studio 中使用 SQL Explorer。SQL Explorer 是 Workspace 中的一个单页工具，可帮助您了解 EMR 集群数据目录中的数据源。您可以使用 SQL Explorer 浏览数据、运行 SQL 查询以检索数据以及下载查询结果。

SQL Explorer 支持 Presto。要使用 SQL Explorer，您必须确保您的集群使用的是 Amazon EMR 版本 5.34.0 或更高版本或者版本 6.4.0 或更高版本，并且安装了 Presto。Amazon EMR Studio SQL Explorer 不支持配置了传输中加密的 Presto 集群。这是因为 Presto 在这些集群上将以 TLS 模式运行。

浏览集群的数据目录

SQL Explorer 提供了一个目录浏览器界面，您可以使用该界面来探索和了解数据的组织方式。例如，在编写 SQL 查询之前，您可以使用数据目录浏览器验证表和列名称。

浏览数据目录

1. 在 Workspace 中打开 SQL Explorer。
2. 确保您的 Workspace 已挂载到某个在 EC2 上运行、使用 Amazon EMR 6.4.0 或更高版本并且安装了 Presto 的 EMR 集群。您可以选择一个现有的集群或创建一个新的集群。有关更多信息，请参阅将计算资源附加到 EMR Studio Workspace。
3. 请从下拉列表中选择一个要浏览的 Database（数据库）。
4. 展开数据库中的表以查看表的列名称。您还可在搜索栏中输入一个关键词以筛选表结果。

运行 SQL 查询以检索数据

使用 SQL 查询检索数据并下载结果

1. 在 Workspace 中打开 SQL Explorer。
2. 确保您的 Workspace 已挂载到某个在 EC2 上运行并且安装了 Presto 和 Spark 的 EMR 集群。您可以选择一个现有的集群或创建一个新的集群。有关更多信息，请参阅将计算资源附加到 EMR Studio Workspace。
3. 选择 Open editor（打开编辑器）以在您的 Workspace 中打开一个新的编辑器选项卡。
4. 在编辑器选项卡中编写 SQL 查询。
5. 选择 运行。
6. 在 Result preview（结果预览）下查看查询结果。预设情况下，SQL Explorer 会显示前 100 个结果。您可以使用 Preview first 100 query results（预览前 100 个查询结果）下拉列表以选择要求显示的不同数量的结果（最高 1000）。
7. 选择 Download results（下载结果）以 CSV 格式下载结果。您最多可以下载 1000 行结果。

将计算资源附加到 EMR Studio Workspace

Amazon EMR Studio 使用 EMR 集群上的内核运行笔记本命令。您应将 Workspace 附加到某个使用 Amazon EC2 实例的集群，或某个 Amazon EMR on EKS 集群，或某个 EMR Serverless 应用程序，然后再选择内核。EMR Studio 允许您将 Workspaces 附加到新的或现有的集群，并让您无需关闭 Workspace 即可灵活地更改集群。

本节涵盖以下主题，可帮助您使用和预置 EMR Studio 集群：

- 将 Amazon EC2 集群附加到 EMR Studio Workspace
- 将 Amazon EMR on EKS 集群附加到 EMR Studio Workspace

- 将 Amazon EMR Serverless 应用程序附加到 EMR Studio Workspace
- 创建新的 EMR 集群并将其附加到 EMR Studio Workspace
- 从 EMR Studio Workspace 分离计算资源

将 Amazon EC2 集群附加到 EMR Studio Workspace

您可以在创建 Workspace 时将在 Amazon EC2 上运行的现有 EMR 集群附加到 Workspace，也可以将集群附加到现有 Workspace。如果您要创建和附加新集群，请参阅创建新的 EMR 集群并将其附加到 EMR Studio Workspace。

On create

在创建 Workspace 时附加 Amazon EMR 计算集群

1. 在 Create a Workspace (创建 Workspace) 对话框中，确保您已经为新 Workspace 选择了子网。展开 Advanced configuration (高级配置) 部分。
2. 在 Create a Workspace (创建 Workspace) 对话框中，确保您已经为新 Workspace 选择了子网。展开 Advanced configuration (高级配置) 部分。
3. 选择 Attach Workspace to an EMR cluster (将 Workspace 附加到 EMR 集群)。
4. 在 EMR 集群下拉列表中，选择现有 EMR 集群以将其附加到 Workspace。

附加集群后，您可以完成 Workspace 创建过程。首次打开新 Workspace 并选择 EMR 集群面板时，您应该会看到自己选择的集群已附加。

On launch

在启动 Workspace 时附加 Amazon EMR 计算集群

1. 导航到 Workspace 列表，然后选择要启动的 Workspace 所在行。然后，选择启动 Workspace > 使用选项启动。
2. 选择要附加到 Workspace 的 EMR 集群。

附加集群后，您可以完成 Workspace 创建过程。首次打开新 Workspace 并选择 EMR 集群面板时，您应该会看到自己选择的集群已附加。

In JupyterLab

在 JupyterLab 中将 Workspace 附加到 Amazon EMR 计算集群

1. 选择自己的 Workspace，然后选择启动 Workspace > 快速启动。
2. 在 JupyterLab 中，打开左侧边栏中的集群选项卡。
3. 选择 EMR on EC2 集群下拉列表，或者选择 Amazon EMR on EKS 集群。
4. 选择附加将集群附加到 Workspace。

附加集群后，您可以完成 Workspace 创建过程。首次打开新 Workspace 并选择 EMR 集群面板时，您应该会看到自己选择的集群已附加。

In the Workspace UI

从 Workspace 用户界面将 Workspace 附加 Amazon EMR 计算集群

1. 在要附加到集群的 Workspace 中，从左侧边栏中选择 EMR 集群图标打开集群面板。
2. 在集群类型下，展开下拉列表并选择 EMR on EC2 集群。
3. 从下拉列表中选择一个集群。您可能需要先分离现有集群才能启用集群选择下拉列表。
4. 选择 Attach (附加)。附加集群后，您应该会看到一条成功消息。

将 Amazon EMR on EKS 集群附加到 EMR Studio Workspace

除了使用在 Amazon EC2 上运行的 Amazon EMR 集群之外，您还可以将 Workspace 附加到 Amazon EMR on EKS 集群以运行笔记本代码。有关 Amazon EMR on EKS 的更多信息，请参阅 What is Amazon EMR on EKS。

在您将 Workspace 连接到 Amazon EMR on EKS 集群之前，您的 Studio 管理员必须授予您访问权限。

On create

在创建 Workspace 时附加 Amazon EMR on EKS 集群

1. 在 Create a Workspace（创建 Workspace）对话框中，展开 Advanced configuration（高级配置）部分。
2. 选择将 Workspace 附加到 Amazon EMR on EKS 集群。
3. 在 Amazon EMR on EKS 集群下，从下拉列表中选择集群。

4. 在 Select an endpoint (选择终端节点) 下，选择要附加到 Workspace 的托管式终端节点。托管式终端节点是允许 EMR Studio 与您选择的集群进行通信的网关。
5. 选择创建 Workspace 完成 Workspace 创建过程并附加选定的集群。

附加集群后，您可以完成 Workspace 创建过程。首次打开新 Workspace 并选择 EMR 集群面板时，您应该会看到自己选择的集群已附加。

In the Workspace UI

从 Workspace 用户界面附加 Amazon EMR on EKS 集群

1. 在要附加到集群的 Workspace 中，从左侧边栏中选择 EMR 集群图标打开集群面板。
2. 展开集群类型下拉列表并选择 Amazon EMR on EKS 集群。
3. 在 EMR on EKS 集群下，从下拉列表中选择集群。
4. 在 Endpoint (终端节点) 下，选择要附加到 Workspace 的托管式终端节点。托管式终端节点是允许 EMR Studio 与您选择的集群进行通信的网关。
5. 选择 Attach (附加)。附加集群后，您应该会看到一条成功消息。

将 Amazon EMR Serverless 应用程序附加到 EMR Studio Workspace

您可以将 Workspace 附加到 EMR Serverless 应用程序来运行交互式工作负载。有关更多信息，请参阅通过 Using notebooks to run interactive workloads with EMR Serverless through EMR Studio。

Example 将 Workspace 附加到 JupyterLab 中的 EMR Serverless 应用程序

您的账户管理员必须首先按 Required permissions for interactive workloads 中所述向您授予访问权限，然后才能将 Workspace 附加到 EMR Serverless 应用程序。

1. 导航至 EMR Studio，选择您的 Workspace，然后选择启动 Workspace > 快速启动。
2. 在 JupyterLab 中，打开左侧边栏中的集群选项卡。
3. 选择将 EMR Serverless 作为计算选项，然后选择一个 EMR Serverless 应用程序和一个运行时系统角色。
4. 要将集群附加到您的 Workspace，请选择附加。

现在打开此 Workspace 时，您应该会看到已经附加了所选的应用程序。

创建新的 EMR 集群并将其附加到 EMR Studio Workspace

高级 EMR Studio 用户可以预置在 Amazon EC2 上运行的新 EMR 集群，以便与 Workspace 一起使用。预设情况下，新集群将安装 EMR Studio 所需的所有大数据应用程序。

要创建集群，您的 Studio 管理员必须首先使用会话策略授予您权限。有关更多信息，请参阅 为 EMR Studio 用户创建权限策略。

您可以在 Create a Workspace (创建 Workspace) 对话框中或从 Workspace UI 的 Cluster (集群) 面板中创建新集群。无论哪种方式，您都有两个集群创建选项：

1. Create an EMR cluster (创建 EMR 集群) – 通过选择 Amazon EC2 实例类型和数量来创建 EMR 集群。
2. Use a cluster template (使用集群模板) – 通过选择预定义的集群模板预置集群。当您拥有使用集群模板的权限时，才会显示此选项。

通过提供集群配置创建 EMR 集群

1. 选择一个起点。

要……	请执行此操作...
使用 Create a Workspace (创建 Workspace) 对话框在创建 Workspace 时创建集群。	展开 Create a Workspace (创建 Workspace) 对话框中的 Advanced configuration (高级配置) 部分，然后选择 Create an EMR cluster (创建 EMR 集群)。
创建 Workspace 后，从 Workspace UI 中的 EMR 集群面板创建集群。	从打开的 Workspace 的左侧边栏中选择 EMR 集群选项卡，展开高级配置部分，然后选择创建集群。

2. 输入 Cluster name (集群名称)。命名集群有助于您稍后在 EMR Studio 集群列表中找到它。
3. 对于 Amazon EMR 版本，请为集群选择一个 Amazon EMR 发行版本。
4. 对于 Instance (实例)，为集群选择 Amazon EC2 实例的类型和数量。有关选择实例类型的更多信息，请参阅配置 Amazon EC2 实例。一个实例将用作主节点。
5. 选择 EMR Studio 可以在其中启动新集群的子网。每个子网选项都经过您的 Studio 管理员的预先批准，您的 Workspace 应该能够连接到列出的任何子网中的集群。
6. 选择用于日志存储的 S3 URI。

7. 选择 Create EMR cluster (创建 EMR 集群) 来预置集群。如果您使用创建 Workspace 对话框，请选择创建 Workspace 来创建 Workspace 并预置集群。EMR Studio 预置新集群后，它会将集群附加到 Workspace。

使用集群模板创建集群

1. 选择一个起点。

要……	请执行此操作...
使用 Create a Workspace (创建 Workspace) 对话框在创建 Workspace 时创建集群。	展开 Create a Workspace (创建 Workspace) 对话框中的 Advanced configuration (高级配置) 部分，然后选择 Use a cluster template (使用集群模板)。
从 Workspace UI 中的 EMR 集群面板创建集群。	从打开的 Workspace 的左侧边栏中选择 EMR 集群选项卡，展开高级配置部分，然后选择集群模板。

2. 从下拉列表中选择集群模板。每个可用的集群模板都包含一个简要说明，以帮助您进行选择。
3. 您选择的集群模板可能有其它参数，例如 Amazon EMR 发行版或集群名称。您可以选择或插入值，或使用管理员选择的默认值。
4. 选择 EMR Studio 可以在其中启动新集群的子网。每个子网选项都经过您的 Studio 管理员的预先批准，您的 Workspace 应该能够连接到任何子网中的集群。
5. 选择 Use cluster template (使用集群模板) 来预置集群并将其附加到 Workspace。EMR Studio 将需要几分钟时间来创建集群。如果您使用创建 Workspace 对话框，请选择创建 Workspace 来创建 Workspace 并预置集群。EMR Studio 预置您的新集群后，它会将集群附加到 Workspace。

从 EMR Studio Workspace 分离计算资源

要交换附加到 Workspace 的集群，您可以从 Workspace UI 分离集群。

从 Workspace 分离集群

1. 在要分离集群的 Workspace 中，从左侧边栏中选择 EMR 集群图标打开集群面板。
2. 在 Select cluster (选择集群) 下，选择 Detach (分离) 并等待 EMR Studio 分离集群。集群分离后，您将看到一条成功消息。

从 EMR Studio Workspace 分离 EMR Serverless 应用程序

要交换附加到 Workspace 的计算资源，您可以通过 Workspace UI 分离应用程序。

1. 在要从集群分离的 Workspace 中，从左侧边栏中选择 Amazon EMR 计算图标，从而打开计算面板。
2. 在选择计算资源下，选择分离，然后等待 EMR Studio 完成应用程序的分离。应用程序分离完成后，您将看到一条成功消息。

将基于 Git 的存储库链接到 EMR Studio Workspace

关于 EMR Studio 的 Git 存储库

您最多可以将三个 Git 存储库与一个 EMR Studio Workspace 相关联。默认情况下，每个 Workspace 都允许您从与 Studio 相同的 Amazon 账户关联的 Git 存储库列表中进行选择。您还可以创建新的 Git 存储库作为 Workspace 资源。

在连接到集群的主节点时，您可以使用终端命令运行如下所示的 Git 命令。

```
!git pull origin <branch-name>
```

此外，您也可以使用 jupyterlab-git 扩展程序。从左侧边栏中选择 Git 图标打开它。有关适用于 JupyterLab 的 jupyterlab-git 扩展的信息，请参阅 jupyterlab-git。

先决条件

- 要将 Git 存储库与 Workspace 关联，Studio 必须配置为允许 Git 存储库链接。您的 Studio 管理员应该执行为基于 Git 的存储库建立访问和权限中所述的步骤。
- 如果您使用 CodeCommit 存储库，则必须使用 Git 凭证和 HTTPS。不支持 SSH 密钥和带有 Amazon Command Line Interface 凭证辅助程序的 HTTPS。CodeCommit 也不支持个人访问令牌（PAT）。有关更多信息，请参阅《IAM 用户指南》中的将 IAM 与 CodeCommit 结合使用和《Amazon CodeCommit 用户指南》中的 Setup for HTTPS users using Git credentials。

说明

将关联的 Git 存储库链接到 Workspace

1. 从 Studio 的 Workspaces 列表中打开要链接到存储库的 Workspace。

2. 从左侧边栏中选择 Amazon EMR Git 存储库图标打开 Git 存储库工具面板。
3. 在 Git repositories (Git 存储库) 下，展开下拉列表并最多选择三种存储库以链接到 Workspace。EMR Studio 注册您的选择并开始链接每个存储库。

链接过程可能需要一些时间才能完成。您可以在 Git repository (Git 存储库) 工具面板中查看您选择的每个存储库的状态。EMR Studio 将存储库链接到 Workspace 后，您应该会在 File browser (文件浏览器) 面板中看到属于该存储库的文件。

将新的 Git 存储库作为资源添加到 Workspace

1. 从 Studio 的 Workspaces 列表中打开要链接到存储库的 Workspace。
2. 从左侧边栏中选择 Amazon EMR Git 存储库图标打开 Git 存储库工具面板。
3. 选择 Add new Git repository (添加新的 Git 存储库)。
4. 对于 Repository name (存储库名称)，输入 EMR Studio 中存储库的描述性名称。名称只能包含字母数字字符、连字符和下划线。
5. 对于 Git repository URL (Git 存储库 URL)，输入存储库的 URL。当您使用 CodeCommit 存储库时，这是在您选择 Clone URL (克隆 URL) 和 Clone HTTPS (克隆 HTTPS) 时复制的 URL。例如，`https://git-codecommit.us-west-2.amazonaws.com/v1/repos/[MyCodeCommitRepoName]`。
6. 对于 Branch (分支)，输入要签出的现有分支的名称。
7. 对于 Git credentials (Git 凭证)，请根据以下准则选择选项。EMR Studio 使用存储在 Secrets Manager 中的密钥访问您的 Git 凭证。

> **Note**
>
> 如果您使用 GitHub 存储库，我们建议您使用个人访问令牌（PAT）进行身份验证。从 2021 年 8 月 13 日开始，GitHub 将需要基于令牌的身份验证，并且在对 Git 操作进行身份验证时将不再接受密码。有关更多信息，请参阅 GitHub 博客中的 Token authentication requirements for Git operations 一文。

选项	描述
创建新密钥	选择此选项可将现有 Git 凭证与将在 Amazon Secrets Manager 中为您创建的新密钥相关

选项	描述
	联。根据您用于存储库的 Git 凭证，执行以下操作之一。 如果您使用 Git 用户名和密码访问存储库，请选择 Username and password（用户名和密码），输入要在 Secrets Manager 中使用的 Secret name（密钥名称），然后输入要与密钥关联的 Username（用户名）和 Password（密码）。 –或者– 如果您使用个人访问令牌访问存储库，请选择 Personal access token (PAT) (个人访问令牌 (PAT))，输入要在 Secrets Manager 中使用的 Secret name (密钥名称)，然后输入您的个人访问令牌。有关更多信息，请参阅为 GitHub 的命令行创建个人访问令牌和 Bitbucket 的个人访问令牌。CodeCommit 存储库不支持此选项。
在没有凭证的情况下使用公有存储库	选择此选项可访问公有存储库。
使用现有 Amazon 密钥	如果您已将凭证保存为 Secrets Manager 中的密钥，请选择此选项，然后从列表中选择密钥名称。 如果您选择与 Git 用户名和密码关联的密钥，则密钥必须采用 `{"gitUsername": "MyUserName ", "gitPassword": "MyPassword "}` 格式。

8. 选择 Add repository (添加存储库) 创建新存储库。EMR Studio 创建新存储库后，您将看到一条成功消息。新存储库显示在 Git repositories (Git 存储库) 下的下拉列表中。
9. 要将新存储库链接到您的 Workspace，请从 Git repositories (Git 存储库) 下的下拉列表中选择它。

链接过程可能需要一些时间才能完成。在 EMR Studio 将新存储库链接到 Workspace 后，您应该会在 File Browser (文件浏览器) 面板中看到一个与您的存储库同名的新文件夹。

要打开不同的链接存储库，请在 File browser (文件浏览器) 中导航到其文件夹。

使用 EMR Studio 调试应用程序和任务

借助 Amazon EMR Studio，您可以启动数据应用程序界面，以便在浏览器中分析应用程序和任务运行情况。

您还可以从 Amazon EMR 控制台为在 EC2 集群上运行的 Amazon EMR 启动持久性的集群外用户界面。有关更多信息，请参阅 查看持久性应用程序用户界面。

> **Note**
>
> 根据您的浏览器设置，您可能需要启用弹出窗口才能打开应用程序 UI。

有关配置和使用应用程序界面的信息，请参阅 YARN 时间线服务器、监控和检测或 Tez UI 概览。

调试 Amazon EC2 任务上运行的 Amazon EMR

Workspace UI

从笔记本文件启动集群上 UI

当您使用 Amazon EMR 发行版本 5.33.0 及更高版本时，您可以从 Workspace 中的笔记本启动 Spark Web 用户界面（Spark UI 或 Spark 历史记录服务器）。

集群上 UI 可与 PySpark、Spark 或 SparkR 内核配合使用。Spark 事件日志或容器日志的最大可查看文件大小为 10MB。如果您的日志文件超过 10MB，我们建议您使用持久性 Spark 历史记录服务器而不是集群上 Spark UI 来调试任务。

> **Important**
>
> 为了让 EMR Studio 从 Workspace 启动集群上应用程序用户界面，集群必须能够与 Amazon API Gateway 进行通信。您必须配置 EMR 集群以允许网络流量传出到 Amazon API Gateway，并确保可从集群访问 Amazon API Gateway。
> Spark UI 通过解析主机名来访问容器日志。如果您使用自定义域名，您必须确保您的集群节点的主机名可以由 Amazon DNS 或您指定的 DNS 服务器进行解析。为此，请为与您

的集群关联的 Amazon Virtual Private Cloud（VPC）设置动态主机配置协议（DHCP）选项。有关 DHCP 选项的更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的 DHCP 选项集。

1. 在您的 EMR Studio 中，打开您要使用的 Workspace，并确保它附加到在 EC2 上运行的 Amazon EMR 集群。有关说明，请参阅 将计算资源附加到 EMR Studio Workspace。
2. 打开笔记本文件并使用 PySpark、Spark 或 SparkR 内核。要选择内核，请从笔记本工具栏的右上角选择内核名称以打开 Select Kernel (选择内核) 对话框。如果没有选择任何内核，名称显示为 No Kernel! (无内核！)。
3. 运行笔记本代码。启动 Spark 上下文时，以下内容将作为输出显示在笔记本中。它可能需要几秒钟时间才能显示。如果您已启动 Spark 上下文，则可以随时运行 `%%info` 命令以访问指向 Spark UI 的链接。

> **Note**
>
> 如果 Spark UI 链接不起作用或几秒钟后未显示，请创建一个新的笔记本单元并运行 `%%info` 命令以重新生成链接。

```
[1]: sc
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
2	application_1613085840432_0003	spark	idle	Link	Link	✔

```
SparkSession available as 'spark'.

res1: org.apache.spark.SparkContext = org.apache.spark.SparkContext@58262802
```

4. 要启动 Spark UI，请选择 Spark UI 下的 Link (链接)。如果您的 Spark 应用程序正在运行，Spark UI 将在新选项卡中打开。如果应用程序已完成，则会打开 Spark 历史记录服务器。

   启动 Spark UI 后，您可以在浏览器中修改 URL 以打开 YARN ResourceManager 或 Yarn 时间线服务器。将下列路径之一添加到 `amazonaws.com` 之后。

Web UI	路径	修改后的 URL 示例
YARN ResourceManager	/rm	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com/*rm*
Yarn 时间线服务器	/yts	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com/*yts*
Spark 历史记录服务器	/shs	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com/*shs*

Studio UI

从 EMR Studio UI 中启动持久性 YARN 时间线服务器、Spark 历史记录服务器或 Tez UI

1. 在您的 EMR Studio 中，从页面的左侧选择 Amazon EMR on EC2 打开 Amazon EMR on EC2 集群列表。
2. 通过在搜索框中输入值，按 name (名称)、state (状态) 或 ID 筛选集群列表。您还可以按创建时间范围进行搜索。
3. 选择集群，然后选择 Launch application UIs (启动应用程序 UI) 以选择应用程序用户界面。应用程序 UI 将在新的浏览器选项卡中打开，加载可能需要一些时间。

调试在 EMR Serverless 上运行的 EMR Studio

与在 Amazon EC2 上运行的 Amazon EMR 类似，您可以使用 Workspace 用户界面来分析您的 EMR Serverless 应用程序。在 Workspace UI 中，当您使用 Amazon EMR 发行版 6.14.0 及更高版本时，您可以从 Workspace 中的笔记本电脑实例启动 Spark Web 用户界面（Spark UI 或 Spark 历史记录服务器）。为方便起见，我们还提供了指向驱动程序日志的链接，便于您快速访问 Spark 驱动程序日志。

使用 Spark 历史记录服务器调试 Amazon EMR on EKS 任务运行

当您向 Amazon EMR on EKS 集群提交任务运行时，可以使用 Spark 历史记录服务器访问该任务运行的日志。Spark 历史记录服务器提供了监控 Spark 应用程序的工具，例如：调度流程阶段和任务列表、RDD 大小和内存使用情况概述以及环境信息。您可以通过以下方式启动 Amazon EMR on EKS 任务运行的 Spark 历史记录服务器：

- 当您使用具有 EMR on EKS 托管式端点的 Amazon EMR Studio 提交任务运行时，可以通过 Workspace 的笔记本文件启动 Spark 历史记录服务器。
- 当您使用 Amazon EMR on EKS 的 Amazon CLI 或者 Amazon SDK 提交任务运行时，可以通过 EMR Studio UI 启动 Spark 历史记录服务器。

有关如何使用 Spark 历史记录服务器的信息，请参阅 Apache Spark 文档的监控和仪表。有关任务运行的更多信息，请参阅《Amazon EMR on EKS 开发指南》中的概念和组件。

从 EMR Studio Workspace 中的笔记本文件启动 Spark 历史记录服务器

1. 打开连接到 Amazon EMR on EKS 集群的 Workspace。
2. 在 Workspace 中选择并打开笔记本文件。
3. 选择笔记本文件顶部的 Spark UI，以在新选项卡打开持久性 Spark 历史记录服务器。

通过 EMR Studio UI 启动 Spark 历史记录服务器

> **Note**
>
> EMR Studio UI 中的任务列表仅显示：您使用 Amazon EMR on EKS 的 Amazon CLI 或者 AmazonSDK 提交的任务运行。

1. 在您的 EMR Studio 中，从页面左侧选择 Amazon EMR on EKS。
2. 搜索您常用于提交任务运行的 Amazon EMR on EKS 虚拟集群。您可在搜索框中输入值，按 status (状态) 或 ID 筛选集群列表。
3. 选择集群以打开其详细信息页面。详细信息页面显示有关集群的信息，例如 ID、命名空间和状态。该页面还显示了提交至该集群的所有运行任务的列表。
4. 在集群详细信息页面中选择要调试的任务运行。
5. 在 Jobs (任务) 列表的右上角，选择 Launch Spark History Server (启动 Spark 历史记录服务器)，以在新的浏览器选项卡中打开应用程序界面。

在 EMR Studio Workspace 中安装内核和库

每个 Amazon EMR Studio Workspace 均附带一组预安装的库和内核。

在 Amazon EC2 上运行的集群的内核和库

当您使用在 Amazon EC2 上运行的 EMR 集群时，您还可以通过以下方式为 EMR Studio 自定义环境：

- 在集群主节点上安装 Jupyter notebook 内核和 Python 库 – 使用此选项安装库时，库由附加到同一集群的所有 Workspace 共享。您可以从笔记本单元格中安装内核或库，也可以在使用 SSH 连接到集群的主节点时安装内核或库。
- 使用笔记本范围内的库 — 当 Workspace 用户从笔记本电脑单元中安装和使用库时，这些库仅适用于该笔记本电脑。此选项允许同一集群的不同笔记本工作，而不必担心库版本冲突。

EMR Studio Workspaces 与 EMR Notebooks 具有相同的底层架构。您可以使用 EMR Studio 安装和使用 Jupyter notebook 内核和 Python 库，就像使用 EMR Notebooks 一样。有关说明，请参阅 安装并使用内核和库。

Amazon EMR on EKS 集群上的内核和库

Amazon EMR on EKS 集群包括 PySpark 和 Python 3.7 内核以及一组预安装的库。Amazon EMR on EKS 不支持安装其他的库或集群。

每个 Amazon EMR on EKS 集群都安装了以下 Python 和 PySpark 库：

- Python – boto3, cffi, future, ggplot, jupyter, kubernetes, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn
- PySpark – ggplot, jupyter, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn

EMR Serverless 应用程序的内核和库

每个 EMR Serverless 应用程序都预装了以下 Python 和 PySpark 库：

- Python – ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn
- PySpark – ggplot, matplotlib,numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn

使用 magic 命令增强内核

概述

EMR Studio 和 EMR Notebooks 支持 magic 命令。Magic 命令又称 magics，是 IPython 内核为帮助运行和分析数据而提供的增强功能。IPython 是用 Python 构建的交互式 Shell 环境。

Amazon EMR 还支持 Sparkmagic，该软件包为 Spark 相关内核（PySpark、SparkR 和 Scala 内核）提供特定的 magic 命令，并使用集群上的 Livy 提交 Spark 作业。

只要您的 EMR Notebooks 中有 Python 内核，就可以使用 magic 命令。同样，任何与 Spark 相关的内核都支持 Sparkmagic 命令。

Magic 命令也称作 magics，共有两类：

- 行 magics% – 此类 命令用单个 magic 前缀表示，并在单行代码上运行
- 单元格 magics – 此类 magic 命令用双 %% 前缀表示，并在多行代码上运行

有关所有可用的 magics，请参阅 列出 magic 和 Sparkmagic 命令。

注意事项和限制

- EMR Serverless 不支持使用 %%sh 来运行 spark-submit，也不支持 EMR Notebooks magics。
- Amazon EMR on EKS 集群不支持将 Sparkmagic 命令用于 EMR Studio。这是因为拥有托管式端点的 Spark 内核属于 Kubernetes 内置的内核，不支持 Sparkmagic 和 Livy。作为暂时的解决方法，您可以将 Spark 配置直接设置到 SparkContext 对象，如以下示例所示。

```
spark.conf.set("spark.driver.maxResultSize", '6g')
```

- Amazon 禁止以下 magic 命令和操作：
  - %alias
  - %alias_magic
  - %automagic
  - %macro
  - 使用 %configure 修改 proxy_user
  - 使用 %env 或 %set_env 修改 KERNEL_USERNAME

列出 magic 和 Sparkmagic 命令

使用以下命令列出可用的 magic 命令：

- `%lsmagic` 会列出当前可用的所有 magic 函数。
- `%%help` 会列出 Sparkmagic 软件包提供的当前可用 Spark 相关 magic 函数。

使用 `%%configure` 来配置 Spark

Sparkmagic 最有用的命令之一是 `%%configure` 命令，该命令可配置会话创建参数。借助 `conf` 设置，您可以配置 Apache Spark 配置文档 中提及的任何 Spark 配置。

Example 将外部 JAR 文件从 Maven 存储库或 Amazon S3 添加到 EMR Notebooks

您可以使用以下方法将外部 JAR 文件依赖项添加到 Sparkmagic 支持的任何 Spark 相关内核中。

```
%%configure -f
{"conf": {
    "spark.jars.packages": "com.jsuereth:scala-arm_2.11:2.0,ml.combust.bundle:bundle-
ml_2.11:0.13.0,com.databricks:dbutils-api_2.11:0.0.3",
    "spark.jars": "s3://DOC-EXAMPLE-BUCKET/my-jar.jar"
    }
}
```

Example : 配置 Hudi

您可以使用笔记本编辑器来配置 EMR Notebook 以使用 Hudi。

```
%%configure
{ "conf": {
     "spark.jars": "hdfs://apps/hudi/lib/hudi-spark-bundle.jar,hdfs:///apps/hudi/lib/
spark-spark-avro.jar",
     "spark.serializer": "org.apache.spark.serializer.KryoSerializer",
     "spark.sql.hive.convertMetastoreParquet":"false"
     }
}
```

使用 %%sh 运行 spark-submit

在附加的集群实例上，%%sh magic 会通过子进程来运行 Shell 命令。通常，您需要使用其中一个 Spark 相关的内核在附加的集群上运行 Spark 应用程序。但如果要使用 Python 内核提交 Spark 应用程序，可以使用以下 magic，并将存储桶名称替换为小写的储桶名称。

```
%%sh
spark-submit --master yarn --deploy-mode cluster s3://DOC-EXAMPLE-BUCKET/test.py
```

在此示例中，集群需要访问 s3://*DOC-EXAMPLE-BUCKET*/test.py 的位置，否则该命令将失败。

您可以将 %%sh magic 与任何 Linux 命令结合使用。如果您要运行任何 Spark 或 YARN 命令，请使用以下选项之一创建 emr-notebook Hadoop 用户，并授予用户运行这些命令的权限。

- 您可以通过运行以下命令显式创建新用户。

  ```
  hadoop fs -mkdir /user/emr-notebook
  hadoop fs -chown emr-notebook /user/emr-notebook
  ```

- 您可以在 Livy 中启用用户模拟功能，从而自动创建用户。参阅 启用用户模拟以监控 Spark 用户和任务活动 了解更多信息。

使用 %%display 可视化显示 Spark 数据框

您可以使用 %%display magic 来可视化显示 Spark 数据框。要使用此 magic，请运行以下命令。

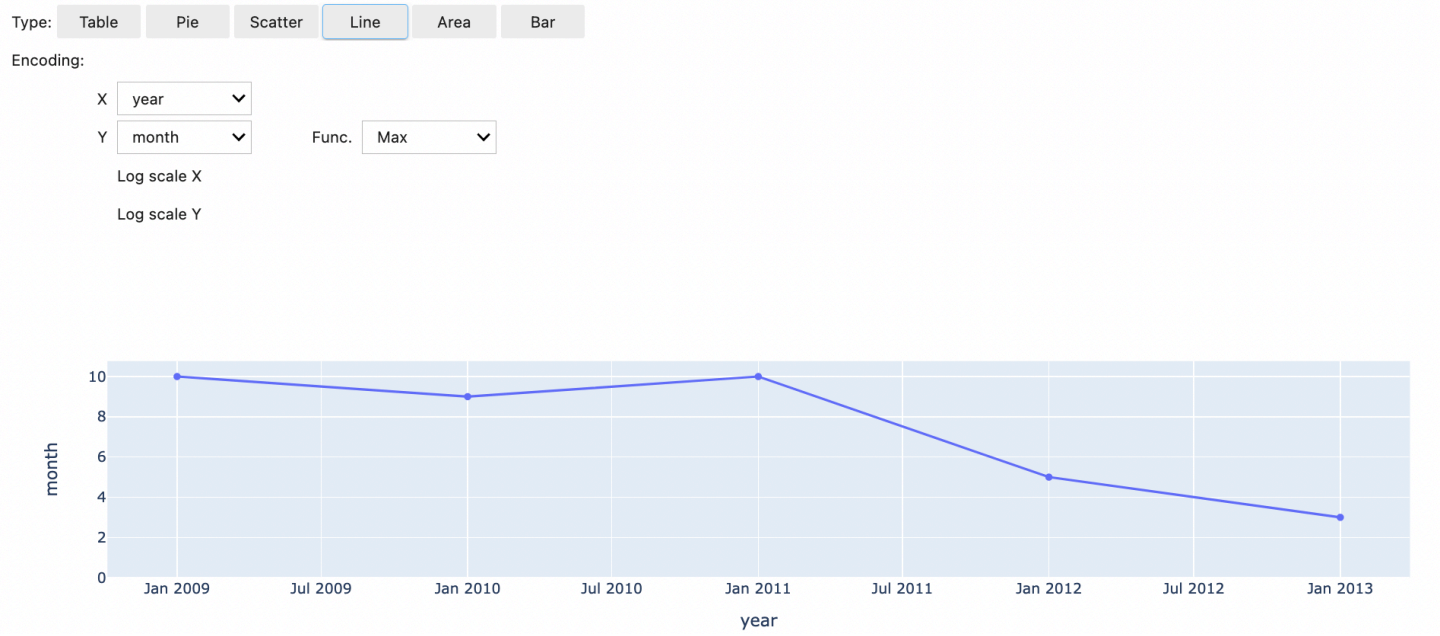
```
%%display df
```

选择以表格格式查看结果，如下图所示。

Type: Table Pie Scatter Line Area Bar

year	month	total_passengers	total_trips
2012-01-01	3	26866837	16146923
2011-01-01	3	26091246	16066350
2013-01-01	3	26965079	15749228
2011-01-01	10	26287953	15707756
2009-01-01	10	26202049	15604551
2012-01-01	5	26278817	15567525
2011-01-01	5	25508952	15554868
2010-01-01	9	25533166	15540209
2010-01-01	5	26002858	15481351
2012-01-01	4	25900645	15477914

您也可以选择使用五种类型的图表来可视化数据。您的选项包括饼图、散点图、折线图、面积图和条形图。



使用 EMR Notebooks magics

Amazon EMR 提供了以下 EMR Notebooks magics，您可以将这些命令用于 Python3 和基于 Spark 的内核：

- `%mount_workspace_dir` – 将 Workspace 目录挂载到集群，以便您可以从 Workspace 中的其他文件导入和运行代码

> **Note**
>
> 使用 `%mount_workspace_dir` 时，只有 Python 3 内核可以访问本地文件系统。Spark 执行器将无法使用此内核访问挂载的目录。

- `%umount_workspace_dir` - 从集群中卸载 Workspace 目录
- `%generate_s3_download_url` - 在笔记本输出中为 Amazon S3 对象生成临时下载链接

先决条件

您需要完成以下任务后才能安装 EMR Notebooks magics：

- 确保您的 集群 EC2 实例（EC2 实例配置文件）的服务角色 拥有 Amazon S3 的读取访问权限。使用 `AmazonElasticMapReduceforEC2Role` 托管策略的 `EMR_EC2_DefaultRole` 满足此要求。如果您使用自定义角色或策略，请确保它拥有必要的 S3 权限。

> **Note**
>
> EMR Notebooks magics 以笔记本电脑实例用户身份在集群上运行，并使用 EC2 实例配置文件与 Amazon S3 进行交互。在 EMR 集群上挂载 Workspace 目录时，拥有附加到该集群权限的所有 Workspace 和 EMR Notebooks 都可以访问已挂载的目录。
> 预设情况下，目录将以只读方式挂载。虽然 `s3fs-fuse` 和 `goofys` 允许读-写挂载，但我们强烈建议您不要将挂载参数修改为以读-写模式挂载目录。如果您允许写入权限，则对目录所做的任何更改都将写入 S3 存储桶。为避免意外删除或覆盖，您可以为 S3 存储桶启用版本控制。要了解更多信息，请参阅在 S3 存储桶中使用版本控制。

- 在集群上运行以下脚本中的一种，从而安装 EMR Notebooks magics 的依赖项。要运行脚本，您可以 使用自定义引导操作，或当您已经有正在运行的集群时，按照在 Amazon EMR 集群上运行命令和脚本中的说明进行操作。

您可以选择要安装哪个依赖项。s3fs-fuse 和 goofys 都是 FUSE（用户空间中的文件系统）工具，这些工具允许您将 Amazon S3 存储桶作为本地文件系统挂载到集群上。s3fs 工具可提供类似于 POSIX 的体验。当您更喜欢性能而不是符合 POSIX 标准的文件系统时，goofys 工具是不错的选择。

```
#!/bin/sh

# Install the s3fs dependency for EMR Notebooks magics sudo amazon-linux-extras
 install epel -y
sudo yum install s3fs-fuse -y
```

或者

```
#!/bin/sh

# Install the goofys dependency for EMR Notebooks magics sudo wget https://
github.com/kahing/goofys/releases/latest/download/goofys -P /usr/bin/
sudo chmod ugo+x /usr/bin/goofys
```

安装 EMR Notebooks magics

> **Note**
>
> 在 Amazon EMR 发行版 6.0 到 6.9.0 和 5.0 到 5.36.0 中，只有 emr-notebooks-magics 程序包版本 0.2.0 及更高版本支持 %mount_workspace_dir magic。

完成以下步骤以安装 EMR Notebooks magics。

1. 在您的笔记本中，运行下面的命令以安装 emr-notebooks-magics 程序包。

   ```
   %pip install boto3 --upgrade
   %pip install botocore --upgrade
   %pip install emr-notebooks-magics --upgrade
   ```

2. 重启内核以加载 EMR Notebooks magics。
3. 使用下面的命令验证安装，该命令应该显示用于 %mount_workspace_dir 的输出帮助文本。

```
%mount_workspace_dir?
```

使用 %mount_workspace_dir 挂载 Workspace 目录

您可以借助 %mount_workspace_dir magic 将 Workspace 目录挂载到 EMR 集群上，以便您能导入和运行存储在目录中的其他文件、模块或程序包。

下面的示例将整个 Workspace 目录挂载到集群上，并指定可选的 *<--fuse-type>* 参数，以使用 goofys 挂载目录。

```
%mount_workspace_dir . <--fuse-type goofys>
```

要验证您的 Workspace 目录是否已挂载，请使用下面的示例，借助 ls 命令来显示当前的工作目录。输出结果应该显示 Workspace 中的所有文件。

```
%%sh
ls
```

在 Workspace 中完成更改后，可以使用下面的命令卸载 Workspace 目录：

> **Note**
>
> 即使 Workspace 已停止或已分离，您的 Workspace 目录仍会保持挂载到集群上。您必须显式卸载 Workspace 目录。

```
%umount_workspace_dir
```

使用 %generate_s3_download_url 下载 Amazon S3 对象

generate_s3_download_url 命令可为存储在 Amazon S3 中的对象创建预签名 URL。您可以使用该预签名 URL 将对象下载到本地计算机。例如，您可以运行 generate_s3_download_url 下载代码写入 Amazon S3 的 SQL 查询的结果。

预设情况下，预签名 URL 的有效期为 60 分钟。您可以通过指定 --expires-in 标志的秒数来更改过期时间。例如，--expires-in 1800 创建一个有效期为 30 分钟的 URL。

下面的示例通过指定完整的 Amazon S3 路径来为对象生成下载链接：*s3://EXAMPLE-DOC-BUCKET/path/to/my/object*。

```
%generate_s3_download_url s3://EXAMPLE-DOC-BUCKET/path/to/my/object
```

要了解有关使用 generate_s3_download_url 的更多信息，请运行下面的命令以显示帮助文本。

```
%generate_s3_download_url?
```

使用 **%execute_notebook** 在无头模式下运行笔记本

您可以使用 %execute_notebook magic 在无头模式下运行另一个笔记本电脑实例，并查看运行的每个单元格的输出。此 magic 命令需要拥有 Amazon EMR 和 Amazon EC2 共享的实例角色的额外权限。有关如何授予额外权限的更多信息，请运行命令 %execute_notebook?。

如果有长时间运行的任务，系统可能会因为不活动而进入睡眠状态，也可能暂时失去互联网连接。这可能会中断浏览器与 Jupyter 服务器之间的连接。如果出现这种情况，您可能会丢失自己运行的单元格和 Jupyter 服务器发送的单元格的输出内容。

如果使用 %execute_notebook magic 在无头模式下运行笔记本电脑实例，即使本地网络出现中断，EMR Notebooks 也会捕获已运行单元格的输出。EMR Notebooks 将输出以增量方式保存在与您运行的笔记本同名的新笔记本中。然后，EMR Notebooks 将笔记本放入 Workspace 内的新文件夹。无头运行发生在同一个集群上并使用 EMR_Notebook_DefaultRole 服务角色，但其他参数可能会更改默认值。

要在无头模式下运行笔记本，请使用以下命令：

```
%execute_notebook <relative-file-path>
```

要为无头运行指定集群 ID 和服务角色，请使用以下命令：

```
%execute_notebook <notebook_name>.ipynb --cluster-id <emr-cluster-id> --service-role
 <emr-notebook-service-role>
```

当 Amazon EMR 和 Amazon EC2 共享一个实例角色时，该角色需要以下额外权限：

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:StartNotebookExecution",
                "elasticmapreduce:DescribeNotebookExecution",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::<AccoundId>:role/EMR_Notebooks_DefaultRole"
        }
    ]
}
```

> **Note**
>
> 要使用 `%execute_notebook` magic，请安装 `emr-notebooks-magics` 程序包版本 0.2.3 或更高版本。

将多语言笔记本与 Spark 内核配合使用

每个 Jupyter notebook 内核都有默认语言。例如，Spark 内核的默认语言是 Scala，而 PySpark 内核的默认语言是 Python。借助 Amazon EMR 6.4.0 及更高版本，EMR Studio 支持多语言笔记本电脑。这意味着除了默认语言 Python、Spark、R 和 Spark SQL 之外，EMR Studio 中的每个内核都可以支持以下语言。

要激活此功能，请在任何单元格的开头指定以下一种 magic 命令。

语言	命令
Python	`%%pyspark`
Scala	`%%scalaspark`

语言	命令
R	%%rspark 不支持将交互式工作负载与 EMR Serverless 结合使用。
Spark SQL	%%sql

调用后，这些命令会使用相应语言的解释器在同一 Spark 会话中执行整个单元格。

用户可以使用 %%pyspark 单元格 magic 在所有 Spark 内核中编写 PySpark 代码。

```
%%pyspark
a = 1
```

用户可以使用 %%sql 单元格 magic 在所有 Spark 内核中执行 Spark-SQL 代码。

```
%%sql
SHOW TABLES
```

用户可以使用 %%rspark 单元格 magic 在所有 Spark 内核中执行 SparkR 代码。

```
%%rspark
a <- 1
```

用户可以使用 %%scalaspark 单元格 magic 在所有 Spark 内核中执行 Spark Scala 代码。

```
%%scalaspark
val a = 1
```

使用临时表格在多个语言解释器之间共享数据

您还可以使用临时表格在语言解释器之间共享数据。以下示例通过单元格的 %%pyspark 用 Python 创建临时表格，并通过下列单元格的 %%scalaspark 从 Scala 该表中读取数据。

```
%%pyspark
df=spark.sql("SELECT * from nyc_top_trips_report LIMIT 20")
```

```
# create a temporary table called nyc_top_trips_report_view in python
df.createOrReplaceTempView("nyc_top_trips_report_view")
```

```
%%scalaspark
// read the temp table in scala
val df=spark.sql("SELECT * from nyc_top_trips_report_view")
df.show(5)
```

Amazon EMR Notebooks 概述

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

您可以将 Amazon EMR Notebooks 与运行 Apache Spark 的 Amazon EMR 集群结合使用，以在 Amazon EMR 控制台中创建和打开 Jupyter notebook和 JupyterLab 界面。EMR 笔记本是一款“无服务器”笔记本，可用于运行查询和代码。与传统笔记本不同，EMR 笔记本的内容（笔记本单元格中的方程、查询、模型、代码和叙述性文本）在客户端中运行。命令是使用 EMR 集群上的内核执行的。笔记本内容也将独立于集群数据保存到 Amazon S3 中以实现持久性和灵活重用。

您可以启动集群，附加 EMR 笔记本以进行分析，然后终止集群。您还可以关闭已附加到一个正在运行的集群的笔记本并切换到另一个笔记本。多个用户可以同时将笔记本附加到同一个集群，并在 Amazon S3 中相互共享笔记本文件。利用这些功能，您可以按需运行集群以节省成本，并减少为不同的集群和数据集重新配置笔记本所花费的时间。

您还可以使用 Amazon EMR API 以编程方式执行 EMR 笔记本，而无需与 Amazon EMR 控制台交互（“无头执行”）。您需要在具有参数标签的 EMR 笔记本中包含一个单元格。该单元格允许脚本将新输入值传递给笔记本。参数化笔记本可以通过不同的输入值集重复使用。无需复制同一笔记本以使用新输入值进行编辑和执行。针对参数化笔记本的每次运行，Amazon EMR 都在 S3 上创建并保存输出笔记本。有关 EMR 笔记本 API 代码示例，请参阅以编程方式执行 EMR Notebooks 的示例命令。

> **Important**
>
> EMR Notebooks 的功能支持使用 Amazon EMR 5.18.0 及更高版本的集群。建议您将 EMR Notebooks 与使用最新版 Amazon EMR（或最低 5.30.0、5.32.0 或 6.2.0 版本）的集群配合使用。在这些版本中，Jupyter 内核在附加的集群而不是 Jupyter 实例上运行。这有助于提高性能并增强您自定义内核和库的能力。有关更多信息，请参阅集群发行版的功能差异。

对于 Amazon S3 存储和 Amazon EMR 集群收取适用的费用。

Amazon EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供

从 EMR Notebooks 过渡到 Workspaces

在 Amazon EMR 新控制台中，我们已将 EMR Notebooks 与 Amazon EMR Studio Workspaces 合并为一种体验。使用 EMR Studio 后，您可以创建和配置不同的 Workspaces 来组织和运行笔记本。如果旧控制台中有 Amazon EMR Notebooks，则会在新控制台中作为 EMR Studio Workspaces 提供。

Amazon EMR 为您创建了这些新的 EMR Studio Workspaces。我们创建的 Studio 数量对应于您从 EMR Notebooks 使用的不同 VPC 的数量。例如，如果您从 EMR Notebooks 连接到两个不同 VPC 中的 EMR 集群，则我们会创建两个新的 EMR Studio。您的笔记本将在新的 Studio 中分发。

> **Important**
>
> 我们关闭了在 Amazon EMR 旧控制台中创建新笔记本的选项。改用 Amazon EMR 新控制台中的 Create Workspace（创建 Workspace）。

有关 Amazon EMR Studio Workspaces 的更多信息，请参阅 学习 WorkSpace 基础知识。有关 EMR Studio 的概念性概述，请参阅 Amazon EMR Studio 工作原理 页面上的 Workspaces。

您需要做什么？

虽然您仍然可以在旧控制台中使用现有的笔记本，但建议您在新控制台中改用 Amazon EMR Studio Workspaces。您必须配置其他角色权限才能开启在 EMR Notebooks 中不可用的 EMR Studio 中的功能。

> **Note**
>
> 至少，要将现有 EMR Notebooks 视为 EMR Studio Workspaces 并创建新的 Workspaces，用户必须对其角色具有 `elasticmapreduce:ListStudios` 和 `elasticmapreduce:CreateStudioPresignedUrl` 权限。要访问所有 EMR Studio 功能，请参阅 为 EMR Notebooks 用户启用 EMR Studio 功能 以获得 EMR Notebooks 用户所需的增加权限的完整列表。

EMR Studio 中超越 EMR Notebooks 的增强功能

借助 Amazon EMR Studio，您可以设置和使用 EMR Notebooks 不具备的以下功能：

- 从 Jupyterlab 中浏览并附加到 EMR 集群
- 从 Jupyterlab 中浏览并附加到 EMR Notebooks 虚拟集群
- 从 Jupyterlab 中连接到 Git 存储库
- 与团队的其他成员合作编写和运行笔记本代码
- 用 SQL Explorer 浏览数据
- 使用 Service Catalog 预置 EMR 集群

有关 Amazon EMR Studio 功能的完整列表，请参阅 EMR Studio 主要功能。

为 EMR Notebooks 用户启用 EMR Studio 功能

作为此合并的一部分，我们将创建的新 EMR Studios，使用现有的 `EMR_Notebooks_DefaultRole` IAM 角色作为 EMR Studio 服务角色。

从 EMR Notebooks 过渡到 EMR Studio 并希望使用 EMR Studio 其他功能的用户需要几个新的角色权限。为计划使用 EMR Studio 的 EMR Notebooks 用户的角色添加以下权限。

> **Note**
>
> 至少，要将现有 EMR Notebooks 视为 EMR Studio Workspaces 并创建新的 Workspaces，用户必须对其角色具有 `elasticmapreduce:ListStudios` 和 `elasticmapreduce:CreateStudioPresignedUrl` 权限。要使用所有 EMR Studio 功能，请添加下面列出的所有权限。管理员用户还需要创建和管理 EMR Studio 的权限。有关更多信息，请参阅创建和管理 EMR Studio 所需的管理员权限。

```
"elasticmapreduce:DescribeStudio",
"elasticmapreduce:ListStudios",
"elasticmapreduce:CreateStudioPresignedUrl",
"elasticmapreduce:UpdateEditor",
"elasticmapreduce:PutWorkspaceAccess",
"elasticmapreduce:DeleteWorkspaceAccess",
"elasticmapreduce:ListWorkspaceAccessIdentities",
```

```
"emr-containers:ListVirtualClusters",
"emr-containers:DescribeVirtualCluster",
"emr-containers:ListManagedEndpoints",
"emr-containers:DescribeManagedEndpoint",
"emr-containers:CreateAccessTokenForManagedEndpoint",
"emr-containers:ListJobRuns",
"emr-containers:DescribeJobRun",
"servicecatalog:SearchProducts",
"servicecatalog:DescribeProduct",
"servicecatalog:DescribeProductView",
"servicecatalog:DescribeProvisioningParameters",
"servicecatalog:ProvisionProduct",
"servicecatalog:UpdateProvisionedProduct",
"servicecatalog:ListProvisioningArtifacts",
"servicecatalog:DescribeRecord",
"servicecatalog:ListLaunchPaths",
"cloudformation:DescribeStackResources"
```

使用 EMR Studio 中的协作功能还需要以下权限，但 EMR Notebooks 不需要这些权限。

```
"sso-directory:SearchUsers",
"iam:GetUser",
"iam:GetRole",
"iam:ListUsers",
"iam:ListRoles",
"sso:GetManagedApplicationInstance"
```

使用 EMR Notebooks 时的注意事项

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

在使用 EMR 笔记本创建集群和开发解决方案时，请考虑以下要求。

集群要求

- 启用 Amazon EMR 阻止公有访问 – 通过对集群进行入站访问，集群用户能够执行笔记本内核。确保只有授权用户才能访问集群。我们强烈建议您将阻止公有访问保持启用状态，并将入站 SSH 流量仅限于可靠来源。有关更多信息，请参阅使用 Amazon EMR 阻止公有访问和使用安全组控制网络流量。
- 使用兼容的集群 – 附加到笔记本的集群必须满足以下要求：
  - 仅支持使用 Amazon EMR 创建的集群。您可以在 Amazon EMR 中单独创建集群，然后附加 EMR 笔记本，也可以在创建 EMR 笔记本时创建可兼容的集群。
  - 仅支持使用 Amazon EMR 发行版 5.18.0 及更高版本创建的集群。请参阅the section called “集群发行版的功能差异”。
  - 不支持使用具有 AMD EPYC 处理器的 Amazon EC2 实例创建的集群（例如 m5a.* 和 r5a.* 实例类型）。
  - EMR Notebooks 仅适用于在 `VisibleToAllUsers` 设置为 `true` 的情况下创建的集群。默认情况下 `VisibleToAllUsers` 为 `true`。
  - 必须在 EC2-VPC 内启动集群。支持公有和私有子网。不支持 EC2-Classic 平台。
  - 启动的集群必须安装有 Hadoop、Spark 和 Livy。可以安装其它应用程序，但 EMR Notebooks 目前仅支持 Spark 集群。

    > **Important**
    >
    > 对于 Amazon EMR 发行版本 5.32.0 及更高版本或 6.2.0 及更高版本，您的集群还必须运行 Jupyter Enterprise Gateway 应用程序才能使用 EMR Notebooks。

  - 不支持使用 Kerberos 身份验证的集群。
  - 与 Amazon Lake Formation 集成的集群仅支持安装笔记本范围的库。不支持在集群上安装内核和库。
  - 不支持具有多个主节点的集群。
  - 不支持使用基于 Amazon Graviton2 的 Amazon EC2 实例的集群

集群发行版的功能差异

我们强烈建议您将 EMR Notebooks 与使用 Amazon EMR 发行版本 5.30.0、5.32.0 或更高版本或 6.2.0 或更高版本创建的集群一起使用。借助这些版本，EMR Notebooks 在附加的 Amazon EMR 集群

上运行内核。内核和库可直接安装到集群主节点上。将 EMR Notebooks 与这些集群版本结合使用具有以下好处：

- 提高了性能 – 笔记本内核在具有所选 EC2 实例类型的集群上运行。早期版本在无法调整大小、访问或自定义的专用实例上运行内核。
- 能够添加和自定义内核 – 您可以使用 conda 和 pip 连接到集群来安装内核包。此外，支持在笔记本单元中使用终端命令来安装 pip。在早期版本中，仅预安装的内核可用（Python、PySpark、Spark 和 SparkR）。有关更多信息，请参阅在集群主节点上安装内核和 Python 库。
- 能够安装 Python 库 - 您可以使用 conda 和 pip 在集群主节点上安装 Python 库。我们建议使用 conda。对于早期版本，仅支持适用于 PySpark 的笔记本范围的库。

集群版本支持的 EMR Notebooks 功能

集群发布版本	适用于 PySpark 的笔记本范围的库	集群上的内核安装	主节点上的 Python 库安装
版本 5.18.0 之前的版本		不支持 EMR Notebooks	
5.18.0–5.25.0	否	否	否
5.26.0–5.29.0	是	否	否
5.30.0	是	是	是
6.0.0	否	否	否
5.32.0 及更高版本，以及 6.2.0 及更高版本	是	是	是

并发附加的 EMR Notebooks 的限制

创建支持笔记本的集群时，请考虑集群主节点的 EC2 实例类型。此 EC2 实例的内存限制决定了可以同时准备好在集群上运行代码和查询的笔记本的数量。

主节点 EC2 实例类型	EMR Notebooks 数量
*.medium	2

主节点 EC2 实例类型	EMR Notebooks 数量
*.large	4
*.xlarge	8
*.2xlarge	16
*.4xlarge	24
*.8xlarge	24
*.16xlarge	24

Jupyter notebook 和 Python 版本

EMR Notebooks 运行 Jupyter notebook 版本 6.0.2 和 Python 3.6.5，而不管附加的集群的 Amazon EMR 发行版本如何。

安全注意事项

使用加密的 S3 位置

如果在 Amazon S3 中指定一个加密位置来存储笔记本文件，则必须将 EMR Notebooks 的服务角色 设置为密钥用户。默认服务角色为 `EMR_Notebooks_DefaultRole`。如果您使用 Amazon KMS 密钥进行加密，请参阅《Amazon Key Management Service 开发人员指南》中的在 Amazon KMS 中使用密钥策略和添加密钥用户的支持文章。

创建笔记本

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio

Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

您可以使用旧 Amazon EMR 控制台创建 EMR 笔记本。不支持使用 Amazon CLI 或 Amazon EMR API 创建笔记本。

创建 EMR Notebooks

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/。
2. 依次选择 Notebooks (笔记本)、Create notebook (创建笔记本)。
3. 在 Notebook name (笔记本名称) 中输入一个名称，在 Notebook description (笔记本描述)（可选）中输入相关描述。
4. 如果您有一个要向其附加笔记本的活动集群，请将默认值 Choose an existing cluster (选择现有集群) 保持选中状态，单击 Choose (选择)，从列表中选择集群，然后单击 Choose cluster (选择集群)。有关 EMR Notebooks 的集群要求的信息，请参阅使用 EMR Notebooks 时的注意事项。

   —或者—

   选择 Create a cluster (创建集群)，输入 Cluster name (集群名称)，然后根据以下准则选择选项。使用按需型实例在账户的默认 VPC 中创建集群。

设置	描述
集群名称	用于标识集群的友好名称。
Release	无法修改。默认为最新的 Amazon EMR 发行版本（5.36.1）。
应用程序	无法修改。列出集群上安装的应用程序。
实例	输入实例数并选择 EC2 实例类型。一个实例用于主节点。其余实例用于核心节点。实例类型决定了可同时附加到集群的笔记本的数量。有关更多信息，请参阅并发附加的 EMR Notebooks 的限制。

设置	描述
EMR 角色	保留默认值或选择用于为 Amazon EMR 指定自定义服务角色的链接。有关更多信息，请参阅Amazon EMR 的服务角色（EMR 角色）。
EC2 实例配置文件	保留默认值或选择用于为 EC2 实例指定自定义服务角色的链接。有关更多信息，请参阅集群 EC2 实例（EC2 实例配置文件）的服务角色。
EC2 密钥对	选择可用于连接到集群实例的 EC2 密钥对。有关更多信息，请参阅使用 SSH 连接到主节点。

5. 对于 Security groups (安全组)，选择 Use default security groups (使用默认安全组)。或者，选择 Choose security groups (选择安全组)，然后选择集群的 VPC 中可用的自定义安全组。您应为主实例选择一个安全组，为笔记本客户端实例选择另一个安全组。有关更多信息，请参阅the section called “EMR Notebooks 的安全组”。
6. 对于 Amazon Service Role (亚马逊云科技服务角色)，保留默认设置或从列表中选择一个自定义角色。笔记本的客户端实例将使用此角色。有关更多信息，请参阅EMR Notebooks 的服务角色。
7. 对于 Notebook location (笔记本位置)，选择笔记本文件在 Amazon S3 中的保存位置，或指定自己的位置。如果存储桶和文件夹不存在，Amazon EMR 会进行创建。

   Amazon EMR 会创建一个文件夹名称为笔记本 ID 的文件夹，并将此笔记本保存到名为 *NotebookName*.ipynb 的文件中。例如，如果为名为 MyFirstEMRManagedNotebook 的笔记本指定 Amazon S3 位置 s3://MyBucket/MyNotebooks，则该笔记本文件将保存到 s3://MyBucket/MyNotebooks/*NotebookID*/MyFirstEMRManagedNotebook.ipynb 下。

   如果在 Amazon S3 中指定一个加密位置，则必须将 EMR Notebooks 的服务角色 设置为密钥用户。默认服务角色为 EMR_Notebooks_DefaultRole。如果您使用 Amazon KMS 密钥进行加密，请参阅《Amazon Key Management Service 开发人员指南》中的在 Amazon KMS 中使用密钥策略和添加密钥用户的支持文章。
8. （可选）如果您已将基于 Git 的存储库添加到要与此笔记本关联的 Amazon EMR，请选择 Git repository (Git 存储库)，单击 Choose repository (选择存储库)，然后从列表中选择一个存储库。有关更多信息，请参阅将基于 Git 的存储库与 EMR Notebooks 关联。
9. （可选）选择 Tags (标签)，然后为该笔记本添加其它键-值标签。

> **Important**
>
> 其中 Key (键) 字符串设置为 `creatorUserID`，值设置为您的 IAM 用户 ID 的默认标签用于访问目的。我们建议您不要更改或删除此标签，因为它可用于控制访问权限。有关更多信息，请参阅将集群和笔记本标签与访问控制 IAM policy 结合使用。

10. 选择 Create Notebook (创建笔记本)。

使用 EMR Notebooks

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

创建 EMR 笔记本后，在很短的时间内便可启动该笔记本。Notebooks (笔记本) 列表中的 Status (状态) 显示为 Starting (正在启动)。笔记本的状态为 Ready (就绪) 时，便可将其打开。如果您随笔记本一同创建了集群，则笔记本变为 Ready (就绪) 状态可能需要较长的时间。

> **Tip**
>
> 刷新浏览器或选择笔记本列表上方的刷新图标来刷新笔记本状态。

了解笔记本状态

EMR 笔记本可以在 Notebooks (笔记本) 列表中具有以下 Status (状态)。

状态	含义
Ready (就绪)	您可以使用笔记本编辑器打开笔记本。笔记本的状态为 Ready (就绪) 时，您可以将其停止或删

状态	含义
	除。要更改集群，您必须先停止笔记本。如果处于 Ready (就绪) 状态的笔记本长时间空闲，它将自动停止。
Starting (正在启动)	正在创建笔记本并将其附加到集群。笔记本处于“Starting (正在启动)”状态时，您无法打开笔记本编辑器，也无法停止、删除它或更改集群。
Pending (待处理)	已创建笔记本，正在等待与待完成的集群集成。集群可能仍在预配置资源，也可能在响应其它请求。您可以在本地模式中使用笔记本打开笔记本编辑器。依赖于集群过程的任何代码均不会执行和失败。
Stopping (正在停止)	正在关闭笔记本或正在终止该笔记本附加到的集群。笔记本处于“Stopping (正在停止)”状态时，您无法打开笔记本编辑器，也无法停止、删除它或更改集群。
Stopped (已停止)	笔记本已关闭。只要集群仍在运行，您就可以在同一个集群上启动笔记本。您可以更改集群，也可以删除该集群。
Deleting (正在删除)	正在从可用集群列表中删除该集群。笔记本文件*NotebookName* `.ipynb` 会保留在 Amazon S3 中，并继续产生适用的存储费用。

使用笔记本编辑器

使用 EMR 笔记本的一项优势是，您可以直接从控制台启动 Jupyter 或 JupyterLab 中的笔记本。

对于 EMR Notebooks，从 Amazon EMR 控制台访问的笔记本编辑器就是您熟悉的开源 Jupyter notebook 编辑器或 JupyterLab。由于笔记本编辑器是在 Amazon EMR 控制台内启动的，与使用 Amazon EMR 集群上托管的笔记本相比，配置访问权限更高效。您无需将用户客户端配置为通过 SSH、安全组规则以及代理配置进行 Web 访问。如果用户具有足够的权限，他们只需在 Amazon EMR 控制台内打开笔记本编辑器即可。

一次只能有一个用户从 Amazon EMR 中打开 EMR 笔记本。如果其它用户尝试打开已打开的 EMR 笔记本，则会发生错误。

> **Important**
>
> Amazon EMR 为每个笔记本编辑器会话创建一个唯一的预签名 URL，该 URL 仅在短时间内有效。我们建议您不要共享笔记本编辑器 URL。这样做会产生安全风险，因为 URL 的收件人会采用您的权限来编辑笔记本并在 URL 的生命周期内运行笔记本代码。如果其他人需要访问笔记本，请通过权限策略向其用户提供权限，并确保 EMR Notebooks 的服务角色有权访问 Amazon S3 位置。有关更多信息，请参阅the section called “安全性”和EMR Notebooks 的服务角色。

为 EMR 笔记本打开笔记本编辑器

1. 从 Notebooks（笔记本）列表中选择 Status（状态）为 Ready（就绪）或 Pending（待处理）的笔记本。
2. 选择 Open in JupyterLab (在 JupyterLab 中打开) 或 Open in Jupyter (在 Jupyter 中打开)。

   JupyterLab 或 Jupyter notebook 编辑器将在一个新的浏览器选项卡中打开。
3. 从 Kernel (内核) 菜单中，选择 Change kernel (选择内核)，然后选择与您的编程语言相对应的内核。

   现在，您已准备就绪，可以从笔记本编辑器编写和运行代码了。

保存笔记本内容

在笔记本编辑器中执行操作时，笔记本单元内容和输出在 Amazon S3 中会自动定期保存到笔记本文件中。自上次编辑单元以来未进行任何更改的笔记本在编辑器的笔记本名称旁会显示 (autosaved) ([自动保存])。如果更改尚未保存，系统将显示 unsaved changes (未保存的更改)。

您可以手动保存笔记本。从 File（文件）菜单中，选择 Save and Checkpoint（保存和检查点）或按 CTRL+S。这会在 Amazon S3 的笔记本文件夹中 checkpoints 文件夹下创建一个名为 *NotebookName*.ipynb 的文件。例如，s3://*MyBucket*/*MyNotebookFolder*/*NotebookID*/checkpoints/*NotebookName*.ipynb。该位置只保存最新的检查点文件。

更改集群

您可以更改已附加 EMR 笔记本的集群，无需更改笔记本本身的内容。您可以只针对状态为 Stopped (已停止) 的笔记本更改集群。

更改 EMR 笔记本的集群

1. 如果要更改的笔记本正在运行中，请从 Notebooks (笔记本) 列表中选择该笔记本，然后选择 Stop (停止)。
2. 如果笔记本状态为 Stopped (已停止)，请从 Notebooks (笔记本) 列表中选择该笔记本，然后选择 View details (查看详细信息)。
3. 选择 Change cluster (更改集群)。
4. 如果您有一个正在运行 Hadoop、Spark 和 Livy 且要向其附加笔记本的活动集群，保留其默认设置，然后从列表中选择一个集群。列表中只列出符合要求的集群。

   —或者—

   选择 Create a cluster (创建集群)，然后选择集群选项。有关更多信息，请参阅集群要求。
5. 为 Security groups (安全组) 选择一个选项，然后选择 Change cluster and start notebook (更改集群并启动笔记本)。

删除笔记本和笔记本文件

使用 Amazon EMR 控制台删除 EMR 笔记本时，您将从可用笔记本列表中删除该笔记本。但是，笔记本文件会保留在 Amazon S3 中，并继续产生存储费用。

删除笔记本并删除关联文件

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/。
2. 选择 Notebooks (笔记本)，从列表中选择您的笔记本，然后选择 View details (查看详细信息)。
3. 选择 Notebook location (笔记本位置) 旁边的文件夹图标并复制采用格式 `s3://MyNotebookLocationPath/NotebookID/` 的 URL。
4. 选择 Delete (删除)。

   系统将从列表中删除该笔记本，您无法再查看笔记本详细信息。
5. 有关说明，请参阅《Amazon Simple Storage Service 用户指南》中的如何从 S3 存储桶中删除文件夹？。从步骤 3 导航到存储桶和文件夹。

—或者—

如果您已安装 Amazon CLI，请打开命令提示符窗口并键入此段落末尾处的命令。将 Amazon S3 位置替换为您在上面复制的位置。务必使用有权限删除该 Amazon S3 位置的用户的访问密钥配置 Amazon CLI。有关更多信息，请参阅《Amazon Command Line Interface 用户指南》中的配置 Amazon CLI。

```
aws s3 rm s3://MyNotebookLocationPath/NotebookID
```

共享笔记本文件

每个 EMR 笔记本都作为名为 *NotebookName*.ipynb 的文件保存到 Amazon S3。只要笔记本文件与 EMR Notebooks 所基于的同一 Jupyter notebook 版本兼容，您就可以将该笔记本作为 EMR 笔记本进行打开。

打开来自其它用户的笔记本文件的最简单方法是，将来自其它用户的 *.ipynb 文件保存到您的本地文件系统，然后使用 Jupyter 和 Jupyterlab 编辑器中的上载功能。

您可以通过此过程使用其它人共享的 EMR 笔记本、Jupyter 社区中共享的笔记本，或还原从控制台删除的笔记本（如果您仍具有该笔记本文件）。

将不同的笔记本文件用作 EMR 笔记本的基础

1. 继续操作之前，针对要使用的所有笔记本关闭笔记本编辑器，然后停止笔记本（如果是 EMR 笔记本）。
2. 创建一个 EMR 笔记本并为其输入一个名称。您为该笔记本输入的名称将是您需要替换的文件的名称。新文件名称必须与此文件名称完全匹配。
3. 记下您为该笔记本选择的 Amazon S3 位置。您替换的文件位于路径和文件名采用以下格式的文件夹中：s3://*MyNotebookLocation*/*NotebookID*/*MyNotebookName*.ipynb。
4. 停止笔记本。
5. 使用完全相同的名称将 Amazon S3 位置中的旧笔记本文件替换为新的笔记本文件。

   以下适用于 Amazon S3 的 Amazon CLI 命令将保存在名为 SharedNotebook.ipynb 的本地计算机上的文件替换为名为 MyNotebook、ID 为 e-12A3BCDEFJHIJKLMNO45PQRST 的 EMR Notebooks，并使用 Amazon S3 中指定的 MyBucket/MyNotebooksFolder 创建。有关使用 Amazon S3 控制台复制和替换文件的信息，请参阅《Amazon Simple Storage Service 用户指南》中的上传、下载和管理对象。

```
aws s3 cp SharedNotebook.ipynb s3://MyBucket/
MyNotebooksFolder/-12A3BCDEFJHIJKLMNO45PQRST/MyNotebook.ipynb
```

以编程方式执行 EMR Notebooks 的示例命令

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

概述

您可以通过脚本或命令行中的执行 API 来执行 EMR 笔记本。在 Amazon 控制台外启动、停止、列出和描述 EMR 笔记本执行时，您可以通过编程方式控制 EMR 笔记本。您可以将不同的参数值传递给带参数化笔记本单元格的笔记本。这样就不必为每组新参数值创建笔记本副本。有关更多信息，请参阅 Amazon EMR API actions。

您可以使用 Amazon CloudWatch Events 和 Amazon Lambda 安排或批量处理 EMR 笔记本执行。有关更多信息，请参阅将 Amazon Lambda 与 Amazon CloudWatch Events 结合使用。

编程执行的角色权限

要在 EMR Notebooks 中使用编程执行，您必须使用以下策略配置用户权限：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowExecutionActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:StartNotebookExecution",
```

```
                "elasticmapreduce:DescribeNotebookExecution",
                "elasticmapreduce:ListNotebookExecutions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowPassingServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/EMR_Notebooks_DefaultRole"
        }
    ]
}
```

在 EMR Notebooks 集群上以编程方式执行 EMR Notebooks 时，您必须添加以下额外权限：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRetrievingManagedEndpointCredentials",
            "Effect": "Allow",
            "Action": [
                "emr-containers:GetManagedEndpointSessionCredentials"
            ],
            "Resource": [
                "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-
cluster-id/endpoints/managed-endpoint-id"
            ],
            "Condition": {
                "StringEquals": {
                    "emr-containers:ExecutionRoleArn": [
                        "arn:aws:iam::account-id:role/emr-on-eks-execution-role"
                    ]
                }
            }
        },
        {
            "Sid": "AllowDescribingManagedEndpoint",
            "Effect": "Allow",
            "Action": [
```

```
                "emr-containers:DescribeManagedEndpoint"
            ],
            "Resource": [
                "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-
cluster-id/endpoints/managed-endpoint-id"
            ]
        }
    ]
}
```

编程执行的限制

- 每账户每 Amazon Web Services 区域最多支持 100 个并发执行。
- 如果运行超过 30 天，则会终止执行。
- Amazon EMR Serverless 交互式应用程序不支持以编程方式执行笔记本电脑实例。

编程 EMR 笔记本执行示例

本节提供了几个使用 Amazon CLI、Boto3 SDK（Python）和 Ruby 以编程方式执行 EMR 笔记本的示例：

- 笔记本执行 CLI 命令示例
- 笔记本执行 Python 示例
- 笔记本执行 Ruby 示例

您还可以使用编排工具 [例如 Apache Airflow 或 Amazon Managed Workflows for Apache Airflow（MWAA）]，将参数化笔记本作为计划工作流的一部分运行。有关更多信息，请参阅 Amazon 大数据博客中的 Orchestrating analytics jobs on EMR Notebooks using MWAA。

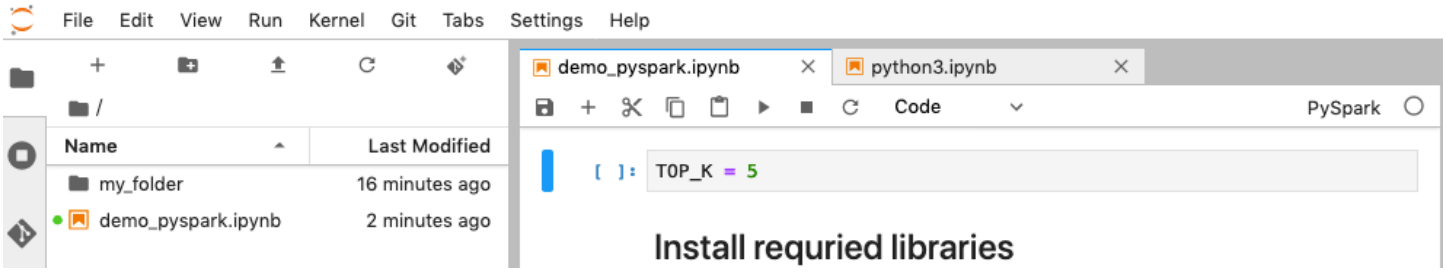
笔记本执行 CLI 命令示例

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio

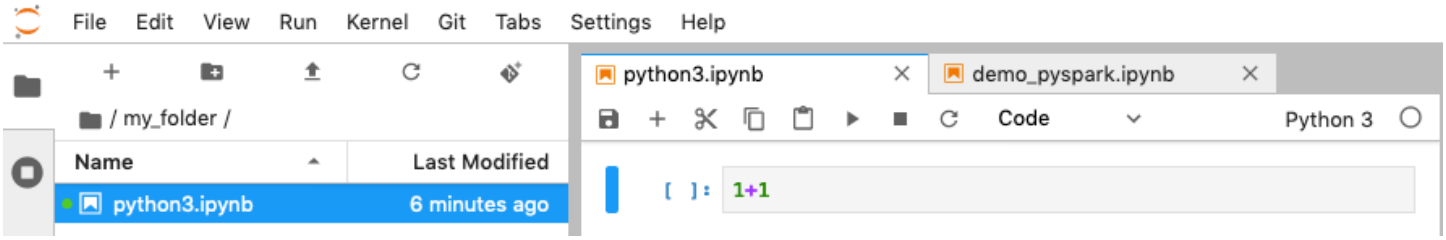
Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

以下示例使用 EMR Notebooks 控制台中的笔记本演示。要查找笔记本，请使用相对于主目录的文件路径。在此示例中，您可以运行两个笔记本文件：demo_pyspark.ipynb 和 my_folder/python3.ipynb。

文件 demo_pyspark.ipynb 的相对路径是 demo_pyspark.ipynb，如下所示。



python3.ipynb 的相对路径是 my_folder/python3.ipynb，如下所示。



有关 Amazon EMR API NotebookExecution 操作的更多信息，请参阅 Amazon EMR API actions。

运行笔记本

您可以使用 Amazon CLI 通过 start-notebook-execution 操作来运行笔记本，示例如下。

Example – 使用（运行在 Amazon EC2 上的）Amazon EMR 集群在 EMR Studio Workspace 中执行 EMR 笔记本。

```
aws emr --region us-east-1 \
start-notebook-execution \
--editor-id e-ABCDEFG123456 \
--notebook-params '{"input_param":"my-value", "good_superhero":["superman", "batman"]}'
 \
--relative-path test.ipynb \
--notebook-execution-name my-execution \
--execution-engine '{"Id" : "j-1234ABCD123"}' \
```

```
--service-role EMR_Notebooks_DefaultRole

{
    "NotebookExecutionId": "ex-ABCDEFGHIJ1234ABCD"
}
```

Example – 使用 EMR Notebooks 集群在 EMR Studio Workspace 中执行 EMR 笔记本。

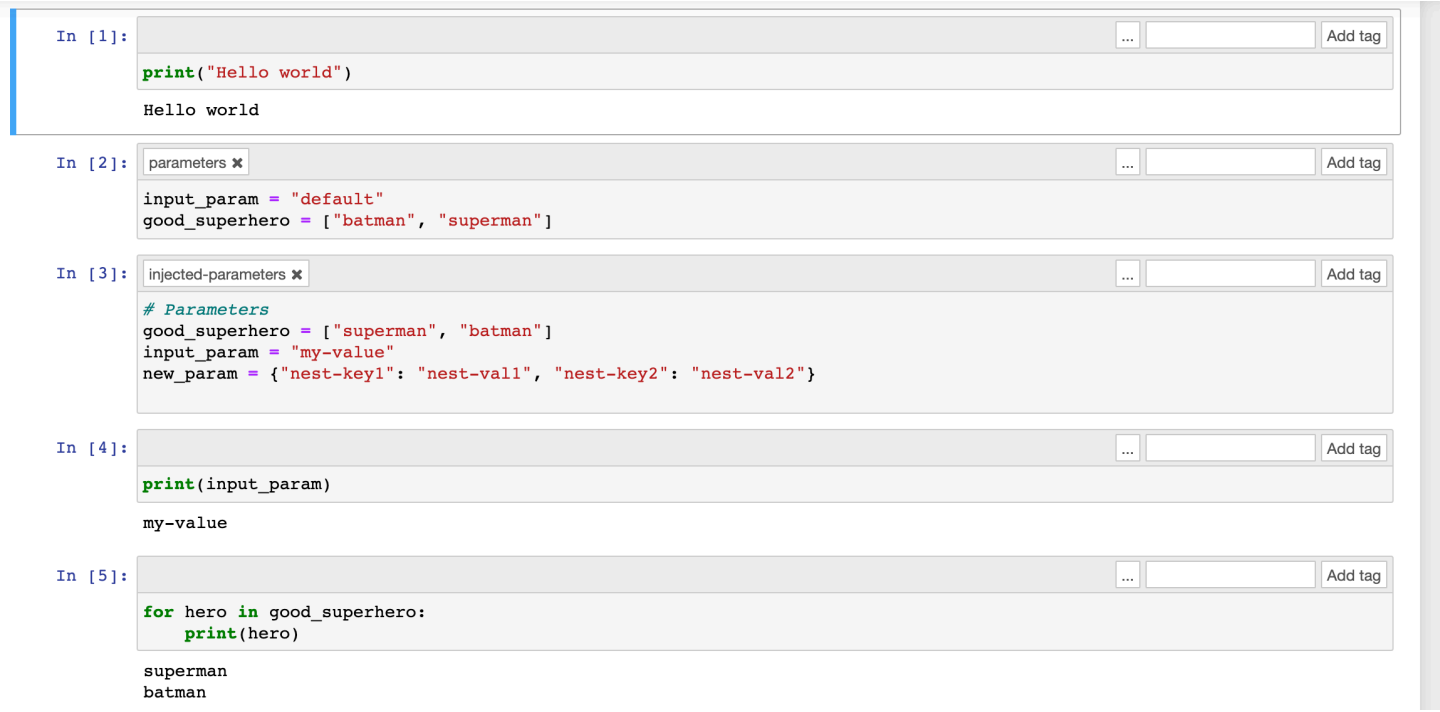
```
aws emr start-notebook-execution \
    --region us-east-1 \
    --service-role EMR_Notebooks_DefaultRole \
    --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
 spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
    --output-notebook-format HTML \
    --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEFG/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
    --editor-id e-ABCDEFG \
    --relative-path EMRonEKS-spark_python.ipynb
```

Example – 执行 EMR 笔记本来指定其 Amazon S3 位置

```
aws emr start-notebook-execution \
    --region us-east-1 \
    --notebook-execution-name my-execution-on-emr-on-eks-cluster \
    --service-role EMR_Notebooks_DefaultRole \
    --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
 spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
    --output-notebook-format HTML \
    --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEF/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
    --notebook-s3-location '{"Bucket": "your-s3-bucket","Key": "s3-prefix-to-notebook-
location/EMRonEKS-spark_python.ipynb"}' \
    --output-notebook-s3-location '{"Bucket": "your-s3-bucket","Key": "s3-prefix-for-
storing-output-notebook"}'
```

笔记本输出

下面是示例笔记本的输出内容。单元格 3 显示新注入的参数值。



描述笔记本

您可以使用 describe-notebook-execution 操作以访问关于特定笔记本执行的信息。

```
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE

{
    "NotebookExecution": {
        "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
        "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
        "ExecutionEngine": {
            "Id": "j-2QMOV6JAX1TS2",
            "Type": "EMR",
            "MasterInstanceSecurityGroupId": "sg-05ce12e58cd4f715e"
        },
        "NotebookExecutionName": "my-execution",
        "NotebookParams": "{\"input_param\":\"my-value\", \"good_superhero\":
[\"superman\", \"batman\"]}",
        "Status": "FINISHED",
        "StartTime": 1593490857.009,
        "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
        "LastStateChangeReason": "Execution is finished for cluster j-2QMOV6JAX1TS2.",
        "NotebookInstanceSecurityGroupId": "sg-0683b0a39966d4a6a",
```

```
        "Tags": []
    }
}
```

停止运行笔记本

如果您想要停止笔记本正在运行的执行，您可以借助 stop-notebook-execution 命令。

```
# stop a running execution
aws emr --region us-east-1 \
stop-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T


# describe it
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T

{
    "NotebookExecution": {
        "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
        "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
        "ExecutionEngine": {
            "Id": "j-2QMOV6JAX1TS2",
            "Type": "EMR"
        },
        "NotebookExecutionName": "my-execution",
        "NotebookParams": "{\"input_param\":\"my-value\", \"good_superhero\":
[\"superman\", \"batman\"]}",
        "Status": "STOPPED",
        "StartTime": 1593490876.241,
        "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:editor-execution/ex-
IZWZX78UVPAATC8LHJR129B1RBN4T",
        "LastStateChangeReason": "Execution is stopped for cluster j-2QMOV6JAX1TS2.
 Internal error",
        "Tags": []
    }
}
```

按开启时间列出笔记本的执行情况

您可以将 --from 参数输入至 list-notebook-executions，按开启时间列出笔记本的执行情况。

```
# filter by start time
```

```
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000

{
    "NotebookExecutions": [
        {
            "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "STOPPED",
            "StartTime": 1593490876.241
        },
        {
            "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "RUNNING",
            "StartTime": 1593490857.009
        },
        {
            "NotebookExecutionId": "ex-IZWZYRS0M14L5V95WZ9OQ399SKMNW",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "STOPPED",
            "StartTime": 1593490292.995
        },
        {
            "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593489834.765
        },
        {
            "NotebookExecutionId": "ex-IZWZXOZF88JWDF9J09GJ91R57VI0N",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FAILED",
            "StartTime": 1593488934.688
        }
    ]
}
```

按开启时间和状态列出笔记本的执行情况

`list-notebook-executions` 命令也可以采用 `--status` 参数来筛选结果。

```
# filter by start time and status
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000 --status FINISHED
{
    "NotebookExecutions": [
        {
            "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593490857.009
        },
        {
            "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593489834.765
        }
    ]
}
```

笔记本执行 Python 示例

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

以下代码示例是一个名为 `demo.py` 的 SDK for Python（Boto3）文件，用于显示笔记本执行 API。

有关 Amazon EMR API `NotebookExecution` 操作的更多信息，请参阅 Amazon EMR API actions。

```
import boto3,time

emr = boto3.client(
    'emr',
    region_name='us-west-1'
)

start_resp = emr.start_notebook_execution(
    EditorId='e-40AC8ZO6EGGCPJ4DLO48KGGGI',
    RelativePath='boto3_demo.ipynb',
    ExecutionEngine={'Id':'j-1HYZS6JQKV11Q'},
    ServiceRole='EMR_Notebooks_DefaultRole'
)

execution_id = start_resp["NotebookExecutionId"]
print(execution_id)
print("\n")

describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)

print(describe_response)
print("\n")

list_response = emr.list_notebook_executions()
print("Existing notebook executions:\n")
for execution in list_response['NotebookExecutions']:
    print(execution)
    print("\n")

print("Sleeping for 5 sec...")
time.sleep(5)

print("Stop execution " + execution_id)
emr.stop_notebook_execution(NotebookExecutionId=execution_id)
describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)
print(describe_response)
print("\n")
```

下面是运行 demo.py 的输出。

```
ex-IZX56YJDW1D29Q1PHR32WABU2SAPK
```

```
{'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
 'EditorId': 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'ExecutionEngine': {'Id':
 'j-1HYZS6JQKV11Q', 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STARTING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()),
 'Arn': 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is starting
 for cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId':
 '70f12c5f-1dda-45b7-adf6-964987d373b7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-
amzn-requestid': '70f12c5f-1dda-45b7-adf6-964987d373b7', 'content-type': 'application/
x-amz-json-1.1', 'content-length': '448', 'date': 'Wed, 19 Aug 2020 00:49:22 GMT'},
 'RetryAttempts': 0}}

Existing notebook executions:

{'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'STARTING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5ABS5PR1E5AHMFYEMX3JJIORRB', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'RUNNING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 48, 36, 373000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5GLVXIU1HNI8BWVW057F6MF4VE', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 45, 14, 646000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 46, 26, 543000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5CV8YDUO8JAIWMXN2VH32RUIT1', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 43, 5, 807000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 44, 31, 632000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5AS0PPW55CEDEURZ9NSOWSUJZ6', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 42, 29, 265000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 43, 48, 320000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX57YF5Q53BKWLR4I5QZ14HJ7DRS', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
```

```
 'StartTime': datetime.datetime(2020, 8, 19, 0, 38, 37, 81000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 40, 39, 646000, tzinfo=tzlocal())}

Sleeping for 5 sec...
Stop execution ex-IZX56YJDW1D29Q1PHR32WABU2SAPK
{'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
 'EditorId': 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'ExecutionEngine': {'Id':
 'j-1HYZS6JQKV11Q', 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STOPPING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()),
 'Arn': 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is being stopped
 for cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId':
 '2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-
amzn-requestid': '2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'content-type': 'application/
x-amz-json-1.1', 'content-length': '453', 'date': 'Wed, 19 Aug 2020 00:49:30 GMT'},
 'RetryAttempts': 0}}
```

笔记本执行 Ruby 示例

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

以下是演示使用笔记本执行 API 的 Ruby 代码示例。

```
# prepare an Amazon EMR client

emr = Aws::EMR::Client.new(
  region: 'us-east-1',
  access_key_id: 'AKIA...JKPKA',
  secret_access_key: 'rLMeu...vU0OLrAC1',
)
```

启动笔记本执行并获取执行 ID

在此示例中，Amazon S3 编辑器和 EMR 笔记本是 `s3://mybucket/notebooks/e-EA8VGAA429FEQTC8HC9ZHWISK/test.ipynb`。

有关 Amazon EMR API `NotebookExecution` 操作的更多信息，请参阅 Amazon EMR API actions。

```
start_response = emr.start_notebook_execution({
    editor_id: "e-EA8VGAA429FEQTC8HC9ZHWISK",
    relative_path: "test.ipynb",

    execution_engine: {id: "j-3U82I95AMALGE"},

    service_role: "EMR_Notebooks_DefaultRole",
})


notebook_execution_id = start_resp.notebook_execution_id
```

描述笔记本执行并打印详细信息

```
describe_resp = emr.describe_notebook_execution({
    notebook_execution_id: notebook_execution_id
})
puts describe_resp.notebook_execution
```

上述命令的输出如下。

```
{
:notebook_execution_id=>"ex-IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:editor_id=>"e-EA8VGAA429FEQTC8HC9ZHWISK",
:execution_engine=>{:id=>"j-3U82I95AMALGE", :type=>"EMR", :master_instance_security_group_id=>n
:notebook_execution_name=>"",
:notebook_params=>nil,
:status=>"STARTING",
:start_time=>2020-07-23 15:07:07 -0700,
:end_time=>nil,
:arn=>"arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:output_notebook_uri=>nil,
:last_state_change_reason=>"Execution is starting for cluster
 j-3U82I95AMALGE.", :notebook_instance_security_group_id=>nil,
```

```
:tags=>[]
}
```

笔记本筛选条件

```
"EditorId": "e-XXXX",           [Optional]
"From" : "1593400000.000",    [Optional]
"To" :
```

停止笔记本执行

```
stop_resp = emr.stop_notebook_execution({
    notebook_execution_id: notebook_execution_id
})
```

启用用户模拟以监控 Spark 用户和任务活动

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

借助 EMR Notebooks，您可以在 Spark 集群上配置用户模拟。此功能可帮助您跟踪从笔记本编辑器内启动的作业活动。此外，EMR Notebooks 有一个内置的 Jupyter notebook 小组件，可用于在笔记本编辑器的查询输出旁边显示 Spark 任务详细信息。默认情况下，此小部件处于可用状态，无需特殊配置。但是，要查看历史记录服务器，您的客户端必须配置为查看主节点上托管的 Amazon EMR Web 界面。

设置 Spark 用户模拟

默认情况下，用户使用笔记本编辑器提交的 Spark 作业似乎源自一个模糊的 `livy` 用户身份。您可以为集群配置用户模拟，以便这些任务与运行代码的用户身份相关联。系统会为在笔记本中运行代码的每

个用户身份在主节点上创建 HDFS 用户目录。例如，如果用户 NbUser1 从笔记本编辑器运行代码，您可以连接主节点并可以看到 hadoop fs -ls /user 显示目录 /user/user_NbUser1。

您可以在 core-site 和 livy-conf 配置分类中设置属性以启用该功能。如果您让 Amazon EMR 同时创建集群和笔记本，那么默认情况下，此功能将不可用。有关使用配置分类自定义应用程序的更多信息，请参阅《Amazon EMR 版本指南》中的配置应用程序。

使用以下配置分类和值为 EMR Notebooks 启用用户模拟：

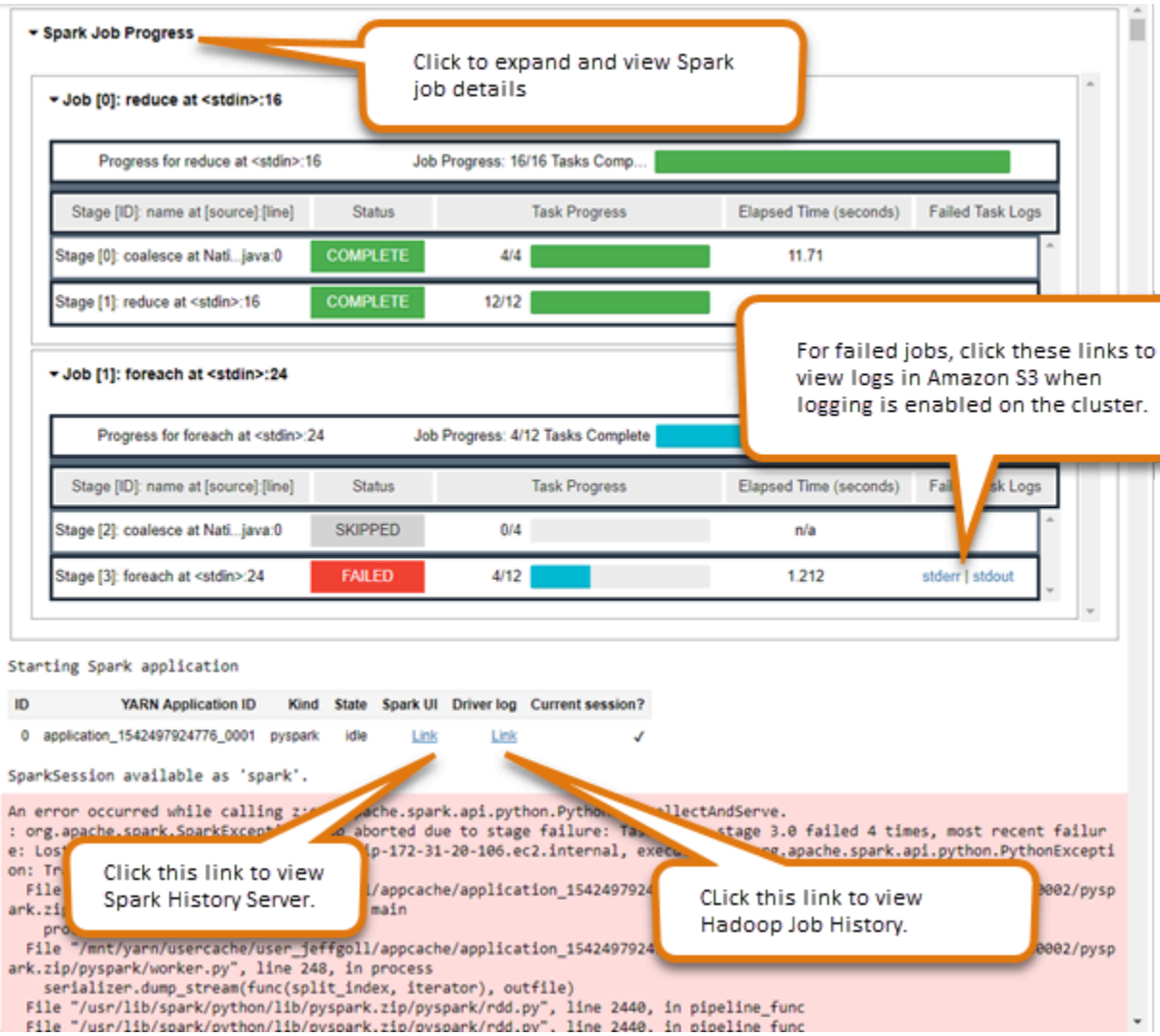
```
[
    {
        "Classification": "core-site",
        "Properties": {
          "hadoop.proxyuser.livy.groups": "*",
          "hadoop.proxyuser.livy.hosts": "*"
        }
    },
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.impersonation.enabled": "true"
        }
    }
]
```

使用 Spark 任务监控小组件

如果运行代码的笔记本编辑器在 EMR 集群上执行 Spark 任务，那么输出中会包含一个用于监控 Spark 任务的 Jupyter notebook 小组件。此小组件会提供任务详细信息、指向 Spark 历史记录服务器页面和 Hadoop 任务历史记录页面的有用链接，以及指向 Amazon S3 中有关任何失败任务的任务日志的便捷链接。

要查看集群主节点上的历史记录服务器页面，您必须根据需要设置 SSH 客户端和代理。有关更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。要查看 Amazon S3 中的日志，必须启用集群日志记录，这是新集群的默认设置。有关更多信息，请参阅查看归档到 Amazon S3 的日志文件。

以下是一个 Spark 任务监控示例。



EMR 笔记本安全性和访问控制

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

有许多功能可帮助您定制 EMR Notebooks 的安全状况。这有助于确保只有获得授权的用户才有权访问和使用 EMR 笔记本，并使用笔记本编辑器在集群上执行代码。这些功能与适用于 Amazon EMR 和 Amazon EMR 集群的安全功能一起使用。有关更多信息，请参阅Amazon EMR 中的安全性。

- 您可以将 Amazon Identity and Access Management 策略语句与笔记本标签一起使用来限制访问权限。有关更多信息，请参阅Amazon EMR 如何与 IAM 配合使用和适用于 EMR Notebooks 的基于身份的策略语句示例。
- Amazon EC2 安全组充当虚拟防火墙，可控制集群主实例与笔记本编辑器之间的网络流量。您可以使用默认设置，也可以自定义这些安全组。有关更多信息，请参阅为 EMR Notebooks 指定 EC2 安全组。
- 指定一个Amazon服务角色来确定 EMR 笔记本与其它Amazon服务交互时所拥有的权限。有关更多信息，请参阅EMR Notebooks 的服务角色。

安装并使用内核和库

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

每个 EMR 笔记本均附带了一组预安装的库和内核。如果 Amazon EMR 集群有权访问内核和库所在的存储库，则您可以在 EMR 集群中安装其他库和内核。例如，对于私有子网中的集群，您可能需要配置网络地址转换 (NAT)，并为集群提供路径来访问公有 PyPI 存储库以安装库。有关为不同网络配置配置外部访问的更多信息，请参阅《Amazon VPC 用户指南》中的场景和示例。

EMR Serverless 应用程序预装了以下适用于 Python 和 PySpark 的库：

- Python 库 – ggplot、matplotlib、numpy、pandas、plotly、bokeh、scikit-learn、scipy、scipy
- PySpark 库 – ggplot、matplotlib、numpy、pandas、plotly、bokeh、scikit-learn、scipy、scipy

在集群主节点上安装内核和 Python 库

对于 Amazon EMR 发行版 5.30.0 及更高版本（不包括 6.0.0），您可以在集群的主节点上安装其他 Python 库和内核。安装后，这些内核和库可供运行已附加到集群的 EMR 笔记本的任何用户使用。通过此方式安装的 Python 库仅适用于主节点上运行的进程。这些库不会安装到核心节点或任务节点上，也不适用于这些节点上运行的执行程序。

> **Note**
>
> 对于 Amazon EMR 版本 5.30.1、5.31.0 和 6.1.0，您必须采取额外的步骤才能在集群的主节点上安装内核和库。
> 要启动此功能，请执行以下操作：
>
> 1. 确保附加到 EMR Notebooks 服务角色的权限策略允许执行以下操作：
>
>    ```
>    elasticmapreduce:ListSteps
>    ```
>
>   有关更多信息，请参阅 EMR Notebooks 的服务角色。
> 2. 使用 Amazon CLI 在集群上运行一个设置 EMR Notebooks 的步骤，如以下示例所示。您必须使用步骤名称 EMRNotebooksSetup。将 *us-east-1* 替换为您的集群所在的区域。有关更多信息，请参阅使用 Amazon CLI 向集群中添加步骤。
>
>    ```
>    aws emr add-steps --cluster-id MyClusterID --steps
>     Type=CUSTOM_JAR,Name=EMRNotebooksSetup,ActionOnFailure=CONTINUE,Jar=s3://us-
>    east-1.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://
>    awssupportdatasvcs.com/bootstrap-actions/EMRNotebooksSetup/emr-notebooks-
>    setup.sh"]
>    ```

您可以通过使用 pip 或 conda 在主节点上的 /emr/notebook-env/bin 目录中安装内核和库。

Example – 安装 Python 库

在 Python3 内核中运行 %pip magic，以作为在笔记本单元中安装 Python 库的命令。

```
%pip install pmdarima
```

您可能需要重启内核才能使用更新后的软件包。您也可以使用 %%sh Spark magic 命令来调用 pip。

```
%%sh
```

```
/emr/notebook-env/bin/pip install -U matplotlib
/emr/notebook-env/bin/pip install -U pmdarima
```

使用 PySpark 内核时，您可以使用 pip 命令在集群上安装库，也可在 PySpark 笔记本中使用笔记本范围的库。

要从终端在集群上运行 pip 命令，请首先通过 SSH 连接到主节点，如以下命令所示。

```
sudo pip3 install -U matplotlib
sudo pip3 install -U pmdarima
```

此外，您还可以使用笔记本范围的库。借助笔记本范围的库，库安装将仅限于会话范围，并在所有 Spark 执行器上进行。有关更多信息，请参阅 使用笔记本范围的库。

如果需要在一个 PySpark 内核中封装多个 Python 库，您还可以创建一个隔离的 Python 虚拟环境。有关使用示例，请参阅 使用 Virtualenv。

要在会话中创建 Python 虚拟环境，请笔记本中第一个单元中使用 %%configure magic 命令的 Spark 属性 spark.yarn.dist.archives，如以下示例所示。

```
%%configure -f
{
   "conf": {
   "spark.yarn.appMasterEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON":"./environment/bin/python",
   "spark.yarn.dist.archives":"s3://DOC-EXAMPLE-BUCKET/prefix/
my_pyspark_venv.tar.gz#environment",
   "spark.submit.deployMode":"cluster"
   }
}
```

同样，您也可以创建一个 Spark 执行器环境。

```
%%configure -f
{
   "conf": {
   "spark.yarn.appMasterEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON":"./environment/bin/python",
   "spark.executorEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.dist.archives":"s3://DOC-EXAMPLE-BUCKET/prefix/
my_pyspark_venv.tar.gz#environment",
   "spark.submit.deployMode":"cluster"
```

```
    }
}
```

您还可以使用 conda 来安装 Python 库。您不需要 sudo 访问权限即可使用 conda。您必须使用 SSH 连接到主节点，然后从终端运行 conda。有关更多信息，请参阅使用 SSH 连接到主节点。

Example – 安装内核

以下示例演示在连接到集群的主节点时如何使用终端命令安装 Kotlin 内核：

```
sudo /emr/notebook-env/bin/conda install kotlin-jupyter-kernel -c jetbrains
```

> **Note**
>
> 这些说明不安装内核依赖项。如果您的内核具有第三方依赖项，您可能需要执行额外的设置步骤，然后才能在笔记本上使用内核。

笔记本范围的库注意事项和限制

使用笔记本范围的库时，请注意以下几点：

- 使用 Amazon EMR 发行版 5.26.0 及更高版本创建的集群现已提供笔记本电脑实例范围的库。
- 笔记本范围的库只能与 PySpark 内核结合使用。
- 任何用户均可从笔记本单元中安装其它笔记本范围的库。这些库仅在单个笔记本会话期间对笔记本用户可用。如果其它用户需要相同的库，或者同一用户在不同的会话中需要相同的库，则必须重新安装该库。
- 您只能卸载使用 `install_pypi_package` API 安装的库，而不能卸载集群上预装的任何库。
- 如果不同版本的相同库分别安装在了集群上和用作笔记本范围的库，则笔记本范围的库版本将覆盖集群库版本。

使用笔记本范围的库

要安装库，您的 Amazon EMR 集群必须能够访问库所在的 PyPI 存储库。

以下示例演示了一些简单命令，这些命令使用 PySpark 内核和 API 从笔记本单元中列出、安装和卸载库。有关其它示例，请参阅 Amazon 大数据博客上的 Install Python libraries on a running cluster with EMR Notebooks 一文。

Example – 列出当前库

以下命令列出了可用于当前 Spark 笔记本会话的 Python 包。这将列出集群上安装的库以及笔记本范围的库。

```
sc.list_packages()
```

Example – 安装 Celery 库

以下命令安装 Celery 库作为笔记本范围的库。

```
sc.install_pypi_package("celery")
```

安装库后，以下命令将确认该库在 Spark 驱动程序和执行程序上是否可用。

```
import celery
sc.range(1,10000,1,100).map(lambda x: celery.__version__).collect()
```

Example – 安装 Arrow 库，指定版本和存储库

以下命令安装 Arrow 库作为笔记本范围的库，并指定库版本和存储库 URL。

```
sc.install_pypi_package("arrow==0.14.0", "https://pypi.org/simple")
```

Example – 卸载库

以下命令卸载 Arrow 库，将其作为笔记本范围的库从当前会话中删除。

```
sc.uninstall_package("arrow")
```

将基于 Git 的存储库与 EMR Notebooks 关联

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM

角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

可以将基于 Git 的存储库与 Amazon EMR Notebooks 关联，以将笔记本保存在版本控制的环境中。一个笔记本最多可与三个存储库关联。支持以下基于 Git 的服务：

- Amazon CodeCommit
- GitHub
- Bitbucket
- GitLab

将基于 Git 的存储库与您的笔记本相关联后，将会获得以下优势。

- 版本控制 – 您可以在版本控制系统中记录代码更改，以便查看更改历史记录并有选择地撤消一些更改。
- 协作 – 在不同笔记本中工作的同事可以通过基于 Git 的远程存储库共享代码。笔记本可以从远程存储库克隆或合并代码，然后将更改推送回这些远程存储库。
- 代码重用 – 许多演示数据分析或机器学习技术的 Jupyter notebook 可在公开托管的存储库上提供，例如 GitHub。可以将您的笔记本与存储库关联，以便重用包含在该存储库中的 Jupyter notebook。

要将基于 Git 的存储库与 EMR Notebooks 结合使用，请在 Amazon EMR 控制台中添加存储库作为资源，关联需要身份验证的存储库的凭证，然后将这些存储库与笔记本链接起来。您可在 Amazon EMR 控制台中查看存储在账户中的存储库列表以及各个存储库的详细信息。在创建基于 Git 的现有存储库时，可以将这些存储库与笔记本关联。

主题

- 先决条件和注意事项
- 将基于 Git 的存储库添加到 Amazon EMR
- 更新或删除基于 Git 的存储库。
- 链接或取消链接基于 Git 的存储库
- 使用关联的 Git 存储库创建新的笔记本
- 在笔记本中使用 Git 存储库

先决条件和注意事项

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

在计划将基于 Git 的存储库与 EMR Notebooks 集成时，请考虑以下事项。

Amazon CodeCommit

如果您使用 CodeCommit 存储库，则必须对 CodeCommit 使用 Git 凭证和 HTTPS。不支持 SSH 密钥和带有 Amazon CLI 凭证辅助程序的 HTTPS。CodeCommit 不支持个人访问令牌（PAT）。有关更多信息，请参阅《IAM 用户指南》中的将 IAM 与 CodeCommit 结合使用：Git 凭证、SSH 密钥和 Amazon 访问密钥和《Amazon CodeCommit 用户指南》中的适用于使用 Git 凭证的 HTTPS 用户的设置。

访问和权限注意事项

在将存储库与笔记本关联之前，请确保集群、EMR Notebooks 的 IAM 角色和安全组具有正确的设置和权限。您还可以按照为 EMR Notebooks 配置私有托管的 Git 存储库中的说明配置在专用网络中托管的基于 Git 的存储库。

- 集群互联网访问 – 启动的网络接口只有一个私有 IP 地址。这意味着笔记本连接到的集群必须位于具有网络地址转换 (NAT) 网关的私有子网中，或者必须能够通过虚拟私有网关访问 Internet。有关更多信息，请参阅 Amazon VPC 选项。

  笔记本的安全组必须包含出站规则，以便允许笔记本将流量从集群路由到 Internet。建议您创建自己的安全组。有关更多信息，请参阅为 EMR Notebooks 指定 EC2 安全组。

  > **Important**
  >
  > 如果网络接口启动到公有子网中，它将无法通过互联网网关（IGW）与互联网通信。

- Amazon Secrets Manager 的权限 – 如果您使用 Secrets Manager 来存储用于访问存储库的密钥，则 the section called “EMR Notebooks 角色” 必须附加允许执行 `secretsmanager:GetSecretValue` 操作的权限策略。

为 EMR Notebooks 配置私有托管的 Git 存储库

使用以下说明为 EMR Notebooks 配置私有托管的存储库。您必须提供一个配置文件，其中包含有关您的 DNS 和 Git 服务器的信息。Amazon EMR 使用此信息配置将流量路由到您的私有托管存储库的 EMR 笔记本。

先决条件

在为 EMR Notebooks 配置私有托管的 Git 存储库之前，您必须具备以下条件：

- 将保存 EMR 笔记本文件的 Amazon S3 Control 位置。

为 EMR Notebooks 配置一个或多个私有托管的 Git 存储库

1. 使用提供的模板创建配置文件。为要在配置中指定的每个 Git 服务器包括以下值：

   - **`DnsServerIpV4`** – 您的 DNS 服务器的 IPv4 地址。如果您同时提供 `DnsServerIpV4` 和 `GitServerIpV4List` 的值，则 `DnsServerIpV4` 的值优先，并将用于解析您的 `GitServerDnsName`。

     > **Note**
     >
     > 要使用私有托管的 Git 存储库，您的 DNS 服务器必须允许来自 EMR Notebooks 的入站访问。我们强烈建议您保护您的 DNS 服务器，防止其它未经授权的访问。

   - **`GitServerDnsName`** – 您的 Git 服务器的 DNS 名称。例如 `"git.example.com"`。
   - **`GitServerIpV4List`** – 属于您的 Git 服务器的 IPv4 地址列表。

```
[
    {
        "Type": "PrivatelyHostedGitConfig",
        "Value": [
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<enterprise.git.com>",
```

```
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            },
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<git.example.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            }
        ]
    }
]
```

2. 将配置文件另存为 `configuration.json`。
3. 将配置文件上载到名为 `life-cycle-configuration` 的文件夹中的指定 Amazon S3 存储位置。例如，如果您的默认 S3 位置是 `s3://DOC-EXAMPLE-BUCKET/notebooks`，则您的配置文件应位于 `s3://DOC-EXAMPLE-BUCKET/notebooks/`**`life-cycle-configuration/configuration.json`**。

> **Important**
>
> 我们强烈建议您将 `life-cycle-configuration` 文件夹的访问权限限制为仅您的 EMR Notebooks 管理员以及 EMR Notebooks 的服务角色。您还应该确保 `configuration.json` 免受未经授权的访问。有关说明，请参阅使用用户策略控制对存储桶的访问或 Amazon S3 的安全最佳实践。

有关上载说明，请参阅《Amazon Simple Storage Service 用户指南》中的创建文件夹和上载对象。

将基于 Git 的存储库添加到 Amazon EMR

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

有关如何将基于 Git 的存储库添加到旧控制台中的 EMR 笔记本或新控制台中的 EMR Studio Workspace，请参阅以下各节的内容。

New console

EMR Notebooks 便是新控制台中的 EMR Studio Workspaces，因此您可以按照 将基于 Git 的存储库链接到 EMR Studio Workspace 中的说明将最多三个 Git 存储库关联到 Workspace。

此外，您也可以使用 JupyterLab Git 扩展程序。从 Jupyterlab 笔记本的左侧边栏中选择 Git 图标来访问扩展程序。有关扩展程序的信息，请参阅 jupyterlab-git GitHub 存储库。

要将 Git 存储库关联到 Workspace，Studio 管理员必须配置 Studio 以允许 Git 存储库链接。有关更多信息，请参阅为基于 Git 的存储库建立访问和权限。

Old console

使用旧控制台将基于 Git 的存储库添加为 Amazon EMR 账户中的资源

1. 通过以下链接打开 Amazon EMR 旧控制台：https://console.aws.amazon.com/elasticmapreduce。
2. 选择 Git repositories (Git 存储库)，然后选择 Add repository (添加存储库)。
3. 对于 Repository name (存储库名称)，输入在 Amazon EMR 中为存储库使用的名称。

   名称只能包含字母数字字符、连字符 (-) 或下划线 (_)。
4. 对于 Git repository URL (Git 存储库 URL)，输入存储库的 URL。使用 CodeCommit 存储库时，这是当您选择 Clone URL (克隆 URL)、然后选择 Clone HTTPS (克隆 HTTPS) 时复制的 URL，例如 `https://git-codecommit.us-west-2.amazonaws.com/v1/repos/MyCodeCommitRepoName`。

5. 对于 Branch (分支)，输入分支名称。
6. 对于 Git credentials (Git 凭证)，请根据以下准则选择选项。可以使用 Git 用户名和密码或个人访问令牌（PAT）对您的存储库进行身份验证。EMR Notebooks 使用存储在 Secrets Manager 中的密钥访问您的 Git 凭证。

> **Note**
>
> 如果您使用 GitHub 存储库，我们建议您使用个人访问令牌（PAT）进行身份验证。从 2021 年 8 月 13 日开始，GitHub 在对 Git 操作进行身份验证时将不再接受密码。有关更多信息，请参阅 GitHub 博客中的 Token authentication requirements for Git operations 一文。

选项	描述
使用现有 Amazon 密钥	如果您已将凭证保存为 Secrets Manager 中的密钥，请选择此选项，然后从列表中选择密钥名称。 如果您选择与 Git 用户名和密码关联的密钥，则密钥必须采用 `{"gitUsername": "MyUserName ", "gitPassword": "MyPassword "}` 格式。
创建新密钥	选择此选项可将现有 Git 凭证与您在 Secrets Manager 中创建的新密钥相关联。根据您用于存储库的 Git 凭证，执行以下操作之一。 如果您使用 Git 用户名和密码访问存储库，请选择 Username and password（用户名和密码），输入要在 Secrets Manager 中使用的 Secret name（密钥名称），然后输入要与密钥关联的 Username（用户名）和 Password（密码）。 –或者–

选项	描述
	如果您使用个人访问令牌访问存储库，请选择 Personal access token (PAT) (个人访问令牌 (PAT))，输入要在 Secrets Manager 中使用的 Secret name (密钥名称)，然后输入您的个人访问令牌。 有关更多信息，请参阅为 GitHub 的命令行创建个人访问令牌和 Bitbucket 的个人访问令牌。CodeCommit 存储库不支持此选项。
在没有凭证的情况下使用公有存储库	选择此选项可访问公有存储库。

7. 选择 Add repository (添加存储库)。

更新或删除基于 Git 的存储库。

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

要了解如何从旧控制台中的 EMR 笔记本或新控制台中的 EMR Studio Workspace 删除基于 Git 的存储库，请参阅以下各节的内容。

New console

EMR Notebooks 便是新控制台中的 EMR Studio Workspaces，因此您可以参考 将基于 Git 的存储库链接到 EMR Studio Workspace 来获取有关在 Workspace 中使用 Git 存储库的更多信息。但是目前，您无法从 Workspace 中删除 Git 存储库。

Old console

在旧控制台中更新基于 Git 的存储库

1. 在 Git repositories (Git 存储库) 页面上，选择要更新的存储库。
2. 在存储库页面上，选择 Edit repository (编辑存储库)。
3. 更新存储库页面上的 Git credentials (Git 凭证)。

在旧控制台中删除 Git 存储库

1. 在 Git repositories (Git 存储库) 页面上，选择要删除的存储库。
2. 在存储库页面上，选择当前已链接到存储库的所有笔记本。选择 Unlink notebook (取消笔记本链接)。
3. 在存储库页面上，选择 Delete (删除)。

> **Note**
>
> 要从 Amazon EMR 中删除本地 Git 存储库，您必须先从该存储库取消所有笔记本的链接。有关更多信息，请参阅链接或取消链接基于 Git 的存储库。删除 Git 存储库将不会删除为存储库创建的任何密钥。您可以删除 Amazon Secrets Manager 中的密钥。

链接或取消链接基于 Git 的存储库

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

按照以下步骤将基于 Git 的存储库链接到或取消链接到旧控制台中的 EMR 笔记本，或新控制台中的 EMR Studio Workspace。

New console

EMR Notebooks 便是新控制台中的 EMR Studio Workspaces，因此您可以参考 将基于 Git 的存储库链接到 EMR Studio Workspace 来获取有关在 Workspace 中使用 Git 存储库的更多信息。但是目前，您无法从 Workspace 中删除 Git 存储库。

Old console

将基于 Git 的存储库链接到 EMR Notebooks

一旦笔记本处于 Ready (就绪) 状态，便能将存储库链接到笔记本。

1. 从 Notebooks (笔记本) 列表中，选择要更新的笔记本。
2. 在 Notebook (笔记本) 页面的 Git repositories (Git 存储库) 部分，选择 Link new repository (链接新存储库)。
3. 在 Link Git repository to notebook (将 Git 存储库链接到笔记本) 窗口的存储库列表中，选择一个或多个要链接到笔记本的存储库，然后选择 Link repository (链接存储库)。

或者

1. 在 Git repositories (Git 存储库) 页面上，选择要链接到笔记本的存储库。
2. 在 EMR notebooks (EMR Notebooks) 列表中，选择 Link new notebook (链接新笔记本) 以将此存储库链接到现有笔记本。

从 EMR Notebooks 取消 Git 存储库的链接

1. 从 Notebooks (笔记本) 列表中，选择要更新的笔记本。
2. 在 Git repositories (Git 存储库) 列表中，选择要从笔记本取消链接的存储库，然后选择 Unlink repository (取消链接存储库)。

或者

1. 在 Git repositories (Git 存储库) 页面上，选择要更新的存储库。
2. 在 EMR notebooks (EMR Notebooks) 列表中，选择要从存储库取消链接的笔记本，然后选择 Unlink notebook (取消链接笔记本)。

> **Note**
>
> 将 Git 存储库链接到笔记本会将远程存储库克隆到本地 Jupyter notebook。从笔记本取消 Git 存储库的链接只会断开笔记本与远程存储库的连接，而不会删除本地 Git 存储库。

了解存储库状态

Git 存储库在存储库列表中可能具有以下任何状态。有关将 EMR Notebooks 与 Git 存储库关联的更多信息，请参阅链接或取消链接基于 Git 的存储库。

状态	含义
链接	Git 存储库正在链接到笔记本。当存储库处于 Linking (正在链接) 状态时，您无法停止笔记本。
已链接	Git 存储库将链接到笔记本。如果存储库具有 Linked (已链接) 状态时，则它已连接到远程存储库。
链接失败	Git 存储库未能链接到笔记本。您可以重新尝试链接它。
正在取消链接	正在从笔记本取消 Git 存储库的链接。当存储库处于 Unlinking (正在取消链接) 状态时，您无法停止笔记本。从笔记本取消 Git 存储库的链接只会断开它与远程存储库的连接，而不会从笔记本中删除任何代码。
取消链接失败	未能从笔记本取消 Git 存储库的链接。您可以重新尝试取消该存储库的链接。

使用关联的 Git 存储库创建新的笔记本

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

在 Amazon EMR 旧控制台中创建一个笔记本并将之关联到 Git 存储库

1. 按照创建笔记本中的说明进行操作。
2. 对于 Security group (安全组)，选择 Use your own security group (使用您自己的安全组)。

   > **Note**
   >
   > 笔记本的安全组必须包含出站规则，以允许笔记本通过集群将流量路由到 Internet。建议您创建自己的安全组。有关更多信息，请参阅为 EMR Notebooks 指定 EC2 安全组。

3. 对于 Git repositories (Git 存储库)，选择要与笔记本关联的存储库。

   1. 选择作为资源存储在您账户中的存储库，然后选择 Save (保存)。
   2. 要在账户中添加新存储库作为资源，请选择 add a new repository (添加新存储库)。在新窗口中完成 Add repository (添加存储库) 工作流。

在笔记本中使用 Git 存储库

> **Note**
>
> EMR Notebooks 在新控制台中作为 Amazon EMR Studio Workspaces 提供。您仍然可以在旧控制台中使用现有笔记本，但无法在其中创建新笔记本。新控制台中的创建 Workspace 按钮将取代此功能。要访问或创建 Workspaces，EMR Notebooks 用户需要额外的 IAM 角色权限。有关更多信息，请参阅 Amazon EMR Notebooks are Amazon EMR Studio

Workspaces in new console（Amazon EMR Notebooks 在新控制台中为 Amazon EMR Studio Workspaces）和 What's new in the console?（控制台中有哪些新功能？）

在打开笔记本时，您可以选择 Open in JupyterLab (在 JupyterLab 中打开) 或 Open in Jupyter (在 Jupyter 中打开)。

如果您选择在 Jupyter 中打开笔记本，则会显示笔记本中的可扩展文件和文件夹的列表。您可以在笔记本单元格中手动运行 Git 命令，如下所示。

```
!git pull origin primary
```

要打开任何额外的存储库，请导航到其它文件夹。

如果您选择打开具有 JupyterLab 界面的笔记本，则可以使用预先安装的 JupyterLab Git 扩展程序。有关扩展程序的更多信息，请参阅 jupyterlab-git。

计划和配置集群

此部分说明了有关使用 Amazon EMR 计划、配置和启动集群的配置选项和说明。在启动集群之前，您根据所处理的数据以及成本、速度、容量、可用性、安全性和可管理性的要求做出有关系统的选择。您的选择包括：

- 集群运行所在的区域、在何处以及如何存储数据，以及如何输出结果。请参阅配置集群位置和数据存储。
- 无论您是在 Outposts 还是在 Local Zones 运行 Amazon EMR 集群。请参阅Amazon Outposts 上的 EMR 集群或Amazon Local Zones 上的 EMR 集群。
- 集群是长时间运行的还是临时的，以及它运行哪个软件。请参阅 将集群配置为在步骤执行后继续或终止 和 配置集群软件。
- 集群是单个主节点模式还是三个主节点模式。请参阅计划和配置主节点。
- 用于优化应用程序的成本、性能和可用性的硬件和联网选项。请参阅配置集群硬件和联网。
- 如何设置集群以便您能更轻松地管理集群，以及监控活动、性能和运行状况。请参阅 配置集群日志记录和调试 和 为集群添加标签。
- 如何对集群资源进行身份验证和授予对该资源的访问权，以及如何加密数据。请参阅Amazon EMR 中的安全性。
- 如何与其它软件和服务集成。请参阅驱动程序和第三方应用程序集成。

快速启动集群

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台快速启动集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr/clusters。

2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Create Cluster（创建集群）页面上，输入或选择所提供字段的值。持久性摘要面板显示您当前选择的集群选项的实时视图。在摘要面板中选择一个标题，以导航到相应部分并进行调整。您必须先完成所有必需的配置，然后才能选择 Create Cluster（创建集群）。
4. 选择 Create Cluster（创建集群）以接受配置，如图所示。
5. 集群详细信息页面随即打开。找到集群名称旁边的集群 Status（状态）。在集群创建过程中，状态应从 Starting (正在启动) 变为 Running (正在运行) 再变为 Waiting (正在等待)。您可能需要选择右上角的刷新图标或刷新您的浏览器才能接收更新。

   当状态变更为 Waiting (正在等待) 后，表明您的集群已启动、正在运行，并准备接受步骤和 SSH 连接。

Old console

使用 Amazon EMR 旧控制台中的 Create Cluster - Quick Options（创建集群 – 快速选项）页面，快速创建用于简单任务或用于评估或测试的集群。Quick Options (快速选项) 使用集群配置选项（如软件、联网和安全等）的默认值。

使用旧控制台的 Quick Options（快速选项）启动集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Clusters (集群)，然后选择 Create cluster (创建集群)，打开 Quick Options (快速选项) 页面。
3. 在 Create Cluster - Quick Options (创建集群 – 快速选项) 页面上，输入或选择所提供字段的值。
4. 选择 Create cluster (创建集群)，启动集群并打开集群状态页面。
5. 在集群状态页面上，找到集群名称旁边的 Status (状态)。在集群创建过程中，状态应从 Starting (正在启动) 变为 Running (正在运行) 再变为 Waiting (正在等待)。您可能需要选择右侧的刷新图标或刷新您的浏览器才能接收更新。

   当状态变更为 Waiting (正在等待) 后，表明您的集群已启动、正在运行，并准备接受步骤和 SSH 连接。

配置集群位置和数据存储

此部分会介绍如何为集群配置区域、使用 Amazon EMR 时可用的其它文件系统以及如何使用这些文件系统。它还会介绍如何准备数据或将数据上载到 Amazon EMR（如果需要），以及如何为日志文件和您配置的所有输出数据文件准备输出位置。

主题

- 选择一个Amazon区域
- 使用存储和文件系统
- 准备输入数据
- 配置输出位置

选择一个Amazon区域

Amazon Web Services 在全球范围内的数据中心的服务器上运行。数据中心按地理区域进行管理。启动 Amazon EMR 集群时，必须指定区域。您可以选择一个区域，以便减少延迟、尽可能降低成本或满足法规要求。有关 Amazon EMR 支持的所有区域和端点的列表，请参阅《Amazon Web Services 一般参考》 中的 Regions and endpoints。

为了获得最佳性能，您应在与您的数据相同的区域启动集群。例如，如果存储您的输入数据的 Amazon S3 存储桶位于美国西部（俄勒冈州）区域，那么您应在美国西部（俄勒冈州）区域启动集群，以避免缴纳跨区域数据传输费。如果您使用 Amazon S3 存储桶接收集群输出，那么也应该在美国西部（俄勒冈州）区域创建该集群输出。

如果您打算将某个 Amazon EC2 密钥对与集群相关联（这是使用 SSH 登录主节点所必需的操作），那么必须在集群所在区域创建密钥对。与之类似，Amazon EMR 创建的用以管理集群的安全组也应在集群所在区域内创建。

如果您是在 2017 年 5 月 17 日或之后注册的 Amazon Web Services 账户，则您从 Amazon Web Services Management Console 访问资源的默认区域是美国东部（俄亥俄）（us-east-2）；对于旧账户，默认区域是美国西部（俄勒冈）（us-west-2）或美国东部（弗吉尼亚北部）（us-east-1）。有关更多信息，请参阅 区域和端点。

某些 Amazon 功能仅在部分区域可用。例如，集群计算实例仅在美国东部（弗吉尼亚州北部）区域可用，而亚太地区（悉尼）区域仅支持 Hadoop 1.0.3 及更高版本。当您选择一个区域时，请核实该区域是否支持您想要使用的功能。

为了获得最佳性能，对于所有将用于集群的 Amazon 资源，请使用同一区域。下表列出了区域名称和服务的映射关系。有关 Amazon EMR 区域的列表，请参阅《Amazon Web Services 一般参考》 中的 Amazon Web Services 区域 and endpoints。

使用控制台选择区域

默认区域显示在导航栏上账户信息的左侧。要在新旧控制台中切换区域，请选择“区域”下拉菜单并选择一个新选项。

使用 Amazon CLI 指定区域

可在 Amazon CLI 中使用 aws configure 命令或 AWS_DEFAULT_REGION 环境变量指定默认区域。有关更多信息，请参阅《Amazon Command Line Interface 用户指南》中的配置Amazon区域。

选择一个使用 SDK 或 API 的区域

要使用软件开发工具包选择区域，请将应用程序配置为使用该区域的端点。如果您正在使用 Amazon SDK 创建客户端应用程序，那么您可以通过调用 setEndpoint 来变更客户终端节点，如下例所示：

```
client.setEndpoint("elasticmapreduce.us-west-2.amazonaws.com");
```

在您的应用程序通过设置端点指定一个区域后，您可以为集群的 EC2 实例设置可用区。可用区是专用于隔离其他可用区内故障的区分位置，可向相同区域中的其他可用区提供低延迟的便宜网络连接。一个区域包含一个或多个可用区。为了优化性能和降低延迟，所有资源应与使用它们的集群位于同一可用区。

使用存储和文件系统

Amazon EMR 和 Hadoop 提供了各种文件系统，供您在处理集群步骤时使用。通过用于访问数据的 URI 前缀，您可以指定要使用的文件系统类型。例如，s3://*DOC-EXAMPLE-BUCKET1*/path 会引用使用 EMRFS 的 Amazon S3 存储桶。下表列出了可用的文件系统以及关于最适合用途的建议。

Amazon EMR 和 Hadoop 处理集群时通常会使用两个或多个以下文件系统。HDFS 和 EMRFS 是与 Amazon EMR 配合使用的两种主要文件系统。

> **Important**
>
> 从 Amazon EMR 发行版 5.22.0 开始，Amazon EMR 专门使用 Amazon 签名版本 4 来对 Amazon S3 的请求进行身份验证。除非发布说明指出需专门使用签名版本 4，否则早期 Amazon EMR 发行版在某些情况下使用 Amazon 签名版本 2。有关更多信息，请参阅

《Amazon Simple Storage Service 开发人员指南》中的对请求进行身份验证（Amazon签名版本 4）和对请求进行身份验证（Amazon签名版本 2）。

文件系统	Prefix	描述
HDFS	`hdfs://`（或无前缀）	HDFS 是适用于 Hadoop 的一种可扩展的分布式便携文件系统。HDFS 的一项优势是管理集群的 Hadoop 集群节点与管理单一步骤的 Hadoop 集群节点之间的数据感知。有关更多信息，请参阅 Hadoop 文档。 通过主节点和核心节点使用 HDFS。优势是快；劣势是它是短暂存储，会在集群终止时回收。它最适合用于缓存由中间任务流程步骤产生的结果。
EMRFS	`s3://`	EMRFS 是 Hadoop 文件系统的一种实现方式，用于从 Amazon EMR 读取常规文件并将其直接写入到 Amazon S3。通过 EMRFS 可以方便地将持久性数据存储在 Amazon S3 中，以便用于 Hadoop，同时它还提供诸如 Amazon S3 服务器端加密、先写后读一致性和列表一致性等功能。 **Note** 之前，Amazon EMR 使用 `s3n` 和 `s3a` 文件系统。尽管这两种文件系统仍然有效，不过我们建议您使用 `s3` URI 方案以获得最佳性能、安全性和可靠性。
本地文件系统		本地文件系统指的是本地连接的磁盘。创建 Hadoop 集群时，会从 EC2 实例上创建各个节点，这些节点附带了预先配置的数据块，这些数据块属于称为实例存储 的预先附加的磁盘存储。实例存储卷上的数据仅在 EC2 实例的生命周期内保留。实例存储卷适合于存储不断变化的临时数据（如缓冲区、缓存、Scratch 数据和其他临时内容）。有关更多信息，请参阅 Amazon EC2 实例存储。

文件系统	Prefix	描述
		HDFS 使用本地文件系统，但 Python 也从本地文件系统运行，您可以选择在实例存储卷上存储其他应用程序文件。
（早期）Amazon S3 数据块文件系统	`s3bfs://`	Amazon S3 数据块文件系统是一种早期文件存储系统。我们强烈反对使用此系统。 **Important** 我们不推荐使用此文件系统，因为它可能触发竞争条件，此条件可能会导致集群故障。但是，早期应用程序可能会需要它。

访问文件系统

通过用于访问数据的统一资源标识符 (URI) 前缀，您可以指定要使用的文件系统。以下步骤介绍了如何引用几种不同类型的文件系统。

访问本地 HDFS

- 在 URI 中指定 `hdfs:///` 前缀。Amazon EMR 会将未在 URI 中指定前缀的路径解析为本地 HDFS。例如，以下两个 URI 都会解析至 HDFS 中的相同位置。

```
hdfs:///path-to-data

/path-to-data
```

访问远程 HDFS

- 将主节点 IP 地址包含在 URI 内，如以下示例所示。

```
hdfs://master-ip-address/path-to-data
```

```
master-ip-address/path-to-data
```

访问 Amazon S3

- 使用 s3:// 前缀。

```
s3://bucket-name/path-to-file-in-bucket
```

访问 Amazon S3 数据块文件系统

- 仅用于需要 Amazon S3 数据块文件系统的旧版应用程序。要使用此文件系统访问或存储数据，请在 URI 中使用 s3bfs:// 前缀。

  Amazon S3 数据块文件系统是一种旧版文件系统，用于支持向 Amazon S3 上载大于 5GB 的文件。借助 Amazon EMR 通过 Amazon Java SDK 提供的分段上载功能，您可以上载最大为 5TB 的文件到 Amazon S3 本机文件系统，而 Amazon S3 数据块文件系统已被弃用。

  > **Warning**
  >
  > 因为此早期文件系统可能创建会导致文件系统损坏的竞争条件，所以您应当避免使用此形式，而改用 EMRFS。

```
s3bfs://bucket-name/path-to-file-in-bucket
```

准备输入数据

多数集群会先加载输入数据，然后再处理这些数据。为加载数据，必须定位在集群可以访问的位置，并且格式必须为集群可以处理的类型。最常见的情形是将输入数据上载至 Amazon S3。Amazon EMR 为您的集群提供了从 Amazon S3 导入或读取数据的工具。

Hadoop 中的默认输入格式是文本文件，然而您也可以自定义 Hadoop 从而使用工具导入以其它格式存储的数据。

主题

- Amazon EMR 可以接受的输入类型
- 如何将数据导入 Amazon EMR

Amazon EMR 可以接受的输入类型

集群的默认输入格式是文本文件，每行都使用换行符 (\n) 分隔，这是最常用的输入格式。

如果您输入的数据格式不是默认的文本文件，可以使用 Hadoop 接口 `InputFormat` 指定其它输入类型。您甚至可以创建 `FileInputFormat` 类型的子类，用于处理自定义数据类型。有关更多信息，请参阅 http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/InputFormat.html。

如果您正在使用 Hive，可以使用串行器/解串器 (SerDe) 将给定格式的数据读入 HDFS。有关更多信息，请参阅 https://cwiki.apache.org/confluence/display/Hive/SerDe。

如何将数据导入 Amazon EMR

Amazon EMR 提供了几种方法将数据导入到集群上。最常见的方法是将数据上载到 Amazon S3，然后使用 Amazon EMR 的内置功能将数据加载到集群上。您还可以使用 Hadoop 的 DistributedCache 功能将文件从分布式文件系统传输到本地文件系统。Amazon EMR（Hive 版本 0.7.1.1 及更高版本）提供的 Hive 实施包括一项功能，您可以使用此功能在 DynamoDB 和 Amazon EMR 集群之间导入和导出数据。如果有大量的本地数据要处理，您会发现 Amazon Direct Connect 服务非常有用。

主题

- 将数据上载到 Amazon S3
- 使用 Amazon DataSync 上载数据
- 使用分布式缓存导入文件
- 如何处理压缩文件
- 将 DynamoDB 数据导入到 Hive
- 使用 Amazon Direct Connect 连接到数据
- 使用 Amazon Snowball 上载大量数据

将数据上载到 Amazon S3

有关如何将文件上传到 Amazon S3 的说明，请参阅《Amazon Simple Storage Service 用户指南》中的向存储桶添加对象。有关将 Amazon S3 与 Hadoop 结合使用的更多信息，请参阅 http://wiki.apache.org/hadoop/AmazonS3。

主题

- 创建和配置 Amazon S3 存储桶
- 为 Amazon S3 配置分段上载
- 最佳实践

创建和配置 Amazon S3 存储桶

Amazon EMR 将 Amazon SDK for Java 与 Amazon S3 结合使用来存储输入数据、日志文件和输出数据。Amazon S3 将这些存储位置称为存储桶。存储桶具有某些限制以符合 Amazon S3 和 DNS 要求。有关更多信息，请参阅 《Amazon Simple Storage Service 用户指南》中的存储桶限制。

本部分会介绍如何使用 Amazon S3 Amazon Web Services Management Console创建存储桶，并为 Amazon S3 存储桶设置权限。您还可以使用 Amazon S3 API 或 Amazon CLI 为 Amazon S3 存储桶创建和设置权限。您还可以将 curl 与修改一起使用来传递 Amazon S3 的相应身份验证参数。

请参阅以下资源：

- 要使用控制台创建存储桶，请参阅《Amazon S3 用户指南》https://docs.amazonaws.cn/AmazonS3/latest/userguide/create-bucket.html中的创建存储桶。
- 要使用 Amazon CLI 创建并使用存储桶，请参阅《Amazon S3 用户指南》中的通过 Amazon Command Line Interface 使用高级别 S3 命令。
- 要使用 SDK 创建存储桶，请参阅《Amazon Simple Storage Service 用户指南》中的创建存储桶的示例。
- 要通过 curl 使用存储桶，请参阅 curl 的 Amazon S3 身份验证工具。
- 有关指定特定于区域的存储桶的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的访问存储桶。
- 要使用 Amazon S3 访问点处理存储桶，请参阅《Amazon S3 用户指南》中的为接入点使用存储桶式别名。您可以轻松地将 Amazon S3 接入点与 Amazon S3 访问点别名一起使用，而不使用 Amazon S3 存储桶名称。您可以将 Amazon S3 接入点别名用于现有应用程序和新的应用程序，包括 Spark、Hive、Presto 等其他应用程序。

> **Note**
>
> 如果您为存储桶启用日志记录，则只启用存储桶访问日志，而不会启用 Amazon EMR 集群日志。

在存储桶创建期间或创建后，您可以根据应用程序设置适当权限来访问存储桶。一般情况下，您可以给自己（所有者）授予读取和写入权限，并给已验证用户授予读取访问权限。

在创建集群之前，所需的 Amazon S3 存储桶必须存在。您必须将集群中引用的所有必要的脚本或数据上载至 Amazon S3。下表介绍了示例数据、脚本和日志文件位置。

为 Amazon S3 配置分段上载

Amazon EMR 通过 Amazon SDK for Java 支持 Amazon S3 分段上载。分段上载可让您将单独对象作为一组片段上载。您可以独立上载以及按任意顺序上载这些对象分段。如果任意分段传输失败，可以重新传输该分段且不会影响其它分段。上载完所有的对象分段后，Amazon S3 将汇集这些分段并创建对象。

有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 分段上传概述。

此外，Amazon EMR 提供的属性可让您更精确地对分段上载失败的分段进行清理。

下表描述了分段上载的 Amazon EMR 配置属性。您可以使用 `core-site` 配置分类配置这些属性。有关更多信息，请参阅《Amazon EMR 版本指南》中的配置应用程序。

配置参数名称	默认值	描述
`fs.s3n.multipart.uploads.enabled`	`true`	布尔值类型，指示是否要启用分段上载。当 EMRFS 一致视图处于启用状态时，系统会默认启用分段上传，而将此值设置为 `false` 会被忽略。
`fs.s3n.multipart.uploads.split.size`	`134217728`	在 EMRFS 启动新的分段上载之前且分段上载处于启用状态时，指定分段的大小上限（以字节为单位）。最小值为 `5242880` (5 MB)。如果指定了较小值，则使用 `5242880`。最大值为 `5368709120` (5 GB)。如果指定了较大值，则使用 `5368709120` 。

配置参数名称	默认值	描述
		如果 EMRFS 客户端加密功能已禁用且 Amazon S3 优化的提交程序也处于禁用状态，则此值还可以控制数据文件的大小上限，相应数据文件可能会增大，直到 EMRFS 使用分段上载（而非 PutObject 请求）来上载文件。有关更多信息，请参阅
fs.s3n.ssl.enabled	true	布尔值类型，指示使用 http 还是 https。
fs.s3.buckets.create.enabled	false	布尔值类型，指示是否应在存储桶不存在时创建存储桶。设置为 false 会导致 CreateBucket 操作出现异常。
fs.s3.multipart.clean.enabled	false	布尔值类型，指示是否要启用后台定期清理未完成的分段上载。
fs.s3.multipart.clean.age.threshold	604800	long 类型，用于指定分段上载的最小有效期（以秒为单位），然后考虑对其进行清理。默认有效期为一周。
fs.s3.multipart.clean.jitter.max	10000	整数类型，指定在安排下一轮清理之前最大随机抖动的延迟（以秒为单位）增至 15 分钟的固定延迟。

禁用分段上传

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台禁用分段上传

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Software Settings（软件设置）下，输入下面的配置：`classification=core-site,properties=[fs.s3n.multipart.uploads.enabled=false]`。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台禁用分段上传

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 依次选择 Create cluster (创建集群)、Go to advanced options (转到高级选项)。
3. 在 Edit Software Settings（编辑软件设置）下，输入下面的配置：`classification=core-site,properties=[fs.s3n.multipart.uploads.enabled=false]`
4. 继续创建集群。

CLI

使用 Amazon CLI 禁用分段上载

此过程介绍如何使用 Amazon CLI 禁用分段上载。要禁用分段上载，请键入带 `create-cluster` 参数的 `--bootstrap-actions` 命令。

1. 使用以下内容创建一个文件 (`myConfig.json`)，并将其保存到您在其中运行该命令的同一目录中：

```
[
  {
    "Classification": "core-site",
```

```
    "Properties": {
      "fs.s3n.multipart.uploads.enabled": "false"
    }
  }
]
```

2. 请键入以下命令，将 *myKey* 替换为您的 EC2 密钥对的名称。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\ ) 。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^)。

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-5.36.1 --applications Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --configurations file://myConfig.json
```

CLI

使用 API 禁用分段上传

- 有关以编程方式使用 Amazon S3 分段上传的信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用 Amazon SDK for Java 进行分段上传。

  有关 Amazon SDK for Java 的更多信息，请参阅 Amazon SDK for Java。

最佳实践

以下是将 Amazon S3 存储桶与 EMR 集群结合使用的建议。

启用版本控制

版本控制是针对 Amazon S3 存储桶的建议的配置。一旦启用版本控制，则可确保即使数据被意外删除或覆盖，也可恢复数据。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用版本控制。

清理失败的分段上载

默认情况下，EMR 集群组件通过 Amazon SDK for Java 与 Amazon S3 API 结合来使用分段上载，以将日志文件和输出数据写入 Amazon S3。有关使用 Amazon EMR 更改与此配置相关的属性的信息，请参阅为 Amazon S3 配置分段上载。有时，上载大型文件会生成未完成的 Amazon S3 分段上载。在分段上载无法成功完成时，进行中的分段上载将继续占用存储桶并会产生存储费用。我们建议采用下列选项，以避免文件存储过多：

- 对于与 Amazon EMR 结合使用的存储桶，启用 Amazon S3 中的生命周期配置，以在上载启动日期三天后删除仍未完成的分段上载。利用生命周期配置规则，您可以创建控制对象的存储类和生命周期。有关更多信息，请参阅对象生命周期管理和使用存储桶生命周期策略中止未完成的分段上载。
- 通过将 `fs.s3.multipart.clean.enabled` 设置为 `TRUE` 并调整其它清理参数来启用 Amazon EMR 的分段清理功能。此功能可用于正常运行时间有限的高容量、大规模集群。在这种情况下，生命周期配置规则的 `DaysAfterIntitiation` 参数可能太长，即使设置为最小参数，也会导致 Amazon S3 存储中迅猛增加。Amazon EMR 的分段清理功能有便于实现更精确的控制。有关更多信息，请参阅为 Amazon S3 配置分段上载。

管理版本标记

我们建议您在 Amazon S3 中启用生命周期配置规则，以删除您与 Amazon EMR 结合使用的启用了版本控制的存储桶的过期对象删除标记。删除受版本控制的存储桶中的对象时将创建删除标记。如果对象的所有早期版本随后过期，则存储桶中将保留过期的对象删除标记。虽然您无需为删除标记付费，但删除过期的标记可提高 LIST 请求的性能。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的在启用了版本控制的存储桶中进行生命周期配置。

性能最佳实践

根据您的工作负载，对 EMR 集群及其上应用程序的特定类型使用会导致对存储桶的大量请求。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的请求率和性能注意事项。

使用 Amazon DataSync 上载数据

Amazon DataSync 是一种在线数据传输服务，它可简化、自动化和加快在本地存储与 Amazon 存储服务之间或者在多项 Amazon 存储服务之间移动数据的过程。DataSync 支持各种本地存储系统，例如 Hadoop Distributed File System（HDFS）、NAS 文件服务器和自行管理的对象存储。

数据导入到集群的最常见方法是将数据上载到 Amazon S3，然后使用 Amazon EMR 的内置功能将数据加载到集群上。

DataSync 可以帮助您完成以下任务：

- 将 Hadoop 集群上的 HDFS 复制到 Amazon S3 以实现业务连续性
- 将 HDFS 复制到 Amazon S3 以填充数据湖
- 在 Hadoop 集群的 HDFS 与 Amazon S3 之间传输数据以进行分析和处理

要将数据上传到 S3 存储桶，请首先在与本地存储相同的网络中部署一个或多个 DataSync 代理。代理是用于从自行管理位置读取数据或向自行管理位置写入数据的虚拟机 (VM)。然后，您可以在 S3 存储桶所在的 Amazon Web Services 账户 和 Amazon Web Services 区域 中激活代理。

激活代理后，您可以为本地存储创建源位置，为 S3 存储桶创建目标位置，还要创建一个任务。任务由两个位置（源和目标）和一组用于控制任务行为的默认选项组成。

最后，运行 DataSync 任务，将数据从源位置传输到目标位置。

有关更多信息，请参阅 Getting started with Amazon DataSync。

使用分布式缓存导入文件

主题

- 受支持的文件类型
- 缓存文件的位置
- 从 Streaming 应用程序访问缓存文件
- 从 Streaming 应用程序访问缓存文件

DistributedCache 是一种 Hadoop 功能，可以在映射或归约任务需要访问通用数据时提高效率。如果集群所依赖的现有应用程序或二进制文件在创建集群时尚未安装，您可以使用 DistributedCache 导入这些文件。此功能让集群节点可以从它的本地文件系统中读取导入的文件，而不是从其它集群节点中检索文件。

有关更多信息，请转到 http://hadoop.apache.org/docs/stable/api/org/apache/hadoop/filecache/DistributedCache.html。

创建集群时，可以调用 DistributedCache。在即将开始 Hadoop 作业前，将文件读入缓存区，并且在作业持续期间文件会保留在缓存中。您可以对存储在任何 Hadoop 兼容文件系统（如 HDFS 或 Amazon S3）上的文件执行缓存操作。文件缓存的默认大小是 10 GB。要更改缓存大小，可使用引导操作重新配置 Hadoop 参数 `local.cache.size`。有关更多信息，请参阅创建引导操作以安装其它软件。

受支持的文件类型

DistributedCache 适用于单个文件和存档。单个文件缓存为只读模式。可执行文件和二进制文件具有执行权限设置。

存档是通过实用工具 (如 gzip) 打包的一个或多个文件。DistributedCache 将压缩文件传递到每个核心节点，然后将存档作为缓存的一部分进行解压缩。DistributedCache 支持以下压缩格式：

- zip
- tgz
- tar.gz
- tar
- jar

缓存文件的位置

DistributedCache 仅将文件复制到核心节点。如果集群中没有核心节点，DistributedCache 会将文件复制到主节点。

利用符号链接，DistributedCache 将缓存文件与映射器和折叠器（reducer）的当前工作目录相关联。符号链接是文件位置的一个别名，而不是实际的文件位置。参数值（yarn-site.xml 中的 yarn.nodemanager.local-dirs），可指定临时文件的位置。Amazon EMR 或基于实例类型和 EMR 版本的某些变体，将此参数设置为 /mnt/mapred。例如，某个设置可能包含 /mnt/mapred 和 /mnt1/mapred，因为此实例类型有两个短暂卷。缓存文件位于临时文件位置的子目录中：/mnt/mapred/taskTracker/archive。

如果您将单个文件读入缓存，那么 DistributedCache 会将此文件放在 archive 目录下。如果您将一份归档读入缓存，那么 DistributedCache 会解压缩该文件，然后使用与归档文件相同的名称在 /archive 下创建子目录。单个文件位于新的子目录下。

仅当使用 Streaming 时，才可以使用 DistributedCache。

从 Streaming 应用程序访问缓存文件

要从映射器或 Reducer 应用程序中访问缓存文件，您需要确保已将当前工作目录 (./) 添加到应用程序路径中，并引用了缓存文件，就好像这些文件存在于当前工作目录下一样。

从 Streaming 应用程序访问缓存文件

您可以使用 Amazon Web Services Management Console 和 Amazon CLI 创建使用分布式缓存的集群。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台指定分布式缓存文件

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Steps（步骤）下，选择 Add step（添加步骤）。Add step（添加步骤）对话框随即打开。在 Arguments（参数）字段，将要保存至缓存的文件和存档包含在内。文件大小 (或存档文件中文件的总计大小) 必须小于分配的缓存大小。

   如果要将单个文件添加到分布式缓存，请指定 `-cacheFile`，后跟文件名称和位置、井号（#）以及将文件放入本地缓存时要为其指定的名称。以下示例说明了如何将单个文件添加到分布式缓存。

   ```
   -cacheFile \
   s3://DOC-EXAMPLE-BUCKET/file-name#cache-file-name
   ```

   如果要将存档文件添加到分布式缓存，请输入 `-cacheArchive`，后跟文件在 Amazon S3 中的位置、井号（#）以及您要为本地缓存中的文件集合指定的名称。以下示例说明了如何将存档文件添加到分布式缓存。

   ```
   -cacheArchive \
   s3://DOC-EXAMPLE-BUCKET/archive-name#cache-archive-name
   ```

   在其他对话框字段中输入相应值。选项因步骤类型而异。要添加步骤并退出对话框，请选择 Add step（添加步骤）。

4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台指定分布式缓存文件

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Step execution (步骤执行) 作为启动模式。
4. 在 Steps (步骤) 部分的 Add step (添加步骤) 字段中，从列表中选择 Streaming program (流式程序)，并单击 Configure and add (配置并添加)。
5. 在 Arguments（参数）字段，将要保存至缓存的文件和存档包含在内，然后选择 Add（添加）。文件大小 (或存档文件中文件的总计大小) 必须小于分配的缓存大小。

   如果要将单个文件添加到分布式缓存，请指定 `-cacheFile`，后跟文件名称和位置、井号（#）以及将文件放入本地缓存时要为其指定的名称。以下示例说明了如何将单个文件添加到分布式缓存。

   ```
   -cacheFile \
   s3://DOC-EXAMPLE-BUCKET/file_name#cache_file_name
   ```

   如果要将存档文件添加到分布式缓存，请输入 `-cacheArchive`，后跟文件在 Amazon S3 中的位置、井号（#）以及您要为本地缓存中的文件集合指定的名称。以下示例说明了如何将存档文件添加到分布式缓存。

   ```
   -cacheArchive \
   s3://DOC-EXAMPLE-BUCKET/archive_name#cache_archive_name
   ```
6. 继续配置和启动您的集群。在处理任何集群步骤前，您的集群会将文件复制到缓存位置。

CLI

使用 Amazon CLI 指定分布式缓存文件

- 要在创建集群时提交 Streaming 步骤，请键入带 `create-cluster` 参数的 `--steps` 命令。要使用 Amazon CLI 指定分布式缓存文件，请在提交 Streaming 步骤时指定相应的参数。

  如果要将单个文件添加到分布式缓存，请指定 `-cacheFile`，后跟文件名称和位置、井号（#）以及将文件放入本地缓存时要为其指定的名称。

  如果要将存档文件添加到分布式缓存，请输入 `-cacheArchive`，后跟文件在 Amazon S3 中的位置、井号（#）以及您要为本地缓存中的文件集合指定的名称。以下示例说明了如何将存档文件添加到分布式缓存。

  有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

Example 1

键入以下命令启动集群并提交一个 Streaming 步骤，该步骤使用 `-cacheFile` 将文件 `sample_dataset_cached.dat` 添加到缓存。

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
 --instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
 program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py
 s3://my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py","-
input","s3://my_bucket/my_input","-output","s3://my_bucket/my_output", "-
cacheFile","s3://my_bucket/sample_dataset.dat#sample_dataset_cached.dat"]
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

如果您之前未创建默认 EMR 服务角色和 EC2 实例配置文件，请先键入 `aws emr create-default-roles` 创建它们，然后再键入 `create-cluster` 子命令。

Example 2

以下命令显示了流式集群的创建过程，并使用 `-cacheArchive` 将一份存档的文件添加到缓存。

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
 --instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
 program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py
 s3://my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py","-
input","s3://my_bucket/my_input","-output","s3://my_bucket/my_output", "-
cacheArchive","s3://my_bucket/sample_dataset.tgz#sample_dataset_cached"]
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

如果您之前未创建默认 EMR 服务角色和 EC2 实例配置文件，请先键入 `aws emr create-default-roles` 创建它们，然后再键入 `create-cluster` 子命令。

如何处理压缩文件

Hadoop 会检查文件扩展名以检测压缩文件。Hadoop 支持的压缩类型是：gzip、bzip2 和 LZO。不需要执行任何额外操作来解压缩使用这些压缩类型的文件；Hadoop 会为您执行这些操作。

要索引 LZO 文件，您可以使用 hadoop-lzo 库，可以从 https://github.com/kevinweil/hadoop-lzo 上下载该库。请注意，因为这是第三方库，所以 Amazon EMR 不会为开发人员提供有关如何使用此工具的支持。有关使用信息，请参阅 hadoop-lzo readme 文件。

将 DynamoDB 数据导入到 Hive

Amazon EMR 提供的 Hive 实施包括一项功能，您可以使用此功能在 DynamoDB 和 Amazon EMR 集群之间导入和导出数据。如果您的输入数据存储在 DynamoDB 中，这会非常有用。有关详细信息，请参阅使用 Amazon EMR 导出、导入、查询和连接 DynamoDB 中的表。

使用 Amazon Direct Connect 连接到数据

Amazon Direct Connect 是一种服务，您可以用它在数据中心、办公室或主机托管环境与 Amazon Web Services 之间建立私有专用网络连接。如果您有大量输入数据，那么使用 Amazon Direct Connect 可以降低网络成本、提高带宽吞吐量，并且比基于 Internet 的连接提供更加一致的网络体验。有关更多信息，请参阅《Amazon Direct Connect 用户指南》。

使用 Amazon Snowball 上载大量数据

通过使用 Amazon Snowball 服务，您可以用快于互联网的速度，在 Amazon Simple Storage Service（Amazon S3）和您的现场数据存储位置间传输大量数据。Snowball 支持两种任务类型：导入任务和导出任务。导入任务涉及将数据从本地部署源传输到 Amazon S3 存储桶。导出任务涉及将数

据从 Amazon S3 存储桶传输到本地部署源。对于这两种任务类型，Snowball 设备会保护您的数据安全，而区域承运商则在 Amazon S3 和您的现场数据存储位置之间传输这些数据。Snowball 设备坚固耐用，并由 Amazon Key Management Service（Amazon KMS）提供保护。有关更多信息，请参阅 Amazon Snowball Edge 开发人员指南。

配置输出位置

Amazon EMR 集群最常见的输出格式是压缩或未压缩的文本文件。通常，把这些文件写入 Amazon S3 存储桶。启动该集群前，必须先创建此存储桶。启动集群时，指定 S3 存储桶作为输出位置。

有关更多信息，请参阅以下主题：

主题

- 创建和配置 Amazon S3 存储桶
- Amazon EMR 可以返回什么格式？
- 如何将数据写入您并不拥有的 Amazon S3 存储桶
- 压缩集群的输出

创建和配置 Amazon S3 存储桶

Amazon EMR（Amazon EMR）使用 Amazon S3 存储输入数据、日志文件和输出数据。Amazon S3 将这些存储位置称为存储桶。存储桶具有某些限制以符合 Amazon S3 和 DNS 要求。有关更多信息，请转至《Amazon Simple Storage Service 开发人员指南》中的存储桶限制。

要创建 Amazon S3 存储桶，请按照《Amazon Simple Storage Service 开发人员指南》中的创建存储桶页面中的说明进行操作。

> **Note**
>
> 如果在 Create a Bucket (创建存储桶) 向导中启用了日志记录，则只启用存储桶访问日志，不会启用集群日志。

> **Note**
>
> 有关指定特定于区域的存储桶的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的存储桶和区域以及 Amazon SDK 的可用区域终端节点。

创建存储桶后，您可以为其设置适当的权限。一般情况下，您可以给自己（所有者）授予读取和写入权限。在配置存储桶时，强烈建议您关注Amazon S3 的安全最佳实践。

在创建集群之前，所需的 Amazon S3 存储桶必须存在。您必须将集群中引用的所有必要的脚本或数据上载至 Amazon S3。下表介绍了示例数据、脚本和日志文件位置。

信息	Amazon S3 中的示例位置
脚本或程序	s3://*DOC-EXAMPLE-BUCKET1* /script/MapperScript.py
日志文件	s3://*DOC-EXAMPLE-BUCKET1* /logs
输入数据	s3://*DOC-EXAMPLE-BUCKET1* /input
输出数据	s3://*DOC-EXAMPLE-BUCKET1* /output

Amazon EMR 可以返回什么格式？

集群的默认输出格式是将密钥和值对写入单独文本文件行的文本。这是最常用的输出格式。

如果默认输出数据需要采用默认文本文件以外的格式，则可以使用 Hadoop 接口 `OutputFormat` 指定其它输出类型。您甚至可以创建 `FileOutputFormat` 类型的子类，用于处理自定义数据类型。有关更多信息，请参阅 http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/OutputFormat.html。

如果您正在启动 Hive 集群，可以使用串行器/解串器 (SerDe) 将来自 HDFS 的数据以给定格式输出。有关更多信息，请参阅 https://cwiki.apache.org/confluence/display/Hive/SerDe。

如何将数据写入您并不拥有的 Amazon S3 存储桶

默认情况下，当您将文件写入到 Amazon Simple Storage Service（Amazon S3）存储桶时，您是唯一一个能够读取该文件的用户。假设您将文件写入到您自己的存储桶，此默认设置会保护您的文件隐私。

然而，如果您正在运行集群，并且要让输出写入到另一个Amazon用户的 Amazon S3 存储桶以及要让其它Amazon用户能够读取此输出，您必须做两件事情：

- 让其它Amazon用户授予您写入他们的 Amazon S3 存储桶的权限。您启动的集群会在Amazon凭证下运行，因此您启动的任何集群也将能够写入其它Amazon用户的存储桶。
- 为其它Amazon用户设置文件（您或集群可以写入 Amazon S3 存储桶）读取权限。设置这些读取权限的最简单方法是使用预装访问控制列表（ACL），这是一组由 Amazon S3 预先定义的访问策略。

有关其他Amazon用户可以如何授予您向其他用户的 Amazon S3 存储桶写入文件的权限，请参阅《Amazon Simple Storage Service 用户指南》中的编辑存储桶权限。

对于在将文件写入到 Amazon S3 时使用预装 ACL 的集群，将 `fs.s3.canned.acl` 集群配置选项设置为预装 ACL 后再使用。下表列出了当前定义的预装 ACL。

标准 ACL	描述
`AuthenticatedRead`	指定为所有者授予 `Permission.FullControl` 访问权限及为 `GroupGrantee.AuthenticatedUsers` 组被授权者授予 `Permission.Read` 访问权限。
`BucketOwnerFullControl`	指定为存储桶的所有者授予 `Permission.FullControl` 访问权限。存储桶的所有者不必与对象的所有者相同。
`BucketOwnerRead`	指定为存储桶的所有者授予 `Permission.Read` 访问权限。存储桶的所有者不必与对象的所有者相同。
`LogDeliveryWrite`	指定为所有者授予 `Permission.FullControl` 访问权限及为 `GroupGrantee.LogDelivery` 组被授权者授予 `Permission.Write` 访问权限，以便能够传递访问日志。
`Private`	指定为所有者授予 `Permission.FullControl` 访问权限。
`PublicRead`	指定为所有者授予 `Permission.FullControl` 访问权限及为 `GroupGrantee.AllUsers` 组被授权者授予 `Permission.Read` 访问权限。
`PublicReadWrite`	指定为所有者授予 `Permission.FullControl` 访问权限及为 `GroupGrantee.AllUsers` 组被授权者授予 `Permission.Read` 和 `Permission.Write` 访问权限。

根据您正在运行的集群类型，有多种方法可设置集群配置选项。以下步骤介绍了如何设置常见案例的选项。

在 Hive 中使用预装 ACL 写文件

- 通过 Hive 命令提示符，将 `fs.s3.canned.acl` 配置选项设置为您想在其写入 Amazon S3 的文件上设置集群的预装 ACL。要访问 Hive 命令提示符，可通过 SSH 连接到主节点，然后在 Hadoop 命令提示符处键入 Hive。有关更多信息，请参阅使用 SSH 连接到主节点。

  以下示例将 `fs.s3.canned.acl` 配置选项设置为 `BucketOwnerFullControl`，这可给予 Amazon S3 存储桶的所有者对该文件的完全控制权。请注意，设置命令区分大小写，且不含引号或空格。

```
hive> set fs.s3.canned.acl=BucketOwnerFullControl;
create table acl (n int) location 's3://acltestbucket/acl/';
insert overwrite table acl select count(*) from acl;
```

  示例的最后两行创建了一个存储在 Amazon S3 中的表格，并向该表格中写入数据。

在 Pig 中使用预装 ACL 写文件

- 通过 Pig 命令提示符，将 `fs.s3.canned.acl` 配置选项设置为您想在其写入 Amazon S3 的文件上设置集群的预装 ACL。要访问 Pig 命令提示符，可通过 SSH 连接到主节点，然后在 Hadoop 命令提示符处键入 Pig。有关更多信息，请参阅使用 SSH 连接到主节点。

  以下示例将 `fs.s3.canned.acl` 配置选项设置为 BucketOwnerFullControl，这可给予 Amazon S3 存储桶的所有者对该文件的完全控制权。请注意，设置命令包含一个位于预装 ACL 名称前面的空格，并且不包含引号。

```
pig> set fs.s3.canned.acl BucketOwnerFullControl;
store some data into 's3://acltestbucket/pig/acl';
```

在 自定义 JAR 中使用预装 ACL 写文件

- 利用带有 -D 标志的 Hadoop 设置 `fs.s3.canned.acl` 配置选项。如以下示例所示。

```
hadoop jar hadoop-examples.jar wordcount
```

```
-Dfs.s3.canned.acl=BucketOwnerFullControl s3://mybucket/input s3://mybucket/output
```

压缩集群的输出

主题

- 输出数据压缩
- 中间数据压缩
- 在 Amazon EMR 中使用 Snappy 库

输出数据压缩

这会压缩 Hadoop 作业的输出。如果您使用的是 TextOutputFormat，返回结果是 gzip 处理后的文本文件。如果您向 SequenceFiles 写入，那么，返回结果是内部压缩的 SequenceFile。将配置设置 mapred.output.compress 设置为 True，可启用该功能。

如果您运行的是流式作业，则可以将这些参数传递给流式作业的方式启用该功能。

```
-jobconf mapred.output.compress=true
```

您还可以使用引导操作，自动压缩所有的作业输出。此处介绍如何使用 Ruby 客户端做到这点。

```
--bootstrap-actions s3://elasticmapreduce/bootstrap-actions/configure-hadoop \
--args "-s,mapred.output.compress=true"
```

最后，如果写入自定义 JAR，可以在创建作业时使用以下行启用输出压缩。

```
FileOutputFormat.setCompressOutput(conf, true);
```

中间数据压缩

如果您的作业有海量的数据在映射器会 Reducer 之间来回发送，那么启用中间压缩可以提高性能。压缩映射输出，并在其到达核心节点时进行解压缩。配置设置是 mapred.compress.map.output。您同样可以对输出压缩启用这个功能。

当写入自定义 JAR 时，使用下列命令：

```
conf.setCompressMapOutput(true);
```

在 Amazon EMR 中使用 Snappy 库

Snappy 是一种压缩和解压缩库，已为提高速度而进行了优化。它在 Amazon EMR AMI 版本 2.0 实例和更高版本中提供，用作中间压缩的默认值。有关 Snappy 库的详细信息，请转到 http://code.google.com/p/snappy/。

计划和配置主节点

启动 Amazon EMR 集群时，可以选择在集群中设置一个或三个主节点。只有 Amazon EMR 版本 5.23.0 及更高版本支持启动具有三个主节点的集群。Amazon EMR 可以利用 EC2 置放群组确保将主节点放置在不同的基础硬件上，从而进一步提高集群可用性。有关更多信息，请参阅Amazon EMR 与 EC2 置放群组的集成。

具有多个主节点的 Amazon EMR 集群提供以下主要优势：

- 主节点不再发生单点故障。如果其中一个主节点发生故障，集群将使用其他两个主节点，不会中断运行。同时，Amazon EMR 会自动使用预置了相同配置和引导操作的新主节点替换发生故障的主节点。
- Amazon EMR 启用 HDFS NameNode 和 YARN ResourceManager 的 Hadoop 高可用性功能，并支持其他一些开源应用程序的高可用性。

  有关具有多个主节点的 Amazon EMR 集群如何支持开源应用程序和其他 Amazon EMR 功能的详细信息，请参阅 支持的应用程序和功能。

> **Note**
>
> 集群只能驻留在一个可用区或子网中。

本部分提供有关具有多个主节点的 Amazon EMR 集群支持的应用程序和功能的信息，以及启动集群的配置详细信息、最佳实践和注意事项。

主题

- 支持的应用程序和功能
- 启动具有多个主节点的 Amazon EMR 集群
- Amazon EMR 与 EC2 置放群组的集成
- 注意事项和最佳实践

支持的应用程序和功能

本主题提供有关 Amazon EMR 集群中 HDFS NameNode 和 YARN ResourceManager 的 Hadoop 高可用性功能的信息，以及高可用性功能如何与开源应用程序和其他 Amazon EMR 功能配合使用。

高可用性 HDFS

具有多个主节点的 Amazon EMR 集群在 Hadoop 中启用 HDFS NameNode 高可用性功能。有关更多信息，请参阅 HDFS 高可用性。

在 Amazon EMR 集群中，两个或更多个单独的节点配置为 NameNodes。一个 NameNode 处于 `active` 状态，另一个则处于 `standby` 状态。如果具有 `active` NameNode 的节点发生故障，Amazon EMR 将会启动自动 HDFS 故障转移过程。具有 `standby` NameNode 的节点将变成 `active` 并接管集群中的所有客户端操作。Amazon EMR 用新的主节点替换故障的主节点，新主节点将作为 `standby` 重新加入。

> **Note**
>
> 在 Amazon EMR 版本 5.23.0 到 5.30.1 中，三个主节点中只有两个运行 HDFS NameNode。

如果需要找出哪个 NameNode 是 `active`，可以使用 SSH 连接到集群中的任何主节点并运行以下命令：

```
hdfs haadmin -getAllServiceState
```

输出列出安装了 NameNode 的节点及其状态。例如：

```
ip-##-#-#-##1.ec2.internal:8020 active
```

```
ip-##-#-#-##2.ec2.internal:8020 standby
ip-##-#-#-##3.ec2.internal:8020 standby
```

高可用性 YARN ResourceManager

具有多个主节点的 Amazon EMR 集群在 Hadoop 中启用 YARN ResourceManager 高可用性功能。有关更多信息，请参阅 ResourceManager 高可用性。

在具有多个主节点的 Amazon EMR 集群中，YARN ResourceManager 在所有三个主节点上运行。一个 ResourceManager 处于 `active` 状态，另两个处于 `standby` 状态。如果具有 `active` ResourceManager 的主节点发生故障，Amazon EMR 将启动自动失效转移过程。具有 `standby` ResourceManager 的主节点会接管所有操作。Amazon EMR 用新的主节点替换故障的主节点，新节点将作为 `standby` 重新加入 ResourceManager 仲裁。

您可以针对任何主节点连接到“http://*master-public-dns-name*:8088/cluster”，它会自动将您引导至 `active` 资源管理器。要找出哪个资源管理器是 `active`，请使用 SSH 连接到集群中的任何主节点。然后运行以下命令以获取三个主节点及其状态的列表：

```
yarn rmadmin -getAllServiceState
```

具有多个主节点的 Amazon EMR 集群中支持的应用程序

您可以在具有多个主节点的 Amazon EMR 集群上安装和运行以下应用程序。对于每个应用程序，主节点失效转移过程各不相同。

应用程序	主节点失效转移期间的可用性	注意
Flink	可用性不受主节点失效转移的影响	在 Amazon EMR 上作为 YARN 应用程序运行的 Flink 任务。Flink 的 JobManager 在核心节点上作为 YARN 的 ApplicationMaster 运行。JobManager 不受主节点失效转移过程的影响。 如果您使用 Amazon EMR 版本 5.27.0 或更早版本，则 JobManager 是单点故障。当 JobManager 失败时，它会失去所有作业状态，并且不会恢复正在运行的作业。通过配置应用程序尝试计数、开展检查点检验并启用 ZooKeeper 作为 Flink 的状态存储，您可以启

应用程序	主节点失效转移期间的可用性	注意
		用 JobManager 高可用性。有关更多信息，请参阅在包括多个主节点的 Amazon EMR 集群中配置 Flink。 从 Amazon EMR 版本 5.28.0 开始，无需手动配置即可启用 JobManager 高可用性。
Ganglia	可用性不受主节点失效转移的影响	Ganglia 在所有主节点上可用，因此 Ganglia 可以在主节点失效转移过程中继续运行。
Hadoop	高可用性	当处于活动状态的主节点发生故障时，HDFS NameNode 和 YARN ResourceManager 会自动故障转移到备用节点。
HBase	高可用性	当处于活动状态的主节点发生故障时，HBase 会自动故障转移到备用节点。 如果通过 REST 或 Thrift 服务器连接到 HBase，则必须在处于活动状态的主节点发生故障时切换到其他主节点。
HCatalog	可用性不受主节点失效转移的影响	HCatalog 是在 Hive 元数据仓上构建的，而 Hive 元数据仓存在于集群之外。在主节点失效转移过程中，HCatalog 仍然可用。
JupyterHub	高可用性	JupyterHub 安装在所有三个主实例上。强烈建议配置笔记本电脑持久性，以防止在出现主节点故障时丢失笔记本。有关更多信息，请参阅在 Amazon S3 中为笔记本配置持久性。
Livy	高可用性	Livy 安装在所有三个主节点上。当处于活动状态的主节点发生故障时，您将无法访问当前的 Livy 会话，并需要在不同的主节点或新的替换节点上创建新的 Livy 会话。

应用程序	主节点失效转移期间的可用性	注意
Mahout	可用性不受主节点失效转移的影响	由于 Mahout 没有进程守护程序，因此不受主节点失效转移过程的影响。
MXNet	可用性不受主节点失效转移的影响	由于 MXNet 没有进程守护程序，因此不受主节点失效转移过程的影响。
Phoenix	高可用性	Phoenix 的查询服务器仅在三个主节点之一上运行。所有三个主节点上的 Phoenix 都配置为连接到 Phoenix 查询服务器。您可以通过使用 `/etc/phoenix/conf/phoenix-env.sh` 文件找到 Phoenix 查询服务器的私有 IP。
Pig	可用性不受主节点失效转移的影响	由于 Pig 没有进程守护程序，因此不受主节点失效转移过程的影响。
Spark	高可用性	所有 Spark 应用程序都在 YARN 容器中运行，并且可以像高可用性 YARN 功能一样对主节点失效转移做出反应。
Sqoop	高可用性	默认情况下，sqoop-job 和 sqoop-metastore 将数据（作业描述）存储在运行该命令的主节点的本地磁盘上，如果您想在外部数据库上保存元数据仓数据，请参阅 Apache Sqoop 文档
Tez	高可用性	由于 Tez 容器在 YARN 上运行，因此在主节点失效转移过程中，Tez 的行为与 YARN 相同。
TensorFlow	可用性不受主节点失效转移的影响	由于 TensorFlow 没有进程守护程序，因此不受主节点失效转移过程的影响。

应用程序	主节点失效转移期间的可用性	注意
Zeppelin	高可用性	Zeppelin 安装在所有三个主节点上。Zeppelin 默认将注释和解释器配置存储在 HDFS 中，以防止数据丢失。解释器会话在所有三个主实例间完全隔离。发生主节点故障时，会话数据将丢失。建议不要在不同的主实例上同时修改同一注释。
ZooKeeper	高可用性	ZooKeeper 是 HDFS 自动故障转移功能的基础。ZooKeeper 提供高可用性服务，用于维护协调数据、通知客户端数据的更改以及监控客户端的故障。有关更多信息，请参阅 HDFS 自动故障转移。

要在具有多个主节点的 Amazon EMR 集群中运行以下应用程序，您必须配置外部数据库。外部数据库位于集群外部，在主节点失效转移过程中能够永久保存数据。对于以下应用程序，服务组件将在主节点失效转移过程中自动恢复，但活动任务可能会失败，需要重试。

应用程序	主节点失效转移期间的可用性	注意
Hive	仅适用于服务组件的高可用性	Hive 的外部元数据仓是必需的。这必须是一个 MySQL 外部元数据仓，因为 PostgreSQL 不支持多主集群。有关更多信息，请参阅为 Hive 配置外部元存储。
Hue	仅适用于服务组件的高可用性	Hue 的外部数据库是必需的。有关更多信息，请参阅在 Amazon RDS 中将 Hue 与远程数据库结合使用。
Oozie	仅适用于服务组件的高可用性	Oozie 的外部数据库是必需的。有关更多信息，请参阅在 Amazon RDS 中将 Oozie 与远程数据库结合使用。

应用程序	主节点失效转移期间的可用性	注意
		Oozie-server 和 oozie-client 安装在所有三个主节点上。默认情况下，oozie 客户端配置为连接到正确的 oozie 服务器。
PrestoDB 或 PrestoSQL/ Trino	仅适用于服务组件的高可用性	PrestoDB（或 Amazon EMR 6.1.0 至 6.3.0 版本的 PrestoSQL 或 Amazon EMR 6.4.0 以及跟高版本的 Trino）的外部 Hive 元数据是必需的。您可以将 Presto 与 Amazon Glue 数据目录结合使用，也可以使用适用于 Hive 的外部 MySQL 数据库。 由于 Presto CLI 安装在所有三个主节点上，因此您可以使用它从任何主节点访问 Presto 协调器。Presto 协调器仅安装在一个主节点上。您可以通过调用 Amazon EMR `describe-cluster` API 并读取响应中 `MasterPublicDnsName` 字段的返回值，以查找安装有 Presto 协调器的主节点的 DNS 名称。

> **Note**
>
> 当主节点发生故障时，Java 数据库连接（JDBC）或开放式数据库连接（ODBC）将终止其与主节点的连接。您可以连接到任何剩余的主节点以继续工作，因为 Hive 元存储进程守护程序在所有主节点上运行。或者您可以等待故障的主节点被替换。

Amazon EMR 功能如何在具有多个主节点的集群中工作

使用 SSH 连接到主节点

您可以采用与连接到单个主节点相同的方式，使用 SSH 连接到 Amazon EMR 集群中三个主节点中的任何一个。有关更多信息，请参阅使用 SSH 连接到主节点。

如果主节点发生故障，与该主节点的 SSH 连接将终止。要继续工作，您可以连接到另外两个主节点之一。或者，您可以在 Amazon EMR 用新的主节点替换故障的节点后访问新的主节点。

> **Note**
>
> 替换主节点的私有 IP 地址与前一个相同。替换主节点的公有 IP 地址可能会更改。您可以在控制台中或使用 Amazon CLI 中的 `describe-cluster` 命令检索新的 IP 地址。
> NameNode 仅在两个主节点上运行。但是，您可以运行 `hdfs` CLI 命令并操作任务以访问所有三个主节点上的 HDFS。

在具有多个主节点的 Amazon EMR 集群中的处理步骤

您可以采用与具有单个主节点的集群中的步骤相同的方式，向具有多个主节点的 Amazon EMR 集群提交步骤。有关更多信息，请参阅向集群提交工作。

以下是在具有多个主节点的 Amazon EMR 集群中处理步骤的注意事项：

- 如果主节点发生故障，主节点上运行的步骤将标记为 FAILED。任何本地写入的数据都将丢失。但是，状态 FAILED 可能无法反映步骤的实际状态。
- 如果在主节点发生故障时一个正在运行的步骤已启动 YARN 应用程序，由于主节点的自动失效转移，该步骤可以继续并成功。
- 建议您通过参考作业的输出来检查步骤的状态。例如，MapReduce 作业使用 `_SUCCESS` 文件来确定作业是否成功完成。
- 建议您将 ActionOnFailure 参数设置为 CONTINUE 或 CANCEL_AND_WAIT，而不是 TERMINATE_JOB_FLOW 或 TERMINATE_CLUSTER。

自动终止保护

Amazon EMR 会为具有多个主节点的所有集群自动启用终止保护，并覆盖您在创建集群时应用的所有步骤执行设置。您可以在集群启动后禁用终止保护。请参阅为正在运行的集群配置终止保护。要关闭具有多个主节点的集群，您必须先修改集群属性，才能禁用终止保护。有关说明，请参阅 终止具有多个主节点的 Amazon EMR 集群。

有关终止保护的更多信息，请参阅使用终止保护。

具有多个主节点的 Amazon EMR 集群中不支持的功能

具有多个主节点的 Amazon EMR 集群目前不支持以下 Amazon EMR 功能：

- EMR Notebooks

- 实例集
- 一键访问持久性 Spark 历史记录服务器
- 持久性应用程序用户界面
- 目前，具有多个主节点的 Amazon EMR 集群或与 Amazon Lake Formation 集成的 Amazon EMR 集群不支持一键式访问持久性应用程序用户界面。

> **Note**
>
> 要在集群中使用 Kerberos 身份验证，您必须配置外部 KDC。
> 从 Amazon EMR 版本 5.27.0 开始，您可以在具有多个主节点的 Amazon EMR 集群上配置 HDFS 透明加密。有关更多信息，请参阅 Amazon EMR 上的 HDFS 中的透明加密。

启动具有多个主节点的 Amazon EMR 集群

本主题提供了启动具有多个主节点的 Amazon EMR 集群的配置详细信息和示例。

> **Note**
>
> Amazon EMR 会为具有多个主节点的所有集群自动启用终止保护，并覆盖您在创建集群时应用的所有自动终止设置。要关闭具有多个主节点的集群，您必须先修改集群属性，才能禁用终止保护。有关说明，请参阅 终止具有多个主节点的 Amazon EMR 集群。

先决条件

- 您可以同时在公有 VPC 子网和私有 VPC 子网中启动具有多个主节点的 Amazon EMR 集群。不支持 EC2-Classic。要在公有子网中启动具有多个主节点的 Amazon EMR 集群，必须通过在控制台中选择 Auto-assign IPv4（自动分配 IPv4）或运行以下命令来启用此子网中的实例，以接收公有 IP 地址。将 *22XXXX01* 替换为您的子网 ID。

```
aws ec2 modify-subnet-attribute --subnet-id subnet-22XXXX01 --map-public-ip-on-launch
```

- 要在具有多个主节点的 Amazon EMR 集群上运行 Hive、Hue 或 Oozie，您必须创建外部元存储。有关更多信息，请参阅为 Hive 配置外部元存储、在 Amazon RDS 中将 Hue 与远程数据库结合使用或者 Apache Oozie。

- 要在集群中使用 Kerberos 身份验证，您必须配置外部 KDC。有关更多信息，请参阅[在 Amazon Amazon EMR 上配置 Kerberos]。

启动具有多个主节点的 Amazon EMR 集群

启动具有多个主节点的 Amazon EMR 集群时，必须为主节点实例组指定实例计数值 3。以下示例演示了如何使用默认 AMI 或自定义 AMI 启动集群。

> **Note**
>
> 使用 Amazon CLI 启动具有多个主节点的 Amazon EMR 集群时，必须指定子网 ID。在以下示例中将 *22XXXX01* 替换为您的子网 ID。

Example – 使用默认 AMI 启动具有多个主节点的 Amazon EMR 集群

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-5.36.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

Example – 使用自定义 AMI 启动具有多个主节点的 Amazon EMR 集群

```
aws emr create-cluster \
--name "custom-ami-ha-cluster" \
--release-label emr-5.36.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID
```

Example – 通过外部 Hive 元存储启动具有多个主节点的 Amazon EMR 集群

要在具有多个主节点的 Amazon EMR 集群上运行 Hive，您必须为 Hive 指定外部元存储，如下例所示。

1. 创建一个包含 Hive 元数据仓凭证的临时 hiveConfiguration.json 文件。

```
[
  {
    "Classification": "hive-site",
    "Properties": {
      "javax.jdo.option.ConnectionURL": "jdbc:mysql:\/\/hostname:3306\/hive?
createDatabaseIfNotExist=true",
      "javax.jdo.option.ConnectionDriverName": "org.mariadb.jdbc.Driver",
      "javax.jdo.option.ConnectionUserName": "username",
      "javax.jdo.option.ConnectionPassword": "password"
    }
  }
]
```

2. 启动具有 Hive 元数据仓的集群。

```
aws emr create-cluster \
--name "ha-cluster-with-hive-metastore" \
--release-label emr-5.36.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-
22XXXX01 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name= Spark Name=Hive \
--configurations ./hiveConfiguration.json
```

终止具有多个主节点的 Amazon EMR 集群

要终止具有多个主节点的 Amazon EMR 集群，必须在终止集群之前禁用终止保护，如下例所示。将 *j-3KVTXXXXXX7UG* 替换为您的集群 ID。

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-
protected
```

```
aws emr terminate-clusters --cluster-id j-3KVTXXXXXX7UG
```

Amazon EMR 与 EC2 置放群组的集成

在 Amazon EC2 上启动 Amazon EMR 多主节点集群时，您可以选择使用置放群组策略，指定您希望部署的主节点实例，以防止硬件故障。

从 Amazon EMR 版本 5.23.0 开始支持置放群组策略，该策略可作为多主节点集群选项。目前，置放群组策略仅支持主节点类型，并会将 SPREAD 策略应用于这些主节点。SPREAD 策略将一小组实例放置在单独的基础硬件上，以防止发生硬件故障时出现多个主节点丢失的问题。请注意：如果没有足够的唯一硬件来满足请求，实例启动请求可能会失败。有关 EC2 置放策略和限制的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的置放群组。

Amazon EC2 的初始限制是每个 Amazon 区域可以启动 500 个已启用置放群组策略的集群。联系 Amazon 支持，以请求增加允许置放群组的数量。通过跟踪 Amazon EMR 与 Amazon EMR 置放组策略关联的键值对，可以识别 Amazon EMR 创建的 EC2 置放组 EMR。有关 EC2 集群实例类型的更多信息，请参阅查看 Amazon EC2 中的集群实例。

将置放群组托管策略附加到 Amazon EMR 角色

置放群组策略需要一个名为 `AmazonElasticMapReducePlacementGroupPolicy` 的托管式策略，以允许 Amazon EMR 在 Amazon EC2 上创建、删除和描述置放群组。您必须将 `AmazonElasticMapReducePlacementGroupPolicy` 附加到 Amazon EMR 的服务角色，然后才能启动 Amazon EMR 多主节点集群。

您还可以将 `AmazonEMRServicePolicy_v2` 托管式策略 附加到 Amazon EMR 服务角色，而不使用置放群组托管式策略。`AmazonEMRServicePolicy_v2` 会与 `AmazonElasticMapReducePlacementGroupPolicy` 一样授予访问 Amazon EC2 上置放群组的相同权限。有关更多信息，请参阅Amazon EMR 的服务角色（EMR 角色）。

`AmazonElasticMapReducePlacementGroupPolicy` 托管式策略是由 Amazon EMR 创建和管理的以下 JSON 文本。

> **Note**
>
> 由于 `AmazonElasticMapReducePlacementGroupPolicy` 托管式策略是自动更新的，因此，此处显示的策略可能已过期。使用 Amazon 管理控制台查看当前的策略。

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Resource":"*",
         "Effect":"Allow",
         "Action":[
            "ec2:DeletePlacementGroup",
            "ec2:DescribePlacementGroups"
         ]
      },
      {
         "Resource":"arn:aws:ec2:*:*:placement-group/pg-*",
         "Effect":"Allow",
         "Action":[
            "ec2:CreatePlacementGroup"
         ]
      }
   ]
}
```

使用置放群组策略启动 Amazon EMR 多主节点集群

要使用置放群组策略启动 Amazon EMR 多主节点集群，请将置放群组托管策略 `AmazonElasticMapReducePlacementGroupPolicy` 附加到 Amazon EMR 角色。有关更多信息，请参阅[将置放群组托管策略附加到 Amazon EMR 角色](#)。

每次使用此角色启动 Amazon EMR 多主节点集群时，Amazon EMR 都会尝试启动将 SPREAD 策略应用于其主节点的集群。如果您使用的角色未将置放群组托管式策略 `AmazonElasticMapReducePlacementGroupPolicy` 附加到自身，Amazon EMR 会尝试在没有置放群组策略的情况下启动 Amazon EMR 多主节点集群。

如果您使用 EMR API 或 CLI 通过 `placement-group-configs` 参数来启动 Amazon EMR 多主节点集群，则 Amazon EMR 仅在 Amazon EMR 角色已附加置放群组托管策略 `AmazonElasticMapReducePlacementGroupPolicy` 时启动集群。如果 Amazon EMR 角色没有附加策略，则 Amazon EMR 多主节点集群将启动失败。

Example – 使用 Amazon EMR API 启动已附加置放群组策略的 Amazon EMR 多主节点集群。

使用 RunJobFlow 操作创建 Amazon EMR 多主节点集群时，请将 `PlacementGroupConfigs` 属性进行如下设置。目前，`MASTER` 实例角色会自动使用 `SPREAD` 作为置放群组策略。

```
{
```

```
   "Name":"ha-cluster",
   "PlacementGroupConfigs":[
      {
         "InstanceRole":"MASTER"
      }
   ],
   "ReleaseLabel":"emr-5.30.1",
   "Instances":{
      "ec2SubnetId":"subnet-22XXXX01",
      "ec2KeyName":"ec2_key_pair_name",
      "InstanceGroups":[
         {
            "InstanceCount":3,
            "InstanceRole":"MASTER",
            "InstanceType":"m5.xlarge"
         },
         {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
         }
      ]
   },
   "JobFlowRole":"EMR_EC2_DefaultRole",
   "ServiceRole":"EMR_DefaultRole"
}
```

- 将 *ha-cluster* 替换为您的高可用性集群名称。
- 将 *subnet-22XXXX01* 替换为您的子网 ID。
- 将 *ec2_key_pair_name* 替换为您用于此集群的 EC2 密钥对的名称。EC2 密钥对是可选的，仅当您希望使用 SSH 访问集群时是必需的。

Example – 使用 Amazon EMR CLI 启动已附加置放群组策略的多主节点集群。

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER \
--release-label emr-5.30.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
```

```
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

- 将 *ha-cluster* 替换为您的高可用性集群名称。
- 将 *subnet-22XXXX01* 替换为您的子网 ID。
- 将 *ec2_key_pair_name* 替换为您用于此集群的 EC2 密钥对的名称。EC2 密钥对是可选的，仅当您希望使用 SSH 访问集群时是必需的。

在未附加置放群组策略的情况下启动多主节点集群

要让多主节点集群在未附加置放群组策略的情况下启动主节点，您需要执行以下操作之一：

- 从 Amazon EMR 角色中删除置放群组托管策略 `AmazonElasticMapReducePlacementGroupPolicy`，或
- 使用 Amazon EMR API 或 CLI 通过 `placement-group-configs` 参数启动多主节点集群，选择 `NONE` 作为置放群组策略。

Example – 使用 Amazon EMR API 启动未附加置放群组策略的多主节点集群。

使用 RunJobFlow 操作创建多主节点集群时，请将 `PlacementGroupConfigs` 属性进行如下设置。

```
{
   "Name":"ha-cluster",
   "PlacementGroupConfigs":[
      {
         "InstanceRole":"MASTER",
         "PlacementStrategy":"NONE"
      }
   ],
   "ReleaseLabel":"emr-5.30.1",
   "Instances":{
      "ec2SubnetId":"subnet-22XXXX01",
      "ec2KeyName":"ec2_key_pair_name",
      "InstanceGroups":[
         {
            "InstanceCount":3,
```

```
            "InstanceRole":"MASTER",
            "InstanceType":"m5.xlarge"
         },
         {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
         }
      ]
   },
   "JobFlowRole":"EMR_EC2_DefaultRole",
   "ServiceRole":"EMR_DefaultRole"
}
```

- 将 *ha-cluster* 替换为您的高可用性集群名称。
- 将 *subnet-22XXXX01* 替换为您的子网 ID。
- 将 *ec2_key_pair_name* 替换为您用于此集群的 EC2 密钥对的名称。EC2 密钥对是可选的，仅当您希望使用 SSH 访问集群时是必需的。

Example – 使用 Amazon EMR CLI 启动未附加置放群组策略的多主节点集群。

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER,PlacementStrategy=NONE \
--release-label emr-5.30.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

- 将 *ha-cluster* 替换为您的高可用性集群名称。
- 将 *subnet-22XXXX01* 替换为您的子网 ID。
- 将 *ec2_key_pair_name* 替换为您用于此集群的 EC2 密钥对的名称。EC2 密钥对是可选的，仅当您希望使用 SSH 访问集群时是必需的。

检查附加多主节点集群的置放群组策略配置

您可以使用 Amazon EMR 描述集群 API，来查看附加多主节点集群的置放群组策略配置。

Example

```
aws emr describe-cluster --cluster-id "j-xxxxx"
{
   "Cluster":{
      "Id":"j-xxxxx",
      ...
      ...
      "PlacementGroups":[
         {
            "InstanceRole":"MASTER",
            "PlacementStrategy":"SPREAD"
         }
      ]
   }
}
```

注意事项和最佳实践

具有多个主节点的 Amazon EMR 集群的限制：

- 您无法将 Amazon EMR 集群与具有实例集的多个主节点结合使用。有关适用于多个主节点的 Amazon EMR 功能的更多信息，请参阅 支持的应用程序和功能。
- 如果任意两个主节点同时发生故障，Amazon EMR 将无法恢复集群。
- 具有多个主节点的 Amazon EMR 集群无法容忍可用区故障。在可用区中断的情况下，您将无法访问 EMR 集群。
- 除了 具有多个主节点的 Amazon EMR 集群中支持的应用程序 中指定的开源应用程序之外，Amazon EMR 不保证其他开源应用程序的高可用性功能。
- 在 Amazon EMR 版本 5.23.0 到 5.30.1 中，三个主节点中只有两个运行 HDFS NameNode。

配置子网的注意事项：

- 具有多个主节点的 Amazon EMR 集群仅可驻留在一个可用区或子网中。如果在发生失效转移时子网被充分利用或超额订阅，则 Amazon EMR 无法替换发生故障的主节点。为避免出现这种情况，建议您将整个子网专用于 Amazon EMR 集群。此外，请确保子网中有足够的私有 IP 地址。

配置核心节点的注意事项：

- 为确保核心节点实例组也具有高可用性，我们建议您至少启动四个核心节点。如果您决定启动具有三个或更少核心节点的较小集群，请通过将 `dfs.replication parameter` 设置为至少 2 以使 HDFS 具有足够的 DFS 复制。有关更多信息，请参阅 HDFS 配置。

> **Warning**
>
> 1. 如果单个节点出现故障，则在少于四个节点的集群上将 `dfs.replication` 设置为 1 可能会导致 HDFS 数据丢失。建议您使用具有至少四个核心节点的集群来处理生产工作负载。
> 2. Amazon EMR 不允许集群扩展 `dfs.replication` 下方的核心节点。例如，如果是 `dfs.replication = 2`，则最小核心节点数为 2。
> 3. 当您使用托管扩缩、自动扩缩或选择手动调整集群大小时，建议您将设置 `dfs.replication` 为 2 或更高。

在指标上设置警报的注意事项：

- Amazon EMR 不提供有关 HDFS 或 YARN 的应用程序特定指标。我们建议您设置告警以监控主节点实例计数。使用以下 Amazon CloudWatch 指标来配置警报：`MultiMasterInstanceGroupNodesRunning`、`MultiMasterInstanceGroupNodesRunningPe` 或 `MultiMasterInstanceGroupNodesRequested`。如果主节点发生故障和替换，CloudWatch 将通知您。
  - 如果 `MultiMasterInstanceGroupNodesRunningPercentage` 低于 1.0 和大于 0.5，集群可能会丢失主节点。在这种情况下，Amazon EMR 将尝试替换主节点。
  - 如果 `MultiMasterInstanceGroupNodesRunningPercentage` 低于 0.5，两个主节点可能会发生故障。在这种情况下，仲裁丢失，集群无法恢复。您必须手动将数据迁出此集群。

  有关更多信息，请参阅在指标上设置警报。

Amazon Outposts 上的 EMR 集群

从 Amazon EMR 版本 5.28.0 开始，您可以在 Amazon Outposts 上创建和运行 EMR 集群。Amazon Outposts 启用本地设施中的本地Amazon服务、基础设施和操作模型。在 Amazon Outposts 环境中，您可以使用与Amazon云中相同的 Amazon API、工具和基础设施。Amazon Outposts 上的 Amazon

EMR 是低延迟工作负载的理想选择，这些工作负载需要在靠近本地部署数据和应用程序的地方运行。有关 Amazon Outposts 的更多信息，请参阅 Amazon Outposts 用户指南。

先决条件

以下是使用 Amazon Outposts 上的 Amazon EMR 的先决条件：

- 您必须在本地数据中心中安装并配置了 Amazon Outposts。
- 您的 Outpost 环境和Amazon区域之间必须具有可靠的网络连接。
- 您必须具有足够的容量来支持 Outpost 中提供的 EMR 支持的实例类型。

限制

以下是 Amazon Outposts 上使用 Amazon EMR 的限制：

- 按需实例是 Amazon EC2 实例唯一支持的选项。Spot 实例不适用于 Amazon Outposts 上的 Amazon EMR。
- 如果您需要额外的 Amazon EBS 存储卷，则仅支持通用型 SSD（GP2）。
- 将对象存储在您指定的 Amazon Web Services 区域中的 S3 存储桶，是 Amazon EMR on Outposts 唯一支持的 S3 选项。对于 Amazon EMR on Amazon Outposts，S3 on Outposts 不受支持。
- Amazon Outposts 上的 Amazon EMR 仅支持以下实例类型：

实例类	实例类型
通用型	`m5.xlarge |m5.2xlarge | m5.4xlarge |m5.12xlarge | m5.24xlarge |m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge |m5d.24xlarge`
计算优化	`c5.xlarge |c5.2xlarge | c5.4xlarge |c5.18xlarge | c5d.xlarge |c5d.2xlarge | c5d.4xlarge | c5d.18xlarge`
内存优化	`r5.xlarge |r5.2xlarge | r5.4xlarge |r5.12xlarge | r5d.xlarge |r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge`

实例类	实例类型
存储优化	`i3en.xlarge |i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge |i3en.24xlarge`

网络连接注意事项

- 如果 Outpost 与其Amazon区域之间的网络连接断开，集群将继续运行。但是，在恢复连接之前，您无法创建新集群或对现有集群执行新操作。如果实例出现故障，则不会自动替换该实例。此外，向正在运行的集群添加步骤、检查步骤执行状态以及发送 CloudWatch 指标和事件等操作将会延迟。
- 建议在您的 Outpost 与Amazon区域之间提供可靠且高度可用的网络连接。如果您的 Outpost 与其Amazon区域之间的网络连接断开了若干小时以上，则启用了终止保护的集群将继续运行，并且已禁用终止保护的集群可能会终止。
- 如果网络连接会因日常维护而受到影响，我们建议主动启用终止保护。一般而言，连接中断意味着无法访问 Outpost 或客户网络本地的任何外部依赖关系。如果具有多个主节点的 Amazon EMR 集群使用区域内实例，则包括 Amazon S3、与 EMRFS 一致性视图结合使用的 DynamoDB 以及 Amazon RDS。

在 Amazon Outposts 上创建 Amazon EMR 集群

在 Amazon Outposts 上创建 Amazon EMR 集群与在 Amazon 云上创建 Amazon EMR 集群相类似。在 Amazon Outposts 上创建 Amazon EMR 集群时，您必须指定与您的 Outpost 关联的 Amazon EC2 子网。

Amazon VPC 可以跨Amazon区域中的所有可用区。Amazon Outposts 是可用区的扩展，您可以将账户中的 Amazon VPC 扩展为以跨多个可用区和关联的 Outpost 位置。在配置 Outpost 时，您将一个子网与它相关联，以将区域 VPC 环境扩展到本地设施。Outpost 实例和相关服务显示为区域 VPC 的一部分，类似于具有关联子网的可用区。有关更多信息，请参阅 Amazon Outposts 用户指南。

控制台

要在 Amazon Outposts 上使用Amazon Web Services Management Console创建新 Amazon EMR 集群，请指定与您的 Outpost 关联的 Amazon EC2 子网。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台在 Amazon Outposts 上创建集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster configuration（集群配置）下，选择 Instance groups（实例组）或 Instance fleets（实例集）。然后，从 Choose EC2 instance type（选择 EC2 实例类型）下拉菜单中选择实例类型，或依次选择 Actions（操作）、Add EBS volumes（添加 EBS 卷）。Amazon Outposts 上的 Amazon EMR 支持有限的 Amazon EBS 卷和实例类型。
4. 在 Networking（联网）下，选择一个 EC2 子网，其 Outpost ID 格式如下：op-123456789。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台在 Amazon Outposts 上创建集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 在 Software Configuration (软件配置) 下，对于 Release (版本)，选择 5.28.0 或更高版本。
5. 在 Hardware Configuration (硬件配置) 下，对于 EC2 Subnet (EC2 子网)，选请择具有以下格式的 Outpost ID 的 EC2 子网：op-123456789。
6. 为统一的实例组或实例集选择实例类型或添加 Amazon EBS 存储卷。Amazon Outposts 上的 Amazon EMR 支持有限的 Amazon EBS 卷和实例类型。

CLI

使用 Amazon CLI 在 Amazon Outposts 上创建集群

- 要在 Amazon Outposts 上使用 Amazon CLI 创建新的 Amazon EMR 集群，请指定与您的 Outpost 关联的 EC2 子网，如下例所示。将 *subnet-22XXXX01* 替换为您自己的 EC2 子网 ID。

```
aws emr create-cluster \
--name "Outpost cluster" \
--release-label emr-5.36.1 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey SubnetId=subnet-22XXXX01 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Amazon Local Zones 上的 EMR 集群

从 Amazon EMR 版本 5.28.0 开始，您可以在 Amazon Local Zones 子网上创建和运行 Amazon EMR 集群，作为支持 Local Zones 的Amazon区域的逻辑扩展。本地区域使得 Amazon EMR 功能和Amazon服务的子集（如计算和存储服务）在位置上与用户更近，从而为本地运行的应用程序提供非常低的延迟访问。有关可用的 Local Zones 列表，请参阅 Amazon Local Zones。有关访问可用Amazon Local Zones 的信息，请参阅区域、可用区和 Local Zones。

支持的实例类型

下列实例类型可用于 Local Zones 的 Amazon EMR 集群。实例类型可用性可能因区域而异。

实例类	实例类型
通用型	m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge
计算优化	c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge | c5d.xlarge | c5d.2xlarge | c5d.4xlarge| c5d.9xlarge | c5d.18xlarge

实例类	实例类型
内存优化	r5.xlarge | r5.2xlarge | r5.4xlarge | r5.12xlarge | r5d.xlarge | r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge
存储优化	i3en.xlarge | i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge | i3en.24xlarge

在 Local Zones 上创建 Amazon EMR 集群

通过以下方法在 Amazon Local Zones 中创建 Amazon EMR 集群：将 Amazon EMR 集群启动到与本地区域关联的 Amazon VPC 子网中。您可以使用美国西部（俄勒冈）控制台中的 Local Zones 名称（如 us-west-2-lax-1a）访问集群。

Local Zones 目前不支持 Amazon EMR Notebooks，也不支持使用接口 VPC 端点（Amazon PrivateLink）直接连接到 Amazon EMR。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台在本地区域创建集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Networking（网络）下，选择一个 EC2 子网，其具有以下格式的 Local Zone ID：subnet 123abc | us-west-2-lax-1a。
4. 为统一的实例组或实例集选择实例类型或添加 Amazon EBS 存储卷。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台在本地区域创建集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 在 Software Configuration (软件配置) 下，对于 Release (版本)，选择 5.28.0 或更高版本。
5. 在 Hardware Configuration (硬件配置) 下，对于 EC2 Subnet (EC2 子网)，选择具有以下格式的 Local Zones ID 的 EC2 子网：subnet 123abc | us-west-2-lax-1a。
6. 为统一的实例组或实例集添加 Amazon EBS 存储卷并选择实例类型。

CLI

使用 Amazon CLI 在本地区域创建集群

- 使用 create-cluster 命令以及 Local Zones 的 SubnetId，如以下示例中所示。将 subnet-22XXXX1234567 替换为本地区域 SubnetId，并根据需要替换其它选项。有关更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr/create-cluster.html。

```
aws emr create-cluster \
--name "Local Zones cluster" \
--release-label emr-5.29.0 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey,SubnetId=subnet-22XXXX1234567 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

配置 Docker

Amazon EMR 6.x 支持 Hadoop 3，它允许 YARN NodeManager 直接在 Amazon EMR 集群上或在 Docker 容器内启动容器。Docker 容器提供了自定义执行环境，应用程序代码在其中运行。自定义执行环境，与 YARN NodeManager、其它应用程序的执行环境相互隔离。

Docker 容器可以包含应用程序使用的特殊库，并且还可以提供不同版本的本机工具和库（如 R 和 Python）。您可以使用熟悉的 Docker 工具，以定义应用程序的库和运行时依赖项。

默认情况下，将 Amazon EMR 6.x 集群配置为允许 YARN 应用程序（如 Spark）使用 Docker 容器运行。要自定义容器配置，请编辑 `yarn-site.xml` 中定义的 Docker 支持选项和 `/etc/hadoop/conf` 目录中提供的 `container-executor.cfg` 文件。有关每个配置选项及其使用方式的详细信息，请参阅使用 Docker 容器启动应用程序。

提交任务时，您可以选择使用 Docker。请使用以下变量指定 Docker 运行时和 Docker 镜像。

- `YARN_CONTAINER_RUNTIME_TYPE=docker`
- `YARN_CONTAINER_RUNTIME_DOCKER_IMAGE={DOCKER_IMAGE_NAME}`

您使用 Docker 容器运行 YARN 应用程序时，YARN 会下载您在提交任务时指定的 Docker 镜像。要想让 YARN 解析此 Docker 镜像，必须使用 Docker 注册表进行配置。Docker 注册表的配置选项取决于您使用公有子网还是私有子网来部署集群。

Docker 注册表

Docker 注册表是 Docker 镜像的存储和分配系统。对于 Amazon EMR，我们建议您使用 Amazon ECR，这是一个完全托管式的 Docker 容器注册表，让您能够创建自己的自定义镜像，并在高度可用且可扩展的架构中托管。

部署注意事项

Docker 注册表要求从集群中的每个主机进行网络访问。这是因为当您的 YARN 应用程序在集群上运行时，每个主机都会从 Docker 注册表下载镜像。这些网络连接要求可能会限制您对 Docker 注册表的选择，具体取决于您是将 Amazon EMR 集群部署到公有子网还是私有子网。

公有子网

当 EMR 集群在公有子网中部署时，运行 YARN NodeManager 的节点可以直接访问互联网上任何可用的注册表。

私有子网

当 EMR 集群在私有子网中部署时，运行 YARN NodeManager 的节点无法直接访问互联网。Docker 镜像可以托管在 Amazon ECR 中，并通过 Amazon PrivateLink 访问。

有关如何使用 Amazon PrivateLink 允许在私有子网场景中访问 Amazon ECR 的信息，请参阅为 Amazon ECS 和 Amazon ECR 设置 Amazon PrivateLink。

配置 Docker 注册表

要在 Amazon EMR 中使用 Docker 注册，必须将 Docker 配置为信任要用于解析 Docker 镜像的特定注册表。原定设置的信任注册表是本地（私有）和 centos 注册表。要使用其它公有存储库或 Amazon ECR，您可以使用带有 `container-executor` 分类键的 EMR 分类 API 来覆盖 `/etc/hadoop/conf/container-executor.cfg` 中的 `docker.trusted.registries` 设置。

以下示例演示了如何将集群配置为信任名为 `your-public-repo` 的公有存储库和 ECR 注册表终端节点 `123456789123.dkr.ecr.us-east-1.amazonaws.com`。如果您使用 ECR，请将此终端节点替换为您的特定 ECR 终端节点。

```
[
  {
    "Classification": "container-executor",
    "Configurations": [
        {
            "Classification": "docker",
            "Properties": {
                "docker.trusted.registries": "local,centos,your-public-
repo,123456789123.dkr.ecr.us-east-1.amazonaws.com",
                "docker.privileged-containers.registries": "local,centos,your-public-
repo,123456789123.dkr.ecr.us-east-1.amazonaws.com"
            }
        }
    ]
  }
]
```

要使用 Amazon Command Line Interface（Amazon CLI）启动具有此配置的 Amazon EMR 6.0.0 集群，请创建名为 `container-executor.json` 的文件，其中包含上述 ontainer-executor JSON 配置的内容。然后，使用以下命令启动集群。

```
export KEYPAIR=<Name of your Amazon EC2 key-pair>
export SUBNET_ID=<ID of the subnet to which to deploy the cluster>
export INSTANCE_TYPE=<Name of the instance type to use>
export REGION=<Region to which to deploy the cluster>

aws emr create-cluster \
    --name "EMR-6.0.0" \
    --region $REGION \
    --release-label emr-6.0.0 \
```

```
    --applications Name=Hadoop Name=Spark \
    --service-role EMR_DefaultRole \
    --ec2-attributes KeyName=$KEYPAIR,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=
$SUBNET_ID \
    --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=
$INSTANCE_TYPE InstanceGroupType=CORE,InstanceCount=2,InstanceType=$INSTANCE_TYPE \
    --configuration file://container-executor.json
```

将 YARN 配置为访问 Amazon ECR on EMR 6.0.0 及更早版本

如果您是 Amazon ECR 的新用户，请按照 Amazon ECR 入门中的说明进行操作，并验证您是否有权从 Amazon EMR 集群中的每个实例访问 Amazon ECR。

在 EMR 6.0.0 及更早版本上，要使用 Docker 命令访问 Amazon ECR，必须首先生成凭证。要验证 YARN 是否可以从 Amazon ECR 访问镜像，请使用容器环境变量 `YARN_CONTAINER_RUNTIME_DOCKER_CLIENT_CONFIG` 来传递对您所生成凭证的引用。

在其中一个核心节点上运行以下命令，获取 ECR 账户的登录行。

```
aws ecr get-login --region us-east-1 --no-include-email
```

`get-login` 命令会生成正确的 Docker CLI 命令来运行，以创建凭证。从 `get-login` 复制输出并运行它。

```
sudo docker login -u AWS -p <password> https://<account-id>.dkr.ecr.us-
east-1.amazonaws.com
```

此命令在 `/root/.docker` 文件夹中生成 `config.json` 文件。将此文件复制到 HDFS，以便提交到集群的任务可以使用它对 Amazon ECR 进行身份验证。

运行以下命令，将 `config.json` 文件复制到您的主目录。

```
mkdir -p ~/.docker
sudo cp /root/.docker/config.json ~/.docker/config.json
sudo chmod 644 ~/.docker/config.json
```

请运行以下将 config.json 放入 HDFS 的命令，以便在集群上运行任务时使用。

```
hadoop fs -put ~/.docker/config.json /user/hadoop/
```

YARN 可以将 ECR 作为 Docker 镜像注册表访问，并在任务执行期间提取容器。

配置 Docker 注册表和 YARN 后，您可以使用 Docker 容器来运行 YARN 应用程序。有关更多信息，请参阅使用 Amazon EMR 6.0.0 通过 Docker 运行 Spark 应用程序。

在 EMR 6.1.0 及更高版本中，您无需手动设置对 Amazon ECR 的身份验证。如果在 `container-executor` 分类键中检测到 Amazon ECR 注册表，将激活 Amazon ECR 自动身份验证功能，并且 YARN 会在您提交带有 ECR 镜像的 Spark 任务时处理身份验证过程。您可以通过在 yarn-site 中检查 `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` 来确认是否启用了自动身份验证。如果 `docker.trusted.registries` 包含 ECR 注册表 URL，将启用自动身份验证，并将 YARN 身份验证设置设置为 `true`。

对 Amazon ECR 使用自动身份验证的先决条件

- EMR 版本 6.1.0 或更高版本
- 配置中包含的 ECR 注册表与集群位于同一个区域
- IAM 角色，具有获取授权令牌并提取任何镜像的权限

有关更多信息，请参阅使用 Amazon ECR 进行设置。

如何启用自动身份验证

按照 配置 Docker 注册表 将 Amazon ECR 注册表设置为受信任的注册表，并确保 Amazon ECR 存储库和集群位于同一区域。

若要启用此功能（即使未在受信任的注册表中设置 ECR 注册表），请使用配置分类将 `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` 设置为 `true`。

如何禁用自动身份验证

默认情况下，如果在受信任的注册表中未检测到 Amazon ECR 注册表，则表明已禁用自动身份验证。

要禁用自动身份验证（即使在受信任的注册表中设置了 Amazon ECR 注册表），请使用配置分类将 `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` 设置为 `false`。

如何检查是否在集群上启用了自动身份验证

在主节点上，使用文本编辑器（如 `vi`）来查看文件内容：`vi /etc/hadoop/conf.empty/yarn-site.xml`。检查 `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` 的值。

控制集群终止

本部分介绍了关闭 Amazon EMR 集群的选项。它涵盖了自动终止和终止保护，以及它们与其他 Amazon EMR 功能的交互方式。

您可以通过以下方式关闭 Amazon EMR 集群：

- 最后一步执行后终止- 在所有步骤完成后，创建一个关闭的暂时集群。
- 自动终止（空闲时间后）- 在指定的空闲时间后，创建具有关闭的自动终止策略的集群。有关更多信息，请参阅使用自动终止策略。
- 手动终止- 创建长时间运行的集群，可以运行到您故意终止。有关如何手动终止集群的信息，请参阅终止集群。

您还可以在集群上设置终止保护，以避免意外或错误关闭 EC2 实例。

当 Amazon EMR 关闭集群时，集群的所有 Amazon EC2 实例都将关闭。实例存储和 EBS 卷中的数据已不再可用，且无法恢复。如果要制定一个策略以通过向 Amazon S3 写入数据并平衡成本来管理和保留数据，那么了解和管理集群终止就至关重要。

主题

- 将集群配置为在步骤执行后继续或终止
- 使用自动终止策略
- 使用终止保护

将集群配置为在步骤执行后继续或终止

本主题解阐释了使用长时间运行的集群与创建在最后一步运行后关闭的临时集群之间的区别。它还介绍了如何为集群配置步骤执行。

创建长时间运行的集群

默认情况下，使用控制台或 Amazon CLI 创建的集群是长时间运行的集群。长时间运行的集群将继续运行、接受工作并累积费用，直到您采取措施关闭它们。

长时间运行的集群在以下情况十分有效：

- 当你需要交互式或自动查询数据时。

- 当您需要与集群上托管的大数据应用程序持续交互时。
- 当您定期处理大型数据集或需要频繁处理时，每次都无法有效启动新集群和加载数据。

您还可以在长时间运行的集群上设置终止保护，以避免意外或错误关闭 EC2 实例。有关更多信息，请参阅使用终止保护。

> **Note**
>
> Amazon EMR 会为具有多个主节点的所有集群自动启用终止保护，并覆盖您在创建集群时应用的所有步骤执行设置。您可以在集群启动后禁用终止保护。请参阅为正在运行的集群配置终止保护。要关闭具有多个主节点的集群，您必须先修改集群属性，才能禁用终止保护。有关说明，请参阅 终止具有多个主节点的 Amazon EMR 集群。

将集群配置为在步骤执行后终止

在步骤执行后配置终止时，集群将启动、运行引导启动操作，然后运行您指定的步骤。最后一步完成后，Amazon EMR 将终止集群的 Amazon EC2 实例。默认情况下，使用 Amazon EMR API 启动的集群会启动步骤执行。

步骤执行后终止，对于定期执行处理任务（例如日常数据处理运行）的集群来说，极其有效。步骤执行还可帮助您确保您只需按处理数据所需的时间付费。有关步骤的更多信息，请参阅 向集群提交工作。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台启动步骤执行

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。

3. 在 Steps（步骤）下，选择 Add step（添加步骤）。在 Add Step（添加步骤）对话框中，输入相应的字段值。选项因步骤类型而异。要添加步骤并退出对话框，请选择 Add step（添加步骤）。
4. 在 Cluster termination（集群终止）下，选中 Terminate cluster after last step completes（最后一步完成后终止集群）复选框。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台启动步骤执行

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Step execution (步骤执行)。
4. 选择适用于您的应用程序的其它设置，然后选择 Create Cluster (创建集群)。

Amazon CLI

使用 Amazon CLI 启动步骤执行

- 使用 `--auto-terminate` 命令时指定 `create-cluster` 参数以创建暂时性集群。

  以下示例说明了如何使用 `--auto-terminate` 参数。您可以键入以下命令，将 *myKey* 替换为您的 EC2 密钥对的名称。

  > **Note**
  >
  > 为了便于读取，包含 Linux 行继续符（\）。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^)。

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.36.1 \
--applications Name=Hive Name=Pig --use-default-roles --ec2-attributes
 KeyName=myKey \
--steps Type=PIG,Name="Pig Program",ActionOnFailure=CONTINUE,\
```

```
Args=[-f,s3://mybucket/scripts/pigscript.pig,-p,\
INPUT=s3://mybucket/inputdata/,-p,OUTPUT=s3://mybucket/outputdata/,\
$INPUT=s3://mybucket/inputdata/,$OUTPUT=s3://mybucket/outputdata/]
--instance-type m5.xlarge --instance-count 3 --auto-terminate
```

API

使用 Amazon EMR API 关闭步骤执行

- 使用 RunJobFlow 操作创建集群时，应将 KeepJobFlowAliveWhenNoSteps 属性设置为 `true`。

使用自动终止策略

自动终止策略允许您编排集群清理，而无需监控和手动终止未使用的集群。向集群添加自动终止策略时，您可以指定空闲时间，在此时间之后集群应自动关闭。

根据发布版本，Amazon EMR 使用不同的条件将集群标记为空闲。下表概述了 Amazon EMR 如何确定集群空闲状态。

当您使用以下应用程序时：	当符合以下条件时，集群被视为处于空闲状态：
Amazon EMR 5.34.0 及更高版本、6.4.0 及更高版本	• 没有活跃的 YARN 应用程序 • HDFS 利用率低于 10% • 没有活跃的 EMR 笔记本或 EMR Studio 连接 • 没有群集上的应用程序用户界面正被使用
Amazon EMR 版本 5.30.0 - 5.33.0 和 6.1.0 - 6.3.0	• 没有活跃的 YARN 应用程序 • 集群没有活跃的 Spark 作业

当您使用以下应用程序时：	当符合以下条件时，集群被视为处于空闲状态：
	Note Amazon EMR 将集群标记为空闲，即使您有活跃的 Python3 内核，也可能会自动终止该集群。这是因为执行 Python3 内核不会在该集群上提交 Spark 作业。要将自动终止与 Python3 内核配合使用，我们建议您使用 Amazon EMR 版本 6.4.0 或更高版本。

Note

Amazon EMR 6.4.0 及更高版本支持用于检测主节点上活动的集群文件：`/emr/metricscollector/isbusy`。当你使用集群运行 shell 脚本或非 YARN 应用程序时，您可以定期接触或更新`isbusy`，以告知Amazon EMR 集群不处于空闲状态。

您可以在创建集群时附加自动终止策略，也可以将策略添加到现有集群。要更改或禁用自动终止，您可以更新或删除策略。

注意事项

在使用自动终止策略之前，请注意以下功能和限制：

- 对于亚太地区（雅加达）区域，Amazon EMR 自动终止功能现已在 Amazon EMR 6.14.0 及更高版本中推出。
- 在下列 Amazon Web Services 区域，Amazon EMR 5.30.0 和 6.1.0 及更高版本提供 Amazon EMR 自动终止功能：

  美国东部（弗吉尼亚州北部和俄亥俄州）、美国西部（俄勒冈州和加利福尼亚州北部）、南美洲（圣保罗）、欧洲地区（法兰克福、爱尔兰、伦敦、米兰、巴黎和斯德哥尔摩）、加拿大（中部）、亚太地区（香港、孟买、首尔、新加坡、悉尼和东京）、中东（巴林）、非洲（开普敦）、Amazon GovCloud（美国东部）、Amazon GovCloud（美国西部）、中国（北京，由光环新网运营）和中国（宁夏，由西云数据运营）。

- 如果您没有指定金额，则空闲超时默认为 60 分钟（1 小时）。您可以指定最小空闲超时为 1 分钟，最长空闲超时时间为 7 天。
- 在 Amazon EMR 6.4.0 及更高版本中，当您使用 Amazon EMR 控制台创建新集群时，默认情况下会启用自动终止。
- 当您为集群启用自动终止时，Amazon EMR 将发布高分辨率 Amazon CloudWatch 指标。您可以使用这些指标来跟踪集群的活动和空闲状态。有关更多信息，请参阅集群容量指标。
- 当您使用基于非基于 YARN 的应用程序（例如 Presto、Trino 或 HBase）时，将不支持自动终止。
- 要使用自动终止，指标收集器进程必须能够连接到公有 API 端点，以便在 API Gateway 中执行自动终止。如果您通过 Amazon Virtual Private Cloud 使用私有 DNS 名称，自动终止将无法正常运行。为确保自动终止正常运行，我们建议您执行以下操作之一：
  - 从您的 Amazon VPC 中删除 API Gateway 接口 VPC 终端节点。
  - 按照我从 VPC 连接到 API Gateway API 时，为什么会收到 HTTP 403 禁止错误？中的说明禁用私有 DNS 名称设置。
  - 在您的私有子网中启动集群。有关更多信息，请参阅 私有子网 中的主题。
- 对于 EMR 5.30.0 及更高版本，如果您将主要安全组默认的 Allow All（允许所有）出站规则删除为 0.0.0.0/，则您必须添加一条规则，以允许出站 TCP 连接到安全组，从而在端口 9443 上进行服务访问。您的服务访问安全组应允许来自主要安全组端口 9443 上的入站 TCP 流量。有关配置安全组的更多信息，请参阅适用于主实例（私有子网）的 Amazon EMR 托管安全组。

使用自动终止的权限

您需要将以下示例 IAM 权限策略中列出的权限附加到管理您的 EMR 集群的 IAM 资源，然后才能应用和管理 Amazon EMR 的自动终止策略。

```
{
  "Version": "2012-10-17",
  "Statement": {
      "Sid": "AllowAutoTerminationPolicyActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:PutAutoTerminationPolicy",
        "elasticmapreduce:GetAutoTerminationPolicy",
        "elasticmapreduce:RemoveAutoTerminationPolicy"
      ],
      "Resource": "<your-resources>"
    }
}
```

附加、更新或删除自动终止策略

本部分包括从 Amazon EMR 集群附加、更新或删除自动终止策略的说明。在使用自动终止策略之前，请确保您有必要的 IAM 权限。请参阅使用自动终止的权限。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

在使用新控制台创建集群时，附加自动终止策略

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster termination（集群终止）下，选择 Terminate cluster after idle time（空闲时间后终止集群）。
4. 指定空闲时数和分钟数，在此之后，集群将自动终止。默认空闲时间为 1 小时。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

使用新控制台在正在运行的集群上附加、更新或删除自动终止策略

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Properties（属性）选项卡上，找到 Cluster termination（集群终止），然后选择 Edit（编辑）。
4. 选择或清除 Enable auto-termination（启用自动终止）以打开或关闭该功能。如果启用自动终止，请指定空闲时数和分钟数，在此之后，集群将自动终止。然后选择 Save changes（保存更改）进行确认。

Old console

在使用旧控制台创建集群时，附加自动终止策略

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 在硬件配置下，选择自动终止。
4. 指定空闲时数和分钟数，在此之后，集群应自动终止。默认空闲时间为 1 小时。
5. 选择适用于您的应用程序的其它设置，然后选择 Create Cluster (创建集群)。

使用旧控制台在正在运行的集群上附加、更新或删除自动终止策略

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择集群，然后选择要更新的集群。
3. 在 Cluster Details (集群详细信息) 页面上，选择 Hardware (硬件)标签。
4. 选择或清除 Enable auto-termination（启用自动终止）以打开或关闭该功能。如果启用自动终止，请指定空闲时数和分钟数，在此之后，集群将自动终止。

Amazon CLI

开始之前

在您使用自动终止策略之前，我们建议您更新到 Amazon CLI 的最新版本。有关说明，请参阅安装、更新和卸载 Amazon CLI。

使用Amazon CLI附加或更新自动终止策略

- 您可以使用`aws emr put-auto-termination-policy`命令附加或更新集群上的自动终止策略。

  以下示例指定了 3600 秒`######`。如果不指定`######`，该值默认为 1 小时。

```
aws emr put-auto-termination-policy \
--cluster-id <your-cluster-id> \
--auto-termination-policy IdleTimeout=3600
```

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字符 (^)。

当你使用`aws emr create-cluster`命令时，您还可以为`--auto-termination-policy`指定值。有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅 Amazon CLI命令参考。

使用 Amazon CLI 删除自动终止策略

- 使用`aws emr remove-auto-termination-policy`命令从集群中删除自动终止策略。有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅 Amazon CLI命令参考。

```
aws emr remove-auto-termination-policy --cluster-id <your-cluster-id>
```

使用终止保护

在长时间运行的集群上启用终止保护时，您仍然可以终止集群，但您必须先明确地从集群中删除终止保护。这有助于确保 EC2 实例不会因事故或错误而关闭。如果集群可能在本地磁盘中存储了数据，而您又需要在这些实例终止前予以恢复，那么这种终止保护尤其有用。您可以在创建集群时启用终止保护，并可以更改运行的集群上的设置。

启用终止保护后，Amazon EMR API 中的 `TerminateJobFlows` 操作将不起作用。用户不能使用此 API 或 Amazon CLI 中的 `terminate-clusters` 命令终止集群。该 API 会返回错误，而 CLI 会通过非零返回代码退出。使用 Amazon EMR 控制台终止集群时，系统会提示您采取额外步骤来关闭终止保护。

> **Warning**
>
> 终止保护不保证在发生人为错误或实施解决方法时保留数据 – 例如，使用 SSH 连接到实例时从命令行发出重新启动命令、实例上运行的应用程序或脚本发出重新启动命令，或者使用 Amazon EC2 或 Amazon EMR API 禁用终止保护。即使启用了终止保护，保存到实例存储的数据（包括 HDFS 数据）也可能会丢失。将数据输出写入 Amazon S3 位置，并根据您的业务连续性要求创建适当的备份策略。

终止保护不影响您使用以下任意操作扩展集群资源的能力：

- 使用 Amazon Web Services Management Console 或 Amazon CLI 手动调整集群大小。有关更多信息，请参阅手动调整正在运行的集群的大小。
- 搭配使用缩减策略和自动扩展，将实例从核心或任务实例组中删除。有关更多信息，请参阅将自动伸缩与实例组的自定义策略结合使用 。
- 通过减少目标容量，将实例从实例队列中删除。有关更多信息，请参阅实例集选项。

终止保护和 Amazon EC2

启用终止保护的 Amazon EMR 集群已为集群中的所有 Amazon EC2 实例设置了 `disableAPITermination` 属性。如果终止请求源自 Amazon EMR，且实例的 Amazon EMR 和 Amazon EC2 设置相冲突，则 Amazon EMR 设置将覆盖 Amazon EC2 设置。例如，假设您使用 Amazon EC2 控制台在 disabled（已禁用）终止保护的集群中的 Amazon EC2 实例上 enable（启用）终止保护，则当您使用 Amazon EMR 控制台、适用于 Amazon EMR 的 Amazon CLI 命令或 Amazon EMR API 终止该集群时，Amazon EMR 会将 `DisableApiTermination` 设置为 `false`，并终止该实例以及其他实例。

> **Important**
>
> 如果在具有终止保护功能的 Amazon EMR 集群中创建实例，并使用 Amazon EC2 API 或 Amazon CLI 命令来修改该实例，那么 `DisableApiTermination` 会设置为 `false`，之后 Amazon EC2 API 或 Amazon CLI 命令会执行 `TerminateInstances` 操作， Amazon EC2 实例也会终止。

终止保护和不正常的 YARN 节点

Amazon EMR 会定期检查集群中核心和任务 Amazon EC2 实例上运行的节点的 Apache Hadoop YARN 状态。运行状况由 NodeManager 运行状况检查程序服务报告。如果某一节点报告 `UNHEALTHY`，则 Amazon EMR 实例控制器将拒绝列出该节点，并且在该节点再次恢复正常之前不会为其分配 YARN 容器。节点不正常的常见原因是，磁盘利用率超过 90%。有关识别不正常节点并进行恢复的更多信息，请参阅资源错误。

如果节点保持 `UNHEALTHY` 状态超过 45 分钟，Amazon EMR 将根据终止保护的状态执行以下操作。

终止保护	结果

终止保护	结果
已启用（推荐）	Amazon EC2 核心实例仍处于拒绝列出状态，并将继续计入集群容量。您可以连接到 Amazon EC2 核心实例以进行配置和数据恢复，并调整集群大小以增加容量。有关更多信息，请参阅资源错误。 运行状况不佳的任务节点免受终止保护，并将被终止。
已禁用	Amazon EC2 实例已终止。Amazon EMR 根据实例组中实例的指定数量或实例集的目标容量预置新实例。如果所有核心节点处于 UNHEALTHY 状态超过 45 分钟，集群将终止，并报告 NO_SLAVES_LEFT 状态。 **Important** 如果核心实例因状态不正常而被终止，那么 HDFS 数据可能会丢失。如果节点存储了没有复制到其它节点的数据块，那么这些块都将丢失，这可能会导致数据丢失。我们建议您使用终止保护，以便您可以在必要时连接到实例并恢复数据。

终止保护和步骤执行

当你启用步骤执行并启用终止保护时，Amazon EMR 将忽略终止保护。

向集群提交步骤时，您可以设置 ActionOnFailure 属性，以确定步骤因错误而无法完成执行时发生的情况。此设置的可能值为 TERMINATE_CLUSTER（早期版本的值为 TERMINATE_JOB_FLOW）、CANCEL_AND_WAIT 和 CONTINUE。有关更多信息，请参阅向集群提交工作。

当其中 ActionOnFailure 设置为 CANCEL_AND_WAIT 的步骤失败时，如果已启用步骤执行，则集群会终止，且不会执行后续步骤。

当其中 ActionOnFailure 设置为 TERMINATE_CLUSTER 的步骤失败时，请使用下面的设置表来确定结果。

ActionOnFailure	步骤执行	终止保护	结果
TERMINATE_CLUSTER	启用	已禁用	集群会终止
	启用	启用	集群会终止
	已禁用	启用	集群继续运行
	已禁用	已禁用	集群会终止

终止保护和竞价型实例

当 Spot 价格上涨并超过最高 Spot 价格时，Amazon EMR 终止保护不会阻止 Amazon EC2 Spot 实例终止。

启动集群时配置终止保护

您可以使用控制台、Amazon CLI 或 API 在启动集群时启用或禁用终止保护。

默认终止保护设置取决于您启动集群的方式：

- Amazon EMR 控制台（新）：默认情况下，“终止保护”为已启用。
- Amazon EMR 控制台（旧）快速选项：默认情况下，“终止保护”为禁用。
- Amazon EMR 控制台（旧）高级选项：默认情况下，“终止保护”为已启用。
- Amazon CLI `aws emr create-cluster` – 除非指定 `--termination-protected`，否则 Termination Protection (终止保护) 是禁用的。
- Amazon EMR API RunJobFlow 命令 – 除非 `TerminationProtected` 布尔值设定为 `true`，否则 Termination Protection (终止保护) 是禁用的。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台创建集群时打开或关闭终止保护

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 对于 EMR release version（EMR 发行版），选择 emr-6.6.0 或更高版本。
4. 在 Cluster termination（集群终止）下，确保预先选择 Use termination protection（使用终止保护），或者清除该选项将其关闭。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台创建集群时打开或关闭终止保护

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 对于 Step 3: General Cluster Settings (步骤 3: 常规集群设置)，在 General Options (常规选项) 下，确保选中 Termination protection (终止保护) 以将其启用，或清除选中此选项以将其禁用。
5. 选择适用于您的应用程序的其它设置，再选择 Next (下一步)，然后完成集群配置。

Amazon CLI

使用 Amazon CLI 创建集群时打开或关闭终止保护

- 借助 Amazon CLI，您可以通过使用带有 `--termination-protected` 参数的 `create-cluster` 命令启动已启用终止保护的集群。默认情况下，终止保护处于禁用状态。

  以下示例创建启用了终止保护的集群：

  > **Note**
  >
  > 为了便于读取，包含 Linux 行继续符（\）。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号（^）。

```
aws emr create-cluster --name "TerminationProtectedCluster" --release-
label emr-5.36.1 \
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --termination-protected
```

  有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

为正在运行的集群配置终止保护

您可以使用控制台或 Amazon CLI 为运行中的集群配置终止保护。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台为正在运行的集群打开或关闭终止保护

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Properties（属性）选项卡上，找到 Cluster termination（集群终止），然后选择 Edit（编辑）。
4. 选中或清除 Use termination protection（使用终止保护）复选框以打开或关闭该功能。然后选择 Save changes（保存更改）进行确认。

Old console

使用旧控制台为正在运行的集群打开或关闭终止保护

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 在 Clusters (集群) 页面上，选择集群的 Name (名称)。
3. 在 Summary (摘要) 选项卡上，对于 Termination protection (终止保护)，选择 Change (更改)。
4. 要启用终止保护，请选择 On (开启)。要禁用终止保护，请选择 Off (关闭)。然后选择绿色复选标记进行确认。

Amazon CLI

使用 Amazon CLI 为正在运行的集群打开或关闭终止保护

- 要使用 Amazon CLI 为正在运行的集群启用终止保护，请使用带有 `--termination-protected` 参数的 `modify-cluster-attributes` 命令。要禁用终止保护，请使用 `--no-termination-protected` 参数。

  以下示例在 ID 为 *`j-3KVTXXXXXX7UG`* 的集群上启用了终止保护：

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --termination-protected
  ```

以下示例在相同集群上禁用了终止保护：

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
```

在 Amazon EMR 中使用 Amazon Linux AMI

Amazon Linux 亚马逊机器映像（AMI）

创建和启动集群时，Amazon EMR 使用 Amazon Linux Amazon Machine Image（AMI）初始化 Amazon EC2 实例。AMI 包含 Amazon Linux 操作系统、其它软件和每个用于托管集群应用程序的实例所需的配置。

默认情况下，创建集群时，Amazon EMR 运用专为您使用 Amazon EMR 发行版而创建的默认 Amazon Linux AMI。有关默认 Amazon Linux AMI 的更多信息，请参阅在 Amazon EMR 使用默认的 Amazon Linux AMI。使用 Amazon EMR 5.7.0 或更高版本时，您可以选择为 Amazon EMR 指定自定义 Amazon Linux AMI，而不是默认的 Amazon Linux AMI。作为使用引导操作的替代方案，您可以借助自定义 AMI 来加密根设备卷，并对应用程序和配置进行自定义。您可以为 Amazon EMR 集群的实例组或实例队列配置中的每种实例类型，指定自定义 AMI。多个自定义 AMI 支持使您能够灵活地在集群中使用多种架构类型。请参阅使用自定义 AMI。

Amazon EMR 自动将 Amazon EBS 通用型 SSD 卷附加为所有 AMI 的根设备。使用由 EBS 提供支持的 AMI 增强性能。根据运行集群的区域中 gp2 卷的每月 Amazon EBS 费用按小时按比例收取 EBS 成本。例如，按 0.10 美元/GB/月收费的区域中的每个集群实例上的根卷的每小时成本约为 0.00139 美元/小时（0.10 美元/GB/月除以 30 天除以 24 小时乘以 10 GB）。无论您使用默认 Amazon Linux AMI 还是自定义 Amazon Linux AMI，都可将 Amazon EBS 根设备卷的大小指定在 10-100GiB 之间。

有关Amazon Linux AMI 的更多信息，请参阅 Amazon Machine Image（AMI）。有关 Amazon EMR 实例的实例存储的更多信息，请参阅实例存储。

在 Amazon EMR 使用默认的 Amazon Linux AMI

除非您指定自定义 AMI，否则每个 Amazon EMR 发行版都会在 Amazon EMR 使用默认的 Amazon Linux AMI。从 Amazon EMR 5.36 和 Amazon EMR 6.6 版本开始，在 Amazon EMR 默认 AMI 中更新 Amazon Linux 2（AL2）的默认行为是自动为默认 Amazon EMR AMI 应用最新的 AL2 版本。

Amazon EMR 版本的 Amazon Linux 2 自动更新

当您使用 Amazon EMR 5.36 或更高版本或 6.6 或更高版本的最新补丁版本启动集群时，Amazon EMR 会使用最新的 Amazon Linux 2 版本作为默认 Amazon EMR AMI。例如：

- 如果有 `x.x.0` 和 `x.x.1` 版本，则 `x.x.0` 版本在 `x.x.1` 发布时会停止获取 AMI 更新。
- 同样，`x.x.2` 启动时，`x.x.1` 会停止获取 AMI 更新。
- 稍后，当 `x.y.0` 发布时，`x.x.[latest]` 还会继续收到 AMI 更新与 `x.y.[latest]`。

要查看您是否使用的是 Amazon EMR 版本的第二位小数点后的数字 (`6.8.1`) 表示的最新补丁版本，请参阅 Amazon EMR Release Guide 中的可用版本，在控制台中创建集群时查看 Amazon EMR 版本下拉菜单，或者使用 `ListReleaseLabels` API 或 `list-release-labels` 操作。要在我们发布新 Amazon EMR 版本时获取更新，请在 Release Guide 中的 What's new? 页面订阅 RSS 源。

如果愿意，您可以选择使用 Amazon EMR 版本首次附带的 Amazon Linux 版本来启动集群。有关如何为您的集群指定 Amazon Linux 版本的信息，请参阅 创建集群时更改 Amazon Linux 发行版。

默认 Amazon Linux 版本

下表列出了有关 Amazon EMR 6.x 版本 6.6 及更高版本的最新补丁版本的 Amazon Linux 信息。

OsRelea Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 727.0	4.14.320	2023 年 8 月 14 日	us-east-1 、us-east-2 、us-west-1 、us-west-2 、eu-north- 1 、eu-west-1 、eu-west-2 、eu-west-3 、eu-centra l-1 、eu-centra l-2 （6.10+）、eu-south-

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
			1 、eu-south-2 （6.10+）、ap-east-1 、ap-south-1 、ap-south-2 （6.10+）、ap-southeast-3 、ap-southeast-4 （6.8+ 和 5.36.1）、ap-northeast-1 、ap-northeast-2 、ap-northeast-3 、ap-southeast-1 、ap-southeast-2 、af-south-1 、sa-east-1 、me-central-1 （6.10+）、me-south-1 、ca-central-1 、il-central-1 （6.9+ 和 5.36.1）

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230719.0	4.14.320	2023 年 8 月 2 日	us-east-1 、us-east-2 、us-west-1 、us-west-2 、eu-north-1 、eu-west-1 、eu-west-2 、eu-west-3 、eu-central-1 、eu-central-2 （6.10+）、eu-south-1 、eu-south-2 （6.10+）、ap-east-1 、ap-south-1 、ap-south-2 （6.10+）、ap-southeast-3 、ap-southeast-4 （6.8+ 和 5.36.1）、ap-northeast-1 、ap-northeast-2 、ap-northeast-3 、ap-southeast-1 、ap-southeast-2 、af-south-1 、sa-east-1 、me-central-1 （6.10+）、me-south-1 、ca-central-1 、il-central-1 （6.9+ 和 5.36.1）

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230628.0	4.14.318	2023 年 7 月 12 日	us-east-1 、us-east-2 、us-west-1 、us-west-2 、eu-north-1 、eu-west-1 、eu-west-2 、eu-west-3 、eu-central-1 、eu-central-2 （6.10+）、eu-south-1 、eu-south-2 （6.10+）、ap-east-1 、ap-south-1 、ap-south-2 （6.10+）、ap-southeast-3 、ap-northeast-1 、ap-northeast-2 、ap-northeast-3 、ap-southeast-1 、ap-southeast-2 、af-south-1 、sa-east-1 、me-central-1 （6.10+）、me-south-1 、ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230612.0	4.14.314	2023 年 6 月 23 日	us-east-1 、us-east-2 、us-west-1 、us-west-2 、eu-north-1 、eu-west-1 、eu-west-2 、eu-west-3 、eu-central-1 、eu-central-2 （6.10+）、eu-south-1 、eu-south-2 （6.10+）、ap-east-1 、ap-south-1 、ap-south-2 （6.10+）、ap-southeast-3 、ap-northeast-1 、ap-northeast-2 、ap-northeast-3 、ap-southeast-1 、ap-southeast-2 、af-south-1 、sa-east-1 、me-central-1 （6.10+）、me-south-1 、ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230504.1	4.14.313	2023 年 5 月 16 日	us-east-1 、us-east-2 、us-west-1 、us-west-2 、eu-north-1 、eu-west-1 、eu-west-2 、eu-west-3 、eu-central-1 、eu-central-2 （6.10+）、eu-south-1 、eu-south-2 （6.10+）、ap-east-1 、ap-south-1 、ap-south-2 （6.10+）、ap-southeast-3 、ap-northeast-1 、ap-northeast-2 、ap-northeast-3 、ap-southeast-1 、ap-southeast-2 、af-south-1 、sa-east-1 、me-central-1 、me-south-1 、ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230418.0	4.14.311	2023 年 5 月 3 日	us-east-1 、us-east-2 、us-west-1 、us-west-2 、eu-north-1 、eu-west-1 、eu-west-2 、eu-west-3 、eu-central-1 、eu-central-2 （仅限 6.10）、eu-south-1 、eu-south-2 （仅限 6.10）、ap-east-1 、ap-south-1 、ap-south-2 （仅限 6.10）、ap-southeast-3 、ap-northeast-1 、ap-northeast-2 、ap-northeast-3 、ap-southeast-1 、ap-southeast-2 、af-south-1 、sa-east-1 、me-central-1 、me-south-1 、ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230404.1	4.14.311	2023 年 4 月 18 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-central-2 ，eu-south-1 ，eu-south-2 ，ap-east-1 ，ap-south-1 ，ap-south-2 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-central-1 ，me-south-1 ，ca-central-1
2.0.20230404.0	4.14.311	2023 年 4 月 10 日	us-east-1 ，eu-west-3

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230320.0	4.14.309	2023 年 3 月 30 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-central-2 ，eu-south-1 ，eu-south-2 ，ap-east-1 ，ap-south-1 ，ap-south-2 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-central-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230307.0	4.14.305	2023 年 3 月 15 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-central-2 ，eu-south-1 ，eu-south-2 ，ap-east-1 ，ap-south-1 ，ap-south-2 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-central-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230207.0	4.14.304	2023 年 3 月 3 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-central-2 ，eu-south-1 ，eu-south-2 ，ap-east-1 ，ap-south-1 ，ap-south-2 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-central-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20230119.1	4.14.301	2023 年 2 月 9 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20221210.1	4.14.301	2023 年 1 月 12 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20221103.3	4.14.296	2022 年 12 月 5 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20220004.0	4.14.294	2022 年 11 月 2 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20220912.1	4.14.291	2022 年 10 月 7 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1
2.0.20220805.0	4.14.287	2022 年 8 月 30 日	us-west-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20220719.0	4.14.287	2022 年 8 月 10 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1

OsRelease Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.20220426.0	4.14.281	2022 年 6 月 10 日	us-east-1 ，us-east-2 ，us-west-1 ，us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ，eu-west-3 ，eu-central-1 ，eu-south-1 ，ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ，sa-east-1 ，me-south-1 ，ca-central-1

OsRelea Label Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2022 406.1	4.14.275	2022 年 5 月 2 日	us-east-1 ，us-east-2 ，us-west-1 ， us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ， eu-west-3 ，eu-central-1 ，eu-south-1 ， ap-east-1 ，ap-south-1 ，ap-southeast-3 ，ap-northeast-1 ，ap-northeast-2 ，ap-northeast-3 ，ap-southeast-1 ，ap-southeast-2 ，af-south-1 ， sa-east-1 ，me-south-1 ，ca-central-1

下表列出了有关 Amazon EMR 5.x 版本 5.36 及更高版本的最新补丁版本的 Amazon Linux 信息。

OsRelea Label（A zon Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 504.1	4.14.313	2023 年 5 月 16 日	美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（北加利福尼亚）、美国西部（俄勒冈州）、加拿大（中部）、欧

OsRelea Label（A zon Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
			洲地区（斯德哥尔摩）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（巴黎）、欧洲地区（法兰克福）、欧洲地区（米兰）、亚太地区（香港）、亚太地区（孟买）、亚太地区（雅加达）、亚太地区（东京)、亚太地区（首尔）、亚太地区（大阪）、亚太地区（新加坡）、亚太地区（悉尼）、非洲（开普敦）、南美洲（圣保罗）、中东（巴林）、中东（阿联酋）

OsRelea Label（A zon Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 418.0	4.14.311	2023 年 5 月 3 日	美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（北加利福尼亚）、美国西部（俄勒冈州）、加拿大（中部）、欧洲地区（斯德哥尔摩）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（巴黎）、欧洲地区（法兰克福）、欧洲地区（米兰）、亚太地区（香港）、亚太地区（孟买）、亚太地区（雅加达）、亚太地区（东京)、亚太地区（首尔）、亚太地区（大阪）、亚太地区（新加坡）、亚太地区（悉尼）、非洲（开普敦）、南美洲（圣保罗）、中东（巴林）、中东（阿联酋）

OsRelea Label（A zon Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 404.1	4.14.311	2023 年 4 月 18 日	美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（北加利福尼亚）、美国西部（俄勒冈州）、加拿大（中部）、欧洲地区（斯德哥尔摩）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（巴黎）、欧洲地区（法兰克福）、欧洲地区（米兰）、亚太地区（香港）、亚太地区（孟买）、亚太地区（雅加达）、亚太地区（东京)、亚太地区（首尔）、亚太地区（大阪）、亚太地区（新加坡）、亚太地区（悉尼）、非洲（开普敦）、南美洲（圣保罗）、中东（巴林）
2.0.2023 404.0	4.14.311	2023 年 4 月 10 日	美国东部（弗吉尼亚州北部）、欧洲地区（巴黎）

OsRelea Label（A zon Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 320.0	4.14.309	2023 年 3 月 30 日	美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（北加利福尼亚）、美国西部（俄勒冈州）、加拿大（中部）、欧洲地区（斯德哥尔摩）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（巴黎）、欧洲地区（法兰克福）、欧洲地区（米兰）、亚太地区（香港）、亚太地区（孟买）、亚太地区（雅加达）、亚太地区（东京)、亚太地区（首尔）、亚太地区（大阪）、亚太地区（新加坡）、亚太地区（悉尼）、非洲（开普敦）、南美洲（圣保罗）、中东（巴林）

OsRelea Label（A zon Linux 版本）	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 307.0	4.14.305	2023 年 3 月 15 日	美国东部（弗吉尼亚州北部）、美国东部（俄亥俄州）、美国西部（北加利福尼亚）、美国西部（俄勒冈州）、加拿大（中部）、欧洲地区（斯德哥尔摩）、欧洲地区（爱尔兰）、欧洲地区（伦敦）、欧洲地区（巴黎）、欧洲地区（法兰克福）、欧洲地区（米兰）、亚太地区（香港）、亚太地区（孟买）、亚太地区（雅加达）、亚太地区（东京)、亚太地区（首尔）、亚太地区（大阪）、亚太地区（新加坡）、亚太地区（悉尼）、非洲（开普敦）、南美洲（圣保罗）、中东（巴林）

OsRelea Label (A zon Linux 版本)	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2023 207.0	4.14.304	2023 年 3 月 3 日	us-east-1 , us-east-2 , us-west-1 , us-west-2 , eu-north-1 , eu-west-1 , eu-west-2 , eu-west-3 , eu-centra l-1 , eu-south-1 , ap-east-1 , ap-south-1 , ap-southe ast-3 , ap-northe ast-1 , ap-northe ast-2 , ap-northe ast-3 , ap-southe ast-1 , ap-southe ast-2 , af-south-1 , sa-east-1 , me-south-1 , ca-central-1

OsRelea Label (A zon Linux 版本)	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2022 210.1	4.14.301	2023 年 1 月 12 日	us-east-1 , us-east-2 , us-west-1 , us-west-2 , eu-north-1 , eu-west-1 , eu-west-2 , eu-west-3 , eu-centra l-1 , eu-south-1 , ap-east-1 , ap-south-1 , ap-southe ast-3 , ap-northe ast-1 , ap-northe ast-2 , ap-northe ast-3 , ap-southe ast-1 , ap-southe ast-2 , af-south-1 , sa-east-1 , me-south-1 , ca-central-1

OsRelea Label (A zon Linux 版本)	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2022 103.3	4.14.296	2022 年 12 月 5 日	us-east-1 , us-east-2 , us-west-1 , us-west-2 , eu-north-1 , eu-west-1 , eu-west-2 , eu-west-3 , eu-central-1 , eu-south-1 , ap-east-1 , ap-south-1 , ap-southeast-3 , ap-northeast-1 , ap-northeast-2 , ap-northeast-3 , ap-southeast-1 , ap-southeast-2 , af-south-1 , sa-east-1 , me-south-1 , ca-central-1

OsRelea Label (A zon Linux 版本)	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2022 004.0	4.14.294	2022 年 11 月 2 日	us-east-1 , us-east-2 , us-west-1 , us-west-2 , eu-north-1 , eu-west-1 , eu-west-2 , eu-west-3 , eu-centra l-1 , eu-south-1 , ap-east-1 , ap-south-1 , ap-southe ast-3 , ap-northe ast-1 , ap-northe ast-2 , ap-northe ast-3 , ap-southe ast-1 , ap-southe ast-2 , af-south-1 , sa-east-1 , me-south-1 , ca-central-1

OsRelea Label (A zon Linux 版本)	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2022 912.1	4.14.291	2022 年 10 月 7 日	us-east-1 , us-east-2 , us-west-1 , us-west-2 , eu-north-1 , eu-west-1 , eu-west-2 , eu-west-3 , eu-centra l-1 , eu-south-1 , ap-east-1 , ap-south-1 , ap-southe ast-3 , ap-northe ast-1 , ap-northe ast-2 , ap-northe ast-3 , ap-southe ast-1 , ap-southe ast-2 , af-south-1 , sa-east-1 , me-south-1 , ca-central-1
2.0.2022 719.0	4.14.287	2022 年 8 月 10 日	us-west-1 , eu-west-3 , eu-north-1 , eu-central-1 , ap-south-1 , me-south-1

OsRelea Label (A zon Linux 版本)	Amazon Linux 内核版本	可用日期	支持的区域
2.0.2022 426.0	4.14.281	2022 年 6 月 14 日	us-east-1 ，us-east-2 ，us-west-1 ， us-west-2 ，eu-north-1 ，eu-west-1 ，eu-west-2 ， eu-west-3 ，eu-centra l-1 ，eu-south-1 ， ap-east-1 ，ap-south-1 ，ap-southe ast-3 ，ap-northe ast-1 ，ap-northe ast-2 ，ap-northe ast-3 ，ap-southe ast-1 ，ap-southe ast-2 ，af-south-1 ， sa-east-1 ，me-south-1 ，ca-central-1

软件更新注意事项

以下是需要注意的默认软件更新行为：

Amazon EMR 5.35.0 及更早版本，以及 6.5.0 及更早版本 – Amazon Linux AMI“已锁定”为 Amazon EMR 版本

对于 Amazon EMR 5.35.0 及更早版本以及 6.5.0 及更早版本，默认 AMI 基于发行 Amazon EMR 时最新的可用 Amazon Linux AMI。AMI 已使用大数据应用程序和该版本中附带的 Amazon EMR 功能针对兼容性进行测试。

每个 Amazon EMR 5.35.0 及更早版本、6.5.0 及更早的 Amazon EMR 版本均“已锁定”为分配的 Amazon Linux AMI 版本，以保持兼容性。这意味着，即使有更新版本的 Amazon Linux AMI 可用，系

统也会为更早的 Amazon EMR 版本使用较早的 Amazon Linux AMI 版本。因此，我们建议您使用最新的 Amazon EMR 版本，除非您需要使用早期版本以确保兼容性且无法迁移。如果您必须使用早期版本的 Amazon EMR 来确保兼容性，我们建议您使用系列中的最新版本。例如，如果您必须使用 5.12 系列，请使用 5.12.2，而不是 5.12.0 或 5.12.1。如果系列中有新版本可用，请考虑将您的应用程序迁移到新版本。

有关 Amazon EMR 5.36.0 及更高版本以及 6.6.0 及更高版本引入的自动更新行为的更多信息，请参阅 Amazon EMR 版本的 Amazon Linux 2 自动更新。

默认引导行为不包括内核更新

当基于 Amazon EMR 默认 Amazon Linux AMI 的集群中的 Amazon EC2 实例首次启动时，它会检查 Amazon Linux 的已启用的软件包存储库以及适用于 AMI 版本的软件更新的 Amazon EMR。与其它 Amazon EC2 实例一样，这些存储库中的关键和重要安全更新会自动安装。但是，如果您使用旧版本的 Amazon Linux AMI，则可能无法自动安装最新的安全更新。这是因为 EMR 引用的存储库对于每个版本的 Amazon Linux AMI 都是固定的。同样，请注意，您的网络配置必须允许 HTTP 和 HTTPS 传出到 Amazon S3 中的 Amazon Linux 存储库，否则安全更新将失败。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的程序包存储库。默认情况下，需要重新启动的其它软件包和内核更新（包括 NVIDIA 和 CUDA）将在首次启动时从自动下载中被排除。

> **Important**
>
> 运行 Amazon Linux 或 Amazon Linux 2 AMI（Amazon Linux Machine Image）的 Amazon EMR 集群使用默认的 Amazon Linux 行为，且不会自动下载和安装需要重新启动的重要关键内核更新。这与运行默认 Amazon Linux AMI 的其它 Amazon EC2 实例的行为相同。如果需要重新启动的新 Amazon Linux 软件更新（例如内核、NVIDIA 和 CUDA 更新）在 Amazon EMR 版本发布后可用，则运行默认 AMI 的 Amazon EMR 集群实例不会自动下载和安装这些更新。要获取内核更新，您可以自定义 Amazon EMR AMI，以使用最新的 Amazon Linux AMI。

集群启动时有无更新

请注意，如果由于在集群首次启动时无法访问软件包存储库而导致无法安装软件更新，则集群实例仍会完成启动。例如，存储库可能由于 S3 暂时不可用而无法访问，或者您可能将 VPC 或防火墙规则配置为了阻止访问。

请勿运行 **`sudo yum update`**

当您使用 SSH 连接到集群实例时，屏幕输出的前几行会提供一个链接，该链接指向实例使用的 Amazon Linux AMI 的发布说明、最新 Amazon Linux AMI 版本的通知、可供已启用存储库中的更新使用的软件包数量的通知，以及可用于运行 `sudo yum update` 的指令。

> **Important**
>
> 我们强烈建议您不要在使用 SSH 或使用引导操作连接的同时，在集群实例上运行 `sudo yum update`。这可能会导致不兼容，因为所有软件包都是不加区分地安装的。

软件更新最佳实践

管理软件更新的最佳实践

- 如果您使用早期版本的 Amazon EMR，请考虑对迁移到最新版本进行测试，然后再更新软件包。
- 如果您迁移到更新的版本或升级软件包，请首先在非生产环境中测试实施。使用 Amazon EMR 控制台克隆集群的选项对此大有帮助。
- 逐个为您的应用程序和 Amazon Linux AMI 版本评估软件更新。仅在生产环境中测试和安装程序包，该生产环境是您确定安全状况、应用程序功能或性能时绝对需要的。
- 请访问 Amazon Linux 安全中心，获取更新。
- 避免通过使用 SSH 连接到单个集群实例来安装软件包。相反，根据需要使用引导操作来安装和更新所有集群实例上的软件包。这要求您终止并重新启动集群。有关更多信息，请参阅创建引导操作以安装其它软件。

使用自定义 AMI

使用 Amazon EMR 5.7.0 或更高版本时，您可以选择为 Amazon EMR 指定自定义 Amazon Linux AMI，而不是默认的 Amazon Linux AMI。如果您要执行以下操作，自定义 AMI 会非常有用：

- 预安装应用程序并执行其它自定义设置，而不使用引导操作。这可缩短集群启动时间并简化启动工作流程。有关更多信息以及示例，请参阅 从预置的实例创建自定义 Amazon Linux AMI。
- 实施比引导操作允许的集群和节点配置更高级的集群和节点配置。
- 如果您使用的 Amazon EMR 版本早于 5.24.0，则加密集群中的 EC2 实例的 EBS 根设备卷（引导卷）。与默认的 AMI 一样，自定义 AMI 的最小根卷大小为 10GiB。有关更多信息，请参阅创建带加密 Amazon EBS 根设备卷的自定义 AMI。

> **Note**
>
> 从 Amazon EMR 版本 5.24.0 开始，您指定 Amazon KMS 作为密钥提供程序时，可以使用安全配置选项加密 EBS 根设备和存储卷。有关更多信息，请参阅本地磁盘加密。

自定义 AMI 必须与您创建的集群位于同一Amazon区域中。它还应该与 EC2 实例架构匹配。例如，m5.xlarge 实例具有 x86_64 架构。因此，要使用自定义 AMI 预置 m5.xlarge，您的自定义 AMI 还应具有 x86_64 架构。同样，要预置具有 arm64 架构的 m6g.xlarge 实例，您的自定义 AMI 应该具有 arm64 架构。有关为实例类型识别 Linux AMI 的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的查找 Linux AMI。

> **Important**
>
> 运行 Amazon Linux 或 Amazon Linux 2 AMI（Amazon Linux Machine Image）的 Amazon EMR 集群使用默认的 Amazon Linux 行为，且不会自动下载和安装需要重新启动的重要关键内核更新。这与运行默认 Amazon Linux AMI 的其它 Amazon EC2 实例的行为相同。如果需要重新启动的新 Amazon Linux 软件更新（例如内核、NVIDIA 和 CUDA 更新）在 Amazon EMR 版本发布后可用，则运行默认 AMI 的 Amazon EMR 集群实例不会自动下载和安装这些更新。要获取内核更新，您可以自定义 Amazon EMR AMI，以使用最新的 Amazon Linux AMI。

从预置的实例创建自定义 Amazon Linux AMI

预安装软件并执行其它配置以便为 Amazon EMR 创建自定义 Amazon Linux AMI 的基本步骤如下所示：

- 从基础 Amazon Linux AMI 启动实例。
- 连接到实例以安装软件和执行其它自定义操作。
- 创建已配置实例的新映像 (AMI 快照)。

在基于自定义实例创建映像之后，您可将此映像复制到已加密的目标，如创建带加密 Amazon EBS 根设备卷的自定义 AMI中所述。

教程：从带有已安装自定义软件的实例创建 AMI

基于最新的 Amazon Linux AMI 启动 EC2 实例

1. 使用 Amazon CLI 运行以下命令，这将从现有 AMI 创建实例。将 *MyKeyName* 替换为连接到实例所用的密钥对，并将 *MyAmiId* 替换为相应的 Amazon Linux AMI 的 ID。有关最新的 AMI ID，请参阅 Amazon Linux AMI。

   > **Note**
   >
   > 为了便于读取，包含 Linux 行继续符（\）。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号（^）。

   ```
   aws ec2 run-instances --image-id MyAmiID \
   --count 1 --instance-type m5.xlarge \
   --key-name MyKeyName --region us-west-2
   ```

   `InstanceId` 输出值在下一步中用作 *MyInstanceId*。

2. 运行以下命令：

   ```
   aws ec2 describe-instances --instance-ids MyInstanceId
   ```

   `PublicDnsName` 输出值在下一步中用于连接到实例。

连接到实例并安装软件

1. 使用可让您在 Linux 实例上运行 shell 命令的 SSH 连接。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用 SSH 连接到您的 Linux 实例。
2. 执行任何所需的自定义操作。例如：

   ```
   sudo yum install MySoftwarePackage
   sudo pip install MySoftwarePackage
   ```

从自定义映像创建快照

- 在自定义实例之后，使用 `create-image` 命令从实例创建 AMI。

```
aws ec2 create-image --no-dry-run --instance-id MyInstanceId --name MyEmrCustomAmi
```

在您启动集群或创建加密快照时，将使用 `imageID` 输出值。有关更多信息，请参阅在 EMR 集群中使用单个自定义 AMI和创建带加密 Amazon EBS 根设备卷的自定义 AMI。

如何在 Amazon EMR 集群中使用自定义 AMI

您可以通过以下两种方式使用自定义 AMI 来预置 Amazon EMR 集群：

- 对集群中的所有 EC2 实例，使用单个自定义 AMI。
- 对集群中使用的不同 EC2 实例类型，使用不同的自定义 AMI。

预置 EMR 集群时，您只能使用两个选项中的一个，而且在集群启动后就无法更改它。

在 Amazon EMR 集群中使用单个和多个自定义 AMI 的注意事项

考虑因素	单个自定义 AMI	多个自定义 AMI
在同一集群中使用 x86 和 Graviton2 处理器和自定义 AMI	不支持 X	支持 ✓
AMI 自定义取决于实例类型	不支持 X	支持 ✓
向正在运行的集群添加新的任务实例组/队列时，更改自定义 AMI。注意：您不能更改现有实例组/队列的自定义 AMI。	不支持 X	支持 ✓
使用 Amazon 控制台启动集群	支持 ✓	不支持 X
使用 Amazon CloudFormation 启动集群	支持 ✓	支持 ✓

在 EMR 集群中使用单个自定义 AMI

要在创建集群时指定自定义 AMI ID，请使用以下方法之一：

- Amazon Web Services Management Console
- Amazon CLI
- Amazon EMR 开发工具包
- Amazon EMR API RunJobFlow
- Amazon CloudFormation（请参阅集群 InstanceGroupConfig、集群 InstanceTypeConfig、资源 InstanceGroupConfig 或资源 InstanceFleetConfig-InstanceTypeConfig 的 `CustomAmiID` 属性）

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台指定单个自定义 AMI

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Name and applications（名称和应用程序）下，找到 Operating system options（操作系统选项）。选择 Custom AMI（自定义 AMI），然后在 Custom AMI（自定义 AMI）字段中输入 AMI ID。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台指定单个自定义 AMI

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 依次选择 Create cluster (创建集群)、Go to advanced options (转到高级选项)。
3. 在软件配置下，对于版本，选择 emr-5.7.0 或更高版本，然后选择适合您的应用程序的其他选项。选择 Next (下一步) 。
4. 在 Hardware Configuration (硬件配置) 下，选择适合您的应用程序的值，然后选择 Next (下一步)。
5. 在其他选项中，针对自定义 AMI ID，输入一个值并确保重启时更新所有安装的软件包选项已选中。有关更改更新选项的更多信息，请参阅管理 AMI 程序包存储库更新。
6. 要启动集群，请选择 Next (下一步) 并完成其它配置选项。

Amazon CLI

使用 Amazon CLI 指定单个自定义 AMI

- 在运行 `--custom-ami-id` 命令时，使用 `aws emr create-cluster` 参数可指定 AMI ID。

  以下示例指定一个将单个自定义 AMI 与 20 GiB 启动卷结合使用的集群。有关更多信息，请参阅指定 Amazon EBS 根设备卷大小。

  > **Note**
  >
  > 为了便于读取，包含 Linux 行继续符 (\ ) 。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字符 (^)。

```
aws emr create-cluster --name "Cluster with My Custom AMI" \
--custom-ami-id MyAmiID --ebs-root-volume-size 20 \
--release-label emr-5.7.0 --use-default-roles \
--instance-count 2 --instance-type m5.xlarge
```

在 Amazon EMR 集群中使用多个自定义 AMI

要使用多个自定义 AMI 创建群集，请使用以下方法之一：

- Amazon CLI 版本 1.20.21 或更高版本
- Amazon SDK
- Amazon EMR API 参考中的 Amazon EMR RunJobFlow
- Amazon CloudFormation（请参阅集群 InstanceGroupConfig、集群 InstanceTypeConfig、资源 InstanceGroupConfig 或资源 InstanceFleetConfig-InstanceTypeConfig 的 `CustomAmiID` 属性）

Amazon 管理控制台目前不支持使用多个自定义 AMI 创建集群。

Example -使用Amazon CLI 通过多个自定义 AMI 创建实例组集群

使用 Amazon CLI 版本 1.20.21 或更高版本，您可以将单个自定义 AMI 分配给整个集群，也可以向集群中每个实例节点分配多个自定义 AMI。

以下示例显示了使用跨节点类型（主节点、核心、任务）使用两种实例类型 (m5.xlarge) 创建的统一实例组集群。每个节点都有多个自定义 AMI。该示例说明了多个自定义 AMI 配置的几个功能：

- 集群级别没有分配自定义 AMI。这是为了避免多个自定义 AMI 和单个自定义 AMI 之间的冲突，这将导致集群启动失败。
- 集群可以在主节点、核心和单个任务节点之间拥有多个自定义 AMI。这允许单个 AMI 自定义，例如：预安装的应用程序、复杂的集群配置和加密的 Amazon EBS 根设备卷。
- 实例组核心节点只能有一种实例类型和相应的自定义 AMI。同样，主节点 (master node) 只能有一种实例类型和相应的自定义 AMI。
- 集群可以有多个任务节点。

```
aws emr create-cluster --instance-groups
InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567
InstanceGroupType=TASK,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-456789
```

Example – 使用 Amazon CLI 版本 1.20.21 或更高版本将任务节点添加到包含多种实例类型和多个自定义 AMI 的正在运行的实例组集群

使用 Amazon CLI 版本 1.20.21 或更高版本，您可以将多个自定义 AMI 添加到实例组（您要将该实例组添加到正在运行的集群）。CustomAmiId参数可以与add-instance-groups命令一起使用，如以下示例所示。请注意，在多个节点中使用相同的多个自定义 AMI ID (ami-123456)。

```
aws emr create-cluster --instance-groups
InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567

{
    "ClusterId": "j-123456",
    ...
}

aws emr add-instance-groups --cluster-id j-123456 --instance-groups
 InstanceGroupType=Task,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
```

Example – 使用 Amazon CLI 版本 1.20.21 或更高版本创建实例集集群、多个自定义 AMI、多种实例类型、按需主节点、按需核心、多个核心和任务节点

```
aws emr create-cluster --instance-fleets
InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge
 CustomAmiId=ami-123456}']
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,C
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,Custo
{InstanceType=m6g.xlarge, CustomAmiId=ami-567890}']
```

Example – 使用 Amazon CLI 版本 1.20.21 或更高版本将任务节点添加到包含多种实例类型和多个自定义 AMI 的正在运行的集群

```
aws emr create-cluster --instance-fleets
InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge
 CustomAmiId=ami-123456}']
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,C
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']

{
    "ClusterId": "j-123456",
```

```
    ...
}

aws emr add-instance-fleet --cluster-id j-123456 --instance-fleet
InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,Custo
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
```

管理 AMI 程序包存储库更新

在首次启动时，默认情况下，Amazon Linux AMI 将在其它服务启动之前连接到程序包存储库以安装安全更新。根据您的环境，在为 Amazon EMR 指定自定义 AMI 时，可以选择禁用这些更新。用于禁用此功能的选项仅在您使用自定义 AMI 时可用。默认情况下，Amazon Linux 内核更新以及其它需要重新启动的软件包不会更新。请注意，您的联网配置必须允许 HTTP 和 HTTPS 传出到 Amazon S3 中的 Amazon Linux 存储库，否则安全更新将失败。

> **Warning**
>
> 我们强烈建议您在指定自定义 AMI 时选择重新启动以更新所有安装的程序包。选择不更新程序包将产生额外的安全风险。

使用 Amazon Web Services Management Console，您可以在选择 Custom AMI（自定义 AMI ID）时选择用于禁用更新的选项。

使用 Amazon CLI，您可以在使用 create-cluster 命令时指定 `--repo-upgrade-on-boot NONE` 以及 `--custom-ami-id`。

使用 Amazon EMR API 时，您可以为 RepoUpgradeOnBoot 参数指定 `NONE`。

创建带加密 Amazon EBS 根设备卷的自定义 AMI

要为 Amazon EMR 加密 Amazon Linux AMI 的 Amazon EBS 根设备卷，请将未加密的 AMI 中的快照镜像复制到已加密的目标。有关创建加密 EBS 卷的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EBS 加密。快照的源 AMI 可以是基础 Amazon Linux AMI，您也可从派生自您自定义的基础 Amazon Linux AMI 的 AMI 复制快照。

> **Note**
>
> 从 Amazon EMR 版本 5.24.0 开始，您指定 Amazon KMS 作为密钥提供程序时，可以使用安全配置选项加密 EBS 根设备和存储卷。有关更多信息，请参阅本地磁盘加密。

您可以使用外部密钥提供程序或 Amazon KMS 密钥来加密 EBS 根卷。至少必须允许 Amazon EMR 使用的服务角色（通常为默认 EMR_DefaultRole）为 Amazon EMR 加密和解密卷，这样才能使用 AMI 创建集群。如果使用 Amazon KMS 作为密钥提供程序，则意味着必须允许以下操作：

- kms:encrypt
- kms:decrypt
- kms:ReEncrypt*
- kms:CreateGrant
- kms:GenerateDataKeyWithoutPlaintext"
- kms:DescribeKey"

执行此操作的最简单方式是添加角色作为密钥用户，如以下教程中所述。如果您需要自定义角色策略，请参考以下示例策略语句。

```
{
"Version": "2012-10-17",
"Statement": [
   {
   "Sid": "EmrDiskEncryptionPolicy",
   "Effect": "Allow",
   "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:CreateGrant",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:DescribeKey"
      ],
   "Resource": [
      "*"
      ]
   }
]
}
```

教程：使用 KMS 密钥创建带加密根设备卷的自定义 AMI

此示例中的第一步是查找 KMS 密钥的 ARN 或创建新的 ARN。有关创建密钥的更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的创建密钥。以下过程说明如何将默认服务角色 `EMR_DefaultRole` 作为密钥用户添加到密钥策略。当创建或编辑密钥时，记下密钥的 ARN 值。随后，在创建 AMI 时将使用此 ARN 更高版本。

使用控制台将 Amazon EC2 的服务角色添加到加密密钥用户列表

1. 登录到 Amazon Web Services Management Console，然后通过以下网址打开 Amazon Key Management Service (Amazon KMS) 控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 选择要使用的 KMS 密钥的别名。
4. 在密钥详细信息页面的 Key Users (密钥用户) 下，选择 Add (添加)。
5. 在 Attach (附加) 对话框中，选择 Amazon EMR 服务角色。默认角色的名称为 `EMR_DefaultRole`。
6. 选择 Attach (附加)。

使用 Amazon CLI 创建加密的 AMI

- 在 Amazon CLI 中使用 `aws ec2 copy-image` 命令创建具有加密 EBS 根设备卷和已修改的密钥的 AMI。将指定的 `--kms-key-id` 值替换为您在早期版本创建或修改的密钥的完整 ARN。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^) 。

```
aws ec2 copy-image --source-image-id MyAmiId \
--source-region us-west-2 --name MyEncryptedEMRAmi \
--encrypted --kms-key-id arn:aws:kms:us-west-2:12345678910:key/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

此命令的输出将提供您创建的 AMI 的 ID (可在创建集群时指定此 ID)。有关更多信息，请参阅在 EMR 集群中使用单个自定义 AMI。您也可选择通过安装软件并执行其它配置来自定义此 AMI。有关更多信息，请参阅从预置的实例创建自定义 Amazon Linux AMI。

最佳实践和注意事项

在为 Amazon EMR 创建自定义 AMI 时，请考虑以下事项：

- Amazon EMR 5.30.0 及更高版本，以及 Amazon EMR 6.x 系列都基于 Amazon Linux 2。对于这些 Amazon EMR 版本，您需要对自定义 AMI 使用基于 Amazon Linux 2 的映像。要查找基本自定义 AMI，请参阅查找 Linux AMI。
- 对于 5.30.0 和 6.x 之前的 Amazon EMR 版本，不支持 Amazon Linux 2 AMI。
- 您必须使用 64 位 Amazon Linux AMI。不支持 32 位 AMI。
- 不支持具有多个 Amazon EBS 卷的 Amazon Linux AMI。
- 使您的自定义项基于由 EBS 支持的最新 Amazon Linux AMI。有关 Amazon Linux AMI 和对应 AMI ID 的列表，请参阅 Amazon Linux AMI。
- 请不要复制现有 Amazon EMR 实例的快照来创建自定义 AMI。这将导致错误。
- 仅支持 HVM 虚拟化类型和与 Amazon EMR 兼容的实例。在您执行 AMI 自定义过程时，务必选择 HVM 镜像和与 Amazon EMR 兼容的实例类型。有关兼容的实例和虚拟化类型，请参阅支持的实例类型。
- 您的服务角色在 AMI 上必须具有启动许可，因此 AMI 必须为公用 AMI，或者您必须是 AMI 的所有者或所有者已与您共享 AMI。
- 在 AMI 上创建与应用程序同名的用户将导致错误 (例如，`hadoop`、`hdfs`、`yarn` 或 `spark`)。
- `/tmp`、`/var` 和 `/emr`（如果它们存在于 AMI 上）的内容将在启动时分别移至 `/mnt/tmp`、`/mnt/var` 和 `/mnt/emr`。文件将保留，但如果有大量数据，则启动时间可能长于预期时间。
- 如果您使用基于 Amazon Linux AMI（创建日期为 2018-08-11）的自定义 Amazon Linux AMI，则 Oozie 服务器无法启动。如果您使用 Oozie，请根据具有不同创建日期的 Amazon Linux AMI ID 创建自定义 AMI。您可以使用以下 Amazon CLI 命令返回所有 2018.03 版本的 HVM Amazon Linux AMI 的镜像 ID 列表以及发布日期，以便您可以根据需要选择合适的 Amazon Linux AMI。将 MyRegion 替换为您的区域标识符，如 us-west-2。

```
aws ec2 --region MyRegion describe-images --owner amazon --query 'Images[?
Name!=`null`]|[?starts_with(Name, `amzn-ami-hvm-2018.03`) == `true`].
[CreationDate,ImageId,Name]' --output text | sort -rk1
```

- 如果您使用的 VPC 采用非标准域名和 AmazonProvidedDNS，则不应使用操作系统 DNS 配置中的 `rotate` 选项。

有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的创建由 Amazon EBS 支持的 Linux AMI。

创建集群时更改 Amazon Linux 发行版

当您使用 Amazon EMR 6.6.0 或更高版本启动集群时，它会自动使用已针对默认 Amazon EMR AMI 验证的最新 Amazon Linux 2 版本。您可以使用 Amazon EMR 控制台或 Amazon CLI 为集群指定不同的 Amazon Linux 发行版。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

在使用新控制台创建集群时更改 Amazon Linux 发行版

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 对于 EMR 版本，选择 emr-6.6.0 或更高版本。
4. 在 Operating system options（操作系统选项）下，选择 Amazon Linux version（Amazon Linux 版本），然后选中 Automatically apply latest Amazon Linux updates（自动应用最新的 Amazon Linux 更新）复选框。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

Old console

在使用旧控制台创建集群时更改 Amazon Linux 发行版

1. 打开 Amazon EMR 控制台（https://console.amazonaws.cn/elasticmapreduce）。
2. 选择 Create cluster（创建集群）并转到 Advanced options（高级选项）。
3. 在软件配置下，对于版本，选择 emr-6.6.0 或更高版本，然后选择适合您的应用程序的其他选项。选择 Next（下一步）。
4. 在 Hardware Configuration (硬件配置) 下，选择适合您的应用程序的值，然后选择 Next (下一步)。
5. 在 Additional Options（其他选项）下，选择 Amazon Linux Release（Amazon Linux 发行版），然后选择适用于您集群的 Amazon Linux 发行版。
6. 要启动集群，请选择 Next (下一步) 并完成其它配置选项。

Amazon CLI

在使用 Amazon CLI 创建集群时更改 Amazon Linux 发行版

- 在您运行 `aws emr` create-cluster 命令时，请使用 `--os-release-label` 参数指定 Amazon Linux 发行版。

```
aws emr create-cluster --name "Cluster with Different Amazon Linux Release" \
--os-release-label 2.0.20210312.1 \
--release-label emr-6.6.0 --use-default-roles \
--instance-count 2 --instance-type m5.xlarge
```

指定 Amazon EBS 根设备卷大小

此选项仅在 Amazon EMR 版本 4.x 及更高版本中提供。当您使用Amazon Web Services Management Console、Amazon CLI 或 Amazon EMR API 创建集群时，可指定从 10GiB（默认值）到 100GiB（最大值）的卷大小。此大小调整仅适用于 EBS 根设备卷，并且适用于集群中的所有实例。它不适用于您在创建集群时为每个实例类型单独指定的存储卷。

有关 Amazon EBS 的更多信息，请参阅 Amazon EC2 根设备卷。

> **Note**
>
> 如果您使用默认 AMI，Amazon EMR 将挂载通用型 SSD（gp2）作为根设备卷类型。自定义 AMI 可能具有其它根设备卷类型。不过，自定义 AMI 的最小根卷大小也是 10GiB。有关更多信息，请参阅使用自定义 AMI。

基于从中运行集群的区域中的卷类型的每月 EBS 费用按小时比例收取 EBS 根设备卷成本。存储卷也是如此。费用按 GB 计算，但如果您以 GiB 为单位指定根卷大小，可能需要在估算值中考虑到这一点 (1 GB 等于 0.931323 GiB)。要估算与集群中的 EBS 根设备卷关联的费用，请使用以下公式：

（$EBS GB/月）* 0.931323 / 30 / 24 * EMR_EBSRootGiB * InstanceCount

例如，采用一个具有一个主节点和一个核心节点并使用带默认 10GiB 根设备卷的基础 Amazon Linux AMI 的集群。如果区域中的 EBS 成本为每月每 GB 0.10 USD，则每实例每小时约 0.00129 USD，集群每小时约 0.00258 USD（每月每 GB 0.10 USD 除以 30 天，除以 24 小时，乘以 10GB，乘以 2 个集群实例）。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台指定 Amazon EBS 根设备卷大小

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster configuration（集群配置）下，转到 Cluster scaling and provisioning option（集群扩展和配置选项）部分。展开 EBS root volume（EBS 根卷）的箭头，然后输入一个介于 10 GiB 和 100 GiB 之间的值。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台指定 Amazon EBS 根设备卷大小

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 在软件配置下，对于版本，选择 Amazon EMR 版本 4.x 或更高版本。选择适用于您应用程序的其他选项，然后选择 Next（下一步）。
5. 在 Hardware Configuration (硬件配置) 下，对于 Root device EBS volume size (根设备 EBS 卷大小)，输入一个介于 10 GiB 和 100 GiB 之间的值。

CLI

使用 Amazon CLI 指定 Amazon EBS 根设备卷大小

- 使用 create-cluster 命令的 `--ebs-root-volume-size` 参数，如以下示例所示。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符（\）。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号（^）。

```
aws emr create-cluster --release-label emr-5.7.0 \
--ebs-root-volume-size 20 --instance-groups InstanceGroupType=MASTER,\
InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge
```

配置集群软件

选择软件版本后，Amazon EMR 使用 Amazon Machine Image（AMI）与 Amazon Linux 在您启动集群时安装选中的软件，如 Hadoop、Spark、Hive 等。Amazon EMR 会定期提供新版本，来添加新功能、新应用程序并提供常规更新。我们建议在可能的情况下尽量使用最新的版本启动集群。在从控制台启动集群时，默认选项为使用最新版本。

有关 Amazon EMR 版本和每个版本可用的软件版本的更多信息，请访问 Amazon EMR 版本指南。有关如何编辑安装在集群上的应用程序和软件的默认配置的更多信息，请转到《Amazon EMR 版本指南》中的配置应用程序。Amazon EMR 版本中包含的开源 Hadoop 和 Spark 生态系统组件的某些版本拥有补丁和改进（详见 Amazon EMR 版本指南中的文档）。

除了可在集群上安装的标准软件和应用程序外，您还可以使用引导操作安装自定义软件。引导操作是启动集群时在实例上运行的脚本，以及在创建集群时添加到集群中的新节点上运行的脚本。此外，要在每个节点上调用 Amazon CLI 命令以将对象从 Amazon S3 复制到集群中的每个节点时，引导操作也非常有用。

> **Note**
>
> 在 Amazon EMR 版本 4.x 及更高版本中，引导操作的用法发生了变化。有关 Amazon EMR AMI 版本 2.x 和 3.x 区别的更多信息，请转到《Amazon EMR 版本指南》中的 4.x 区别概述。

创建引导操作以安装其它软件

您可以使用引导操作安装其它软件或自定义集群实例的配置。引导操作是在 Amazon EMR 使用 Amazon Linux Amazon Machine Image（AMI）启动实例之后在集群上运行的脚本。引导操作在 Amazon EMR 安装您在创建集群时指定的应用程序之前和在集群节点开始处理数据之前运行。如果您向正在运行的集群添加节点，则引导操作也会以相同方式在这些节点上运行。您可以创建自定义引导操作并指定何时创建集群。

Amazon EMR 版本 4.x 不支持适用于 Amazon EMR AMI 版本 2.x 和 3.x 的大多数预定义引导操作。例如，Amazon EMR 版本 4.x 不支持 `configure-Hadoop` 和 `configure-daemons`。相反，Amazon EMR 版本 4.x 自身就提供了此功能。有关如何将引导操作从 Amazon EMR AMI 版本 2.x 和 3.x 迁移到 Amazon EMR 发行版 4.x 的更多信息，请访问《Amazon EMR 发行之南》中的使用 Amazon EMR 的早期 AMI 版本自定义集群和应用程序配置。

引导操作基础

默认以 Hadoop 用户的身份执行引导操作。您可以使用 `sudo` 以根特权执行引导操作。

所有 Amazon EMR 管理界面都支持引导操作。您可以通过从控制台、Amazon CLI 或 API 中提供多个 `bootstrap-actions` 参数的方式，为每个集群指定最多 16 个引导操作。

在 Amazon EMR 控制台，您可以选择在创建集群时指定引导操作。

如果使用 CLI，通过在使用 `create-cluster` 命令创建集群时添加 `--bootstrap-actions` 参数，可将对引导操作脚本的引用传递给 Amazon EMR。

```
--bootstrap-actions Path="s3://mybucket/filename",Args=[arg1,arg2]
```

如果引导操作返回非零错误代码，Amazon EMR 会将其当作故障处理，并终止实例。如果太多实例导致引导操作失败，Amazon EMR 就会终止集群。如果只有几个实例失败，Amazon EMR 会尝试重新分配失败的实例并继续执行。使用集群 `lastStateChangeReason` 错误代码标识引导操作所导致的故障。

带条件运行引导操作

如需仅在主节点上运行引导操作，您可以使用带有某种逻辑的自定义引导操作来确定该节点是否为主节点。

```
#!/bin/bash
if grep isMaster /mnt/var/lib/info/instance.json | grep false;
then
    echo "This is not master node, do nothing,exiting"
    exit 0
fi
echo "This is master, continuing to execute script"
# continue with code logic for master node below
```

以下输出将来自核心节点。

```
This is not master node, do nothing, exiting
```

以下输出将来自主节点。

```
This is master, continuing to execute script
```

要使用此逻辑，请将您的引导操作（包括上述代码）上载到 Amazon S3 存储桶。在 Amazon CLI 上，将 `--bootstrap-actions` 参数添加到 `aws emr create-cluster` API 调用并将引导脚本位置指定为 `Path`。

关闭操作

引导操作脚本可以通过向 `/mnt/var/lib/instance-controller/public/shutdown-actions/` 目录编写的方式，创建一个或多个关闭操作。当集群终止时，此目录上的所有脚本会并行执行。每个脚本必须在 60 秒钟内运行和完成。

如果节点因为错误而终止，则不保证关闭操作脚本的运行。

> **Note**
>
> 使用 Amazon EMR 版本 4.0 及更高版本时，必须在主节点上手动创建 `/mnt/var/lib/instance-controller/public/shutdown-actions/` 目录。默认情况下，该目录不存在；但一经创建，该目录中的脚本可在关机前运行。有关连接主节点以创建目录的更多信息，请参阅使用 SSH 连接到主节点。

使用自定义引导操作

您可以创建自定义脚本以执行自定义引导操作。任何 Amazon EMR 界面都可以引用自定义引导操作。

> **Note**
>
> 为了获得最佳性能，我们建议您将要与 Amazon EMR 配合使用的自定义引导操作、脚本和其他文件存储在 Amazon S3 存储桶中，该存储桶与您的集群位于相同的 Amazon Web Services 区域 中。

目录

- 添加自定义引导操作
- 使用自定义引导操作将对象从 Amazon S3 复制到每个节点

添加自定义引导操作

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台通过引导操作创建集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Bootstrap actions（引导操作）下，选择 Add（添加），为操作指定名称、脚本位置和可选参数。选择 Add bootstrap action（添加引导操作）。
4. （可选）添加更多引导操作。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台通过自定义引导操作创建集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 单击 Go to advanced options (转到高级选项)。
4. 在“Create Cluster - Advanced Options, Steps 1 and 2 (创建集群 - 高级选项、步骤 1 和 2)”中，选择所需的选项，然后继续执行 Step 3: General Cluster Settings (步骤 3：常规集群设置)。
5. 在 Bootstrap Actions (引导操作) 下，选择 Configure and add (配置并添加) 来为您的引导操作指定名称、JAR 位置和参数。选择 Add（添加）。
6. (可选) 根据需要添加更多引导操作。
7. 继续创建集群。您的引导操作将在预置和初始化集群后执行。

   当集群主节点正在运行时，您可以连接到主节点，查看引导操作脚本在 `/mnt/var/log/bootstrap-actions/1` 目录中生成的日志文件。

CLI

使用 Amazon CLI 通过自定义引导操作创建集群

使用 Amazon CLI 包含引导操作时，请将 `Path` 和 `Args` 指定为逗号分隔列表。以下示例未使用参数列表。

- 要通过自定义引导操作启动集群，请键入以下命令，并使用您的 EC2 密钥对的名称替换 *myKey*。包含 `--bootstrap-actions` 参数并将引导脚本位置指定为 `Path`。

  - Linux、UNIX 和 Mac OS X 用户：

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://elasticmapreduce/bootstrap-actions/download.sh"
```

  - Windows 用户：

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --use-
default-roles --ec2-attributes KeyName=myKey --applications Name=Hive Name=Pig
 --instance-count 3 --instance-type m5.xlarge --bootstrap-actions Path="s3://
elasticmapreduce/bootstrap-actions/download.sh"
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

> **Note**
>
> 如果您之前未创建默认 Amazon EMR 服务角色和 EC2 实例配置文件，请先键入 `aws emr create-default-roles` 创建它们，然后再键入 `create-cluster` 子命令。

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

使用自定义引导操作将对象从 Amazon S3 复制到每个节点

安装应用程序前，您可以使用引导操作将对象从 Amazon S3 复制到集群中的每个节点。Amazon CLI 安装在集群的每个节点上，因此您的引导操作可以调用 Amazon CLI 命令。

下面的示例演示了一个简单的引导操作脚本，即将一个文件 `myfile.jar` 从 Amazon S3 复制到每个集群节点上的本地文件夹 `/mnt1/myfolder` 中。此脚本保存到 Amazon S3 上，使用的文件名为 `copymyfile.sh`，其中包含以下内容。

```
#!/bin/bash
aws s3 cp s3://mybucket/myfilefolder/myfile.jar /mnt1/myfolder
```

在启动集群时，您可以指定脚本。以下 Amazon CLI 示例展示了这一点：

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.36.1 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://mybucket/myscriptfolder/copymyfile.sh"
```

配置集群硬件和联网

在创建 Amazon EMR 集群时，应重点考虑的是，如何配置 Amazon EC2 实例和网络选项。本章将详细介绍这些选项，然后将它们与最佳实践和指南相关联。

- 节点类型 – EMR 集群中的 Amazon EC2 实例分为三种节点类型，即主节点、核心节点和任务节点。每个节点类型都执行一组由安装在集群上的分布式应用程序定义的角色。例如，在 Hadoop MapReduce 或 Spark 任务期间，核心节点和任务节点上的组件处理数据、将输出传输到 Amazon S3 或 HDFS 并将状态元数据返回到主节点。对于单节点集群，所有组件都在主节点上运行。有关更多信息，请参阅了解节点类型：主节点、核心节点和任务节点。
- EC2 实例 – 创建集群时，您可以选择每种类型节点在哪个 Amazon EC2 实例上运行。EC2 实例类型决定了节点的处理和存储配置文件。为节点选择 Amazon EC2 实例非常重要，因为它决定了集群中各节点类型的性能配置文件。有关更多信息，请参阅配置 Amazon EC2 实例。
- 联网 – 您可以使用公有子网、私有子网或共享子网将 Amazon EMR 集群启动到 VPC。您的联网配置决定了客户和服务如何连接到集群来执行工作、集群如何连接到数据存储和其它Amazon资源，以及用于控制这些连接上的流量的选项。有关更多信息，请参阅配置联网。

- 实例分组 – 托管每个节点类型的 EC2 实例集合称作实例集或统一实例组。实例分组配置是您在创建集群时所做的选择。此选择决定了在集群运行时向集群添加节点的方式。配置适用于所有节点类型。它可以在稍后更改。有关更多信息，请参阅使用实例集或统一实例组创建集群。

> **Note**
>
> 实例集配置仅在 Amazon EMR 发行版 4.8.0 及更高版本（不包括 5.0.0 和 5.0.3）中可用。

了解节点类型：主节点、核心节点和任务节点

通过此节，您可以了解 Amazon EMR 如何使用其中每个节点类型并将其作为集群容量规划的基础。

主节点

主节点管理集群，通常运行分布式应用程序的主要组件。例如，主节点运行 YARN Resource Manager 服务以管理应用程序的资源。此外，它还运行 HDFS NameNode 服务、跟踪提交到集群的任务的状态，并监控实例组的运行状况。

要监控集群的进度并直接与应用程序交互，您可以作为 Hadoop 用户通过 SSH 连接到主节点。有关更多信息，请参阅使用 SSH 连接到主节点。通过连接到主节点，您可以直接访问目录和文件，如 Hadoop 日志文件。有关更多信息，请参阅查看日志文件。您还可以查看应用程序以主节点上运行的网站形式发布的用户界面。有关更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

> **Note**
>
> 使用 Amazon EMR 5.23.0 及更高版本，您可以启动具有三个主节点的集群，以支持应用程序的高可用性（如 YARN Resource Manager、HDFS NameNode、Spark、Hive 和 Ganglia）。使用此功能，主节点不再发生潜在的单点故障。如果其中一个主节点出现故障，Amazon EMR 会自动故障转移到备用主节点，并将出现故障的主节点替换为具有相同配置和引导操作的新主节点。有关更多信息，请参阅计划和配置主节点。

核心节点

核心节点由主节点进行管理。核心节点运行数据节点守护程序以将数据存储作为 Hadoop Distributed File System（HDFS）的一部分进行协调。它们还运行任务跟踪守护程序，并对安装的应用程序所需的数据执行其它并行计算任务。例如，核心节点运行 YARN NodeManager 守护程序、Hadoop MapReduce 任务和 Spark 执行器。

每个集群只能有一个核心实例组或实例机群，但可在实例组或实例机群中的多个 Amazon EC2 实例上运行多个节点。对于实例组，您可以在集群运行时添加和删除 Amazon EC2 实例。您还可以设置弹性伸缩，以根据指标的值来添加实例。有关使用实例组配置添加和删除 Amazon EC2 实例的更多信息，请参阅使用集群扩展。

对于实例集，您可以通过为按需实例与竞价型实例修改实例集的目标容量来高效地添加和删除实例。有关目标容量的更多信息，请参阅实例集选项。

> **Warning**
>
> 从运行的核心节点删除 HDFS 守护程序或终止核心节点将面临数据丢失的风险。配置核心节点以使用竞价型实例时，请务必小心。有关更多信息，请参阅什么时候应该使用竞价型实例？。

任务节点

您可以使用任务节点来支持对数据执行并行计算任务，例如 Hadoop MapReduce 任务和 Spark 执行程序。任务节点不运行数据节点守护程序，也不在 HDFS 中存储数据。与使用核心节点一样，您可以将任务节点添加到集群，方式是将 Amazon EC2 实例添加到现有的统一实例组或修改任务实例机群的目标容量。

对于统一实例组配置，您最多可拥有 48 个任务实例组。通过此方式添加实例组，使您能够将 Amazon EC2 实例类型和定价选项混合在一起，例如按需型实例和 Spot 实例。这使您能够灵活且经济有效地响应工作负载要求。

对于实例集配置，提供了混合实例类型和购买选项的内置功能，因此仅有一个任务实例集。

由于竞价型实例通常用于运行任务节点，而 Amazon EMR 具有调度 YARN 任务的默认功能，因此在竞价型实例上运行的任务节点终止时，正在运行的任务不会失败。Amazon EMR 通过允许应用程序主进程仅在核心节点上运行来实现这一目标。应用程序主进程控制正在运行的任务，并且需要在任务的整个生命周期内保持活动状态。

Amazon EMR 发行版 5.19.0 及更高版本使用内置 YARN 节点标签功能来实现这一目标。（早期版本使用代码补丁）。`yarn-site` 中的属性和 `capacity-scheduler` 配置分类是默认配置，以便 YARN capacity-scheduler 和 fair-scheduler 利用节点标注。Amazon EMR 自动为核心节点添加 `CORE` 标注，并设置属性，以便只在具有 CORE 标签的节点上安排应用程序主节点。手动修改 yarn-site 和 capacity-scheduler 配置分类中的相关属性，或直接在关联的 XML 文件中修改相关属性，可能会破坏此功能或调整此功能。

从 Amazon EMR 6.x 发行版系列开始，默认情况下禁用 YARN 节点标注功能。默认情况下，应用程序主进程可以在核心节点和任务节点上运行。您可以通过配置以下属性来启用 YARN 节点标注功能：

- `yarn.node-labels.enabled: true`
- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

有关特定属性的信息，请参阅用于防止因任务节点竞价型实例终止而导致任务失败的 Amazon EMR 设置。

配置 Amazon EC2 实例

EC2 实例附带了不同的配置，称作实例类型。实例类型均有不同的 CPU、输入/输出和存储容量。除了实例类型之外，您还可以为 Amazon EC2 实例选择不同的购买选项。您可以在统一实例组或实例集中指定不同的实例类型和购买选项。有关更多信息，请参阅使用实例集或统一实例组创建集群。有关为应用程序选择实例类型和购买选项的指导，请参阅 集群配置的最佳实践。

> **Important**
>
> 使用Amazon Web Services Management Console选择实例类型时，每个实例类型显示的 vCPU 数量是该实例的 YARN vcore 数量，而不是该实例类型的 EC2 vCPU 数量。有关每个实例类型 vCPU 数量的更多信息，请参阅 Amazon EC2 实例类型。

主题

- 支持的实例类型
- 配置联网
- 使用实例集或统一实例组创建集群

支持的实例类型

本节介绍 Amazon EMR 支持的实例类型（按 Amazon Web Services 区域 整理）。要了解有关实例类型的更多信息，请参阅 Amazon EC2 实例和 Amazon Linux AMI 实例类型矩阵。

并非所有实例类型在所有区域内都可用，实例可用性取决于指定区域和可用区的可用性和需求。实例的可用区由您用于启动集群的子网决定。

注意事项

为 Amazon EMR 集群选择实例类型时，请考虑以下事项。

> **Important**
>
> 使用Amazon Web Services Management Console选择实例类型时，每个实例类型显示的 vCPU 数量是该实例的 YARN vcore 数量，而不是该实例类型的 EC2 vCPU 数量。有关每个实例类型 vCPU 数量的更多信息，请参阅 Amazon EC2 实例类型。

- 如果您使用在指定区域和可用区中不可用的实例类型创建集群，则您的集群可能无法预置或可能预置失败。有关实例可用性的信息，请参阅 Amazon EMR 定价页面或者查看此页面上的 支持的实例类型（按 Amazon Web Services 区域 整理） 表格。
- 从 Amazon EMR 5.13.0 版本开始，所有实例均为根卷使用 HVM 虚拟化和 EBS 支持的存储。采用 5.13.0 之前的 Amazon EMR 发行版时，有些上一代实例使用 PVM 虚拟化。有关更多信息，请参阅 Linux AMI 虚拟化类型。
- 一些实例类型支持增强联网。有关更多信息，请参阅有关 Linux 的增强联网。
- NVIDIA 和 CUDA 驱动程序默认安装在 GPU 实例类型上。

支持的实例类型（按 Amazon Web Services 区域 整理）

下表介绍 Amazon EMR 支持的实例类型（按 Amazon Web Services 区域 整理）。该表还列出了支持各实例类型的 4.x、5.x 或 6.x 系列中最早的 Amazon EMR 发行版。例如，在 4.8.3 及更高版本、5.2.1 及更高版本和 6.0.0 及更高版本中，EMR 支持在美国东部（弗吉尼亚州北部）内可用的 m4.16xlarge 实例。

美国东部（弗吉尼亚北部）- us-east-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
	p5.48xlarge	emr-6.14.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6a.48xlarge	emr-5.36.1, emr-6.8.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d3en.6xlarge	emr-5.33.0, emr-6.3.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4g.4xlarge	emr-5.36.1, emr-6.10.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

美国东部（俄亥俄）- us-east-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6a.16xlarge	emr-5.36.1, emr-6.8.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.6.0, emr-5.0.3, emr-6.0.0
	d2.2xlarge	emr-4.6.0, emr-5.0.3, emr-6.0.0
	d2.4xlarge	emr-4.6.0, emr-5.0.3, emr-6.0.0
	d2.8xlarge	emr-4.6.0, emr-5.0.3, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	h1.2xlarge	emr-5.17.0, emr-6.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

美国西部（加利福尼亚北部）– us-west-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

美国西部（俄勒冈）- us-west-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
	p5.48xlarge	emr-6.14.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

Amazon GovCloud（美国西部）– us-gov-west-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Amazon GovCloud（美国东部）– us-gov-east-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.16.0, emr-6.0.0
	m5.2xlarge	emr-5.16.0, emr-6.0.0
	m5.4xlarge	emr-5.16.0, emr-6.0.0
	m5.8xlarge	emr-5.16.0, emr-6.0.0
	m5.12xlarge	emr-5.16.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.16xlarge	emr-5.16.0, emr-6.0.0
	m5.24xlarge	emr-5.16.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.16.0, emr-6.0.0
	m5d.2xlarge	emr-5.16.0, emr-6.0.0
	m5d.4xlarge	emr-5.16.0, emr-6.0.0
	m5d.8xlarge	emr-5.16.0, emr-6.0.0
	m5d.12xlarge	emr-5.16.0, emr-6.0.0
	m5d.16xlarge	emr-5.16.0, emr-6.0.0
	m5d.24xlarge	emr-5.16.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.16.0, emr-6.0.0
	c5.2xlarge	emr-5.16.0, emr-6.0.0
	c5.4xlarge	emr-5.16.0, emr-6.0.0
	c5.9xlarge	emr-5.16.0, emr-6.0.0
	c5.12xlarge	emr-5.16.0, emr-6.0.0
	c5.18xlarge	emr-5.16.0, emr-6.0.0
	c5.24xlarge	emr-5.16.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.16.0, emr-6.0.0
	c5d.2xlarge	emr-5.16.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.4xlarge	emr-5.16.0, emr-6.0.0
	c5d.9xlarge	emr-5.16.0, emr-6.0.0
	c5d.18xlarge	emr-5.16.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.16.0, emr-6.0.0
	r5.2xlarge	emr-5.16.0, emr-6.0.0
	r5.4xlarge	emr-5.16.0, emr-6.0.0
	r5.8xlarge	emr-5.16.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.12xlarge	emr-5.16.0, emr-6.0.0
	r5.16xlarge	emr-5.16.0, emr-6.0.0
	r5.24xlarge	emr-5.16.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.16.0, emr-6.0.0
	r5d.2xlarge	emr-5.16.0, emr-6.0.0
	r5d.4xlarge	emr-5.16.0, emr-6.0.0
	r5d.8xlarge	emr-5.16.0, emr-6.0.0
	r5d.12xlarge	emr-5.16.0, emr-6.0.0
	r5d.16xlarge	emr-5.16.0, emr-6.0.0
	r5d.24xlarge	emr-5.16.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-5.16.0, emr-6.0.0
	d2.2xlarge	emr-5.16.0, emr-6.0.0
	d2.4xlarge	emr-5.16.0, emr-6.0.0
	d2.8xlarge	emr-5.16.0, emr-6.0.0
	i3.xlarge	emr-5.16.0, emr-6.0.0
	i3.2xlarge	emr-5.16.0, emr-6.0.0
	i3.4xlarge	emr-5.16.0, emr-6.0.0
	i3.8xlarge	emr-5.16.0, emr-6.0.0
	i3.16xlarge	emr-5.16.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

非洲（开普敦）- af-south-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.29.0, emr-6.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0
	m5.12xlarge	emr-5.29.0, emr-6.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0
	m5d.12xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.16xlarge	emr-5.29.0, emr-6.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.29.0, emr-6.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0
	c5.12xlarge	emr-5.29.0, emr-6.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.29.0, emr-6.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.8xlarge	emr-5.29.0, emr-6.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
存储优化	d2.xlarge	emr-5.29.0, emr-6.0.0
	d2.2xlarge	emr-5.29.0, emr-6.0.0
	d2.4xlarge	emr-5.29.0, emr-6.0.0
	d2.8xlarge	emr-5.29.0, emr-6.0.0
	i3.xlarge	emr-5.29.0, emr-6.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0
	i3en.3xlarge	emr-5.29.0, emr-6.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.12xlarge	emr-5.29.0, emr-6.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太地区（香港）- ap-east-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.20.0, emr-6.0.0
	m5.2xlarge	emr-5.20.0, emr-6.0.0
	m5.4xlarge	emr-5.20.0, emr-6.0.0
	m5.8xlarge	emr-5.20.0, emr-6.0.0
	m5.12xlarge	emr-5.20.0, emr-6.0.0
	m5.16xlarge	emr-5.20.0, emr-6.0.0
	m5.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.2xlarge	emr-5.20.0, emr-6.0.0
	m5d.4xlarge	emr-5.20.0, emr-6.0.0
	m5d.8xlarge	emr-5.20.0, emr-6.0.0
	m5d.12xlarge	emr-5.20.0, emr-6.0.0
	m5d.16xlarge	emr-5.20.0, emr-6.0.0
	m5d.24xlarge	emr-5.20.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
计算优化	c5.xlarge	emr-5.20.0, emr-6.0.0
	c5.2xlarge	emr-5.20.0, emr-6.0.0
	c5.4xlarge	emr-5.20.0, emr-6.0.0
	c5.9xlarge	emr-5.20.0, emr-6.0.0
	c5.12xlarge	emr-5.20.0, emr-6.0.0
	c5.18xlarge	emr-5.20.0, emr-6.0.0
	c5.24xlarge	emr-5.20.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.20.0, emr-6.0.0
	c5d.2xlarge	emr-5.20.0, emr-6.0.0
	c5d.4xlarge	emr-5.20.0, emr-6.0.0
	c5d.9xlarge	emr-5.20.0, emr-6.0.0
	c5d.18xlarge	emr-5.20.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.20.0, emr-6.0.0
	r5.2xlarge	emr-5.20.0, emr-6.0.0
	r5.4xlarge	emr-5.20.0, emr-6.0.0
	r5.8xlarge	emr-5.20.0, emr-6.0.0
	r5.12xlarge	emr-5.20.0, emr-6.0.0
	r5.16xlarge	emr-5.20.0, emr-6.0.0
	r5.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.20.0, emr-6.0.0
	r5d.2xlarge	emr-5.20.0, emr-6.0.0
	r5d.4xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.8xlarge	emr-5.20.0, emr-6.0.0
	r5d.12xlarge	emr-5.20.0, emr-6.0.0
	r5d.16xlarge	emr-5.20.0, emr-6.0.0
	r5d.24xlarge	emr-5.20.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-5.20.0, emr-6.0.0
	d2.2xlarge	emr-5.20.0, emr-6.0.0
	d2.4xlarge	emr-5.20.0, emr-6.0.0
	d2.8xlarge	emr-5.20.0, emr-6.0.0
	i3.xlarge	emr-5.20.0, emr-6.0.0
	i3.2xlarge	emr-5.20.0, emr-6.0.0
	i3.4xlarge	emr-5.20.0, emr-6.0.0
	i3.8xlarge	emr-5.20.0, emr-6.0.0
	i3.16xlarge	emr-5.20.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太地区（雅加达）– ap-southeast-3

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.30.2, emr-6.0.1
	m5.2xlarge	emr-5.30.2, emr-6.0.1
	m5.4xlarge	emr-5.30.2, emr-6.0.1
	m5.8xlarge	emr-5.30.2, emr-6.0.1
	m5.12xlarge	emr-5.30.2, emr-6.0.1
	m5.16xlarge	emr-5.30.2, emr-6.0.1
	m5.24xlarge	emr-5.30.2, emr-6.0.1
	m5a.xlarge	emr-5.30.2, emr-6.0.1

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5a.2xlarge	emr-5.30.2, emr-6.0.1
	m5a.4xlarge	emr-5.30.2, emr-6.0.1
	m5a.8xlarge	emr-5.30.2, emr-6.0.1
	m5a.12xlarge	emr-5.30.2, emr-6.0.1
	m5a.16xlarge	emr-5.30.2, emr-6.0.1
	m5a.24xlarge	emr-5.30.2, emr-6.0.1
	m5d.xlarge	emr-5.30.2, emr-6.0.1
	m5d.2xlarge	emr-5.30.2, emr-6.0.1
	m5d.4xlarge	emr-5.30.2, emr-6.0.1
	m5d.8xlarge	emr-5.30.2, emr-6.0.1
	m5d.12xlarge	emr-5.30.2, emr-6.0.1
	m5d.16xlarge	emr-5.30.2, emr-6.0.1
	m5d.24xlarge	emr-5.30.2, emr-6.0.1
	m6g.xlarge	emr-5.30.2, emr-6.1.1
	m6g.2xlarge	emr-5.30.2, emr-6.1.1
	m6g.4xlarge	emr-5.30.2, emr-6.1.1
	m6g.8xlarge	emr-5.30.2, emr-6.1.1
	m6g.12xlarge	emr-5.30.2, emr-6.1.1
	m6g.16xlarge	emr-5.30.2, emr-6.1.1
	m6gd.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.30.2, emr-6.0.1
	c5.2xlarge	emr-5.30.2, emr-6.0.1
	c5.4xlarge	emr-5.30.2, emr-6.0.1
	c5.9xlarge	emr-5.30.2, emr-6.0.1
	c5.12xlarge	emr-5.30.2, emr-6.0.1
	c5.18xlarge	emr-5.30.2, emr-6.0.1
	c5.24xlarge	emr-5.30.2, emr-6.0.1

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.xlarge	emr-5.30.2, emr-6.0.1
	c5d.2xlarge	emr-5.30.2, emr-6.0.1
	c5d.4xlarge	emr-5.30.2, emr-6.0.1
	c5d.9xlarge	emr-5.30.2, emr-6.0.1
	c5d.12xlarge	emr-5.30.2, emr-6.0.1
	c5d.18xlarge	emr-5.30.2, emr-6.0.1
	c5d.24xlarge	emr-5.30.2, emr-6.0.1
	c5n.xlarge	emr-5.30.2, emr-6.0.1
	c5n.2xlarge	emr-5.30.2, emr-6.0.1
	c5n.4xlarge	emr-5.30.2, emr-6.0.1
	c5n.9xlarge	emr-5.30.2, emr-6.0.1
	c5n.18xlarge	emr-5.30.2, emr-6.0.1
	c6g.xlarge	emr-5.31.1, emr-6.1.1
	c6g.2xlarge	emr-5.31.1, emr-6.1.1
	c6g.4xlarge	emr-5.31.1, emr-6.1.1
	c6g.8xlarge	emr-5.31.1, emr-6.1.1
	c6g.12xlarge	emr-5.31.1, emr-6.1.1
	c6g.16xlarge	emr-5.31.1, emr-6.1.1
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
内存优化	r5.xlarge	emr-5.30.2, emr-6.0.1
	r5.2xlarge	emr-5.30.2, emr-6.0.1
	r5.4xlarge	emr-5.30.2, emr-6.0.1
	r5.8xlarge	emr-5.30.2, emr-6.0.1
	r5.12xlarge	emr-5.30.2, emr-6.0.1
	r5.16xlarge	emr-5.30.2, emr-6.0.1
	r5.24xlarge	emr-5.30.2, emr-6.0.1
	r5d.xlarge	emr-5.30.2, emr-6.0.1
	r5d.2xlarge	emr-5.30.2, emr-6.0.1
	r5d.4xlarge	emr-5.30.2, emr-6.0.1
	r5d.8xlarge	emr-5.30.2, emr-6.0.1
	r5d.12xlarge	emr-5.30.2, emr-6.0.1
	r5d.16xlarge	emr-5.30.2, emr-6.0.1
	r5d.24xlarge	emr-5.30.2, emr-6.0.1
	r6g.xlarge	emr-5.31.1, emr-6.1.1
	r6g.2xlarge	emr-5.31.1, emr-6.1.1
	r6g.4xlarge	emr-5.31.1, emr-6.1.1
	r6g.8xlarge	emr-5.31.1, emr-6.1.1
	r6g.12xlarge	emr-5.31.1, emr-6.1.1
	r6g.16xlarge	emr-5.31.1, emr-6.1.1

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-5.30.2, emr-6.0.1
	d2.2xlarge	emr-5.30.2, emr-6.0.1
	d2.4xlarge	emr-5.30.2, emr-6.0.1
	d2.8xlarge	emr-5.30.2, emr-6.0.1

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3.xlarge	emr-5.30.2, emr-6.0.1
	i3.2xlarge	emr-5.30.2, emr-6.0.1
	i3.4xlarge	emr-5.30.2, emr-6.0.1
	i3.8xlarge	emr-5.30.2, emr-6.0.1
	i3.16xlarge	emr-5.30.2, emr-6.0.1
	i3en.xlarge	emr-5.30.2, emr-6.0.1
	i3en.2xlarge	emr-5.30.2, emr-6.0.1
	i3en.3xlarge	emr-5.30.2, emr-6.0.1
	i3en.6xlarge	emr-5.30.2, emr-6.0.1
	i3en.12xlarge	emr-5.30.2, emr-6.0.1
	i3en.24xlarge	emr-5.30.2, emr-6.0.1
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太地区（孟买）– ap-south-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d2.4xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太（海得拉巴）- ap-south-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.36.0, emr-6.7.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.16xlarge	emr-5.36.0, emr-6.7.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0
计算优化	c5.xlarge	emr-5.36.0, emr-6.7.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.4xlarge	emr-5.36.0, emr-6.7.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0
	c6i.xlarge	emr-5.36.0, emr-6.7.0
	c6i.2xlarge	emr-5.36.0, emr-6.7.0
	c6i.4xlarge	emr-5.36.0, emr-6.7.0
	c6i.8xlarge	emr-5.36.0, emr-6.7.0
	c6i.12xlarge	emr-5.36.0, emr-6.7.0
	c6i.16xlarge	emr-5.36.0, emr-6.7.0
	c6i.24xlarge	emr-5.36.0, emr-6.7.0
	c6i.32xlarge	emr-5.36.0, emr-6.7.0
内存优化	r5.xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.2xlarge	emr-5.36.0, emr-6.7.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.2xlarge	emr-5.36.0, emr-6.7.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	i3.xlarge	emr-5.36.0, emr-6.7.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3.8xlarge	emr-5.36.0, emr-6.7.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太地区（大阪）– ap-northeast-3

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-5.10.0, emr-6.0.0
	d2.2xlarge	emr-5.10.0, emr-6.0.0
	d2.4xlarge	emr-5.10.0, emr-6.0.0
	d2.8xlarge	emr-5.10.0, emr-6.0.0
	i3.xlarge	emr-5.10.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3.2xlarge	emr-5.10.0, emr-6.0.0
	i3.4xlarge	emr-5.10.0, emr-6.0.0
	i3.8xlarge	emr-5.10.0, emr-6.0.0
	i3.16xlarge	emr-5.10.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太地区（首尔）– ap-northeast-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.1.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.1.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.1.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.1.0, emr-5.0.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

亚太地区（新加坡）– ap-southeast-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

亚太地区（悉尼）- ap-southeast-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

亚太地区（东京）- ap-northeast-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

加拿大（中部）– ca-central-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-4.8.2, emr-5.1.0, emr-6.0.0
	d2.2xlarge	emr-4.8.2, emr-5.1.0, emr-6.0.0
	d2.4xlarge	emr-4.8.2, emr-5.1.0, emr-6.0.0
	d2.8xlarge	emr-4.8.2, emr-5.1.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

中国（宁夏）- cn-northwest-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.9.1, emr-5.5.3, emr-6.0.0
	d2.2xlarge	emr-4.9.1, emr-5.5.3, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d2.4xlarge	emr-4.9.1, emr-5.5.3, emr-6.0.0
	d2.8xlarge	emr-4.9.1, emr-5.5.3, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0

中国（北京）- cn-north-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0

欧洲（法兰克福）– eu-central-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c7gd.xlarge	emr-5.36.1, emr-6.10.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d3en.4xlarge	emr-5.33.0, emr-6.3.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

欧洲（苏黎世）- eu-central-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.36.0, emr-6.7.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.24xlarge	emr-5.36.0, emr-6.7.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.xlarge	emr-5.36.0, emr-6.7.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0
计算优化	c5.xlarge	emr-5.36.0, emr-6.7.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.18xlarge	emr-5.36.0, emr-6.7.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
内存优化	r5.xlarge	emr-5.36.0, emr-6.7.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.12xlarge	emr-5.36.0, emr-6.7.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.12xlarge	emr-5.36.0, emr-6.7.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	i3.xlarge	emr-5.36.0, emr-6.7.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0

欧洲（爱尔兰）– eu-west-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5n.12xlarge	emr-5.31.0, emr-6.1.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6a.16xlarge	emr-5.36.1, emr-6.8.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
存储优化	d2.xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.2xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.4xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d2.8xlarge	emr-4.0.0, emr-5.0.0, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	d3en.xlarge	emr-5.33.0, emr-6.3.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

欧洲（伦敦）– eu-west-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
存储优化	d2.xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0
	d2.2xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0
	d2.4xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0
	d2.8xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

欧洲地区（米兰）- eu-south-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.29.0, emr-6.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0
	m5.12xlarge	emr-5.29.0, emr-6.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0
	m5a.xlarge	emr-5.29.0, emr-6.0.0
	m5a.2xlarge	emr-5.29.0, emr-6.0.0
	m5a.4xlarge	emr-5.29.0, emr-6.0.0
	m5a.8xlarge	emr-5.29.0, emr-6.0.0
	m5a.12xlarge	emr-5.29.0, emr-6.0.0
	m5a.16xlarge	emr-5.29.0, emr-6.0.0
	m5a.24xlarge	emr-5.29.0, emr-6.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0
	m5d.12xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.16xlarge	emr-5.29.0, emr-6.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.29.0, emr-6.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.12xlarge	emr-5.29.0, emr-6.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.9xlarge	emr-5.29.0, emr-6.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.29.0, emr-6.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0
	r5.8xlarge	emr-5.29.0, emr-6.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0
	r5a.xlarge	emr-5.29.0, emr-6.0.0
	r5a.2xlarge	emr-5.29.0, emr-6.0.0
	r5a.4xlarge	emr-5.29.0, emr-6.0.0
	r5a.8xlarge	emr-5.29.0, emr-6.0.0
	r5a.12xlarge	emr-5.29.0, emr-6.0.0
	r5a.16xlarge	emr-5.29.0, emr-6.0.0
	r5a.24xlarge	emr-5.29.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-5.29.0, emr-6.0.0
	d2.2xlarge	emr-5.29.0, emr-6.0.0
	d2.4xlarge	emr-5.29.0, emr-6.0.0
	d2.8xlarge	emr-5.29.0, emr-6.0.0
	i3.xlarge	emr-5.29.0, emr-6.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.3xlarge	emr-5.29.0, emr-6.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

欧洲（西班牙）- eu-south-2

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.36.0, emr-6.7.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.24xlarge	emr-5.36.0, emr-6.7.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
计算优化	c5.xlarge	emr-5.36.0, emr-6.7.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
内存优化	r5.xlarge	emr-5.36.0, emr-6.7.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	i3.xlarge	emr-5.36.0, emr-6.7.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0

欧洲（巴黎）– eu-west-3

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0
	d2.2xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0
	d2.4xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0
	d2.8xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

欧洲（斯德哥尔摩）– eu-north-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.17.0, emr-6.0.0
	m5.2xlarge	emr-5.17.0, emr-6.0.0
	m5.4xlarge	emr-5.17.0, emr-6.0.0
	m5.8xlarge	emr-5.17.0, emr-6.0.0
	m5.12xlarge	emr-5.17.0, emr-6.0.0
	m5.16xlarge	emr-5.17.0, emr-6.0.0
	m5.24xlarge	emr-5.17.0, emr-6.0.0
	m5d.xlarge	emr-5.17.0, emr-6.0.0
	m5d.2xlarge	emr-5.17.0, emr-6.0.0
	m5d.4xlarge	emr-5.17.0, emr-6.0.0
	m5d.8xlarge	emr-5.17.0, emr-6.0.0
	m5d.12xlarge	emr-5.17.0, emr-6.0.0
	m5d.16xlarge	emr-5.17.0, emr-6.0.0
	m5d.24xlarge	emr-5.17.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.17.0, emr-6.0.0
	c5.2xlarge	emr-5.17.0, emr-6.0.0
	c5.4xlarge	emr-5.17.0, emr-6.0.0
	c5.9xlarge	emr-5.17.0, emr-6.0.0
	c5.12xlarge	emr-5.17.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.18xlarge	emr-5.17.0, emr-6.0.0
	c5.24xlarge	emr-5.17.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.17.0, emr-6.0.0
	c5d.2xlarge	emr-5.17.0, emr-6.0.0
	c5d.4xlarge	emr-5.17.0, emr-6.0.0
	c5d.9xlarge	emr-5.17.0, emr-6.0.0
	c5d.12xlarge	emr-5.17.0, emr-6.0.0
	c5d.18xlarge	emr-5.17.0, emr-6.0.0
	c5d.24xlarge	emr-5.17.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
内存优化	r5.xlarge	emr-5.17.0, emr-6.0.0
	r5.2xlarge	emr-5.17.0, emr-6.0.0
	r5.4xlarge	emr-5.17.0, emr-6.0.0
	r5.8xlarge	emr-5.17.0, emr-6.0.0
	r5.12xlarge	emr-5.17.0, emr-6.0.0
	r5.16xlarge	emr-5.17.0, emr-6.0.0
	r5.24xlarge	emr-5.17.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.17.0, emr-6.0.0
	r5d.2xlarge	emr-5.17.0, emr-6.0.0
	r5d.4xlarge	emr-5.17.0, emr-6.0.0
	r5d.8xlarge	emr-5.17.0, emr-6.0.0
	r5d.12xlarge	emr-5.17.0, emr-6.0.0
	r5d.16xlarge	emr-5.17.0, emr-6.0.0
	r5d.24xlarge	emr-5.17.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	d2.xlarge	emr-5.17.0, emr-6.0.0
	d2.2xlarge	emr-5.17.0, emr-6.0.0
	d2.4xlarge	emr-5.17.0, emr-6.0.0
	d2.8xlarge	emr-5.17.0, emr-6.0.0
	i3.xlarge	emr-5.17.0, emr-6.0.0
	i3.2xlarge	emr-5.17.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3.4xlarge	emr-5.17.0, emr-6.0.0
	i3.8xlarge	emr-5.17.0, emr-6.0.0
	i3.16xlarge	emr-5.17.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

中东（巴林）- me-south-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.24.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5.2xlarge	emr-5.24.0, emr-6.0.0
	m5.4xlarge	emr-5.24.0, emr-6.0.0
	m5.8xlarge	emr-5.24.0, emr-6.0.0
	m5.12xlarge	emr-5.24.0, emr-6.0.0
	m5.16xlarge	emr-5.24.0, emr-6.0.0
	m5.24xlarge	emr-5.24.0, emr-6.0.0
	m5d.xlarge	emr-5.24.0, emr-6.0.0
	m5d.2xlarge	emr-5.24.0, emr-6.0.0
	m5d.4xlarge	emr-5.24.0, emr-6.0.0
	m5d.8xlarge	emr-5.24.0, emr-6.0.0
	m5d.12xlarge	emr-5.24.0, emr-6.0.0
	m5d.16xlarge	emr-5.24.0, emr-6.0.0
	m5d.24xlarge	emr-5.24.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.24.0, emr-6.0.0
	c5.2xlarge	emr-5.24.0, emr-6.0.0
	c5.4xlarge	emr-5.24.0, emr-6.0.0
	c5.9xlarge	emr-5.24.0, emr-6.0.0
	c5.12xlarge	emr-5.24.0, emr-6.0.0
	c5.18xlarge	emr-5.24.0, emr-6.0.0
	c5.24xlarge	emr-5.24.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.24.0, emr-6.0.0
	c5d.2xlarge	emr-5.24.0, emr-6.0.0
	c5d.4xlarge	emr-5.24.0, emr-6.0.0
	c5d.9xlarge	emr-5.24.0, emr-6.0.0
	c5d.18xlarge	emr-5.24.0, emr-6.0.0
	c5n.xlarge	emr-5.24.0, emr-6.0.0
	c5n.2xlarge	emr-5.24.0, emr-6.0.0
	c5n.4xlarge	emr-5.24.0, emr-6.0.0
	c5n.9xlarge	emr-5.24.0, emr-6.0.0
	c5n.18xlarge	emr-5.24.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
内存优化	r5.xlarge	emr-5.24.0, emr-6.0.0
	r5.2xlarge	emr-5.24.0, emr-6.0.0
	r5.4xlarge	emr-5.24.0, emr-6.0.0
	r5.8xlarge	emr-5.24.0, emr-6.0.0
	r5.12xlarge	emr-5.24.0, emr-6.0.0
	r5.16xlarge	emr-5.24.0, emr-6.0.0
	r5.24xlarge	emr-5.24.0, emr-6.0.0
	r5d.xlarge	emr-5.24.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5d.2xlarge	emr-5.24.0, emr-6.0.0
	r5d.4xlarge	emr-5.24.0, emr-6.0.0
	r5d.8xlarge	emr-5.24.0, emr-6.0.0
	r5d.12xlarge	emr-5.24.0, emr-6.0.0
	r5d.16xlarge	emr-5.24.0, emr-6.0.0
	r5d.24xlarge	emr-5.24.0, emr-6.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
存储优化	d2.xlarge	emr-5.24.0, emr-6.0.0
	d2.2xlarge	emr-5.24.0, emr-6.0.0
	d2.4xlarge	emr-5.24.0, emr-6.0.0
	d2.8xlarge	emr-5.24.0, emr-6.0.0
	i3.xlarge	emr-5.24.0, emr-6.0.0
	i3.2xlarge	emr-5.24.0, emr-6.0.0
	i3.4xlarge	emr-5.24.0, emr-6.0.0
	i3.8xlarge	emr-5.24.0, emr-6.0.0
	i3.16xlarge	emr-5.24.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

中东（阿联酋）- me-central-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.36.0, emr-6.7.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6g.2xlarge	emr-5.36.0, emr-6.7.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0
计算优化	c5.xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5.2xlarge	emr-5.36.0, emr-6.7.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
内存优化	r5.xlarge	emr-5.36.0, emr-6.7.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5.16xlarge	emr-5.36.0, emr-6.7.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6i.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	i3.xlarge	emr-5.36.0, emr-6.7.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

南美洲（圣保罗）– sa-east-1

实例类	实例类型	支持的 Amazon EMR 最低版本
通用	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
计算优化	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
加速计算	g4dn.xlarge	emr-5.30.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
内存优化	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
存储优化	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0

实例类	实例类型	支持的 Amazon EMR 最低版本
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

上一代实例

Amazon EMR 支持上一代实例，以支持针对这些实例进行了优化且尚未升级的应用程序。有关这些实例类型和升级方法的更多信息，请参阅上一代实例。

实例类	实例类型
General Purpose	m1.small[1] | m1.medium[1] | m1.large[1] | m1.xlarge[1] | m3.xlarge[1] | m3.2xlarge[1] | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge
Compute Optimized	c1.medium[1] [2] | c1.xlarge[1] | c3.xlarge[1] | c3.2xlarge[1] | c3.4xlarge[1] | c3.8xlarge[1] | c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge
GPU Optimized	g2.2xlarge

实例类	实例类型
Memory Optimized	m2.xlarge[1] | m2.2xlarge[1] | m2.4xlarge[1] | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Storage Optimized	i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge

[1] 使用 Amazon EMR 版本早于 5.13.0 的 PVM 虚拟化 AMI。有关更多信息，请参阅 Linux AMI 虚拟化类型。

[2] 在版本 5.15.0 中不受支持。

实例购买选项

当您设置集群时，可以选择 Amazon EC2 实例的购买选项。您可以选择按需型实例和/或 Spot 实例。价格因实例类型和区域而异。Amazon EMR 价格不包括 Amazon EC2 价格（基础服务器的价格）和 Amazon EBS 价格（如果附上 Amazon EBS 卷）。有关当前定价，请参阅 Amazon EMR 定价。

您选择在集群中使用实例组还是实例队列将确定您在集群运行时更改实例购买选项的方式。如果您选择统一实例组，则您只能在创建实例组时为其指定购买选项，且实例类型和购买选项将应用于每个实例组中的所有 Amazon EC2 实例。如果您选择使用实例队列，则可在创建实例队列后更改购买选项，并可以组合购买选项以实现您指定的目标容量。有关这些配置的更多信息，请参阅使用实例集或统一实例组创建集群。

按需实例

对于按需型实例，您可以按小时为计算容量付费。（可选）您可以为这些按需型实例使用预留实例或专用实例购买选项。对于预留实例，您可以对实例进行一次性付款以预留容量。专用实例与属于其它 Amazon账户的实例在主机硬件级别是实体隔离的。有关购买选项的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的实例购买选项。

使用预留实例

要在 Amazon EMR 中使用预留实例，您可以使用 Amazon EC2 购买预留实例并指定预留的参数，包括应用于区域或可用区时的预留范围。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 预留实例和购买预留实例。购买预留实例后，如果满足以下所有条件，则 Amazon EMR 会在集群启动时使用预留实例：

- 在集群配置中指定匹配预留实例规范的按需实例

- 在实例预留范围内启动集群（可用区或区域）
- 预留实例容量仍可用

例如，假设您购买了一个 m5.xlarge 预留实例，其实例预留范围限于美国东部区域。然后，您在美国东部区域启动一个使用两个 m5.xlarge 实例的 Amazon EMR 集群。第一个实例按照预留实例费率计费，第二个实例按照按需费率计费。在创建任何按需型实例之前使用预留实例容量。

使用专用实例

要使用专用实例，您可以先使用 Amazon EC2 购买专用实例，然后使用 Dedicated (专用) 租赁属性创建 VPC。随后，您在 Amazon EMR 中指定集群应在此 VPC 中启动。在集群启动时，集群中符合专用实例规范的任何按需型实例均使用可用的专用实例。

> **Note**
>
> Amazon EMR 不支持对单个实例设置 dedicated 属性。

竞价型实例

利用 Amazon EMR 中的竞价型实例，您能够以比按需购买更低的价格来购买 Amazon EC2 实例容量。使用竞价型实例的缺点是，如果您正在运行的实例类型的竞价型容量不可用，则实例可能会终止。有关何时使用适用于您应用程序的竞价型实例的更多信息，请参阅什么时候应该使用竞价型实例？。

当 Amazon EC2 拥有未利用的容量时，它会以更低的成本（称为 Spot 价格）提供 EC2 实例。此价格会根据可用性和需求发生波动，并且按区域和可用区制定。在选择竞价型实例时，您指定您愿意为每个 EC2 实例类型支付的最高 Spot 价格。当集群的可用区中的 Spot 价格低于为该实例类型指定的最高 Spot 价格时，实例将启动。在实例运行时，您支付的是当前 Spot 价格，而不是您的最高 Spot 价格。

> **Note**
>
> 自 2021 年 7 月 1 日起，新客户不再享有具有定义持续时间（也称为 Spot 块）的竞价型实例。对于以前使用过该功能的客户，我们将继续支持具有定义持续时间的竞价型实例，直到 2022 年 12 月 31 日。

要查看当前定价，请参阅 Amazon EC2竞价型实例定价。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的竞价型实例https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/

using-spot-instances.html。在创建和配置集群时，您指定用于最终确定集群从中启动的可用区的网络选项。有关更多信息，请参阅配置联网。

> **Tip**
>
> 在使用 Advanced Options (高级选项) 创建集群的情况下，将鼠标悬停在 Spot 购买选项旁边的信息工具提示上方，即可在控制台中看到实时 Spot 价格。这将显示所选区域中每个可用区的价格。最低价格位于绿色行中。因为可用区之间的 Spot 价格会波动，所以，选择初始价格最低的可用区并不保证集群在整个使用期限的使用价格最低。要获得最佳结果，请在选择之前了解可用区定价的历史记录。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的竞价型实例定价历史记录。

Spot 实例选项取决于您在集群配置中使用统一实例组还是实例队列。

统一实例组中的竞价型实例

当您使用统一实例组中的竞价型实例时，实例组中的所有实例必须是竞价型实例。您为集群指定一个子网或可用区。对于每个实例组，您指定一个竞价型实例和一个最高 Spot 价格。如果集群的区域和可用区中的 Spot 价格低于最高 Spot 价格，则该类型的竞价型实例将启动。如果 Spot 价格高于您的最高 Spot 价格，则实例将终止。仅在配置实例组时设置最高 Spot 价格。它可以在稍后更改。有关更多信息，请参阅使用实例集或统一实例组创建集群。

实例集中的竞价型实例

在使用实例队列配置时，可使用其它选项更好地控制竞价型实例启动和终止的方式。基本上，实例队列使用与统一实例组不同的方法来启动实例。其工作方式是，您为竞价型实例（和按需实例）确立目标容量 并创建最多 5 个实例类型。此外，您还可以为每个实例类型指定权重容量，或使用实例类型的 vCPU (YARN vcore) 作为权重容量。在预置该类型的实例时，此权重容量将计入目标容量。Amazon EMR 使用两个购买选项预置实例，直至达到每个目标的目标容量。此外，您可以为 Amazon EMR 定义一系列可用区，以便在启动实例时从中选择。您还为每个队列提供其他 Spot 选项，包括预置超时。有关更多信息，请参阅配置实例集。

实例存储

实例存储和 Amazon EBS 卷存储用于 HDFS 数据，以及缓冲区、缓存、Scratch 数据和一些应用程序可能“溢出”到本地文件系统中的其他临时内容。

Amazon EBS 在 Amazon EMR 中的工作方式与常规 Amazon EC2 实例的工作方式不同。附加到 Amazon EMR 集群的 Amazon EBS 卷是临时的：这些卷在集群和实例终止时（例如，在缩减实例组

时）将被删除，因此请务必不要指望数据能持久保留。虽然数据是临时存在的，但有可能可以复制 HDFS 中的数据，具体取决于集群中的节点数目和规范。在添加 Amazon EBS 存储卷时，这些卷将作为附加卷进行安装。它们不是引导卷的一部分。YARN 已配置为使用所有附加卷，但您负责将附加卷作为本地存储 (例如，针对本地日志文件) 分配。

有关将 Amazon EBS 与 Amazon EMR 集群结合使用的其他警告包括：

- 您无法拍摄 Amazon EBS 卷快照，然后在 Amazon EMR 中将其还原。要创建可重复使用的自定义配置，请使用自定义 AMI（在 Amazon EMR 版本 5.7.0 和更高版本中提供)。有关更多信息，请参阅使用自定义 AMI。
- 只有在使用自定义 AMI 时，才支持加密的 Amazon EBS 根设备卷。有关更多信息，请参阅创建带加密 Amazon EBS 根设备卷的自定义 AMI。
- 如果使用 Amazon EMR API 应用标签，则这些操作将适用于 EBS 卷。
- 每个实例最多有 25 个卷。
- 核心节点上的 Amazon EBS 卷不得小于 5GB。

实例的默认 Amazon EBS 存储

Amazon EMR 自动将 Amazon EBS 通用型 SSD（gp2）10GB 卷附加为其 AMI 的根设备以提高性能。此外，对于仅具有 EBS 存储的 EC2 实例，Amazon EMR 会将 Amazon EBS gp2 存储卷分配给实例。在使用 Amazon EMR 发行版 5.22.0 和更高版本创建集群时，默认 Amazon EBS 存储量根据实例大小而增加。我们跨多个卷拆分增加的存储，从而提高了 IOPS 性能，这相应又会提高某些标准化工作负载的性能。如果要使用其他 Amazon EBS gp2 实例存储配置，您可以在创建 Amazon EMR 集群或将节点添加到现有集群时指定该配置。目前，Amazon EBS gp3 卷不能用作 Amazon EMR 集群上的根卷。您只能将 Amazon EBS gp2 卷用作根卷，并将 gp3 卷添加为附加卷。下表列出了每种实例类型的默认 Amazon EBS gp2 存储卷数量、大小以及总大小。

Amazon EBS 的成本将根据运行集群的 Amazon Web Services 区域中 gp2 卷的月度费用，按小时比例收取。例如，按 0.10 USD/GB/月收费的区域中的每个集群节点上的根卷的每小时 Amazon EBS 费用约为 0.00139 USD/小时（0.10 USD/GB/月除以 30 天除以 24 小时乘以 10 GB）。

Amazon EMR 5.22.0 和更高版本的默认 Amazon EBS gp2 存储卷和大小（按实例类型划分）

实例大小	卷数	卷大小 (GiB)	总大小（GiB）
*.large	1	32	32

实例大小	卷数	卷大小 (GiB)	总大小（GiB）
*.xlarge	2	32	64
*.2xlarge	4	32	128
*.4xlarge	4	64	256
*.8xlarge	4	128	512
*.9xlarge	4	144	576
*.10xlarge	4	160	640
*.12xlarge	4	192	768
*.16xlarge	4	256	1024
*.18xlarge	4	288	1152
*.24xlarge	4	384	1536

指定额外的 EBS 存储卷

在 Amazon EMR 中配置实例类型时，可以指定其它 EBS 卷来添加实例存储（如果存在）和默认 EBS 卷之外添加容量。Amazon EBS 提供以下卷类型：通用型（SSD）、预置 IOPS（SSD）、吞吐优化（SSD）、冷数据（SSD）和磁性介质。它们的性能特点和价格不同，您可根据应用程序的分析和业务需求定制您的存储。例如，一些应用程序可能需要溢出到磁盘，而其它应用程序可在内存中或使用 Amazon S3 安全工作。

您只能在集群启动时以及添加额外的任务节点实例组时将 Amazon EBS 卷附加到实例。如果 Amazon EMR 集群中的某个实例失败，则该实例及其附加的 Amazon EBS 卷将被替换为新卷。因此，如果您手动分离 Amazon EBS 卷，Amazon EMR 会将此操作视为失败并替换实例存储（如果适用）和卷存储。

Amazon EMR 不允许您将现有 EMR 集群的卷类型从 gp2 修改为 gp3。要将 gp3 用于工作负载/使用案例，您需要启动一个新的 EMR 集群。此外，我们不建议更新正在使用或正在预调配的集群的吞吐量和 IOPS，因为 Amazon EMR 将您在集群启动时指定的吞吐量和 IOPS 值用于在集群纵向扩展期间添加的任何新实例。请参阅 比较 Amazon EBS 卷类型 gp2 和 gp3 和 迁移到 gp3 时选择 IOPS 和吞吐量。

> **Important**
>
> 要将 gp3 卷用于您的 EMR 集群，请使用 API、开发工具包或 CLI 启动新的 EMR 集群。

比较 Amazon EBS 卷类型 gp2 和 gp3

以下是 us-east-1（弗吉尼亚北部）区域 gp2 和 gp3 卷的成本比较

卷类型	gp3	gp2
卷大小	1GiB – 16TiB	1GiB – 16TiB
默认/基准 IOPS	3000	3 IOPS/GiB（至少 100 IOPS）到最大 16000 IOPS。小于 1TiB 的卷最高可突增至 3000 IOPS。
每个卷的最大 IOPS	16000	16000
默认/基准吞吐量	125 MiB/s	吞吐量限制介于 128 MiB/s 和 250 MiB/s 之间，具体取决于卷大小。
每个卷的最大吞吐量	1,000 MiB/s	250 MiB/s
价格	0.08 USD/GiB-月，每月免费使用 3000 IOPS，超过 3000 的部分 0.005 USD/预调配 IOPS-月；免费提供 125 MiB/s，超过 125MiB/s 的部分 0.04 USD/预调配 MiB/s-月	0.10 USD/GiB-月

迁移到 gp3 时选择 IOPS 和吞吐量

预调配 gp2 卷时，您必须计算出卷的大小才能获得成比例的 IOPS 和吞吐量。使用 gp3，您不必预调配更大的卷即可获得更高的性能。您可以根据应用需求选择所需的大小和性能。选择合适的大小和正确的性能参数（IOPS、吞吐量）可以最大限度地降低成本，而不会影响性能。

以下表格可帮助您选择 gp3 配置选项：

卷大小	IOPS	吞吐量
1–170GiB	3000	125 MiB/s
170–334GiB	3000	125 MiB/s，如果所选的 EC2 实例类型支持 125MiB/s 或更低，则根据使用量使用更高的速率，最大为 250 MiB/s*。
334–1000GiB	3000	125 MiB/s，如果所选的 EC2 实例类型支持 125MiB/s 或更低，则根据使用量使用更高的速率，最大为 250 MiB/s*。
1000+ GiB	匹配 gp2 IOPS（大小（以 GiB 为单位）x 3）或由当前 gp2 卷驱动的最大 IOPS	125 MiB/s，如果所选的 EC2 实例类型支持 125MiB/s 或更低，则根据使用量使用更高的速率，最大为 250 MiB/s*。

*Gp3 能够提供高达 1000 MiB/s 的吞吐量。由于 gp2 提供的最大吞吐量为 250Mib/s，因此在使用 gp3 时可能不需要超过此限制。

配置联网

大多数集群都使用 Amazon Virtual Private Cloud（Amazon VPC）启动到虚拟网络。VPC 是Amazon中在逻辑上与Amazon账户隔离的独立虚拟网络。您可以对私有 IP 地址范围、子网、路由表和网络网关等进行配置。有关更多信息，请参阅《Amazon VPC 用户指南》。

VPC 提供以下功能：

- 处理敏感数据

在 VPC 中启动集群类似于使用额外的工具（如路由表和网络 ACL）在私有网络中启动集群，以便定义可以访问网络的人员。如果您正在处理集群中的敏感数据，您可能希望获得在 VPC 中启动集群所提供的额外访问控制。此外，您还可以选择在私有子网中启动您的资源，其中，所有这些资源都没有直接的 Internet 连接。

- 访问内部网络上的资源

  如果您的数据源位于私有网络中，可能无法或者不愿将这些数据上载到Amazon，以便导入 Amazon EMR，这要么是因为要传输的数据量，要么是因为数据的敏感性质。相反，您可以在 VPC 中启动集群，并通过 VPN 连接将您的数据中心连接到 VPC，从而允许集群访问内部网络上的资源。例如，如果数据中心内有 Oracle 数据库，则在通过 VPN 连接到该网络的 VPC 中启动集群，可使集群能够访问 Oracle 数据库。

公有子网和私有子网

您可以同时在公有 VPC 子网和私有 VPC 子网中启动 Amazon EMR 集群。这意味着，您不需要互联网连接即可运行 Amazon EMR 集群；但是，您可能需要配置网络地址转换（NAT）和 VPN 网关来访问位于 VPC 外部的服务或资源，例如，在公司内部网或公有Amazon服务终端节点（如 Amazon Key Management Service）中。

> **Important**
>
> Amazon EMR 仅支持在版本 4.2 及更高版本中的私有子网中启动集群。

有关 Amazon VPC 的更多信息，请参阅 Amazon VPC 用户指南。

主题

- Amazon VPC 选项
- 设置 VPC 以托管集群
- 在 VPC 中启动集群
- 私有子网的最低 Amazon S3 策略
- 用于了解 VPC 的更多资源

Amazon VPC 选项

在 VPC 中启动 Amazon EMR 集群时，您可以在公有子网、私有子网或共享子网中启动它。配置上存在少量但明显的区别，具体取决于您为集群选择的子网类型。

公有子网

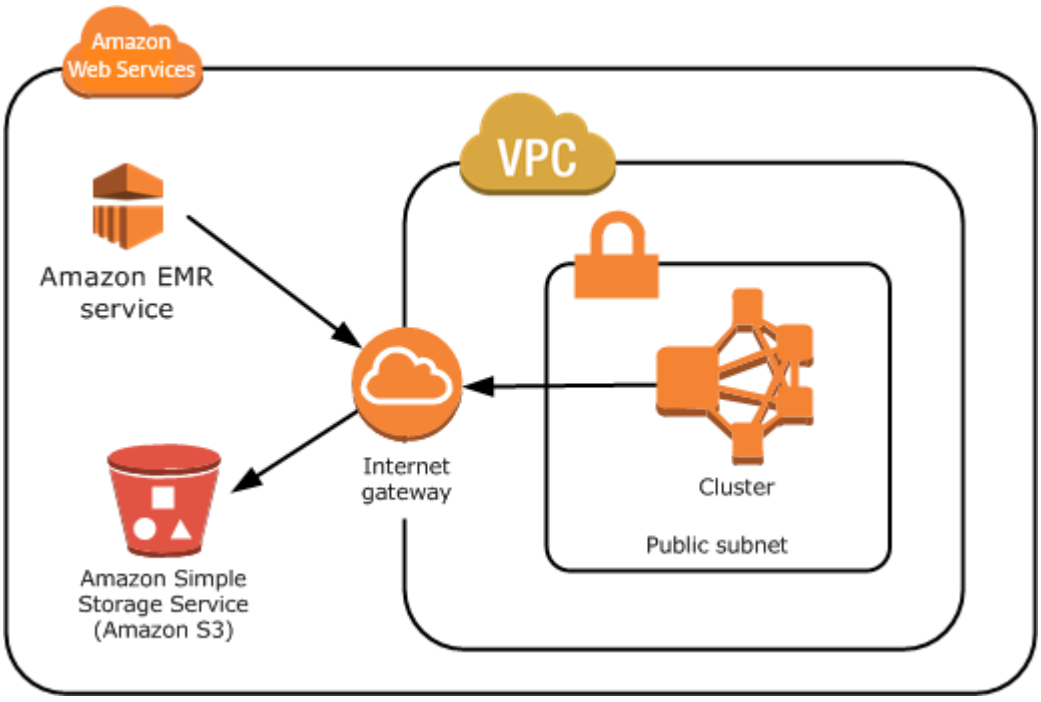
公有子网中的 EMR 集群需要已连接的 Internet 网关。这是因为 Amazon EMR 集群必须访问Amazon服务和 Amazon EMR。如果可利用某项服务（如 Amazon S3）创建 VPC 终端节点，则可使用该终端节点访问这些服务，而不是通过互联网网关访问公有终端节点。此外，Amazon EMR 无法通过网络地址转换（NAT）设备与公有子网中的集群通信。为此，需要一个 Internet 网关，但您仍可在更复杂的场景中为其它流量使用 NAT 实例或网关。

集群中的所有实例都通过 VPC 终端节点或互联网网关连接到 Amazon S3。目前不支持 VPC 终端节点的其它Amazon服务仅使用互联网网关。

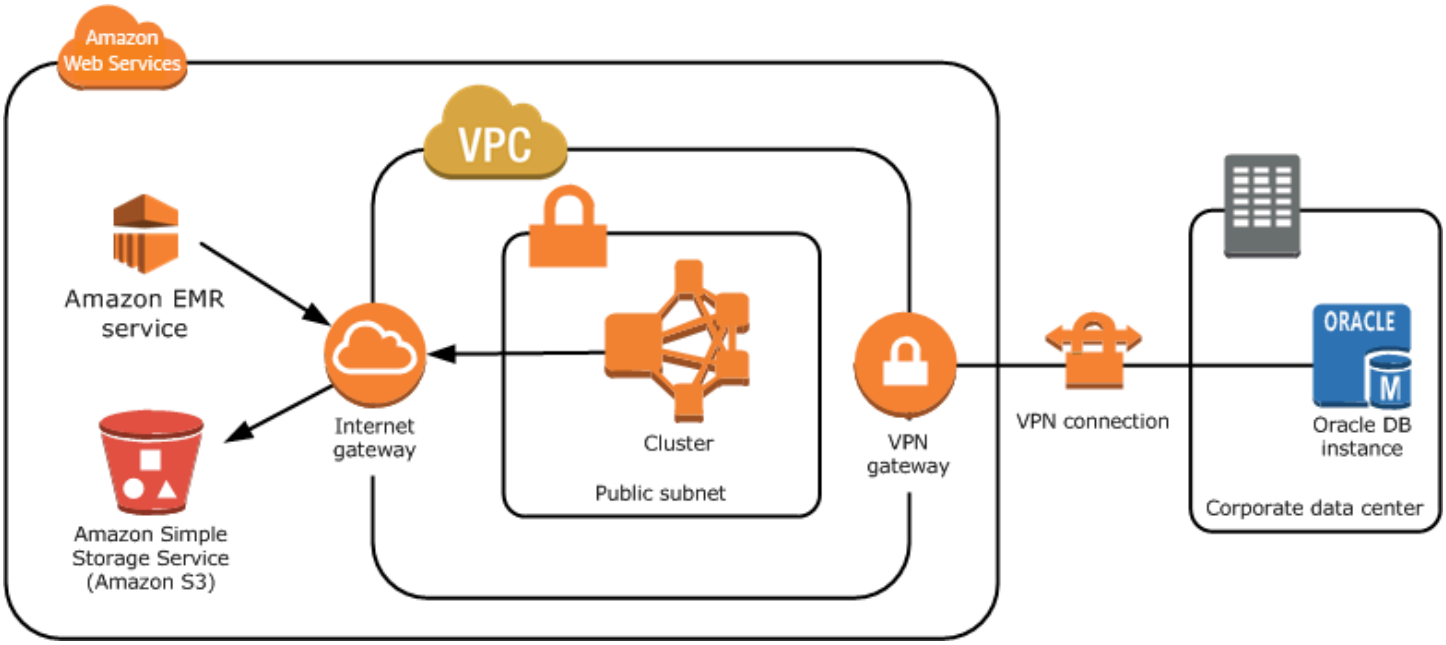
如果您不希望其它Amazon资源连接到互联网网关，则可以在 VPC 内创建的私有子网中启动这些组件。

在公有子网中运行的集群将使用两个安全组：一个用于主节点，另一个用于核心节点和任务节点。有关更多信息，请参阅使用安全组控制网络流量。

下图说明 Amazon EMR 集群如何在带公有子网的 VPC 中运行。该集群能够通过互联网网关连接到其它Amazon资源，如 Amazon S3 存储桶。



下图演示如何设置 VPC，以便 VPC 中的集群可以访问您自己的网络中的资源 (如 Oracle 数据库)。



私有子网

借助私有子网，无需挂载互联网网关的子网也可启动Amazon资源。Amazon EMR 支持在发行版本 4.2.0 及更高版本的私有子网中启动集群。

> **Note**
>
> 当您在私有子网中设置 Amazon EMR 集群时，我们建议您也为 Amazon S3 设置 VPC 端点。如果您的 EMR集群处于没有适用于 Amazon S3 的 VPC 端点的私有子网中，则您需要为与 S3 流量关联的其他 NAT 网关付费，因为您的 EMR 集群与 S3 之间的流量不是位于您的 VPC 内。

私有子网与公有子网的区别在以下方面：

- 要访问未提供 VPC 终端节点的Amazon服务，您仍必须使用 NAT 实例或互联网网关。
- 您至少必须在 Amazon S3 中提供到 Amazon EMR 服务日志存储桶和 Amazon Linux 存储库的路由。有关更多信息，请参阅私有子网的最低 Amazon S3 策略。
- 如果您使用 EMRFS 功能，则需要有一个 Amazon S3 VPC 终端节点和一个从私有子网到 DynamoDB 的路由。
- 调试仅在您提供一个从私有子网到公有 Amazon SQS 终端节点的情况下起作用。
- 仅支持使用Amazon Web Services Management Console在公有子网中通过 NAT 实例或网关创建私有子网配置。为 EMR 集群添加和配置 NAT 实例和 Amazon S3 VPC 终端节点的最简单方法是，使

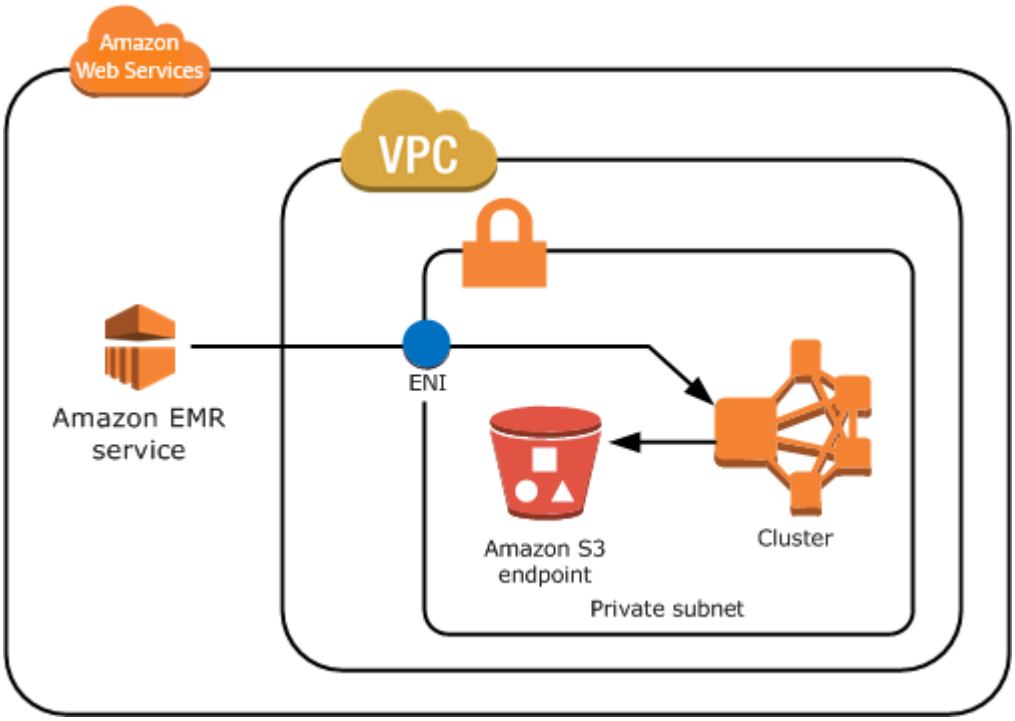
用 Amazon EMR 控制台中的 VPC Subnets List (VPC 子网列表) 页面。要配置 NAT 网关，请参阅《Amazon VPC 用户指南》中的 NAT 网关。

- 您无法利用现有 Amazon EMR 集群将子网从公有子网更改为私有子网，反之亦然。要在私有子网中找到 Amazon EMR 集群，则必须在私有子网中启动该集群。

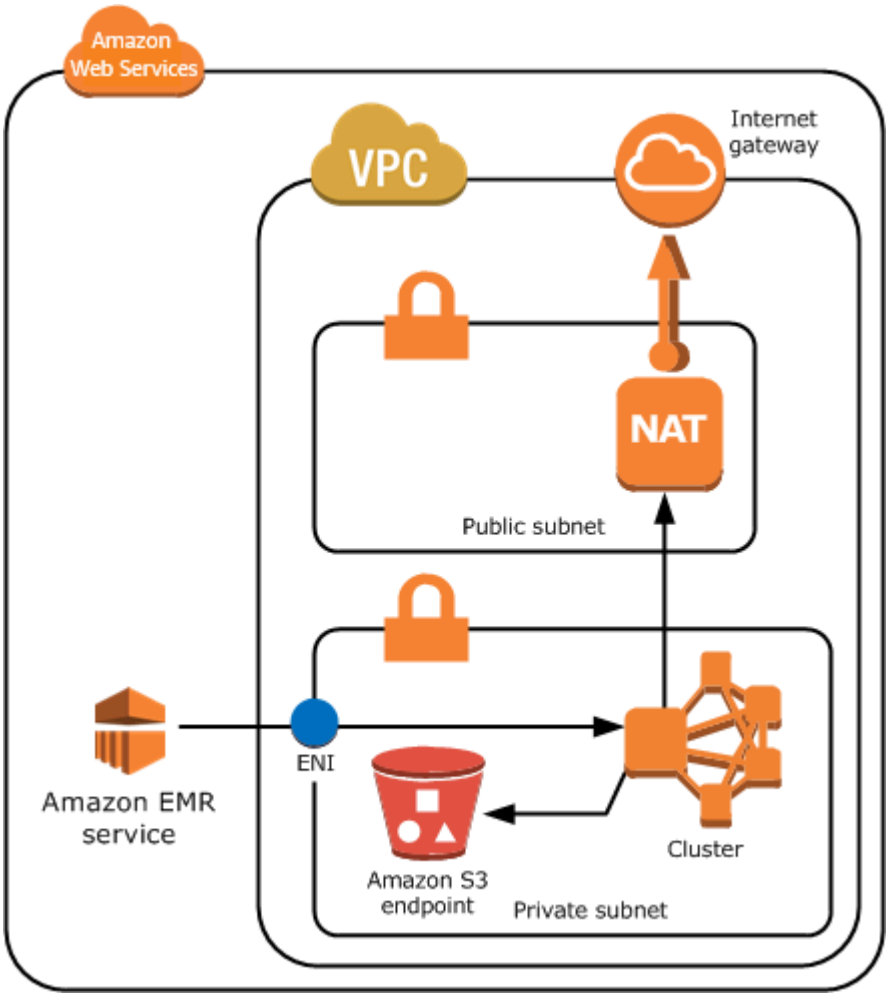
Amazon EMR 为私有子网中的集群创建并使用不同的默认安全组：ElasticMapReduce-Master-Private、ElasticMapReduce-Slave-Private、and ElasticMapReduce-ServiceAccess。有关更多信息，请参阅使用安全组控制网络流量。

有关您的集群的 NACL 的完整列表，请选择 Amazon EMR 控制台的 Cluster Details（集群详细信息）页面上的 Security groups for Primary（主节点的安全组）和 Security groups for Core & Task（核心与任务节点的安全组）。

下图说明如何在私有子网中配置 Amazon EMR 集群。子网外部的唯一通信是到 Amazon EMR 的通信。



下图显示已连接到公有子网中的 NAT 实例的私有子网中的 Amazon EMR 集群的示例配置。



共享子网

VPC 共享允许客户与同一Amazon组织内的其它Amazon账户共享子网。您可以将 Amazon EMR 集群启动到公有共享子网和私有共享子网中，但需要注意以下几点。

子网拥有者必须与您共享一个子网才能在其中启动 Amazon EMR 集群。但是，共享子网稍后可以取消共享。有关更多信息，请参阅使用共享 VPC。当某个集群启动到共享子网并且该共享子网随后取消共享时，您可以在子网取消共享时根据 Amazon EMR 集群状态观察特定行为。

- 在集群成功启动之前，子网未取消共享 - 如果拥有者在参与者启动集群时停止共享 Amazon VPC 或子网，则集群可能无法启动或部分初始化，而无需预置请求的所有实例。
- 集群成功启动之后，子网未取消共享 - 当拥有者停止与参与者共享子网或 Amazon VPC 时，参与者的集群将无法调整大小以添加新实例或替换不健康的实例。

当您启动一个 Amazon EMR 集群时，系统会创建多个安全组。在共享子网中，子网参与者控制这些安全组。子网拥有者可以看到这些安全组，但不能对其执行任何操作。如果子网拥有者想要删除或修改安全组，则创建安全组的参与者必须执行该操作。

使用 IAM 控制 VPC 权限

默认情况下，所有用户都可以查看该账户的所有子网，且任何用户都可以在任何子网中启动集群。

当您在 VPC 中启动集群时，可以使用 Amazon Identity and Access Management（IAM）控制集群的访问权限，并使用策略限制操作，就像您操作在 Amazon EC2 Classic 中启动的集群一样。有关 IAM 的更多信息，请参阅 IAM 用户指南。

您还可以使用 IAM 控制可以创建和管理子网的人员。例如，您可以创建一个账户来管理子网，再创建第二个账户可以启动集群但无法修改 Amazon VPC 设置的用户账户。有关管理 Amazon EC2 和 Amazon VPC 中的策略和操作的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的适用于 Amazon EC2 的 IAM policy。

设置 VPC 以托管集群

您必须先创建 VPC 和子网，然后才能在 VPC 中启动集群。对于公有子网，您必须创建一个 Internet 网关并将它连接到子网。以下说明介绍如何创建能够托管 Amazon EMR 集群的 VPC。

为 Amazon EMR 集群创建带有子网的 VPC

1. 通过以下网址打开 Amazon VPC 控制台：https://console.aws.amazon.com/vpc/。
2. 在页面右上角，为 VPC 选择 Amazon Web Services 区域。
3. 选择 Create VPC（创建 VPC）。
4. 在 VPC settings（VPC 设置）页面上，选择 VPC and more（VPC 等）。
5. 在 Name tag auto-generation（名称标签自动生成）下，启用 Auto-generate（自动生成）然后输入 VPC 的名称。这可帮助您在创建 VPC 和子网后在 Amazon VPC 控制台中识别它们。
6. 在 IPv4 CIDR block（IPv4 CIDR 块）字段中，为 VPC 输入私有 IP 地址空间，以确保正确解析 DNS 主机名；否则，可能会出现 Amazon EMR 集群故障。该地址空间包括以下 IP 地址范围：
   - 10.0.0.0 - 10.255.255.255
   - 172.16.0.0 - 172.31.255.255
   - 192.168.0.0 - 192.168.255.255
7. 在 Number of Availability Zones (AZs)（可用区数量）下，选择要在其中启动子网的可用区数量。

8. 在 Number of public subnets（公有子网数量）下，选择要添加到 VPC 的单个公有子网。如果集群中使用的数据可在 Internet 上（例如，在 Amazon S3 或 Amazon RDS 中）使用，则只需使用公有子网，无需添加私有子网。
9. 在 Number of private subnets（私有子网数量）下，选择要添加到 VPC 的私有子网数量。如果您的应用程序数据存储在您自己的网络中（例如，在 Oracle 数据库中），则选择一个或多个私有子网。对于私有子网中的 VPC，所有 Amazon EC2 实例都必须至少有一个通过弹性网络接口到 Amazon EMR 的路由。在控制台中，将自动为您进行配置。
10. 在 NAT gateways（NAT 网关）下，可选择添加 NAT 网关。仅当您的私有子网需要与 Internet 通信的情况下，才需要使用它们。
11. 在 VPC endpoints（VPC 端点）下，选择向子网添加 Amazon S3 的端点。
12. 验证是否已选中 Enable DNS hostnames（启用 DNS 主机名）和 Enable DNS resolution（启用 DNS 解析）。有关更多信息，请参阅将 DNS 与您的 VPC 一起使用。
13. 选择 Create VPC（创建 VPC）。
14. 状态窗口会显示工作的进度。工作完成后，选择 View VPC（查看 VPC）导航到 Your VPCs（您的 VPC）页面，该页面将显示默认 VPC 和刚创建的 VPC。您创建的 VPC 是非默认 VPC，因此 Default VPC 列将显示 No。
15. 如果您想将 VPC 与不包含域名的 DNS 条目相关联，请导航到 DHCP option sets（DHCP 选项集），选择 Create DHCP options set（创建 DHCP 选项集）然后省略域名。创建选项集后，导航到新 VPC，选择 Actions（操作）菜单下的 Edit DHCP options set（编辑 DHCP 选项集），然后选择新选项集。创建了 DNS 选项集之后，您无法使用控制台编辑域名。

    这是针对 Hadoop 和相关应用程序的最佳做法，可确保解析节点的完全限定域名 (FQDN)。要确保正确解析 DNS，请配置 VPC，使其包含将参数设置为以下值的 DHCP 选项集：

    - domain-name = **`ec2.internal`**

      如果您所在的区域是美国东部（弗吉尼亚北部），请使用 **`ec2.internal`**。对于其它区域，请使用 ***`region-name`*`.compute.internal`**。例如，在 `us-west-2` 中，请使用 **`us-west-2.compute.internal`**。对于 Amazon GovCloud (US-West) 区域，请使用 **`us-gov-west-1.compute.internal`**。
    - domain-name-servers = **`AmazonProvidedDNS`**

    有关更多信息，请参阅《Amazon VPC 用户指南》中的 DHCP 选项集。
16. 创建 VPC 之后，转到 Subnets（子网）页面并记下新 VPC 的一个子网的 Subnet ID（子网 ID）。在该 VPC 中启动 Amazon EMR 集群时，您将使用此信息。

在 VPC 中启动集群

在具有一个配置为托管 Amazon EMR 集群的子网后，通过在创建集群时指定关联的子网标识符来在该子网中启动集群。

> **Note**
>
> Amazon EMR 支持版本 4.2 及更高版本中的私有子网。

启动集群时，Amazon EMR 基于集群是在 VPC 私有子网还是公有子网中启动来添加安全组。所有安全组都允许端口 8443 的入口与 Amazon EMR 服务进行通信，但 IP 地址范围因公有子网和私有子网而异。Amazon EMR 管理这些所有安全组，而且随着时间的推移，可能需要向 Amazon 范围添加其它 IP 地址。有关更多信息，请参阅使用安全组控制网络流量。

为了管理 VPC 上的集群，Amazon EMR 为主节点附加了一个网络设备，并通过该设备对主节点进行管理。您可以使用 Amazon EC2 API 操作 `DescribeInstances` 查看此设备。如果您以任何方式修改此设备，则集群可能会失败。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台在 VPC 中启动集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Networking（联网）下，转到 Virtual private cloud (VPC) [虚拟私有云（VPC）] 字段。输入 VPC 名称或选择 Browse（浏览）以选择 VPC。或者，选择 Create VPC（创建 VPC）以创建可用于集群的 VPC。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台在 VPC 中启动集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 在 Hardware Configuration (硬件配置) 部分，对于 Network (网络)，选择您之前创建的 VPC 网络的 ID。
5. 对于 EC2 Subnet (EC2 子网)，选择您之前创建的子网的 ID。
   a. 如果使用 NAT 实例和 S3 终端节点选项正确地配置您的私有子网，将在子网名称和标识符上方显示 (EMR Ready)。
   b. 如果您的私有子网没有 NAT 实例和/或 S3 终端节点，则可通过选择 Add S3 endpoint and NAT instance (添加 S3 终端节点和 NAT 实例)、Add S3 endpoint (添加 S3 终端节点) 或 Add NAT instance (添加 NAT 实例) 进行配置。为您的 NAT 实例和 S3 终端节点选择所需的选项，然后选择 Configure (配置)。

   > **Important**
   >
   > 要从 Amazon EMR 创建 NAT 实例，您需要 ec2:CreateRoute、`ec2:RevokeSecurityGroupEgress`、`ec2:AuthorizeSecurityG` 和 `cloudformation:CreateStack` 权限。

   > **Note**
   >
   > 为您的 NAT 设备启动 Amazon EC2 实例会产生额外费用。
6. 继续创建集群。

Amazon CLI

使用 Amazon CLI 在 VPC 中启动集群

> **Note**
>
> Amazon CLI 未提供自动创建 NAT 实例并将该实例连接到私有子网的方法。但是，要在子网中创建 S3 端点，您可以使用 Amazon VPC CLI 命令。使用控制台可在私有子网中创建 NAT 实例和启动集群。

配置 VPC 之后，可以使用带 `create-cluster` 参数的 `--ec2-attributes` 子命令在 VPC 中启动 Amazon EMR 集群。使用 `--ec2-attributes` 参数可为集群指定 VPC 子网。

- 要在特定子网中创建集群，请键入以下命令，将 *myKey* 替换为您的 Amazon EC2 密钥对的名称，并将 *77XXXX03* 替换为您的子网 ID。

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --
applications Name=Hadoop Name=Hive Name=Pig --use-default-roles --ec2-attributes
 KeyName=myKey,SubnetId=subnet-77XXXX03 --instance-type m5.xlarge --instance-
count 3
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

> **Note**
>
> 如果您之前未创建默认 Amazon EMR 服务角色和 EC2 实例配置文件，请先键入 `aws emr create-default-roles` 创建它们，然后再键入 `create-cluster` 子命令。

私有子网的最低 Amazon S3 策略

对于私有子网，您必须让 Amazon EMR 至少能够访问 Amazon Linux 存储库。此私有子网策略是用于访问 Amazon S3 的 VPC 终端节点策略的一部分。对于 Amazon EMR 5.25.0 或更高版本，要启用一键访问持久性 Spark 历史记录服务器，您必须允许 Amazon EMR 访问收集 Spark 事件日志的系统存储桶。如果启用日志记录，请为 `aws157-logs-*` 存储桶授予 PUT 权限。有关更多信息，请参阅[一键访问持久性 Spark 历史记录服务器]。

由您决定满足业务需求的策略限制。例如，您可以指定区域packages.us-east-1.amazonaws.com以避免 Amazon S3 存储桶名称混淆。以下示例策略提供了访问 Amazon Linux 存储库和 Amazon EMR 系统存储桶以收集 Spark 事件日志的权限。将 *MyRegion* 替换为您的日志存储桶所在的区域，例如 us-east-1。

有关将 IAM policy 与 Amazon VPC 终端节点结合使用的更多信息，请参阅Amazon S3 终端节点策略。

```
{
   "Version": "2008-10-17",
   "Statement": [
       {
           "Sid": "AmazonLinuxAMIRepositoryAccess",
           "Effect": "Allow",
           "Principal": "*",
           "Action": "s3:GetObject",
           "Resource": [
            "arn:aws:s3:::packages.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::repo.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::repo.MyRegion.emr.amazonaws.com/*"
           ]
       },
       {
           "Sid": "EnableApplicationHistory",
           "Effect": "Allow",
           "Principal": "*",
           "Action": [
               "s3:Put*",
               "s3:Get*",
               "s3:Create*",
               "s3:Abort*",
               "s3:List*"
           ],
           "Resource": [
            "arn:aws:s3:::prod.MyRegion.appinfo.src/*"
           ]
       }
   ]
}
```

以下示例策略提供了访问 Amazon Linux 2 存储库所需的权限。默认值是 Amazon Linux 2 AMI。

```
{
   "Statement": [
       {
           "Sid": "AmazonLinux2AMIRepositoryAccess",
           "Effect": "Allow",
           "Principal": "*",
           "Action": "s3:GetObject",
           "Resource": [
            "arn:aws:s3:::amazonlinux.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::amazonlinux-2-repos-MyRegion/*"
           ]
       }
   ]
}
```

用于了解 VPC 的更多资源

要了解有关 VPC 和子网的更多信息，请参阅以下主题。

- VPC 中的私有子网
  - 场景 2：带有公有子网和私有子网 (NAT) 的 VPC
  - NAT 实例
  - Amazon VPC NAT 实例的高可用性：示例
- VPC 中的公有子网
  - 场景 1：带单个公有子网的 VPC
- 一般 VPC 信息
  - Amazon VPC User Guide
  - VPC 对等
  - 在您的 VPC 中使用弹性网络接口
  - 安全地连接到私有 VPC 中运行的 Linux 实例

使用实例集或统一实例组创建集群

当您创建集群并指定主节点、核心节点和任务节点的配置时，有两个配置选项。您可以使用实例队列 或统一实例组。您选择的配置选项适用于所有节点，将在集群的生命周期内应用，并且实例队列和实例组无法在集群中共存。实例集配置在 Amazon EMR 版本 4.8.0 和更高版本中可用（不包括 5.0.x 版）。

您可以使用 Amazon EMR 控制台、Amazon CLI 或 Amazon EMR API 创建带任一配置的集群。从 Amazon CLI 使用 `create-cluster` 命令时，可使用 `--instance-fleets` 参数通过实例队列创建集群，也可使用 `--instance-groups` 参数通过统一实例组创建集群。

使用 Amazon EMR API 时的情况也是如此。您使用 `InstanceGroups` 配置指定一组 `InstanceGroupConfig` 对象，或使用 `InstanceFleets` 配置指定一组 `InstanceFleetConfig` 对象。

在 Amazon EMR 新控制台中，您可以在创建集群时选择使用实例组或实例集，并且可以选择对每个实例组或实例集使用竞价型实例。在 Amazon EMR 旧控制台中，如果您在创建集群时使用默认 Quick Options（快速选项）设置，Amazon EMR 会将统一实例组配置应用于集群并使用按需型实例。要将竞价型实例与统一实例组结合使用，或要配置实例队列或其它自定义设置，请选择 Advanced Options (高级选项)。

实例机群

实例集配置提供了各种面向 Amazon EC2 实例的预置选项。每个节点类型均有一个实例集，使用任务实例集是可选的。您在使用 Amazon CLI、Amazon EMR API、按需型实例和竞价型实例的分配策略创建实例时，可以为每个实例集指定最多五个 EC2 实例类型，或为每个实例集指定最多 30 个 EC2 实例类型。对于核心实例和任务实例集，您可以为按需型实例指定一个目标容量，再为竞价型实例指定一个目标容量。。Amazon EMR 选择指定实例类型的任意组合来满足目标容量，预置按需实例和竞价型实例。

对于主节点类型，Amazon EMR 会从实例列表中选择一个实例类型，然后您指定将其预配置为按需型实例还是竞价型实例。实例集还为竞价型实例和按需型购买提供了其它选项。Spot 实例选项包括指定在无法预置 Spot 容量时要执行的操作的超时以及启动 Spot 实例机群的首选分配策略（容量优化）。此外，还可以使用分配策略（最低价格）选项来启动按需型实例机群。如果您使用的服务角色不是 EMR 默认服务角色，或者在服务角色中使用了 EMR 托管式策略，则需要向自定义集群服务角色添加其它权限，以启用分配策略选项。有关更多信息，请参阅Amazon EMR 的服务角色（EMR 角色）。

有关配置实例集的更多信息，请参阅 配置实例集。

统一实例组

统一实例组可提供比实例集简单的设置。每个 Amazon EMR 集群可包含最多 50 个实例组：一个主实例组（包含一个 Amazon EC2 实例）、一个核心实例组（包含一个或多个 EC2 实例）和最多 48 个可选任务实例组。每个核心和任务实例组可以包含任意数量的 Amazon EC2 实例。您可以通过手动添加和删除 Amazon EC2 实例来扩展每个实例组，也可以设置自动扩展。有关添加和删除实例的信息，请参阅使用集群扩展。

有关配置统一实例组的更多信息，请参阅配置统一实例组。

使用实例集和实例组

主题

- 配置实例集
- 将容量预留与实例集结合使用
- 配置统一实例组
- 实例和可用区灵活性的最佳实践
- 集群配置的最佳实践

配置实例集

> **Note**
>
> 实例集配置仅在 Amazon EMR 发行版 4.8.0 及更高版本（不包括 5.0.0 和 5.0.3）中可用。

Amazon EMR 集群的实例集配置允许您为 Amazon EC2 实例选择各种预置选项，并帮助您为集群中的各节点类型制定灵活而富有弹性的资源配置策略。

在实例集配置中，您可以为每个队列内的按需实例和竞价型实例https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/using-spot-instances.html指定目标容量。集群启动时，Amazon EMR 预置实例，直至达到目标。如果 Amazon EC2 在集群运行时因价格上涨或实例失败而回收 Spot 实例，Amazon EMR 会尝试将该实例替换为您指定的任何实例类型。这使得可在 Spot 价格高峰期间更轻松地重新获取容量。

您可以为每个队列指定最多 5 个 Amazon EC2 实例类型以便 Amazon EMR 在实现目标时使用，或者在您使用 Amazon CLI 或 Amazon EMR API 以及按需实例和 Spot 实例的分配策略创建集群时，可以为每个队列指定最多 30 个 Amazon EC2 实例类型。

您也可以为不同的可用区选择多个子网。当 Amazon EMR 启动集群时，它会在这些子网中查找您指定的实例和购买选项。如果 Amazon EMR 在一个或多个可用区检测到某个Amazon大规模事件，Amazon EMR 会自动尝试将流量从受影响的可用区路由出去，并尝试根据您的选择在备用可用区启动您创建的新集群。请注意，只有在创建集群时才能选择集群可用区。如果发生可用区中断，现有的集群节点不会在新的可用区中自动重新启动。

注意事项

在 Amazon EMR 中使用实例集时，请考虑以下事项。

- 对于每个节点类型（主节点、核心节点和任务节点），您可以有一个实例集。您在 Amazon Web Services Management Console上最多可为每个实例集指定 5 个 Amazon EC2 实例类型（使用 Amazon CLI 或 Amazon EMR API 以及 实例集的分配策略 创建集群时，最多可为每个实例集指定 30 个类型）。
- Amazon EMR 选择任一或所有指定的 Amazon EC2 实例类型，使用 Spot 和按需购买选项进行预置。
- 对于核心队列和任务队列，您可以设定 Spot 和按需型实例的目标容量。请使用分配给计入目标的每个 Amazon EC2 实例的 vCPU 或通用单位。Amazon EMR 预置实例，直至达到每个目标容量。对于主实例集，目标始终为一。
- 您可以选择一个子网（可用区）或范围。如果您选择一个范围，Amazon EMR 会预置最适合的可用区中的容量。
- 在指定竞价型实例的目标容量时：
  - 对于每个实例类型，指定最高 Spot 价格。如果 Spot 价格低于最高 Spot 价格，Amazon EMR 会预置竞价型实例。您按 Spot 价格付费，但这不一定是最高 Spot 价格。
  - 对于每个队列，定义预置竞价型实例的超时期限。如果 Amazon EMR 无法预置 Spot 容量，您可以改为终止集群或切换到预置按需容量。这仅适用于预置集群，而不适用于调整集群大小。如果超时期限在集群大小调整过程中结束，则未预置的竞价请求将无效，且不会转移到按需容量。
- 对于每个实例集，您可以为竞价型实例指定以下分配策略之一：价格容量优化、容量优化、最低价格或在所有池中实现多元化。
- 对于每个实例集，您可以为按需型实例应用最低价格分配策略；您无法为按需型实例自定义分配策略。
- 对于每个采用按需型 `allocation strategy - lowest-price` 的实例集，您可以选择应用容量预留选项。
- 启动集群之前，请检查您的子网大小。如果您使用任务队列预置集群，但相应子网中没有足够的可用 IP 地址，则该队列将进入暂停状态，而不是终止集群并显示错误。为了避免此问题，我们建议您增加子网中的 IP 地址数量。

实例集选项

使用以下准则来了解实例队列选项。

主题

- 设置目标容量
- 启动选项

- 多个子网（可用区）选项
- 主节点（master node）配置

设置目标容量

为核心队列和任务队列指定所需的目标容量。此操作将确定 Amazon EMR 预置的按需实例和竞价型实例的数量。在指定实例时，您将决定每个实例计入目标的量。在预置按需实例时，该容量将计入按需目标。Spot 实例也是如此。与核心实例集和任务实例集不同，主实例集始终为一个实例。因此，此队列的目标容量始终是一。

在使用控制台时，默认情况下，Amazon EC2 实例类型的 vCPU 数量将用于目标容量的计数。您可以将此项更改为 Generic units (通用单位)，然后为每个 EC2 实例类型指定计数。在使用 Amazon CLI 时，您将为每个实例类型手动分配通用单位。

> **Important**
>
> 使用Amazon Web Services Management Console选择实例类型时，每个实例类型显示的 vCPU 数量是该实例的 YARN vcore 数量，而不是该实例类型的 EC2 vCPU 数量。有关每个实例类型 vCPU 数量的更多信息，请参阅 Amazon EC2 实例类型。

对于每个队列，可指定最多 5 个 Amazon EC2 实例类型。如果您使用 实例集的分配策略 并使用 Amazon CLI 或 Amazon EMR API 创建集群，可以为每个实例集指定最多 30 个 EC2 实例类型。Amazon EMR 选择这些 EC2 实例类型的任意组合来达到您的目标容量。由于 Amazon EMR 希望完全填满目标容量，因此可能会超过目标。例如，如果存在 2 个未填满的单位，并且 Amazon EMR 只能使用 5 个单位来预置实例，则仍将预置该实例，这意味着将超出目标容量 3 个单位。

如果减少目标容量以调整正在运行的集群的大小，Amazon EMR 会尝试完成应用程序任务并终止实例以满足新目标。有关更多信息，请参阅在任务完成时终止。

启动选项

对于竞价型实例，您可以在队列中指定每个实例类型的最高 Spot 价格。您可以采用按需价格的百分比形式或特定美元金额形式设置此价格。如果可用区中的当前 Spot 价格低于您的最高 Spot 价格，Amazon EMR 会预置竞价型实例。您按 Spot 价格付费，但这不一定是最高 Spot 价格。

> **Note**
>
> 自 2021 年 7 月 1 日起，新客户不再享有具有定义持续时间（也称为 Spot 块）的竞价型实例。对于以前使用过该功能的客户，我们将继续支持具有定义持续时间的竞价型实例，直到 2022 年 12 月 31 日。

在 Amazon EMR 5.12.1 及更高版本中，您可以选择启动采用优化容量分配的竞价型和按需型实例集。可在旧 Amazon Web Services Management Console 中或使用 API `RunJobFlow` 对此分配策略选项进行设置。请注意，您无法在新控制台中自定义分配策略。使用分配策略选项需要额外的服务角色权限。如果您为集群使用默认 Amazon EMR 服务角色和托管式策略（`EMR_DefaultRole` 和 `AmazonEMRServicePolicy_v2`），则已包含分配策略选项的权限。如果您不使用默认 Amazon EMR 服务角色和托管式策略，则必须添加它们以后才能使用此选项。请参阅Amazon EMR 的服务角色（EMR 角色）。

有关竞价型实例的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的竞价型实例https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/using-spot-instances.html。有关按需实例的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的按需实例。

如果您选择使用价格最低的分配策略来启动按需型实例集，则可以选择使用容量预留。可以使用 Amazon EMR API `RunJobFlow` 对容量预留选项进行设置。容量预留需要额外的服务角色权限，您必须添加这些权限后才能使用这些选项。请参阅分配策略权限。请注意，您无法在新控制台中自定义容量预留。

多个子网（可用区）选项

在使用实例队列时，您可以在 VPC 中指定多个 Amazon EC2 子网，每个子网均对应一个不同的可用区。如果您使用 EC2-Classic，则需明确指定可用区。Amazon EMR 将根据您的队列规范确定最佳可用区以启动实例。始终仅在一个可用区中预置实例。您可以选择私有子网或公有子网，但不能将二者混合，并且指定的子网必须位于同一 VPC 中。

主节点（master node）配置

由于主实例集只是一个实例，因此其配置与核心实例集和任务实例集的配置略有不同。由于主实例集仅包含一个实例，因此您只能为主实例集选择按需型或竞价型实例。如果您使用控制台创建实例队列，则您选定的购买选项的目标容量将设置为 1。如果您使用 Amazon CLI，请始终根据需要将 `TargetSpotCapacity` 或 `TargetOnDemandCapacity` 设置为 1。您仍然可以为主实例集选择最多 5 个实例类型（为按需型实例或竞价型实例使用分配策略选项时，最多可选择 30 个实例类型）。但与

核心实例集和任务实例集（Amazon EMR 可能会预置多个不同类型的实例）不同，Amazon EMR 会为主实例集选择要预置的单个实例类型。

实例集的分配策略

对于 Amazon EMR 5.12.1 及更高版本，您可以将分配策略选项与每个集群节点的按需型实例和竞价型实例结合使用。当您使用 Amazon CLI、Amazon EMR API 或带有分配策略的 Amazon EMR 控制台创建集群时，您可以为每个实例集指定最多 30 个 Amazon EC2 实例类型。使用默认 Amazon EMR 集群实例集配置，每个实例集最多可以有 5 种实例类型。我们建议您使用分配策略选项，它可以加快集群配置、更准确地进行竞价型实例分配并减少竞价型实例中断。

主题

- 按需型实例的分配策略
- 竞价型实例分配策略
- 分配策略权限
- 使用分配策略所需的 IAM 权限

按需型实例的分配策略

当您使用分配策略时，您的按需型实例使用最低价格策略。这将首先启动价格最低的实例。启动按需型实例时，可以选择在您的账户中使用开放或目标容量预留。您可以对主节点、核心节点和任务节点使用开放容量预留。如果使用的按需型实例采用实例集分配策略，您可能会遇到容量不足的问题。我们建议指定大量实例类型，以实现多样化并降低出现容量不足问题的概率。有关更多信息，请参阅将容量预留与实例集结合使用。

竞价型实例分配策略

您可以为竞价型实例指定以下分配策略之一：

`price-capacity-optimized`（推荐）

价格容量优化的分配策略从竞价型实例池中启动竞价型实例，这些实例池的可用容量最大，启动的实例数量价格最低。因此，价格容量优化策略获得竞价型容量的几率通常更高，中断率也更低。

`capacity-optimized`

容量优化的分配策略将竞价型实例启动到可用性最高的池中，短期内中断几率最低。对于与重新启动的作业相关的中断成本可能更高的工作负载来说，这是一个不错的选择。这是 Amazon EMR 6.9.0 及更低版本的默认策略。

diversified

通过多元化分配策略，Amazon EC2 在所有竞价型容量池中分配竞价型实例。

lowest-price

最低价格分配策略从具有可用容量的最低价格池中启动竞价型实例。如果价格最低的池没有可用容量，则竞价型实例将来自价格次低且具有可用容量的池。如果某个池在满足您所需的容量之前耗尽容量，则 Amazon EC2 Fleet 将继续从价格次低的池中提取实例，以满足您的请求。为确保达到所需容量，您可能会获得来自多个容量池的竞价型实例。由于此策略仅考虑实例价格而不考虑容量可用性，因此可能会导致较高的中断率。

分配策略权限

使用分配策略选项需要多个 IAM 权限，默认的 Amazon EMR 服务角色和 Amazon EMR 托管式策略（EMR_DefaultRole 和 AmazonEMRServicePolicy_v2）中会自动提供这些权限。如果您为集群使用自定义服务角色或托管式策略，必须添加这些权限后才能创建集群。有关更多信息，请参阅分配策略权限。

使用按需分配策略选项时，可以使用可选的按需容量预留（ODCR）。容量预留选项允许您为 Amazon EMR 集群首先使用预留容量指定首选项。此功能可确保您的关键工作负载使用您已经使用开放或目标 ODCR 预留的容量。对于非关键工作负载，容量预留首选项允许您指定是否使用预留容量。

容量预留只能由匹配其属性（实例类型、平台、可用区）的实例使用。默认情况下，Amazon EMR 在配置与实例属性匹配的按需型实例时会自动使用开放容量预留。如果您没有任何正在运行的实例与容量预留的容量匹配，则这些容量将保持未使用的状态，直至启动与其属性相匹配的实例。如果您不想在启动集群时使用任何容量预留，则必须在启动选项中将容量预留首选项设置为 none (无)。

不过，您还可以将容量预留定位到特定工作负载。这使您可以明确控制允许哪些实例在预留容量中运行。有关按需容量预留的更多信息，请参阅 将容量预留与实例集结合使用。

使用分配策略所需的 IAM 权限

您的 Amazon EMR 的服务角色（EMR 角色） 需要额外的权限才能创建使用按需型或竞价型实例集的分配策略选项的集群。

我们会自动将这些权限包含在默认 Amazon EMR 服务角色 EMR_DefaultRole 和 Amazon EMR 托管策略 AmazonEMRServicePolicy_v2 中。

如果您为集群使用自定义服务角色或托管式策略，则必须添加以下权限：

```
{
```

```
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteLaunchTemplate",
    "ec2:CreateLaunchTemplate",
    "ec2:DescribeLaunchTemplates",
    "ec2:CreateLaunchTemplateVersion",
    "ec2:CreateFleet"
   ],
   "Resource": "*"
  }
}
```

以下是创建使用开放容量或目标容量预留的集群所需的服务角色权限。除了使用分配策略选项所需的权限之外，还必须提供这些权限。

Example 服务角色容量预留的策略文档

要使用开放容量预留，必须提供以下额外权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplateVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

Example

要使用目标容量预留，必须提供以下额外权限。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplateVersions",
                "resource-groups:ListGroupResources"
            ],
            "Resource": "*"
        }
    ]
}
```

为您的集群配置实例集

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台通过实例集创建集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster configuration（集群配置）下，选择 Instance fleets（实例集）。
4. 对于每个 Node group（节点组），选择 Add instance type（添加实例类型），为主实例集和核心实例集选择最多 5 个实例类型，为任务实例集选择最多 15 个实例类型。Amazon EMR 可在启动集群时预置这些实例类型的任意组合。
5. 在每个节点组类型下，选择每个实例旁边的 Actions（操作）下拉菜单以更改以下设置：

   添加 EBS 卷

   在 Amazon EMR 预置实例类型后，指定要附加到实例类型的 EBS 卷。

编辑权重容量

对于核心节点组，将此值更改为适用于应用程序的任意单位数量。每个实例集实例类型的 YARN vCore 数量用作默认的权重容量单位。您无法编辑主节点的权重容量。

编辑最高 Spot 价格

为实例集中的每个实例类型指定最高 Spot 价格。您可以采用按需价格的百分比形式或特定美元金额形式设置此价格。如果可用区中的当前 Spot 价格低于最高 Spot 价格，Amazon EMR 会预置竞价型实例。您按 Spot 价格付费，但这不一定是最高 Spot 价格。

6. （可选）要为节点添加安全组，请在 Networking（联网）部分展开 EC2 security groups (firewall) [EC2 安全组（防火墙）]，然后为每个节点类型选择安全组。
7. 或者，要使用分配策略选项，请选中应用分配策略旁边的复选框，然后选择想要为竞价型实例指定的分配策略。如果 Amazon EMR 服务角色没有所需权限，则不应选择此选项。有关更多信息，请参阅实例集的分配策略。
8. 选择适用于集群的任何其他选项。
9. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台通过实例集创建集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 在控制台窗口的顶部，选择 Go to advanced options (转到高级选项)，输入 Software Configuration (软件配置) 选项，然后选择 Next (下一步)。
4. 在 Cluster Composition (集群构成) 下，选择 Instance fleets (实例集)。选择实例集选项时，您可看到用于指定按需型实例和竞价型实例的 Target capacity (目标容量) 选项，该选项显示在 Cluster Nodes and Instances (集群节点和实例表) 中。
5. 对于 Network (网络)，输入一个值。如果您为 Network (网络) 选择 VPC，请选择单个 Amazon EC2 Subnet (EC2 子网) 或按住 Ctrl 并单击以选择多个 EC2 子网。您选择的子网必须属于同一类型 (公有或私有)。如果您仅选择了一个子网，则您的集群将在该子网中启动。如果您选择了一组，则当集群启动时，会从该组中选择具有最佳拟合度的子网。

> **Note**
>
> 您的账户和区域可能为您提供了为 Network (网络) 选择 Launch into EC2-Classic (在 EC2-Classic 中启动) 的选项。如果您选择该选项，请从 EC2 Availability Zones (EC2 可用区) 而非 EC2 Subnets (EC2 子网) 中选择一个或多个项。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 和 Amazon VPC。

6. 如果要使用分配策略选项，请在 Allocation Strategy（分配策略）下选中复选框以应用分配策略。有关更多信息，请参阅实例集的分配策略。
7. 如果您想要更改实例集的默认名称，请在每个 Node type (节点类型) 下，选择铅笔图标，然后输入易识别的名称。如果想要删除 Task (任务) 实例集，请选择“Task (任务)”行右侧的 X 图标。
8. 选择 Add/remove instance types to fleet（为实例集添加/移除实例类型），然后从主实例集和核心实例集的列表中选择最多 5 个实例类型；为任务实例集添加最多 15 个实例类型。Amazon EMR 可在启动集群时预置这些实例类型的任意组合。
9. 对于每个核心和任务实例类型，选择您定义该实例加权容量（每个实例都计为 X 个单位）的方式。会将各实例集实例类型的 YARN vCore 数量用作默认的加权容量单位，但您可以将该值更改为对应用程序有意义的任何单位。
10. 在 Target capacity (目标容量) 下，定义每个实例集所需的按需型和竞价型实例总数。EMR 可确保实例集中的实例满足按需型和竞价型实例目标容量的请求单位。如果没有为实例集指定按需型或竞价型单位，则不会为该实例集配置容量。
11. 如果使用 Spot 的目标容量配置队列，您可以输入您的最高 Spot 价格作为“% of On-Demand (% 按需价格)”定价，也可以输入“Dollars（$）(美元（$）)”金额（以 USD 为单位）。
12. 要在预置实例类型时使 EBS 卷挂载到该类型，请选择“EBS Storage (EBS 存储)”旁边的铅笔图标，然后输入 EBS 配置选项。
13. 如果您为 Spot units (竞价型单位) 设定了一个即时计数，将根据以下准则设置 Advanced Spot options (高级竞价型实例选项)。

    - Provisioning timeout (预置超时) – 使用这些设置可控制 Amazon EMR 在无法从您指定的 Fleet instance types (队列实例类型) 中预置竞价型实例时执行的操作。您输入一个超时期限 (以分钟为单位)，然后选择 Terminate the cluster (终止集群) 还是 Switch to provisioning On-Demand Instances (切换到预置按需型实例)。如果您选择切换到按需实例，则分配给按需实例的容量将计入竞价型实例的目标容量，并且 Amazon EMR 将预置按需实例，直至达到竞价型实例的目标容量。

14. 选择 Next (下一步)，修改其它集群设置，然后选择 Next (下一步)。
15. 如果您选择应用新的分配策略选项，请在 Security Options (安全选项) 设置中，选择一个 EMR role (EMR 角色)，并选择 EC2 instance profile (EC2 实例配置文件)，其中包含分配策略选项所需的权限。否则，集群创建将失败。
16. 选择创建集群。

Amazon CLI

要使用 Amazon CLI 创建和启动带有实例集的集群，请遵循以下指南：

- 要使用实例队列创建和启动集群，请将 `create-cluster` 命令与 `--instance-fleet` 参数结合使用。
- 要获取有关集群中的实例队列的配置详细信息，请使用 `list-instance-fleets` 命令。
- 要向正在创建的集群添加多个自定义 Amazon Linux AMI，请使用`CustomAmiId`选项和每个`InstanceType`规范。您可以使用多种实例类型和多个自定义 AMI 配置实例队列节点，以满足您的要求。请参阅示例：使用实例集配置创建集群。
- 要更改实例队列的目标容量，请使用 `modify-instance-fleet` 命令。
- 要向不具有任务实例队列的集群添加此队列，请使用 `add-instance-fleet` 命令。
- 使用附加实例队列命令的 CustomAmiId 参数，可以将多个自定义 AMI 添加到任务实例队列。请参阅示例：使用实例集配置创建集群。
- 要在创建实例队列时使用分配策略选项，请更新服务角色以在下一部分内容包含示例策略文档。
- 要在使用按需分配策略创建实例队列时使用容量预留选项，请更新服务角色以在下一部分内容包含示例策略文档。
- 默认的 EMR 服务角色和 Amazon EMR 托管式策略（`EMR_DefaultRole` 和 `AmazonEMRServicePolicy_v2`）中将自动提供实例集。如果您为集群使用自定义服务角色或自定义托管式策略，则必须在下一部分内容添加分配策略的新权限。

示例：使用实例集配置创建集群

以下示例演示了 `create-cluster` 命令以及可组合使用的各种选项。

> **Note**
>
> 如果您之前未创建默认 Amazon EMR 服务角色和 EC2 实例配置文件，请先使用 `aws emr create-default-roles` 创建它们，然后再使用 `create-cluster` 命令。

Example 示例：按需型主实例、带单个实例类型的按需型核心实例、默认 VPC

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}
```

Example 示例：竞价型主实例、带单个实例类型的竞价型核心实例、默认 VPC

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

Example 示例：按需型主实例、带单个实例类型的混合核心实例、单个 EC2 子网

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c'] \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=2,TargetSpotCapacity=6,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=2}']
```

Example 示例：按需型主实例、带多个权重实例类型的竞价型核心实例、竞价型实例的超时、EC2 子网的范围

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=11,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
```

```
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
```

Example 示例：按需型主实例、带多个权重实例类型的混合核心实例和任务实例、核心竞价型实例的超时、EC2 子网的范围

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 \
    InstanceFleetType=CORE,TargetOnDemandCapacity=8,TargetSpotCapacity=6,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
 \
    InstanceFleetType=TASK,TargetOnDemandCapacity=3,TargetSpotCapacity=3,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}']
```

Example 示例：竞价型主实例、无核心实例或任务实例、Amazon EBS 配置、默认 VPC

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole
 \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetSpotCapacity=1,\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=60,TimeoutAction=TERMINATE_CLU
\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,\
EbsConfiguration={EbsOptimized=true,EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=gp2
\
SizeIn GB=100}},{VolumeSpecification={VolumeType=io1,SizeInGB=100,Iop
 s=100},VolumesPerInstance=4}]}}']
```

Example 示例：多个自定义 AMI、多种实例类型、按需型主实例、按需型核心实例

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole
 \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
```

```
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}']
```

Example 示例：将任务节点添加到具有多种实例类型和多个自定义 AMI 的正在运行的集群

```
aws emr add-instance-fleet --cluster-id j-123456 --release-label Amazon EMR 5.3.1 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleet \
    InstanceFleetType=Task,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}',\
'{InstanceType=m6g.xlarge,CustomAmiId=ami-234567}']
```

Example 示例：使用 JSON 配置文件

您可以在 JSON 文件中配置实例队列参数，然后将 JSON 文件引用为实例队列的唯一参数。例如，以下命令引用 JSON 配置文件 *my-fleet-config.json*：

```
aws emr create-cluster --release-label emr-5.30.0 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets file://my-fleet-config.json
```

my-fleet-config.json 文件指定主实例集、核心实例集和任务实例集，如下例所示。核心实例集使用以按需价格的百分比表示的最高 Spot 价格 (BidPrice)，而任务实例集和主实例集使用以 USD 表示的最高 Spot 价格 (BidPriceAsPercentageofOnDemandPrice)（字符串形式）。

```
[
    {
        "Name": "Masterfleet",
        "InstanceFleetType": "MASTER",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "SWITCH_TO_ON_DEMAND"
            }
        },
        "InstanceTypeConfigs": [
```

```
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    },
    {
        "Name": "Corefleet",
        "InstanceFleetType": "CORE",
        "TargetSpotCapacity": 1,
        "TargetOnDemandCapacity": 1,
        "LaunchSpecifications": {
          "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price",
            "CapacityReservationOptions":
            {
                "UsageStrategy": "use-capacity-reservations-first",
                "CapacityReservationResourceGroupArn": "String"
            }
        },
            "SpotSpecification": {
                "AllocationStrategy": "capacity-optimized",
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPriceAsPercentageOfOnDemandPrice": 100
            }
        ]
    },
    {
        "Name": "Taskfleet",
        "InstanceFleetType": "TASK",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
          "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price",
            "CapacityReservationOptions":
            {
                "CapacityReservationPreference": "none"
            }
```

```
        },
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    }
]
```

修改实例集的目标容量

使用 modify-instance-fleet 命令可指定实例队列的新目标容量。您必须指定集群 ID 和实例队列 ID。使用 list-instance-fleets 命令可检索实例队列 ID。

```
aws emr modify-instance-fleet --cluster-id <cluster-id> \
  --instance-fleet \
    InstanceFleetId='<instance-fleet-id>',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

将任务实例集添加到集群

如果集群仅具有主实例集和核心实例集，您可以使用 add-instance-fleet 命令添加任务实例集。您只能使用此命令添加任务实例队列。

```
aws emr add-instance-fleet --cluster-id <cluster-id>
  --instance-fleet \
    InstanceFleetType=TASK,TargetSpotCapacity=1,\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=20,TimeoutAction=TERMINATE_CLU
\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

获取集群中的实例集的配置详细信息

使用 list-instance-fleets 命令可获取集群中的实例队列的配置详细信息。该命令将集群 ID 用作输入。以下示例演示了针对包含主任务实例组和核心任务实例组的集群的命令及其输出。有关完整响应语法，请参阅《Amazon EMR API 参考》中的 ListInstanceFleets。

```
list-instance-fleets --cluster-id <cluster-id>
```

```
{
    "InstanceFleets": [
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759094.637,
                    "CreationDateTime": 1488758719.817
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 6,
            "Name": "CORE",
            "InstanceFleetType": "CORE",
            "LaunchSpecifications": {
                "SpotSpecification": {
                    "TimeoutDurationMinutes": 60,
                    "TimeoutAction": "TERMINATE_CLUSTER"
                }
            },
            "ProvisionedOnDemandCapacity": 2,
            "InstanceTypeSpecifications": [
                {
                    "BidPrice": "0.5",
                    "InstanceType": "m5.xlarge",
                    "WeightedCapacity": 2
                }
            ],
            "Id": "if-1ABC2DEFGHIJ3"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759058.598,
                    "CreationDateTime": 1488758719.811
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
```

```
                }
            },
            "ProvisionedSpotCapacity": 0,
            "Name": "MASTER",
            "InstanceFleetType": "MASTER",
            "ProvisionedOnDemandCapacity": 1,
            "InstanceTypeSpecifications": [
                {
                    "BidPriceAsPercentageOfOnDemandPrice": 100.0,
                    "InstanceType": "m5.xlarge",
                    "WeightedCapacity": 1
                }
            ],
           "Id": "if-2ABC4DEFGHIJ4"
        }
    ]
}
```

将容量预留与实例集结合使用

要启动带有容量预留选项的按需型实例集，请附加使用容量预留选项所需的额外服务角色权限。由于容量预留选项必须与按需分配策略一起使用，因此您还必须在服务角色和托管式策略中提供分配策略所需的权限。有关更多信息，请参阅分配策略权限。

Amazon EMR 支持开放容量预留和目标容量预留。以下主题显示了可与 RunJobFlow 操作或 `create-cluster` 命令结合使用的实例集配置，以还使用按需型容量预留启动实例集。

尽力使用开放容量预留

如果集群的按需型实例与您账户中可用的开放容量预留属性（实例类型、平台、租赁和可用区）相匹配，则会自动应用容量预留。但是，不能保证使用您的容量预留。为了预置集群，Amazon EMR 评估启动请求中指定的所有实例池，并使用价格最低且具有足够容量的实例池来启动所有请求的核心节点。将自动应用与实例池匹配的可用开放容量预留。如果可用的开放容量预留与实例池不匹配，它们将保持未使用状态。

预置核心节点后，就会选择并修复可用区。Amazon EMR 将任务节点预置到选定可用区中的实例池中，首先从价格最低的节点开始，直到预置完所有任务节点。将自动应用与实例池匹配的可用开放容量预留。

以下是尽力使用开放容量预留 Amazon EMR 容量分配逻辑的使用案例。

示例 1: 启动请求中价格最低的实例池具有可用的开放容量预留

在这种情况下，Amazon EMR 在具有按需型实例的价格最低的实例池中启动容量。将自动使用此实例池中可用的开放容量预留。

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available Open capacity reservations	150	100	100
On-Demand Price	$	$$	$$$
预置的实例	100	-	-
已使用的开放容量预留	100	-	-
可用的开放容量预留	50	100	100

在启动队列后，您可以运行 `describe-capacity-reservations` 来查看剩余的未使用的容量预留量。

示例 2: 启动请求中价格最低的实例池没有可用的开放容量预留

在这种情况下，Amazon EMR 在具有按需型实例的价格最低的实例池中启动容量。但是，您的开放容量预留仍处于未使用状态。

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
可用的开放容量预留	-	-	100
On-Demand Price	$	$$	$$$

预置的实例	100	-	-
已使用的开放容量预留	-	-	-
可用的开放容量预留	-	-	100

配置实例机群以便尽力使用开放容量预留

您使用 RunJobFlow 操作创建基于实例集的集群时，请将按需分配策略设置为 lowest-price 和 CapacityReservationPreference，以让容量预留选项保持 open。或者，如果您将此字段留空，Amazon EMR 会将按需型实例的容量预留首选项默认设置为 open。

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "CapacityReservationPreference": "open"
         }
        }
    }
```

此外，您还可以使用 Amazon EMR CLI 创建使用开放容量预留、基于实例集的集群。

```
aws emr create-cluster \
 --name 'open-ODCR-cluster' \
 --release-label emr-5.30.0 \
 --service-role EMR_DefaultRole \
 --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
 --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
   LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={CapacityReservationPreference=open}}'}
```

其中，

- `open-ODCR-cluster` 替换为使用开放容量预留的集群的名称。
- `subnet-22XXXX01` 替换为子网 ID。

首先使用开放容量预留

在预置 Amazon EMR 集群时，您可以选择覆盖价格最低的分配策略，并将首先使用可用的开放容量预留作为首要优先级。在这种情况下，Amazon EMR 会评估在启动请求中指定容量预留的所有实例池，并使用具有足够容量且价格最低的实例池来启动所有请求的核心节点。如果具有容量预留的实例池没有足够容量用于请求的核心节点，Amazon EMR 会回退到上一主题中描述的尽力案例。也就是说，Amazon EMR 会重新评估启动请求中指定的所有实例池，并使用价格最低且具有足够容量的实例池启动所有请求的核心节点。将自动应用与实例池匹配的可用开放容量预留。如果可用的开放容量预留与实例池不匹配，它们将保持未使用状态。

预置核心节点后，就会选择并修复可用区。Amazon EMR 将任务节点预置到选定可用区中具有容量预留的实例池中，首先从价格最低的实例池开始，直到预置完所有任务节点。Amazon EMR 首先使用选定可用区中每个实例池中可用的开放容量预留，并且仅在需要时使用价格最低的策略来预置所有剩余的任务节点。

以下是首先使用开放容量预留的 Amazon EMR 容量分配逻辑的使用案例。

示例 1：启动请求中具有可用开放容量预留的实例池具有足够的核心节点容量

在这种情况下，Amazon EMR 在具有可用开放容量预留的实例池中启动容量，而不考虑实例池价格。因此，会在预置所有核心节点之前，尽可能使用您的开放容量预留。

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Usage Strategy	use-capacity-reservations-first		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available Open capacity reservations	-	-	150
On-Demand Price	$	$$	$$$
预置的实例	-	-	100

已使用的开放容量预留	-	-	100
可用的开放容量预留	-	-	50

示例 2：启动请求中具有可用开放容量预留的实例池没有足够的容量用于核心节点

在这种情况下，Amazon EMR 回退到使用价格最低的策略启动核心节点，并尽力使用容量预留。

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Usage Strategy	use-capacity-reservations-first		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available Open capacity reservations	10	50	50
On-Demand Price	$	$$	$$$
预置的实例	100	-	-
已使用的开放容量预留	10	-	-
可用的开放容量预留	-	50	50

在启动队列后，您可以运行 describe-capacity-reservations 来查看剩余的未使用的容量预留量。

将实例机群配置为首先使用开放容量预留

使用 RunJobFlow 操作创建基于势例集的集群时，请将按需分配策略设置为 lowest-price，并将 CapacityReservationOptions 的 UsageStrategy 设置为 use-capacity-reservations-first。

```
"LaunchSpecifications":
```

```
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "UsageStrategy": "use-capacity-reservations-first"
         }
       }
    }
```

您还可以使用 Amazon EMR CLI 创建首先使用容量预留的基于实例机群的集群。

```
aws emr create-cluster \
  --name 'use-CR-first-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-first}}'}
```

其中，

- `use-CR-first-cluster` 替换为使用开放容量预留的集群的名称。
- `subnet-22XXXX01` 替换为子网 ID。

首先使用目标容量预留

预置 Amazon EMR 集群时，您可以选择覆盖价格最低的分配策略，并将首先使用可用的目标容量预留作为首要优先级。在这种情况下，Amazon EMR 会评估在启动请求中指定目标预留的所有实例池，并选择具有足够容量的价格最低的实例池来启动所有请求的核心节点。如果具有目标容量预留的实例池没有足够容量用于请求的核心节点，Amazon EMR 会回退到上一主题中描述的尽力案例。也就是说，Amazon EMR 会重新评估启动请求中指定的所有实例池，并选择价格最低且具有足够容量的实例池启动所有请求的核心节点。将自动应用与实例池匹配的可用开放容量预留。但是，目标容量预留仍处于使用状态。

预置核心节点后，就会选择并修复可用区。Amazon EMR 将任务节点预置到选定可用区中具有目标容量预留的实例池中，首先从价格最低的实例池开始，直到预置完所有任务节点。Amazon EMR 首先尝试使用选定可用区中每个实例池中可用的目标容量预留。然后，Amazon EMR 仅在需要时使用价格最低的策略来预置所有剩余的任务节点。

以下是首先使用目标容量预留的 Amazon EMR 容量分配逻辑的使用案例。

示例 1: 启动请求中具有可用目标容量预留的实例池具有足够的核心节点容量

在这种情况下，Amazon EMR 在具有可用目标容量预留的实例池中启动容量，而不考虑实例池价格。因此，会在预置所有核心节点之前，尽可能使用您的目标容量预留。

On-Demand Strategy	lowest-price		
Usage Strategy	use-capacity-reservations-first		
Requested Capacity	100		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available targeted capacity reservations	-	-	150
On-Demand Price	$	$$	$$$
预置的实例	-	-	100
已使用的目标容量预留	-	-	100
可用的目标容量预留	-	-	50

Example 示例 2：启动请求中具有可用目标容量预留的实例池没有足够的容量用于核心节点

On-Demand Strategy	lowest-price
Requested Capacity	100
Usage Strategy	use-capacity-reservations-first

Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available targeted capacity reservations	10	50	50
On-Demand Price	$	$$	$$$
预置的实例	100	-	-
已使用的目标容量预留	-	-	-
可用的目标容量预留	10	50	50

在启动队列后，您可以运行 describe-capacity-reservations 来查看剩余的未使用的容量预留量。

将实例机群配置为首先使用目标容量预留

使用 RunJobFlow 操作创建基于实例集的集群时，请将按需分配策略设置为 lowest-price，并将 CapacityReservationOptions 的 UsageStrategy 设置为 use-capacity-reservations-first、将 CapacityReservationOptions 的 CapacityReservationResourceGroupArn 设置为 <your resource group ARN>。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用容量预留。

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "UsageStrategy": "use-capacity-reservations-first",
            "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:sa-
east-1:123456789012:group/MyCRGroup"
         }
       }
    }
```

其中，将 arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup 替换为您的资源组 ARN。

您还可以使用 Amazon EMR CLI 创建首先使用目标容量预留的基于实例机群的集群。

```
aws emr create-cluster \
  --name 'targeted-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \
    InstanceFleetType=CORE,TargetOnDemandCapacity=100,\
InstanceTypeConfigs=['{InstanceType=c5.xlarge},{InstanceType=m5.xlarge},
{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-
first,CapacityReservationResourceGroupArn=arn:aws:resource-groups:sa-
east-1:123456789012:group/MyCRGroup}}'}
```

其中，

- 将 `targeted-CR-cluster` 替换为使用目标容量预留的集群的名称。
- `subnet-22XXXX01` 替换为子网 ID。
- 将 `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` 替换为您的资源组 ARN。

避免使用可用的开放容量预留

Example

如果您希望避免在启动 Amazon EMR 集群时意外使用任何开放容量预留，请将按需分配策略设置为 `lowest-price`，并将 `CapacityReservationOptions` 的 `CapacityReservationPreference` 设置为 `none`。否则，Amazon EMR 会将按需型实例的容量预留首选项默认设置为 `open`，并尝试尽力使用可用的开放容量预留。

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "CapacityReservationPreference": "none"
         }
```

```
        }
    }
```

您还可以在不使用任何开放容量预留的情况下，使用 Amazon EMR CLI 创建基于实例集的集群。

```
aws emr create-cluster \
  --name 'none-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={CapacityReservationPreference=none}}'}
```

其中，

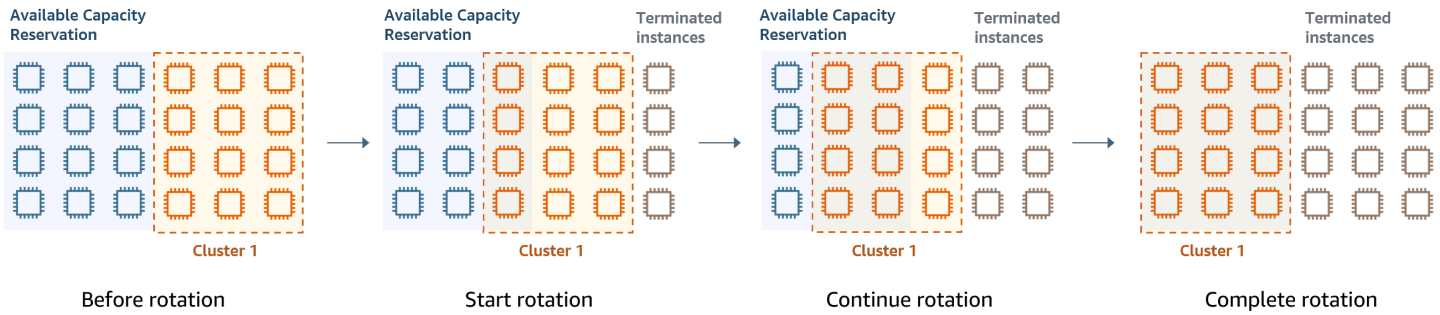
- 将 `none-CR-cluster` 替换为不使用任何开放容量预留的集群的名称。
- `subnet-22XXXX01` 替换为子网 ID。

使用容量预留的方案

在以下情况下，使用容量预留可以获得益处。

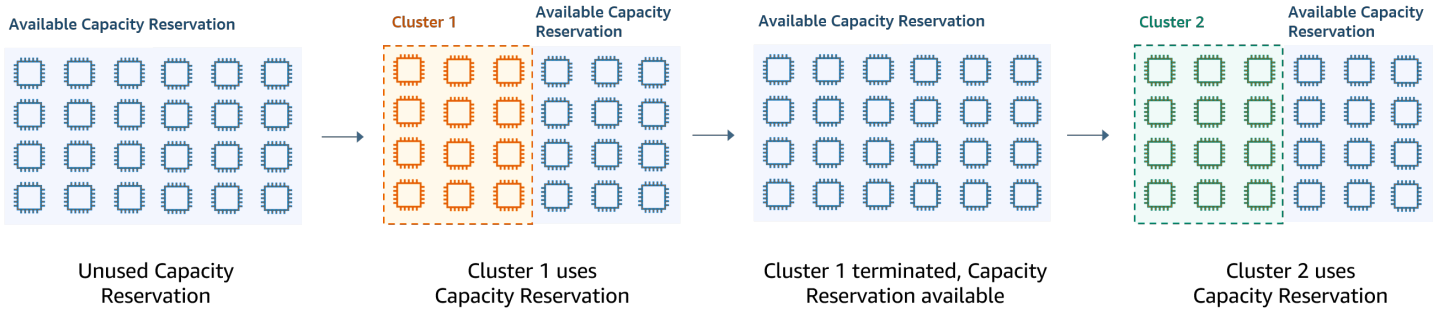
情况 1：使用容量预留轮替长时间运行的集群

在轮替一个长时间运行的集群时，您可能对预置新实例的实例类型和可用区有严格的要求。通过容量预留，您可以使用容量保证来完成集群轮替，且不会发生中断。



方案 2：使用容量预留预置连续的短期集群

您还可以使用容量预留为单个工作负载配置一组连续的短期集群，以便在终止集群时，下一个集群可以使用容量预留。您可以使用目标容量预留来确保仅预期集群使用容量预留。



配置统一实例组

对于实例组配置，每个节点类型（主节点、核心节点或任务节点）均包含相同的实例类型和相同的实例购买选项：按需或 Spot。在创建实例组时，您可以指定这些设置。它们在以后均不可更改。但您可以将类型和购买选项相同的实例添加到核心和任务实例组。您也可以删除实例。

如果集群的按需型实例与您账户中可用的开放容量预留属性（实例类型、平台、租赁和可用区）相匹配，则会自动应用容量预留。您可以对主节点、核心节点和任务节点使用开放容量预留。但是，使用实例组预置集群时，不能使用目标容量预留，也不能阻止在具有匹配属性的开放容量预留中启动实例。如果要使用目标容量预留或阻止在开放容量预留中启动实例，请改用实例集。有关更多信息，请参阅将容量预留与实例集结合使用。

要在创建集群后添加不同的实例类型，您可以添加其它任务实例组。您可以为每个实例组选择不同的实例类型和购买选项。有关更多信息，请参阅使用集群扩展。

启动实例时，按需实例的容量预留首选项默认设置为 open，这样按需实例才能在任何具有匹配属性（实例类型、平台、可用区）的开放容量预留中运行。有关按需容量预留的更多信息，请参阅将容量预留与实例集结合使用。

此部分介绍如何使用统一实例组创建集群。有关通过手动添加或删除实例或使用自动扩展来修改现有实例组的更多信息，请参阅管理集群。

使用控制台配置统一实例组

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台通过实例组创建集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster configuration（集群配置）下，选择 Instance groups（实例组）。
4. 在 Node groups（节点组）下，每种类型的节点组都有一个部分。对于主节点组，如果需要 3 个主节点，请选中 Use multiple primary nodes（使用多个主节点）复选框。如果要使用 Spot 购买，请选中 Use Spot purchasing option（使用 Spot 购买选项）复选框。
5. 对于主节点组和核心节点组，选择 Add instance type（添加实例类型）并选择最多 5 个实例类型。对于任务组，选择 Add instance type（添加实例类型）并选择最多 15 个实例类型。Amazon EMR 可在启动集群时预置这些实例类型的任意组合。
6. 在每个节点组类型下，选择每个实例旁边的 Actions（操作）下拉菜单以更改以下设置：

   添加 EBS 卷

   在 Amazon EMR 预置实例类型后，指定要附加到实例类型的 EBS 卷。

   编辑最高 Spot 价格

   为实例集中的每个实例类型指定最高 Spot 价格。您可以采用按需价格的百分比形式或特定美元金额形式设置此价格。如果可用区中的当前 Spot 价格低于最高 Spot 价格，Amazon EMR 会预置竞价型实例。您按 Spot 价格付费，但这不一定是最高 Spot 价格。
7. 或者，扩展节点配置以输入 JSON 配置或从 Amazon S3 加载 JSON。
8. 选择适用于集群的任何其他选项。
9. 要启动集群，选择 Create cluster（创建集群）。

Old console

在创建集群时，以下过程涵盖了 Advanced options (高级选项)。此外，借助 Quick options (快速选项)，可使用实例组配置创建集群。

使用旧控制台通过统一实例组创建集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)，输入 Software Configuration (软件配置) 选项，然后选择 Next (下一步)。
4. 在 Hardware Configuration (硬件配置) 屏幕中，将 Uniform instance groups (统一实例组) 保持选中状态。
5. 选择 Network (网络)，然后选择您的集群的 EC2 Subnet (EC2 子网)。您选择的子网将与可用性组关联，后者与每个子网一起列出。有关更多信息，请参阅配置联网。

   > **Note**
   >
   > 您的账户和区域可能为您提供了为 Network (网络) 选择 Launch into EC2-Classic (在 EC2-Classic 中启动) 的选项。如果您选择该选项，请选择 EC2 Availability Zone (EC2 可用区) 而非 EC2 Subnet (EC2 子网)。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 和 Amazon VPC。

6. 在每个 Node type (节点类型) 行中：

   - 如果您需要更改实例组的默认名称，请在 Node type (节点类型) 下，选择铅笔图标，然后输入易识别的名称。若要删除 Task (任务) 实例组，请选择 X 图标。选择 Add task instance group (添加任务实例组) 以添加其它 Task (任务) 实例组。
   - 在 Instance type (实例类型) 下面，选择铅笔图标，然后选择要用于该节点类型的实例类型。

     > **Important**
     >
     > 使用Amazon Web Services Management Console选择实例类型时，每个实例类型显示的 vCPU 数量是该实例的 YARN vcore 数量，而不是该实例类型的 EC2 vCPU 数量。有关每个实例类型 vCPU 数量的更多信息，请参阅 Amazon EC2 实例类型。

   - 在 Instance type (实例类型) 下面，选择 Configurations (配置) 的铅笔图标，然后编辑每个实例组的应用程序配置。
   - 在 Instance count (实例计数) 下，输入要用于每个节点类型的实例的数量。

- 在 Purchasing option (购买选项) 下，选择 On-demand (按需) 或 Spot。如果选择 Spot，请为竞价型实例选择最高价的选项。默认情况下，选择 Use on-demand as max price (使用按需价格作为最高价格)。您可以选择 Set max $/hr (设定最高价($/小时))，然后输入您的最高价。您选择的 EC2 Subnet (EC2 子网) 的可用区价格低于 Maximum Spot price (最高 Spot 价格)。

  > **Tip**
  >
  > 将鼠标移到 Spot 的信息工具提示上方可查看当前区域中可用区的当前 Spot 价格。最低 Spot 价格用绿色显示。您可能需要使用此信息来更改您的 EC2 Subnet (EC2 子网) 选择。

- 在 Auto Scaling for Core and Task node types (自动扩展核心和任务节点类型) 下，选择铅笔图标，然后配置自动扩展选项。有关更多信息，请参阅将自动伸缩与实例组的自定义策略结合使用 。

7. 根据需要选择 Add task instance group (添加任务实例组) 并按照上一步中所述配置设置。
8. 选择 Next (下一步)，修改其它集群设置，然后启动集群。

使用 Amazon CLI 通过统一实例组创建集群

要使用 Amazon CLI 指定集群的实例组配置，请将 `create-cluster` 命令与 `--instance-groups` 参数结合使用。除非您为实例组指定 `BidPrice` 实参，否则 Amazon EMR 将使用按需型实例选项。有关使用按需实例和各种集群选项启动统一实例组的 `create-cluster` 命令的示例，请在命令行上键入 `aws emr create-cluster help` ，或参阅《Amazon CLI 命令参考》中的 create-cluster。

您可以使用 Amazon CLI 在集群中创建使用竞价型实例的统一实例组。提供的 Spot 价格取决于可用区。在使用 CLI 或 API 时，您可以使用 `--ec2-attributes` 形参的 `AvailabilityZone` 实参 (如果您使用的是 EC2-classic 网络) 或 `SubnetID` 实参来指定可用区。您选择的可用区或子网适用于集群，因此可用于所有实例组。如果您未明确指定可用区或子网，Amazon EMR 将在启动集群时选择具有最低 Spot 价格的可用区。

以下示例演示了 `create-cluster` 命令，该命令创建主实例组、核心实例组和两个任务实例组，这些实例组均使用竞价型实例。将 *`myKey`* 替换为您的 Amazon EC2 密钥对的名称。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字符 (^)。

```
aws emr create-cluster --name "MySpotCluster" \
  --release-label emr-5.36.1 \
  --use-default-roles \
  --ec2-attributes KeyName=myKey \
  --instance-groups \
    InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,BidPrice=0.25 \
    InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.03 \
    InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=4,BidPrice=0.03 \
    InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.04
```

通过 CLI，您可以创建统一的实例组集群，为实例组的每种实例类型指定唯一的自定义 AMI。这允许您在同一实例组中使用不同的实例架构。每种实例类型都必须使用具有匹配架构的自定义 AMI。例如，您可以使用 x86_64 架构自定义 AMI 配置 m5.xlarge 实例类型和带有相应的AWS AARCH64 (ARM) 架构自定义 AMI 的 m6g.xlarge 实例类型。

以下示例显示了使用两种实例类型创建的统一实例组集群，每种实例类型都有自定义 AMI。请注意，自定义 AMI 仅在实例类型级别指定，而不是在集群级别指定。这是为了避免集群级别的实例类型 AMI 和 AMI 之间的冲突，这将导致集群启动失败。

```
aws emr create-cluster
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-groups \

 InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
 \

 InstanceGroupType=CORE,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-234567
```

您可以向添加到正在运行的集群的实例组中添加多个自定义 AMI。`CustomAmiId`参数可以与`add-instance-groups`命令一起使用，如以下示例所示。

```
aws emr add-instance-groups --cluster-id j-123456 \
```

```
  --instance-groups \

 InstanceGroupType=Task,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
```

使用 Java SDK 创建实例组

您实例化一个 InstanceGroupConfig 对象，此对象指定集群的实例组配置。要使用竞价型实例，您可以在 withBidPrice 对象上设置 withMarket 和 InstanceGroupConfig 属性。以下代码演示了如何定义运行竞价型实例的主实例组、核心实例组和任务实例组。

```
InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
 .withInstanceCount(1)
 .withInstanceRole("MASTER")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.25");

InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
 .withInstanceCount(4)
 .withInstanceRole("CORE")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.03");

InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
 .withInstanceCount(2)
 .withInstanceRole("TASK")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.10");
```

实例和可用区灵活性的最佳实践

每个 Amazon Web Services 区域 都有多个相互隔离的位置，称为可用区。当您启动实例时，可以选择指定您所用 Amazon Web Services 区域 中的可用区（AZ）。可用区灵活性是指多个可用区中的实例分配。如果一个实例发生故障，您可以设计您的应用程序，使另一个可用区中的实例可以处理请求。有关可用区的更多信息，请参阅《Amazon EC2 用户指南》中的区域和可用区文档。

实例灵活性是指使用多种实例类型来满足容量要求。使用实例表现灵活性时，可以跨实例大小、系列和代系使用总容量。与使用单一实例类型的集群相比，更大的灵活性有助于提高查找和分配所需计算容量的能力。

与具有单一实例类型或可用区的集群相比，实例和可用区灵活性能够减少容量不足错误（ICE）和竞价型中断。在了解初始实例系列和大小后，请使用此处介绍的最佳实践来确定需要实现多样化的实例。这种方法可实现最低的性能和成本差异，同时最大限度提高 Amazon EC2 容量池的可用性。

灵活选择可用区

建议配置所有可用区以在虚拟私有云（VPC）中使用，并为 EMR 集群选择这些可用区。集群必须仅存在于一个可用区中，但通过 Amazon EMR 实例集，您可以为不同的可用区选择多个子网。当 Amazon EMR 启动集群时，它会在这些子网中查找您指定的实例和购买选项。当您为多个子网配置 EMR 集群时，与单个子网中的集群相比，您的集群可以访问更充足的 Amazon EC2 容量池。

如果您必须优先考虑一定数量的可用区，以便在 EMR 集群的虚拟私有云（VPC）中使用，则可以利用 Amazon EC2 的竞价型放置分数功能。使用竞价型放置评分，您可以指定竞价型实例的计算要求，然后 EC2 会按照 1 到 10 的等级返回得分排名前十的 Amazon Web Services 区域 或可用区。分数为 10 表示您的竞价型请求成功的可能性很大；分数为 1 表示您的竞价型请求不太可能成功。有关如何使用竞价放置分数的更多信息，请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的竞价放置分数。

灵活选择实例类型

实例灵活性是指使用多种实例类型来满足容量要求。实例灵活性有利于 Amazon EC2 竞价型实例和按需型实例的使用。借助竞价型实例，实例灵活性允许 Amazon EC2 使用实时容量数据从更充足的容量池启动实例。此外，实例灵活性还可以预测哪些实例的可用性最高，从而减少中断并降低工作负载的整体成本。借助按需型实例，当在更多的实例池中配置总容量时，实例灵活性可减少容量不足错误（ICE）。

对于 Instance Group（实例组）集群，最多可指定 50 个 EC2 实例类型。对于具有分配策略的 Instance Fleets（实例集），每个主节点组、核心节点组和任务节点组最多可指定 30 个 EC2 实例类型。实例范围越大，实例灵活性的优势就越多。

表现实例灵活性

请考虑以下最佳实践来表现应用程序的实例灵活性。

主题

- 确定实例系列和大小
- 包含其他实例

确定实例系列和大小

Amazon EMR 支持针对不同用例的多个实例类型。支持的实例类型 文档中列出了这些实例类型。每个实例类型都属于一个实例系列，该系列描述了该类型针对哪些应用程序进行了优化。

对于新的工作负载，应该使用通用系列中的实例类型进行基准测试，例如 m5 或 c5。然后，从 Ganglia 和 Amazon CloudWatch 监控操作系统和 YARN 指标，以确定负载达到峰值时的系统瓶颈。瓶颈包括 CPU、内存、存储和 I/O 操作。确定瓶颈后，为实例类型选择计算优化、内存优化、存储优化或其他合适的实例系列。有关更多详细信息，请参阅 GitHub 上 Amazon EMR best practices guide（《Amazon EMR 最佳实践指南》）中的 Determine right infrastructure for your Spark workloads（为 Spark 工作负载确定合适的基础设施）页面。

接下来，确定应用程序需要的最小 YARN 容器或 Spark 执行程序。这是适合容器的最小实例大小，也是集群的最小实例大小。使用此指标来确定可进一步实现多元化的实例。实例越小，实例灵活性就越高。

为了获得最大的实例灵活性，您应该利用尽可能多的实例。建议使用硬件规格相似的实例实现多元化。这可以最大限度地访问 EC2 容量池，同时将成本和性能差异降至最低。实现不同大小实例的多元化。为此，首先要确定 Amazon Graviton 和上一代实例的优先级。一般来说，对于每种工作负载，应在至少 15 种实例类型之间进行灵活选择。建议从通用型实例、计算优化型实例或内存优化型实例开始。这些实例类型的灵活性最大。

包含其他实例

包含其他实例类型，以便最大限度地实现多元化。首先确定实例大小、Graviton 和代系灵活性的优先级。这将允许访问其他具有类似成本和性能配置的 EC2 容量池。如果由于 ICE 或竞价型实例中断而需要提高灵活性，请考虑变体和系列灵活性。每种方法都需要根据您的使用案例和要求进行权衡。

- 大小灵活性：首先，使用同系列不同大小的实例实现多元化。尽管同系列的实例成本和性能相同，但可以在每台主机上启动不同数量的容器。例如，如果所需的最小执行程序大小为 2vCPU 和 8Gb 内存，则最小实例大小为 m5.xlarge。为实现大小灵活性，请将 m5.xlarge、m5.2xlarge、m5.4xlarge、m5.8xlarge、m5.12xlarge、m5.16xlarge 和 m5.24xlarge 包含在内。
- Graviton 灵活性：除大小外，您还可以通过 Graviton 实例实现多元化。Graviton 实例由 Amazon Graviton2 处理器提供支持，可为 Amazon EC2 中的云工作负载提供最佳性价比。例如，如果最小实例大小为 m5.xlarge，则可将 m6g.xlarge、m6g.2xlarge、m6g.4xlarge、m6g.8xlarge 和 m6g.16xlarge 包含在内，以确保实现 Graviton 的灵活性。

- 代系灵活性：与 Graviton 和大小灵活性类似，上一代系列中的实例具有相同的硬件规格。因此，成本和性能配置相似，同时增加了可访问的 Amazon EC2 池总量。为确保代系灵活性，请将 `m4.xlarge`、`m4.2xlarge`、`m4.10xlarge` 和 `m4.16xlarge` 包含在内。
- 系列和变体灵活性
  - 容量：为了优化容量，建议在实例系列中灵活使用实例。来自不同实例系列的通用实例的实例池更充足，有助于满足容量要求。但是，实例系列不同，vCPU 与内存比率也有所不同。如果预期的应用程序容器大小适合不同的实例，会导致利用不足。例如，对于 `m5.xlarge`，可将计算优化型实例（如 `c5`）或内存优化型实例（如 `r5`）包含在内，以确保实例系列灵活性。
  - 成本：为了优化成本，我们建议在不同变体之间灵活使用实例。这些实例的内存与 vCPU 比率与初始实例相同。变体灵活性的劣势在于，这些实例的容量池较小，可能导致额外容量受限或竞价型实例中断增加。例如对于 `m5.xlarge`，可将基于 AMD 的实例 (`m5a`)、基于 SSD 的实例 (`m5d`) 或网络优化实例 (`m5n`) 包含在内，以实现变体灵活性。

集群配置的最佳实践

使用此部分中的准则可帮助您决定要为 EMR 集群中的每个节点类型预置的实例类型、购买选项和存储量。

您应使用哪个实例类型？

可通过多种方式将 Amazon EC2 实例添加到集群。选取何种方式取决于您对集群使用实例组配置还是实例集配置。

- 实例组
  - 手动将同一类型的实例添加到现有的核心和任务实例组。
  - 手动添加任务实例组，该组可使用其它实例类型。
  - 在 Amazon EMR 中为实例组设置自动伸缩，并根据您指定的 Amazon CloudWatch 指标值来自动添加和删除实例。有关更多信息，请参阅使用集群扩展。
- 实例队列
  - 添加一个任务实例队列。
  - 对于现有的核心和任务实例队列，更改按需实例和竞价型实例的目标容量。有关更多信息，请参阅配置实例集。

规划集群实例的一个方法是，使用代表性的示例数据集运行测试集群并监控集群中节点的利用。有关更多信息，请参阅查看和监控集群。另一种方法是计算您在考虑的实例的容量，并将该容量值与数据的大小做比较。

通常，分配任务的主节点类型不需要 EC2 实例和很高的处理能力；核心节点类型的 Amazon EC2 实例（处理任务并将数据存储在 HDFS 中）需要处理能力和存储容量；任务节点类型的 Amazon EC2 实例（不存储数据）只需要处理能力。有关可用的 Amazon EC2 实例及其配置的准则，请参阅配置 Amazon EC2 实例。

以下指导方针适用于大部分 Amazon EMR 集群。

- 对于您可以在每个Amazon Web Services 区域的每个Amazon账户中运行的按需 Amazon EC2 实例的总数，存在 vCPU 限制。有关 vCPU 限制以及如何请求为账户提高限制的更多信息，请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的按需实例。
- 主节点通常没有大型计算需求。对于具有大量节点的集群，或者对于主节点上专门部署了应用程序（JupyterHub、Hue 等）的集群，可能需要一个较大的主节点，这有助于提高集群性能。例如，对于小型集群（50 个或更少的节点），考虑使用 m5.xlarge 实例；对于较大的实例类型，考虑增加更大的集群类型。
- 核心及任务节点的计算需求取决于您的应用程序将执行的处理类型。许多任务能够在通用型实例类型上运行，该类型在 CPU、磁盘空间和输入/输出方面提供了均衡的性能。计算密集型集群可能受益于在 CPU 增强型实例上运行，这些实例的 CPU 在比例上比 RAM 更高。数据库和内存缓存应用程序可能受益于在内存增强型实例上运行。网络密集型和 CPU 密集型应用程序（如解析、NLP 和机器学习）可能受益于在集群计算实例上运行，这些实例提供高比例的 CPU 资源和增强的网络性能。
- 如果集群的不同阶段有不同的容量需求，您可以开始使用少量核心节点，并随时增加或减少任务节点的数量来满足任务流程不断变化的容量要求。
- 您可以处理的数据量取决于核心节点的容量以及作为输入的数据、处理期间的数据和作为输出的数据的大小。在处理期间，输入、中间和输出数据集都存储在集群上。

什么时候应该使用竞价型实例？

在 Amazon EMR 中启动集群时，您可以选择在竞价型实例上启动主实例、核心实例或任务实例。因为每类实例组在集群中发挥的作用不同，所以，在竞价型实例上启动每个节点类型都会有影响。集群在运行时，您不能更改实例的购买选项。要将主节点和核心节点从按需型实例更改为竞价型实例或者反向操作，您必须终止该集群，并启动一个新集群。对于任务节点，您可以启动新的任务实例组或实例队列，并删除旧的任务实例组或实例队列。

主题

- 用于防止因任务节点竞价型实例终止而导致任务失败的 Amazon EMR 设置
- 竞价型实例上的主节点
- Spot 实例上的核心节点

- Spot 实例上的任务节点
- 适用于应用程序场景的实例配置

用于防止因任务节点竞价型实例终止而导致任务失败的 Amazon EMR 设置

由于竞价型实例通常用于运行任务节点，而 Amazon EMR 具有调度 YARN 任务的默认功能，因此在竞价型实例上运行的任务节点终止时，正在运行的任务不会失败。Amazon EMR 通过允许应用程序主进程仅在核心节点上运行来实现这一目标。应用程序主进程控制正在运行的任务，并且需要在任务的整个生命周期内保持活动状态。

Amazon EMR 发行版 5.19.0 及更高版本使用内置 YARN 节点标签功能来实现这一目标。（早期版本使用代码补丁）。`yarn-site` 中的属性和 `capacity-scheduler` 配置分类是默认配置，以便 YARN capacity-scheduler 和 fair-scheduler 利用节点标注。Amazon EMR 自动为核心节点添加 `CORE` 标注，并设置属性，以便只在具有 CORE 标签的节点上安排应用程序主节点。手动修改 yarn-site 和 capacity-scheduler 配置分类中的相关属性，或直接在关联的 XML 文件中修改相关属性，可能会破坏此功能或调整此功能。

默认情况下，Amazon EMR 会配置以下属性和值。配置这些属性时要谨慎。

- yarn-site (yarn-site.xml)（在所有节点上）
  - `yarn.node-labels.enabled: true`
  - `yarn.node-labels.am.default-node-label-expression: 'CORE'`
  - `yarn.node-labels.fs-store.root-dir: '/apps/yarn/nodelabels'`
  - `yarn.node-labels.configuration-type: 'distributed'`
- yarn-site (yarn-site.xml) 在主节点和核心节点上
  - `yarn.nodemanager.node-labels.provider: 'config'`
  - `yarn.nodemanager.node-labels.provider.configured-node-partition: 'CORE'`
- capacity-scheduler (capacity-scheduler.xml)（在所有节点上）
  - `yarn.scheduler.capacity.root.accessible-node-labels: '*'`
  - `yarn.scheduler.capacity.root.accessible-node-labels.CORE.capacity: 100`
  - `yarn.scheduler.capacity.root.default.accessible-node-labels: '*'`
  - `yarn.scheduler.capacity.root.default.accessible-node-labels.CORE.capacity: 100`

> **Note**
>
> 从 Amazon EMR 6.x 发行版系列开始，默认情况下禁用 YARN 节点标注功能。默认情况下，应用程序主进程可以在核心节点和任务节点上运行。您可以通过配置以下属性来启用 YARN 节点标注功能：
>
> - `yarn.node-labels.enabled: true`
> - `yarn.node-labels.am.default-node-label-expression: 'CORE'`

竞价型实例上的主节点

主节点会控制和引导集群。当它终止时，集群也随之终止。因此，如果您正在运行可接受突然终止的集群，那么，您应当只以竞价型实例方式启动主节点。如果您在测试新的应用程序，或者您拥有的集群会定期地将数据保存到外部存储器（如 Amazon S3）。或者在运行的集群中成本比确保集群完成更重要，就可能属于这种情况。

以竞价型实例方式启动主实例组时，集群在竞价型实例请求实现后才会启动。在您选择最高 Spot 价格时，请考虑这个因素。

启动集群时，只能添加竞价型实例主节点。在正在运行的集群上，不能添加或删除主节点。

一般情况下，如果以竞价型实例方式运行整个集群（所有的实例组），就只能以竞价型实例方式运行主节点。

Spot 实例上的核心节点

核心节点处理数据并使用 HDFS 存储信息。终止一个核心实例会面临数据丢失的风险。因此，仅当允许丢失部分 HDFS 数据时，才能在竞价型实例上运行核心节点。

当以竞价型实例方式启动核心实例组时，Amazon EMR 会持续等候，直到可以在启动实例组前预置所有已请求的核心实例。换言之，如果您请求六个 Amazon EC2 实例，但在您的最高 Spot 价格水平及该水平以下只能获得五个节点，那么实例组将不会启动。Amazon EMR 将继续等待，直到这六个 Amazon EC2 实例都可用，或者直到您终止集群。您可以更改核心实例组中竞价型实例的数量以增加运行中集群的容量。有关如何使用实例组，以及竞价型实例如何与实例队列结合使用的更多信息，请参阅the section called “配置实例集或实例组”。

Spot 实例上的任务节点

任务节点处理数据，但不使用 HDFS 保存持久性数据。如果它们因为 Spot 价格上涨超过最高 Spot 价格而终止，那么不会丢失数据，对您的集群的影响会降低到最低程度。

当您以竞价型实例的方式启动一个或多个任务实例组时，Amazon EMR 会预置使用您的最高 Spot 价格所能获得的最多任务节点。这意味着，如果您请求一个具有六个节点的任务实例组，但在您的最高 Spot 价格水平及该水平以下只能获得五个竞价型实例，那么 Amazon EMR 将使用五个节点启动该实例组，等到以后情况允许了再添加第六个。

以竞价型实例方式启动任务实例组是在尽可能降低成本的同时扩展集群容量的一种策略。如果您以按需型实例方式启动主实例组和核心实例组，则其容量可以保证集群运行。您可以根据需要向任务实例组添加任务实例，从而处理峰值流量或者加快数据处理速度。

可以使用控制台、Amazon CLI 或 API 添加或删除任务节点。还可以添加更多任务组，但任务组一经创建将不可删除。

适用于应用程序场景的实例配置

下表是通常适用于各种应用程序场景的节点类型购买选项和配置的快速参考。选择该链接以查看有关每个场景类型的更多信息。

应用程序场景	主节点购买选项	核心节点购买选项	任务节点购买选项
长时间运行的集群和数据仓库	按需	按需或实例队列组合	Spot 或实例队列组合
成本驱动型工作负载	Spot 实例	Spot 实例	Spot 实例
数据关键型工作负载	按需	按需	Spot 或实例队列组合
应用程序测试	Spot 实例	Spot 实例	Spot 实例

有几种情况应使用竞价型实例运行 Amazon EMR 集群。

长时间运行的集群和数据仓库

如果您运行的是持续性的、可预测计算容量变化的 Amazon EMR 集群，如数据仓库，就可以通过竞价型实例以更低的成本处理峰值需求。您可以按需型实例方式启动主实例组和核心实例组来处理正常容量，并以竞价型实例方式启动任务实例组来处理峰值负载需求。

成本驱动型工作负载

如果您运行的是临时集群，降低成本比完成时间更为重要，并且可以接受丢失部分工作，那么，以竞价型实例方式运行整个集群（主实例组、核心实例组和任务实例组）可以享受最大的成本节省优势。

数据关键型工作负载

如果对于您运行的集群，降低成本比保证完成时间更为重要，但不能接受丢失部分工作，则建议您以按需型实例方式启动主实例组和核心实例组，并辅以采用竞价型实例的一个或多个任务实例组。以按需型实例方式运行主实例组和核心实例组，可确保您的数据持久保存到 HDFS 中，并且可保护集群不会因为 Spot 市场上的价格波动而终止，同时可通过以竞价型实例方式运行任务实例组来实现成本节省。

应用程序测试

当为生产环境做启动准备而测试新的应用程序时，您可以按竞价型实例的方式运行整个集群（主实例组、核心实例组和任务实例组），从而降低测试成本。

计算集群的必需的 HDFS 容量

可供集群使用的 HDFS 存储量取决于以下因素：

- 用于核心节点的 Amazon EC2 实例的数量。
- 用于使用的实例类型的 Amazon EC2 实例存储的容量。有关实例存储卷的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon Amazon EC2 实例存储。
- 附加到核心节点的 Amazon EBS 卷的数量和大小。
- 一个重复因子，它将考虑每个数据块存储在 HDFS 中以实现类似 RAID 的冗余的方式。默认情况下，对于包含 10 个或更多核心节点的集群，重复因子为 3；对于包含 4-9 个核心节点的集群，重复因子为 2；对于包含 3 个或更少核心节点的集群，重复因子为 1。

要为每个核心节点计算集群的 HDFS 容量，请将实例存储卷容量添加到 Amazon EBS 存储容量 (如果已使用)。将结果乘以核心节点数，然后将乘积除以基于核心节点数确定的复制因子。例如，某个集群具有 10 个 i2.xlarge 类型的核心节点，这些节点具有 800 GB 的实例存储 (没有附加的 Amazon EBS 卷)，该集群总共有约 2,666 GB 可用于 HDFS 的存储 (10 节点 x 800 GB ÷ 3 复制因子)。

如果计算出的 HDFS 容量值小于您的数据，则可通过以下方式增大 HDFS 存储量：

- 创建带其他 Amazon EBS 卷的集群或将带附加的 Amazon EBS 卷的实例组添加到现有集群
- 添加多个核心节点
- 选择带更大存储容量的 Amazon EC2 实例类型
- 使用数据压缩
- 更改 Hadoop 配置设置以减小重复因子

减少重复因子的方法应谨慎使用，因为这会降低 HDFS 数据的冗余程度以及集群从丢失或损坏的 HDFS 数据块中恢复的能力。

配置集群日志记录和调试

在规划集群时要确定的事项之一是您希望提供多少调试支持。当首次开发数据处理应用程序时，我们建议您在集群上测试该应用程序，处理一小部分但具有代表性的数据子集。执行此操作时，您可能希望利用 Amazon EMR 提供的所有调试工具，如将日志文件存档到 Amazon S3。

在完成开发和将数据处理应用程序投入全面生产后，您可以选择缩减调试。这样做可以节省在 Amazon S3 中存储日志文件存档的成本，并降低数据库上的处理负载，因为不再需要向 Amazon S3 写入状态。当然，如果这样做，在出现错误时可用于调查问题的工具则会相对较少。

默认日志文件

默认情况下，每个集群都会将日志文件写到主节点上。这些文件将写入 `/mnt/var/log/` 目录。如 使用 SSH 连接到主节点 中所述，您可以通过使用 SSH 连接到主节点来访问这些日志。

> **Note**
>
> 如果使用 Amazon EMR 6.8.0 发行版或更早版本，集群终止期间日志文件会保存到 Amazon S3，因此主节点终止后，将无法访问日志文件。如果使用 Amazon EMR 6.9.0 发行版及更高版本，集群缩减期间日志文件会存档到 Amazon S3，因此即使在节点终止后，集群上生成的日志文件仍然存在。

您无需启用任何功能或工具即可将日志文件写到主节点上。这是 Amazon EMR 和 Hadoop 的默认行为。

一个集群可产生多种类型的日志文件，包括：

- 步骤日志 – 这些日志由 Amazon EMR 服务生成，包含集群的相关信息和每个步骤的结果。日志文件存储在主节点上的 `/mnt/var/log/hadoop/steps/` 目录中。每个步骤都将其结果记录在单独编号的子目录中：第一步的子目录为 `/mnt/var/log/hadoop/steps/s-stepId1/`，第二步为 `/mnt/var/log/hadoop/steps/s-stepId2/`，依此类推。13 个字符的步骤标识符 (如 stepId1、stepId2) 对集群来说是唯一的。
- Hadoop 和 YARN 组件日志 – 与 Apache YARN 和 MapReduce 关联的组件日志包含在 `/mnt/var/log` 中的单独文件夹中。`/mnt/var/log` 下 Hadoop 组件的日志文件位置有：hadoop-

hdfs、hadoop-mapreduce、hadoop-httpfs 和 hadoop-yarn。hadoop-state-pusher 目录用于 Hadoop 状态推送进程的输出。

- 引导操作日志 – 如果您的任务使用引导操作，则会记录这些操作的结果。日志文件存储在主节点上的 /mnt/var/log/bootstrap-actions/ 中。每个引导操作都将其结果记录在单独编号的子目录中：第一个引导操作的子目录为 `/mnt/var/log/bootstrap-actions/1/`、第二个引导操作为 `/mnt/var/log/bootstrap-actions/2/`，依此类推。
- 实例状态日志 – 这些日志提供有关 CPU、内存状态和节点的垃圾收集器线程的信息。日志文件存储在主节点上的 `/mnt/var/log/instance-state/` 中。

将日志文件归档到 Amazon S3

> **Note**
>
> 您目前不能通过 `yarn logs` 实用工具对 Amazon S3 使用日志聚合。

如果使用 Amazon EMR 6.9.0 发行版及更高版本，集群缩减期间日志文件会存档到 Amazon S3，因此即使在节点终止后，集群上生成的日志文件仍然存在。此行为会自动启用，因此您无需执行任何操作即可将其打开。对于 Amazon EMR 发行版 6.8.0 及更早版本，您可以配置集群以定期将存储在主节点上的日志文件存档到 Amazon S3。这可确保集群终止后（无论是正常关闭或是由于错误所致），日志文件仍可用。Amazon EMR 每隔 5 分钟便将日志文件归档到 Amazon S3。

要将 Amazon EMR 发行版 6.8.0 及更早版本的日志文件存档到 Amazon S3，您必须在启动集群时启用此功能。您可以使用控制台、CLI 或 API 执行此操作。默认情况下，使用控制台启动的集群已启用日志存档。对于使用 CLI 或 API 启动的集群，必须手动启用到 Amazon S3 的日志记录。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台将日志文件存档到 Amazon S3

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。

2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster logs（集群日志）下，选中 Publish cluster-specific logs to Amazon S3（将集群特定日志发布到 Amazon S3）复选框。
4. 在 Amazon S3 location（Amazon S3 位置）字段中，键入（或浏览到）存储日志的 Amazon S3 路径。如果键入的文件夹名称在存储桶中不存在，Amazon S3 将创建该文件夹。

   设定此值后，Amazon EMR 会将日志文件从集群中的 EC2 实例复制到 Amazon S3。这可以防止在集群终止及托管集群的 EC2 实例终止时丢失日志文件。这些日志在排除故障时非常有用。有关更多信息，请参阅查看日志文件。
5. （可选）选中 Encrypt cluster-specific logs（加密集群特定的日志）复选框。然后，从列表中选择 Amazon KMS 密钥，输入密钥 ARN 或创建新密钥。此选项仅适用于 Amazon EMR 5.30.0 及更高版本（不包括 6.0.0）。要使用此选项，请为您的 EC2 实例配置文件和 Amazon EMR 角色向 Amazon KMS 添加权限。有关更多信息，请参阅使用 Amazon KMS 客户托管密钥加密存储在 Amazon S3 中的日志文件。
6. 选择适用于集群的任何其他选项。
7. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台将日志文件存档到 Amazon S3

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 在一般选项部分中的日志记录字段中，接受默认选项：已启用。

   此字段决定 Amazon S3 是否将详细日志数据捕获到 Amazon EMR。您只能在创建集群时设置此选项。有关更多信息，请参阅查看日志文件。
5. 在 S3 folder 字段中，键入（或浏览到）用于存储日志的 Amazon S3 路径。您也可以让控制台为您生成一个 Amazon S3 路径。如果键入的文件夹名称在存储桶中不存在，系统将创建该文件夹。

   设定此值后，Amazon EMR 会将日志文件从集群中的 EC2 实例复制到 Amazon S3。这可以防止在集群结束及托管集群的 EC2 实例终止时丢失日志文件。这些日志在排除故障时非常有用。

有关更多信息，请参阅查看日志文件。

6. 在 Log encryption (日志加密) 字段中，选择 Encrypt logs stored in S3 with an Amazon KMS customer managed key (使用 Amazon KMS 客户托管密钥加密存储在 S3 中的日志)。然后，从列表中选择 Amazon KMS 密钥或输入密钥 ARN。您也可以创建新的 Amazon KMS 密钥。

   此选项仅适用于 Amazon EMR 5.30.0 及更高版本（不包括 6.0.0）。要使用此选项，请为您的 EC2 实例配置文件和 Amazon EMR 角色向 Amazon KMS 添加权限。有关更多信息，请参阅使用 Amazon KMS 客户托管密钥加密存储在 Amazon S3 中的日志文件。

7. 按照计划和配置集群中所述继续创建集群。

CLI

使用 Amazon CLI 将日志文件存档到 Amazon S3

要使用 Amazon CLI 将日志文件归档到 Amazon S3，请键入 `create-cluster` 命令并使用 `--log-uri` 参数指定 Amazon S3 日志路径。

1. 要将文件记录到 Amazon S3，请键入以下命令并将 *myKey* 替换为您的 EC2 密钥对的名称。

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.36.1 --log-uri s3://DOC-EXAMPLE-BUCKET/logs --applications Name=Hadoop Name=Hive Name=Pig
 --use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

2. 如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

> **Note**
>
> 如果您之前未创建默认 Amazon EMR 服务角色和 EC2 实例配置文件，请先输入 `aws emr create-default-roles` 进行创建，然后键入 `create-cluster` 子命令。

使用 Amazon KMS 客户托管密钥加密存储在 Amazon S3 中的日志文件

利用 Amazon EMR 版本 5.30.0 及更高版本（Amazon EMR 6.0.0 除外），您可以使用 Amazon KMS 客户托管密钥加密存储在 Amazon S3 中的日志文件。要在控制台中启用此选项，请按照将日志文件归

档到 Amazon S3中的步骤进行操作。您的 Amazon EC2 实例配置文件和 Amazon EMR 角色必须满足以下先决条件：

- 用于您的集群的 Amazon EC2 实例配置文件必须有权使用 `kms:GenerateDataKey`。
- 用于您的集群的 Amazon EMR 角色必须有权使用 `kms:DescribeKey`。
- 必须将 Amazon EC2 实例配置文件和 Amazon EMR 角色添加到指定的 Amazon KMS 客户托管密钥的密钥用户列表中，如以下步骤所示：
  1. 从 https://console.aws.amazon.com/kms 打开 Amazon Key Management Service（Amazon KMS）控制台。
  2. 要更改 Amazon 区域，请使用页面右上角的区域选择器。
  3. 选择要修改的 KMS 密钥的别名。
  4. 在密钥详细信息页面的 Key Users (密钥用户) 下，选择 Add (添加)。
  5. 在 Add key users (添加密钥用户)对话框中，选择您的 Amazon EC2 实例配置文件和 Amazon EMR 角色。
  6. 选择 Add（添加）。

有关更多信息，请参阅 Amazon EMR 使用的 IAM 服务角色、《Amazon Key Management Service 开发人员指南》中的使用密钥策略。

使用 Amazon CLI 在 Amazon S3 中聚合日志

> **Note**
>
> 您目前不能通过 `yarn logs` 实用工具使用日志聚合。您只能使用此过程支持的聚合。

日志聚合 (Hadoop 2.x) 将日志从单个应用程序的所有容器汇总到单个文件中。要使用 Amazon CLI 启用对 Amazon S3 的日志聚合，请在集群启动时使用引导操作启用日志聚合并指定用于存储日志的存储桶。

- 要启用日志聚合，请创建名为 `myConfig.json` 的下列配置文件，其中包含以下内容：

```
[
  {
    "Classification": "yarn-site",
    "Properties": {
      "yarn.log-aggregation-enable": "true",
```

```
      "yarn.log-aggregation.retain-seconds": "-1",
      "yarn.nodemanager.remote-app-log-dir": "s3:\/\/DOC-EXAMPLE-BUCKET\/logs"
    }
  }
]
```

请键入以下命令，将 *myKey* 替换为您的 EC2 密钥对的名称。您还可以将任何红色文本替换为自己的配置。

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-5.36.1 \
--applications Name=Hadoop \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-type m5.xlarge \
--instance-count 3 \
--configurations file://./myConfig.json
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

> **Note**
>
> 如果您之前未创建默认 EMR 服务角色和 EC2 实例配置文件，请先运行 `aws emr create-default-roles` 以创建它们，然后再运行 `create-cluster` 子命令。

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅 Amazon CLI 命令参考。

日志位置

以下列表包括了所有日志类型及其在 Amazon S3 中的位置。您可以使用这些日志来排查 Amazon EMR 问题。

步骤日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/steps/<step-id>/
```

应用程序日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/containers/
```

此位置包含容器 stderr 以及 stdout、directory.info、prelaunch.out 和 launch_container.sh 日志。

资源管理器日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/
applications/hadoop-yarn/
```

Hadoop HDFS

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/
applications/hadoop-hdfs/
```

此位置包含 NameNode、DataNode 和 YARN TimelineServer 日志。

节点管理器日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/
applications/hadoop-yarn/
```

实例状态日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/daemons/
instance-state/
```

Amazon EMR 预调配日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/
provision-node/*
```

Hive 日志

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/
applications/hive/*
```

- 要查找集群上的 Hive 日志，请删除星号（*）并将 /var/log/hive/ 附加上面的链接后。
- 要查找 HiveServer2 日志，请删除星号（*）并将 var/log/hive/hiveserver2.log 附加上面的链接后。
- 要查找 HiveCLI 日志，请删除星号（*）并将 /var/log/hive/user/hadoop/hive.log 附加上面的链接后。
- 要查找 Hive Metastore Server 日志，请删除星号（*）并将 /var/log/hive/user/hive/hive.log 附加上面的链接后。

如果故障位置在 Tez 应用程序的主节点或任务节点上，请提供相应的 Hadoop 容器日志。

启用调试工具

借助调试工具，您可以更轻松地从 Amazon EMR 控制台浏览日志文件。有关更多信息，请参阅查看调试工具中的日志文件。当您对集群启用调试时，Amazon EMR 将日志文件存档到 Amazon S3，然后为这些文件建立索引。然后，您就可以使用控制台以直观的方式浏览集群的步骤、作业、任务和任务尝试日志。

要在 Amazon EMR 控制台中使用调试工具，您必须在使用控制台、CLI 或者 API 启动集群时启用调试。请注意，Amazon EMR 新控制台不提供调试工具。

Old console

使用旧控制台打开调试工具

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 选择 Go to advanced options (转到高级选项)。
4. 在 Cluster Configuration (集群配置) 部分的 Logging (日志记录) 字段中，选择 Enabled (已启用)。不能在未启用日志记录的情况下启用调试。
5. 在 Log folder S3 location 字段中，键入用于存储日志的 Amazon S3 路径。
6. 在 Debugging (调试) 字段中，选择 Enabled (已启用)。此调试选项创建一个 Amazon SQS 交换，以将调试消息发布到 Amazon EMR 服务后端。将消息发布到此交换功能可能发生费用。有关更多信息，请参阅 Amazon SQS 产品页面。
7. 按照计划和配置集群中所述继续创建集群。

Amazon CLI

使用 Amazon CLI 打开调试工具

要使用 Amazon CLI 启用调试，请键入带 `--enable-debugging` 参数的 `create-cluster` 子命令。您还必须在启用调试时指定 `--log-uri` 参数。

- 要使用 Amazon CLI 启用调试，请键入以下命令并将 *myKey* 替换为您的 EC2 密钥对的名称。

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-5.36.1 \
--log-uri s3://DOC-EXAMPLE-BUCKET/logs \
```

```
--enable-debugging \
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-type m5.xlarge \
--instance-count 3
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

> **Note**
>
> 如果您之前未创建默认 EMR 服务角色和 EC2 实例配置文件，请先键入 `aws emr create-default-roles` 创建它们，然后再键入 `create-cluster` 子命令。

API

使用 Amazon EMR API 打开调试工具

- 使用以下 Java SDK 配置启用调试。

```
    StepFactory stepFactory = new StepFactory();
StepConfig enabledebugging = new StepConfig()
    .withName("Enable debugging")
    .withActionOnFailure("TERMINATE_JOB_FLOW")
    .withHadoopJarStep(stepFactory.newEnableDebuggingStep());
```

在此示例中，`new StepFactory()` 使用 `us-east-1` 作为默认区域。如果您的集群是在其它区域中启动的，则需要使用 `new StepFactory("region.elasticmapreduce")`（例如，`new StepFactory("ap-northeast-2.elasticmapreduce")`）指定该区域。

调试选项信息

Amazon EMR 发行版 4.1.0 至 5.27.0 支持在所有区域进行调试。其他 Amazon EMR 版本不支持调试选项。自 2023 年 1 月 23 日起，Amazon EMR 将停用所有版本的调试工具。

Amazon EMR 创建一个 Amazon SQS 队列来处理调试数据。消息可能发生费用。但是，Amazon SQS 拥有含多达 100 万个请求的免费套餐。有关更多信息，请参阅https://www.amazonaws.cn/https://aws.amazon.com/sqs。

调试需要用到角色；您的服务角色和实例配置文件必须让您能够使用所有的 Amazon SQS API 操作。如果您的角色挂载到 Amazon EMR 托管式策略，则您无需对角色做出任何修改。如果您有自定义角色，则需要添加 `sqs:*` 权限。有关更多信息，请参阅为 Amazon EMR 配置 IAM 服务角色对Amazon 服务和资源的权限。

为集群添加标签

可以方便地通过不同方式对 Amazon 分类；例如，按用途、拥有者或环境分类。您可以通过使用标签将自定义元数据分配给 Amazon EMR，在 Amazon EMR 中实现分类。标签包含您定义的一个键和一个值。对于 Amazon EMR，集群是您可以进行标记的资源级内容。例如，您可以为账户的集群定义一组标签，用于帮助您跟踪每个集群的拥有者，或者标识是生产集群还是测试集群。我们建议您创建一组一致的标签以满足您的组织要求。

向 Amazon EMR 集群添加某个标签时，该标签也会传播到与该集群关联的每个活动 Amazon EC2 实例。同样，从 Amazon EMR 集群中删除某个标签时，该标签会从每个关联的活动 Amazon EC2 实例中删除。

> **Important**
>
> 应使用 Amazon EMR 控制台或 CLI（而不是 Amazon EC2 控制台或 CLI）对属于集群的 Amazon EC2 实例进行标签管理，因为在 Amazon EC2 中进行的更改不会同步回 Amazon EMR 标记系统。

您可以通过查找以下系统标签来标识属于 Amazon EMR 集群的 Amazon EC2 实例。在此示例中，*`CORE`* 是实例组角色的值，*`j-12345678`* 是示例任务流程（集群）标识符值：

- aws:elasticmapreduce:instance-group-role=*`CORE`*
- aws:elasticmapreduce:job-flow-id=*`j-12345678`*

> **Note**
>
> Amazon EMR 和 Amazon EC2 将您的标签解释为没有语义含义的字符串。

可以使用 Amazon Web Services Management Console、CLI 和 API 处理标签。

可以在创建新 Amazon EMR 集群时添加标签，并且可以对正在运行的 Amazon EMR 集群添加、编辑或删除标签。编辑标签是适用于 Amazon EMR 控制台的概念，但是使用 CLI 和 API 时，要编辑标签，请删除旧标签并添加新标签。您可以编辑标签的键和值，并且可以在集群正在运行期间随时从资源中删除标签。但是，您不能对已终止集群或是曾与仍处于活动状态的集群关联的已终止实例添加、编辑或删除标签。此外，您还可以将标签的值设为空的字符串，但是不能将其设为 null。

如果按标签将 Amazon Identity and Access Management（IAM）与 Amazon EC2 实例一起用于基于资源的权限，则 IAM policy 会应用于 Amazon EMR 传播到集群的 Amazon EC2 实例的标签。要使 Amazon EMR 标签能够传播到您的 Amazon EC2 实例，需要向您用于 Amazon EC2 的 IAM policy 授予调用 Amazon EC2 CreateTags 和 DeleteTags API 的权限。传播的标签还可能会影响 Amazon EC2 的基于资源的权限。传播到 Amazon EC2 的标签可以读取为 IAM policy 中的条件，正如其它 Amazon EC2 标签一样。向 Amazon EMR 集群添加标签时应考虑 IAM policy，以免用户拥有不正确的集群权限。要避免这些问题，请确保 IAM policy 不包含您还计划用于 Amazon EMR 集群的标签的条件。有关更多信息，请参阅控制对Amazon EC2 资源的访问。

标签限制

下面是适用于标签的基本限制：

- 适用于 Amazon EC2 资源的限制也适用于 Amazon EMR。有关更多信息，请参阅https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions。
- 请勿在标签名称和值中使用 `aws:` 前缀，因为它是保留标签，专供 Amazon 使用。此外，您还无法编辑或删除带此前缀的标签名称或值。
- 您无法对已终止的集群更改或编辑标签。
- 标签值可以是空字符串，但是不能是 null。此外，标签的键不能是空字符串。
- 键和值可以包含任何语言的任何字母字符、任何数字字符、空格、不可见的分隔符和以下符号：_ . : / = + - @

有关使用 Amazon Web Services Management Console添加标签的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的在控制台中使用标签。有关使用 Amazon EC2 API 或命令行添加标记的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 API 和 CLI 概览。

针对账单为资源添加标签

您可以使用标签来组织Amazon账单，使其反映您自己的成本结构。要执行此操作，请注册以获取包含标签键值的 Amazon 账户账单。随后可以按标签的键值组织账单信息，以查看组合资源的成本。虽然

Amazon EMR 和 Amazon EC2 具有不同的账单语句，但是每个集群上的标签也设置在每个关联实例上，以便您可以使用标签来链接相关 Amazon EMR 和 Amazon EC2 成本。

例如，您可以将特定的应用程序名称用作几个资源的标签，然后组织账单信息，以查看在数个服务中的使用该应用程序的总成本。有关更多信息，请参阅《Amazon Billing 用户指南》中的成本分配和添加标签。

向集群添加标签

您可以在创建集群时向其添加标签。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

创建集群时使用新控制台添加标签

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Tags（标签）下，选择 Add new tag（添加新标签）。在 Key（密钥）字段中指定标签。或者在 Value（值）字段中指定标签。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

创建集群时使用旧控制台添加标签

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 依次选择 Create cluster (创建集群)、Go to advanced options (转到高级选项)。

3. 在 Step 3: General Cluster Settings (步骤 3: 一般集群设置) 页面上的 Tags (标签) 部分中，键入标签的 Key (密钥)。

   开始键入 Key (密钥) 时，另一个新行自动出现，以便为下一个新标签提供空间。

4. （可选）键入标签的 Value (值)。
5. 对要向集群添加的每个标签键/值对重复之前的步骤。集群启动时，您输入的所有标签都会自动与集群关联。

Amazon CLI

创建集群时使用 Amazon CLI 添加标签

以下示例演示如何使用 Amazon CLI 向新集群添加标签。要在创建集群时添加标签，请键入带有 `create-cluster` 参数的 `--tags` 子命令。

- 要在创建集群时添加键值为 *`marketing`* 且名为 *`costCenter`* 的标签，请键入以下命令，将 *`myKey`* 替换为您的 EC2 密钥对的名称。

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hadoop Name=Hive Name=Pig --tags "costCenter=marketing" --
use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --
instance-count 3
```

如果不使用 `--instance-groups` 参数指定实例计数，则将启动单个主节点，其余实例将作为核心节点启动。所有节点都使用该命令中指定的实例类型。

> **Note**
>
> 如果您之前未创建默认 EMR 服务角色和 EC2 实例配置文件，请先键入 aws emr `create-default-roles` 创建它们，然后再键入 `create-cluster` 子命令。

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

您也可以向现有集群添加标签。

New console

使用新控制台向现有集群添加标签

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Tags（标签）选项卡上，选择 Manage tags（管理标签）。在 Key（密钥）字段中指定标签。或者在 Value（值）字段中指定标签。
4. 选择保存更改。Tags（标签）选项卡会更新为集群拥有的新标签数。例如，如果您现在有两个标签，则选项卡的标签为 Tags (2) [标签（2）]。

Old console

使用旧控制台向现有集群添加标签

1. 在 Amazon EMR 控制台中，选择 Cluster List (集群列表) 页面，然后单击要添加标签的集群。
2. 在 Cluster Details (集群详细信息) 页面上的 Tags (标签) 字段中，单击 View All/Edit (查看全部/编辑)。
3. 在 View All/Edit (查看全部/编辑) 页面上，单击 Add (添加)。
4. 单击 Key (键) 列中的空字段，然后键入您的键名称。
5. 可以选择单击 Value (值) 列中的空字段，然后键入您的值名称。
6. 您每开始一个新标签，就会有另一个空标签行出现在您当前编辑的标签下。对每个要添加的标签在新标签行上重复之前的步骤。

Amazon CLI

使用 Amazon CLI 向正在运行的集群添加标签

- 输入带有 `--tag` 参数的 `add-tags` 子命令可将标签分配给集群 ID。可以使用控制台或 `list-clusters` 子命令查找集群 ID。`add-tags` 子命令当前仅接收一个资源 ID。

  例如，要向正在运行的集群添加两个标签（一个标签的键名为 *`costCenter`*，键值为 *`marketing`*；另一个标签的键名为 *`other`*，键值为 *`accounting`*），请输入下面的命令，并将 *`j-KT4XXXXXXXX1NM`* 替换为您的集群 ID。

```
aws emr add-tags --resource-id j-KT4XXXXXXXX1NM --tag "costCenter=marketing" --
tag "other=accounting"
```

请注意，使用 Amazon CLI 添加标签时，命令没有输出。有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

查看集群上的标签

如果您要查看与集群关联的所有标签，可以使用控制台或 Amazon CLI 进行查看。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台查看集群上的标签

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 要查看您的所有标签，请在集群详细信息页面上选择 Tags（标签）选项卡。

Old console

使用旧控制台查看集群上的标签

1. 在 Amazon EMR 控制台中，选择 Cluster List (集群列表) 页面，然后单击集群以查看标签。
2. 在 Cluster Details (集群详细信息) 页面上的 Tags (标签) 字段中，一些标签会显示在此处。单击 View All/Edit (查看全部/编辑) 以显示集群上的所有可用标签。

Amazon CLI

使用 Amazon CLI 查看集群上的标签

要使用 Amazon CLI 查看集群上的标签，请键入带有 `--query` 参数的 `describe-cluster` 子命令。

- 要查看集群的标签，请键入以下命令，将 *j-KT4XXXXXXXX1NM* 替换为您的集群 ID。

```
aws emr describe-cluster --cluster-id j-KT4XXXXXX1NM --query Cluster.Tags
```

输出会显示有关集群的所有标签信息，类似于以下内容：

```
Value: accounting     Value: marketing
Key: other            Key: costCenter
```

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

从集群中删除标签

如果您不再需要标签，则可以从集群中将其删除。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台删除集群上的标签

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Tags（标签）选项卡上，选择 Manage tags（管理标签）。
4. 对于要删除的每个键值对，选择 Remove（删除）。

5. 选择 Save changes（保存更改）。

Old console

使用旧控制台删除集群上的标签

1. 在 Amazon EMR 控制台中，选择 Cluster List (集群列表) 页面，然后单击要从中删除标签的集群。
2. 在 Cluster Details (集群详细信息) 页面上的 Tags (标签) 字段中，单击 View All/Edit (查看全部/编辑)。
3. 在 View All/Edit (查看全部/编辑) 对话框中，单击要删除的标签旁边的 X 图标，然后单击 Save (保存)。
4. （可选）对要从集群中删除的每个标签键值对重复之前的步骤。

Amazon CLI

使用 Amazon CLI 删除集群上的标签

键入带有 `--tag-keys` 参数的 `remove-tags` 子命令。删除标签时，只指定键名称。

- 要从集群中删除标签，请键入以下命令，将 *j-KT4XXXXXXXX1NM* 替换为集群 ID。

```
aws emr remove-tags --resource-id j-KT4XXXXXX1NM --tag-keys "costCenter"
```

> **Note**
>
> 当前无法使用一个命令删除多个标签。

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

驱动程序和第三方应用程序集成

您可以在 Amazon EMR 上运行几款流行的大数据应用程序，承担实用工具定价。这意味着，您在集群运行期间为第三方应用程序额外支付标称小时费用。这可以让您使用应用程序，而不必购买年度许可。以下部分介绍您可以与 EMR 结合使用的一些工具。

主题

- 将商业智能工具与 Amazon EMR 结合使用

将商业智能工具与 Amazon EMR 结合使用

您可以将常用商业智能工具（如 Microsoft Excel、MicroStrategy、QlikView 和 Tableau）与 Amazon EMR 结合使用来探索和显示您的数据。许多这类工具都需要 ODBC (开放式数据库连接) 或 JDBC (Java 数据库连接) 驱动程序。下载并安装最新驱动程序，请参阅 http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/。

要查找较旧版本的驱动程序，请参阅 http://awssupportdatasvcs.com/bootstrap-actions/Simba/。

Amazon EMR 中的安全性

Amazon 十分重视云安全性。作为 Amazon 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是 Amazon 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云的安全性 – Amazon负责保护在Amazon云中运行Amazon服务的基础设施。Amazon还向您提供可安全使用的服务。作为 Amazon 合规性计划的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于 Amazon EMR 的合规性计划，请参阅合规性计划范围内的Amazon服务。
- 云中的安全性 - 您的责任由您使用的 Amazon 服务决定。您还需要对其它因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

该文档帮助您了解如何在使用 Amazon EMR 时应用责任共担模式。当您在 Amazon EMR 上开发解决方案时，根据您的业务需求，使用以下技术帮助确保集群资源和数据的安全。本章中的主题向您演示如何配置 Amazon EMR 和使用其它Amazon服务，以满足您的安全性和合规性目标。

安全配置

Amazon EMR 中的安全配置是不同安全设置的模板。您可以创建一个安全配置以在创建集群时很方便地重复使用安全设置。有关更多信息，请参阅使用安全配置设置集群安全性。

数据保护

您可以实施数据加密来帮助保护 Amazon S3 中的静态数据、集群实例存储中的静态数据以及传输中的数据。有关更多信息，请参阅加密静态数据和传输中的数据。

Amazon Identity and Access Management 与 Amazon EMR 结合使用

Amazon Identity and Access Management (IAM) 是一种 Amazon 服务，可以帮助管理员安全地控制对 Amazon 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和授权（具有权限）使用 Amazon EMR 资源。IAM 是一个可以免费使用的 Amazon 服务。

- IAM 基于身份的策略 – IAM policy 允许或拒绝用户和组执行操作的权限。策略可以与标签结合使用来按集群控制访问。有关更多信息，请参阅适用于 Amazon EMR 的 Amazon Identity and Access Management。
- IAM 角色 – Amazon EMR 服务角色、实例配置文件和服务相关角色控制 Amazon EMR 访问其它 Amazon服务的方式。有关更多信息，请参阅为 Amazon EMR 配置 IAM 服务角色对Amazon服务和资源的权限。
- 用于处理 EMRFS 对 Amazon S3 请求的 IAM 角色 – 当 Amazon EMR 访问 Amazon S3 时，您可以根据用户、组或 Amazon S3 中 EMRFS 数据的位置指定要使用的 IAM 角色。这让您能够精确地控制集群用户是否可以访问 Amazon EMR 中的文件。有关更多信息，请参阅为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色。

Kerberos

您可以将 Kerberos 设置为通过私有密钥加密来提供强大的身份验证。有关更多信息，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。

Lake Formation

您可以将 Lake Formation 权限与 Amazon Glue 数据目录结合使用，在 Amazon Glue 数据目录中提供对数据库和表的精细列级别访问。Lake Formation 可实现从企业身份系统对 EMR Notebooks 或 Apache Zeppelin 的联合身份单点登录。有关更多信息，请参阅将 Amazon EMR 与 Amazon Lake Formation 集成。

Secure Socket Shell（SSH）

SSH 帮助为用户提供连接到集群实例上的命令行的安全方式。它还提供了隧道来查看应用程序在主节点上托管的 Web 界面。客户端可以使用 Kerberos 或 Amazon EC2 密钥对进行身份验证。有关更多信息，请参阅对 SSH 凭证使用 EC2 密钥对和连接到集群。

Amazon EC2 安全组

安全组充当 EMR 集群实例的虚拟防火墙，可限制入站和出站网络流量。有关更多信息，请参阅使用安全组控制网络流量。

更新适用于 Amazon EMR 的默认 Amazon Linux AMI

> **Important**
>
> 运行 Amazon Linux 或 Amazon Linux 2 AMI（Amazon Linux Machine Image）的 Amazon EMR 集群使用默认的 Amazon Linux 行为，且不会自动下载和安装需要重新启动的重要关键内核更新。这与运行默认 Amazon Linux AMI 的其它 Amazon EC2 实例的行为相同。如果需要重新启动的新 Amazon Linux 软件更新（例如内核、NVIDIA 和 CUDA 更新）在 Amazon EMR 版本发布后可用，则运行默认 AMI 的 Amazon EMR 集群实例不会自动下载和安装这些更新。要获取内核更新，您可以自定义 Amazon EMR AMI，以使用最新的 Amazon Linux AMI。

根据您的应用程序的安全状况和集群运行的时长，您可选择定期重启集群以应用安全更新，或创建引导操作以自定义软件包安装和更新。也可以选择在正在运行的集群实例上测试然后安装所选安全更新。有关更多信息，请参阅在 Amazon EMR 使用默认的 Amazon Linux AMI。请注意，您的联网配置必须允许 HTTP 和 HTTPS 传出到 Amazon S3 中的 Amazon Linux 存储库，否则安全更新将失败。

使用安全配置设置集群安全性

利用 Amazon EMR 发行版 4.8.0 或更高版本，可以使用安全配置来配置数据加密、Kerberos 身份验证（在发行版 5.10.0 及更高版本中提供）和 EMRFS 的 Amazon S3 授权（在发行版 5.10.0 及更高版本中提供）。

在创建安全配置之后，在创建集群时指定它，并且可以为任意数量的集群重复使用它。

您可以使用控制台、Amazon Command Line Interface (Amazon CLI) 或 Amazon 开发工具包创建安全配置。您还可以使用 Amazon CloudFormation 模板创建安全配置。有关更多信息，请参阅 Amazon CloudFormation 用户指南和 AWS::EMR::SecurityConfiguration 的模板参考。

主题

- 创建安全配置
- 为集群指定安全配置

创建安全配置

本主题介绍使用 EMR 控制台和 Amazon CLI 创建安全配置的一般过程，并介绍有关加密、身份验证和 EMRFS 的 IAM 角色的参数的参考信息。有关这些功能的更多信息，请参阅以下主题：

- 加密静态数据和传输中的数据
- 使用 Kerberos 通过 Amazon EMR 进行身份验证
- 为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色

使用控制台创建安全配置

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/emr。
2. 在导航窗格中，选择 Security Configurations (安全配置)、Create security configuration (创建安全配置)。
3. 键入安全配置的 Name (名称)。
4. 按照以下部分中所述选择 Encryption (加密)、Authentication (身份认证) 选项，然后选择 Create (创建)。

使用 Amazon CLI 创建安全配置

- 使用以下示例中所示的 `create-security-configuration` 命令。
  - 对于 *SecConfigName*，指定安全配置的名称。这是您在创建使用此安全配置的集群时指定的名称。
  - 对于 *SecConfigDef*，指定内联 JSON 结构或本地 JSON 文件的路径，例如 *file://MySecConfig.json*。JSON 参数定义 Encryption (加密)、IAM Roles for EMRFS access to Amazon S3 (用于处理 EMRFS 对 Amazon S3 的访问的 IAM 角色) 和 Authentication (身份验证) 的选项，如下面部分所述。

```
aws emr create-security-configuration --name "SecConfigName" --security-
configuration SecConfigDef
```

配置数据加密

在配置安全配置中的加密之前，创建用于加密的密钥和凭证。有关更多信息，请参阅提供通过 Amazon EMR 加密静态数据的密钥和为通过 Amazon EMR 加密来加密传输中的数据提供凭证。

在创建安全配置时，需要指定两组加密选项：静态数据加密和传输中数据加密。静态数据加密选项包括采用 EMRFS 的 Amazon S3 和本地磁盘加密。传输中加密选项为支持传输层安全性 (TLS) 的某些应用

程序启用开源加密功能。静态选项和传输中选项可以同时启用或分别启用。有关更多信息，请参阅加密静态数据和传输中的数据。

> **Note**
>
> 使用 Amazon KMS 时，加密密钥的存储和使用将产生费用。有关更多信息，请参阅 Amazon KMS 定价。

使用控制台指定加密选项

根据以下准则选择 Encryption (加密) 下的选项。

- 选择静态加密下的选项以加密存储在文件系统中的数据。

  您可以选择加密 Amazon S3 和/或本地磁盘中的数据。
- 在 S3 data encryption (S3 数据加密) 下，对于 Encryption mode (加密模式)，选择一个值以确定 Amazon EMR 如何使用 EMRFS 加密 Amazon S3 数据。

  下一步取决于所选的加密模式：

  - SSE-S3

    指定使用 Amazon S3 托管密钥进行服务器端加密。您无需执行任何其它操作，因为 Amazon S3 将为您处理密钥。
  - SSE-KMS 或 CSE-KMS

    指定使用 Amazon KMS 托管的密钥 (KMS-CSE) 进行服务器端加密或使用 Amazon KMS 托管的密钥 (CSE-KMS) 进行客户端加密。对于 Amazon KMS key，选择一个键。该密钥必须与您的 EMR 集群同在一个区域中。有关密钥要求，请参阅使用 Amazon KMS keys 加密。
  - CSE-Custom (自定义 CSE)

    指定使用自定义客户端根密钥（自定义 CSE）进行客户端加密。对于 S3 object (S3 对象)，输入您的自定义密钥提供程序 JAR 文件在 Amazon S3 中或 Amazon S3 ARN 的位置。之后，对于 Key provider class (密钥提供程序类别)，输入在实现 EncryptionMaterialsProvider 接口的应用程序中声明的类的完整类名。
- 在 Local disk encryption (本地磁盘加密) 下，为 Key provider type (密钥提供程序类型) 选择一个值。
  - Amazon KMS key

选择此选项以指定一个 Amazon KMS key。对于 Amazon KMS key，选择一个键。该密钥必须与您的 EMR 集群同在一个区域中。有关密钥要求的更多信息，请参阅使用 Amazon KMS keys 加密。

EBS 加密

当您指定 Amazon KMS 作为密钥提供程序时，可以启用 EBS 加密来加密 EBS 根设备和存储卷。要启用此选项，您必须向 EMR 服务角色 `EMR_DefaultRole` 使用您指定的 Amazon KMS key 的权限。有关密钥要求的更多信息，请参阅通过为 KMS 密钥提供额外的权限来启用 EBS 加密。

- Custom (自定义)

  选择此选项可指定自定义密钥提供程序。对于 S3 object (S3 对象)，输入您的自定义密钥提供程序 JAR 文件在 Amazon S3 中或 Amazon S3 ARN 的位置。对于 Key provider class (密钥提供程序类别)，输入在实现 EncryptionMaterialsProvider 接口的应用程序中声明的类的完整类名。这里提供的类名必需与为 CSE-Custom 提供的类名不同。

- 选择 In-transit encryption (传输中加密) 为传输中的数据启用开源 TLS 加密功能。根据以下指南选择 Certificate provider type (凭证提供程序类型)：

  - PEM

    选择此选项将使用您在 zip 文件中提供的 PEM 文件。zip 文件中需要有两个构件：privateKey.pem 和 certificateChain.pem。第三个文件 trustedCertificates.pem 是可选文件。有关详细信息，请参阅为通过 Amazon EMR 加密来加密传输中的数据提供凭证。对于 S3 object (S3 对象)，指定 zip 文件字段在 Amazon S3 中的位置或 Amazon S3 ARN。

  - Custom (自定义)

    选择此选项可指定一个自定义凭证提供程序，然后在 S3 object (S3 对象) 中输入您的自定义凭证提供程序 JAR 文件在 Amazon S3 中的位置或 Amazon S3 ARN。对于 Key provider class (密钥提供程序类别)，输入在实现 TLSArtifactsProvider 接口的应用程序中声明的类的完整类名。

使用 Amazon CLI 指定加密选项

下面部分使用示例方案，说明对不同配置和密钥提供程序的正确 --security-configuration JSON 格式，再提供 JSON 参数和相应值的参考信息。

传输中数据加密选项示例

下面的示例将说明以下情景：

- 传输中的数据加密已启用，静态数据加密已禁用。
- Amazon S3 中一个包含凭证的 zip 文件用作密钥提供程序（请参阅为通过 Amazon EMR 加密来加密传输中的数据提供凭证了解凭证要求）。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": false,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "s3://MyConfigStore/artifacts/MyCerts.zip"
   }
  }
 }
}'
```

下面的示例将说明以下情景：

- 传输中的数据加密已启用，静态数据加密已禁用。
- 使用了自定义密钥提供程序 (请参阅为通过 Amazon EMR 加密来加密传输中的数据提供凭证了解凭证要求)。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": false,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "CertificateProviderClass": "com.mycompany.MyCertProvider"
   }
  }
  }
}'
```

静态数据加密选项示例

下面的示例将说明以下情景：

- 传输中的数据加密已禁用，而静态数据加密已启用。
- SSE-S3 已用于 Amazon S3 加密。
- 本地磁盘加密使用 Amazon KMS 作为密钥提供程序。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": false,
  "EnableAtRestEncryption": true,
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "SSE-S3"
   },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "AwsKms",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   }
  }
  }
}'
```

下面的示例将说明以下情景：

- 传输中数据加密已启用且使用 ARN 引用了 Amazon S3 中包含 PEM 凭证的 ZIP 文件。
- SSE-KMS 已用于 Amazon S3 加密。
- 本地磁盘加密使用 Amazon KMS 作为密钥提供程序。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": true,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
```

```
    "S3Object": "arn:aws:s3:::MyConfigStore/artifacts/MyCerts.zip"
   }
  },
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "SSE-KMS",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "AwsKms",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   }
  }
 }
}'
```

下面的示例将说明以下情景：

- 传输中数据加密已启用且引用了 Amazon S3 中包含 PEM 凭证的 ZIP 文件。
- CSE-KMS 已用于 Amazon S3 加密。
- 本地磁盘加密使用了通过其 ARN 引用的自定义密钥提供程序。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": true,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "s3://MyConfigStore/artifacts/MyCerts.zip"
   }
  },
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "CSE-KMS",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   },
```

```
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "Custom",
    "S3Object": "arn:aws:s3:::artifacts/MyKeyProvider.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
   }
  }
 }
}'
```

下面的示例将说明以下情景：

- 利用自定义密钥提供程序启用了传输中的数据加密。
- CSE-Custom 已用于 Amazon S3 数据。
- 本地磁盘加密使用了自定义密钥提供程序。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": "true",
  "EnableAtRestEncryption": "true",
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "CertificateProviderClass": "com.mycompany.MyCertProvider"
   }
  },
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "CSE-Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
    },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
   }
  }
 }
}'
```

下面的示例将说明以下情景：

- 传输中的数据加密已禁用，而静态数据加密已启用。
- 通过 SSE-KMS 启用了 Amazon S3 加密。
- 使用多个 Amazon KMS 密钥，每个 S3 存储桶一个密钥，并且加密例外应用于这些单独的 S3 存储桶。
- 本地磁盘加密已禁用。

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
   "EncryptionConfiguration": {
     "AtRestEncryptionConfiguration": {
             "S3EncryptionConfiguration": {
           "EncryptionMode": "SSE-KMS",
           "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012",
           "Overrides": [
              {
                "BucketName": "sse-s3-bucket-name",
                "EncryptionMode": "SSE-S3"
              },
              {
                "BucketName": "cse-kms-bucket-name",
                "EncryptionMode": "CSE-KMS",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
              },
              {
                "BucketName": "sse-kms-bucket-name",
                "EncryptionMode": "SSE-KMS",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
              }
             ]
             }
           },
     "EnableInTransitEncryption": false,
      "EnableAtRestEncryption": true
  }
}'
```

下面的示例将说明以下情景：

- 传输中的数据加密已禁用，而静态数据加密已启用。
- 通过 SSE-S3 启用了 Amazon S3 加密，并且本地磁盘加密已禁用。

```
aws emr create-security-configuration --name "MyS3EncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
        "AtRestEncryptionConfiguration": {
            "S3EncryptionConfiguration": {
                "EncryptionMode": "SSE-S3"
            }
        }
     }
}'
```

下面的示例将说明以下情景：

- 传输中的数据加密已禁用，而静态数据加密已启用。
- 本地磁盘加密已启用并将 Amazon KMS 作为密钥提供程序，Amazon S3 加密已禁用。

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
        "AtRestEncryptionConfiguration": {
            "LocalDiskEncryptionConfiguration": {
                "EncryptionKeyProviderType": "AwsKms",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
     }
}'
```

下面的示例将说明以下情景：

- 传输中的数据加密已禁用，而静态数据加密已启用。

- 本地磁盘加密已启用并将 Amazon KMS 作为密钥提供程序，Amazon S3 加密已禁用。
- EBS 加密已启用。

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
        "AtRestEncryptionConfiguration": {
            "LocalDiskEncryptionConfiguration": {
                "EnableEbsEncryption": true,
                "EncryptionKeyProviderType": "AwsKms",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
     }
}'
```

加密设置的 JSON 引用

下表列出了用于加密设置的 JSON 参数并提供了每个参数的可接受值描述。

参数	描述
"EnableInTransitEncryption" : ## | false	Specify ## to enable in-transit encryption and false to disable it. If omitted, false is assumed, and in-transit encryption is disabled.
"EnableAtRestEncryption": true | false	Specify ## to enable at-rest encryption and false to disable it. If omitted, false is assumed and at-rest encryption is disabled.
传输中加密参数	
"InTransitEncryptionConfiguration" :	Specifies a collection of values used to configure in-transit encryption when EnableInTransitEncryption is ##.

参数	描述
"CertificateProviderType": "PEM" | "Custom"	Specifies whether to use PEM certificates referenced with a zipped file, or a ### certificate provider. If PEM is specified, S3Object must be a reference to the location in Amazon S3 of a zip file containing the certificates. If Custom is specified, S3Object must be a reference to the location in Amazon S3 of a JAR file, followed by a CertificateProviderClass entry.
"S3Object" : " *ZipLocation* "| "*JarLocation* "	Provides the location in Amazon S3 to a zip file when PEM is specified, or to a JAR file when ### is specified. The format can be a path (for example, s3://MyConfig/artifacts/CertFiles.zip) or an ARN (for example, arn:aws:s3:::Code/MyCertProvider.jar) . If a zip file is specified, it must contain files named exactly privateKey.pem and certificateChain.pem . A file named trustedCertificates.pem is optional.
"CertificateProviderClass" : "*MyClassID* "	Required only if ### is specified for CertificateProviderType . *MyClassID* specifies a full class name declared in the JAR file, which implements the TLSArtifactsProvider interface. For example, com.mycompany.MyCertProvider .
静态加密参数	
"AtRestEncryptionConfiguration" :	Specifies a collection of values for at-rest encryption when EnableAtRestEncryption is ##, including Amazon S3 encryption and local disk encryption.

参数	描述
Amazon S3 加密参数	
"S3EncryptionConfiguration" :	Specifies a collection of values used for Amazon S3 encryption with the EMR File System (EMRFS).
"EncryptionMode" : "SSE-S3" | "SSE-KMS" | "CSE-KMS" | "CSE-Custom"	Specifies the type of Amazon S3 encryption to use. If SSE-S3 is specified, no further Amazon S3 encryption values are required. If either SSE-KMS or CSE-KMS is specified, an Amazon KMS key ARN must be specified as the AwsKmsKey value. If CSE-Custom (### CSE) is specified, S3Object and EncryptionKeyProviderClass values must be specified.
"AwsKmsKey" : " *MyKeyARN*"	Required only when either SSE-KMS or CSE-KMS is specified for EncryptionMode . *MyKeyARN* must be a fully specified ARN to a key (for example, arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012).
"S3Object" : "*JarLocation* "	Required only when CSE-Custom (### CSE) is specified for CertificateProviderType . *JarLocation* provides the location in Amazon S3 to a JAR file. The format can be a path (for example, s3://MyConfig/artifacts/MyKeyProvider.jar) or an ARN (for example, arn:aws:s3:::Code/MyKeyProvider.jar) .

参数	描述
"EncryptionKeyProviderClass" : "*MyS3KeyClassID* "	Required only when CSE-Custom (### CSE) is specified for EncryptionMode . *MyS3KeyClassID* specifies a full class name of a class declared in the application that implements the EncryptionMaterialsProvider interface; for example, *com.mycompany.MyS3KeyProvider* .
本地磁盘加密参数	
"LocalDiskEncryptionConfiguration"	Specifies the key provider and corresponding values to be used for local disk encryption.
“EnableEbsEncryption”: true|false	Specify ## to enable EBS encryption. EBS encryption encrypts the EBS root device volume and attached storage volumes. To use EBS encryption, you must specify AwsKms as your EncryptionKeyProviderType .
"EncryptionKeyProviderType" : "AwsKms"| "Custom"	Specifies the key provider. If AwsKms is specified, an KMS key ARN must be specified as the AwsKmsKey value. If **###** is specified, S3Object and EncryptionKeyProviderClass values must be specified.
"AwsKmsKey : " *MyKeyARN*"	Required only when AwsKms is specified for ##. *MyKeyARN* must be a fully specified ARN to a key (for example, arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-456789012123).

参数	描述
"S3Object" : "*JarLocation* "	Required only when CSE-Custom (### CSE) is specified for CertificateProviderType . *JarLocation* provides the location in Amazon S3 to a JAR file. The format can be a path (for example, s3://MyConfig/artifacts/MyKeyProvider.jar) or an ARN (for example, arn:aws:s3:::Code/MyKeyProvider.jar) .
"EncryptionKeyProviderClass" : "*MyLocalDiskKeyClassID* "	Required only when ### is specified for ##. *MyLocalDiskKeyClassID* specifies a full class name of a class declared in the application that implements the EncryptionMaterialsProvider interface; for example, *com.mycompany.MyLocalDiskKeyProvider* .

配置 Kerberos 身份验证

包含 Kerberos 设置的安全配置只能由使用 Kerberos 属性创建的集群使用，否则会发生错误。有关更多信息，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。仅在 Amazon EMR 发行版 5.10.0 及更高版本中提供了 Kerberos。

使用控制台指定 Kerberos 设置

根据以下准则选择 Kerberos authentication (Kerberos 身份验证) 下的选项。

参数		描述
Kerberos		指定为使用此安全配置的集群启用 Kerberos。如果集群使用此安全配置，则集群还必须指定 Kerberos 设置，否则会发生错误。
Provider	集群专用 KDC	指定 Amazon EMR 在使用此安全配置的任何集群的主节点上创建 KDC。您可以在创建集群时指定领域名称和 KDC 管理员密码。

参数		描述
		如果需要，您可以从其它集群引用此 KDC。使用不同的安全配置创建这些集群，指定外部 KDC，并使用您为集群专用 KDC 指定的领域名称和 KDC 管理员密码。
	外部 KDC	仅在 Amazon EMR 5.20.0 版及更高版本中可用。指定使用此安全配置的集群使用集群外部的 KDC 服务器对 Kerberos 主体进行身份验证。未在集群上创建 KDC。创建集群时，您需要为外部 KDC 指定领域名称和 KDC 管理员密码。
票证生命周期		可选。指定 KDC 颁发的 Kerberos 票证在使用此安全配置的集群上有效的期间。 出于安全考虑，限制票证生命周期。集群应用程序和服务在过期后自动续订票证。使用 Kerberos 凭证通过 SSH 连接到集群的用户在票证过期后需要通过主节点命令行运行 `kinit` 来续订。
跨领域信任		指定使用此安全配置的集群上的集群专用 KDC 与不同 Kerberos 领域中的 KDC 之间的跨领域信任。 来自另一个领域的委托人（通常是用户）将通过使用此配置的集群进行身份验证。需要在其它 Kerberos 领域中进行其它配置。有关更多信息，请参阅教程：配置与 Active Directory 域的跨领域信任。
跨领域信任属性	领域	指定信任关系中其它领域的 Kerberos 领域名称。按照惯例，Kerberos 领域名称与域名相同，但全部使用大写字母。
	域	指定信任关系中其它领域的域名。

参数		描述
	管理服务器	指定信任关系的其它领域中的管理服务器的完全限定域名（FQDN）或 IP 地址。管理服务器和 KDC 服务器通常运行在具有相同 FQDN 的同一台计算机上，但通过不同的端口进行通信。 如果未指定端口，则使用端口 749，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com :749`)。
	KDC 服务器	指定信任关系的其它领域中 KDC 服务器的完全限定域名（FQDN）或 IP 地址。KDC 服务器和管理服务器通常运行在具有相同 FQDN 的同一台计算机上，但是使用不同的端口。 如果未指定端口，则使用端口 88，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com :88`)。
外部 KDC		指定集群使用集群外部 KDC。
外部 KDC 属性	管理服务器	指定外部管理服务器的完全限定域名（FQDN）或 IP 地址。管理服务器和 KDC 服务器通常运行在具有相同 FQDN 的同一台计算机上，但通过不同的端口进行通信。 如果未指定端口，则使用端口 749，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com :749`)。
	KDC 服务器	指定外部 KDC 服务器的完全限定域名（FQDN）。KDC 服务器和管理服务器通常运行在具有相同 FQDN 的同一台计算机上，但是使用不同的端口。 如果未指定端口，则使用端口 88，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com :88`)。

参数			描述
	Active Directory 集成		指定 Kerberos 委托人身份验证与 Microsoft Active Directory 域集成。
	Active Directory 集成属性	Active Directory 领域	指定 Active Directory 域的 Kerberos 领域名称。按照惯例，Kerberos 领域名称通常与域名相同，但全部使用大写字母。
		Active Directory 域	指定 Active Directory 域名。
		Active Directory 服务器	指定 Microsoft Active Directory 域控制器的完全限定域名（FQDN）。

使用 Amazon CLI 指定 Kerberos 设置

以下参考表显示了安全配置中 Kerberos 设置的 JSON 参数。有关示例配置，请参阅配置示例。

参数			描述
`"AuthenticationConfiguration": {`			对于 Kerberos 是必需的。指定身份验证配置是此安全配置的一部分。
	`"KerberosConfiguration": {`		对于 Kerberos 是必需的。指定 Kerberos 配置属性。

		`"Provider": "ClusterDedicatedKdc",`
		—或者—
		`"Provider": "ExternalKdc",`
			ClusterDedicatedKdc 指定 Amazon EMR 在使用此安全配置的任何集群的主节点上创建 KDC。您可以在创建集群时指定领域名称和 KDC 管理员密码。如果需要，您可以从其它集群引用此 KDC。使用不同的安全配置创建这些集群，指定外部 KDC，并使用您在使用集群专用 KDC 创建集群

参数	描述
	时指定的领域名称和 KDC 管理员密码。 *ExternalKdc* 指定集群使用外部 KDC。Amazon EMR 未在主节点上创建 KDC。使用此安全配置的集群必须指定外部 KDC 的领域名称和 KDC 管理员密码。
"ClusterDedicatedKdcConfiguration": {	在指定 *ClusterDedicatedKdc* 时是必需的。
"TicketLifetimeInHours": *24*,	可选。指定 KDC 颁发的 Kerberos 票证在使用此安全配置的集群上有效的期间。 出于安全考虑，限制票证生命周期。集群应用程序和服务在过期后自动续订票证。使用 Kerberos 凭证通过 SSH 连接到集群的用户在票证过期后需要通过主节点命令行运行 `kinit` 来续订。
"CrossRealmTrustConfiguration": {	指定使用此安全配置的集群上的集群专用 KDC 与不同 Kerberos 领域中的 KDC 之间的跨领域信任。 来自另一个领域的委托人（通常是用户）将通过使用此配置的集群进行身份验证。需要在其它 Kerberos 领域中进行其它配置。有关更多信息，请参阅教程：配置与 Active Directory 域的跨领域信任。

参数	描述
"Realm": "*KDC2.COM*",	指定信任关系中其它领域的 Kerberos 领域名称。按照惯例，Kerberos 领域名称与域名相同，但全部使用大写字母。
"Domain": "*kdc2.com*",	指定信任关系中其它领域的域名。
"AdminServer": "*kdc.com:749* ",	指定信任关系的其它领域中的管理服务器的完全限定域名（FQDN）或 IP 地址。管理服务器和 KDC 服务器通常运行在具有相同 FQDN 的同一台计算机上，但通过不同的端口进行通信。 如果未指定端口，则使用端口 749，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，domain.example.com *:749*)。
"KdcServer": "*kdc.com:88* "	指定信任关系的其它领域中 KDC 服务器的完全限定域名（FQDN）或 IP 地址。KDC 服务器和管理服务器通常运行在具有相同 FQDN 的同一台计算机上，但是使用不同的端口。 如果未指定端口，则使用端口 88，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，domain.example.com *:88*)。
}	
}	
"ExternalKdcConfiguration": {	在指定 *ExternalKdc* 时是必需的。

参数				描述
			"TicketLifetimeInHours": *24*,	可选。指定 KDC 颁发的 Kerberos 票证在使用此安全配置的集群上有效的期间。 出于安全考虑，限制票证生命周期。集群应用程序和服务在过期后自动续订票证。使用 Kerberos 凭证通过 SSH 连接到集群的用户在票证过期后需要通过主节点命令行运行 kinit 来续订。
			"KdcServerType": "Single",	指定引用单个 KDC 服务器。Single 是目前唯一支持的值。
			"AdminServer": "*kdc.com:749* ",	指定外部管理服务器的完全限定域名（FQDN）或 IP 地址。管理服务器和 KDC 服务器通常运行在具有相同 FQDN 的同一台计算机上，但通过不同的端口进行通信。 如果未指定端口，则使用端口 749，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，domain.example.com *:749*)。
			"KdcServer": "*kdc.com:88* ",	指定外部 KDC 服务器的完全限定域名（FQDN）。KDC 服务器和管理服务器通常运行在具有相同 FQDN 的同一台计算机上，但是使用不同的端口。 如果未指定端口，则使用端口 88，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，domain.example.com *:88*)。

参数	描述
"AdIntegrationConfiguration": {	指定 Kerberos 委托人身份验证与 Microsoft Active Directory 域集成。
"AdRealm": "*AD.DOMAIN.COM* ",	指定 Active Directory 域的 Kerberos 领域名称。按照惯例，Kerberos 领域名称通常与域名相同，但全部使用大写字母。
"AdDomain": "*ad.domain.com* "	指定 Active Directory 域名。
"AdServer": "*ad.domain.com* "	指定 Microsoft Active Directory 域控制器的完全限定域名（FQDN）。
}	
}	
}	
}	

为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色

EMRFS 的 IAM 角色让您能够为 Amazon S3 中的 EMRFS 数据提供不同权限。您可以创建角色映射，来指定在访问请求包含您指定的标识符时用来获得权限的 IAM 角色。标识符可以是 Hadoop 用户或角色，或 Amazon S3 前缀。

有关更多信息，请参阅为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色。

使用 Amazon CLI 指定 EMRFS 的 IAM 角色

以下是一个示例 JSON 代码段，用于在安全配置中为 EMRFS 指定自定义 IAM 角色。它演示了三种不同标识符类型的角色映射，后跟参数引用。

```
{
```

```
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_MyBuckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://MyBucket/","s3://MyOtherBucket/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

参数	描述
"AuthorizationConfiguration":	必需。
"EmrFsConfiguration":	必需。包含角色映射。
"RoleMappings":	必需。包含一个或多个角色映射定义。按照角色映射的先后顺序对角色映射进行评估。对于 Amazon S3 中的 EMRFS 数据调用，如果角色映射评估为 true，则不会评估其它角色映射，并且 EMRFS 将使用指定的 IAM 角色进行请求。角色映射由以下必需参数组成：
"Role":	以 arn:aws:iam:: *account-id* :role/*role-name* 格式指定 IAM 角色的 ARN 标识符。如果 EMRFS 对 Amazon S3 的请求与指定的任何 Identifiers 匹配，则这是 Amazon EMR 承担的 IAM 角色。
"IdentifierType":	可以是以下值之一：

参数	描述
	• "User" 指定标识符是一个或多个 Hadoop 用户，可以是 Linux 账户用户或 Kerberos 主体。如果 EMRFS 请求来自一个或多个指定用户，则担任 IAM 角色。 • "Prefix" 指定该标识符是 Amazon S3 位置。用于调用具有指定前缀的一个或多个位置时代入 IAM 角色。例如，前缀 s3://mybucket/ 匹配 s3://mybucket/mydir 和 s3://mybucket/yetanotherdir 。 • "Group" 指定标识符是一个或多个 Hadoop 组。如果请求来自指定组或组中的用户，则代入 IAM 角色。
"Identifiers":	指定一个或多个相应标识符类型的标识符。用逗号分隔多个标识符，不含空格。

配置到 Amazon EC2 实例的元数据服务请求

实例元数据 是有关您的实例的数据，可以用来配置或管理正在运行的实例。您可以使用以下其中一种方法，从正在运行的实例中访问实例元数据：

- 实例元数据服务版本 1（IMDSv1）– 一种请求/响应方法
- 实例元数据服务版本 2（IMDSv2）– 一种面向会话的方法

虽然 Amazon EC2 同时支持 IMDSv1 和 IMDSv2，但 Amazon EMR 在 Amazon EMR 5.23.1、5.27.1、5.32 或更高版本以及 6.2 或更高版本中支持 IMDSv2。在这些版本中，Amazon EMR 组件对所有 IMDS 调用都使用 IMDSv2。对于应用程序代码中的 IMDS 调用，您可以同时使用 IMDSv1 和 IMDSv2，或者将 IMDS 配置为仅使用 IMDSv2，以提高安全性。在指定必须使用 IMDSv2 时，IMDSv1 不再起作用。

有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的配置实例元数据服务。

> **Note**
>
> 在早期的 Amazon EMR 5.x 或 6.x 版本中，关闭 IMDSv1 会导致集群启动失败，因为 Amazon EMR 组件对所有 IMDS 调用都使用 IMDSv1。关闭 IMDSv1 时，请确保任何使用 IMDSv1 的自定义软件都已更新为 IMDSv2。

指定实例元数据服务配置，使用 Amazon CLI。

以下是一个示例 JSON 代码段，用于在安全配置中指定 Amazon EC2 实例元数据服务（IMDS）。

```
{
  "InstanceMetadataServiceConfiguration" : {
      "MinimumInstanceMetadataServiceVersion": integer,
      "HttpPutResponseHopLimit": integer
   }
}
```

参数	描述
`"InstanceMetadataServiceConfiguration":`	必需。
`"MinimumInstanceMetadataServiceVersion":`	必需。指定 1 或 2。1 的一个值允许 IMDSv1 和 IMDSv2。2 的一个值仅允许 IMDSv2。
`"HttpPutResponseHopLimit":`	必需。实例元数据请求的所需 HTTP PUT 响应跃点限制。该数字越大，实例元数据请求传播得越远。默认值：1。您可以指定 1 到 64 之间的整数。

使用控制台指定实例元数据服务配置

当您从 Amazon EMR 控制台启动集群时，您可以为集群配置 IMDS 的使用。
Amazon EMR 控制台中的 IMDS 安全配置控件

要使用控制台配置 IMDS 的使用，请执行以下操作：

1. 在创建新的安全配置时，Security configurations (安全配置) 页面上，在 EC2 Instance Metadata Service (EC2 实例元数据服务) 设置下选择 Configure EC2 Instance metadata service (配置 EC2 实例元数据服务)。此配置仅在 Amazon EMR 5.23.1、5.27.1、5.32 或更高版本以及 6.2 或更高版本中受支持。
2. 对于 Minimum Instance Metadata Service Version (最低实例元数据服务版本) 选项，请选择以下任一项：
   - Turn off IMDSv1 and only allow IMDSv2 (关闭 IMDSv1 并且仅允许使用 IMDSv2)（如果您只想在此集群上允许 IMDSv2）。请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的过渡到使用实例元数据服务版本 2。
   - 如果您希望在此集群上允许 IMDSv1 和面向会话的 IMDSv2，则允许集群上同时使用 IMDSv1 和 IMDSv2。
3. 对于 IMDSv2，您还可以通过将 HTTP put response hop limit (HTTP PUT 响应跃点限制) 设置为 1 和 64 之间的整数来为元数据令牌设置允许的网络跃点数。

有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的配置实例元数据服务。

请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的配置实例详细信息和配置实例元数据服务。

为集群指定安全配置

创建集群时，您可以通过指定安全配置来指定加密设置。您可以使用 Amazon Web Services Management Console或 Amazon CLI。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台指定安全配置

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。

2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Security configuration and permissions（安全配置和权限）下，找到 Security configuration（安全配置）字段。选择下拉菜单或选择 Browse（浏览）以选择您之前创建的安全配置的名称。或者，选择 Create security configuration（创建安全配置）以创建可用于集群的配置。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台指定安全配置

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/emr。
2. 依次选择 Create cluster (创建集群)、Go to advanced options (转到高级选项)。
3. 在 Step 1: Software and Steps (步骤 1: 软件和步骤) 屏幕上，从 Release (版本) 列表中选择 emr-4.8.0 或更高版本。选择您需要的设置并选择 Next (下一步)。
4. 在 Step 2: Hardware (步骤 2: 硬件) 屏幕上，选择您需要的设置并选择 Next (下一步)。在 Step 3: General Cluster Settings (步骤 3: 常规集群设置) 中执行相同的操作。
5. 在 Step 4: Security (步骤 4: 安全) 屏幕上，在 Encryption Options (加密选项) 下为 Security configuration (安全配置) 选择一个值。
6. 根据需要配置其它安全选项，然后选择 Create cluster (创建集群)。

CLI

使用 Amazon CLI 指定安全配置

- 使用 `aws emr create-cluster` 以通过 `--security-configuration` *`MySecConfig`* 选择应用安全配置（其中 *`MySecConfig`* 是该安全配置的名称），如下例所示。您指定的 `--release-label` 必须是 4.8.0 或更高版本，`--instance-type` 可以是任意可用类型。

```
aws emr create-cluster --instance-type m5.xlarge --release-label emr-5.0.0 --
security-configuration mySecConfig
```

Amazon EMR 中的数据保护

Amazon责任共担模式应用于 Amazon EMR 中的数据保护。如该模式中所述，Amazon负责保护运行所有Amazon云的全球基础设施。您负责维护对托管在此基础设施上的内容的控制。此内容包括您所使用的Amazon的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅Amazon安全性博客上的 Amazon shared responsibility model and GDPR 一文。

出于数据保护目的，我们建议您保护 Amazon 账户凭证，并使用 Amazon Identity and Access Management 设置各个账户。这仅向每个用户授予履行其工作职责所需的权限。我们还建议您通过以下方式保护您的数据：

- 对每个账户使用 multi-factor authentication（MFA）。
- 使用 TLS 与 Amazon 资源进行通信。我们要求使用 TLS 1.2。
- 使用 Amazon CloudTrail 设置 API 和用户活动日志记录。
- 使用 Amazon 加密解决方案以及 Amazon 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Simple Storage Service（Amazon S3）中的个人数据。
- 如果在通过命令行界面或 API 访问 Amazon 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 终端节点的更多信息，请参阅美国联邦信息处理标准 (FIPS) 第 140-2 版。

我们强烈建议您切勿将敏感的可识别信息（例如您客户的账号）放入自由格式字段（例如 Name (名称) 字段）。这包括使用控制台、API、Amazon CLI 或 Amazon SDK 处理 Amazon EMR 或其它Amazon服务时。您输入到 Amazon EMR 或其它服务中的任何数据都可能被选取以包含在诊断日志中。当您向外部服务器提供 URL 时，请勿在 URL 中包含凭证信息来验证您对该服务器的请求。

加密静态数据和传输中的数据

数据加密有助于防止未经授权的用户在集群和关联的数据存储系统中读取数据。这包括保存到持久性媒体的数据（称为静态数据）和在网络中传输时可能被拦截的数据（称为传输中的数据）。

从 Amazon EMR 版本 4.8.0 开始，您可以使用 Amazon EMR 安全配置更轻松地为集群配置数据加密设置。安全配置提供用于为 Amazon S3 上 Amazon Elastic Block Store（Amazon EBS）卷和 EMRFS 中的传输中数据和静态数据增强安全性的设置。

（可选）从 Amazon EMR 版本 4.1.0 和更高版本开始，您可以选择在 HDFS 中配置透明加密，此操作不使用安全配置进行配置。有关更多信息，请参阅《Amazon EMR 版本指南》中的 Amazon EMR 上的 HDFS 中的透明加密。

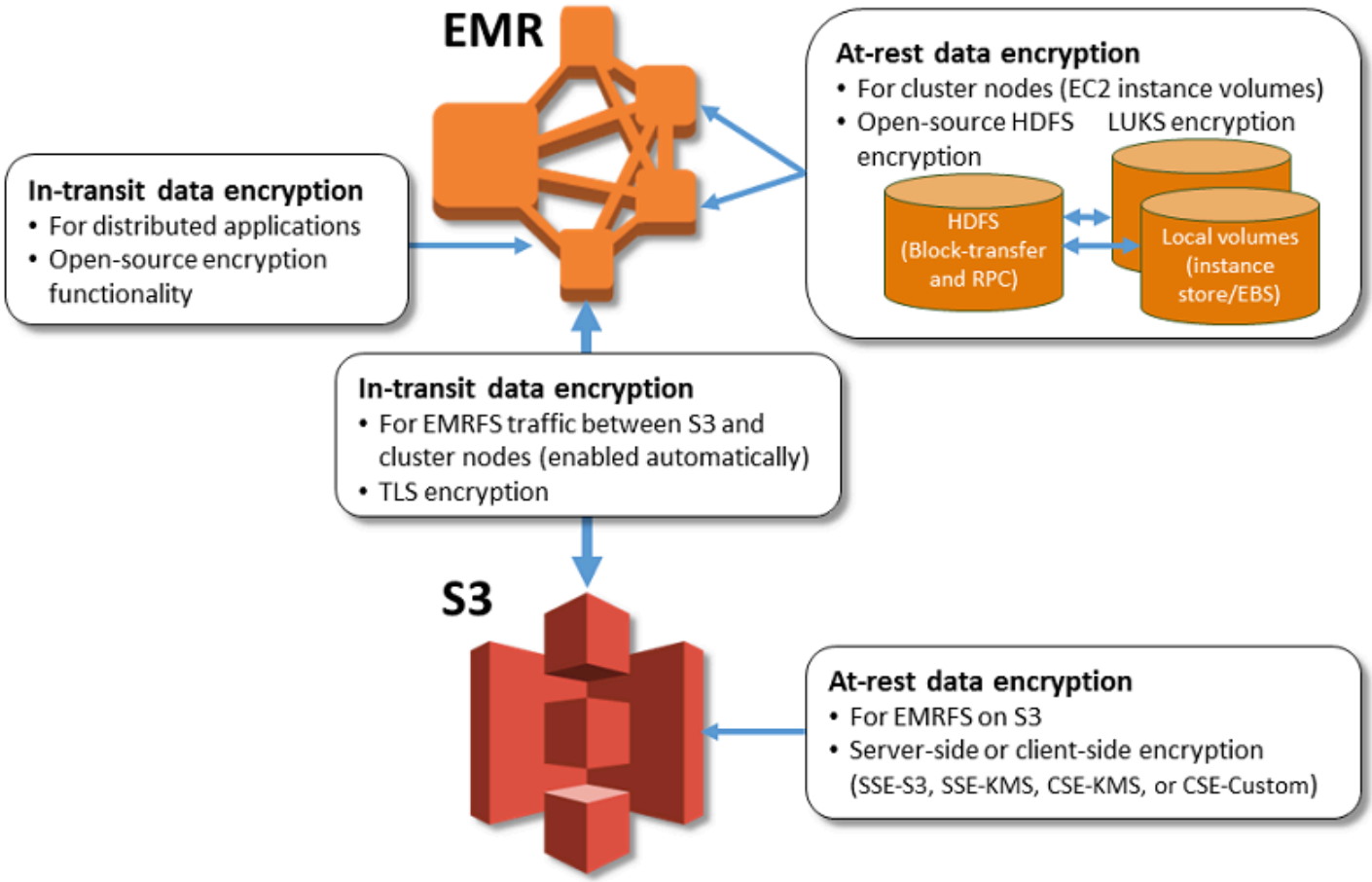
主题

- 加密选项
- 为数据加密创建密钥和凭证

加密选项

使用 Amazon EMR 版本 4.8.0 和更高版本，您可以使用安全配置来指定用于对静态数据和/或传输中的数据进行加密的设置。当启用静态数据加密时，您可以选择加密 Amazon S3 中的 EMRFS 数据和/或本地磁盘中的数据。您创建的每个安全配置都存储在 Amazon EMR 中而不是集群配置中，因此无论何时创建集群，您都可以轻松地重复使用某个配置来指定数据加密设置。有关更多信息，请参阅创建安全配置。

下图显示了安全配置可用的不同数据加密选项。



以下加密选项也可用，并且不是使用安全配置来配置的：

- （可选）使用 Amazon EMR 版本 4.1.0 和更高版本，您可以选择在 HDFS 中配置透明加密。有关更多信息，请参阅《Amazon EMR 版本指南》中的 Amazon EMR 上的 HDFS 中的透明加密。
- 如果您使用的是不支持安全配置的 Amazon EMR 版本，则可以在 Amazon S3 中为 EMRFS 数据手动配置加密。有关更多信息，请参阅使用 EMRFS 属性指定 Amazon S3 加密。
- 如果您使用 5.24.0 之前的 Amazon EMR 版本，则只有在使用自定义 AMI 时，才支持加密的 EBS 根设备卷。有关更多信息，请参阅《Amazon EMR 管理指南》中的使用加密的 Amazon EBS 根设备卷创建自定义 AMI。

> **Note**
>
> 从 Amazon EMR 版本 5.24.0 开始，您指定 Amazon KMS 作为密钥提供程序时，可以使用安全配置选项加密 EBS 根设备和存储卷。有关更多信息，请参阅本地磁盘加密。

数据加密需要密钥和凭证。安全配置可让您从多个选项中灵活选择，这些选项包括 Amazon Key Management Service 托管的密钥、Amazon S3 托管的密钥以及来自您提供的自定义提供程序的密钥和凭证。当使用 Amazon KMS 作为您的密钥提供程序时，您需支付加密密钥的存储和使用费用。有关更多信息，请参阅Amazon KMS定价。

在指定加密选项前，要确定所用的密钥和凭证管理系统，因此您可以先创建密钥和凭证或您在加密设置中指定的自定义提供程序。

Amazon S3 中 EMRFS 数据的静态加密

Amazon S3 加密适用于在 Amazon S3 中读取和写入的 EMR 文件系统（EMRFS）对象。在您启用静态加密时，请指定 Amazon S3 服务器端加密（SSE）或客户端加密（CSE）作为默认加密模式。（可选）您可以使用 Per bucket encryption overrides (每存储桶加密覆盖) 为单个存储桶指定不同的加密方法。无论是否启用了 Amazon S3 加密，传输层安全性 (TLS) 都会对 EMR 集群节点和 Amazon S3 之间正在传输的 EMRFS 对象进行加密。有关 Amazon S3 加密的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用加密保护数据。

> **Note**
>
> 使用 Amazon KMS 时，加密密钥的存储和使用将产生费用。有关更多信息，请参阅 Amazon KMS 定价。

Amazon S3 服务器端加密

设置 Amazon S3 服务器端加密时，Amazon S3 在向磁盘写入数据时会在对象级别加密数据，并在访问数据时对数据进行解密。有关 SSE 的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用服务器端加密保护数据。

在 Amazon EMR 中指定 SSE 时，可以在两个不同的密钥管理系统之间进行选择：

- SSE-S3 – Amazon S3 为您管理密钥。
- SSE-KMS - 您使用适用于 Amazon EMR 的策略设置一个 Amazon KMS key。有关 Amazon EMR 的密钥要求的更多信息，请参阅使用 Amazon KMS keys 进行加密。

客户提供密钥的 SSE（SSE-C）不能用于 Amazon EMR。

Amazon S3 客户端加密

对于 Amazon S3 客户端加密，Amazon S3 加密和解密过程在您的 EMR 集群上的 EMRFS 客户端中进行。在对象上载到 Amazon S3 之前对其进行加密，并在下载后对其进行解密。您指定的提供程序会提供客户端使用的加密密钥。客户端可以使用 Amazon KMS 提供的密钥（CSE-KMS）或提供客户端根密钥（CSE-C）的自定义 Java 类。CSE-KMS 和 CSE-C 之间的加密细节略有不同，具体取决于指定的提供程序以及正在解密或加密对象的元数据。有关这些区别的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用客户端加密保护数据。

> **Note**
>
> Amazon S3 CSE 仅确保与 Amazon S3 交换的 EMRFS 数据已加密；不确保集群实例卷上的所有数据都已加密。此外，由于 Hue 不使用 EMRFS，因此 Hue S3 文件浏览器写入 Amazon S3 的对象没有加密。

本地磁盘加密

在您使用 Amazon EMR 安全配置启用本地磁盘加密时，以下机制协同工作来加密本地磁盘。

开源 HDFS 加密

在分布式处理期间，HDFS 在集群实例之间交换数据。它还从实例存储卷以及附加到实例的 EBS 卷中读写数据。启用本地磁盘加密会激活以下开源 Hadoop 加密选项：

- 安全 Hadoop RPC 设置为 `Privacy`，它使用简单身份验证安全层 (SASL)。

- 对 HDFS 块数据传输的数据加密设置为 `true` 并配置为使用 AES 256 加密。

> **Note**
>
> 您也可以通过启用传输中加密激活其他 Apache Hadoop 加密。有关更多信息，请参阅传输中加密。这些加密设置不会激活 HDFS 透明加密，您可以手动配置透明加密。有关更多信息，请参阅《Amazon EMR 版本指南》中的 Amazon EMR 上的 HDFS 中的透明加密。

实例存储加密

对于使用基于 NVMe 的 SSD 作为实例存储卷的 EC2 实例类型，不论 Amazon EMR 加密设置如何，都将使用 NVMe 加密。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 NVMe SSD 卷。对于其它实例存储卷，在启用了本地磁盘加密时，Amazon EMR 使用 LUKS 加密实例存储卷，而不论使用 EBS 加密还是 LUKS 加密了 EBS 卷。

EBS 卷加密

如果您在默认为账户启用 EBS 卷的 Amazon EC2 加密的区域中创建集群，即使未启用本地磁盘加密，EBS 卷仍会加密。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的默认加密。在安全配置中启用了本地磁盘加密时，对于集群 EC2 实例，Amazon EMR 设置要优先于 Amazon EC2 encryption-by-default 设置。

利用以下选项，可以使用安全配置加密 EBS 卷：

- EBS 加密 – 从 Amazon EMR 版本 5.24.0 开始，您可以选择启用 EBS 加密。EBS 加密选项加密 EBS 根设备卷以及附加的存储卷。EBS 加密选项仅在您指定 Amazon Key Management Service 作为密钥提供程序时可用。我们建议使用 EBS 加密。
- LUKS 加密 – 如果您为 Amazon EBS 卷选择使用 LUKS 加密，LUKS 加密仅应用于附加的存储卷，而不应用于根设备卷。有关 LUKS 加密的更多信息，请参阅 LUKS 磁盘规格。

  对于密钥提供程序，您可以使用适用于 Amazon EMR 的策略设置一个 Amazon KMS key，也可以使用提供加密构件的自定义 Java 类。使用 Amazon KMS 时，加密密钥的存储和使用将产生费用。有关更多信息，请参阅Amazon KMS定价。

> **Note**
>
> 要检查您的集群上是否启用了 EBS 加密，建议您使用 `DescribeVolumes` API 调用。有关更多信息，请参阅 DescribeVolumes。在集群上运行 `lsblk` 只会检查 LUKS 加密的状态，而不检查 EBS 加密。

传输中加密

传输中加密会启用几种加密机制。这些是开源功能，是特定于应用程序的，可能因 Amazon EMR 版本而异。使用 Apache 应程序用配置可以启用以下特定于应用程序的加密功能。有关更多信息，请参阅配置应用程序。

Hadoop

- Hadoop MapReduce Encrypted Shuffle 使用 TLS。
- 安全 Hadoop RPC 设置为“Privacy”并使用 SASL（当静态加密启用时在 Amazon EMR 中激活）。
- 对 HDFS 块数据传输的数据加密使用 AES 256（当安全配置中启用了静态加密时在 Amazon EMR 中激活）。
- 有关更多信息，请参阅 Apache Hadoop 文档中的 Hadoop in secure mode（安全模式下的 Hadoop）。

HBase

- 如果启用了 Kerberos，`hbase.rpc.protection` 属性会设置为 `privacy` 用于加密通信。
- 有关更多信息，请参阅 Apache HBase 文档中的用于安全操作的客户端配置。
- 有关 Kerberos 与 Amazon EMR 的更多信息，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。

Hive

- JDBC/ODBC 客户端与 HiveServer2（HS2）的通信使用 Amazon EMR 发行版 6.9.0 及更高版本中的 SSL 配置进行加密。
- 有关更多信息，请参阅 Apache Hive 文档的 SSL encryption（SSL 加密）部分。

Spark

- 在 Amazon EMR 版本 5.9.0 及更高版本中，使用 AES-256 密码对 Spark 组件（例如，数据块传输服务和外部随机服务）之间的内部 RPC 通信进行加密。在早期版本中，使用将 DIGEST-MD5 作为密码的 SASL 对内部 RPC 通信进行加密。
- 使用 Spark 的 SSL 配置对与用户界面（如 Spark History Server 和支持 HTTPS 的文件服务器）的 HTTP 协议通信进行加密。有关更多信息，请参阅 Spark 文档中的 SSL 配置。
- 有关更多信息，请参阅 Apache Spark 文档中的 Spark security settings（Spark 安全设置）。

Tez

- Tez shuffle handler 使用 TLS（`tez.runtime.ssl.enable`）。

Presto

- Presto 节点之间的内部通信使用 SSL/TLS (仅限 Amazon EMR 版本 5.6.0 及更高版本)。

您可以通过两种方式指定用于传输中加密的加密构件：一种是提供上载到 Amazon S3 的凭证的压缩文件，另一种是引用提供加密构件的自定义 Java 类。有关更多信息，请参阅为通过 Amazon EMR 加密来加密传输中的数据提供凭证。

为数据加密创建密钥和凭证

在使用安全配置指定加密选项前，确定要用于密钥和加密构件的提供程序。例如，您可以使用 Amazon KMS 或您创建的自定义提供程序。接下来，创建密钥或密钥提供程序，如本节中所述。

提供通过 Amazon EMR 加密静态数据的密钥

在 Amazon EMR 中，您可以将 Amazon Key Management Service（Amazon KMS）或自定义密钥提供程序用于静态数据加密。使用 Amazon KMS 时，加密密钥的存储和使用将产生费用。有关更多信息，请参阅Amazon KMS定价。

本主题介绍可与 Amazon EMR 结合使用的 KMS 密钥的密钥策略详细信息，以及编写适用于 Amazon S3 加密的自定义密钥提供程序类的准则和代码示例。有关创建密钥的更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的创建密钥。

使用 Amazon KMS keys 加密

Amazon KMS 加密密钥，必须在与您的 Amazon EMR 集群实例和与 EMRFS 一起使用的 Amazon S3 存储桶相同的区域创建。如果指定的密钥没有位于用于配置集群的账户中，则必须使用它的 ARN 指定密钥。

Amazon EC2 实例配置文件的角色必须具有使用您指定的 KMS 密钥的权限。Amazon EMR 中实例配置文件的默认角色是 `EMR_EC2_DefaultRole`。如果您对实例配置文件使用不同的角色，或者对 Amazon S3 的 EMRFS 请求使用 IAM 角色，请确保根据需要将每个角色添加为密钥用户。这会为该角色授予使用该 KMS 密钥的权限。有关更多信息，请参阅《Amazon Key Management Service开发人员指南》和为向 Amazon S3 发出的 EMRFS 请求配置 IAM 角色中的使用密钥策略。

您可以使用 Amazon Web Services Management Console 将实例配置文件或 EC2 实例配置文件添加到指定 KMS 密钥的密钥用户列表中，也可以使用 Amazon CLI 或 Amazon SDK 来附加适当的密钥策略。

请注意 Amazon EMR 仅支持对称 KMS 密钥。不能使用非对称 KMS 密钥加密 Amazon EMR 集群中的静态数据。要获取确定 KMS 密钥是对称还是非对称的帮助，请参阅识别对称密钥和非对称密钥。

以下步骤介绍了如何使用 Amazon Web Services Management Console 将 Amazon EMR 实例配置文件 `EMR_EC2_DefaultRole` 作为密钥用户添加。它假定您已创建一个 KMS 密钥。要创建新的 KMS 密钥，请参阅《Amazon Key Management Service 开发人员指南》中的创建密钥。

将 Amazon EMR 的 EC2 实例配置文件添加到加密密钥用户列表中

1. 登录到 Amazon Web Services Management Console，然后通过以下网址打开 Amazon Key Management Service (Amazon KMS) 控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 选择要修改的 KMS 密钥的别名。
4. 在密钥详细信息页面的 Key Users (密钥用户) 下，选择 Add (添加)。
5. 在 Add key users (添加密钥用户) 对话框中，选择适当的角色。默认角色的名称为 `EMR_EC2_DefaultRole`。
6. 选择 Add（添加）。

通过为 KMS 密钥提供额外的权限来启用 EBS 加密

从 Amazon EMR 版本 5.24.0 开始，您可以使用安全配置选项加密 EBS 根设备和存储卷。要启用此选项，您必须指定 Amazon KMS 作为密钥提供程序。此外，您必须向 EMR 服务角色 `EMR_DefaultRole` 授予使用您指定的 Amazon KMS key 的权限。

您可以使用 Amazon Web Services Management Console 将 EMR 服务角色添加到指定 KMS 密钥的密钥用户列表中，也可以使用 Amazon CLI 或 Amazon SDK 来附加适当的密钥策略。

以下步骤介绍了如何使用Amazon Web Services Management Console添加默认 EMR 实例服务角色 `EMR_DefaultRole`，并将其作为密钥用户。它假定您已创建一个 KMS 密钥。要创建新的 KMS 密钥，请参阅《Amazon Key Management Service 开发人员指南》中的创建密钥。

将 EMR 服务角色添加到加密密钥用户列表中

1. 登录到 Amazon Web Services Management Console，然后通过以下网址打开 Amazon Key Management Service (Amazon KMS) 控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在左侧边栏中选择 Customer managed keys (客户托管密钥)。
4. 选择要修改的 KMS 密钥的别名。
5. 在密钥详细信息页面的 Key Users (密钥用户) 下，选择 Add (添加)。
6. 在 Add key users (添加密钥用户) 对话框中，选择适当的角色。默认 EMR 服务角色的名称为 `EMR_DefaultRole`。
7. 选择 Add（添加）。

创建自定义密钥提供程序

当使用安全配置时，您必须为本地磁盘加密和 Amazon S3 加密指定不同的提供程序类名。

创建自定义密钥提供程序时，应用程序应实施 EncryptionMaterialsProvider 接口，它在 Amazon SDK for Java 1.11.0 版及更高版本中可用。为提供加密材料，该实施可以使用任何策略。例如，您可以选择提供静态加密材料，也可以选择与更复杂的密钥管理系统集成。

用于自定义加密材料的加密算法必须是 AES/GCM/NoPadding。

EncryptionMaterialsProvider 类通过加密上下文获取加密材料。Amazon EMR 在运行时填充加密上下文信息，以帮助调用者确定要返回的正确加密材料。

Example 示例：通过 EMRFS 使用自定义密钥提供程序对 Amazon S3 进行加密

当 Amazon EMR 从 EncryptionMaterialsProvider 类中获取加密材料以执行加密时，EMRFS 可以选择使用两个字段填充 materialsDescription 参数：对象的 Amazon S3 URI 和集群的 JobFlowId，这两个字段可供 EncryptionMaterialsProvider 类使用来选择性地返回加密材料。

例如，提供程序可能会为不同的 Amazon S3 URI 前缀返回不同的密钥。它是最终与 Amazon S3 对象一起存储的返回加密材料的描述，而不是 EMRFS 生成并传递给提供程序的 materialsDescription 值。解密 Amazon S3 对象时，加密材料描述将传递给 EncryptionMaterialsProvider 类，以便它可以再次选择性地返回匹配的密钥以解密对象。

以下是 EncryptionMaterialsProvider 参考实施。另一个自定义提供程序 EMRFSRSAEncryptionMaterialsProvider 可从 GitHub 获取。

```
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.EncryptionMaterialsProvider;
import com.amazonaws.services.s3.model.KMSEncryptionMaterials;
import org.apache.hadoop.conf.Configurable;
import org.apache.hadoop.conf.Configuration;

import java.util.Map;

/**
 * Provides KMSEncryptionMaterials according to Configuration
 */
public class MyEncryptionMaterialsProviders implements EncryptionMaterialsProvider,
 Configurable{
  private Configuration conf;
  private String kmsKeyId;
  private EncryptionMaterials encryptionMaterials;

  private void init() {
    this.kmsKeyId = conf.get("my.kms.key.id");
    this.encryptionMaterials = new KMSEncryptionMaterials(kmsKeyId);
  }

  @Override
  public void setConf(Configuration conf) {
    this.conf = conf;
    init();
  }

  @Override
  public Configuration getConf() {
    return this.conf;
  }

  @Override
  public void refresh() {
```

```
  }

  @Override
  public EncryptionMaterials getEncryptionMaterials(Map<String, String>
 materialsDescription) {
    return this.encryptionMaterials;
  }

  @Override
  public EncryptionMaterials getEncryptionMaterials() {
    return this.encryptionMaterials;
  }
}
```

为通过 Amazon EMR 加密来加密传输中的数据提供凭证

使用 Amazon EMR 发行版 4.8.0 或更高版本，您有两个选项可指定使用安全配置对传输中的数据进行加密的构件：

- nin 可以手动创建 PEM 凭证，将它们包含一个 zip 文件中，然后在 Amazon S3 中引用该 .zip 文件。
- 可以将一个自定义凭证提供程序作为 Java 类实现。在 Amazon S3 中指定应用程序的 JAR 文件，然后提供与应用程序声明中相同的完整提供程序类名。该类必需实现 TLSArtifactsProvider 接口，该接口是从Amazon SDK for Java版本 1.11.0 开始提供的。

Amazon EMR 自动将构件下载到集群中的每个节点，然后使用它们实现开源传输中加密功能。有关可用选项的更多信息，请参阅传输中加密。

使用 PEM 凭证

在为传输中加密指定 .zip 文件时，安全配置要求 .zip 文件中的 PEM 文件的命名方式与下面完全一致：

传输中加密凭证

文件名	必需/可选	详细信息
privateKey.pem	必填	私有密钥
certificateChain.pem	必填	凭证链

文件名	必需/可选	详细信息
trustedCertificates.pem	可选	如果 Java 默认可信根凭证颁发机构 (CA) 或可链接到 Java 默认可信根 CA 的中间 CA 没有对提供的凭证签名，则此选项是必需的。可在 `jre/lib/security/cacerts` 中找到 Java 默认可信根 CA。

您可能需要将私有密钥 PEM 文件配置为支持访问您的集群实例所在的 Amazon VPC 域的通配符凭证。例如，如果您的集群位于 us-east-1（弗吉尼亚北部），则可以选择通过在凭证主题定义中指定 `CN=*.ec2.internal` 在凭证配置中指定一个通用名称以允许访问集群。如果您的集群位于 us-west-2（俄勒冈），则可以指定 `CN=*.us-west-2.compute.internal`。

如果加密构件中提供的 PEM 文件在域的 CN 中没有通配符，则必须将 `hadoop.ssl.hostname.verifier` 的值改为 `ALLOW_ALL`。将配置提交到集群时使用 `core-site` 分类，或者通过将此值添加到 `core-site.xml` 文件中来完成此操作。必须进行此更改，因为默认主机名验证程序将不接受没有通配符的主机名，从而导致错误。有关 Amazon VPC 中的 EMR 集群配置的更多信息，请参阅 配置联网。

以下示例说明了如何使用 OpenSSL 生成一个带 1024 位 RSA 私有密钥的自签名 X.509 凭证。该密钥允许访问发布者在 `us-west-2`（俄勒冈）区域中的 Amazon EMR 集群实例，这是用 *`*.us-west-2.compute.internal`* 域名作为通用名称来表示的。

指定其它可选项（如国家/地区 (C)、州/省 (S)、区域设置 (L) 等）。由于生成了自签名凭证，因此，该示例中的第二个命令会将 `certificateChain.pem` 文件复制到 `trustedCertificates.pem` 文件。第三个命令使用 `zip` 创建包含这些凭证的 `my-certs.zip` 文件。

> **Important**
>
> 此示例仅为概念验证演示。不建议使用自签名凭证，可能存在潜在安全风险。对于生产系统，应使用可信凭证颁发机构 (CA) 颁发凭证。

```
$ openssl req -x509 -newkey rsa:1024 -keyout privateKey.pem -out certificateChain.pem
 -days 365 -nodes -subj '/C=US/ST=Washington/L=Seattle/O=MyOrg/OU=MyDept/CN=*.us-
west-2.compute.internal'
$ cp certificateChain.pem trustedCertificates.pem
$ zip -r -X my-certs.zip certificateChain.pem privateKey.pem trustedCertificates.pem
```

适用于 Amazon EMR 的 Amazon Identity and Access Management

Amazon Identity and Access Management (IAM) 是一项 Amazon Web Service，可以帮助管理员安全地控制对 Amazon 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和授权（具有权限）使用 Amazon EMR 资源。IAM 是一项无需额外费用即可使用的 Amazon Web Service。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- Amazon EMR 如何与 IAM 配合使用
- Amazon EMR 步骤的运行时角色
- 为 Amazon EMR 配置 IAM 服务角色对Amazon服务和资源的权限
- Amazon EMR 基于身份的策略示例

受众

如何使用 Amazon Identity and Access Management（IAM）因您在 Amazon EMR 中执行的操作而异。

服务用户 – 如果您使用 Amazon EMR 服务来完成任务，则您的管理员会为您提供所需的凭证和权限。随着您使用更多 Amazon EMR 功能来完成工作，您可能需要额外权限。了解如何管理访问权限可帮助您向管理员请求适合的权限。如果您无法访问 Amazon EMR 中的功能，请参阅 Amazon EMR 身份和访问问题排查。

服务管理员 – 如果您在公司负责管理 Amazon EMR 资源，您可能对 Amazon EMR 具有完全访问权限。您有责任确定您的服务用户应访问哪些 Amazon EMR 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 Amazon EMR 搭配使用的更多信息，请参阅Amazon EMR 如何与 IAM 配合使用。

IAM 管理员 – 如果您是 IAM 管理员，您可能需要了解如何编写策略以管理对 Amazon EMR 的访问的详细信息。要查看您可在 IAM 中使用的 Amazon EMR 基于身份的策略示例，请参阅Amazon EMR 基于身份的策略示例。

使用身份进行身份验证

身份验证是您使用身份凭证登录 Amazon 的方法。您必须作为 Amazon Web Services 账户根用户、IAM 用户或通过代入 IAM 角色进行身份验证（登录到 Amazon）。

如果您以编程方式访问 Amazon，则 Amazon 将提供软件开发工具包（SDK）和命令行界面（CLI），以便使用您的凭证以加密方式签署您的请求。如果您不使用 Amazon 工具，则必须自行对请求签名。有关使用推荐的方法自行签署请求的更多信息，请参阅《IAM 用户指南》中的签署 Amazon API 请求。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，Amazon 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《IAM 用户指南》中的 在 Amazon 中使用多重身份验证（MFA）。

Amazon Web Services 账户根用户

当您创建 Amazon Web Services 账户时，最初使用的是一个对账户中所有 Amazon Web Services和资源拥有完全访问权限的登录身份。此身份称为 Amazon Web Services 账户根用户，使用您创建账户时所用的电子邮件地址和密码登录，即可获得该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）结合使用联合身份验证和身份提供程序，以使用临时凭证来访问 Amazon Web Services。

联合身份是来自企业用户目录、Web 身份提供程序、Amazon Directory Service、的用户，或任何使用通过身份源提供的凭证来访问 Amazon Web Services 的用户。当联合身份访问 Amazon Web Services 账户时，他们代入角色，而角色提供临时凭证。

IAM 用户和组

IAM 用户是Amazon Web Services 账户内对某个人员或应用程序具有特定权限的一个身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，我们建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM 用户指南中的何时创建 IAM 用户（而不是角色）。

IAM 角色

IAM 角色是Amazon Web Services 账户中具有特定权限的身份。它类似于 IAM 用户，但与特定人员不关联。您可以通过切换角色，在 Amazon Web Services Management Console 中暂时代入 IAM 角色。您可以调用 Amazon CLI 或 Amazon API 操作或使用自定义 URL 以担任角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- Federated user access（联合用户访问）– 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的 为第三方身份提供商创建角色。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户访问 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些 Amazon Web Services，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。
- 跨服务访问 – 某些 Amazon Web Services使用其它 Amazon Web Services中的特征。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service（Amazon S3）中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 主体权限 – 当您使用 IAM 用户或角色在 Amazon 中执行操作时，您将被视为主体。策略向主体授予权限。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中触发另一个操作。在这种情况下，您必须具有执行这两个操作的权限。要查看某个操作是否要依赖策略中的其他相关操作，请参阅《服务授权参考》中的 Amazon EMR 的操作、资源和条件键。
  - 服务角色 – 服务角色是服务代表您在您的账户中执行操作而担任的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的 创建向 Amazon Web Service 委派权限的角色。

- 服务相关角色 – 服务相关角色是与 Amazon Web Service关联的一种服务角色。服务可以代入代表您执行操作的角色。服务相关角色显示在您的 Amazon Web Services 账户 中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 – 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 Amazon CLI 或 Amazon API 请求的应用程序的临时凭证。这优先于在 EC2 实例中存储访问密钥。要将 Amazon 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅 IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您将创建策略并将其附加到 Amazon 身份或资源，以控制 Amazon 中的访问。策略是 Amazon 中的对象；在与身份或资源相关联时，策略定义它们的权限。在主体（用户、根用户或角色会话）发出请求时，Amazon 将评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略在 Amazon 中存储为 JSON 文档。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南 中的 JSON 策略概览。

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么条件 下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM policy。然后，管理员可以向角色添加 IAM policy，并且用户可以代入角色。

IAM policy 定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。具有该策略的用户可以从 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的 创建 IAM policy。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管式策略是可以附加到Amazon Web Services 账户中的多个用户、组和角色的独立策略。托管式策

略包括 Amazon 托管式策略和客户托管式策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅 IAM 用户指南 中的在托管式策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合身份用户或 Amazon Web Services。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用来自 IAM 的 Amazon 托管式策略。

访问控制列表 (ACL)

访问控制列表 (ACL) 控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Simple Storage Service（Amazon S3）、Amazon WAF 和 Amazon VPC 是支持 ACL 的服务示例。要了解有关 ACL 的更多信息，请参阅 Amazon Simple Storage Service 开发人员指南 中的访问控制列表 (ACL) 概览。

其它策略类型

Amazon 支持额外的、不太常用的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 – 权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅 IAM 用户指南中的 IAM 实体的权限边界。
- 服务控制策略 (SCP) – SCP 是 JSON 策略，指定了组织或组织单位 (OU) 在 Amazon Organizations 中的最大权限。Amazon Organizations 服务可以分组和集中管理您的企业拥有的多个 Amazon Web Services 账户 账户。如果在组织内启用了所有特征，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中实体（包括每个 Amazon Web Services 账户根用户）的权限。有关 Organizations 和 SCP 的更多信息，请参阅 Amazon Organizations 用户指南中的 SCP 的工作原理。
- 会话策略 – 会话策略是当您以编程方式为角色或联合身份用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资

源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解 Amazon 如何确定在涉及多种策略类型时是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

Amazon EMR 如何与 IAM 配合使用

在使用 IAM 管理对 Amazon EMR 的访问权限之前，您应该了解哪些 IAM 功能可用于 Amazon EMR。

将 IAM 功能与 Amazon EMR 一起使用

IAM 特征	Amazon EMR 支持
基于身份的策略	是
基于资源的策略	是
策略操作	是
策略资源	是
策略条件键	是
ACL	否
ABAC（策略中的标签）	是
临时凭证	是
委托人权限	是
服务角色	否
服务相关角色	是

要大致了解 Amazon EMR 和其他 Amazon 服务如何与大多数 IAM 功能一起使用，请参阅《IAM 用户指南》中的与 IAM 一起使用的 Amazon 服务。

Amazon EMR 基于身份的策略

支持基于身份的策略	是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM policy。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Amazon EMR 基于身份的策略示例

要查看 Amazon EMR 基于身份的策略的示例，请参阅Amazon EMR 基于身份的策略示例。

Amazon EMR 基于资源的策略

支持基于资源的策略。	是

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合身份用户或 Amazon Web Services。

要启用跨账户存取，您可以将整个账户或其它账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当主体和资源处于不同的 Amazon Web Services 账户中时，则信任账户中的 IAM 管理员还必须授予主体实体（用户或角色）对资源的访问权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

Amazon EMR 的策略操作

支持策略操作	是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么 条件 下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 Amazon API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行相关操作的权限。

要查看 Amazon EMR 操作的列表，请参阅《服务授权参考》中的 Amazon EMR 的操作、资源和条件键。

Amazon EMR 中的策略操作在操作前面使用以下前缀：

```
EMR
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "EMR:action1",
      "EMR:action2"
         ]
```

要查看 Amazon EMR 基于身份的策略的示例，请参阅Amazon EMR 基于身份的策略示例。

Amazon EMR 的策略资源

支持策略资源	是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么条件 下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 Amazon EMR 的资源类型及其 ARN 的列表，请参阅《服务授权参考》中的由 Amazon EMR 定义的资源。要了解可以为每个资源的 ARN 指定哪些操作，请参阅 Amazon EMR 的操作、资源和条件键。

要查看 Amazon EMR 基于身份的策略的示例，请参阅Amazon EMR 基于身份的策略示例。

Amazon EMR 的策略条件键

支持特定于服务的策略条件键	是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么内容。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 `Condition` 元素（或 `Condition` 块）中，可以指定语句生效的条件。`Condition` 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 `Condition` 元素，或在单个 `Condition` 元素中指定多个键，则 Amazon 使用逻辑 `AND` 运算评估它们。如果您为单个条件键指定多个值，则 Amazon 使用逻辑 `OR` 运算来评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM policy 元素：变量和标签。

Amazon支持全局条件键和特定于服务的条件键。要查看所有 Amazon 全局条件键，请参阅《IAM 用户指南》中的 Amazon 全局条件上下文键。

要查看 Amazon EMR 条件键的列表，并了解您可以对哪些操作和资源使用条件键，请参阅《服务授权参考》中的 Amazon EMR 的操作、资源和条件键。

要查看 Amazon EMR 基于身份的策略的示例，请参阅Amazon EMR 基于身份的策略示例。

Amazon EMR 中的访问控制列表（ACL）

支持 ACL	否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

使用 Amazon EMR 基于属性的访问权限控制（ABAC）

支持 ABAC（策略中的标签）	是

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在 Amazon 中，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）以及Amazon资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为 Yes（是）。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为 Partial（部分）。

有关 ABAC 的更多信息,请参阅《IAM 用户指南》中的什么是 ABAC？。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问控制（ABAC）。

使用 Amazon EMR 的临时凭证

支持临时凭证	是

某些 Amazon Web Services在您使用临时凭证登录时无法正常工作。有关更多信息，包括 Amazon Web Services与临时凭证配合使用，请参阅 IAM 用户指南中的使用 IAM 的 Amazon Web Services。

如果您不使用用户名和密码而用其它方法登录到Amazon Web Services Management Console，则使用临时凭证。例如，当您使用贵公司的单点登录（SSO）链接访问Amazon时，该过程将自动创建临时凭证。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的 切换到角色（控制台）。

您可以使用 Amazon CLI 或者 Amazon API 创建临时凭证。之后，您可以使用这些临时凭证访问 Amazon。Amazon建议您动态生成临时凭证，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

Amazon EMR 的跨服务主体权限

支持委托人权限	是

当您使用 IAM 用户或角色在Amazon中执行操作时，您将被视为委托人。策略向主体授予权限。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中触发另一个操作。在这种情况下，您必须具有执行这两个操作的权限。要查看某个操作是否要依赖策略中的其他相关操作，请参阅《服务授权参考》中的 Amazon EMR 的操作、资源和条件键。

Amazon EMR 的服务角色

支持服务角色	否

Amazon EMR 的服务相关角色

支持服务相关角色	是

有关创建或管理服务相关角色的详细信息，请参阅使用 IAM 的Amazon服务。在表中查找服务相关角色列中包含 Yes 额表。选择 Yes 链接以查看该服务的服务相关角色文档。

将集群和笔记本标签与访问控制 IAM policy 结合使用

可以将基于标签的访问控制与基于身份的 IAM policy 结合使用，精细调整与 EMR Notebooks 和 EMR 集群关联的 Amazon EMR 操作的权限。您可以使用 `Condition` 元素中的条件键（也称作 `Condition` 数据块），来允许仅在笔记本和/或集群具有特定标签或键-值组合时执行某些操作。您也可以限制 `CreateEditor` 操作（创建 EMR Notebooks）和 `RunJobFlow` 操作（创建集群），这样在创建资源时必须提交标签请求。

在 Amazon EMR 中，可用于 `Condition` 元素中的条件键仅应用于 `ClusterID` 或 `NotebookID` 为必需请求参数的 Amazon EMR API 操作。例如，ModifyInstanceGroups 操作不支持上下文密钥，因为 `ClusterID` 是可选参数。

您创建 EMR Notebooks 时，系统会应用一个默认标签，其中 `creatorUserId` 的键字符串会设置为创建笔记本的 IAM 用户 ID 的值。这在将笔记本的允许操作仅限定为创建者时非常有用。

以下条件键可用于 Amazon EMR 中：

- 使用 `elasticmapreduce:ResourceTag/TagKeyString` 条件上下文键可允许或拒绝用户对带有您指定的 *TagKeyString* 的标签的集群或笔记本执行操作。如果某个操作同时传递 `ClusterID` 和 `NotebookID`，则该条件同时适用于集群和笔记本。这意味着这两个资源都必须具有您指定的标签键字符串或键-值组合。您可以使用 `Resource` 元素来限制语句，以便它能根据需要仅适用于集群或笔记本。有关更多信息，请参阅Amazon EMR 基于身份的策略示例。
- 使用 `elasticmapreduce:RequestTag/TagKeyString` 条件上下文密钥要求带操作/API 调用的特定标签。例如，您可以将此条件上下文键与 `CreateEditor` 操作一起使用，以在创建笔记本时对其应用 *TagKeyString*。

示例

要查看 Amazon EMR 操作的列表，请参阅《IAM 用户指南》中的 Amazon EMR 定义的操作。

Amazon EMR 步骤的运行时角色

运行时角色是一个 Amazon Identity and Access Management（IAM）角色，您可以在向 Amazon EMR 集群提交作业或查询时指定该角色。您提交给 Amazon EMR 集群的作业或查询使用运行时角色访问 Amazon 资源，例如 Amazon S3 中的对象。您可以使用 Amazon EMR 为 Spark 和 Hive 作业指定运行时角色。

当您在 Amazon SageMaker 内连接到 Amazon EMR 集群时，以及在将 Amazon EMR Studio 工作区连接到 EMR 集群时，您也可以指定运行时系统角色。有关更多信息，请参阅 Connect to an Amazon EMR cluster from Studio 和 使用运行时系统角色运行 EMR Studio Workspace。

以前，Amazon EMR 集群运行 Amazon EMR 作业或查询，其权限基于附加到您用于启动集群的实例配置文件的 IAM policy。这意味着策略必须包含在 Amazon EMR 集群上所运行所有作业和查询的所有权限的联合。使用运行时角色，您现在可以单独管理每个作业或查询的访问控制，而不是共享集群的 Amazon EMR 实例配置文件。

在具有运行时角色的 Amazon EMR 集群上，您还可以将基于 Amazon Lake Formation 的访问控制应用于针对数据湖的 Spark、Hive 和 Presto 作业和查询。要了解有关如何与 Amazon Lake Formation 集成的更多信息，请参阅 将 Amazon EMR 与 Amazon Lake Formation 集成。

> **Note**
>
> 当您为 Amazon EMR 步骤指定运行时角色时，您提交的作业或查询只能访问附加到运行时角色的策略所允许的 Amazon 资源。这些作业和查询无法访问该集群 EC2 实例上的 Instance Metadata Service，也无法使用集群的 EC2 实例配置文件来访问任何 Amazon 资源。

使用运行时角色启动 Amazon EMR 集群的先决条件

主题

- 步骤 1：在 Amazon EMR 中设置安全配置
- 步骤 2：为 Amazon EMR 集群设置 EC2 实例配置文件
- 步骤 3：设置信任策略

步骤 1：在 Amazon EMR 中设置安全配置

使用以下 JSON 结构在 Amazon Command Line Interface (Amazon CLI) 上创建安全配置，并将 `EnableApplicationScopedIAMRole` 设置为 `true`。有关安全配置的更多信息，请参阅 使用安全配置设置集群安全性。

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true
        }
    }
}
```

建议您始终在安全配置中启用传输中加密选项，以便对通过互联网传输的数据进行加密，而不是以纯文本形式传输。如果您不想从 SageMaker Studio 或 EMR Studio 使用运行时系统角色连接到 Amazon EMR 集群，则可以跳过这些选项。要配置数据加密，请参阅配置数据加密。

或者，您可以使用 Amazon Web Services Management Console 创建具有自定义设置的安全配置。

步骤 2：为 Amazon EMR 集群设置 EC2 实例配置文件

Amazon EMR 集群使用 Amazon EC2 实例配置文件角色来带入运行时角色。要将运行时角色与 Amazon EMR 步骤配合使用，请将以下策略添加到计划用作实例配置文件角色的 IAM 角色。要向 IAM 角色添加策略或编辑现有的内联式或托管式策略，请参阅添加和删除 IAM 身份权限。

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowRuntimeRoleUsage",
            "Effect":"Allow",
            "Action":[
```

```
                "sts:AssumeRole",
                "sts:TagSession"
            ],
            "Resource":[
                <runtime-role-ARN>
            ]
        }
    ]
}
```

步骤 3：设置信任策略

对于您计划用作运行时角色的每个 IAM 角色，设置以下信任策略，将 EMR_EC2_DefaultRole 替换为您的实例配置文件角色。要修改 IAM 角色的信任策略，请参阅修改角色信任策略。

```
{
    "Sid":"AllowAssumeRole",
    "Effect":"Allow",
    "Principal":{
        "AWS":"arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
    },
    "Action":"sts:AssumeRole"
}
```

通过基于角色的访问控制启动 Amazon EMR 集群

设置配置后，您可以使用 步骤 1：在 Amazon EMR 中设置安全配置 安全配置来启动 Amazon EMR 集群。要将运行时角色与 Amazon EMR 步骤结合使用，请使用发行版本标签 emr-6.7.0 或者更高版本，然后选择 Hive 和/或 Spark 作为集群应用程序。要从 SageMaker Studio 进行连接，请使用发行版本 emr-6.9.0 或更高版本，然后选择 Livy、Spark、Hive 或 Presto 作为集群应用程序。有关如何启动集群的说明，请参阅 为集群指定安全配置。

使用 Amazon EMR 步骤提交 Spark 任务

下面示范了如何运行 Apache Spark 包含的 HdfsTest 示例。只有在提供的 Amazon EMR 运行时角色有权访问 S3_LOCATION 时，API 调用才会成功。

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
S3_LOCATION=<s3-path>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>
```

```
aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Spark Example", "ActionOnFailure": "CONTINUE","HadoopJarStep":
 { "Jar":"command-runner.jar","Args" : ["spark-example","HdfsTest",
 "$S3_LOCATION"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

> **Note**
>
> 我们建议您关闭对 Amazon EMR 集群的 SSH 访问权限，并且只允许 Amazon EMR `AddJobFlowSteps` API 访问集群。

使用 Amazon EMR 步骤提交 Hive 任务

以下示例使用带有 Amazon EMR 步骤的 Apache Hive 提交任务以运行 `QUERY_FILE.hql` 文件。仅当提供的运行时角色可以访问查询文件的 Amazon S3 路径时，此查询才会成功。

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>

aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Run hive query using command-runner.jar - simple
 select","ActionOnFailure":"CONTINUE","HadoopJarStep": { "Jar": "command-
runner.jar","Args" :["hive -
f","s3://DOC_EXAMPLE_BUCKET/QUERY_FILE.hql"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

从 SageMaker Studio 笔记本使用运行时角色连接到 Amazon EMR 集群

您可以从 SageMaker Studio 将 Amazon EMR 运行时角色应用于在 Amazon EMR 集群中运行的查询。为此，请执行以下步骤。

1. 按照 Launch Amazon SageMaker Studio（启动 Amazon SageMaker Studio）中的说明创建 SageMaker Studio。
2. 在 SageMaker Studio UI 中，启动具有支持的内核的笔记本。例如，启动具有 PySpark 内核的 SparkMagic 映像。
3. 在 SageMaker Studio 中选择 Amazon EMR 集群，然后选择 Connect（连接）。

4. 选择运行时角色，然后选择 Connect（连接）。

这将使用魔术命令创建一个 SageMaker 笔记本单元，以使用所选的 Amazon EMR 运行时角色连接到您的 Amazon EMR 集群。在笔记本单元中，您可以使用运行时角色和基于 Lake Formation 的访问控制输入和运行查询。有关更多详细示例，请参阅使用 Amazon SageMaker Studio 中的 Amazon Lake Formation 和 Amazon EMR 应用精细数据访问控制。

控制对 Amazon EMR 运行时角色的访问

您可以使用条件键 `elasticmapreduce:ExecutionRoleArn` 控制对运行时角色的访问。以下策略允许 IAM 主体使用名为 `Caller` 的 IAM 角色，或任何以字符串 `CallerTeamRole` 开头的 IAM 角色，作为运行时角色。

> **Important**
>
> 向调用者授予调用 `AddJobFlowSteps` 或 `GetClusterSessionCredentials` API 的访问权限时，您必须创建基于 `elasticmapreduce:ExecutionRoleArn` 上下文键的条件，如以下示例所示。

```
{
    "Sid":"AddStepsWithSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:AddJobFlowSteps"
    ],
    "Resource":"*",
    "Condition":{
        "StringEquals":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::<AWS_ACCOUNT_ID>:role/Caller"
            ]
        },
        "StringLike":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::<AWS_ACCOUNT_ID>:role/CallerTeamRole*"
            ]
        }
    }
}
```

在运行时角色和 Amazon EMR 集群之间建立信任

Amazon EMR 为激活运行时角色授权的每种安全配置生成唯一标识符 `ExternalId`。该授权使每个用户都拥有一组运行时角色，可以在属于自己的集群上使用。例如，在企业中，每个部门都可以使用其外部 ID 更新自己的一组运行时角色的信任策略。

您可以使用 Amazon EMR `DescribeSecurityConfiguration` API 查找外部 ID，如以下示例所示。

```
aws emr describe-security-configuration --name 'iamconfig-with-lf'{"Name": "iamconfig-
with-lf",
    "SecurityConfiguration":
        "{\"AuthorizationConfiguration\":{\"IAMConfiguration\":
{\"EnableApplicationScopedIAMRole\
        ":true,\"ApplicationScopedIAMRoleConfiguration\":{\"PropagateSourceIdentity
\":true,\"Exter
        nalId\":\"FXH5TSACFDWUCDSR3YQE2O7ETPUSM4OBCGLYWODSCUZDNZ4Y\"}},\"Lake
        FormationConfiguration\":{\"AuthorizedSessionTagValue\":\"Amazon EMR\"}}}",
    "CreationDateTime": "2022-06-03T12:52:35.308000-07:00"
}
```

有关如何使用外部 ID 的信息，请参阅[如何在向第三方授予对 Amazon 资源的访问权时使用外部 ID](#)。

审核

要使用 IAM 角色监控和控制最终用户执行的操作，您可以开启源身份功能。了解有关源身份的更多信息，请参阅[监控和控制使用所担任角色执行的操作](#)。

要跟踪源身份，请在安全配置中将 `ApplicationScopedIAMRoleConfiguration/PropagateSourceIdentity` 设置为 `true`，如下所示。

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true,
            "ApplicationScopedIAMRoleConfiguration":{
                "PropagateSourceIdentity":true
            }
        }
    }
}
```

将 PropagateSourceIdentity 设置为 true 时，Amazon EMR 会将调用凭证的源身份用于您使用运行时角色创建的任务或查询会话。如果调用凭证中不存在源身份，Amazon EMR 将不会设置源身份。

要使用此属性，请为您的实例配置文件提供 sts:SetSourceIdentity 权限，如下所示。

```
{ // PropagateSourceIdentity statement
    "Sid":"PropagateSourceIdentity",
    "Effect":"Allow",
    "Action":"sts:SetSourceIdentity",
    "Resource":[
        <runtime-role-ARN>
    ],
    "Condition":{
        "StringEquals":{
            "sts:SourceIdentity":<source-identity>
        }
    }
}
```

您还必须将 AllowSetSourceIdentity 语句添加到运行时角色的信任策略中。

```
{ // AllowSetSourceIdentity statement
    "Sid":"AllowSetSourceIdentity",
    "Effect":"Allow",
    "Principal":{
        "AWS":"arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
    },
    "Action":[
        "sts:SetSourceIdentity",
        "sts:AssumeRole"
    ],
    "Condition":{
        "StringEquals":{
            "sts:SourceIdentity":<source-identity>
        }
    }
}
```

其它注意事项

> **Note**
>
> 使用 Amazon EMR 发行版本 `emr-6.9.0`，当您从 SageMaker Studio 连接到 Amazon EMR 集群时，可能会遇到间歇性故障。要解决此问题，您可以在启动集群时通过引导操作来安装补丁。有关补丁的详细信息，请参阅 Amazon EMR release 6.9.0 known issues（Amazon EMR 发行版本 6.9.0 已知问题）。

此外，在为 Amazon EMR 配置运行时角色时，请考虑以下事项。

- Amazon EMR 支持所有商业 Amazon Web Services 区域 中的运行时角色。
- 当您使用发行版本 `emr-6.7.0` 或更高版本时，Amazon EMR 步骤支持具有运行时角色的 Apache Spark 和 Apache Hive 作业。
- 当您使用发行版本 `emr-6.9.0` 或更高版本时，SageMaker Studio 支持具有运行时角色的 Spark、Hive 和 Presto 查询。
- SageMaker 中的以下笔记本内核支持运行时角色：
  - DataScience - Python 3 内核
  - DataScience 2.0 - Python 3 内核
  - DataScience 3.0 - Python 3 内核
  - SparkAnalytics 1.0 - SparkMagic 和 PySpark 内核
  - SparkAnalytics 2.0 - SparkMagic 和 PySpark 内核
  - SparkMagic - PySpark 内核
- Amazon EMR 支持仅在创建集群时使用 `RunJobFlow` 的步骤。此 API 不支持运行时角色。
- Amazon EMR 不支持您配置为高可用性的集群上的运行时角色。
- 运行时角色不支持控制对集群上资源（例如 HDFS 和 HMS）的访问。

为 Amazon EMR 配置 IAM 服务角色对Amazon服务和资源的权限

Amazon EMR 和应用程序（如 Hadoop 和 Spark）在运行时需要访问其它Amazon资源和执行操作的权限。Amazon EMR 中的每个集群必须有一个服务角色和一个适用于 Amazon EC2 实例配置文件的角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 角色和使用实例配置文件。挂载到这些角色的 IAM policy 为集群提供了代表用户与其它Amazon服务进行互操作的权限。

如果您在 Amazon EMR 中的集群使用了自动伸缩，则需要一个额外的角色，即 Auto Scaling 角色。如果您使用 EMR Notebooks，则需要 EMR Notebooks 的Amazon服务角色。

Amazon EMR 为确定权限的每个角色提供默认角色和默认托管式策略。托管式策略由 Amazon 创建和维护，因此如果服务要求发生变化，这些策略将自动更新。请参阅《IAM 用户指南》中的Amazon托管式策略。

如果您是首次在账户中创建集群或笔记本，那么适用于 Amazon EMR 的角色尚不存在。创建它们后，您可以在 IAM 控制台中（https://console.aws.amazon.com/iam/）查看角色、挂载到角色的策略以及策略允许或拒绝的权限。您可以指定要创建和使用的 Amazon EMR 的默认角色，也可以创建自己的角色并在创建集群时分别指定它们以自定义权限，还可以指定在使用 Amazon CLI 创建集群时要使用的默认角色。有关更多信息，请参阅自定义 IAM 角色。

修改基于身份的策略以获取传递 Amazon EMR 服务角色的权限

Amazon EMR 完全权限默认托管策略包含 `iam:PassRole` 安全配置，包括以下内容：

- `iam:PassRole` 权限仅适用于特定默认 Amazon EMR 角色。
- `iam:PassedToService` 条件允许您仅将策略用于指定的Amazon服务，例如 `elasticmapreduce.amazonaws.com` 和 `ec2.amazonaws.com`。

您可以在 IAM 控制台中查看 AmazonEMRFullAccessPolicy_v2 和 AmazonEMRServicePolicy_v2 策略的 JSON 版本。建议您使用 v2 托管式策略创建新的集群。

服务角色摘要

下表列出了与 Amazon EMR 关联的 IAM 服务角色，以供快速参考。

函数	默认角色	描述	默认托管式策略
Amazon EMR 的服务角色（EMR 角色）	`EMR_DefaultRole_V2`	允许 Amazon EMR 在配置资源和执行服务级别操作时代表您调用其它Amazon服务。所有集群都需要该角色。	`AmazonEMRServicePolicy_v2` **Important** 请求竞价型实例需要服务相关角色。如果

函数	默认角色	描述	默认托管式策略
			此角色不存在，Amazon EMR 服务角色必须拥有创建它的权限，否则会出现权限错误。如果您计划请求竞价型实例，您必须更新此策略，以包含一个允许创建此服务相关角色的语句。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的Amazon EMR 的服务角色（EMR 角色）和竞价型实例请求的服务相关角色。

函数	默认角色	描述	默认托管式策略
集群 EC2 实例（EC2 实例配置文件）的服务角色	EMR_EC2_DefaultRole	集群实例上运行在 Hadoop 生态系统之上的应用程序进程在调用其它 Amazon 服务时将使用该角色。要使用 EMRFS 访问 Amazon S3 中的数据，您可以根据数据在 Amazon S3 中的位置指定代入不同的角色。例如，多个团队可以访问单个 Amazon S3 数据“存储账户”。有关更多信息，请参阅为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色。所有集群都需要该角色。	AmazonElasticMapReduceforEC2Role （有关更多信息，请参阅 集群 EC2 实例（EC2 实例配置文件）的服务角色。）
Amazon EMR 中的自动扩展服务角色（Auto Scaling 角色）	EMR_AutoScaling_DefaultRole	允许用于动态扩展环境的其它操作。只有在 Amazon EMR 中使用自动伸缩的集群才需要。有关更多信息，请参阅将自动伸缩与实例组的自定义策略结合使用 。	AmazonElasticMapReduceforAutoScalingRole （有关更多信息，请参阅 Amazon EMR 中的自动扩展服务角色（Auto Scaling 角色）。）

函数	默认角色	描述	默认托管式策略

EMR Notebooks 的服务角色	EMR_Notebooks_DefaultRole	提供 EMR Notebooks 访问其它Amazon资源和执行操作所需的权限。仅在使用 EMR Notebooks 时需要。	AmazonElasticMapReduceEditorsRole （有关更多信息，请参阅 EMR Notebooks 的服务角色。）
			
			默认情况下也附加 S3FullAccessPolicy 。本策略的内容如下所示。
			
			{
			"Version": "2012-10-17",
			"Statement": [
			{
			"Effect": "Allow",
			"Action": "s3:*",
			"Resource": "*"
			}
			]
			}

函数	默认角色	描述	默认托管式策略
服务相关角色	AWSServiceRoleForEMRCleanup	Amazon EMR 自动创建服务相关角色。如果 Amazon EMR 的服务失去了清除 Amazon EC2 资源的能力，Amazon EMR 可以使用此角色来进行清除操作。如果集群使用竞价型实例，则附加到 Amazon EMR 的服务角色（EMR 角色）的权限策略必须允许创建服务相关角色。有关更多信息，请参阅Amazon EMR 的服务相关角色权限。	AmazonEMRCleanupPolicy

主题

- Amazon EMR 使用的 IAM 服务角色
- 自定义 IAM 角色
- 为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色
- 将针对 Amazon EMR 访问的基于资源的策略用于 Amazon Glue 数据目录
- 将 IAM 角色用于直接调用Amazon服务的应用程序
- 允许用户和组创建和修改角色

Amazon EMR 使用的 IAM 服务角色

在预置集群资源、运行应用程序、动态扩展资源以及创建和运行 EMR Notebooks 时，Amazon EMR 使用 IAM 服务角色代表您执行操作。Amazon EMR 与其它Amazon服务交互时使用以下角色。每个角色在 Amazon EMR 中都有唯一的功能。本部分中的主题介绍角色功能，并提供默认角色和每个角色的权限策略。

如果集群上的应用程序代码直接调用 Amazon 服务，则可能需要使用开发工具包来指定角色。有关更多信息，请参阅将 IAM 角色用于直接调用Amazon服务的应用程序。

主题

- Amazon EMR 的服务角色（EMR 角色）
- 集群 EC2 实例（EC2 实例配置文件）的服务角色
- Amazon EMR 中的自动扩展服务角色（Auto Scaling 角色）
- EMR Notebooks 的服务角色
- 对 Amazon EMR 使用服务相关角色

Amazon EMR 的服务角色（EMR 角色）

Amazon EMR 角色定义了在预置资源，以及执行对于在集群中运行的特定 Amazon EC2 实例而言不会执行的服务级任务时，将允许 Amazon EMR 执行的操作。例如，服务角色用于在集群启动时配置 EC2 实例。

- 默认角色名为 `EMR_DefaultRole_V2`。
- 附加到 `EMR_DefaultRole_V2` 的 Amazon EMR 范围的默认托管式策略是 `AmazonEMRServicePolicy_v2`。此 v2 策略将取代弃用的默认托管式策略 `AmazonElasticMapReduceRole`。

`AmazonEMRServicePolicy_v2` 取决于对 Amazon EMR 预调配或使用的资源的范围缩小访问权限。使用此策略时，您需要在预置集群时传递用户标签 `for-use-with-amazon-emr-managed-policies = true`。Amazon EMR 将自动传播这些标签。此外，您可能需要手动向特定类型的资源添加用户标签，例如不是由 Amazon EMR 创建的 EC2 安全组。请参阅标记资源以使用托管式策略。

> **Important**
>
> Amazon EMR 使用此 Amazon EMR 服务角色和 `AWSServiceRoleForEMRCleanup` 角色来清理账户中不再使用的集群资源，例如 Amazon EC2 实例。您必须在角色策略包含删除或终止资源所需的操作。否则，Amazon EMR 将无法执行这些清理操作，并且集群上剩余的未使用资源可能会产生费用。

下面显示的是当前 `AmazonEMRServicePolicy_v2` 策略的内容。您还可以在 IAM 控制台上查看 `AmazonEMRServicePolicy_v2` 托管式策略的当前内容。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "CreateInTaggedNetwork",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:RunInstances",
    "ec2:CreateFleet",
    "ec2:CreateLaunchTemplate",
    "ec2:CreateLaunchTemplateVersion"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateWithEMRTaggedLaunchTemplate",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateFleet",
    "ec2:RunInstances",
    "ec2:CreateLaunchTemplateVersion"
   ],
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateEMRTaggedLaunchTemplate",
   "Effect": "Allow",
   "Action": "ec2:CreateLaunchTemplate",
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
```

```
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateEMRTaggedInstancesAndVolumes",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances",
     "ec2:CreateFleet"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "ResourcesToLaunchEC2",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances",
     "ec2:CreateFleet",
     "ec2:CreateLaunchTemplate",
     "ec2:CreateLaunchTemplateVersion"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*::image/ami-*",
     "arn:aws:ec2:*:*:key-pair/*",
     "arn:aws:ec2:*:*:capacity-reservation/*",
     "arn:aws:ec2:*:*:placement-group/pg-*",
     "arn:aws:ec2:*:*:fleet/*",
     "arn:aws:ec2:*:*:dedicated-host/*",
     "arn:aws:resource-groups:*:*:group/*"
    ]
   },
   {
```

```
    "Sid": "ManageEMRTaggedResources",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplateVersion",
     "ec2:DeleteLaunchTemplate",
     "ec2:DeleteNetworkInterface",
     "ec2:ModifyInstanceAttribute",
     "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "ManageTagsOnEMRTaggedResources",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags",
     "ec2:DeleteTags"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateNetworkInterfaceNeededForPrivateSubnet",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateNetworkInterface"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*"
    ],
```

```
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "TagOnCreateTaggedEMRResources",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": [
       "RunInstances",
       "CreateFleet",
       "CreateLaunchTemplate",
       "CreateNetworkInterface"
      ]
     }
    }
   },
   {
    "Sid": "TagPlacementGroups",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags",
     "ec2:DeleteTags"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:placement-group/pg-*"
    ]
   },
   {
    "Sid": "ListActionsForEC2Resources",
    "Effect": "Allow",
    "Action": [
```

```
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeCapacityReservations",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeImages",
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeNetworkAcls",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribePlacementGroups",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes",
    "ec2:DescribeVolumeStatus",
    "ec2:DescribeVpcAttribute",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcs"
   ],
   "Resource": "*"
  },
  {
   "Sid": "CreateDefaultSecurityGroupWithEMRTags",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSecurityGroup"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:security-group/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateDefaultSecurityGroupInVPCWithEMRTags",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSecurityGroup"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:vpc/*"
   ],
```

```
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "TagOnCreateDefaultSecurityGroupWithEMRTags",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
      "ec2:CreateAction": "CreateSecurityGroup"
     }
    }
   },
   {
    "Sid": "ManageSecurityGroups",
    "Effect": "Allow",
    "Action": [
     "ec2:AuthorizeSecurityGroupEgress",
     "ec2:AuthorizeSecurityGroupIngress",
     "ec2:RevokeSecurityGroupEgress",
     "ec2:RevokeSecurityGroupIngress"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateEMRPlacementGroups",
    "Effect": "Allow",
    "Action": [
     "ec2:CreatePlacementGroup"
    ],
    "Resource": "arn:aws:ec2:*:*:placement-group/pg-*"
   },
```

```
{
 "Sid": "DeletePlacementGroups",
 "Effect": "Allow",
 "Action": [
  "ec2:DeletePlacementGroup"
 ],
 "Resource": "*"
},
{
 "Sid": "AutoScaling",
 "Effect": "Allow",
 "Action": [
  "application-autoscaling:DeleteScalingPolicy",
  "application-autoscaling:DeregisterScalableTarget",
  "application-autoscaling:DescribeScalableTargets",
  "application-autoscaling:DescribeScalingPolicies",
  "application-autoscaling:PutScalingPolicy",
  "application-autoscaling:RegisterScalableTarget"
 ],
 "Resource": "*"
},
{
 "Sid": "ResourceGroupsForCapacityReservations",
 "Effect": "Allow",
 "Action": [
  "resource-groups:ListGroupResources"
 ],
 "Resource": "*"
},
{
 "Sid": "AutoScalingCloudWatch",
 "Effect": "Allow",
 "Action": [
  "cloudwatch:PutMetricAlarm",
  "cloudwatch:DeleteAlarms",
  "cloudwatch:DescribeAlarms"
 ],
 "Resource": "arn:aws:cloudwatch:*:*:alarm:*_EMR_Auto_Scaling"
},
{
 "Sid": "PassRoleForAutoScaling",
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
```

```
    "Condition": {
     "StringLike": {
      "iam:PassedToService": "application-autoscaling.amazonaws.com*"
     }
    }
   },
   {
    "Sid": "PassRoleForEC2",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
    "Condition": {
     "StringLike": {
      "iam:PassedToService": "ec2.amazonaws.com*"
     }
    }
   }
  ]
}
```

您的服务角色应该使用下面的信任策略。

> **Important**
>
> 下面的信任策略包括 aws:SourceArn 和 aws:SourceAccount 全局条件密钥，该密钥将限制您为 Amazon EMR 授予的、针对您账户中特定资源的权限。使用这些策略可以防止混淆代理问题。

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
```

```
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

集群 EC2 实例（EC2 实例配置文件）的服务角色

集群 EC2 实例的服务角色（又称为 Amazon EMR 的 EC2 实例配置文件）是一种特殊类型的服务角色，在实例启动时分配给 Amazon EMR 集群中的每个 EC2 实例。在 Hadoop 生态系统之上运行的应用程序进程代入该角色来获得与其它Amazon服务交互的权限。

有关 EC2 实例的服务角色的更多信息，请参阅《IAM 用户指南》中的使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

> **Important**
>
> 集群 EC2 实例的默认服务角色及其关联的 Amazon 默认托管式策略 `AmazonElasticMapReduceforEC2Role` 即将弃用，没有提供Amazon替代托管式策略。您需要创建并指定实例配置文件以替换弃用的角色和默认策略。

默认角色和托管式策略

- 默认角色名为 `EMR_EC2_DefaultRole`。
- `EMR_EC2_DefaultRole` 默认托管式策略 `AmazonElasticMapReduceforEC2Role` 的支持即将结束。不要为 EC2 实例配置文件使用默认托管式策略，而是将基于资源的策略应用于 Amazon EMR 需要的 S3 桶和其他资源，或者使用您自己的客户管理型策略和 IAM 角色作为实例配置文件。有关更多信息，请参阅创建具有最小权限的集群 EC2 实例服务角色。

下面显示了 `AmazonElasticMapReduceforEC2Role` 版本 3 的内容。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Resource": "*",
            "Action": [
                "cloudwatch:*",
                "dynamodb:*",
                "ec2:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSteps",
                "kinesis:CreateStream",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:MergeShards",
                "kinesis:PutRecord",
                "kinesis:SplitShard",
                "rds:Describe*",
                "s3:*",
                "sdb:*",
                "sns:*",
                "sqs:*",
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetTableVersions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:UpdatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition",
                "glue:CreateUserDefinedFunction",
```

```
                "glue:UpdateUserDefinedFunction",
                "glue:DeleteUserDefinedFunction",
                "glue:GetUserDefinedFunction",
                "glue:GetUserDefinedFunctions"
            ]
        }
    ]
}
```

您的服务角色应该使用下面的信任策略。

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

创建具有最小权限的集群 EC2 实例服务角色

作为一项最佳实践，我们强烈建议您为集群 EC2 实例和权限策略创建一个服务角色，该角色对您的应用程序所需的其它Amazon服务具有最低权限。

默认托管式策略 AmazonElasticMapReduceforEC2Role 提供可轻松启动初始集群的权限。但是，AmazonElasticMapReduceforEC2Role 即将弃用，Amazon EMR 不会为弃用的角色提供替代Amazon托管默认策略。要启动初始集群，您需要提供基于客户管理的资源或基于 ID 的策略。

以下策略语句提供了 Amazon EMR 不同功能所需权限的示例。我们建议您使用这些权限创建权限策略，将访问权限限制为您的集群所需的功能和资源。所有示例策略语句都使用 *us-west-2* 区域和虚构的 Amazon 账户 ID *123456789012*。请将它们替换为适合您集群的值。

有关创建和指定自定义角色的更多信息，请参阅自定义 IAM 角色。

> **Note**
>
> 如果您为 EC2 创建自定义 EMR 角色，请按照基本工作流程自动创建同名实例配置文件。Amazon EC2 允许您创建不同名称的实例配置文件和角色，但 Amazon EMR 不支持此配置，并且在您创建集群时会导致“invalid instance profile”错误。

使用 EMRFS 读写 Amazon S3 中的数据

当在 Amazon EMR 集群上运行的应用程序引用 `s3://mydata` 格式的数据时，Amazon EMR 使用 EC2 实例配置文件发出请求。集群通常以这种方式在 Amazon S3 中读取和写入数据。默认情况下，Amazon EMR 使用附加到集群 EC2 实例的服务角色的权限。有关更多信息，请参阅为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色。

由于 EMRFS 的 IAM 角色将回退到附加到集群 EC2 实例服务角色的权限，因此作为一项最佳实践，我们建议您对 EMRFS 使用 IAM 角色，并限制附加到集群 EC2 实例服务角色的 EMRFS 和 Amazon S3 权限。

下面的示例语句演示了 EMRFS 向 Amazon S3 发出请求所需的权限。

- *my-data-bucket-in-s3-for-emrfs-reads-and-writes* 指定 Amazon S3 中的桶，在该桶中，集群使用 */** 读取和写入数据和所有子文件夹。请仅添加您的应用程序需要的存储桶和文件夹。
- 只有在启用 EMRFS 一致视图时，才需要允许 dynamodb 操作的策略声明。*EmrFSMetadata* 为 EMRFS 一致视图指定默认文件夹。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:CreateBucket",
                "s3:DeleteObject",
                "s3:GetBucketVersioning",
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
```

```
                "s3:ListBucketVersions",
                "s3:ListMultipartUploadParts",
                "s3:PutBucketVersioning",
                "s3:PutObject",
                "s3:PutObjectTagging"
            ],
            "Resource": [
                "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes",
                "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:BatchGetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:DescribeTable",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteTable",
                "dynamodb:UpdateTable"
            ],
            "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/EmrFSMetadata"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "dynamodb:ListTables",
                "s3:ListBucket"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:DeleteQueue",
```

```
                "sqs:SendMessage",
                "sqs:CreateQueue"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:EMRFS-Inconsistency-*"
        }
    ]
}
```

将日志文件存档到 Amazon S3

以下策略语句允许 Amazon EMR 集群将日志文件存档到指定的 Amazon S3 位置。在下面的示例中，在集群创建后，通过控制台的日志文件夹 S3 位置、Amazon CLI 中的 `--log-uri` 选项或 `RunJobFlow` 命令中的 `LogUri` 参数来指定 *s3://MyLoggingBucket/MyEMRClusterLogs*。有关更多信息，请参阅将日志文件归档到 Amazon S3。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::MyLoggingBucket/MyEMRClusterLogs/*"
        }
    ]
}
```

使用调试工具

以下策略语句允许在启用 Amazon EMR 调试工具时所需的操作。调试需要将日志文件存档到 Amazon S3，以及上面示例中所示的相关权限。有关更多信息，请参阅启用调试工具。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:SendMessage"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:AWS-ElasticMapReduce-*"
```

```
        }
    ]
}
```

使用 Amazon Glue 数据目录

以下策略语句允许在将 Amazon Glue 数据目录用作应用程序的元存储时所需的操作。有关更多信息，请参阅《Amazon EMR 版本指南》中的使用 Amazon Glue 数据目录作为 Spark SQL 的元存储、使用 Amazon Glue 数据目录作为 Hive 的元存储和使用 Presto 与 Amazon Glue 数据目录。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetTableVersions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:UpdatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition",
                "glue:CreateUserDefinedFunction",
                "glue:UpdateUserDefinedFunction",
                "glue:DeleteUserDefinedFunction",
                "glue:GetUserDefinedFunction",
                "glue:GetUserDefinedFunctions"
            ],
            "Resource": "*",
        }
```

```
    ]
}
```

Amazon EMR 中的自动扩展服务角色（Auto Scaling 角色）

Amazon EMR 的 Auto Scaling 角色的作用与服务角色相似，但前者允许用于动态扩展环境的其它操作。

- 默认角色名为 `EMR_AutoScaling_DefaultRole`。
- 附加到 `EMR_AutoScaling_DefaultRole` 的默认托管式策略是 `AmazonElasticMapReduceforAutoScalingRole`。

`AmazonElasticMapReduceforAutoScalingRole` 版本 1 的内容如下所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudwatch:DescribeAlarms",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

您的服务角色应该使用下面的信任策略。

> **Important**
>
> 下面的信任策略包括 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件密钥，该密钥将限制您为 Amazon EMR 授予的、针对您账户中特定资源的权限。使用这些策略可以防止混淆代理问题。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service":  [
                    "elasticmapreduce.amazonaws.com",
                    "application-autoscaling.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

EMR Notebooks 的服务角色

每个 EMR Notebooks 都需要权限来访问其它Amazon资源和执行操作。附加到该服务角色的 IAM policy 为笔记本提供了与其它Amazon服务进行交互操作的权限。当您使用Amazon Web Services Management Console创建笔记本时，您指定一个Amazon服务角色。您可以使用默认角色 EMR_Notebooks_DefaultRole，也可以指定您创建的角色。如果之前尚未创建笔记本，则可以选择创建默认角色。

- 默认角色名为 EMR_Notebooks_DefaultRole。
- 默认情况下，附加到 EMR_Notebooks_DefaultRole 的托管式策略是 AmazonElasticMapReduceEditorsRole 和 S3FullAccessPolicy。

您的服务角色应该使用下面的信任策略。

> **Important**
>
> 下面的信任策略包括 aws:SourceArn 和 aws:SourceAccount 全局条件密钥，该密钥将限制您为 Amazon EMR 授予的、针对您账户中特定资源的权限。使用这些策略可以防止混淆代理问题。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

版本 1 的内容 AmazonElasticMapReduceEditorsRole 如下所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:CreateNetworkInterface",
```

```
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeNetworkInterfaces",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DescribeTags",
                "ec2:DescribeInstances",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListSteps"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "aws:elasticmapreduce:editor-id",
                        "aws:elasticmapreduce:job-flow-id"
                    ]
                }
            }
        }
    ]
}
```

下面是 S3FullAccessPolicy 的内容。S3FullAccessPolicy 允许 EMR Notebooks 的服务角色对 Amazon Web Services 账户 中的对象执行所有 Amazon S3 操作。当您为 EMR Notebooks 创建自定义服务角色时，必须向服务角色授予 Amazon S3 权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        }
```

```
    ]
}
```

您可以将服务角色的读写权限范围缩小到要保存笔记本文件的 Amazon S3 位置。使用以下最小 Amazon S3 权限集。

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

如果您的 Amazon S3 存储桶已加密，您必须为 Amazon Key Management Service 包含以下权限。

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

当您将 Git 存储库链接到笔记本并需要为存储库创建密钥时，您必须在附加到 Amazon EMR Notebooks 的服务角色的 IAM policy 中添加 `secretsmanager:GetSecretValue` 权限。下面演示了一个示例策略：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

EMR Notebooks 服务角色权限

此表列出了 EMR Notebooks 使用服务角色执行的操作，以及每个操作所需的权限。

操作	权限

在笔记本和 Amazon EMR 集群之间建立安全的网络通道，并执行必要的清理操作。	`"ec2:CreateNetworkInterface",`
	`"ec2:CreateNetworkInterfacePermission",`
	`"ec2:DeleteNetworkInterface",`
	`"ec2:DeleteNetworkInterfacePermission",`
	`"ec2:DescribeNetworkInterfaces",`
	`"ec2:ModifyNetworkInterfaceAttribute",`
	`"ec2:AuthorizeSecurityGroupEgress",`
	`"ec2:AuthorizeSecurityGroupIngress",`
	`"ec2:CreateSecurityGroup",`
	`"ec2:DescribeSecurityGroups",`
	`"ec2:RevokeSecurityGroupEgress",`
	`"ec2:DescribeTags",`
	`"ec2:DescribeInstances",`
	`"ec2:DescribeSubnets",`
	`"ec2:DescribeVpcs",`
	`"elasticmapreduce:ListInstances",`
	`"elasticmapreduce:DescribeCluster",`
	`"elasticmapreduce:ListSteps"`

使用存储在 Amazon Secrets Manager 中的 Git 凭证将 Git 存储库链接到笔记本。	`"secretsmanager:GetSecretValue"`
将Amazon标签应用于 EMR Notebooks 在设置安全网络通道时创建的网络接口和默认安全组。有关更多信息，请参阅标记Amazon资源。	`"ec2:CreateTags"`

访问笔记本文件和元数据或将它们上载到 Amazon S3。	`"s3:PutObject",`
	`"s3:GetObject",`
	`"s3:GetEncryptionConfiguration",`
	`"s3:ListBucket",`
	`"s3:DeleteObject"`
	
	仅当您使用加密的 Amazon S3 存储桶时才需要以下权限。
	
	`"kms:Decrypt",`
	`"kms:GenerateDataKey",`

操作	权限
	"kms:ReEncryptFrom", "kms:ReEncryptTo", "kms:DescribeKey"

Amazon 托管策略的 EMR Notebooks 更新

查看有关自 2021 年 3 月 1 日以来针对 EMR Notebooks 的 Amazon 托管策略更新的详细信息。

更改	说明	日期
AmazonElasticMapReduceEditorsRole - Added permissions	EMR Notebooks 已将 ec2:describeVPCs 和 elastmicmapreduce:ListSteps 权限添加至 AmazonElasticMapReduceEditorsRole 。	2023 年 2 月 8 日
EMR Notebooks 开启了跟踪修订	EMR Notebooks 为 Amazon 托管策略开启了跟踪修订。	2023 年 2 月 8 日

对 Amazon EMR 使用服务相关角色

Amazon EMR 使用 Amazon Identity and Access Management（IAM）服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon EMR 直接相关。服务相关角色由 Amazon EMR 预定义，包含 Amazon EMR 代表您调用 Amazon EC2 来清除不再使用的集群资源所需的权限。服务相关角色与 Amazon EMR 服务角色和 Amazon EMR 的 Amazon EC2 实例配置文件一起使用。有关服务角色和实例配置文件的更多信息，请参阅为 Amazon EMR 配置 IAM 服务角色对Amazon服务和资源的权限。

Amazon EMR 定义此服务相关角色的权限，除非另外定义，否则只有 Amazon EMR 可以代入该角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其它 IAM 实体的权限策略。只能在终止账户中的所有 EMR 集群后删除角色。

有关支持服务相关角色的其他服务的信息，请参阅可与 IAM 配合使用的 Amazon 服务，并查找 Service-Linked Role（服务相关角色）列中为 Yes（是）的服务。请选择 Yes 与查看该服务的服务相关角色文档的链接。

Amazon EMR 的服务相关角色权限

Amazon EMR 使用 AWSServiceRoleForEMRCleanup 角色，这是一个基于服务的角色。如果 Amazon EMR 服务角色失去该能力，它允许 Amazon EMR 代表您终止和删除 Amazon EC2 资源。如果该角色尚不存在，Amazon EMR 在集群创建期间会自动创建该角色。

AWSServiceRoleForEMRCleanup 服务相关角色信任以下服务以代入该角色：

- `elasticmapreduce.amazonaws.com`

AWSServiceRoleForEMRCleanup 服务相关角色权限策略允许 Amazon EMR 对指定资源完成以下操作：

- 操作：`ec2` 上的 `DescribeInstances`
- 操作：`DescribeSpotInstanceRequests` 上的 `ec2`
- 操作：`ModifyInstanceAttribute` 上的 `ec2`
- 操作：`TerminateInstances` 上的 `ec2`
- 操作：`CancelSpotInstanceRequests` 上的 `ec2`
- 操作：`DeleteNetworkInterface` 上的 `ec2`
- 操作：`DescribeInstanceAttribute` 上的 `ec2`
- 操作：`DescribeVolumeStatus` 上的 `ec2`
- 操作：`DescribeVolumes` 上的 `ec2`
- 操作：`DetachVolume` 上的 `ec2`
- 操作：`ec2` 上的 `DeleteVolume`

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。

允许 IAM 实体创建 AWSServiceRoleForEMRCleanup 服务关联角色

将以下语句添加到需要创建服务相关角色的 IAM 实体的权限策略：

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
```

```
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

允许 IAM 实体编辑 AWSServiceRoleForEMRCleanup 服务关联角色的描述

将以下语句添加到需要编辑服务相关角色的描述的 IAM 实体的权限策略：

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

允许 IAM 实体删除 AWSServiceRoleForEMRCleanup 服务关联角色

将以下语句添加到需要删除服务相关角色的 IAM 实体的权限策略：

```
{
    "Effect": "Allow",
    "Action": [
```

```
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

为 Amazon EMR 创建服务相关角色

您无需手动创建 AWSServiceRoleForEMRCleanup 角色。当您启动集群时，无论是首次启动还是当服务相关角色不存在时，Amazon EMR 都会为您创建服务相关角色。您必须具备创建服务相关角色的 权限。有关将此功能添加到 IAM 实体（例如，用户、组或角色）的权限策略的示例语句，请参阅Amazon EMR 的服务相关角色权限。

> **Important**
>
> 如果您在 2017 年 10 月 24 日之前使用 Amazon EMR，当服务相关角色不受支持时，Amazon EMR 会在您的账户中创建 AWSServiceRoleForEMRCleanup 角色。有关更多信息，请参阅我的 IAM 账户中出现新角色。

为 Amazon EMR 编辑服务相关角色

Amazon EMR 不允许您编辑 AWSServiceRoleForEMRCleanup 服务相关角色。创建服务相关角色后，将无法更改角色名称，因为可能有多个实体引用该角色。但是可以使用 IAM 编辑角色说明。

编辑服务相关角色描述（IAM 控制台）

您可以使用 IAM 控制台编辑服务相关角色的描述。

编辑服务相关角色的描述 (控制台)

1. 在 IAM 控制台的导航窗格中，选择 Roles。

2. 以下代码示例显示如何将 IAM policy 附加到用户。
3. 在 Role description (角色描述) 的右侧，选择 Edit (编辑)。
4. 在框中键入新描述，然后选择 Save changes (保存更改)。

编辑服务相关角色描述（IAM CLI）

您可以从 Amazon Command Line Interface 使用 IAM 命令编辑服务相关角色的描述。

更改服务相关角色的描述 (CLI)

1. (可选) 要查看角色的当前描述，请使用以下命令：

   ```
   $ aws iam get-role --role-name role-name
   ```

   通过 CLI 命令使用角色名称（并非 ARN）指向角色。例如，如果某个角色的 ARN 为 `arn:aws:iam::123456789012:role/myrole`，则将该角色称为 **myrole**。
2. 要更新服务相关角色的描述，请使用以下命令之一：

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

编辑服务相关角色描述（IAM API）

您可以使用 IAM API 编辑服务相关角色的描述。

更改服务相关角色的描述 (API)

1. (可选) 要查看角色的当前描述，请使用以下命令：

   IAM API：GetRole
2. 要更新角色的描述，请使用以下命令：

   IAM API：UpdateRoleDescription

删除适用于 Amazon EMR 的服务相关角色

如果不再需要使用某个需要服务相关角色的特征或服务，我们建议您删除该角色。这样，您就不会有未被主动监控或维护的未使用实体。但是，您必须先清除您的服务相关角色，然后才能将其删除。

清除服务相关角色

必须先确认服务相关角色没有活动会话并删除该角色使用的任何资源，然后才能使用 IAM 删除服务相关角色。

在 IAM 控制台中检查服务相关角色是否具有活动会话

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）。选择 AWSServiceRoleForEMRCleanup 角色的名称（不是复选框）。
3. 在选定角色的 Summary (摘要) 页面上，选择 Access Advisor (访问顾问)。
4. 在 Access Advisor (访问顾问) 选项卡上，查看服务相关角色的近期活动。

> **Note**
>
> 如果您不确定 Amazon EMR 是否正在使用 AWSServiceRoleForEMRCleanup 角色，可以尝试删除该角色。如果服务正在使用该角色，则删除操作会失败，并且您可以查看正在使用该角色的 区域。如果该角色已被使用，则您必须等待会话结束，然后才能删除该角色。您无法撤销服务相关角色对会话的权限。

删除 AWSServiceRoleForEMRCleanup 使用的 Amazon EMR 资源

- 终止您的账户中的所有集群。有关更多信息，请参阅终止集群。

删除服务相关角色（IAM 控制台）

您可以使用 IAM 控制台删除服务相关角色。

删除服务相关角色（控制台）

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择 Roles（角色）。选中 AWSServiceRoleForEMRCleanup 旁边的复选框，而不是选择名称或行本身。
3. 对于页面顶部的角色操作，请选择删除角色。
4. 在确认对话框中，查看上次访问服务数据，该数据显示每个选定角色上次访问Amazon服务的时间。这样可帮助您确认角色当前是否处于活动状态。要继续，请选择 Yes, Delete（是的，删除）。

5. 监视 IAM 控制台通知，以监控服务相关角色的删除进度。由于 IAM 服务相关角色删除是异步的，因此，在您提交角色进行删除后，删除任务可能成功，也可能失败。如果任务失败，您可以从通知中选择 View details 或 View Resources 以了解删除失败的原因。如果因角色正在使用服务中的资源而导致删除操作失败，则失败原因将包含一个资源列表。

删除服务相关角色（IAM CLI）

您可以从 Amazon Command Line Interface 使用 IAM 命令删除服务相关角色。如果服务相关角色正被使用或具有关联的资源，则无法删除它，因此您必须提交删除请求。如果不满足这些条件，该请求可能会被拒绝。

删除服务相关角色 (CLI)

1. 要查看删除任务的状态，您必须从响应中捕获 `deletion-task-id`。键入以下命令以提交服务相关角色的删除请求：

   ```
   $ aws iam delete-service-linked-role --role-name AWSServiceRoleForEMRCleanup
   ```

2. 键入以下命令以检查删除任务的状态：

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   删除任务的状态可能是 `NOT_STARTED`、`IN_PROGRESS`、`SUCCEEDED` 或 `FAILED`。如果删除失败，则调用会返回失败的原因，以便您进行问题排查。

删除服务相关角色（IAM API）

您可以使用 IAM API 删除服务相关角色。如果服务相关角色正被使用或具有关联的资源，则无法删除它，因此您必须提交删除请求。如果不满足这些条件，该请求可能会被拒绝。

删除服务相关角色 (API)

1. 要提交服务相关角色的删除请求，请调用 DeleteServiceLinkedRole。在请求中，指定 AWSServiceRoleForEMRCleanup 角色名称。

   要查看删除任务的状态，您必须从响应中捕获 `DeletionTaskId`。

2. 要检查删除的状态，请调用 GetServiceLinkedRoleDeletionStatus。在请求中，指定 `DeletionTaskId`。

删除任务的状态可能是 NOT_STARTED、IN_PROGRESS、SUCCEEDED 或 FAILED。如果删除失败，则调用会返回失败的原因，以便您进行问题排查。

Amazon EMR 服务相关角色支持的区域

Amazon EMR 支持在以下区域中使用服务相关角色。

区域名称	区域标识	Amazon EMR 支持
美国东部（弗吉尼亚州北部）	us-east-1	是
US East (Ohio)	us-east-2	是
美国西部（北加利福尼亚）	us-west-1	是
美国西部（俄勒冈州）	us-west-2	是
亚太地区（孟买）	ap-south-1	是
亚太地区（大阪）	ap-northeast-3	是
亚太地区（首尔）	ap-northeast-2	是
亚太地区（新加坡）	ap-southeast-1	是
Asia Pacific (Sydney)	ap-southeast-2	是
亚太地区（东京)	ap-northeast-1	是
Canada (Central)	ca-central-1	是
欧洲（法兰克福）	eu-central-1	是
Europe (Ireland)	eu-west-1	是
欧洲（伦敦）	eu-west-2	是
欧洲（巴黎）	eu-west-3	是
South America (São Paulo)	sa-east-1	是

区域名称	区域标识	Amazon EMR 支持
中国（北京）	cn-north-1	否

自定义 IAM 角色

根据安全要求，您可能需要自定义 IAM 服务角色和权限，从而对权限加以限制。要自定义权限，我们建议您创建新的角色和策略。从托管式策略中的默认角色 (例如 `AmazonElasticMapReduceforEC2Role` 和 `AmazonElasticMapReduceRole`) 的权限开始着手。将相应内容复制并粘贴到新的策略语句中，根据具体情况修改权限，并将修改后的权限策略附加到您创建的角色。您必须拥有合适的 IAM 权限才能使用角色和策略。有关更多信息，请参阅允许用户和组创建和修改角色。

如果您为 EC2 创建自定义 EMR 角色，请按照基本工作流程自动创建同名实例配置文件。Amazon EC2 允许您创建不同名称的实例配置文件和角色，但 Amazon EMR 不支持此配置，并且在您创建集群时会导致“invalid instance profile”错误。

> **Important**
>
> 当服务要求发生变化时，内联策略不会自动更新。如果您创建并附加内联策略，请注意可能会发生突然导致权限错误的服务更新。有关更多信息，请参阅《IAM 用户指南》中的托管式策略与内联策略和在创建集群时指定自定义 IAM 角色。

有关处理 IAM 角色的更多信息，请参阅《IAM 用户指南》中的以下主题：

- 创建向 Amazon 服务委派权限的角色
- 修改角色
- 删除角色

在创建集群时指定自定义 IAM 角色

在创建集群时，您可以指定适用于 Amazon EMR 的服务角色和适用于 Amazon EC2 实例配置文件的角色。创建集群的用户需要拥有检索并向 Amazon EMR 和 EC2 实例分配角色的权限。否则，会出现未授权该账户调用 EC2 错误。有关更多信息，请参阅允许用户和组创建和修改角色。

使用控制台指定自定义角色

在创建集群时，您可以使用 Advanced options (高级选项) 指定适用于 Amazon EMR 的自定义服务角色、适用于 EC2 实例配置文件的自定义角色和自定义 Auto Scaling 角色。当您使用 Quick options (快速选项) 时，将指定默认服务角色和适用于 EC2 实例配置文件的默认角色。有关更多信息，请参阅Amazon EMR 使用的 IAM 服务角色。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台指定自定义 IAM 角色

使用新控制台创建集群时，您必须指定适用于 Amazon EMR 的自定义服务角色和适用于 EC2 实例配置文件的自定义角色。有关更多信息，请参阅Amazon EMR 使用的 IAM 服务角色。

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Security configuration and permissions（安全配置和权限）下，找到 IAM role for instance profile（适用于实例配置文件的 IAM 角色）和 Service role for Amazon EMR（适用于 Amazon EMR 的服务角色）字段。对于每个角色类型，从列表中选择一个角色。系统只会列出您的账户中具有针对相应角色类型的适当信任策略的角色。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

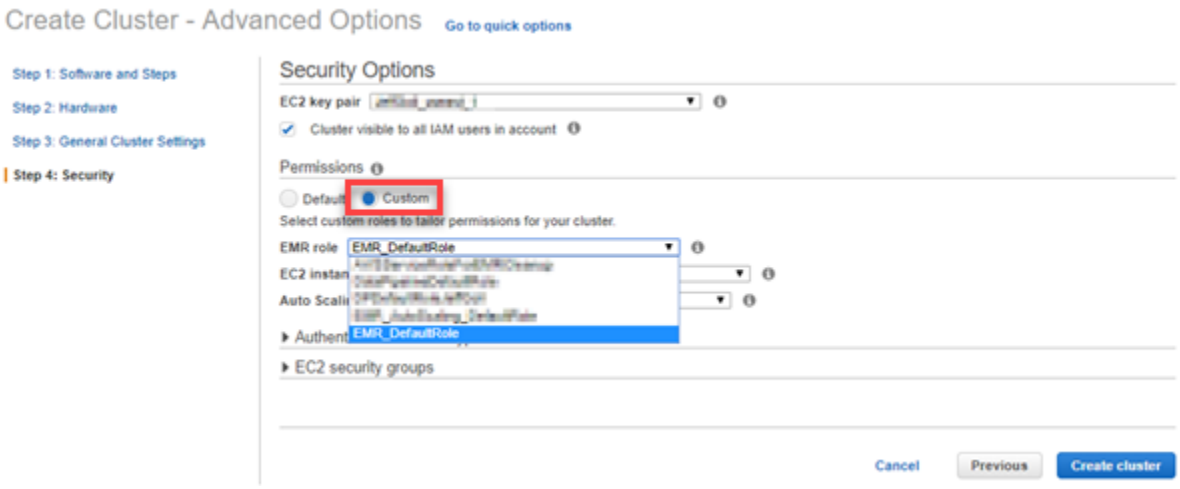
Old console

使用旧控制台指定自定义 IAM 角色

使用旧控制台创建集群时，您可以使用 Advanced options（高级选项）指定适用于 Amazon EMR 的自定义服务角色、适用于 EC2 实例配置文件的自定义角色和自定义自动扩展角色。当您使用

Quick options (快速选项) 时，将指定默认服务角色和适用于 EC2 实例配置文件的默认角色。有关更多信息，请参阅Amazon EMR 使用的 IAM 服务角色。

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 依次选择 Create cluster (创建集群)、Go to advanced options (转到高级选项)。
3. 选择适合您应用程序的集群设置，直至到达 Security Options (安全选项)。在 Permissions (权限) 下，选择适用于 Amazon EMR 的 Default (默认) 角色。
4. 选择 Custom (自定义)。
5. 对于每个角色类型，从列表中选择一个角色。系统只会列出您的账户中具有针对相应角色类型的适当信任策略的角色。



6. 选择适用于您集群的其它选项，然后选择 Create Cluster (创建集群)。

使用 Amazon CLI 指定自定义角色

您可以在 Amazon CLI 中使用 `create-cluster` 命令的选项显式指定 Amazon EMR 的服务角色和集群 EC2 实例的服务角色。使用 `--service-role` 选项指定服务角色。使用 `InstanceProfile` 选项的 `--ec2-attributes` 参数指定适用于 EC2 实例配置文件的角色。

将使用单独的选项（`--auto-scaling-role`）指定 Auto Scaling 角色。有关更多信息，请参阅将自动伸缩与实例组的自定义策略结合使用 。

使用 Amazon CLI 指定自定义 IAM 角色

- 以下命令将在启动集群时指定自定义服务角色 *MyCustomServiceRoleForEMR* 和适用于 EC2 实例配置文件的自定义角色 *MyCustomServiceRoleForClusterEC2Instances*。本示例使用默认 Amazon EMR 角色。

  > **Note**
  >
  > 为了便于读取，包含 Linux 行继续符 (\ )。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^)。

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.36.1 \
--applications Name=Hive Name=Pig --service-role MyCustomServiceRoleForEMR \
--ec2-attributes InstanceProfile=MyCustomServiceRoleForClusterEC2Instances,\
KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

您可以使用这些选项 (而不是使用 `--use-default-roles` 选项) 来显式指定默认角色。`--use-default-roles` 选项指定服务角色以及 Amazon CLI 的 `config` 文件中定义的 EC2 实例配置文件的角色。

以下示例显示了 Amazon CLI 的 `config` 文件的内容，该文件指定了 Amazon EMR 的自定义角色。使用此配置文件，指定 `--use-default-roles` 选项后，将使用 *MyCustomServiceRoleForEMR* 和 *MyCustomServiceRoleForClusterEC2Instances* 创建集群。默认情况下，`config` 文件将默认 `service_role` 指定为 `AmazonElasticMapReduceRole`，将默认 `instance_profile` 指定为 `EMR_EC2_DefaultRole`。

```
[default]
output = json
region = us-west-1
aws_access_key_id = myAccessKeyID
aws_secret_access_key = mySecretAccessKey
emr =
     service_role = MyCustomServiceRoleForEMR
     instance_profile = MyCustomServiceRoleForClusterEC2Instances
```

为处理 EMRFS 对 Amazon S3 的请求配置 IAM 角色

当在集群上运行的应用程序引用使用 `s3://mydata` 格式的数据时，Amazon EMR 使用 EMRFS 发出请求。为了与 Amazon S3 交互，EMRFS 会代入附加到您的 Amazon EC2 实例配置文件的权限策略。无论运行应用程序的用户或组或数据在 Amazon S3 中的位置如何，都会使用相同的 Amazon EC2 实例配置文件。

如果您的集群中有多个用户需要通过 EMRFS 对 Amazon S3 中的数据进行不同级别的访问，则可以为 EMRFS 设置具有 IAM 角色的安全配置。EMRFS 可以根据发出请求的用户或组，或基于数据在 Amazon S3 中的位置，为集群 EC2 实例代入不同的服务角色。EMRFS 的每个 IAM 角色都可以对 Amazon S3 中的数据具有不同访问权限。有关针对集群 EC2 实例的服务角色更多信息，请参阅 集群 EC2 实例（EC2 实例配置文件）的服务角色。

Amazon EMR 版本 5.10.0 和更高版本支持为 EMRFS 使用自定义 IAM 角色。如果您使用的是早期版本或有 EMRFS 的 IAM 角色无法满足的其它要求，可以创建自定义凭证提供程序。有关更多信息，请参阅授权访问 Amazon S3 中的 EMRFS 数据。

当您使用安全配置为 EMRFS 指定 IAM 角色时，请设置角色映射。每个角色映射都指定与标识符对应的 IAM 角色。这些标识符确定通过 EMRFS 访问 Amazon S3 的基础。标识符可以是用户、组或指示数据位置的 Amazon S3 前缀。当 EMRFS 向 Amazon S3 发出请求时，如果请求与访问基础匹配，则 EMRFS 让集群 EC2 实例代入与该请求对应的 IAM 角色。对于集群 EC2 实例，应用的是附加在该角色上的 IAM 权限，而不是附加在服务角色上的 IAM 权限。

角色映射中的用户和组是在集群上定义的 Hadoop 用户和组。用户和组将传递到使用 EMRFS 的应用程序的上下文中的 EMRFS (例如，YARN 用户模拟)。Amazon S3 前缀可以是任意深度的存储桶说明符（例如，`s3://mybucket` 或 `s3://mybucket/myproject/mydata`）。您可以在单个角色映射中指定多个标识符，但它们必须是同一类型。

> **Important**
>
> EMRFS 的 IAM 角色在应用程序的各用户之间提供应用程序级别的隔离。它不提供主机上各用户之间的主机级别的隔离。任何有权访问集群的用户都可以绕过隔离以代入任何角色。

当集群应用程序通过 EMRFS 向 Amazon S3 发出请求时，EMRFS 按角色映射在安全配置中出现的先后顺序评估角色映射。如果通过 EMRFS 发出的请求与任何标识符都不匹配，EMRFS 将回退到使用集群 EC2 实例的服务角色。因此，我们建议附加到此角色的策略限制对 Amazon S3 的权限。有关更多信息，请参阅集群 EC2 实例（EC2 实例配置文件）的服务角色。

配置 角色

在为 EMRFS 的 IAM 角色设置安全配置之前，请计划并创建角色和要附加到角色的权限策略。有关更多信息，请参阅《IAM 用户指南》中的 EC2 实例的角色的工作方式。创建权限策略时，建议先使用附加到 EC2 默认 Amazon EMR 角色的托管式策略，然后根据自己的要求编辑该策略。默认角色名为 EMR_EC2_DefaultRole，要编辑的默认托管式策略为 AmazonElasticMapReduceforEC2Role。有关更多信息，请参阅集群 EC2 实例（EC2 实例配置文件）的服务角色。

更新信任策略以代入角色权限

EMRFS 使用的每个角色必须具有信任策略，以允许 EC2 的集群 Amazon EMR 角色代入该角色。与此类似，EC2 的集群 Amazon EMR 角色必须具有信任策略，以允许 EMR 角色代入该角色。

以下示例信任策略附加到 EMRFS 的角色。该语句允许 EC2 的默认 Amazon EMR 角色代入该角色。例如，如果您有两个虚构 EMRFS 角色 EMRFSRole_First 和 EMRFSRole_Second，此策略语句添加到这两个角色的信任策略。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::AWSAcctID:role/EMR_EC2_DefaultRole"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

此外，以下示例信任策略语句添加到 EMR_EC2_DefaultRole 以允许两个虚构 EMRFS 角色代入该角色。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": ["arn:aws:iam::AWSAcctID:role/EMRFSRole_First",
 "arn:aws:iam::AWSAcctID:role/EMRFSRole_Second"]
```

```
        },
        "Action":"sts:AssumeRole"
      }
   ]
}
```

更新 IAM 角色的信任策略

通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。

1. 选择 Roles (角色)，在 Search (搜索) 中输入角色的名称，然后选择其 Role name (角色名称)。
2. 选择信任关系，然后选择编辑信任关系。
3. 根据 Policy Document（策略文档）和上述指南添加信任语句，然后选择 Update Trust Policy（更新信任策略）。

指定角色作为密钥用户

如果某个角色允许访问 Amazon S3 中使用 Amazon KMS key 进行加密的位置，请务必将该角色指定为密钥用户。这会为该角色授予使用该 KMS 密钥的权限。有关更多信息，请参阅Amazon Key Management Service开发人员指南中的在 Amazon KMS 中使用密钥策略。

为 EMRFS 的 IAM 角色设置安全配置

> **Important**
>
> 如果您指定的 EMRFS 的所有 IAM 角色都不适用，EMRFS 将回退使用 EC2 的 Amazon EMR 角色。在创建集群时，可以考虑根据应用程序的需要自定义角色以限制对 Amazon S3 的权限，然后指定该自定义角色，而不是指定 `EMR_EC2_DefaultRole`。有关更多信息，请参阅自定义 IAM 角色和在创建集群时指定自定义 IAM 角色。

使用控制台指定用于处理 EMRFS 对 Amazon S3 的请求的 IAM 角色

1. 创建一个安全配置来指定角色映射：
   a. 在 Amazon EMR 控制台中，依次选择 Security configurations (安全配置) 和 Create (创建)。
   b. 键入安全配置的 Name (名称)。在创建集群时，使用此名称来指定安全配置。
   c. 选择 Use IAM roles for EMRFS requests to Amazon S3（使用 IAM 角色处理 EMRFS 对 Amazon S3 的请求）。

d. 选择要应用的 IAM role（IAM 角色），在 Basis for access（访问基础）下的列表中选择标识符类型 [Users（用户）、Groups（组）或 S3 prefixes（S3 前缀）]，然后输入相应标识符。如果使用多个标识符，请将它们以逗号分隔，不要包含空格。有关每个标识符类型的更多信息，请参阅以下 JSON configuration reference。

e. 选择Add role (添加角色)设置其它角色映射，如上一步所述。

f. 根据需要设置其它安全配置选项，然后选择 Create (创建)。有关更多信息，请参阅创建安全配置。

2. 创建集群时，指定在上面创建的安全配置。有关更多信息，请参阅为集群指定安全配置。

使用 Amazon CLI 指定用于处理 EMRFS 对 Amazon S3 的请求的 IAM 角色

1. 使用 `aws emr create-security-configuration` 命令，并指定安全配置的名称以及 JSON 格式的安全配置详细信息。

   下面所示的示例命令创建名为 `EMRFS_Roles_Security_Configuration` 的安全配置。它基于保存在执行该命令的目录下的文件 `MyEmrfsSecConfig.json` 中的 JSON 结构。

   ```
   aws emr create-security-configuration --name EMRFS_Roles_Security_Configuration --
   security-configuration file://MyEmrFsSecConfig.json.
   ```

   按照以下准则处理 `MyEmrFsSecConfig.json` 文件的结构。您可以与其它安全配置选项的结构一起指定此结构。有关更多信息，请参阅创建安全配置。

   以下是一个示例 JSON 代码段，用于在安全配置中为 EMRFS 指定自定义 IAM 角色。它演示了三种不同标识符类型的角色映射，后跟参数引用。

   ```
   {
     "AuthorizationConfiguration": {
       "EmrFsConfiguration": {
         "RoleMappings": [{
           "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
           "IdentifierType": "User",
           "Identifiers": [ "user1" ]
         },{
           "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_MyBuckets",
           "IdentifierType": "Prefix",
           "Identifiers": [ "s3://MyBucket/","s3://MyOtherBucket/" ]
         },{
           "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
   ```

```
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

参数	描述
"AuthorizationConfiguration":	必需。
"EmrFsConfiguration":	必需。包含角色映射。
"RoleMappings":	必需。包含一个或多个角色映射定义。按照角色映射的先后顺序对角色映射进行评估。对于 Amazon S3 中的 EMRFS 数据调用，如果角色映射评估为 true，则不会评估其它角色映射，并且 EMRFS 将使用指定的 IAM 角色进行请求。角色映射由以下必需参数组成：
"Role":	以 arn:aws:iam:: *account-id* :role/*role-name* 格式指定 IAM 角色的 ARN 标识符。如果 EMRFS 对 Amazon S3 的请求与指定的任何 Identifiers 匹配，则这是 Amazon EMR 承担的 IAM 角色。

参数	描述
"IdentifierType":	可以是以下值之一： • "User" 指定标识符是一个或多个 Hadoop 用户，可以是 Linux 账户用户或 Kerberos 主体。如果 EMRFS 请求来自一个或多个指定用户，则担任 IAM 角色。 • "Prefix" 指定该标识符是 Amazon S3 位置。用于调用具有指定前缀的一个或多个位置时代入 IAM 角色。例如，前缀 s3://mybucket/ 匹配 s3://mybucket/mydir 和 s3://mybucket/yetanotherdir 。 • "Group" 指定标识符是一个或多个 Hadoop 组。如果请求来自指定组或组中的用户，则代入 IAM 角色。
"Identifiers":	指定一个或多个相应标识符类型的标识符。用逗号分隔多个标识符，不含空格。

2. 使用 aws emr create-cluster 命令创建集群，并指定上一步中创建的安全配置。

以下示例创建一个安装了默认核心 Hadoop 应用程序的集群。该集群使用上面创建的安全配置作为 EMRFS_Roles_Security_Configuration，还使用 EC2 的自定义 Amazon EMR 角色 EC2_Role_EMR_Restrict_S3，该角色由 InstanceProfile 参数的 --ec2-attributes 变量指定。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^) 。

```
aws emr create-cluster --name MyEmrFsS3RolesCluster \
--release-label emr-5.36.1 --ec2-attributes
 InstanceProfile=EC2_Role_EMR_Restrict_S3,KeyName=MyKey \
--instance-type m5.xlarge --instance-count 3 \
```

```
--security-configuration EMRFS_Roles_Security_Configuration
```

将针对 Amazon EMR 访问的基于资源的策略用于 Amazon Glue 数据目录

如果您将 Amazon Glue 与 Amazon EMR 中的 Hive、Spark 或 Presto 结合使用，Amazon Glue 支持使用基于资源的策略来控制对数据目录资源的访问权限。这些资源包括数据库、表、连接和用户定义的函数。有关更多信息，请参阅《Amazon Glue 开发人员指南》中的 Amazon Glue 资源策略。

当使用基于资源的策略来限制从 Amazon EMR 中访问 Amazon Glue 时，在权限策略中指定的委托人必须是与创建集群时指定的 EC2 实例配置文件相关联的角色 ARN。例如，对于附加到目录的基于资源的策略，您可以使用以下示例中显示的格式为集群 EC2 实例的默认服务角色指定角色 ARN，将 *EMR_EC2_DefaultRole* 指定为 `Principal`：

```
arn:aws:iam::acct-id:role/EMR_EC2_DefaultRole
```

acct-id 可以与 Amazon Glue 账户 ID 不同。这允许从不同账户中的 EMR 集群进行访问。您可以指定多个委托人，且每个委托人都可以来自不同的账户。

将 IAM 角色用于直接调用Amazon服务的应用程序

在调用 Amazon 服务时，集群的 EC2 实例上运行的应用程序可以使用 EC2 实例配置文件获取临时安全凭证。

Amazon EMR 版本 2.3.0 和更高版本上的 Hadoop 版本已经过更新，可以使用 IAM 角色。如果您的应用程序仅限于在 Hadoop 架构的顶层运行，并不直接调用Amazon中的任何服务，那么它应使用未做修改的 IAM 角色。

如果您的应用程序在Amazon中直接调用服务，那么您需要更新该应用程序，以便可以利用 IAM 角色。这意味着，您的应用程序现在不是从集群中 EC2 实例上的 `/etc/hadoop/conf/core-site.xml` 获取账户凭证，而是使用 SDK 以 IAM 角色访问资源，或调用 EC2 实例元数据以获取临时凭证。

使用 SDK 以 IAM 角色访问Amazon资源

- 以下主题介绍了如何使用一些 Amazon SDK 以 IAM 角色访问临时凭证。每个主题的开头介绍了不使用 IAM 角色的应用程序的版本，然后向您分步讲解将该应用程序转换为可以使用 IAM 角色的过程。
  - 《Amazon SDK for Java开发人员指南》中的使用 SDK for Java 配置用于 Amazon EC2 实例的 IAM 角色

- 《Amazon SDK for .NET Developer Guide》中的使用 SDK for .NET 配置用于 Amazon EC2 实例的 IAM 角色
- 《Amazon SDK for PHP开发人员指南》中的使用 SDK for PHP 配置用于 Amazon EC2 实例的 IAM 角色
- 《Amazon SDK for Ruby开发人员指南》中的使用 SDK for Ruby 配置用于 Amazon EC2 实例的 IAM 角色

从 EC2 实例元数据中获取临时凭证

- 从以指定 IAM 角色运行的 EC2 实例调用以下 URL，这将返回关联的临时安全凭证（AccessKeyId、SecretAccessKey、SessionToken 和 Expiration）。以下示例使用 Amazon EMR 的默认实例配置文件 `EMR_EC2_DefaultRole`。

```
GET http://169.254.169.254/latest/meta-data/iam/security-
credentials/EMR_EC2_DefaultRole
```

有关编写使用 IAM 角色的应用程序的更多信息，请参阅向 Amazon EC2 实例上运行的应用程序授予访问Amazon资源的权限。

有关临时安全凭证的更多信息，请参阅《使用临时安全凭证》指南中的使用临时安全凭证。

允许用户和组创建和修改角色

必须允许为集群创建、修改和指定角色的 IAM 委托人（用户和组，包括默认角色）执行以下操作。有关每个操作的详细信息，请参阅《IAM API 参考》https://docs.amazonaws.cn/IAM/latest/APIReference/API_Operations.html中的操作。

- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:AddRoleToInstanceProfile`
- `iam:ListRoles`
- `iam:GetPolicy`
- `iam:GetInstanceProfile`

- iam:GetPolicyVersion
- iam:AttachRolePolicy
- iam:PassRole

iam:PassRole 权限允许创建集群。其余权限允许创建默认角色。

有关向用户分配权限的信息，请参阅《IAM 用户指南》中的更改用户的权限。

Amazon EMR 基于身份的策略示例

默认情况下，用户和角色没有创建或修改 Amazon EMR 资源的权限。它们还无法使用 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 执行任务。IAM 管理员必须创建 IAM policy，以便为用户和角色授予权限以对所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的用户或组。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅 IAM 用户指南中的在 JSON 选项卡上创建策略。

主题

- Amazon EMR 的策略最佳实践
- 允许用户查看他们自己的权限
- Amazon EMR 托管式策略
- 对集群和 EMR Notebooks 进行基于标签的访问的 IAM policy
- 拒绝 ModifyInstanceGroup 操作
- Amazon EMR 身份和访问问题排查

Amazon EMR 的策略最佳实践

基于身份的策略非常强大。它们确定某个人是否可以创建、访问或删除您账户中的 Amazon EMR 资源。这些操作可能会使 Amazon 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用Amazon托管式策略 – 要快速开始使用 Amazon EMR，请使用Amazon托管式策略，为您的员工提供他们所需的权限。这些策略已在您的账户中提供，并由 Amazon 维护和更新。有关更多信息，请参阅《IAM 用户指南》中的开始使用Amazon托管式策略中的权限和Amazon EMR 托管式策略。

- 授予最低权限 – 创建自定义策略时，仅授予执行任务所需的许可。最开始只授予最低权限，然后根据需要授予其它权限。这样做比起一开始就授予过于宽松的权限而后再尝试收紧权限来说更为安全。有关更多信息，请参阅《IAM 用户指南》中的授予最低权限。
- 为敏感操作启用 MFA – 为增强安全性，要求用户使用多重身份验证（MFA）来访问敏感资源或 API 操作。要了解更多信息，请参阅 IAM 用户指南中的在 Amazon 中使用多重身份验证 (MFA)。
- 使用策略条件来增强安全性 – 在切实可行的范围内，定义基于身份的策略在哪些情况下允许访问资源。例如，您可编写条件来指定请求必须来自允许的 IP 地址范围。您也可以编写条件，以便仅允许指定日期或时间范围内的请求，或者要求使用 SSL 或 MFA。有关更多信息，请参阅 IAM 用户指南中的 IAM JSON 策略元素：条件。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上完成此操作或者以编程方式使用 Amazon CLI 或 Amazon API 所需的权限。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ViewOwnUserInfo",
         "Effect":"Allow",
         "Action":[
            "iam:GetUser",
            "iam:GetUserPolicy",
            "iam:ListAttachedUserPolicies",
            "iam:ListGroupsForUser",
            "iam:ListUserPolicies"
         ],
         "Resource":[
            "arn:aws:iam:::user/${aws:username}"
         ]
      },
      {
         "Sid":"NavigateInConsole",
         "Effect":"Allow",
         "Action":[
            "iam:GetGroupPolicy",
            "iam:GetPolicy",
            "iam:GetPolicyVersion",
            "iam:ListAttachedGroupPolicies",
```

```
            "iam:ListGroupPolicies",
            "iam:ListGroups",
            "iam:ListPolicies",
            "iam:ListPolicyVersions",
            "iam:ListUsers"
         ],
         "Resource":""
      }
   ]
}
```

Amazon EMR 托管式策略

授予对所需 Amazon EMR 操作的完全访问权限或只读访问权限的最简单方式是使用适用于 Amazon EMR 的 IAM 托管式策略。托管式策略提供了在权限要求发生更改时自动更新的好处。如果您使用内联策略，则可能出现导致权限错误的服务更改。

Amazon EMR 将弃用现有托管式策略（v1 策略）以支持新的托管策略（v2 策略）。新的托管式策略已缩小范围，以与Amazon最佳实践保持一致。弃用现有 v1 托管式策略之后，您将无法将这些策略附加到任何新的 IAM 角色或用户。现有角色和使用弃用策略的用户可以继续使用它们。v2 托管式策略使用标签限制访问。它们仅允许指定的 Amazon EMR 操作，并需要使用 EMR 特定密钥标记的集群资源。我们建议您在使用新的 v2 策略之前仔细阅读文档。

在 IAM 控制台的策略列表中，v1 策略将被标记为已弃用，并在其旁边显示一个警告图标。已弃用的策略具有以下特征：

- 该策略继续为所有当前附加的用户、组和角色工作。什么都不会中断。
- 该策略不能附加到任何新的用户、组或角色。该策略若与当前实体分离,则不能重新附加。
- 从所有当前实体分离 v1 策略后，该策略将不再可见且无法再使用。

下表总结了当前策略 (v1) 与 v2 策略之间的更改。

EMR 托管式策略更改

策略类型	策略名称	策略目的	对 v2 策略的更改
IAM 托管式策略，适用于通过附加的用	V1 策略（将被弃用）：AmazonEla	允许用户执行 EMR 操作的完全权限。包括	策略添加了一个先决条件，即用户必须将用户标记添加到资源

策略类型	策略名称	策略目的	对 v2 策略的更改
户、角色或组完全访问 EMR	**sticMapReduceFullAccess** V2（范围）策略名称：**AmazonEMRFullAccessPolicy_v2**	资源的 iam:PassRole 权限。	，然后才能使用此策略。请参阅标记资源以使用托管式策略。 iam:PassRole 操作需要将 iam:PassedToService 条件设置为指定的服务。默认情况下，不允许访问 Amazon EC2、Amazon S3 和其它服务。请参阅用于完全访问的 IAM 托管式策略（v2 托管默认策略）。
IAM 托管式策略，用于通过附加的用户、角色或组只读访问 EMR	V1 策略（将被弃用）：**AmazonElasticMapReduceReadOnlyAccess** V2（范围）策略名称：**AmazonEMRReadOnlyAccessPolicy_v2**	允许用户执行 Amazon EMR 操作的只读权限。	权限只允许指定的 elasticmapreduce 只读操作。默认情况下，不允许访问 Amazon S3。请参阅用于只读访问的 IAM 托管式策略（v2 托管默认策略）。

策略类型	策略名称	策略目的	对 v2 策略的更改
默认 EMR 服务角色和附加的托管式策略	角色名称：EMR_DefaultRole V1 策略（将被弃用）：AmazonElasticMapReduceRole（EMR 服务角色） v2（缩小范围）策略名称：**AmazonEMRServicePolicy_v2**	允许 Amazon EMR 在配置资源和执行服务级别操作时代表您调用其它Amazon服务。所有集群都需要该角色。	v2 服务角色和 v2 默认策略将替换已弃用的角色和策略。策略添加了一个先决条件，即用户必须将用户标记添加到资源，然后才能使用此策略。请参阅标记资源以使用托管式策略。请参阅 Amazon EMR 的服务角色（EMR 角色）。
集群 EC2 实例（EC2 实例配置文件）的服务角色	V1 策略（将被弃用）：EMR_EC2_DefaultRole（实例配置文件） 已弃用的策略名称：AmazonElasticMapReduceforEC2Role	允许在 EMR 集群上运行的应用程序访问其它Amazon资源，例如 Amazon S3。如果您运行处理来自 Amazon S3 的数据的 Apache Spark 任务，则策略需要允许访问此类资源。	默认角色和默认策略都将逐渐被弃用。没有替代Amazon默认托管角色或策略。您需要提供基于资源或基于身份的策略。这意味着，默认情况下，在 EMR 集群上运行的应用程序无权访问 Amazon S3 或其它资源，除非您手动将其添加到策略中。请参阅默认角色和托管式策略。

策略类型	策略名称	策略目的	对 v2 策略的更改
其它 EC2 服务角色策略	当前策略名称：AmazonElasticMapReduceforAutoScalingRole、AmazonElasticMapReduceEditorsRole、AmazonEMRCleanupPolicy	如果使用 Auto Scaling、Notebooks 或清理 EC2 资源，则需提供 EMR 访问其它 Amazon资源和执行操作所需的权限。	对 v2 没有更改。

保护 iam:PassRole

Amazon EMR 完全权限默认托管策略包含 `iam:PassRole` 安全配置，包括以下内容：

- `iam:PassRole` 权限仅适用于特定默认 Amazon EMR 角色。
- `iam:PassedToService` 条件允许您仅将策略用于指定的Amazon服务，例如 `elasticmapreduce.amazonaws.com` 和 `ec2.amazonaws.com`。

您可以在 IAM 控制台中查看 AmazonEMRFullAccessPolicy_v2 和 AmazonEMRServicePolicy_v2 策略的 JSON 版本。建议您使用 v2 托管式策略创建新的集群。

要创建自定义策略，建议您从托管式策略开始，然后根据您的需求编辑这些策略。

有关如何将策略附加到用户（主体）的信息，请参阅《IAM 用户指南》中的通过 Amazon Web Services Management Console 使用托管式策略。

标记资源以使用托管式策略

AmazonEMRServicePolicy_v2 和 AAmazonEMRFullAccessPolicy_v2 取决于对 Amazon EMR 预置或使用的资源的范围缩小访问权限。缩小范围是通过限制对那些具有预定义用户标签关联的资源的访问权限来实现的。当您使用这两个策略中的任何一个时，您必须在配置集群时传递预定义的用户标签 `for-use-with-amazon-emr-managed-policies = true`。然后，Amazon EMR 将自动传播该标签。此外，您必须将用户标签添加到以下部分中列出的资源。如果您使用 Amazon EMR 控制台启动集群，请参阅 使用 Amazon EMR 控制台启动具有 v2 托管式策略的集群的注意事项。

若要使用托管策略，在使用 CLI、SDK 或其他方法配置集群时传递用户标签 `for-use-with-amazon-emr-managed-policies = true`。

当您传递标签时，Amazon EMR 会将标签传播到它创建的私有子网 ENI、EC2 实例和 EBS 卷。Amazon EMR 还会自动为其创建的安全组添加标签。但是，如果您希望 Amazon EMR 与特定安全组一起启动，则必须对其进行标记。对于非 Amazon EMR 创建的资源，您必须向这些资源添加标签。例如，您必须标记 Amazon EC2 子网、EC2 安全组（如果不是由 Amazon EMR 创建）和 VPC（如果您希望 Amazon EMR 创建安全组）。要在 VPC 中启动具有 v2 托管式策略的集群，必须使用预定义的用户标签标记这些 VPC。请参阅使用 Amazon EMR 控制台启动具有 v2 托管式策略的集群的注意事项。

传播用户指定的标记

Amazon EMR 使用您在创建集群时指定的 Amazon EMR 标签来标记其创建的资源。Amazon EMR 将标签应用于它在集群生命周期内创建的资源。

Amazon EMR 传播以下资源的用户标签：

- 私有子网 ENI（服务访问弹性网络接口）
- EC2 实例
- EBS 卷
- EC2 启动模板

自动标记的安全组

Amazon EMR 使用 Amazon EMR 的 v2 托管式策略所需的标签 `for-use-with-amazon-emr-managed-policies` 标记它创建的 EC2 安全组，而无论您在创建集群命令中指定哪些标签。对于在引入 v2 托管式策略之前创建的安全组，Amazon EMR 不会自动标记。如果要对账户中已存在的默认安全组使用 v2 托管式策略，则需要使用 `for-use-with-amazon-emr-managed-policies = true` 手动标记安全组。

手动标记的集群资源

您必须手动标记一些集群资源，以便 Amazon EMR 默认角色可以访问。

- 您必须使用 Amazon EMR 托管式策略标签 `for-use-with-amazon-emr-managed-policies` 手动标记 EC2 安全组和 EC2 子网。
- 如果您希望 Amazon EMR 创建默认安全组，则必须手动标记 VPC。如果默认安全组不存在，EMR 将尝试使用特定标记创建安全组。

Amazon EMR 会自动为以下资源添加标签：

- EMR 创建的 EC2 安全组

您必须手动为以下资源添加标签：

- EC2 子网
- EC2 安全组

（可选）您可以手动为以下资源添加标签：

- VPC – 仅当您希望 Amazon EMR 创建安全组时

使用 Amazon EMR 控制台启动具有 v2 托管式策略的集群的注意事项

您可以使用 Amazon EMR 控制台预置具有 v2 托管式策略的集群。以下是您使用控制台启动 Amazon EMR 集群时的一些注意事项。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台。自动标记功能在新控制台中尚不可用，新控制台也不会向您显示哪些资源（VPC/子网）需要标记。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的更多信息。

- 您不需要传递预定义标签。Amazon EMR 会自动添加标签并将其传播到恰当组件。
- 对于需要手动标记的组件，如果您具有标记资源所需的权限，Amazon EMR 旧控制台会尝试自动标记这些组件。如果您没有标记资源的权限或如果您想使用新控制台，请让管理员标记这些资源。
- 除非满足所有先决条件，否则无法启动具有 v2 托管式策略的集群。
- Amazon EMR 旧控制台向您显示哪些资源（VPC/子网）需要标记。

用于完全访问的 IAM 托管式策略（v2 托管默认策略）

v2 范围 EMR 默认托管式策略向用户授予特定访问权限。它们需要预定义的 Amazon EMR 资源标签和 Amazon EMR 使用的资源的 `iam:PassRole` 条件键，例如用于启动集群的 `Subnet` 和 `SecurityGroup`。

要为 Amazon EMR 授予对必需操作的权限，可附加 `AmazonEMRFullAccessPolicy_v2` 托管式策略。此更新的默认托管式策略将替换 `AmazonElasticMapReduceFullAccess` 托管式策略。

AmazonEMRFullAccessPolicy_v2 取决于对 Amazon EMR 预置或使用的资源的范围缩小访问权限。使用此策略时，您需要在预置集群时传递用户标签 for-use-with-amazon-emr-managed-policies = true。Amazon EMR 将自动传播标签。此外，您可能需要手动向特定类型的资源添加用户标签，例如不是由 Amazon EMR 创建的 EC2 安全组。有关更多信息，请参阅标记资源以使用托管式策略。

AmazonEMRFullAccessPolicy_v2 策略通过执行以下操作来保护资源：

- 需要使用预定义的 Amazon EMR 托管式策略标签 for-use-with-amazon-emr-managed-policies 标记资源，以便创建集群和访问 Amazon EMR。
- 将 iam:PassRole 操作限制为特定的默认角色和对特定服务的 iam:PassedToService 访问权限。
- 默认情况下，不再提供对 Amazon EC2、Amazon S3 和其它服务的访问权限。

本策略的内容如下所示。

> **Note**
>
> 您还可以使用控制台链接 AmazonEMRFullAccessPolicy_v2 查看该策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RunJobFlowExplicitlyWithEMRManagedTag",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:RunJobFlow"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
                }
            }
        },
        {
            "Sid": "ElasticMapReduceActions",
            "Effect": "Allow",
```

```
            "Action": [
                "elasticmapreduce:AddInstanceFleet",
                "elasticmapreduce:AddInstanceGroups",
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:AddTags",
                "elasticmapreduce:CancelSteps",
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:CreateSecurityConfiguration",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:DeleteSecurityConfiguration",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:DescribeJobFlows",
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeReleaseLabel",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:GetManagedScalingPolicy",
                "elasticmapreduce:GetAutoTerminationPolicy",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListSupportedInstanceTypes",
                "elasticmapreduce:ModifyCluster",
                "elasticmapreduce:ModifyInstanceFleet",
                "elasticmapreduce:ModifyInstanceGroups",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:PutAutoScalingPolicy",
                "elasticmapreduce:PutBlockPublicAccessConfiguration",
                "elasticmapreduce:PutManagedScalingPolicy",
                "elasticmapreduce:RemoveAutoScalingPolicy",
                "elasticmapreduce:RemoveManagedScalingPolicy",
                "elasticmapreduce:RemoveTags",
                "elasticmapreduce:SetTerminationProtection",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:TerminateJobFlows",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "ViewMetricsInEMRConsole",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassRoleForElasticMapReduce",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/EMR_DefaultRole",
                "arn:aws:iam::*:role/EMR_DefaultRole_V2"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ec2.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForAutoScaling",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "application-autoscaling.amazonaws.com*"
                }
```

```
            }
        },
        {
            "Sid": "ElasticMapReduceServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
elasticmapreduce.amazonaws.com*/AWSServiceRoleForEMRCleanup*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "elasticmapreduce.amazonaws.com",
                        "elasticmapreduce.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Sid": "ConsoleUIActions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeImages",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNatGateways",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcEndpoints",
                "s3:ListAllMyBuckets",
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

用于完全访问的 IAM 托管式策略（即将弃用）

AmazonElasticMapReduceFullAccess 和 AmazonEMRFullAccessPolicy_v2 Amazon Identity and Access Management（IAM）托管式策略授予 Amazon EMR 和其他服务所需的所有操作的权限。

> **Important**
>
> AmazonElasticMapReduceFullAccess 托管式策略即将弃用，因此不再建议与 Amazon EMR 结合使用。请改用 AmazonEMRFullAccessPolicy_v2。当 IAM 服务最终弃用 v1 策略时，您将无法将其附加到角色。但是，您可以将现有角色附加到集群，即使该角色使用已弃用的策略。

Amazon EMR 完全权限默认托管策略包含 iam:PassRole 安全配置，包括以下内容：

- iam:PassRole 权限仅适用于特定默认 Amazon EMR 角色。
- iam:PassedToService 条件允许您仅将策略用于指定的Amazon服务，例如 elasticmapreduce.amazonaws.com 和 ec2.amazonaws.com。

您可以在 IAM 控制台中查看 AmazonEMRFullAccessPolicy_v2 和 AmazonEMRServicePolicy_v2 策略的 JSON 版本。建议您使用 v2 托管式策略创建新的集群。

您可以在 AmazonElasticMapReduceFullAccess 查看 Amazon Web Services Management Console 中已弃用的 v1 策略的内容。该策略中的 ec2:TerminateInstances 操作授予用户或角色终止与 IAM 账户关联的任何 Amazon EC2 实例的权限。这包括不属于 EMR 集群的实例。

用于只读访问的 IAM 托管式策略（v2 托管默认策略）。

要向 Amazon EMR 授予只读权限，可附加 AmazonEMRReadOnlyAccessPolicy_v2 托管式策略。此默认托管式策略将替换 AmazonElasticMapReduceReadOnlyAccess 托管式策略。下面的代码段显示了此策略声明的内容。与 AmazonElasticMapReduceReadOnlyAccess 策略相比，AmazonEMRReadOnlyAccessPolicy_v2 策略不使用 elasticmapreduce 元素的通配符。相反，默认的 v2 策略限定了允许的 elasticmapreduce 操作范围。

> **Note**
>
> 您还可以使用Amazon Web Services Management Console链接 AmazonEMRReadOnlyAccessPolicy_v2 查看该策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "ElasticMapReduceActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:DescribeJobFlows",
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeReleaseLabel",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:GetManagedScalingPolicy",
                "elasticmapreduce:GetAutoTerminationPolicy",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListSupportedInstanceTypes",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewMetricsInEMRConsole",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        }
    ]
}
```

用于只读访问的 IAM 托管式策略（即将弃用）

AmazonElasticMapReduceReadOnlyAccess 托管式策略即将弃用。启动新集群时，您无法附加此策略。AmazonElasticMapReduceReadOnlyAccess 已替换 AmazonEMRReadOnlyAccessPolicy_v2 作为 Amazon EMR 的默认托管式策略。下面的代码段显示了此策略声明的内容。elasticmapreduce 元素的通配符指定仅允许以指定字符串开头的操作。请记住，由于此策略不会明确拒绝操作，因此仍可使用其它策略声明来授予对指定操作的访问权限。

> **Note**
>
> 您也可以使用Amazon Web Services Management Console查看策略。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:Describe*",
        "elasticmapreduce:List*",
        "elasticmapreduce:ViewEventsFromAllClustersInConsole",
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sdb:Select",
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon EMR 的Amazon托管式策略

要向用户、组和角色添加权限，与自己编写策略相比，使用 Amazon 托管策略更简单。创建仅为团队提供所需权限的 IAM 客户托管策略需要时间和专业知识。要快速入门，您可以使用我们的 Amazon 托管策略。这些策略涵盖常见使用案例，可在您的 Amazon 账户中使用。有关 Amazon 托管策略的更多信息，请参阅 IAM 用户指南中的 Amazon 托管策略。

Amazon 服务负责维护和更新 Amazon 托管策略。您无法更改 Amazon 托管策略中的权限。服务偶尔会向 Amazon 托管式策略添加额外权限以支持新特征。此类更新会影响附加策略的所有身份（用户、组和角色）。当启动新特征或新操作可用时，服务最有可能会更新 Amazon 托管式策略。服务不会从 Amazon 托管式策略中删除权限，因此策略更新不会破坏您的现有权限。

此外，Amazon 还支持跨多种服务的工作职能的托管策略。例如，ReadOnlyAccess Amazon 托管策略提供对所有 Amazon 服务和资源的只读访问权限。当服务启动新特征时，Amazon 会为新操作和

资源添加只读权限。有关工作职能策略的列表和说明，请参阅 IAM 用户指南中的适用于工作职能的 Amazon 托管策略。

Amazon托管式策略的 Amazon EMR 更新

查看有关 Amazon EMR（自从其开始跟踪更新更改以来）的Amazon托管式策略的更新的详细信息。有关此页面更改的提示，请订阅 Amazon EMR 文档历史记录页面上的 RSS 源。

更改	说明	日期
AmazonEMRFullAccessPolicy_v2 和 **AmazonEMRReadOnlyAccessPolicy_v2** – 对现有策略的更新	增加了 `elasticmapreduce:ListSupportedInstanceTypes` 。	2023 年 7 月 13 日
AmazonEMRFullAccessPolicy_v2 和 **AmazonEMRReadOnlyAccessPolicy_v2** – 对现有策略的更新	添加了`elasticmapreduce:DescribeReleaseLabel` 和`elasticmapreduce:GetAutoTerminationPolicy` 。	2022 年 4 月 21 日
AmazonEMRFullAccessPolicy_v2 – 对现有策略的更新	为 使用自定义 AMI 添加了 `ec2:DescribeImages` 。	2022 年 2 月 15 日
Amazon EMR 托管式策略	更新以明确预定义用户标签的使用情况。 使用Amazon控制台添加的部分启动具有 v2 托管式策略的集群。	2021 年 9 月 29 日

更改	说明	日期
AmazonEMRFullAccessPolicy_v2 – 对现有策略的更新	更改了 PassRoleForAutoScaling 和 PassRoleForEC2 操作，以使用 StringLike 条件运算符分别匹配 "iam:PassedToService":"application-autoscaling.amazonaws.com*" 和 "iam:PassedToService":"ec2.amazonaws.com*" 。	2021 年 5 月 20 日
AmazonEMRFullAccessPolicy_v2 – 对现有策略的更新	删除了无效操作 s3:ListBuckets ，并将其替换为 s3:ListAllMyBuckets 操作。 更新了服务相关角色（SLR）创建方面的内容，以明确将范围缩小到 Amazon EMR 具有的唯一拥有明确服务委托人的 SLR。可以创建的 SLR 与此更改之前完全相同。	2021 年 3 月 23 日

更改	说明	日期
AmazonEMRFullAccessPolicy_v2 – 新策略	Amazon EMR 添加了新权限来确定对资源的访问范围，并添加了一个先决条件，即用户必须先将预定义的用户标签添加到资源中，然后才能使用 Amazon EMR 托管式策略。 `iam:PassRole` 操作需要将 `iam:PassedToService` 条件设置为指定服务。默认情况下，不允许访问 Amazon EC2、Amazon S3 和其它服务。	2021 年 3 月 11 日
AmazonEMRServicePolicy_v2 – 新策略	添加了一个先决条件，即用户必须将用户标记添加到资源，然后才能使用此策略。	2021 年 3 月 11 日
AmazonEMRReadOnlyAccessPolicy_v2 – 新策略	权限只允许指定的 elasticmapreduce 只读操作。默认情况下，不允许访问 Amazon S3。	2021 年 3 月 11 日
Amazon EMR 开始跟踪更改	Amazon EMR 开始跟踪其Amazon托管式策略的更改。	2021 年 3 月 11 日

对集群和 EMR Notebooks 进行基于标签的访问的 IAM policy

您可以在基于身份的策略中使用条件，以便基于标签控制对集群和 EMR Notebooks 的访问。

有关向集群添加标签的更多信息，请参阅为 EMR 集群添加标签。

以下示例说明将条件运算符与 Amazon EMR 条件键结合使用的不同场景和方法。这些 IAM policy 声明仅用于演示目的，并且不应用于生产环境。可通过多种方式来组合策略声明，以根据要求授予和拒绝权限。有关规划和测试 IAM policy 的更多信息，请参阅 IAM 用户指南。

> **Important**
>
> 一个重要注意事项是，应该明确拒绝添加标签操作的权限。这可以防止用户标记资源，从而为其授予您不打算授予的权限。如果您未拒绝资源的标记操作，用户可以修改标签并规避基于标签的策略意图。

适用于集群的基于身份的策略语句示例

以下示例演示基于身份的权限策略，用于控制允许对 EMR 集群执行的操作。

> **Important**
>
> Amazon EMR 中的 `ModifyInstanceGroup` 操作不需要您指定集群 ID。因此，基于集群标签拒绝此操作需要额外的考虑。有关更多信息，请参阅拒绝 ModifyInstanceGroup 操作。

主题

- 仅允许带特定标签值的集群上的操作
- 创建集群时需要集群标签
- 允许带有特定标签的集群上的操作，而不管标签值如何

仅允许带特定标签值的集群上的操作

以下示例演示了一个策略，该策略允许用户使用值 *dev* 根据集群标签 *department* 执行操作，并允许用户对集群添加相同的标签。最后一个策略示例演示如何拒绝向 EMR 集群添加除该相同标签之外的其它标签的权限。

在以下策略示例中，`StringEquals` 条件运算符尝试将 *dev* 与标签 *department* 的值匹配。如果标签 *department* 尚未添加到集群或不包含值 *dev*，则策略不会应用，并且此策略将不允许这些操作。如果任何其它策略声明允许操作，则用户只能使用标签中包含此值的集群。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt12345678901234",
      "Effect": "Allow",
      "Action": [
```

```
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:TerminateJobFlows",
        "elasticmapreduce:SetTerminationProtection",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:DescribeStep"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/department": "dev"
        }
      }
    }
  ]
}
```

您也可以使用条件运算符指定多个标签值。例如，要运行其 *department* 标签包含值 *dev* 或 *test* 的集群上的所有操作，您可以将上一个示例中的条件块替换为以下内容。

```
            "Condition": {
              "StringEquals": {
                "elasticmapreduce:ResourceTag/department":["dev", "test"]
              }
            }
```

创建集群时需要集群标签

在前面的示例中，以下示例策略将查找相同的匹配标签：*department* 标签的值 *dev*。但在此示例中，RequestTag 条件键指定策略在标签创建期间适用。因此，您必须使用与指定值匹配的标签创建集群。

要创建带有标签的集群，您还必须具有 elasticmapredue:AddTags 操作权限。对于此语句，elasticmapreduce:ResourceTag 条件键确保 IAM 仅授予对标签 *department* 上具有值 *dev* 的标签资源的访问权限。Resource 元素用于将此权限限制到集群资源。

对于 PassRole 资源，您必须在 PassRoleForEMR 语句中提供 Amazon 账户 ID 或别名、服务角色名称以及 PassRoleForEC2 语句中的实例配置文件名称。有关 IAM ARN 格式的信息，请参阅 《IAM 用户指南》中的 IAM ARN。

有关匹配的标签键值的更多信息，请参阅《IAM 用户指南》中的 aws:RequestTag/tag-key。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "RunJobFlowExplicitlyWithTag",
        "Effect": "Allow",
        "Action": [
          "elasticmapreduce:RunJobFlow"
        ],
        "Resource": "*",
        "Condition": {
          "StringEquals": {
            "aws:RequestTag/department": "dev"
          }
        }
      },
      {
        "Sid": "AddTagsForDevClusters",
        "Effect": "Allow",
        "Action": "elasticmapreduce:AddTags",
        "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
        "Condition": {
          "StringEquals": {
            "elasticmapreduce:ResourceTag/department": "dev"
          }
        }
      },
      {
        "Sid": "PassRoleForEMR",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
        "Condition": {
          "StringLike": {
            "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
          }
        }
```

```
      },
      {
        "Sid": "PassRoleForEC2",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
        "Condition": {
          "StringLike": {
            "iam:PassedToService": "ec2.amazonaws.com*"
          }
        }
      }
    ]
}
```

允许带有特定标签的集群上的操作，而不管标签值如何

您也可以仅允许具有特定标签的集群上的操作，而不管标签值如何。为此，您可以使用 `Null` 运算符。有关更多信息，请参阅《IAM 用户指南》中的用于检查条件键是否存在的条件运算符。例如，要仅允许包含 *`department`* 标签的 EMR 集群上的操作 (而不管此标签包含的值如何)。您可以将上一示例中的条件块替换为以下内容。`Null` 运算符可检查 EMR 集群上是否存在标签 *`department`*。如果此标签存在，则 `Null` 语句的计算结果为 false，这匹配此策略声明中指定的条件并允许相应的操作。

```
"Condition": {
  "Null": {
    "elasticmapreduce:ResourceTag/department":"false"
  }
}
```

以下策略声明仅允许用户在集群具有 *`department`* 标签 (可包含任何值) 的情况下创建 EMR 集群。针对`PassRole`资源，您需要提供Amazon账户 ID 或别名，以及服务角色名称。有关 IAM ARN 格式的信息，请参阅 《IAM 用户指南》中的 IAM ARN。

有关指定 null (false) 条件运营的更多信息，请参阅《IAM 用户指南》中的用于检查条件键是否存在的条件运算符。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "CreateClusterTagNullCondition",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:RunJobFlow"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/department": "false"
                }
            }
        },
        {
            "Sid": "AddTagsNullCondition",
            "Effect": "Allow",
            "Action": "elasticmapreduce:AddTags",
            "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
            "Condition": {
                "Null": {
                    "elasticmapreduce:ResourceTag/department": "false"
                }
            }
        },
        {
            "Sid": "PassRoleForElasticMapReduce",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ec2.amazonaws.com*"
```

```
                }
            }
        }
    ]
}
```

适用于 EMR Notebooks 的基于身份的策略语句示例

本部分中的示例 IAM policy 语句展示了使用密钥来限制使用 EMR Notebooks 所允许的操作的常见情形。只要与委托人（用户）关联的其它策略都不允许执行这些操作，条件上下文键就会按指示限制允许的操作。

Example – 仅允许访问用户基于标记创建的 EMR Notebooks

下面的示例策略语句附加到角色或用户后，会允许用户仅使用他们创建的笔记本。此策略语句使用创建笔记本时应用的默认标签。

在示例中，`StringEquals` 条件运算符尝试将表示当前用户用户 ID (`{aws:userId}`) 的变量与标签 `creatorUserID` 的值相匹配。如果标签 `creatorUserID` 尚未添加到笔记本或不包含当前用户 ID 的值，则策略不会应用，并且此策略将不允许这些操作。如果没有任何其它策略语句允许这些操作，则用户只能使用此标签中包含此值的笔记本。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
                }
            }
        }
    ]
```

```
}
```

Example – 创建笔记本时需要笔记本标记

此示例使用 RequestTag 上下文键。仅当用户未更改或删除 creatorUserID 标签时，才允许执行 CreateEditor 操作，此标签是默认添加的。变量 ${aws:userId} 用于指定当前活动用户的用户 ID，此 ID 是标签的默认值。

该策略语句可有助于确保用户不会删除 createUserId 标签或更改其值。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/creatorUserId": "${aws:userid}"
                }
            }
        }
    ]
}
```

此示例要求用户使用键字符串为 dept 且值设置为以下之一的标签创建集群：datascience、analytics、operations。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "elasticmapreduce:RequestTag/dept": [
                        "datascience",
                        "analytics",
                        "operations"
                    ]
                }
            }
        }
    ]
}
```

Example – 将笔记本创建限定到已标记的集群并需要笔记本标签

仅当使用键字符串 owner 设置为一个指定值的标签创建笔记本时，此示例才允许创建笔记本。此外，仅当集群具有键字符串 department 设置为一个指定值的标签时，才能创建笔记本。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/owner": [
                        "owner1",
                        "owner2",
                        "owner3"
                    ],
                    "elasticmapreduce:ResourceTag/department": [
                        "dep1",
                        "dep3"
                    ]
                }
            }
        }
    ]
}
```

Example – 将能力限定为启动基于标签的笔记本

此示例将启动笔记本的能力限定为那些具有键字符串 owner 设置为一个指定值的标签的笔记本。因为 Resource 元素仅用于指定 editor，所以此条件不适用于集群，并且它不需要标记。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "owner1",
                        "owner2"
                    ]
                }
            }
        }
    ]
}
```

此示例类似于上一示例。但是，限制仅适用于已标记的集群，不适用于笔记本。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "dep1",
                        "dep3"
                    ]
```

```
                }
            }
        }
    ]
}
```

此示例使用一组不同的笔记本和集群标签。仅当满足以下条件时，它才允许启动笔记本：

- 笔记本具有键字符串 owner 设置为任意指定值的标签

  –和–

- 集群具有键字符串 department 设置为任意指定值的标签

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "user1",
                        "user2"
                    ]
                }
            }
        },
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "datascience",
                        "analytics"
```

```
                    ]
                }
            }
        }
    ]
}
```

Example – 将能力限定为打开基于标签的笔记本编辑器

仅当满足以下条件时，此示例才允许打开笔记本编辑器：

- 笔记本具有键字符串 owner 设置为任意指定值的标签。

  –和–

- 集群具有键字符串 department 设置为任意指定值的标签。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "user1",
                        "user2"
                    ]
                }
            }
        },
        {
            "Action": [
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
```

```
                    "elasticmapreduce:ResourceTag/department": [
                        "datascience",
                        "analytics"
                    ]
                }
            }
        }
    ]
}
```

拒绝 ModifyInstanceGroup 操作

Amazon EMR 中的 ModifyInstanceGroups 操作不要求您在操作时提供集群 ID。相反，您只能指定实例组 ID。因此，基于集群 ID 或集群标签的简单拒绝策略可能不会产生预期效果。考虑以下示例策略。

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEFG67"
        }
    ]

}
```

如果附加了此策略的用户执行 ModifyInstanceGroup 操作并仅指定实例组 ID，则该策略不适用。由于允许在所有其它资源上执行该操作，因此操作成功。

此问题的解决方案是将策略声明附加到身份，该身份使用 NotResource 元素拒绝在没有集群 ID 的情况下发出的任何 ModifyInstanceGroup 操作。以下示例策略添加了这样的拒绝语句，因此除非指定了

集群 ID，否则任何 ModifyInstanceGroups 请求都会失败。由于身份必须在操作中指定集群 ID，因此基于集群 ID 的拒绝语句是有效的。

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEFG67"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "NotResource": "arn:*:elasticmapreduce:*:*:cluster/*"
        }
    ]

}
```

当您希望根据集群标签关联的值拒绝 ModifyInstanceGroups 操作时，也会出现类似问题。解决方案与之相似。除了添加指定标签值的拒绝语句之外，您还可以添加一个策略语句，在您指定的标签不存在时拒绝该 ModifyInstanceGroup 操作，无论值如何。

以下示例演示了一个策略，该策略在附加到身份时，会拒绝执行任何标签设置 department 设置为 dev 的集群上的任何 ModifyInstanceGroups 操作。这个语句之所以有效，是因为拒绝语句使用 StringNotLike 条件来拒绝操作，除非存在 department 标签。

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "dev"
                }
            },
            "Effect": "Deny",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:ResourceTag/department": "?*"
                }
            },
            "Effect": "Deny",
            "Resource": "*"
        }
    ],
}
```

Amazon EMR 身份和访问问题排查

您可以使用以下信息，帮助诊断和修复在使用 Amazon EMR 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 Amazon EMR 中执行操作
- 我无权执行 iam:PassRole

- 我希望允许我的 Amazon 账户以外的人访问我的 Amazon EMR 资源

我无权在 Amazon EMR 中执行操作

如果 Amazon Web Services Management Console 告诉您，您无权执行某个操作，则必须联系您的管理员寻求帮助。您的管理员是指为您提供用户名和密码的那个人。

当 `mateojackson` 用户尝试使用控制台查看有关虚构 *`my-example-widget`* 资源的详细信息，但不拥有虚构 `EMR:`*`GetWidget`* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 EMR:GetWidget on resource: my-example-widget
```

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 `EMR:`*`GetWidget`* 操作访问 *`my-example-widget`* 资源。

我无权执行 iam:PassRole

如果您收到一个错误，指明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 Amazon EMR。

有些 Amazon Web Services允许您将现有角色传递到该服务，而不是创建新服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 Amazon EMR 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 Amazon 管理员。管理员是向您提供登录凭证的人。

我希望允许我的 Amazon 账户以外的人访问我的 Amazon EMR 资源

您可以创建一个角色，以便其它账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 Amazon EMR 是否支持这些功能，请参阅 Amazon EMR 如何与 IAM 配合使用。
- 要了解如何为您拥有的 Amazon Web Services 账户 中的资源提供访问权限，请参阅 IAM 用户指南中的为您拥有的另一个 Amazon Web Services 账户 中的 IAM 用户提供访问权限。
- 要了解如何为第三方Amazon Web Services 账户提供您的资源的访问权限，请参阅 IAM 用户指南中的为第三方拥有的Amazon Web Services 账户提供访问权限。
- 要了解如何通过联合身份验证提供访问权限，请参阅 IAM 用户指南中的为经过外部身份验证的用户（联合身份验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅 IAM 用户指南中的 IAM 角色与基于资源的策略有何不同。

对 Amazon EMR 集群节点进行身份验证

SSH 客户端可以使用 Amazon EC2 密钥对来对集群实例进行身份验证。或者，使用 Amazon EMR 版本 5.10.0 或更高版本，可以配置 Kerberos 来对用户和到主节点的 SSH 连接进行身份验证。而且，在 Amazon EMR 5.12.0 及更高版本中，您可以使用 LDAP 进行身份验证。

主题

- 对 SSH 凭证使用 EC2 密钥对
- 使用 Kerberos 通过 Amazon EMR 进行身份验证
- 使用 Active Directory 或 LDAP 服务器通过 Amazon EMR 进行身份验证。

对 SSH 凭证使用 EC2 密钥对

Amazon EMR 集群节点在 Amazon EC2 实例上运行。您可以通过连接到 Amazon EC2 实例的相同方式连接到集群节点。您可以使用 Amazon EC2 创建密钥对，或者导入密钥对。在创建集群时，您可以指定用于通过 SSH 连接到所有集群实例的 Amazon EC2 密钥对。您也可以在没有密钥对的情况下创建集群。这通常适用于瞬态集群：这种集群启动、运行步骤、然后自动终止。

用于连接到集群的 SSH 客户端需要使用与此密钥对关联的私有密钥文件。这是用于使用 Linux、Unix 和 macOS 的 SSH 客户端的 .pem 文件。您必须设置权限，以便只有密钥所有者才具有访问该文件的权限。这是一个用于使用 Windows 的 SSH 客户端的 .ppk 文件，而此 .ppk 文件通常从 .pem 文件创建。

- 有关创建 Amazon EC2 密钥对的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 密钥对。

- 有关使用 PuTTYgen 从 .pem 文件创建 .ppk 文件的说明，请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的使用 PuTTYgen 转换您的私有密钥。
- 有关设置 .pem 文件权限和如何使用不同方法（包括通过 Linux 或 macOS 的 ssh，通过 Windows 的 PuTTY，或通过任何受支持的操作系统的 Amazon CLI）连接到 EMR 集群主节点 的更多信息，请参阅 使用 SSH 连接到主节点。

使用 Kerberos 通过 Amazon EMR 进行身份验证

Amazon EMR 5.10.0 及更高版本支持 Kerberos。Kerberos 是一种网络身份验证协议，使用私有密钥加密来提供强大的身份验证，以便不会通过网络以未加密的格式发送密码或其他凭证。

在 Kerberos 中，需要进行身份验证的服务和用户称为委托人。委托人存在于 Kerberos 领域 中。在该领域中，被称为密钥分发中心 (KDC) 的 Kerberos 服务器为委托人提供身份验证方法。KDC 通过颁发用于身份验证的票证 来执行此操作。KDC 维护一个包含其领域中的委托人、它们的密码及其它有关每个委托人的管理信息的数据库。KDC 还可以接受来自其它领域中的委托人的身份验证凭证，这称为跨领域信任。此外，EMR 集群可以使用外部 KDC 对委托人进行身份验证。

建立跨域信任或创建外部 KDC 的一种常见场景是从 Active Directory 域对用户进行身份验证。这允许用户在使用 SSH 连接到集群或者使用大数据应用程序时，通过其域账户访问 EMR 集群。

在使用 Kerberos 身份验证时，Amazon EMR 为集群上安装的应用程序、组件和子系统配置 Kerberos，以便它们可以相互进行身份验证。

> **Important**
>
> 在跨领域信任中或作为外部 KDC 时，Amazon EMR 不支持 Amazon Directory Service for Microsoft Active Directory。

在使用 Amazon EMR 配置 Kerberos 之前，我们建议您熟悉 Kerberos 概念、KDC 上运行的服务和用于管理 Kerberos 服务的工具。有关更多信息，请参阅 MIT Kerberos 文档，这由 Kerberos Consortium 发布。

主题

- 支持的应用程序
- Kerberos 架构选项
- 在 Amazon EMR 上配置 Kerberos
- 通过 SSH 连接使用 Kerberos 的集群

- 教程：配置集群专用的 KDC
- 教程：配置与 Active Directory 域的跨领域信任

支持的应用程序

在 EMR 集群中，Kerberos 委托人是所有集群节点上运行的大数据应用程序服务和子系统。Amazon EMR 可以将下面列出的应用程序和组件配置为使用 Kerberos。每个应用程序都与一个 Kerberos 用户委托人相关联。

Amazon EMR 不支持使用 Amazon Directory Service for Microsoft Active Directory 的跨领域信任。

Amazon EMR 只为下列应用程序和组件配置开源 Kerberos 身份验证功能。安装的任何其它应用程序不使用 Kerberos，这可能导致无法与使用 Kerberos 的组件通信并引发应用程序错误。不使用 Kerberos 的应用程序和组件没有启用身份验证。受支持的应用程序和组件可能因 Amazon EMR 版本而异。

Livy 用户界面是 Kerberized 的集群上托管的唯一 Web 用户界面。

- Hadoop MapReduce
- Hbase
- HCatalog
- HDFS
- Hive
  - 不要启用使用 LDAP 身份验证的 Hive。这可能会导致与使用 Kerberos 的 YARN 通信时出现问题。
- Hue
  - 不会自动设置 Hue 用户身份验证，但可以使用配置 API 进行配置。
  - Hue 服务器使用 Kerberos。未配置 Hue 前端 (UI) 的身份验证。可以为 Hue UI 配置 LDAP 身份验证。
- Livy
  - Amazon EMR 版本 5.22.0 及更高版本支持使用 Kerberos 集群进行 Livy 模拟。
- Oozie
- Phoenix
- Presto
  - Presto 在 Amazon EMR 6.9.0 及更高版本中支持 Kerberos 身份验证。

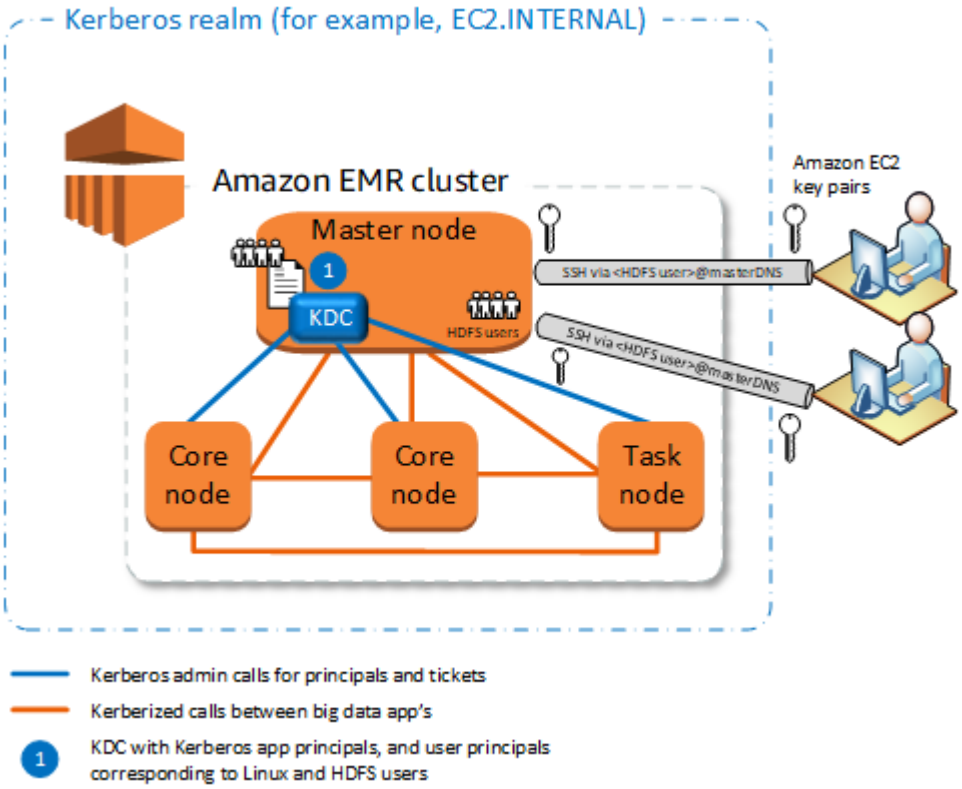
  - 要对 Presto 使用 Kerberos 身份验证，必须启用传输中加密。
- Spark
- Tez
- Trino
  - Trino 在 Amazon EMR 6.11.0 及更高版本中支持 Kerberos 身份验证。
  - 要对 Trino 使用 Kerberos 身份验证，必须启用传输中加密。
- YARN
- Zeppelin
  - Zeppelin 仅配置为将 Kerberos 与 Spark 解释器结合使用。它未配置其它解释器。
  - 除 Spark 之外，使用 Kerberos 的 Zeppelin 解释器不支持用户模拟。
- Zookeeper
  - 不支持 Zookeeper 客户端。

Kerberos 架构选项

在您将 Kerberos 与 Amazon EMR 结合使用时，可以从此部分中列出的架构进行选择。不论选择什么架构，您都使用相同步骤来配置 Kerberos。您需要创建安全配置，在创建集群时指定安全配置和兼容的集群特定 Kerberos 选项，并为集群上与 KDC 中集群委托人用户匹配的 Linux 用户创建 HDFS 目录。有关各个架构的配置选项及示例配置的说明，请参阅在 Amazon EMR 上配置 Kerberos。

集群专用 KDC（主节点上的 KDC）

此配置在 Amazon EMR 版本 5.10.0 及更高版本中提供。



优点

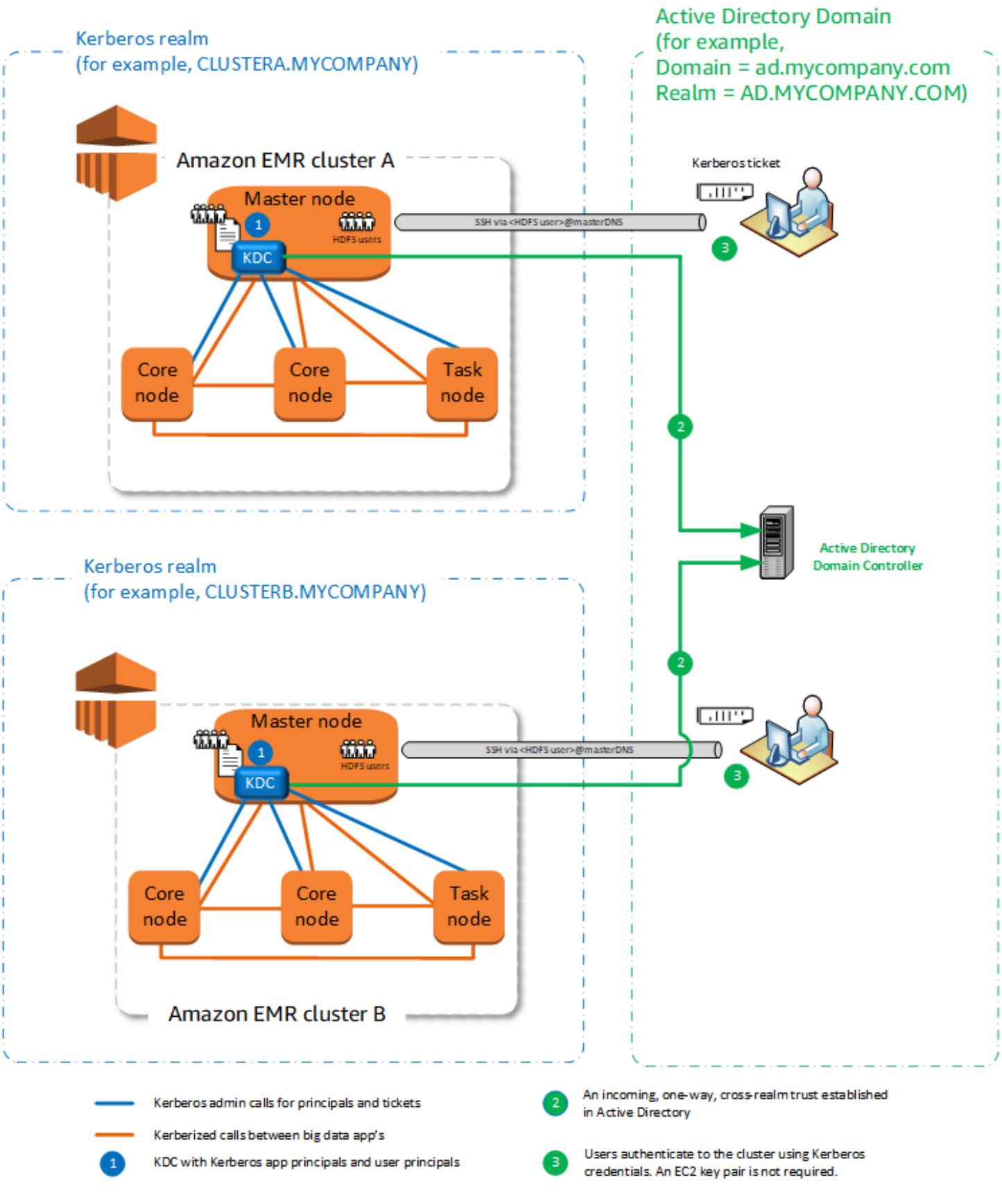
- Amazon EMR 具有 KDC 的完整所有权。
- EMR 集群上的 KDC 独立于集中 KDC 实现，例如 Microsoft Active Directory 或 Amazon Managed Microsoft AD。
- 性能影响降到最低，因为 KDC 仅对集群中的本地节点管理身份验证。
- （可选）其它使用 Kerberos 的集群可以引用 KDC 作为外部 KDC。有关更多信息，请参阅外部 KDC – 不同集群上的主节点。

注意事项和限制

- 使用 Kerberos 的集群无法彼此进行身份验证，因此应用程序无法互操作。如果集群应用程序需要互操作，您必须在集群之间建立跨领域信任，或者将一个集群设置为另一个集群的外部 KDC。如果建立了跨领域信任，则 KDC 必须具有不同的 Kerberos 领域。
- 在主节点的 EC2 实例上，您必须创建与 KDC 用户主体相对应的 Linux 用户，以及各个用户的 HDFS 目录。
- 用户委托人必须使用 EC2 私有密钥文件和 `kinit` 凭证以通过 SSH 连接到集群。

跨领域信任

在此配置中，来自不同 Kerberos 领域的委托人（通产是用户）对使用 Kerberos 的 EMR 集群（具有自己的 KDC）上的应用程序组件进行身份验证。主节点上的 KDC 与另一个 KDC 建立信任关系时所使用的是这两个 KDC 中都存在的跨领域主体。各个 KDC 中的委托人名称和密码完全匹配。跨领域信任在 Active Directory 实现中最常见，如下图所示。还支持与外部 MIT KDC 或其它 Amazon EMR 集群上的 KDC 的跨领域信任。


Active Directory Domain
(for example,
Domain = ad.mycompany.com
Realm = AD.MYCOMPANY.COM)
Kerberos realm
(for example, CLUSTERA.MYCOMPANY)
Amazon EMR cluster A
Master node
1
KDC
HDFS users
SSH via <HDFS user>@masterDNS
Kerberos ticket
3
2
Core node
Core node
Task node
Active Directory Domain Controller
Kerberos realm
(for example, CLUSTERB.MYCOMPANY)
Master node
1
KDC
HDFS users
SSH via <HDFS user>@masterDNS
2
3
Core node
Core node
Task node
Amazon EMR cluster B
Kerberos admin calls for principals and tickets
Kerberized calls between big data app's
1
KDC with Kerberos app principals and user principals
2
An incoming, one-way, cross-realm trust established in Active Directory
3
Users authenticate to the cluster using Kerberos credentials. An EC2 key pair is not required.

优点

- 安装 KDC 的 EMR 集群上维护对 KDC 的完整所有权。
- 使用 Active Directory，Amazon EMR 自动创建与来自 KDC 的用户委托人相对应的 Linux 用户。您仍必须为每个用户创建 HDFS 目录。此外，Active Directory 域中的用户委托人可以使用 `kinit` 凭证，访问使用 Kerberos 的集群，而无需 EC2 私有密钥文件。这消除了在集群用户之间共享私有密钥文件的需求。
- 由于各个集群 KDC 管理集群中节点的身份验证，将大量跨集群节点的网络延迟和处理开销的影响减到了最小。

注意事项和限制

- 如果您使用 Active Directory 领域建立信任，则必须提供 Active Directory 用户名和密码，在您在创建集群时，该用户应有权将委托人加入域。
- 无法在具有相同名称的 Kerberos 领域之间建立跨领域信任。
- 跨领域信任必须明确建立。例如，如果集群 A 和集群 B 均与 KDC 建立了跨领域信任，则不会继承彼此的信任，并且其应用程序无法彼此进行身份验证来实现互操作。
- KDC 必须独立维护并保持协调性，以便完全匹配用户委托人的凭证。

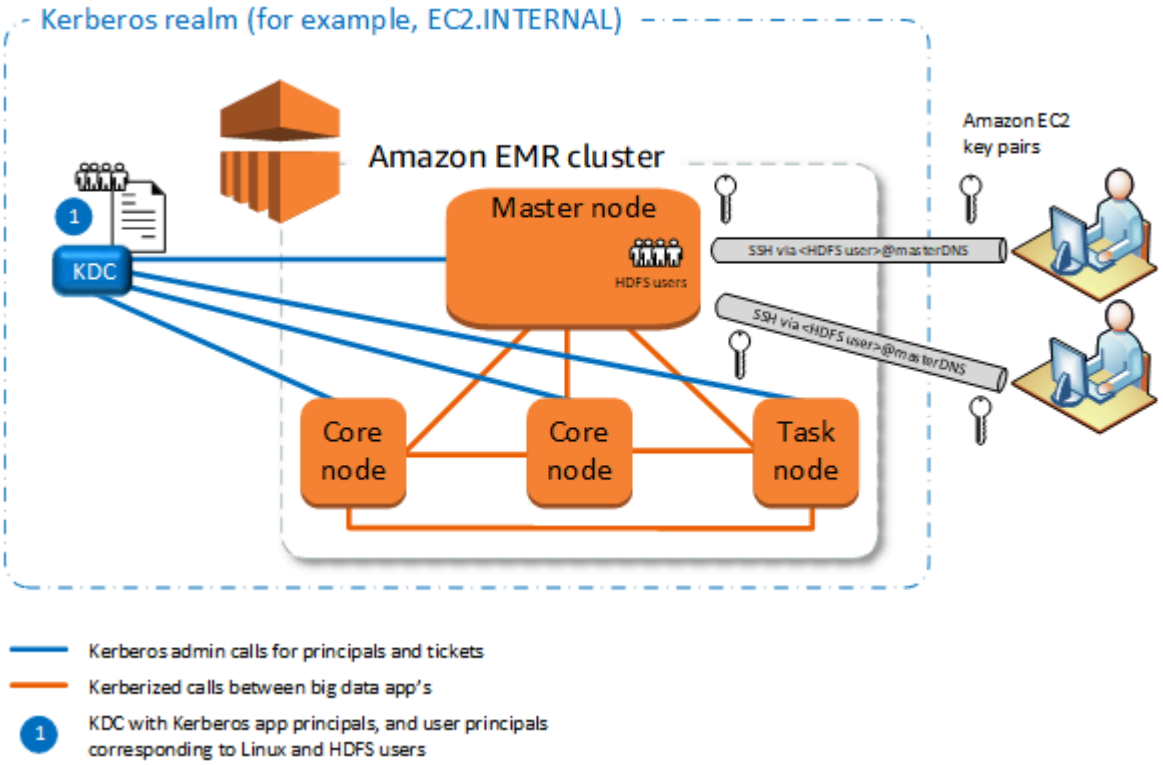
外部 KDC

Amazon EMR 5.20.0 及更高版本支持具有外部 KDC 的配置。

- 外部 KDC – 麻省理工学院（MIT）KDC
- 外部 KDC – 不同集群上的主节点
- 外部 KDC – 不同集群上的集群 KDC 具有 Active Directory 跨领域信任

外部 KDC – 麻省理工学院（MIT）KDC

此配置允许一个或多个 EMR 集群使用在 MIT KDC 服务器中定义和维护的委托人。



优点

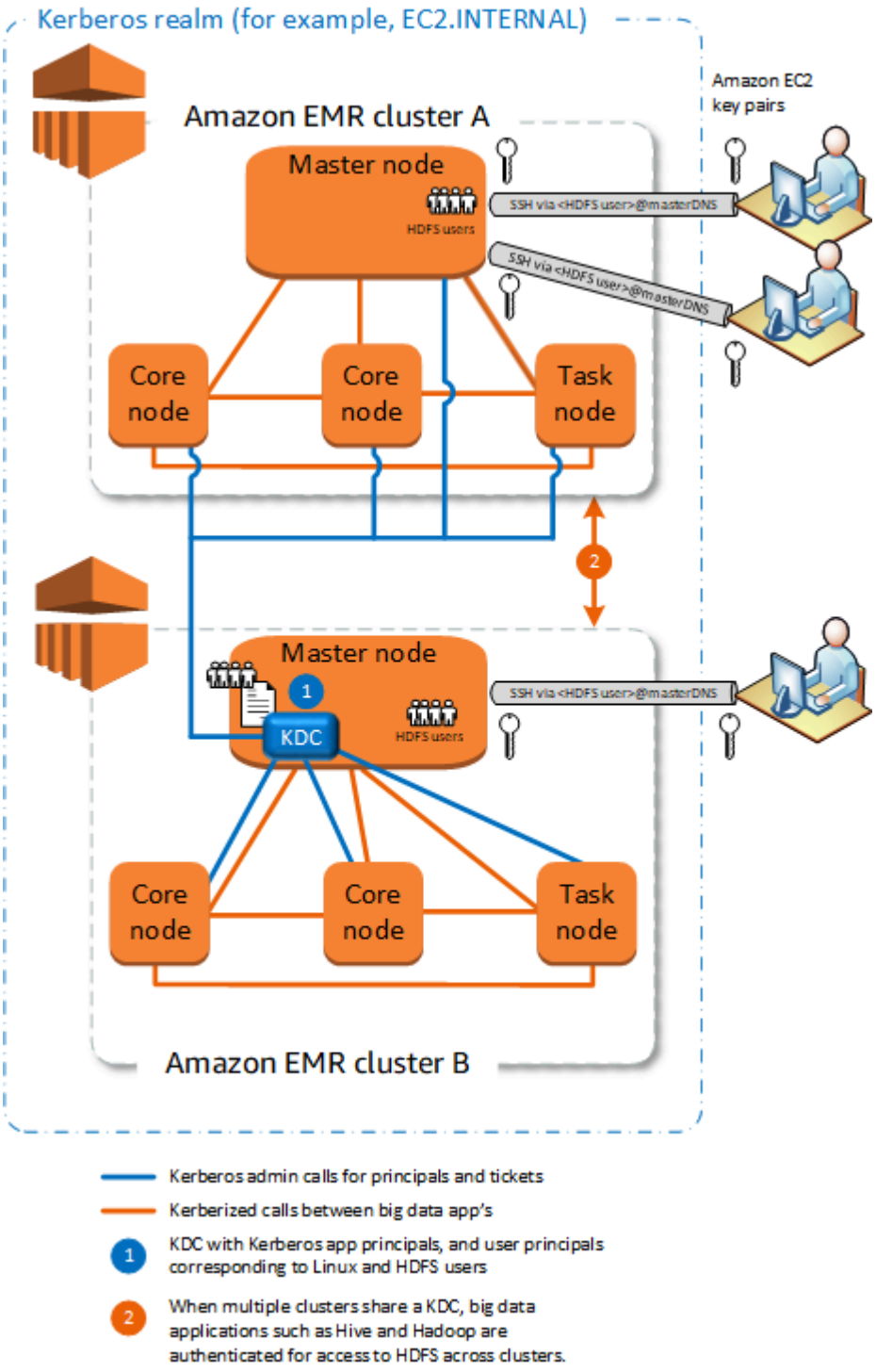
- 在单个 KDC 中，委托人的管理进行了整合。
- 在同一个 Kerberos 领域中，多个集群可以使用相同的 KDC。有关更多信息，请参阅使用具有相同 KDC 的多个集群的要求。
- 使用 Kerberos 的集群上的主节点没有与维护 KDC 相关的性能负担。

注意事项和限制

- 在各个使用 Kerberos 集群主节点的 EC2 实例上，您必须创建与 KDC 用户主体相对应的 Linux 用户，以及各个用户的 HDFS 目录。
- 用户委托人必须使用 EC2 私有密钥文件和 `kinit` 凭证以通过 SSH 连接到使用 Kerberos 的集群。
- 使用 Kerberos 的 EMR 集群中的每个节点都必须具有指向 KDC 的网络路由。
- 使用 Kerberos 的集群中的每个节点对外部 DKC 都会施加身份验证负担，因此 KDC 的配置会影响集群性能。在您配置 KDC 服务器的硬件时，请考虑同时支持的最大 Amazon EMR 节点数。
- 集群性能依赖于使用 Kerberos 的集群中节点与 KDC 之间的网络延迟。
- 由于相互依赖关系，问题排查可能会更加困难。

外部 KDC – 不同集群上的主节点

此配置几乎与上述外部 MIT KDC 实现相同，除了 KDC 位于 EMR 集群的主节点上之外。有关更多信息，请参阅 集群专用 KDC（主节点上的 KDC） 和 教程：配置与 Active Directory 域的跨领域信任：



优点

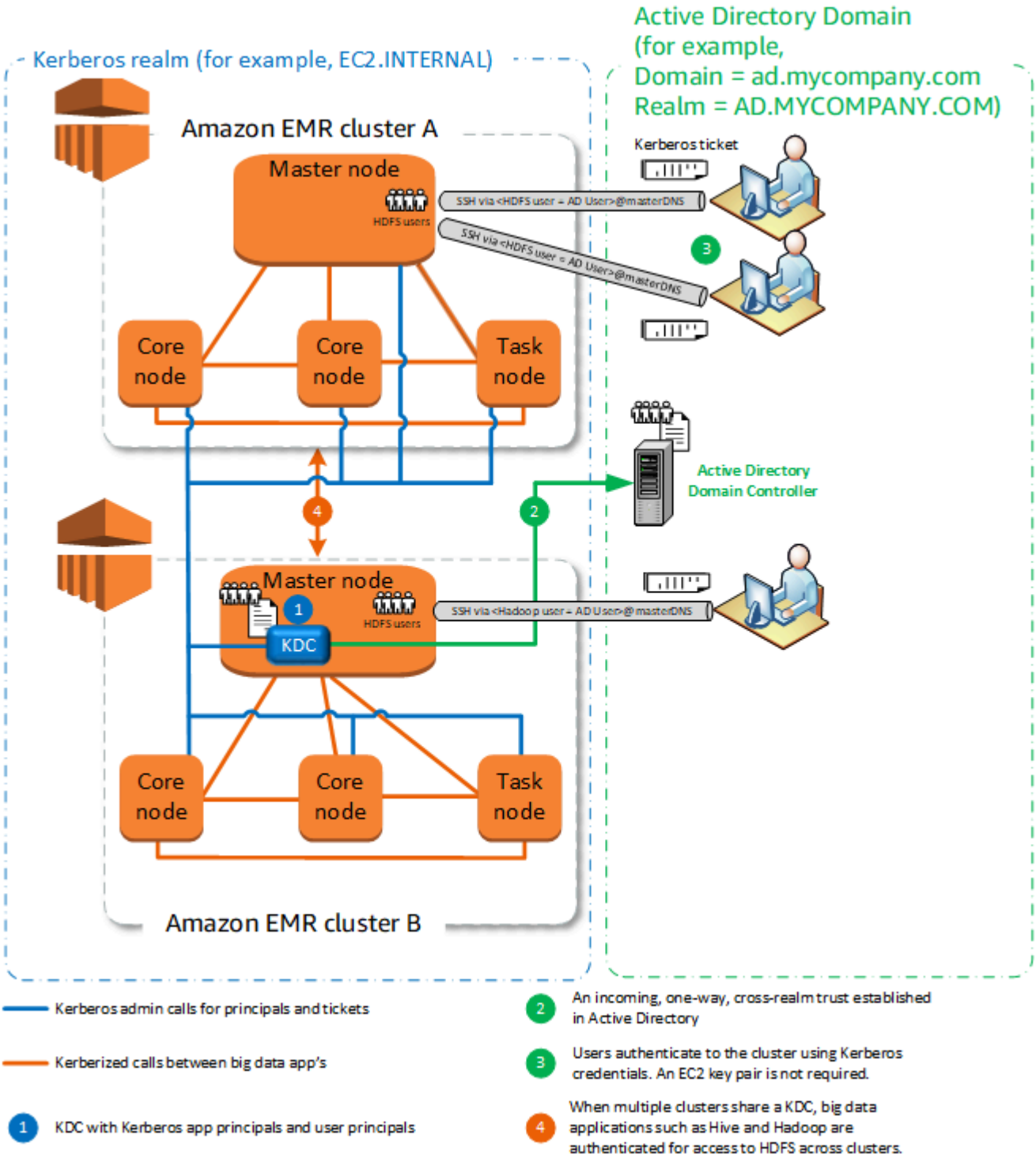
- 在单个 KDC 中，委托人的管理进行了整合。
- 在同一个 Kerberos 领域中，多个集群可以使用相同的 KDC。有关更多信息，请参阅使用具有相同 KDC 的多个集群的要求。

注意事项和限制

- 在各个使用 Kerberos 集群主节点的 EC2 实例上，您必须创建与 KDC 用户主体相对应的 Linux 用户，以及各个用户的 HDFS 目录。
- 用户委托人必须使用 EC2 私有密钥文件和 `kinit` 凭证以通过 SSH 连接到使用 Kerberos 的集群。
- 各个 EMR 集群中的各个节点必须具有指向 KDC 的网络路由。
- 使用 Kerberos 的集群中的每个 Amazon EMR 节点都会对外部 DKC 施加身份验证负担，因此 KDC 的配置会影响集群性能。在您配置 KDC 服务器的硬件时，请考虑同时支持的最大 Amazon EMR 节点数。
- 集群性能依赖于集群中节点与 KDC 之间的网络延迟。
- 由于相互依赖关系，问题排查可能会更加困难。

外部 KDC – 不同集群上的集群 KDC 具有 Active Directory 跨领域信任

在此配置中，您首先要创建一个集群，该集群具有与 Active Directory 存在单向跨领域信任关系的集群专用 KDC。有关详细教程，请参阅教程：配置与 Active Directory 域的跨领域信任。然后，您启动其它集群，引用具有作为外部 KDC 信任关系的集群 KDC。有关示例，请参阅 外部集群 KDC 与 Active Directory 跨领域信任。这允许使用外部 KDC 的每个 Amazon EMR 集群对在 Microsoft Active Directory 域中定义和维护的委托人进行身份验证。



优点

- 在 Active Directory 域中，委托人的管理进行了整合。
- Amazon EMR 加入 Active Directory 领域，这使您无需创建与 Active Directory 用户对应的 Linux 用户。您仍必须为每个用户创建 HDFS 目录。

- 在同一个 Kerberos 领域中，多个集群可以使用相同的 KDC。有关更多信息，请参阅使用具有相同 KDC 的多个集群的要求。
- Active Directory 域中的用户委托人可以通过 `kinit` 凭证访问使用 Kerberos 的集群，而无需 EC2 私有密钥文件。这消除了在集群用户之间共享私有密钥文件的需求。
- 只有一个 Amazon EMR 主节点承担维护 KDC 的负担，并且仅该集群必须通过 Active Directory 凭证创建以实现 KDC 与 Active Directory 之间的跨领域信任。

注意事项和限制

- 每个 EMR 集群中的每个节点都必须具有指向 KDC 和 Active Directory 域控制器的网络路由。
- 每个 Amazon EMR 节点都会对外部 DKC 施加身份验证负担，因此 KDC 的配置会影响集群性能。在您配置 KDC 服务器的硬件时，请考虑同时支持的最大 Amazon EMR 节点数。
- 集群性能依赖于集群中节点与 KDC 服务器之间的网络延迟。
- 由于相互依赖关系，问题排查可能会更加困难。

使用具有相同 KDC 的多个集群的要求

在同一个 Kerberos 领域中，多个集群可以使用相同的 KDC。但是，如果集群并发运行，则如果集群使用的 Kerberos ServicePrincipal 名称存在冲突，则集群可能会失败。

如果您有多个具有相同外部 KDC 的并发集群，请确保这些集群使用不同的 Kerberos 领域。如果集群必须使用相同的 Kerberos 领域，请确保集群位于不同的子网中，并且它们的 CIDR 范围不重叠。

在 Amazon EMR 上配置 Kerberos

此部分提供了设置常见架构的 Kerberos 的配置详细信息和示例。无论您选择哪种架构，配置基本上相同，通过三个步骤完成。如果您使用外部 KDC 或设置跨领域信任，则必须确保集群中的每个节点具有指向外部 KDC 的网络路由，包括适用的安全组的配置以允许入站和出站 Kerberos 流量。

步骤 1：使用 Kerberos 属性创建安全配置

安全配置指定有关 Kerberos KDC 的详细信息，并允许在每次创建集群时重用 Kerberos 配置。您可使用 Amazon EMR 控制台、Amazon CLI 或 EMR API 创建安全配置。安全配置也可以包含其它安全选项，如加密。有关创建安全配置以及在创建集群时指定安全配置的更多信息，请参阅使用安全配置设置集群安全性。有关安全配置中 Kerberos 属性的信息，请参阅安全配置的 Kerberos 设置。

步骤 2：创建集群并指定特定于集群的 Kerberos 属性

在创建集群时，您将指定 Kerberos 安全配置以及集群特定 Kerberos 选项。当您使用 Amazon EMR 控制台时，只有与指定安全配置兼容的 Kerberos 选项可用。当您使用 Amazon CLI 或 Amazon EMR API 时，请确保指定与所指示安全配置兼容的 Kerberos 选项。例如，如果在使用 CLI 创建集群时，您为跨领域信任指定了委托人密码，然而指定的安全配置没有配置跨领域信任参数，则会出错。有关更多信息，请参阅集群的 Kerberos 设置。

步骤 3：配置集群主节点

根据架构和实现的要求，集群上可能会需要额外的设置。您可在创建之后进行这些设置，也可在创建过程使用步骤或引导操作。

对于使用 SSH 连接到集群的每个通过 Kerberos 进行身份验证的用户，请确保创建对应于 Kerberos 用户的 Linux 账户。如果用户主体由 Active Directory 域控制器作为 KDC 或通过跨领域信任提供，Amazon EMR 将自动创建 Linux 账户。如果未使用 Active Directory，则您必须为对应于其 Linux 用户的每个用户创建委托人。有关更多信息，请参阅为通过 Kerberos 进行身份验证的 HDFS 用户和 SSH 连接配置集群。

每个用户还必须具有自己的 HDFS 用户目录，您必须创建该目录。此外，SSH 必须配置为启用 GSSAPI，以允许来自通过 Kerberos 进行身份验证的用户的连接。主节点上必须启用 GSSAPI，并且客户端 SSH 应用程序必须配置为使用 GSSAPI。有关更多信息，请参阅为通过 Kerberos 进行身份验证的 HDFS 用户和 SSH 连接配置集群。

Amazon EMR 上 Kerberos 的安全配置和集群设置

当您创建使用 Kerberos 的集群时，您可以指定安全配置以及特定于集群的 Kerberos 属性。您不能指定一组属性而不指定另一组，否则会发生错误。

本主题提供了在您创建安全配置和集群时可用于 Kerberos 的配置参数概览。此外，针对常见架构提供了创建兼容安全配置和集群的 CLI 示例。

安全配置的 Kerberos 设置

您可以使用 Amazon EMR 控制台、Amazon CLI 或 EMR API 创建指定 Kerberos 属性的安全配置。安全配置也可以包含其它安全选项，如加密。有关更多信息，请参阅创建安全配置。

使用以下参考来了解您选择的 Kerberos 架构的可用安全配置设置。此时将显示 Amazon EMR 控制台设置。有关对应的 CLI 选项，请参阅使用 Amazon CLI 指定 Kerberos 设置或配置示例。

参数		描述
Kerberos		指定为使用此安全配置的集群启用 Kerberos。如果集群使用此安全配置，则集群还必须指定 Kerberos 设置，否则会发生错误。
Provider	集群专用 KDC	指定 Amazon EMR 在使用此安全配置的任何集群的主节点上创建 KDC。您可以在创建集群时指定领域名称和 KDC 管理员密码。 如果需要，您可以从其它集群引用此 KDC。使用不同的安全配置创建这些集群，指定外部 KDC，并使用您为集群专用 KDC 指定的领域名称和 KDC 管理员密码。
	外部 KDC	仅在 Amazon EMR 5.20.0 版及更高版本中可用。指定使用此安全配置的集群使用集群外部的 KDC 服务器对 Kerberos 主体进行身份验证。未在集群上创建 KDC。创建集群时，您需要为外部 KDC 指定领域名称和 KDC 管理员密码。
票证生命周期		可选。指定 KDC 颁发的 Kerberos 票证在使用此安全配置的集群上有效的期间。 出于安全考虑，限制票证生命周期。集群应用程序和服务在过期后自动续订票证。使用 Kerberos 凭证通过 SSH 连接到集群的用户在票证过期后需要通过主节点命令行运行 `kinit` 来续订。
跨领域信任		指定使用此安全配置的集群上的集群专用 KDC 与不同 Kerberos 领域中的 KDC 之间的跨领域信任。 来自另一个领域的委托人（通常是用户）将通过使用此配置的集群进行身份验证。需要在其它 Kerberos 领域中进行其它配置。有关更多信息，请参阅教程：配置与 Active Directory 域的跨领域信任。

参数		描述
跨领域信任属性	领域	指定信任关系中其它领域的 Kerberos 领域名称。按照惯例，Kerberos 领域名称与域名相同，但全部使用大写字母。
	域	指定信任关系中其它领域的域名。
	管理服务器	指定信任关系的其它领域中的管理服务器的完全限定域名（FQDN）或 IP 地址。管理服务器和 KDC 服务器通常运行在具有相同 FQDN 的同一台计算机上，但通过不同的端口进行通信。 如果未指定端口，则使用端口 749，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com:749`)。
	KDC 服务器	指定信任关系的其它领域中 KDC 服务器的完全限定域名（FQDN）或 IP 地址。KDC 服务器和管理服务器通常运行在具有相同 FQDN 的同一台计算机上，但是使用不同的端口。 如果未指定端口，则使用端口 88，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com:88`)。
外部 KDC		指定集群使用集群外部 KDC。
外部 KDC 属性	管理服务器	指定外部管理服务器的完全限定域名（FQDN）或 IP 地址。管理服务器和 KDC 服务器通常运行在具有相同 FQDN 的同一台计算机上，但通过不同的端口进行通信。 如果未指定端口，则使用端口 749，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com:749`)。

参数			描述
	KDC 服务器		指定外部 KDC 服务器的完全限定域名（FQDN）。KDC 服务器和管理服务器通常运行在具有相同 FQDN 的同一台计算机上，但是使用不同的端口。 如果未指定端口，则使用端口 88，这是 Kerberos 默认值。另外，您还可以指定端口 (例如，`domain.example.com :88`)。
	Active Directory 集成		指定 Kerberos 委托人身份验证与 Microsoft Active Directory 域集成。
	Active Directory 集成属性	Active Directory 领域	指定 Active Directory 域的 Kerberos 领域名称。按照惯例，Kerberos 领域名称通常与域名相同，但全部使用大写字母。
		Active Directory 域	指定 Active Directory 域名。
		Active Directory 服务器	指定 Microsoft Active Directory 域控制器的完全限定域名（FQDN）。

集群的 Kerberos 设置

在使用 Amazon EMR 控制台、Amazon CLI 或 EMR API 创建集群时，您可以指定 Kerberos 设置。

使用以下参考来了解您选择的 Kerberos 架构的可用集群配置设置。此时将显示 Amazon EMR 控制台设置。有关对应的 CLI 选项，请参阅配置示例。

参数	描述
领域	集群的 Kerberos 领域名称。Kerberos 约定是将此项设置为与域名相同，但使用大写字母。例如，对于域 `ec2.internal` ，使用 `EC2.INTERNAL` 作为领域名称。

参数	描述
KDC 管理员密码	集群中为 `kadmin` 或 `kadmin.local` 使用的密码。这些是 Kerberos V5 管理系统的命令行接口，该系统保存了集群的 Kerberos 委托人、密码策略和 keytab。
跨领域信任委托人密码 (可选)	在建立跨领域信任时是必需的。跨领域主体密码，这必须跨领域相同。使用强密码。
Active Directory 域加入用户（可选）	在跨领域信任中使用 Active Directory 时必需。这是 Active Directory 账户的用户登录名，该账户具有将计算机加入域的权限。Amazon EMR 使用此身份将集群加入域。有关更多信息，请参阅the section called “步骤 3：将用户账户添加到 EMR 集群的域中”。
Active Directory 域加入密码（可选）	Active Directory 域加入用户的密码。有关更多信息，请参阅the section called “步骤 3：将用户账户添加到 EMR 集群的域中”。

配置示例

以下示例演示了常见场景的安全配置和集群配置。为简洁起见，显示的是 Amazon CLI 命令。

本地 KDC

以下命令创建在主节点上运行集群专用 KDC 的集群。集群上需要额外的配置。有关更多信息，请参阅为通过 Kerberos 进行身份验证的 HDFS 用户和 SSH 连接配置集群。

创建安全配置

```
aws emr create-security-configuration --name LocalKDCSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc",\
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24 }}}}'
```

创建集群

```
aws emr create-cluster --release-label emr-5.36.1 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive --ec2-attributes
 InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole \
--security-configuration LocalKDCSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyPassword
```

集群专用 KDC 与 Active Directory 跨领域信任

以下命令创建在主节点上运行集群专用 KDC 的集群，并与 Active Directory 域具有跨领域信任关系。在集群上和 Active Directory 中需要额外的设置。有关更多信息，请参阅教程：配置与 Active Directory 域的跨领域信任。

创建安全配置

```
aws emr create-security-configuration --name LocalKDCWithADTrustSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc", \
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24, \
"CrossRealmTrustConfiguration": {"Realm":"AD.DOMAIN.COM", \
"Domain":"ad.domain.com", "AdminServer":"ad.domain.com", \
"KdcServer":"ad.domain.com"}}}}}'
```

创建集群

```
aws emr create-cluster --release-label emr-5.36.1 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration KDCWithADTrustSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyClusterKDCAdminPassword,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
CrossRealmTrustPrincipalPassword=MatchADTrustPassword
```

不同集群上的外部 KDC

以下命令创建引用其它集群的主节点上集群专用 KDC 的集群，用于对主体进行身份验证。集群上需要额外的配置。有关更多信息，请参阅为通过 Kerberos 进行身份验证的 HDFS 用户和 SSH 连接配置集群。

创建安全配置

```
aws emr create-security-configuration --name ExtKDCOnDifferentCluster \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSOfKDCMaster:749", \
"KdcServer": "MasterDNSOfKDCMaster:88"}}}}'
```

创建集群

```
aws emr create-cluster --release-label emr-5.36.1 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCOnDifferentCluster \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword
```

外部集群 KDC 与 Active Directory 跨领域信任

以下命令创建没有 KDC 的集群。集群引用运行在其它集群的主节点上的集群专用 KDC，用于对主体进行身份验证。该 KDC 具有与 Active Directory 域控制器的跨领域信任关系。具有 KDC 的主节点上需要额外的配置。有关更多信息，请参阅教程：配置与 Active Directory 域的跨领域信任。

创建安全配置

```
aws emr create-security-configuration --name ExtKDCWithADIntegration \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSofClusterKDC:749", \
"KdcServer": "MasterDNSofClusterKDC.com:88", \
"AdIntegrationConfiguration": {"AdRealm":"AD.DOMAIN.COM", \
"AdDomain":"ad.domain.com", \
"AdServer":"ad.domain.com"}}}}}'
```

创建集群

```
aws emr create-cluster --release-label emr-5.36.1 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
```

```
--service-role EMR_DefaultRole --security-configuration ExtKDCWithADIntegration \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword,\
ADDomainJoinUser=MyPrivilegedADUserName,ADDomainJoinPassword=PasswordForADDomainJoinUser
```

为通过 Kerberos 进行身份验证的 HDFS 用户和 SSH 连接配置集群

Amazon EMR 为集群上运行的应用程序创建已通过 Kerberos 身份验证的客户端，例如，hadoop 用户、spark 用户等。您还可以添加已通过使用 Kerberos 的集群过程的身份验证的用户。然后，已通过身份验证的用户可以使用其 Kerberos 凭证连接到集群并使用应用程序。对于用户向集群进行身份验证，需要以下配置：

- 集群上必须有与 KDC 中 Kerberos 主体匹配的 Linux 账户。Amazon EMR 在与 Active Directory 集成的架构中自动完成此操作。
- 您必须在主节点上为每位用户创建 HDFS 用户目录，并向目录提供用户权限。
- 您必须配置 SSH 服务，以在主节点上启用 GSSAPI。此外，用户必须具有启用了 GSSAPI 的 SSH 客户端。

将 Linux 用户和 Kerberos 主体添加到主节点

如果您不使用 Active Directory，则必须在集群主节点上创建 Linux 账户，并将这些 Linux 用户的主体添加到 KDC。这包括主节点的 KDC 中的主体。除了用户主体之外，运行在主节点上的 KDC 需要本地主机的主体。

当架构包含 Active Directory 集成时，如果适用，将自动创建本地 KDC 上的 Linux 用户和委托人。您可以跳过此步骤。有关更多信息，请参阅跨领域信任和外部 KDC – 不同集群上的集群 KDC 具有 Active Directory 跨领域信任。

> **Important**
>
> 当主节点终止时，由于主节点使用临时存储，KDC 和主体数据库都会丢失。如果您为 SSH 连接创建用户，我们建议您使用配置为高可用性的外部 KDC 来建立跨领域信任。或者，如果您使用 Linux 账户为 SSH 连接创建用户，请使用引导操作和脚本自动执行账户创建过程，以便在创建新集群时重复该过程。

在创建之后或者创建集群时提交步骤到集群，这是添加用户和 KDC 委托人的最简单方法。或者，您可以作为默认 hadoop 用户，使用 EC2 密钥对连接到主节点来运行命令。有关更多信息，请参阅使用 SSH 连接到主节点。

下面的示例引用集群 ID，将 bash 脚本 configureCluster.sh 提交到已有的集群。该脚本会保存到 Amazon S3 中。

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,\
Args=["s3://DOC-EXAMPLE-BUCKET/configureCluster.sh"]
```

以下示例演示了 configureCluster.sh 脚本的内容。该脚本还处理创建 HDFS 用户目录和为 SSH 启用 GSSAPI 的过程，这些内容在下面的部分中介绍。

```
#!/bin/bash
#Add a principal to the KDC for the primary node, using the primary node's returned
 host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([lijuan]=pwd1 [marymajor]=pwd2 [richardroe]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
     name=${i}
     password=${arr[${i}]}

     # Create a principal for each user in the primary node and require a new password
 on first logon
     sudo kadmin.local -q "addprinc -pw $password +needchange $name"

     #Add hdfs directory for each user
     hdfs dfs -mkdir /user/$name

     #Change owner of each user's hdfs directory to that user
     hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

添加用户 HDFS 目录

要允许您的用户登录到集群来运行 Hadoop 任务，您必须为其 Linux 账户添加 HDFS 用户目录，并为每个用户授予对其目录的所有权。

在创建之后或者创建集群时提交步骤到集群，这是创建 HDFS 目录的最简单方法。或者，您可以作为默认 hadoop 用户，使用 EC2 密钥对连接到主节点来运行命令。有关更多信息，请参阅使用 SSH 连接到主节点。

下面的示例引用集群 ID，将 bash 脚本 AddHDFSUsers.sh 提交到已有的集群。该脚本会保存到 Amazon S3 中。

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-BUCKET/AddHDFSUsers.sh"]
```

以下示例演示了 AddHDFSUsers.sh 脚本的内容。

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD, or Linux users created manually on the
 cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
     hdfs dfs -mkdir /user/$username
     hdfs dfs -chown $username:$username /user/$username
done
```

为 SSH 启用 GSSAPI

对于通过 Kerberos 进行身份验证的用户，要使用 SSH 连接到主节点，SSH 服务必须启用了 GSSAPI 身份验证。要启用 GSSAPI，请从主节点命令行运行以下命令或者使用步骤来将其作为脚本运行。重新配置 SSH 之后，您必须重新启动服务。

```
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

通过 SSH 连接使用 Kerberos 的集群

此部分演示了通过 Kerberos 进行身份验证的用户连接到 EMR 集群主节点的步骤。

用于 SSH 连接的每台计算机必须安装了 SSH 客户端和 Kerberos 客户端应用程序。默认情况下，Linux 计算机很可能包括这些内容。例如，大多数 Linux、Unix 和 macOS 操作系统上都会安装 OpenSSH。您可以通过在命令行键入 ssh 来检查 SSH 客户端。如果您的计算机无法识别该命令，请安装 SSH 客户端以连接到主节点。OpenSSH 项目会免费提供一套完整的 SSH 工具执行文件。有关更多信息，请访问 OpenSSH 网站。Windows 用户可以使用 PuTTY 等应用程序作为 SSH 客户端。

有关 SSH 连接的更多信息，请参阅连接到集群。

SSH 使用 GSSAPI 对 Kerberos 客户端进行身份验证，您必须为集群主节点上的 SSH 服务启用 GSSAPI 身份验证。有关更多信息，请参阅为 SSH 启用 GSSAPI。SSH 客户端也必须使用 GSSAPI。

在以下示例中，对于 *MasterPublicDNS*，使用在集群详细信息窗格的摘要选项卡上为主节点公有 DNS显示的值，例如 *ec2-11-222-33-44.compute-1.amazonaws.com*。

krb5.conf（非 Active Directory）的先决条件

使用没有 Active Directory 集成的配置时，除了 SSH 客户端和 Kerberos 客户端应用程序之外，每个客户端计算机必须具有 `/etc/krb5.conf` 文件的副本，该文件与集群主节点上的 `/etc/krb5.conf` 文件匹配。

复制 krb5.conf 文件

1. 使用 SSH 使用 EC2 密钥对和默认用户（例如 hadoop）连接到主节点，例如 hadoop@*MasterPublicDNS*。有关详细说明，请参阅连接到集群。
2. 从主节点，复制 `/etc/krb5.conf` 文件的内容。有关更多信息，请参阅连接到集群。
3. 在将连接到集群的每个客户端计算机上，基于上一步中制作的副本创建一个相同的 `/etc/krb5.conf` 文件。

使用 Kinit 和 SSH

用户每次使用 Kerberos 凭证从客户端计算机连接时，该用户必须首先续订 Kerberos 票证以供自己在客户端计算机上使用。此外，SSH 客户端必须配置为使用 GSSAPI 身份验证。

通过 SSH 连接到使用 Kerberos 的 EMR 集群

1. 使用 `kinit` 续订您的 Kerberos 票证，如以下示例所示

```
kinit user1
```

2. 使用 `ssh` 客户端以及您在集群专用 KDC 中创建的委托人或者 Active Directory 用户名。确保已启用 GSSAPI 身份验证，如以下示例所示。

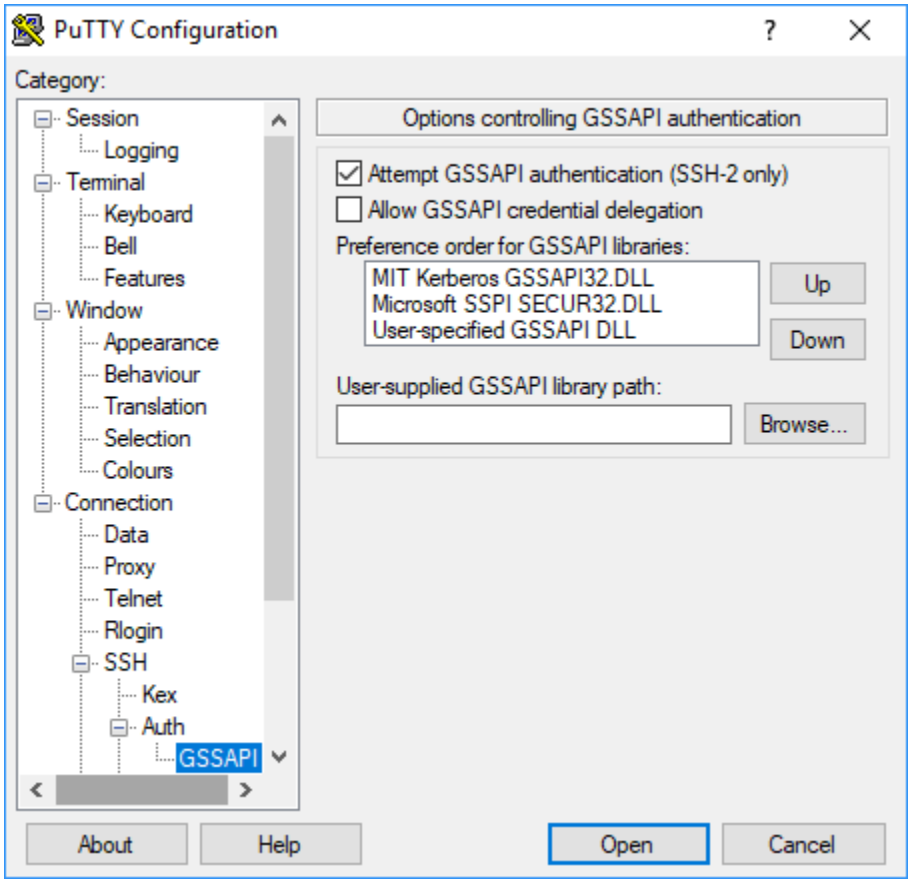
   示例：Linux 用户

   `-K` 选项指定 GSSAPI 身份验证。

```
ssh -K user1@MasterPublicDNS
```

   示例：Windows 用户 (PuTTY)

   确保已启用会话的 GSSAPI 身份验证选项，如下所示：



教程：配置集群专用的 KDC

本主题引导您创建具有集群专用密钥分发中心（KDC）的集群，手动将 Linux 账户添加到所有集群节点，将 Kerberos 主体添加到主节点上的 KDC 并确保客户端计算机已安装 Kerberos 客户端。

有关 Kerberos 和 KDC 的 Amazon EMR 支持的更多信息以及指向 MIT Kerberos 文档的链接，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。

步骤 1：创建使用 Kerberos 的集群

1. 创建启用 Kerberos 的安全配置。以下示例演示使用 Amazon CLI 将安全配置指定为内联 JSON 结构的 `create-security-configuration` 命令。您也可以引用本地保存的文件。

```
aws emr create-security-configuration --name MyKerberosConfig \
--security-configuration '{"AuthenticationConfiguration": {"KerberosConfiguration":
{"Provider": "ClusterDedicatedKdc", "ClusterDedicatedKdcConfiguration":
 {"TicketLifetimeInHours": 24}}}}'
```

2. 创建引用安全配置、指定集群的 Kerberos 属性并使用引导操作添加 Linux 账户的集群。以下示例演示使用 Amazon CLI 的 `create-cluster` 命令。此命令引用您在上面创建的安全配置 `MyKerberosConfig`。它还引用一个简单脚本 `createlinuxusers.sh` 作为引导操作，这是您在创建集群之前创建并上载到 Amazon S3 的脚本。

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-5.36.1 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair \
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd \
--bootstrap-actions Path=s3://DOC-EXAMPLE-BUCKET/createlinuxusers.sh
```

以下编码演示了 `createlinuxusers.sh` 脚本的内容，它将 user1、user2 和 user3 添加到集群中的每个节点。在下一个步骤中，您将这些用户添加为 KDC 委托人。

```
#!/bin/bash
sudo adduser user1
sudo adduser user2
sudo adduser user3
```

步骤 2：将委托人添加到 KDC、创建 HDFS 用户目录并配置 SSH

主节点上运行的 KDC 需要为本地主机和您在集群上创建的每个用户添加主体。您还可以为每个用户创建 HDFS 目录 (如果他们需要连接到集群并运行 Hadoop 任务)。同样，配置 SSH 服务以启用 GSSAPI 身份验证，这是 Kerberos 所必需的。在启用 GSSAPI 后，重新启动 SSH 服务。

完成这些任务的最简单方法是向集群提交步骤。下面的示例将 bash 脚本 `configurekdc.sh` 提交到您在上一步中创建的集群，引用其集群 ID。该脚本会保存到 Amazon S3 中。或者，您可以使用 EC2 密钥对连接到主节点来运行命令或在集群创建过程中提交步骤。

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
myregion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-
BUCKET/configurekdc.sh"]
```

以下编码演示了 configurekdc.sh 脚本的内容。

```
#!/bin/bash
#Add a principal to the KDC for the primary node, using the primary node's returned
 host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([user1]=pwd1 [user2]=pwd2 [user3]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
     name=${i}
     password=${arr[${i}]}

     # Create principal for sshuser in the primary node and require a new password on
 first logon
     sudo kadmin.local -q "addprinc -pw $password +needchange $name"

     #Add user hdfs directory
     hdfs dfs -mkdir /user/$name

     #Change owner of user's hdfs directory to user
     hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

您添加的用户现在应该可以使用 SSH 连接到集群。有关更多信息，请参阅通过 SSH 连接使用 Kerberos 的集群。

教程：配置与 Active Directory 域的跨领域信任

在您设置跨领域信任时，允许来自不同 Kerberos 领域的委托人 (通常为用户) 对 EMR 集群上的应用程序组件进行身份验证。集群专用的密钥分发中心（KDC）与另一个 KDC 建立信任关系时所使用的是这两个 KDC 中都存在的跨领域委托人。委托人名称和密码精确匹配。

跨领域信任要求 KDC 可以通过网络相互访问并解析对方的域名。下面提供了与作为 EC2 实例运行的 Microsoft AD 域控制器建立跨领域信任关系的步骤，以及提供所需连接和域名解析的示例网络设置。任何允许 KDC 之间所需网络流量的网络设置均可以接受。

（可选）在使用一个集群上的 KDC 建立与 Active Directory 的跨领域信任之后，您可以使用不同安全配置创建另一个集群，引用第一个集群上的 KDC 作为外部 KDC。有关示例安全配置和集群设置，请参阅外部集群 KDC 与 Active Directory 跨领域信任。

有关 Kerberos 和 KDC 的 Amazon EMR 支持的更多信息以及指向 MIT Kerberos 文档的链接，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。

> **Important**
>
> Amazon EMR 不支持使用 Amazon Directory Service for Microsoft Active Directory 的跨领域信任。

步骤 1：设置 VPC 和子网

步骤 2：启动并安装 Active Directory 域控制器

步骤 3：将用户账户添加到 EMR 集群的域中

步骤 4：在 Active Directory 域控制器上配置传入信任

步骤 5：使用 DHCP 选项集指定 Active Directory 域控制器作为 VPC DNS 服务器

步骤 6：启动使用 Kerberos 的 EMR 集群

步骤 7：在集群上为 Active Directory 账户创建 HDFS 用户并设置权限

步骤 1：设置 VPC 和子网

以下步骤演示如何创建 VPC 和子网，以便集群专用的 KDC 可以访问 Active Directory 域控制器并解析其域名。在这些步骤中，通过引用 Active Directory 域控制器作为 DHCP 选项集中的域名服务器来提供域名解析。有关更多信息，请参阅步骤 5：使用 DHCP 选项集指定 Active Directory 域控制器作为 VPC DNS 服务器。

KDC 和 Active Directory 域控制器必须能够解析对方的域名。这样，Amazon EMR 才能将计算机加入域并自动在集群实例上配置相应 Linux 账户和 SSH 参数。

如果 Amazon EMR 无法解析域名，您可以使用 Active Directory 域控制器的 IP 地址来引用信任。但是，您必须手动添加 Linux 账户、将相应主体添加到集群专用的 KDC 并配置 SSH。

设置 VPC 和子网

1. 创建带单个公有子网的 Amazon VPC。有关更多信息，请参阅《Amazon VPC 入门指南》中的步骤 1：创建 VPC。

   > **Important**
   >
   > 当您使用 Microsoft Active Directory 域控制器时，请为 EMR 集群选择 CIDR 块，以便所有 IPv4 地址的长度少于 9 个字符 (例如，10.0.0.0/16)。这是因为在计算机加入 Active Directory 目录时使用了集群计算机的 DNS 名称。Amazon 基于 IPv4 地址分配 DNS 主机名，较长的 IP 地址可能导致 DNS 名称超过 15 个字符。在注册已加入的计算机名称时，Active Directory 具有 15 个字符的限制，并且将截断较长的名称，这可能会导致不可预测的错误。

2. 删除分配给 VPC 的默认 DHCP 选项集。有关更多信息，请参阅更改 VPC 以使用 Noo DHCP 选项。稍后，添加一个指定 Active Directory 域控制器作为 DNS 服务器的新 DHCP 选项集。
3. 确认已为 VPC 启用了 DNS 支持，也就是说，同时启用了 DNS 主机名和 DNS 解析。默认处于启用状态。有关更多信息，请参阅更新您的 VPC 的 DNS 支持。
4. 确认您的 VPC 已附加 Internet 网关，这是默认设置。有关更多信息，请参阅创建并附加 Internet 网关。

   > **Note**
   >
   > 此示例中使用 Internet 网关是因为您将为 VPC 建立新的域控制器。您的应用程序可能不需要 Internet 网关。唯一的要求是，集群专用的 KDC 可以访问 Active Directory 域控制器。

5. 创建自定义路由表，添加目标为 Internet 网关的路由，然后将其附加到您的子网。有关更多信息，请参阅创建自定义路由表。
6. 在您启动域控制器的 EC2 实例时，它必须具有静态公有 IPv4 地址，您才能使用 RDP 连接到它。执行此操作的最简单方法是将您的子网配置为自动分配公有 IPv4 地址。这不是创建子网时的默认设置。有关更多信息，请参阅修改子网的公有 IPv4 寻址属性。也可以选择在启动实例时分配地址。有关更多信息，请参阅在实例启动期间分配公有 IPv4 地址。

7. 完成后，记下您的 VPC 和子网 ID。稍后，您在启动 Active Directory 域控制器和集群时会使用它们。

步骤 2：启动并安装 Active Directory 域控制器

1. 基于 Microsoft Windows Server 2016 Base AMI 启动 EC2 实例。我们推荐 m4.xlarge 或更好的实例类型。有关更多信息，请参阅《适用于 Windows 实例的 Amazon EC2 用户指南》中的启动 Amazon Web Services Marketplace 实例。
2. 记下与 EC2 实例关联的安全组的组 ID。您将在 步骤 6：启动使用 Kerberos 的 EMR 集群中需要它。我们使用 *sg-012xrlmdomain345*。或者，您可以为 EMR 集群指定不同的安全组，并指定允许安全组之间流量的此实例。有关更多信息，请参阅《适用于 Linux 实例Amazon EC2 用户指南》中的适用于 Linux 实例的 Amazon EC2 安全组。
3. 使用 RDP 连接到 EC2 实例。有关更多信息，请参阅《适用于 Windows 实例的 Amazon EC2 用户指南》中的连接您的 Windows 实例。
4. 启动 Server Manager (服务器管理器) 以在服务器上安装并配置 Active Directory 域服务角色。将服务器提升为域控制器并分配域名 (我们在此处使用的示例是 *ad.domain.com*)。请记下该域名，因为稍后您创建 EMR 安全配置和集群时会需要它。如果您并不熟悉如何设置 Active Directory，可以按照如何在 Windows Server 2016 中设置 Active Directory（AD）中的说明操作。

   在您完成后，实例会重新启动。

步骤 3：将用户账户添加到 EMR 集群的域中

使用 RDP 连接到 Active Directory 域控制器以在 Active Directory 用户和计算机中为每个集群用户创建账户。有关更多信息，请参阅 Microsoft Learn 网站上的在 Active Directory 用户和计算机中创建用户帐户。请记下每个用户的 User logon name (用户登录名)。稍后，在您配置集群时会需要这些信息。

另外，创建具有将计算机加入域的足够权限的账户。在您创建集群时指定此账户。Amazon EMR 使用它将集群实例加入域。在步骤 6：启动使用 Kerberos 的 EMR 集群 中指定此账户及其密码。要将计算机加入权限委派给账户，我们建议您创建具有加入权限的组，然后将用户分配到该组。有关说明，请参阅《Amazon Directory Service 管理指南》中的委派目录加入权限。

步骤 4：在 Active Directory 域控制器上配置传入信任

下面的示例命令在 Active Directory 中创建信任，这是使用集群专用的 KDC 的单向、传入、不可传递的领域信任。我们对集群的领域使用的示例为 *EC2.INTERNAL*。将 *KDC-FQDN* 替换为托管 KDC 的 Amazon EMR 主节点上列出的 公有 DNS 名称。`passwordt` 参数指定 cross-realm principal

password (跨领域委托人密码)，您在创建集群时指定了它以及集群 realm (领域)。领域名称派生自 us-east-1 中集群的默认域名。Domain 是您正在创建信任的 Active Directory 域，按照惯例，它是小写的。此示例使用 *ad.domain.com*

使用管理员权限打开 Windows 命令提示符并键入以下命令，在 Active Directory 域控制器上创建信任关系：

```
C:\Users\Administrator> ksetup /addkdc EC2.INTERNAL KDC-FQDN
C:\Users\Administrator> netdom trust EC2.INTERNAL /Domain:ad.domain.com /add /realm /
passwordt:MyVeryStrongPassword
C:\Users\Administrator> ksetup /SetEncTypeAttr EC2.INTERNAL AES256-CTS-HMAC-SHA1-96
```

步骤 5：使用 DHCP 选项集指定 Active Directory 域控制器作为 VPC DNS 服务器

现在已配置 Active Directory 域控制器，您必须配置 VPC 以使用它作为您的 VPC 内的域名服务器来解析名称。为此，请附加 DHCP 选项集。指定 Domain name (域名) 作为您的集群的域名，例如，ec2.internal（如果您的集群位于 us-east-1 中）或 *region*.compute.internal（对于其它区域）。对于 Domain name servers，您必须将 Active Directory 域控制器的 IP 地址 (必须可从集群访问) 指定为第一个条目，后跟 AmazonProvidedDNS (例如，***xx.xx.xx.xx***,AmazonProvidedDNS)。有关更多信息，请参阅更改 DHCP 选项集。

步骤 6：启动使用 Kerberos 的 EMR 集群

1. 在 Amazon EMR 中创建安全配置，指定您在之前的步骤中创建的 Active Directory 域控制器。下面显示了一个示例命令。将域 *ad.domain.com* 替换为您在步骤 2：启动并安装 Active Directory 域控制器 中指定的域的名称。

```
aws emr create-security-configuration --name MyKerberosConfig \
--security-configuration '{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
          "Domain": "ad.domain.com",
          "AdminServer": "ad.domain.com",
          "KdcServer": "ad.domain.com"
        }
      }
```

```
    }
  }
}'
```

2. 使用下面的属性创建集群：

   - 使用 `--security-configuration` 选项指定您创建的安全配置。在本例中我们使用 *MyKerberosConfig*。
   - 使用 `--ec2-attributes option` 的 SubnetId 属性来指定您在 步骤 1：设置 VPC 和子网 中创建的子网。在本例中我们使用 *step1-subnet*。
   - 使用 AdditionalMasterSecurityGroups 选项的 AdditionalSlaveSecurityGroups 和 `--ec2-attributes`，指定与步骤 2：启动并安装 Active Directory 域控制器中的 AD 域控制器关联的安全组与集群主节点以及核心节点和任务节点相关联。在本例中我们使用 *sg-012xrlmdomain345*。

   使用 `--kerberos-attributes` 指定以下特定于集群的 Kerberos 属性：

   - 您在设置 Active Directory 域控制器时指定的集群的领域。
   - 您在passwordt 中指定为 步骤 4：在 Active Directory 域控制器上配置传入信任 的跨领域信任委托人密码。
   - KdcAdminPassword，可用于管理集群专用的 KDC。
   - 您在步骤 3：将用户账户添加到 EMR 集群的域中中创建的具有计算机加入权限的 Active Directory 账户的用户登录名和密码。

   以下示例启动使用 Kerberos 的集群。

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-5.10.0 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair,\
SubnetId=step1-subnet, AdditionalMasterSecurityGroups=sg-012xrlmdomain345,
AdditionalSlaveSecurityGroups=sg-012xrlmdomain345\
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
CrossRealmTrustPrincipalPassword=MatchADTrustPwd
```

步骤 7：在集群上为 Active Directory 账户创建 HDFS 用户并设置权限

在设置与 Active Directory 的信任关系时，Amazon EMR 将在集群上为每个 Active Directory 账户创建 Linux 用户。例如，Active Directory 中的用户登录名 `LiJuan` 具有 Linux 账户 `lijuan`。Active Directory 用户名可以包含大写字母，但 Linux 不区分 Active Directory 大小写。

要允许您的用户登录到集群来运行 Hadoop 任务，您必须为其 Linux 账户添加 HDFS 用户目录，并为每个用户授予对其目录的所有权。为此，我们建议您运行以集群步骤的形式保存到 Amazon S3 的脚本。或者，您也可以从主节点上的命令行运行以下脚本中的命令。使用您在创建集群时指定的 EC2 密钥对以 Hadoop 用户身份通过 SSH 连接到主节点。有关更多信息，请参阅对 SSH 凭证使用 EC2 密钥对。

运行以下命令来将步骤添加到运行脚本 *AddHDFSUsers.sh* 的集群中。

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-BUCKET/AddHDFSUsers.sh"]
```

文件 *AddHDFSUsers.sh* 的内容如下所示。

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD or Linux users and KDC principals created
 manually on the cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
     hdfs dfs -mkdir /user/$username
     hdfs dfs -chown $username:$username /user/$username
done
```

映射到 Hadoop 组的 Active Directory 组

Amazon EMR 使用 System Security Services Daemon（SSD）将 Active Directory 组映射到 Hadoop 组。要确认组映射，在您按通过 SSH 连接使用 Kerberos 的集群中所述登录到主节点后，可以使用 `hdfs groups` 命令确认您的 Active Directory 账户所属的 Active Directory 组是否已映射到集群上相应

Hadoop 用户的 Hadoop 组。您还可以使用命令 (例如 `hdfs groups lijuan`)，通过指定一个或多个用户名来查看其它用户的组映射。有关更多信息，请参阅 Apache HDFS 命令指南中的组。

使用 Active Directory 或 LDAP 服务器通过 Amazon EMR 进行身份验证。

在 Amazon EMR 6.12.0 以及更高版本中，您可以使用基于 SSL 的 LDAP（LDAPS）协议来启动与您的企业身份服务器原生集成的集群。LDAP（轻量目录访问协议）是一种开放的、供应商中立的应用程序协议，用于访问和维护数据。LDAP 通常用于针对托管在 Active Directory（AD）和 OpenLDAP 等应用程序上的企业身份服务器进行用户身份验证。通过这种原生集成，您可以使用 LDAP 服务器对 Amazon EMR 上的用户进行身份验证。

Amazon EMR LDAP 集成的亮点包括：

- Amazon EMR 将支持的应用程序配置为代表您使用 LDAP 身份验证进行身份验证。
- Amazon EMR 使用 Kerberos 协议为支持的应用程序配置和维护其安全。您无需输入任何命令或脚本。
- 通过 Apache Ranger 对 Hive 元存储数据库和表的授权，您可以获得精细访问控制 (FGAC)。参阅将 Amazon EMR 与 Apache Ranger 集成 了解更多信息。
- 当您要求 LDAP 凭证才能访问集群时，您可以获得对谁可以通过 SSH 访问您的 EMR 集群的精细访问控制（FGAC）。

以下页面提供了概念性概述、先决条件以及启动集成了 Amazon EMR LDAP 的 EMR 集群的步骤。

主题

- 集成了 Amazon EMR 的 LDAP 概览
- Amazon EMR 的 LDAP 组件
- 适用于 Amazon EMR 的 LDAP 的应用程序支持和注意事项
- 使用 LDAP 配置和启动 EMR 集群
- 在 Amazon EMR 上使用 LDAP 的示例

集成了 Amazon EMR 的 LDAP 概览

轻型目录访问协议（LDAP）是一种软件协议，网络管理员使用该协议通过对公司网络中的用户进行身份验证来管理和控制对数据的访问。LDAP 协议将信息存储在分层的树形目录结构中。有关更多信息，请参阅 LDAP.com 上的 Basic LDAP Concepts。

在公司的网络中，许多应用程序可能会使用 LDAP 协议对用户进行身份验证。通过 Amazon EMR LDAP 集成，EMR 集群可以原生使用相同的 LDAP 协议并添加安全配置。

Amazon EMR 支持两种主要的 LDAP 协议实现：Active Directory 和 OpenLDAP。虽然其他实现也是可能的，但大多数实现都符合与 Active Directory 或 OpenLDAP 相同的身份验证协议。

Active Directory（AD）

Active Directory（AD）是 Microsoft 为 Windows 域网络提供的目录服务。AD 包含在大多数 Windows 服务器操作系统中，并且可以通过 LDAP 和 LDAPS 协议与客户端通信。为了进行身份验证，Amazon EMR 会尝试使用用户主体名称（UPN）作为可分辨名称和密码与您的 AD 实例进行用户绑定。UPN 使用标准格式 `username@domain_name`。

OpenLDAP

OpenLDAP 是 LDAP 协议的免费开源实现。为了进行身份验证，Amazon EMR 会尝试使用完全限定域名（FQDN）作为可分辨名称和密码与您的 OpenLDAP 实例进行用户绑定。FQDN 使用标准格式 `username_attribute=username,LDAP_user_search_base`。通常，`username_attribute` 值为 `uid`，并且 `LDAP_user_search_base` 值包含指向用户的树的属性。例如，`ou=People,dc=example,dc=com`。

LDAP 协议的其他免费和开源实现通常遵循与 OpenLDAP 类似的 FQDN 来标识其用户的可分辨名称。

Amazon EMR 的 LDAP 组件

您可以使用 LDAP 服务器通过以下组件对 Amazon EMR 以及用户直接在 EMR 集群上使用的任何应用程序进行身份验证。

Secret Agent

Secret Agent 是一个集群上进程，用于对所有用户请求进行身份验证。Secret Agent 代表 EMR 集群上支持的应用程序创建与您的 LDAP 服务器的用户绑定。Secret Agent 以 `emrsecretagent` 用户身份运行，将日志写入 `/emr/secretagent/log` 目录。这些日志提供了有关每个用户的身份验证请求状态以及用户身份验证期间可能出现的任何错误的详细信息。

系统安全服务进程守护程序（SSSD）

SSSD 是在支持 LDAP 的 EMR 集群的每个节点上运行的进程守护程序。SSSD 创建并管理 UNIX 用户，以将您的远程企业身份同步到每个节点。基于 Yarn 的应用程序（例如 Hive 和 Spark）要求在为用户运行查询的每个节点上都有一个本地 UNIX 用户。

适用于 Amazon EMR 的 LDAP 的应用程序支持和注意事项

适用于 Amazon EMR 的 LDAP 支持的应用程序

> **Important**
>
> 此页面上列出的应用程序是 Amazon EMR 针对 LDAP 提供支持的唯一应用程序。为确保集群安全，只有在创建启用了 LDAP 的 EMR 集群时，才能包含与 LDAP 兼容的应用程序。如果您尝试安装其他不支持的应用程序，Amazon EMR 将拒绝您对新集群的请求。

Amazon EMR 6.12 及更高版本支持与以下应用程序的 LDAP 集成：

- Apache Livy
- Apache Hive，通过 HiveServer2（HS2）
- Trino
- Presto
- Hue

您还可以将以下应用程序安装在 EMR 集群上，并且进行配置以满足您的安全需求：

- Apache Spark
- Apache Hadoop

适用于 Amazon EMR 的 LDAP 支持的功能

您可以将以下 Amazon EMR 功能与 LDAP 集成结合使用：

> **Note**
>
> 为保障 LDAP 凭证的安全，您必须使用传输中加密来保护集群内外的数据流。有关传输中加密的更多信息，请参阅 加密静态数据和传输中的数据。

- 传输中加密（要求）和静态加密
- 实例组、实例集和竞价型实例
- 在运行中的集群上重新配置应用程序

- EMRFS 服务器端加密（SSE）

不支持的 特征

在使用 Amazon EMR LDAP 集成时，请考虑以下限制：

- Amazon EMR 禁用启用了 LDAP 的集群的步骤。
- Amazon EMR 不支持启用了 LDAP 的集群的运行时系统角色和 Amazon Lake Formation 集成。
- Amazon EMR 不支持带有 StartTLS 的 LDAP。
- Amazon EMR 不支持启用了 LDAP 的集群的高可用性模式（具有多个主节点的集群）。
- 您无法轮换启用了 LDAP 的集群的绑定凭证或证书。如果这些字段中的任何一个已轮换，我们建议您使用更新的绑定凭证或证书启动新集群。
- 必须在 LDAP 中使用精确的搜索库。LDAP 用户和组搜索库不支持 LDAP 搜索筛选条件。

使用 LDAP 配置和启动 EMR 集群

本节介绍如何将 Amazon EMR 配置为与 LDAP 身份验证配合使用。

主题

- 向 Amazon EMR 实例角色添加 Amazon Secrets Manager 权限
- 创建用于 LDAP 集成的 Amazon EMR 安全配置
- 启动通过 LDAP 进行身份验证的 EMR 集群

向 Amazon EMR 实例角色添加 Amazon Secrets Manager 权限

Amazon EMR 使用 IAM 服务角色代表您执行操作以预置和管理集群。集群 EC2 实例的服务角色（又称为 Amazon EMR 的 EC2 实例配置文件）是一种特殊类型的服务角色，在启动时由 Amazon EMR 分配给集群中的每个 EC2 实例。

为定义 EMR 集群与 Amazon S3 数据和其他 Amazon 服务交互的权限，您应该定义一个自定义 Amazon EC2 实例配置文件而不是 `EMR_EC2_DefaultRole`，以在启动集群时使用。有关更多信息，请参阅 集群 EC2 实例（EC2 实例配置文件）的服务角色 和 自定义 IAM 角色：

将以下语句添加到 Amazon EMR 的默认 EC2 实例配置文件中，以便能够标记会话并访问存储 LDAP 证书的 Amazon Secrets Manager。

```
    {
```

```
      "Sid": "AllowAssumeOfRolesAndTagging",
      "Effect": "Allow",
      "Action": ["sts:TagSession", "sts:AssumeRole"],
      "Resource": [
        "arn:aws:iam::111122223333:role/LDAP_DATA_ACCESS_ROLE_NAME",
        "arn:aws:iam::111122223333:role/LDAP_USER_ACCESS_ROLE_NAME"
      ]
    },
    {
        "Sid": "AllowSecretsRetrieval",
        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [
            "arn:aws:secretsmanager:us-east-1:111122223333:secret:LDAP_SECRET_NAME*",
            "arn:aws:secretsmanager:us-
east-1:111122223333:secret:ADMIN_LDAP_SECRET_NAME*"
        ]
    }
```

> **Note**
>
> 如果您在设置 Secrets Manager 权限时忘记了密钥名称末尾的通配符 * 字符，集群请求将失败。通配符代表密钥版本。
> 您还应该将该 Amazon Secrets Manager 策略的范围限制为集群预置实例所需的证书。

创建用于 LDAP 集成的 Amazon EMR 安全配置

在启动集成了 LDAP 的 EMR 集群之前，请使用 创建安全配置 中的步骤为该集群创建 Amazon EMR 安全配置。在 `AuthenticationConfiguration` 下的 `LDAPConfiguration` 区块中，或在 Amazon EMR 控制台安全配置部分中的相应字段中完成以下配置：

`EnableLDAPAuthentication`

控制台选项：身份验证协议：LDAP

要使用 LDAP 集成，请在控制台中创建集群时将此选项设置为 `true` 或选择它作为您的身份验证协议。默认情况下，当您在 Amazon EMR 控制台中创建安全配置时，`EnableLDAPAuthentication` 的值为 `true`。

`LDAPServerURL`

控制台选项：LDAP 服务器位置

LDAP 服务器的位置包括前缀：ldaps://*location_of_server*。

BindCertificateARN

控制台选项：LDAP SSL 证书

包含用于签署 LDAP 服务器使用的 SSL 证书的证书的 Amazon Secrets Manager ARN。如果您的 LDAP 服务器由公共证书颁发机构（CA）签署，则您可以为 Amazon Secrets Manager ARN 提供一个空白文件。有关如何在 Secrets Manager 中存储证书的更多信息，请参阅 将 TLS 凭证存储在 Amazon Secrets Manager 中。

BindCredentialsARN

控制台选项：LDAP 服务器绑定凭证

包含 LDAP 管理员用户绑定凭证的 Amazon Secrets Manager ARN。凭证存储为 JSON 对象。此密钥中只有一个键值对；键值对中的键是用户名，值是密码。例如，{"uid=admin,cn=People,dc=example,dc=com": "AdminPassword1"}。除非您为 EMR 集群启用 SSH 登录，否则此字段为可选字段。在许多配置中，Active Directory 实例需要绑定凭证才能允许 SSSD 同步用户。

LDAPAccessFilter

控制台选项：LDAP 访问筛选条件

指定 LDAP 服务器中可以进行身份验证的对象子集。例如，如果您只想向 LDAP 服务器中具有 posixAccount 对象类的所有用户授予访问权限，请将访问筛选条件定义为 (objectClass=posixAccount)。

LDAPUserSearchBase

控制台选项：LDAP 用户搜索库

您的用户在 LDAP 服务器中所属的搜索库。例如，cn=People,dc=example,dc=com。

LDAPGroupSearchBase

控制台选项：LDAP 组搜索库

您的组在 LDAP 服务器中所属的搜索库。例如，cn=Groups,dc=example,dc=com。

EnableSSHLogin

控制台选项：SSH 登录

指定是否允许使用 LDAP 凭证进行密码身份验证。我们不建议您保持启用此选项。密钥对允许访问 EMR 集群的更安全的路由。此字段是可选的，默认值为 `false`。

`LDAPServerType`

控制台选项：LDAP 服务器类型

指定 Amazon EMR 连接到的 LDAP 服务器的类型。支持的选项有“Active Directory”和“OpenLDAP”。其他 LDAP 服务器类型可能会起作用，但是 Amazon EMR 不正式支持其他服务器类型。有关更多信息，请参阅Amazon EMR 的 LDAP 组件。

`ActiveDirectoryConfigurations`

使用 Active Directory 服务器类型的安全配置的必需子块。

`ADDomain`

控制台选项：Active Directory 域

用于创建用户主体名称（UPN）的域名，用于对使用 Active Directory 服务器类型的安全配置进行用户身份验证。

使用 LDAP 和 Amazon EMR 进行安全配置的注意事项

- 要创建与 Amazon EMR LDAP 集成的安全配置，您必须使用传输中加密。有关传输中加密的信息，请参阅 加密静态数据和传输中的数据。
- 您不能在相同的安全配置中定义 Kerberos 配置。Amazon EMR 会自动预置一个专用的 KDC，并管理此 KDC 的管理员密码。用户无法访问此管理员密码。
- 您不能在相同的安全配置中定义 IAM 运行时系统角色和 Amazon Lake Formation。
- `LDAPServerURL` 的值中必须包含 `ldaps://` 协议。
- `LDAPAccessFilter` 不能为空。

将 LDAP 与 Amazon EMR 的 Apache Ranger 集成一起使用

通过 Amazon EMR 的 LDAP 集成，您可以进一步与 Apache Ranger 集成。当您将您的 LDAP 用户拉取到 Ranger 中时，可以将这些用户与 Apache Ranger 策略服务器关联，以便与 Amazon EMR 和其他应用程序集成。为此，请在与 LDAP 集群一起使用的安全配置中的 `AuthorizationConfiguration` 中定义该 `RangerConfiguration` 字段。有关如何设置安全配置的更多信息，请参阅 创建 EMR 安全配置。

将 LDAP 与 Amazon EMR 配合使用时，您无需为 Apache Ranger 提供带有 Amazon EMR 集成的 `KerberosConfiguration`。

启动通过 LDAP 进行身份验证的 EMR 集群

使用以下步骤启动带有 LDAP 或 Active Directory 的 EMR 集群。

1. 设置您的环境：
   - 确保您的 EMR 集群上的节点可以与 Amazon S3 和 Amazon Secrets Manager 进行通信。有关如何修改您的 EC2 实例配置文件角色以与这些服务通信的更多信息，请参阅 向 Amazon EMR 实例角色添加 Amazon Secrets Manager 权限。
   - 如果您计划在私有子网中运行 EMR 集群，则应使用 Amazon PrivateLink 和 Amazon VPC 端点，或者使用网络地址转换（NAT）将 VPC 配置为与 S3 和 Secrets Manager 通信。有关更多信息，请参阅《Amazon VPC 入门指南》中的 Amazon PrivateLink 和 VPC 端点和 NAT 实例。
   - 确保您的 EMR 集群和 LDAP 服务器之间存在网络连接。您的 EMR 集群必须通过网络访问您的 LDAP 服务器。集群的主节点、核心节点和任务节点与 LDAP 服务器通信以同步用户数据。如果您的 LDAP 服务器在 Amazon EC2 上运行，请更新 EC2 安全组以接受来自 EMR 集群的流量。有关更多信息，请参阅向 Amazon EMR 实例角色添加 Amazon Secrets Manager 权限。
2. 创建用于 LDAP 集成的 Amazon EMR 安全配置 有关更多信息，请参阅创建用于 LDAP 集成的 Amazon EMR 安全配置。
3. 现在您已完成设置，请按照 启动 Amazon EMR 集群 中的步骤启动具有以下配置的集群：
   - 选择 Amazon EMR 6.12 或更高版本。我们建议您使用最新的 Amazon EMR 版本。
   - 仅为您的集群指定或选择支持 LDAP 的应用程序。有关在 Amazon EMR 上使用 LDAP 支持的应用程序的列表，请参阅 适用于 Amazon EMR 的 LDAP 的应用程序支持和注意事项。
   - 应用您在上一步中创建的安全配置。

在 Amazon EMR 上使用 LDAP 的示例

预置使用 LDAP 集成的 EMR 集群后，您可以通过其内置的用户名和密码身份验证机制向任何支持的应用程序提供 LDAP 凭证。本页显示一些示例。

在 Apache Hive 上使用 LDAP 身份验证

Example – Apache Hive

以下示例命令通过 HiveServer2 和 Beeline 启动 Apache Hive 会话：

```
beeline -u "jdbc:hive2://$HOSTNAME:10000/default;ssl=true;sslTrustStore=
$TRUSTSTORE_PATH;trustStorePassword=$TRUSTSTORE_PASS"  -n LDAP_USERNAME -
p LDAP_PASSWORD
```

在 Apache Livy 上使用 LDAP 身份验证

Example – Apache Livy

以下示例命令通过 cURL 启动 Livy 会话。将 *ENCODED-KEYPAIR* 替换为 `username:password` 的 Base64 编码字符串。

```
curl -X POST --data '{"proxyUser":"LDAP_USERNAME","kind": "pyspark"}' -H "Content-Type:
 application/json" -H "Authorization: Basic ENCODED-KEYPAIR" DNS_OF_PRIMARY_NODE:8998/
sessions
```

在 Presto 上使用 LDAP 身份验证

Example – Presto

以下示例命令通过 Presto CLI 启动 Presto 会话：

```
presto-cli --user "LDAP_USERNAME" --password --catalog hive
```

运行此命令后，在系统提示时输入 LDAP 密码。

在 Trino 上使用 LDAP 身份验证

Example Trino

以下示例命令通过 Trino CLI 启动 Trino 会话：

```
trino-cli --user "LDAP_USERNAME" --password --catalog hive
```

运行此命令后，在系统提示时输入 LDAP 密码。

在 Hue 上使用 LDAP 身份验证

您可以通过在集群上创建的 SSH 隧道访问 Hue 用户界面，也可以将代理服务器设置为公开广播与 Hue 的连接。由于 Hue 默认不在 HTTPS 模式下运行，因此我们建议您使用额外的加密层，以确保客

户端与 Hue 用户界面之间的通信使用 HTTPS 加密。这样可以减少您意外以纯文本形式公开用户凭证的可能性。

要使用 Hue 用户界面，请在浏览器中打开 Hue 用户界面，然后输入您的 LDAP 用户名密码进行登录。如果凭证正确，Hue 会将您登录并使用您的身份在所有支持的应用程序中对您进行身份验证。

使用 SSH 进行密码身份验证，对其他应用程序使用 Kerberos 票证

> **Important**
>
> 我们不建议您使用密码身份验证来通过 SSH 访问 EMR 集群。

您可以使用您的 LDAP 凭证通过 SSH 访问 EMR 集群。为此，请在用于启动集群的 Amazon EMR 安全配置中将 `EnableSSHLogin` 配置设置为 `true`。然后，在集群启动后，使用以下命令通过 SSH 连接到集群：

```
ssh username@EMR_PRIMARY_DNS_NAME
```

运行此命令后，在系统提示时输入 LDAP 密码。

Amazon EMR 包含一个集群脚本，允许用户生成 Kerberos keytab 文件和票证，用于不直接接受 LDAP 凭证的受支持应用程序。其中一些应用程序包括 `spark-submit` Spark SQL 和 PySpark。

运行 `ldap-kinit` 并按照提示操作。如果身份验证成功，Kerberos keytab 文件将与有效的 Kerberos 票证一起出现在您的主目录中。使用 Kerberos 票证像在任何使用 Kerberos 的环境中一样运行应用程序。

将 Amazon EMR 与 Amazon Lake Formation 集成

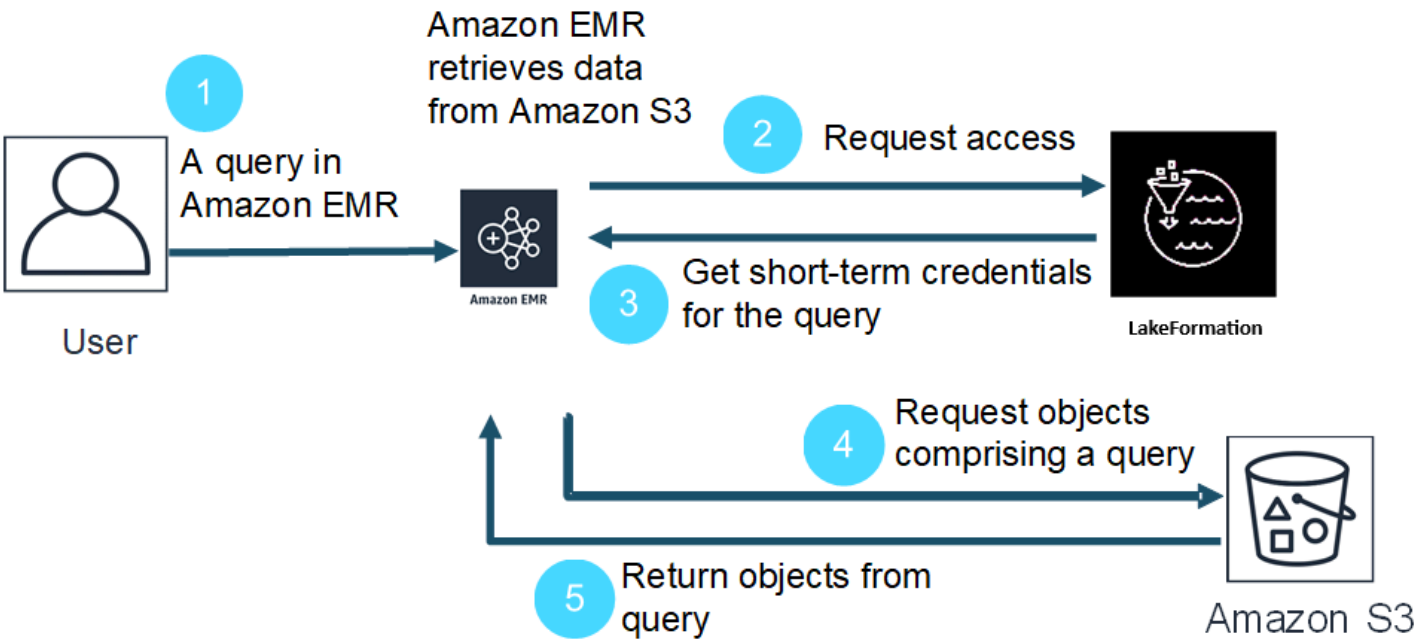
Amazon Lake Formation 是一项托管服务，可帮助您发现、分类、清理和保护 Amazon Simple Storage Service（S3）数据湖中的数据。Lake Formation 提供对 Amazon 数据目录中数据库和表的精细列级别访问。有关更多信息，请参阅什么是 Amazon Lake Formation？

使用 Amazon EMR 发行版本 6.7.0 及更高版本，您可以将基于 Lake Formation 的访问控制应用于您提交到 Amazon EMR 集群的 Spark、Hive 和 Presto 作业。要与 Lake Formation 集成，您必须创建具有运行时角色的 EMR 集群。运行时角色是您与 Amazon EMR 作业或查询关联的 Amazon Identity and Access Management (IAM) 角色。然后，Amazon EMR 使用此角色访问 Amazon 资源。有关更多信息，请参阅Amazon EMR 步骤的运行时角色。

Amazon EMR 如何与 Lake Formation 结合使用

将 Amazon EMR 与 Lake Formation 集成后，您可以使用 Step API 或 SageMaker Studio 对 Amazon EMR 集群执行查询。然后，Lake Formation 通过 Amazon EMR 的临时凭证提供对数据的访问。此过程称为凭证售卖。有关更多信息，请参阅什么是 Amazon Lake Formation？

以下是 Amazon EMR 如何访问受 Lake Formation 安全策略保护的数据的高级概览。



1. 用户对 Lake Formation 中的数据提交 Amazon EMR 查询。
2. Amazon EMR 从 Lake Formation 请求临时凭证以授予用户数据访问权限。
3. Lake Formation 返回临时凭证。
4. Amazon EMR 发送查询请求以从 Amazon S3 检索数据。
5. Amazon EMR 接收来自 Amazon S3 的数据，进行筛选，然后根据用户在 Lake Formation 中定义的用户权限返回结果。

有关将用户和组添加到 Lake Formation 策略中的更多信息，请参阅授予数据目录权限。

先决条件

集成 Amazon EMR 和 Lake Formation 之前，您必须满足以下要求：

- 在 Amazon EMR 集群上开启运行时角色授权。
- 使用 Amazon Glue Data Catalog 作为元数据存储。

- 在 Lake Formation 中定义和管理权限以访问 Amazon Glue 数据目录中的数据库、表和列。有关更多信息，请参阅什么是 Amazon Lake Formation？

主题

- 使用 Amazon EMR 启用 Lake Formation
- Apache Hudi 和 Lake Formation
- 注意事项

使用 Amazon EMR 启用 Lake Formation

在本节中，我们将介绍如何创建安全配置和设置 Lake Formation 以与 Amazon EMR 结合使用。还将介绍如何使用您为 Lake Formation 创建的安全配置启动集群。

步骤 1：为 EMR 集群设置运行时角色

要为 EMR 集群使用运行时角色，您必须创建安全配置。利用安全配置，您可以在集群中应用一致的安全性、授权和身份验证选项。

1. 使用以下安全配置创建名为 `lf-runtime-roles-sec-cfg.json` 的文件。

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true,
            "ApplicationScopedIAMRoleConfiguration":{
                "PropagateSourceIdentity":true
            }
        },
        "LakeFormationConfiguration":{
            "AuthorizedSessionTagValue":"Amazon EMR"
        },
  "EncryptionConfiguration": {
            "EnableInTransitEncryption": true,
            "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {<Certificate-configuration>}
        }
    }
}
```

2. 接下来，为确保会话标签可以授权 Lake Formation，请将 `LakeFormationConfiguration/AuthorizedSessionTagValue` 属性设置为 `Amazon EMR`。
3. 使用以下命令创建 Amazon EMR 安全配置。

```
aws emr create-security-configuration \
--name 'iamconfig-with-iam-lf' \
--security-configuration file://lf-runtime-roles-sec-cfg.json
```

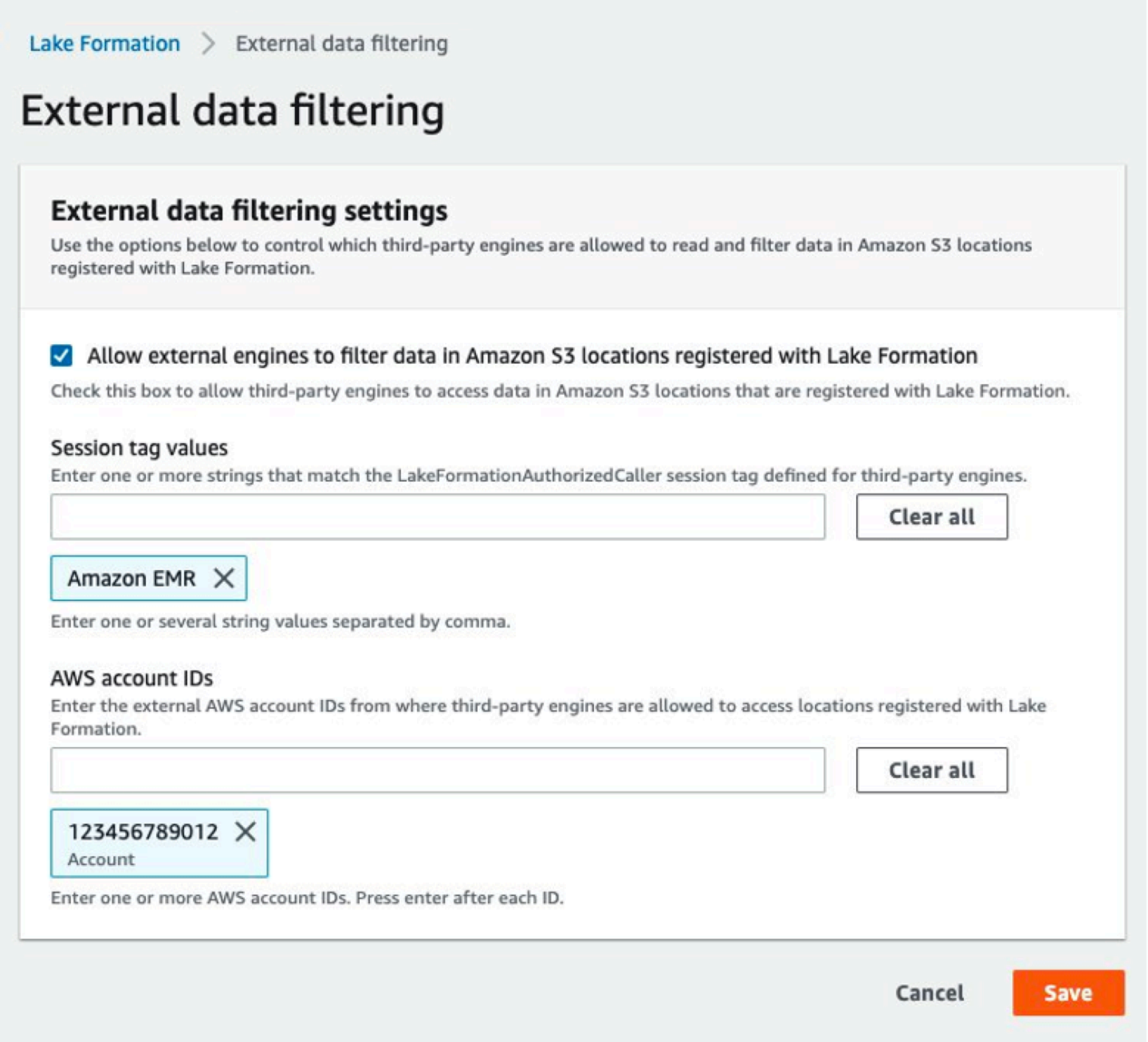
或者，您可以使用 Amazon EMR 控制台创建具有自定义设置的安全配置。

第 2 步：启动一个 Amazon EMR 集群

现在您可以使用您在上一步中创建的安全配置启动 EMR 集群。有关安全配置的更多信息，请参阅 使用安全配置设置集群安全性 和 Amazon EMR 步骤的运行时角色。

步骤 4：使用 Amazon EMR 运行时角色设置基于 Lake Formation 的访问控制

要使用 Lake Formation 应用表和列级别权限，Lake Formation 的数据湖管理员必须将 `Amazon EMR` 设置为会话标签配置的值 `AuthorizedSessionTagValue`。Lake Formation 使用该会话标签来授权调用者并提供对数据湖的访问权限。您可以在 Lake Formation 控制台的 External data filtering（外部数据筛选）部分设置该会话标签。将 *123456789012* 替换为您自己的 Amazon Web Services 账户 ID。



要继续使用 Amazon EMR 运行时角色设置基于 Lake Formation 的访问控制，您必须为 Amazon EMR 运行时角色配置 Amazon Glue 和 Lake Formation 授权。要让您的 IAM 运行时角色与 Lake Formation 进行交互，请使用 `lakeformation:GetDataAccess` 和 `glue:Get*` 向其授予访问权限。

Lake Formation 权限控制对 Amazon Glue 数据目录资源、Amazon S3 位置以及这些位置的基础数据的访问。IAM 权限控制对 Lake Formation 和 Amazon Glue API 及资源的访问。尽管您可能拥有访问数据目录 (SELECT) 中的表的 Lake Formation 权限，但如果没有 `glue:Get*` API 的 IAM 权限，您的操作将失败。有关 Lake Formation 访问控制的更多信息，请参阅 Lake Formation access control overview（Lake Formation 访问控制概述）。

1. 创建 emr-runtime-roles-lake-formation-policy.json 文件并输入以下内容。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "LakeFormationManagedAccess",
        "Effect": "Allow",
        "Action": [
            "lakeformation:GetDataAccess",
            "glue:Get*",
            "glue:Create*",
            "glue:Update*"
        ],
        "Resource": "*"
    }
}
```

2. 创建相关的 IAM policy。

```
aws iam create-policy \
--policy-name emr-runtime-roles-lake-formation-policy \
--policy-document file://emr-runtime-roles-lake-formation-policy.json
```

3. 要将此策略分配给您的 IAM 运行时角色，请按照管理 Amazon Lake Formation 权限中的步骤操作。

现在您可以使用运行时角色和 Lake Formation 来应用表和列级别权限。使用源身份时，您可以控制并监控对 Amazon CloudTrail 执行的操作。有关详细的端到端示例，请参阅 Introducing runtime roles for Amazon EMR steps（引入适用于 Amazon EMR 步骤的运行时角色）。

Apache Hudi 和 Lake Formation

在使用 Spark SQL 读取数据时，Amazon EMR 发行版 6.9.0 及更高版本包括对基于 Lake Formation 的访问控制及 Apache Hudi 的有限支持。Amazon EMR 支持针对使用 Spark SQL 的 SELECT 查询，并且仅限于列级访问控制。借助此功能，您现在可以运行以下命令：

- 对写入时复制表的快照查询，用于在指定的提交或压缩时刻查询表的最新快照。
- “读取时合并”表中的读取优化查询，用于查询最新的压缩数据，其中可能不包括尚未压缩的日志文件中的最新更新。

以下支持矩阵列出了 Apache Hudi 及 Lake Formation 的部分核心功能：

	写入时复制	读取时合并
快照查询 - Spark SQL	是	N
读取优化查询 - Spark SQL	不适用	是
递增查询	N	N
时间旅行查询	N	N
Spark Datasource 查询	N	N
Spark Datasource 写入	N	N
DML/DDL	N	N
元数据表	N	N

查询 Hudi 表

本节介绍如何在启用 Lake Formation 的集群上运行上述支持的查询。该表应为已注册的目录表。

本节介绍如何运行 Lake Formation 集群支持的查询，如上所述

1. 要启动 Spark Shell，请使用以下命令。

```
spark-shell  --jars /usr/lib/hudi/hudi-spark-bundle.jar \
--conf 'spark.serializer=org.apache.spark.serializer.KryoSerializer'
```

```
spark-sql  --jars /usr/lib/hudi/hudi-spark-bundle.jar \
--conf 'spark.serializer=org.apache.spark.serializer.KryoSerializer'
```

2. 要查询写入时复制表的最新快照，请使用以下命令。

```
select * from <my_hudi_cow_table>
```

```
spark.read.table("<my_hudi_cow_table>")
```

3. 要查询 MOR 表的最新压缩数据，可以查询以 _ro 为后缀的读取优化表：

```
SELECT * from <my_hudi_mor_table>_ro
```

```
spark.read.table("<my_hudi_mor_table>_ro")
```

> **Note**
>
> Lake Formation 集群上的读取性能可能会因不支持的优化而变慢。这些功能包括基于 Hudi 元数据的文件列表和数据跳过。我们建议您测试应用程序的性能，以确保其符合您的 SLA。

注意事项

将 Amazon EMR 与 Amazon Lake Formation 结合使用时，请注意以下事项。

- 具有表访问权限的用户可以访问该表的所有属性。如果您对表具有基于 Lake Formation 的访问控制，则请查看该表以确保属性不包含任何敏感数据或信息。
- 当 Spark 收集表统计数据时，带有 Lake Formation 的 Amazon EMR 集群不支持 Spark 回退到 HDFS。这通常有助于优化查询性能。
- 在不受监管的 Apache Spark 和 Apache Hive 上支持基于 Lake Formation 的访问控制的操作（自 Amazon EMR 6.10.0 和更高版本起）包括 `insert into` 和 `insert overwrite`。
- 通过 Apache Spark 和 Apache Hive 支持基于 Lake Formation 的访问控制的操作包括 `select`、`describe`、`show database`、`show table`、`show column` 和 `show partition`。
- Amazon EMR 不支持控制对以下基于 Lake Formation 的操作的访问：
  - 写入受监管表
  - Lake Formation 数据筛选条件
  - DDL 语句，例如 CREATE 或 ALTER 表
- 使用和不使用基于 Lake Formation 的访问控制的同一查询之间存在性能差异。

将 Amazon EMR 与 Apache Ranger 集成

从 Amazon EMR 5.32.0 开始，您可以启动与 Apache Ranger 在本地集成的集群。Apache Ranger 是一个开源框架，可跨 Hadoop 平台启用、监控和管理全面的数据安全。有关更多信息，请参阅 Apache

Ranger。通过本机集成，您可以自带 Apache Ranger，在 Amazon EMR 上强制实施精细数据访问控制。

此部分从概念上简要介绍了 Amazon EMR 与 Apache Ranger 的集成。它还包含启动与 Apache Ranger 集成的 Amazon EMR 集群所需的先决条件和步骤。

将 Amazon EMR 与 Apache Ranger 集成提供了下列主要优势：

- 对 Hive 元存储数据库和表的精细访问控制，使您能够为 Apache Spark 和 Apache Hive 应用程序在数据库、表和列级别定义数据筛选策略。Hive 应用程序支持行级过滤和数据屏蔽。
- 能够将您现有的 Hive 策略直接用于 Hive 应用程序的 Amazon EMR。
- 在前缀和对象级别对 Amazon S3 数据进行访问控制，这使您能够定义数据筛选策略以使用 EMR 文件系统访问 S3 数据。
- 能够使用 CloudWatch Logs 进行集中审计。
- Amazon EMR 代表您安装和管理 Apache Ranger 插件。

Apache Ranger

Apache Ranger 是一个框架，可跨 Hadoop 平台启用、监控和管理全面的数据安全。

Apache Ranger 具有以下功能：

- 集中安全管理以在中央 UI 或使用 REST API 管理所有与安全相关的任务。
- 使用 Hadoop 组件或工具执行特定操作或操作的精细授权，通过集中管理工具进行管理。
- 跨所有 Hadoop 组件的标准化授权方法。
- 增强了对各种授权方法的支持。
- 在 Hadoop 的所有组件中对用户访问和管理操作（安全相关）进行集中审计。

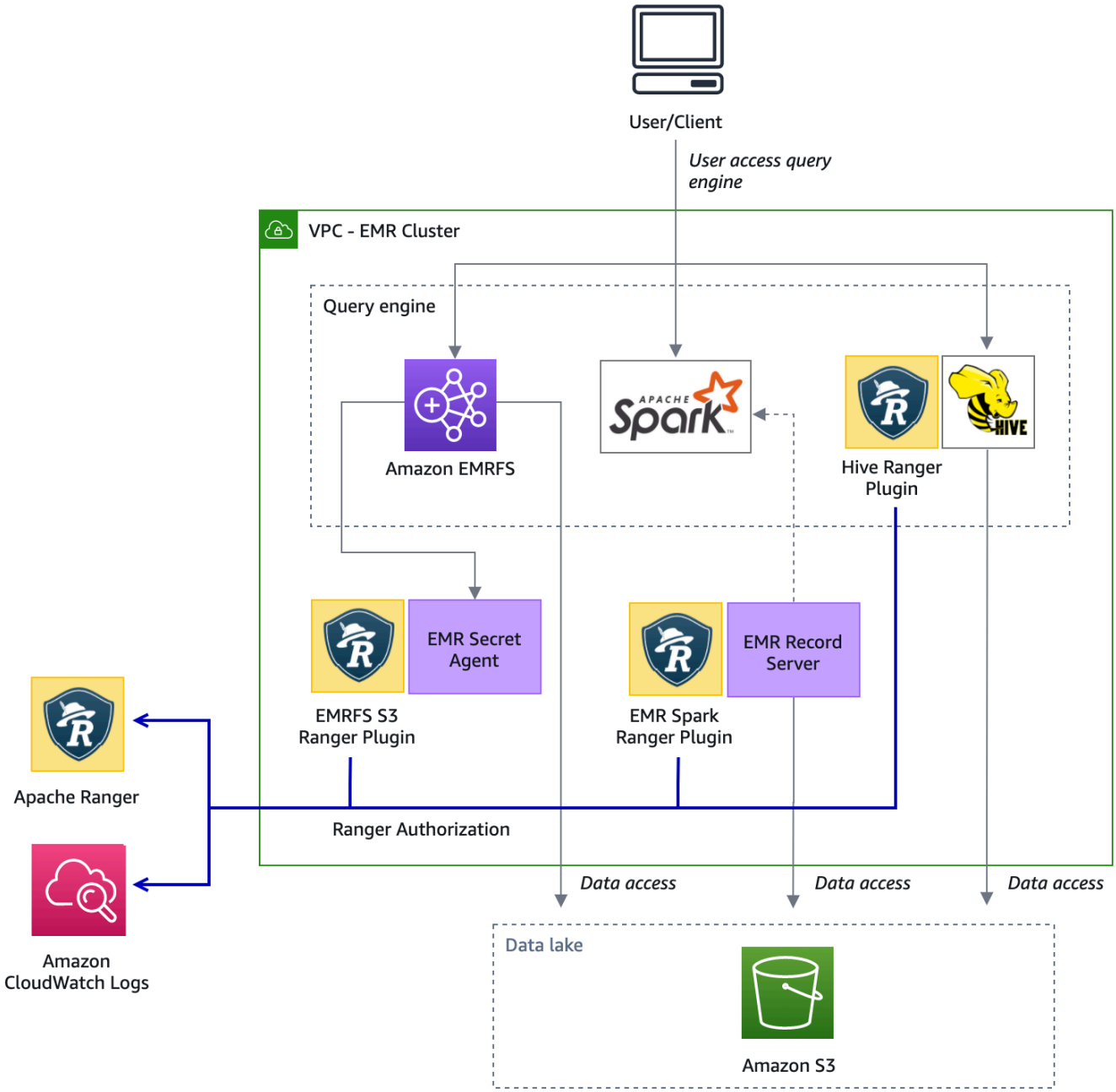
Apache Ranger 使用两个关键组件进行授权：

- Apache Ranger 策略管理服务器 - 此服务器允许您定义 Hadoop 应用程序的授权策略。与 Amazon EMR 集成时，您可以为 Apache Spark 和 Hive 定义和实施策略以访问 Hive Metastore，并访问 Amazon S3 数据 EMR 文件系统。您可以设置新的或使用现有的 Apache Ranger 策略管理服务器与 Amazon EMR 集成。
- Apache Ranger 插件 – 此插件根据 Apache Ranger 策略管理服务器中定义的授权策略验证用户的访问权限。Amazon EMR 会为在 Apache Ranger 配置中选择的每个 Hadoop 应用程序自动安装和配置 Apache Ranger 插件。

主题

- Amazon EMR 与 Apache Ranger 集成的架构
- Amazon EMR 组件

Amazon EMR 与 Apache Ranger 集成的架构



Amazon EMR 组件

Amazon EMR 通过以下组件支持 Apache Ranger 的精细访问控制。有关这些 Amazon EMR 组件与 Apache Ranger 插件的可视化表示，请参阅架构图。

私有代理 – 私有代理安全存储密钥并将密钥分发到其它 Amazon EMR 组件或应用程序。密钥可以包括临时用户凭证、加密密钥或 Kerberos 票证。私有代理在集群中的每个节点上运行并拦截对实例元数据服务的调用。对于对实例配置文件角色凭证的请求，在使用 EMRFS S3 Ranger 插件授权请求后，私有代理根据请求用户和请求的资源提供凭证。私有代理以 *emrsecretagent* 用户身份运行，将日志写入 /emr/secretagent/log 目录。该过程依赖于一组特定 `iptables` 规则来发挥作用。务必要确保 `iptables` 没有被禁用。如果您自定义 `iptables` 配置，则必须保留 NAT 表规则并保持不变。

EMR 记录服务器 – 记录服务器接收从 Spark 访问数据的请求。然后，它通过将请求的资源转发到适用于 Amazon EMR 的 Spark Ranger 插件来授权请求。记录服务器从 Amazon S3 读取数据，并根据 Ranger 策略返回用户有权访问的筛选数据。记录服务器作为 emr_record_server 用户在集群中的每个节点上运行，并将日志写入 /var/log/emr-record-server 目录。

应用程序支持和限制

支持的应用程序

在 Amazon EMR 和 Apache Ranger 的集成中，EMR 安装了 Ranger 插件，目前支持以下应用程序：

- Apache Spark（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Apache Hive（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- 通过 EMRFS 访问 S3（适用于 EMR 5.32 和 EMR 6.3 及更高版本）

以下应用程序可以安装在 EMR 集群上，并且可能需要进行配置以满足您的安全需求：

- Apache Hadoop（适用于 EMR 5.32 和 EMR 6.3 及更高版本，包括 YARN 和 HDFS）
- Apache Livy（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Apache Zeppelin（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Apache Hue（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Ganglia（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- HCatalog（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Mahout（适用于 EMR 5.32 和 EMR 6.3 及更高版本）

- MXNet（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- TensorFlow（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Tez（适用于 EMR 5.32 和 EMR 6.3 及更高版本）
- Trino（可用于 EMR 6.7 及更高版本）
- ZooKeeper（适用于 EMR 5.32 和 EMR 6.3 及更高版本）

> **Important**
>
> 上面列出的应用程序是目前唯一支持的应用程序。为了确保集群安全，启用 Apache Ranger 后，允许您仅使用上述列表中的应用程序创建 EMR 集群。
> 当前不支持其它应用程序。为了确保集群的安全性，尝试安装其它应用程序会导致您的集群被拒绝。

支持的功能

以下 Amazon EMR 功能可用于 Amazon EMR 和 Apache Ranger：

- 静态和动态加密
- Kerberos 身份验证（必需）
- 实例组、实例集和竞价型实例
- 在运行中的集群上重新配置应用程序
- EMRFS 服务器端加密（SSE）

> **Note**
>
> Amazon EMR 加密设置控制 SSE。有关更多信息，请参阅加密选项。

应用程序限制

集成 Amazon EMR 和 Apache Ranger 时，需要记住几个限制：

- 您当前无法使用控制台创建指定 Amazon GovCloud (US) Region 中的 Amazon Ranger 集成选项的安全配置 可以使用 CLI 完成安全配置。

- Kerberos 必须安装在您的集群上。
- 默认情况下，YARN 资源管理器 UI、HDFS NameNode UI 和 Livy UI 等应用程序 UI（用户界面）未设置身份验证。
- 配置了 HDFS 默认权限 `umask`，以便创建的对象默认设置为 `world wide readable`。
- Amazon EMR 不支持 Apache Ranger 的高可用性（多主）模式。
- 有关其他限制，请参阅各个应用程序的具体限制。

> **Note**
>
> Amazon EMR 加密设置控制 SSE。有关更多信息，请参阅加密选项。

插件限制

每个插件都有特定的限制。有关 Apache Hive 插件的限制，请参阅 Apache Hive 插件限制。有关 Apache Spark 插件的限制，请参阅 Apache Spark 插件限制。有关 EMRFS S3 插件的限制，请参阅 EMRFS S3 插件限制。

为 Apache Ranger 设置 Amazon EMR

在安装 Apache Ranger 之前，请查看本部分中的信息，以确保 Amazon EMR 已进行正确配置。

主题

- 设置 Ranger 管理服务器
- 用于与 Apache Ranger 本地集成的 IAM 角色
- 创建 EMR 安全配置
- 将 TLS 凭证存储在 Amazon Secrets Manager 中
- 启动 EMR 集群
- 为启用 Apache Ranger 的 Amazon EMR 集群配置 Zeppelin
- 已知问题

设置 Ranger 管理服务器

对于 Amazon EMR 集成，Apache Ranger 应用程序插件必须使用 TLS/SSL 与 Admin 服务器通信。

先决条件: Ranger Admin 服务器 SSL 启用

Amazon EMR 上的 Apache Ranger 需要插件和 Ranger Admin 服务器之间的双向 SSL 通信。为确保插件通过 SSL 与 Apache Ranger 服务器通信，请在 Ranger Admin 服务器上的 ranger-admin-site.xml 中启用以下属性。

```
<property>
    <name>ranger.service.https.attrib.ssl.enabled</name>
    <value>true</value>
</property>
```

此外，还需进行以下配置。

```
<property>
    <name>ranger.https.attrib.keystore.file</name>
    <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.file</name>
    <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.pass</name>
    <value>_<KEYSTORE_PASSWORD>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.keyalias</name>
    <value><PRIVATE_CERTIFICATE_KEY_ALIAS></value>
</property>

<property>
    <name>ranger.service.https.attrib.clientAuth</name>
    <value>want</value>
</property>

<property>
    <name>ranger.service.https.port</name>
    <value>6182</value>
</property>
```

TLS 凭证

Apache Ranger 与 Amazon EMR 的集成要求使用 TLS 加密从 Amazon EMR 节点到 Ranger Admin 服务器的流量，并且 Ranger 插件使用双向 TLS 身份验证向 Apache Ranger 服务器进行身份验证。Amazon EMR 服务需要 Ranger Admin 服务器的公共凭证（在前面的示例中指定）和私有凭证。

Apache Ranger 插件证书

Apache Ranger 插件公共 TLS 凭证必须可供 Apache Ranger 管理服务器访问，以在插件连接时进行验证。执行此操作共有三种方式。

方法 1: 在 Apache Ranger Admin 服务器中配置信任库

在 ranger-admin-site.xml 中填写以下配置以配置信任库。

```
<property>
    <name>ranger.truststore.file</name>
    <value><LOCATION TO TRUSTSTORE></value>
</property>

<property>
    <name>ranger.truststore.password</name>
    <value><PASSWORD FOR TRUSTSTORE></value>
</property>
```

方法 2: 将证书加载到 Java cacerts 信任库

如果您的 Ranger Admin 服务器没有在其 JVM 选项中指定信任库，那么您可以将插件公共凭证放在默认的 cacerts 存储中。

方法 3: 创建信任库并指定为 JVM 选项的一部分

在 {RANGER_HOME_DIRECTORY}/ews/ranger-admin-services.sh 中，修改 JAVA_OPTS，使其包含 "-Djavax.net.ssl.trustStore=*<TRUSTSTORE_LOCATION>*" 和 "-Djavax.net.ssl.trustStorePassword=*<TRUSTSTORE_PASSWORD>*"。例如，在现有 JAVA_OPTS 之后添加以下行。

```
JAVA_OPTS=" ${JAVA_OPTS} -Djavax.net.ssl.trustStore=${RANGER_HOME}/truststore/
truststore.jck -Djavax.net.ssl.trustStorePassword=changeit"
```

> **Note**
>
> 如果任何用户能够登录 Apache Ranger Admin 服务器并查看正在运行的进程，例如使用 ps 命令时，此规范可能会公开信任库密码。

使用自签名证书

不建议使用自签名凭证作为凭证。自签名凭证不得撤销，自签名凭证可能不符合内部安全要求。

服务定义安装

Ranger Admin 服务器使用服务定义来描述应用程序的策略属性。然后，将策略存储在策略存储库中以供客户端下载。

为了能够配置服务定义，必须对 Ranger Admin 服务器进行 REST 调用。请参阅 Apache Ranger PublicAPIsv2，以获取以下部分所需的 API。

安装 Apache Spark 的服务定义

要安装 Apache Spark 的服务定义，请参阅Apache Spark 插件。

安装 EMRFS 服务定义

要安装 Amazon EMR 的 S3 服务定义，请参阅EMRFS S3 插件。

使用 Hive 服务定义

Apache Hive 可以使用 Apache Ranger 2.0 及更高版本附带的现有 Ranger 服务定义。有关更多信息，请参阅Apache Hive 插件。

网络流量规则

当 Apache Ranger 与您的 EMR 集群集成时，该集群需要与其它服务器和 Amazon 通信。

所有 Amazon EMR 节点（包括核心节点和任务节点）都必须能够与 Apache Ranger Admin 服务器通信以下载策略。如果您的 Apache Ranger 管理员在 Amazon EC2 上运行，您需要更新安全组才能从 EMR 集群获取流量。

除了与 Ranger Admin 服务器通信之外，所有节点都需要能够与以下Amazon服务通信：

- Amazon S3

- Amazon KMS（如果使用 EMRFS SSE-KMS）
- Amazon CloudWatch
- Amazon STS

如果您计划在私有子网中运行 EMR 集群，请配置 VPC 以能够使用《Amazon VPC 用户指南》中的 Amazon PrivateLink 和 VPC 终端节点或使用《Amazon VPC 用户指南》中的网络地址转换（NAT）实例与这些服务进行通信。

用于与 Apache Ranger 本地集成的 IAM 角色

Amazon EMR 和 Apache Ranger 之间的集成依赖于您在启动集群之前应创建的三个关键角色：

- 适用于 Amazon EMR 的自定义 Amazon EC2 实例配置文件。
- Apache Ranger 引擎的 IAM 角色
- 其它Amazon服务的 IAM 角色

此部分概览了这些角色以及需要为每个 IAM 角色提供的策略。有关创建这些角色的信息，请参阅设置 Ranger 管理服务器。

EC2 实例配置文件

Amazon EMR 使用 IAM 服务角色代表您执行操作以预置和管理集群。集群 EC2 实例的服务角色（又称为 Amazon EMR 的 EC2 实例配置文件）是一种特殊类型的服务角色，在启动时分配给集群中的每个 EC2 实例。

要定义 EMR 集群与 Amazon S3 数据以及与受 Apache Ranger 和其他 Amazon 服务保护的 Hive 元存储交互的权限，请定义自定义 EC2 实例配置文件以在您启动集群时使用，而不是使用 `EMR_EC2_DefaultRole`。

有关更多信息，请参阅 集群 EC2 实例（EC2 实例配置文件）的服务角色 和 自定义 IAM 角色：

您需要将以下语句添加到 Amazon EMR 的默认 EC2 实例配置文件中，以便能够标记会话并访问存储 TLS 凭证的 Amazon Secrets Manager。

```
{
  "Sid": "AllowAssumeOfRolesAndTagging",
  "Effect": "Allow",
  "Action": ["sts:TagSession", "sts:AssumeRole"],
  "Resource": [
```

```
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_ENGINE-
PLUGIN_DATA_ACCESS_ROLE_NAME>",
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_USER_ACCESS_ROLE_NAME>"
      ]
    },
    {
        "Sid": "AllowSecretsRetrieval",
        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [

 "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<PLUGIN_TLS_SECRET_NAME>*",

 "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<ADMIN_RANGER_SERVER_TLS_SECRET_NAME>
        ]
    }
```

> **Note**
>
> 对于 Secrets Manager 权限，不要忘记密钥名称末尾的通配符（“*”），否则您的请求将失败。通配符用于密钥版本。

> **Note**
>
> 将 Amazon Secrets Manager 策略的范围限制为仅预置所需的凭证。

Apache Raner 的 IAM 角色

此角色为可信执行引擎（例如 Apache Hive 和 Amazon EMR 记录服务器）提供凭证以访问 Amazon S3 数据。如果您使用 S3 SSE-KMS，则仅使用此角色访问 Amazon S3 数据，包括任何 KMS 密钥。

此角色必须使用以下示例中所述的最低策略创建。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudwatchLogsPermissions",
      "Action": [
        "logs:CreateLogGroup",
```

```
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": [

 "arn:aws:logs:<REGION>:<AWS_ACCOUNT_ID>:<CLOUDWATCH_LOG_GROUP_NAME_IN_SECURITY_CONFIGURATION>:
      ]
    },
    {
      "Sid": "BucketPermissionsInS3Buckets",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
 *"arn:aws:s3:::bucket1",
        "arn:aws:s3:::bucket2"*
        ]
    },
    {
      "Sid": "ObjectPermissionsInS3Objects",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
 *"arn:aws:s3:::bucket1/*",
          "arn:aws:s3:::bucket2/*"
*        ]
    }
  ]
}
```

Important

必须包含 CloudWatch 日志资源末尾的星号“*”以提供写入日志流的权限。

> **Note**
>
> 如果您使用 EMRFS 一致性视图或 S3-SSE 加密，请添加对 DynamoDB 表和 KMS 密钥的权限，以便执行引擎可以与这些引擎交互。

Apache Ranger 的 IAM 角色由 EC2 实例配置文件角色代入。使用以下示例创建一个信任策略，允许 EC2 实例配置文件角色代入 Apache Ranger 的 IAM 角色。

```
   {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2 INSTANCE PROFILE ROLE NAME eg.
EMR_EC2_DefaultRole>"
     },
     "Action": ["sts:AssumeRole", "sts:TagSession"]
   }
```

其它Amazon服务的 IAM 角色。

如果需要，此角色可为不受信任的执行引擎的用户提供与Amazon服务交互的凭证。不要使用此 IAM 角色来允许访问 Amazon S3 数据，除非它是所有用户都应该可以访问的数据。

此角色将由 EC2 实例配置文件角色代入。使用以下示例创建一个信任策略，允许 EC2 实例配置文件角色代入 Apache Ranger 的 IAM 角色。

```
   {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2 INSTANCE PROFILE ROLE NAME eg.
EMR_EC2_DefaultRole>"
     },
     "Action": ["sts:AssumeRole", "sts:TagSession"]
   }
```

验证您的权限

请参阅Apache Ranger 疑难解答了解有关验证权限的说明。

创建 EMR 安全配置

为 Apache Ranger 创建 Amazon EMR 安全配置

在启动与 Apache Ranger 集成的 Amazon EMR 集群之前，请创建安全配置。

Console

创建一个安全配置来指定 Amazon Ranger 集成选项

1. 在 Amazon EMR 控制台中，依次选择 Security configurations (安全配置) 和 Create (创建)。
2. 键入安全配置的 Name (名称)。在创建集群时，使用此名称来指定安全配置。
3. 在 Amazon Ranger Integration (Ranger 集成) 下，选择 Enable fine-grained access control managed by Apache Ranger (启用由 Apache Ranger 管理的细粒度访问控制)。
4. 选择您要应用的 Apache Ranger 的 IAM 角色。有关更多信息，请参阅用于与 Apache Ranger 本地集成的 IAM 角色。
5. 选择要应用的 IAM role for otherr Amazon services (其它亚马逊云科技服务的 IAM 角色)。
6. 通过输入管理服务器的 Secrets Manager ARN 和地址，将插件配置为连接到 Ranger 管理服务器。
7. 选择应用程序以配置 Ranger 插件。填写包含插件私有 TLS 凭证的 Secrets Manager ARN。

   如果您未配置 Apache Spark 或 Apache Hive，并将它们选为集群的应用程序，则请求将失败。
8. 根据需要设置其它安全配置选项，然后选择 Create (创建)。您必须启用使用集群专用 KDC 或外部 KDC 的 Kerberos 身份验证。

> **Note**
>
> 您当前无法使用控制台创建指定 Amazon GovCloud (US) Region 中的 Amazon Ranger 集成选项的安全配置 可以使用 CLI 完成安全配置。

CLI

要为 Apache Ranger 集成创建安全配置

1. 将 *<ACCOUNT ID>* 替换为您的Amazon账户 ID。

2. 将 *<REGION>* 替换为资源所在的区域。
3. 为 TicketLifetimeInHours 指定一个值，用于确定 KDC 发放的 Kerberos 票证的有效期间。
4. 为 AdminServerURL 指定 Ranger Admin 服务器的地址。

```
{
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24
            }
        }
    },
    "AuthorizationConfiguration":{
      "RangerConfiguration":{
         "AdminServerURL":"https://_<RANGER ADMIN SERVER IP>_:6182",
         "RoleForRangerPluginsARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<RANGER PLUGIN
 DATA ACCESS ROLE NAME>_",
         "RoleForOtherAWSServicesARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<USER
 ACCESS ROLE NAME>_",
         "AdminServerSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES ADMIN SERVERS PUBLIC TLS CERTIFICATE
 WITHOUT VERSION>_",
         "RangerPluginConfigurations":[
            {
               "App":"Spark",
               "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES SPARK PLUGIN PRIVATE TLS CERTIFICATE
 WITHOUT VERSION>_",
               "PolicyRepositoryName":"<SPARK SERVICE NAME eg. amazon-emr-spark>"
            },
            {
               "App":"Hive",
               "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES Hive PLUGIN PRIVATE TLS CERTIFICATE WITHOUT
 VERSION>_",
               "PolicyRepositoryName":"<HIVE SERVICE NAME eg. Hivedev>"
            },
            {
               "App":"EMRFS-S3",
```

```
                "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES EMRFS S3 PLUGIN PRIVATE TLS CERTIFICATE
 WITHOUT VERSION>_",
                "PolicyRepositoryName":"<EMRFS S3 SERVICE NAME eg amazon-emr-emrfs>"
            },
        {
                "App":"Trino",
                "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES TRINO PLUGIN PRIVATE TLS CERTIFICATE
 WITHOUT VERSION>_",
                "PolicyRepositoryName":"<TRINO SERVICE NAME eg amazon-emr-trino>"
            }
          ],
          "AuditConfiguration":{
             "Destinations":{
                "AmazonCloudWatchLogs":{
                   "CloudWatchLogGroup":"arn:aws:logs:<REGION>:_<ACCOUNT ID>_:log-
group:_<LOG GROUP NAME FOR AUDIT EVENTS>_"
                }
             }
          }
       }
    }
}
```

PolicyRespositoryNames 是在 Apache Ranger Admin 中指定的服务名称。

使用以下命令创建 Amazon EMR 安全配置。用您选择的名称替换 security-configuration。创建集群时，按名称选择此配置。

```
aws emr create-security-configuration \
--security-configuration file://./security-configuration.json \
--name security-configuration
```

配置其它安全功能

为将 Amazon EMR 与 Apache Ranger 安全集成，您还应该配置以下 EMR 安全功能：

- 启用使用集群专用 KDC 或外部 KDC 的 Kerberos 身份验证。有关说明，请参阅 使用 Kerberos 通过 Amazon EMR 进行身份验证。
- （可选）启用传输中或静态加密。有关更多信息，请参阅加密选项。

有关更多信息，请参阅Amazon EMR 中的安全性。

将 TLS 凭证存储在 Amazon Secrets Manager 中

安装在 Amazon EMR 集群上的 Ranger 插件和 Ranger 管理服务器必须通过 TLS 进行通信，以确保发送的策略数据和其它信息在被拦截时无法读取。EMR 还要求插件通过提供自己的 TLS 凭证并执行双向 TLS 身份验证来对 Ranger Admin 服务器进行身份验证。此设置需要创建四个凭证：两对私有和公有 TLS 凭证。有关将凭证安装到 Ranger Admin 服务器的说明，请参阅设置 Ranger 管理服务器。要完成设置，安装在 EMR 集群上的 Ranger 插件需要两个凭证：管理服务器的公有 TLS 凭证，以及插件将用于对 Ranger 管理服务器进行身份验证的私有凭证。这些 TLS 凭证必须在 Amazon Secrets Manager 中并在 EMR 安全配置中提供。

> **Note**
>
> 强烈建议（但不是必需）为每个应用程序创建一个凭证对，以降低其中一个插件凭证遭到破坏时的影响。

> **Note**
>
> 您需要在凭证过期日期前跟踪和转动凭证。

凭证格式

无论是私有插件凭证还是公有 Ranger Admin 凭证，将凭证导入 Amazon Secrets Manager 都是一样的。在导入 TLS 凭证之前，必须确认凭证采用 509x PEM 格式。

公有凭证的示例格式如下：

```
-----BEGIN CERTIFICATE-----
...Certificate Body...
-----END CERTIFICATE-----
```

私有凭证的示例格式如下：

```
-----BEGIN PRIVATE KEY-----
...Private Certificate Body...
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
```

```
...Trust Certificate Body...
-----END CERTIFICATE-----
```

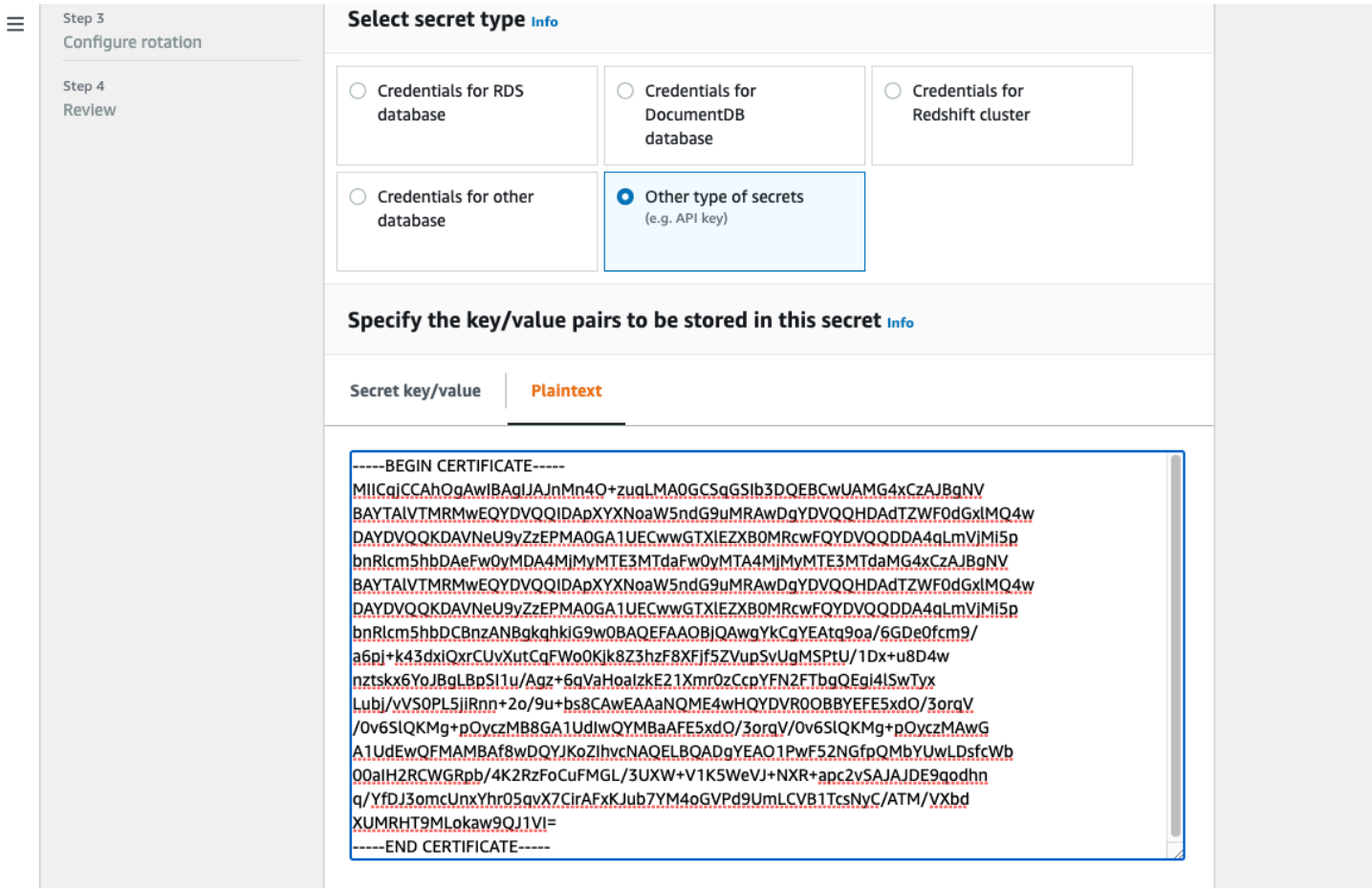
私有凭证还应包含信任凭证。

您可以通过运行以下命令验证是否采用的格式是否正确：

```
openssl x509 -in <PEM FILE> -text
```

将凭证导入到 Amazon Secrets Manager

在 Secrets Manager 中创建 Secret 时，在 secret type（密钥类型）下选择 Other type of secrets（其他类型密钥）并将 PEM 编码的凭证粘贴到 Plaintext（明文）字段中。



启动 EMR 集群

在使用 Apache Ranger 启动 Amazon EMR 集群之前，请确保每个组件都满足以下最低版本要求：

- Amazon EMR 5.32.0 或更高版本，或者 6.3.0 或更高版本。建议您使用最新的 Amazon EMR 版本。

- Apache Ranger Admin 服务器版本 2.x。

完成以下步骤：

- 如果尚未安装 Apache Ranger，请安装它。有关更多信息，请参阅 Apache Ranger 0.5.0 安装。
- 确保您的 Amazon EMR 集群和 Apache Ranger Admin 服务器之间存在网络连接。请参阅 设置 Ranger 管理服务器。
- 创建必要的 IAM 角色。请参阅用于与 Apache Ranger 本地集成的 IAM 角色。
- 为 Apache Ranger 安装创建 EMR 安全配置。有关更多信息，请参阅创建 EMR 安全配置。

为启用 Apache Ranger 的 Amazon EMR 集群配置 Zeppelin

本主题介绍了如何为启用 Apache Ranger 的 Amazon EMR 集群配置 Apache Zeppelin，以便您可以将 Zeppelin 作为笔记本以进行交互式的数据探索。Zeppelin 包含在 Amazon EMR 发行版 5.0.0 及更高版本中。早期版本包括 Zeppelin 作为沙盒应用程序。有关更多信息，请参阅《Amazon EMR 发布指南》中的 Amazon EMR 4.x 版本。

预设情况下，Zeppelin 配置了原定设置登录名和密码，但这在多租户环境中并不安全。

要配置 Zeppelin，请完成以下步骤：

1. 修改身份验证机制。

   修改 `shiro.ini` 文件以实现您首选的身份验证机制。Zeppelin 支持 Active Directory、LDAP、PAM 和 Knox SSO。有关详细信息，请参阅 Apache Zeppelin 的 Apache Shiro 身份验证。
2. 配置 Zeppelin 以模拟终端用户

   如果允许 Zeppelin 模拟终端用户，则 Zeppelin 提交的任务可作为该终端用户运行。将以下配置添加到 `core-site.xml`：

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "hadoop.proxyuser.zeppelin.hosts": "*",
      "hadoop.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
```

```
    ]
  }
]
```

然后将以下配置添加至位于 /etc/hadoop/conf 的 hadoop-kms-site.xml：

```
[
  {
    "Classification": "hadoop-kms-site",
    "Properties": {
      "hadoop.kms.proxyuser.zeppelin.hosts": "*",
      "hadoop.kms.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

您还可以按照在控制台中重新配置实例组中的步骤，使用控制台将这些配置添加到您的 Amazon EMR 集群。

3. 允许 Zeppelin 作为终端用户执行指令

   创建一个名为 /etc/sudoers.d/90-zeppelin-user 的文件，其中包含以下内容：

```
zeppelin ALL=(ALL) NOPASSWD:ALL
```

4. 修改解释器设置以在它们自己的进程中运行用户任务。

   配置所有解释器以在“隔离”进程中实例化“Per User”解释器。

5. 修改 **zeppelin-env.sh**

   将以下内容添加到 zeppelin-env.sh 以便 Zeppelin 以终端用户身份启动解释器：

```
ZEPPELIN_IMPERSONATE_USER=`echo ${ZEPPELIN_IMPERSONATE_USER} | cut -d @ -f1`
```

```
export ZEPPELIN_IMPERSONATE_CMD='sudo -H -u ${ZEPPELIN_IMPERSONATE_USER} bash -c'
```

将以下内容添加到 zeppelin-env.sh 以将默认的笔记本权限改为仅对创建者只读：

```
export ZEPPELIN_NOTEBOOK_PUBLIC="false"
```

最后，将以下配置添加到 zeppelin-env.sh，以在第一个 CLASSPATH 语句之后包含 EMR RecordServer 类路径：

```
export CLASSPATH="$CLASSPATH:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-connector-common.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-spark-connector.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-client.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-common.jar:/usr/share/aws/emr/record-server/lib/jars/secret-agent-interface.jar"
```

6. 重新启动 Zeppelin。

   运行以下命令以重新启动 Zeppelin：

   ```
   sudo systemctl restart zeppelin
   ```

已知问题

已知问题

Amazon EMR 发行版 5.32 中存在一个已知问题，其中更改了 hive-site.xml 的权限，以便只有特权用户才能读取它，因为其中可能存储了凭证。这可能会阻止 Hue 读取 hive-site.xml 并导致网页不断重新加载。如果遇到此问题，请添加以下配置来修复此问题：

```
[
  {
    "Classification": "hue-ini",
    "Properties": {},
    "Configurations": [
      {
        "Classification": "desktop",
        "Properties": {
          "server_group":"hive_site_reader"
         },
        "Configurations":[
```

```
        ]
      }
    ]
  }
]
```

一个已知的问题是适用于 Apache Ranger 的 EMRFS S3 插件当前不支持 Apache Ranger 的安全区功能。使用安全区功能定义的访问控制限制不适用于您的 Amazon EMR 集群。

应用程序 UI

默认情况下，应用程序 UI 不执行身份验证。这包括 ResourceManager UI、NodeManager UI、Livy UI 等。此外，任何有权访问 UI 的用户都能够查看有关所有其它用户的任务相关信息。

如果不需要这种行为，您应该确保使用安全组来限制用户对应用程序 UI 的访问。

HDFS 原定设置的权限

默认情况下，用户在 HDFS 中创建的对象被授予全局可读权限。这可能会导致无权访问数据的用户也可以读取数据。要更改此行为，以便将默认文件权限设置为仅由任务创建者读取和写入，请执行以下步骤。

创建 EMR 集群时，请提供以下配置：

```
[
  {
    "Classification": "hdfs-site",
    "Properties": {
      "dfs.namenode.acls.enabled": "true",
      "fs.permissions.umask-mode": "077",
      "dfs.permissions.superusergroup": "hdfsadmingroup"
    }
  }
]
```

此外，运行以下引导操作：

```
--bootstrap-actions Name='HDFS UMask Setup',Path=s3://elasticmapreduce/hdfs/umask/
umask-main.sh
```

Apache Ranger 插件

Apache Ranger 插件根据 Apache Ranger 策略管理服务器中定义的授权策略验证用户的访问权限。

主题

- Apache Hive 插件
- Apache Spark 插件
- EMRFS S3 插件
- Trino 插件

Apache Hive 插件

Apache Hive 是 Hadoop 生态系统中流行的执行引擎。Amazon EMR 提供了一个 Apache Ranger 插件，能够为 Hive 提供精细访问控制。该插件与开源 Apache Ranger Admin 服务器版本 2.0 及更高版本兼容。

主题

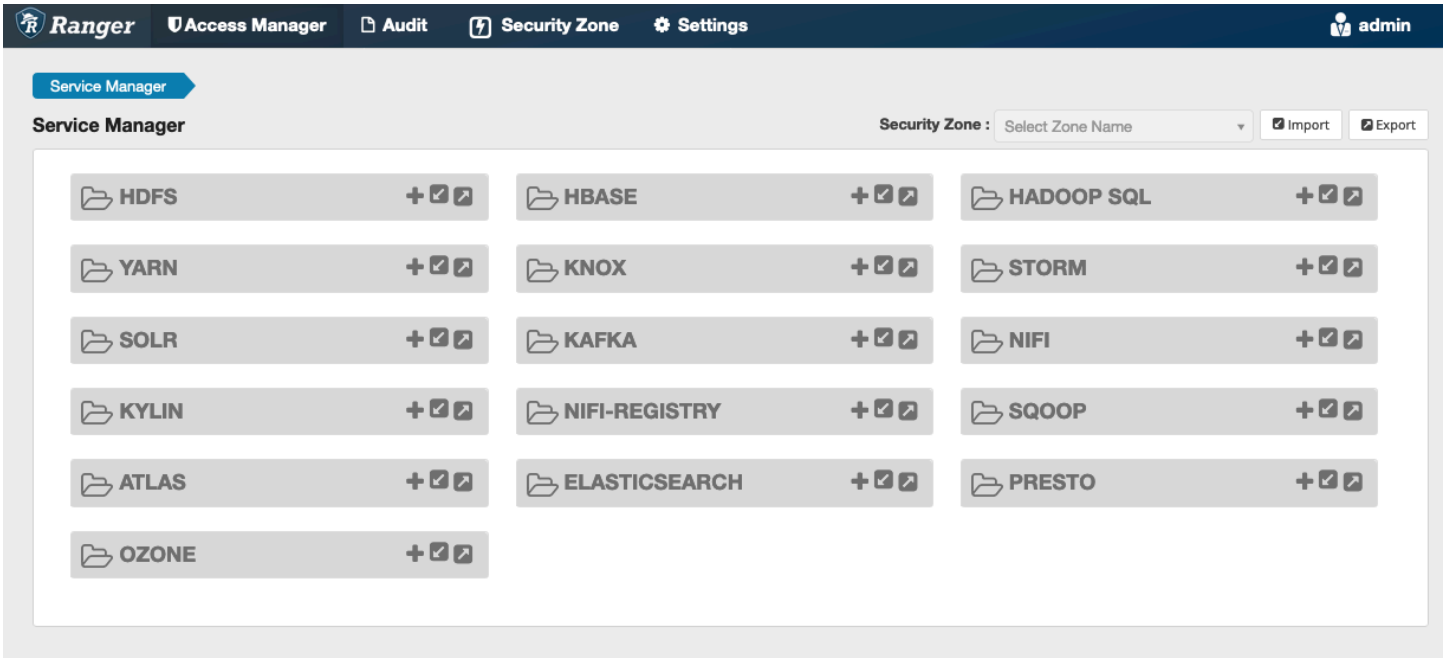
- 支持的特征
- 安装服务配置
- 注意事项
- 限制

支持的特征

EMR 上 Hive 的 Apache Ranger 插件支持开源插件的所有功能，包括数据库、表、列级访问控制以及行筛选和数据掩码。有关 Hive 命令和关联 Ranger 权限的表，请参阅 Hive 命令到 Ranger 权限映射。

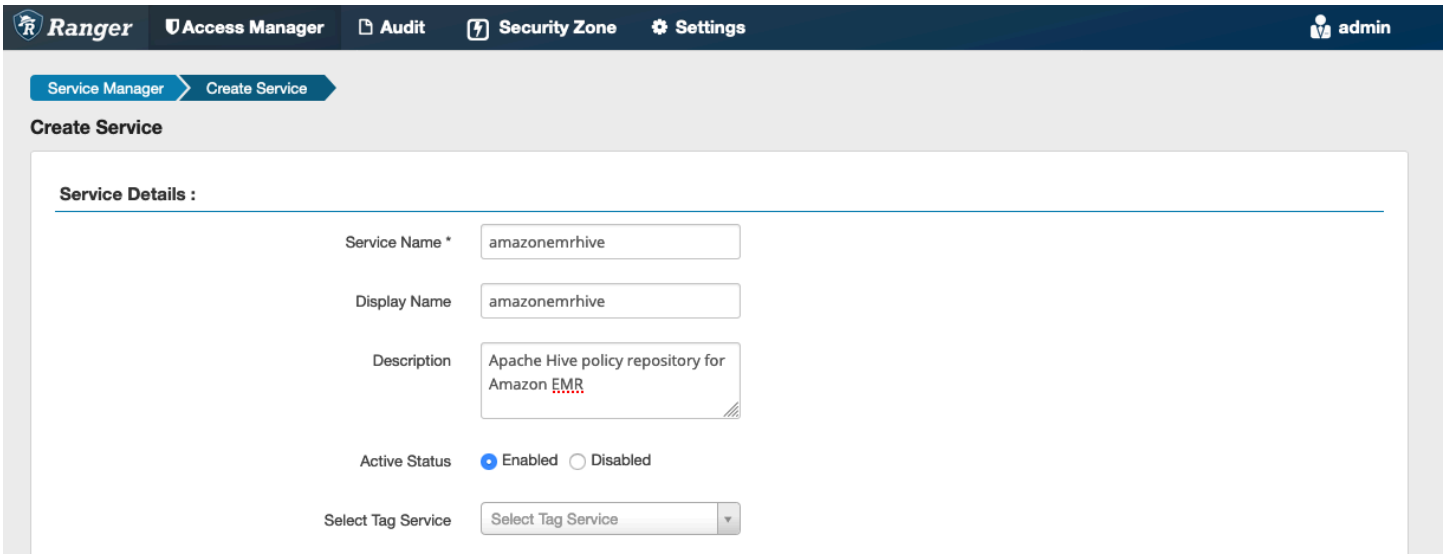
安装服务配置

Apache Hive 插件与 Apache Hive Hadoop SQL 中现有的 Hive 服务定义兼容。



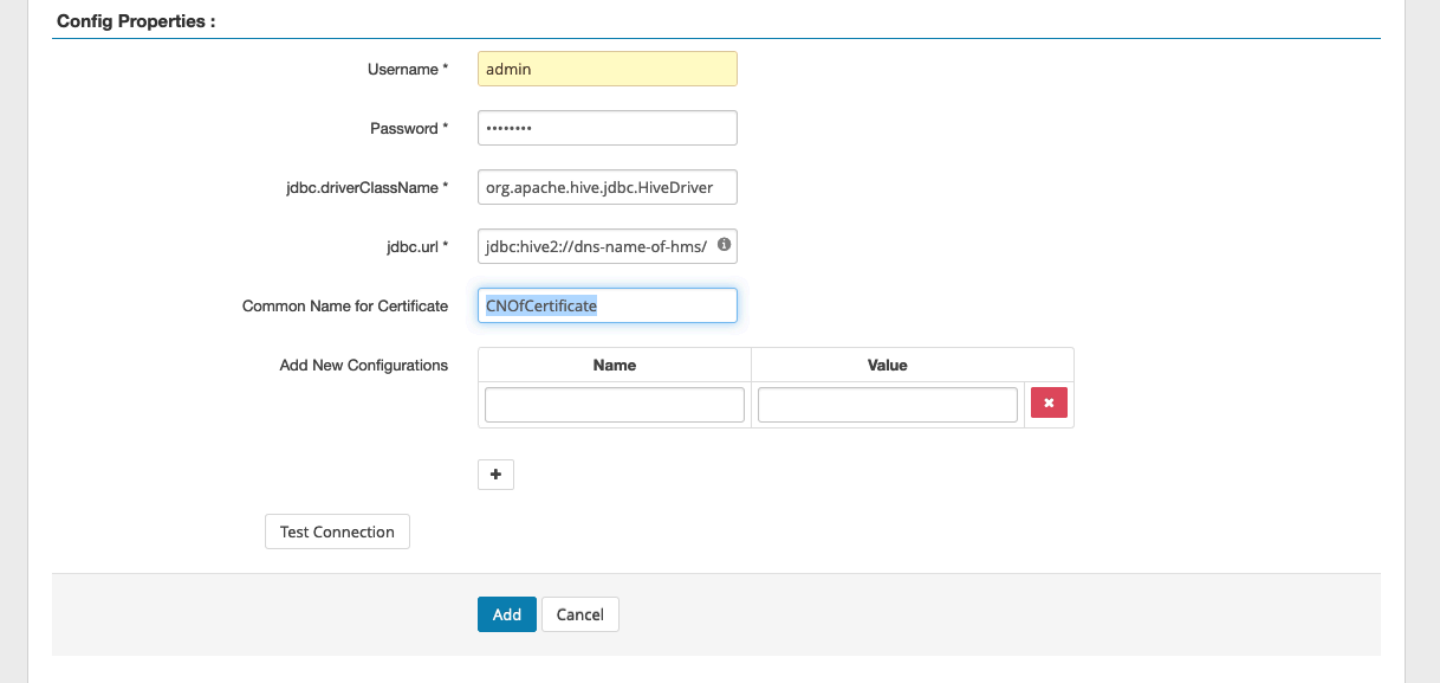
如果您没有 Hadoop SQL 下的服务实例（如上所示），则可以创建一个实例。点击 Hadoop SQL 旁边的 +。

1. Service Name (服务名称)（如果显示）：输入服务名称。建议的值为 **`amazonemrhive`**。记下此服务名称，创建 EMR 安全配置时将要用到。
2. Display Name (显示名称)：输入要为服务显示的名称。建议的值为 **`amazonemrhive`**。

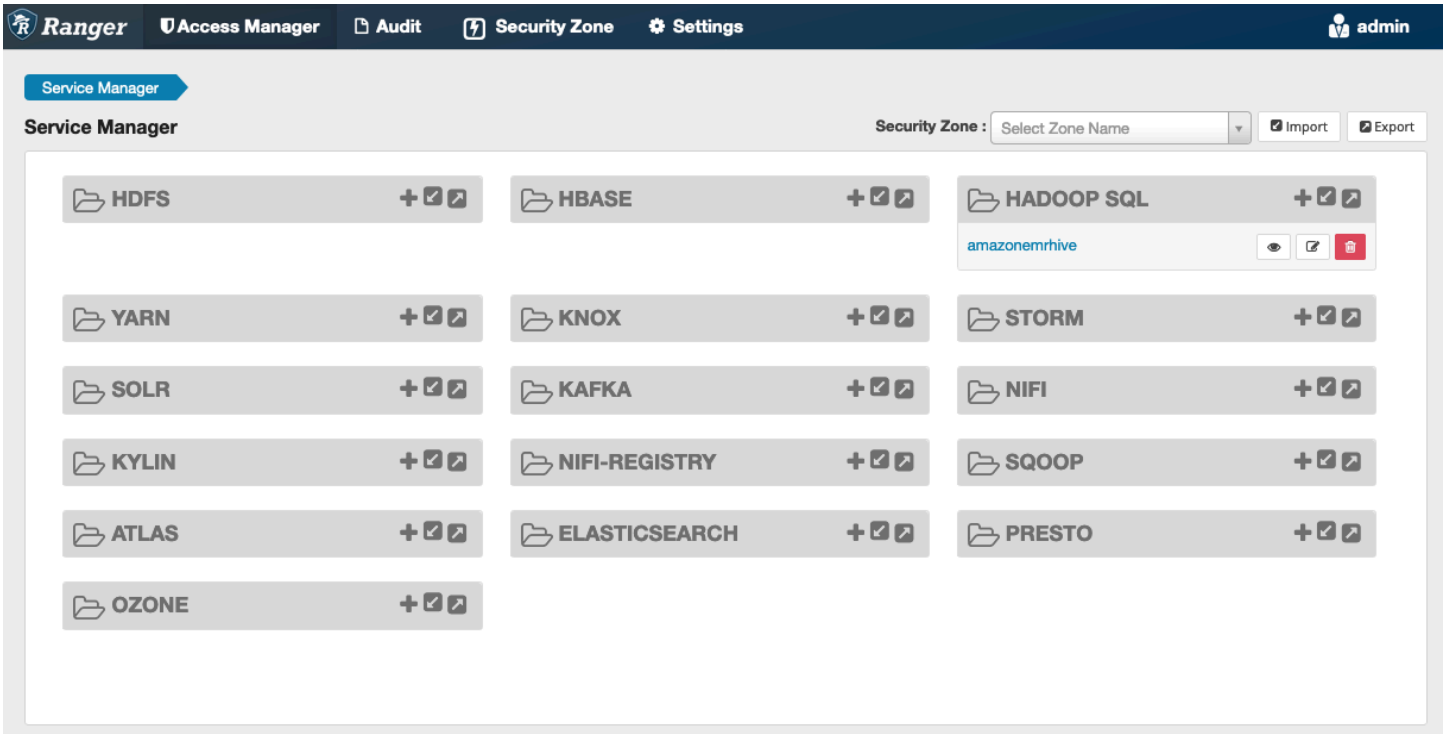


Apache Hive Config Properties 用于通过 HiveServer2 建立与 Apache Ranger Admin 服务器的连接，以在创建策略时实现自动完成。如果您没有持久的 HiveServer2 进程并且可以填充任何信息，则以下属性不需要准确。

- Username (用户名)：为到 HiveServer2 实例的其中一个实例的 JDBC 连接输入用户名。
- Password (密码)：输入上述用户名所对应的密码。
- jdbc.driver.ClassName：输入 Apache Hive 连接的 JDBC 类的类名。可以使用默认值。
- jdbc.url：输入连接到 HiveServer2 时要使用的 JDBC 连接字符串。
- Common Name For Certificate (凭证的公用名称)：凭证中的 CN 字段，用于从客户端插件连接到管理服务器。此值必须与为插件创建的 TLS 凭证中的 CN 字段匹配。



Test Connection (测试连接) 按钮测试是否可以使用上述值成功连接到 HiveServer2 实例。成功创建服务后，服务管理器应如下所示：



注意事项

Hive 元数据服务器

Hive 元数据服务器只能由受信任的引擎访问，特别是 Hive 和 `emr_record_server`，以防未经授权的访问。Hive 元数据服务器也可以通过集群上的所有节点访问。所需的端口 9083 为所有节点提供了主节点的访问权限。

身份验证

默认情况下，Apache Hive 配置为使用 EMR 安全配置中配置的 Kerberos 进行身份验证。HivesServer 2 也可以配置为使用 LDAP 对用户进行身份验证。有关信息，请参阅在多租户 Amazon EMR 集群上为 Hive 实施 LDAP 身份验证。

限制

以下是当前针对 Amazon EMR 5.x 上的 Apache Hive 插件的限制：

- 当前不支持 Hive 角色。不支持 Grant、Revoke 语句。
- 不支持 Hive CLI。JDBC/Beeline 是连接 Hive 的唯一授权方式。
- `hive.server2.builtin.udf.blacklist` 配置中应填充您认为不安全的 UDF。

Apache Spark 插件

Amazon EMR 集成了 EMR RecordServer 以便实现对 SparkSQL 的精细访问控制。EMR 的 RecordServer 是一个特权进程，在启用了 Apache Ranger 的集群的所有节点上运行。当 Spark 驱动程序或执行程序运行 SparkSQL 语句时，所有元数据和数据请求都通过 RecordServer。要详细了解 EMR RecordServer，请参阅 Amazon EMR 组件页面。

主题

- 支持的特征
- 重新部署服务定义以使用 INSERT、ALTER 或 DDL 语句
- 安装服务定义
- 创建 SparkSQL 策略
- 注意事项
- 限制

支持的特征

SQL 语句/Ranger 动作	STATUS	支持的 EMR 版本
SELECT	支持	截至 5.32
SHOW DATABASES	支持	截至 5.32
SHOW COLUMNS	支持	截至 5.32
SHOW TABLES	支持	截至 5.32
SHOW TABLE PROPERTIES	支持	截至 5.32
DESCRIBE TABLE	支持	截至 5.32
INSERT OVERWRITE	支持	自 5.34 和 6.4 起

SQL 语句/Ranger 动作	STATUS	支持的 EMR 版本
INSERT INTO	支持	自 5.34 和 6.4 起
ALTER TABLE	支持	截至 6.4
CREATE TABLE	支持	自 5.35 和 6.7 起
CREATE DATABASE	支持	自 5.35 和 6.7 起
DROP TABLE	支持	自 5.35 和 6.7 起
DROP DATABASE	支持	自 5.35 和 6.7 起
DROP VIEW	支持	自 5.35 和 6.7 起
CREATE VIEW	不支持	

使用 SparkSQL 时支持以下功能：

- 对 Hive Metastore 中的表进行精细访问控制，并且可以在数据库、表和列级别创建策略。
- Apache Ranger 策略可以包括对用户和组的授予策略和拒绝策略。
- 审核事件将提交到 CloudWatch Logs。

重新部署服务定义以使用 INSERT、ALTER 或 DDL 语句

> **Note**
>
> 从 Amazon EMR 6.4 开始，您可以将 Spark SQL 与以下语句结合使用： INSERT INTO （插入）、INSERT OVERWRITE （插入覆盖）或更改 LTER TABLE （表格语句）。从 Amazon EMR 6.7 开始，您可以使用 Spark SQL 创建或删除数据库和表。如果您在部署了 Apache Spark 服务定义的 Apache Ranger 服务器上具有现有安装程序，请使用以下代码重新部署服务定义。

```
# Get existing Spark service definition id calling Ranger REST API and JSON
 processor
curl --silent -f -u <admin_user_login>:<password_for_ranger_admin_user> \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef/
name/amazon-emr-spark' | jq .id

# Download the latest Service definition
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-spark.json

# Update the service definition using the Ranger REST API
curl -u <admin_user_login>:<password_for_ranger_admin_user> -X PUT -d @ranger-
servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/
servicedef/<Spark service definition id from step 1>'
```

安装服务定义

安装 EMR 的 Apache Spark 服务定义要求先安装 Ranger Admin 服务器。请参阅设置 Ranger 管理服务器。

按照以下步骤安装 Apache Spark 服务定义：

步骤 1: SSH 进入 Apache Ranger Admin 服务器

例如：

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

步骤 2: 下载服务定义和 Apache Ranger Admin 服务器插件

在临时目录中，下载服务定义。此服务定义由 Ranger 2.x 版本支持。

```
mkdir /tmp/emr-spark-plugin/
cd /tmp/emr-spark-plugin/
```

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-spark-plugin-2.x.jar
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-spark.json
```

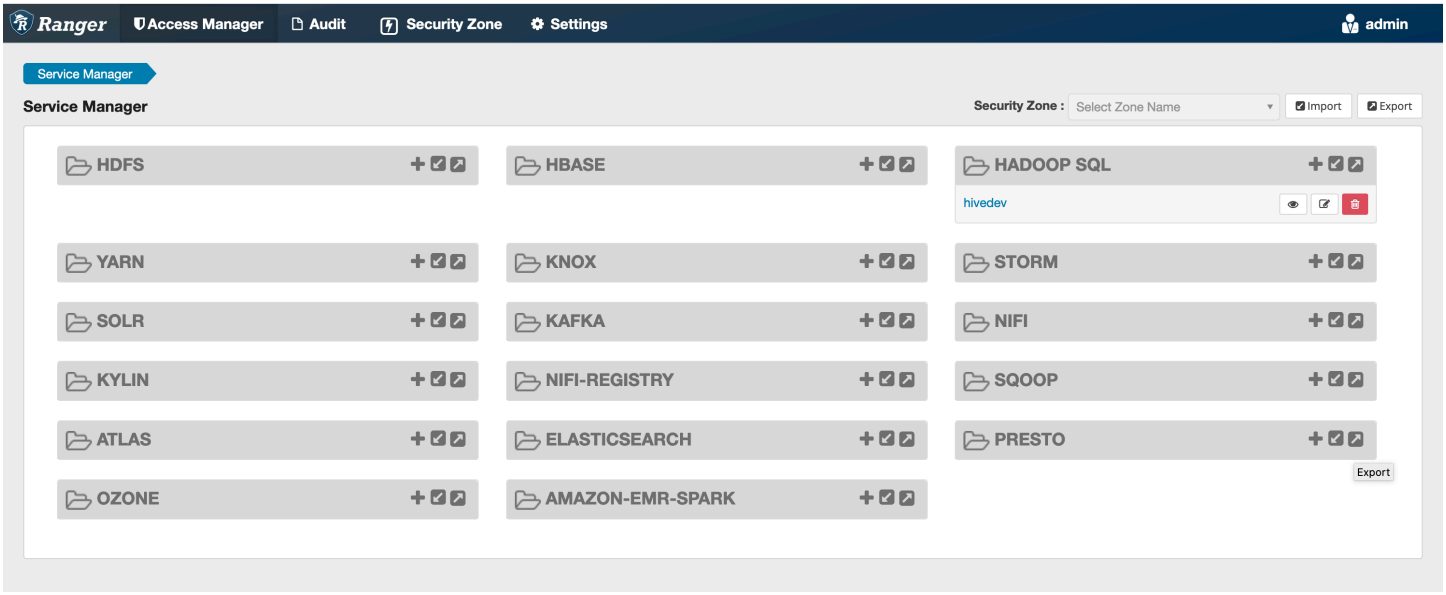
步骤 3: 安装适用于 Amazon EMR 的 Apache Spark 插件

```
export RANGER_HOME=.. # Replace this Ranger Admin's home directory eg /usr/lib/ranger/
ranger-2.0.0-admin
mkdir $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/amazon-emr-spark
mv ranger-spark-plugin-2.x.jar $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/
amazon-emr-spark
```

步骤 4: 注册 Amazon EMR 的 Apache Spark 服务定义

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_* -X
 POST -d @ranger-servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

如果此命令成功运行，您会在 Ranger Admin UI 中看到一个名为“AMAZON-EMR-SPARK”的新服务，如下图所示（显示 Ranger 2.0 版）。

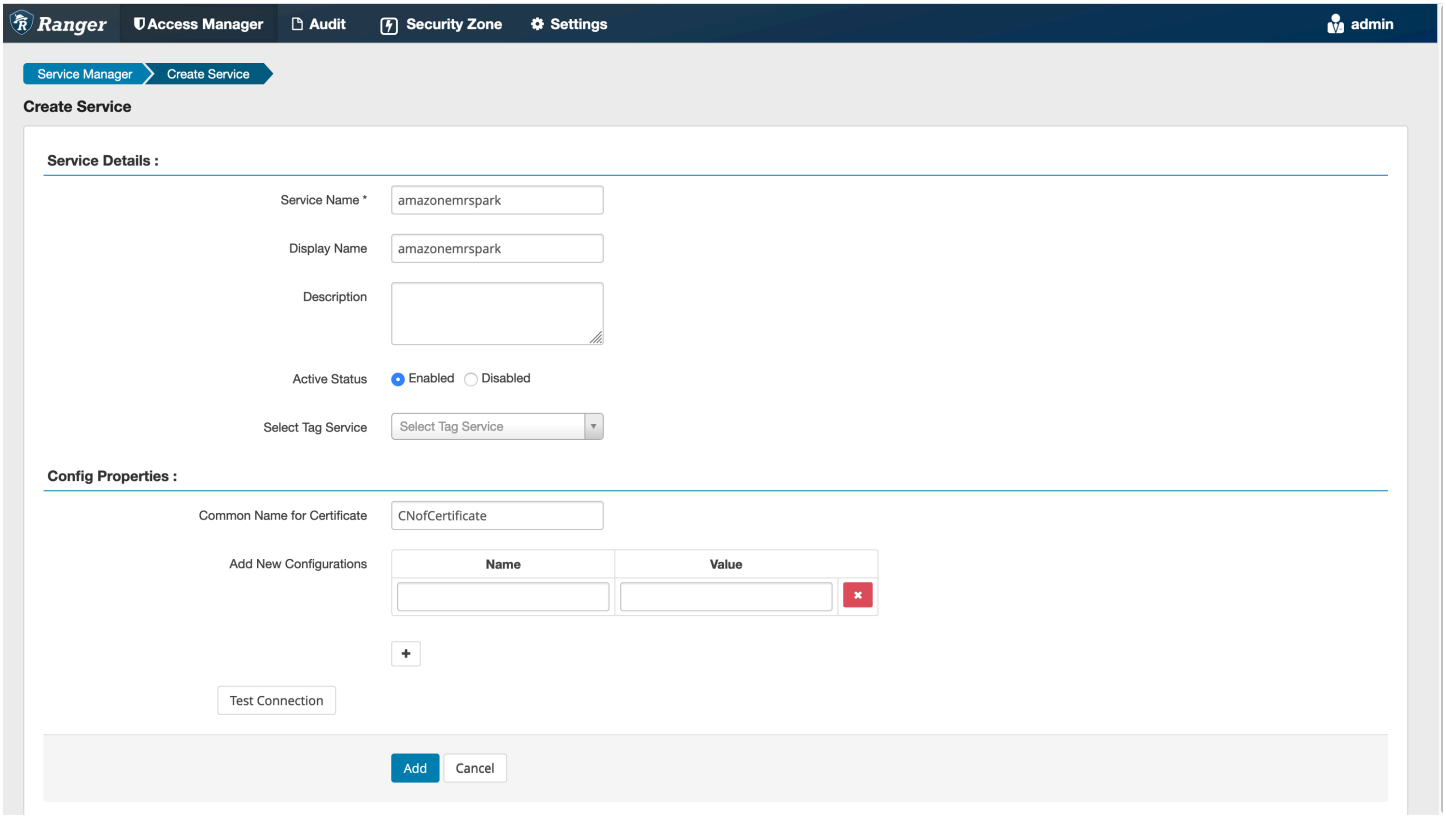


步骤 5: 创建 AMAZON-EMR-SPARK 应用程序的实例

Service Name (服务名称) (如果显示) ：将使用的服务名称。建议的值为 **amazonemrspark**。请记下此服务名称，创建 EMR 安全配置时将要用到。

Display Name (显示名称)：要为此实例显示的名称。建议的值为 **amazonemrspark**。

Common Name For Certificate (凭证的公用名称)：凭证中的 CN 字段，用于从客户端插件连接到管理服务器。此值必须与为插件创建的 TLS 凭证中的 CN 字段匹配。



> **Note**
>
> 此插件的 TLS 凭证应该已在 Ranger Admin 服务器的信任库中注册。有关更多信息，请参阅 TLS 凭证。

创建 SparkSQL 策略

创建新策略时，要填写的字段包括：

Policy Name (策略名称)：此策略的名称。

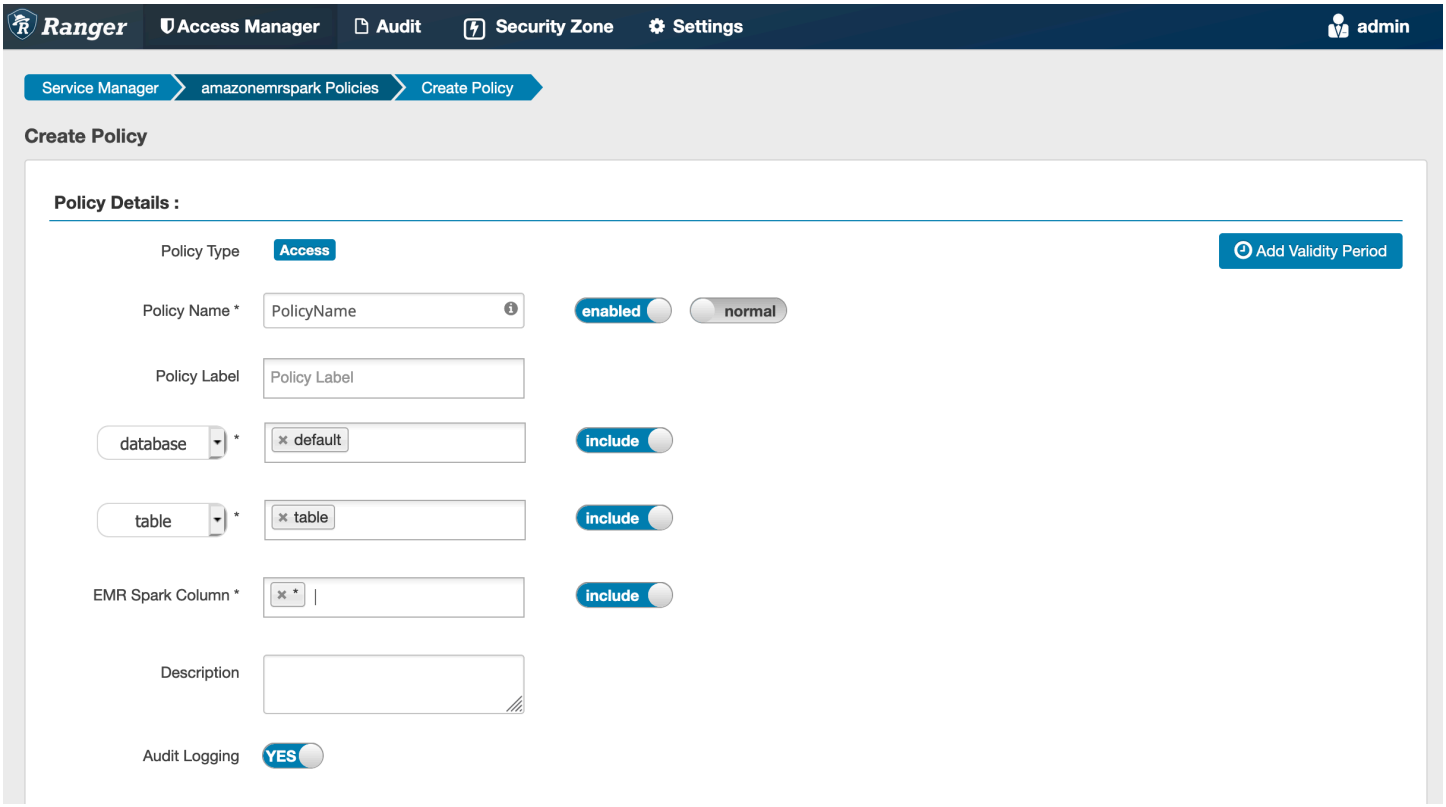
Policy Label (策略标注)：您可以在此策略上放置的标注。

Database (数据库)：应用此策略的数据库。通配符“*”表示所有数据库。

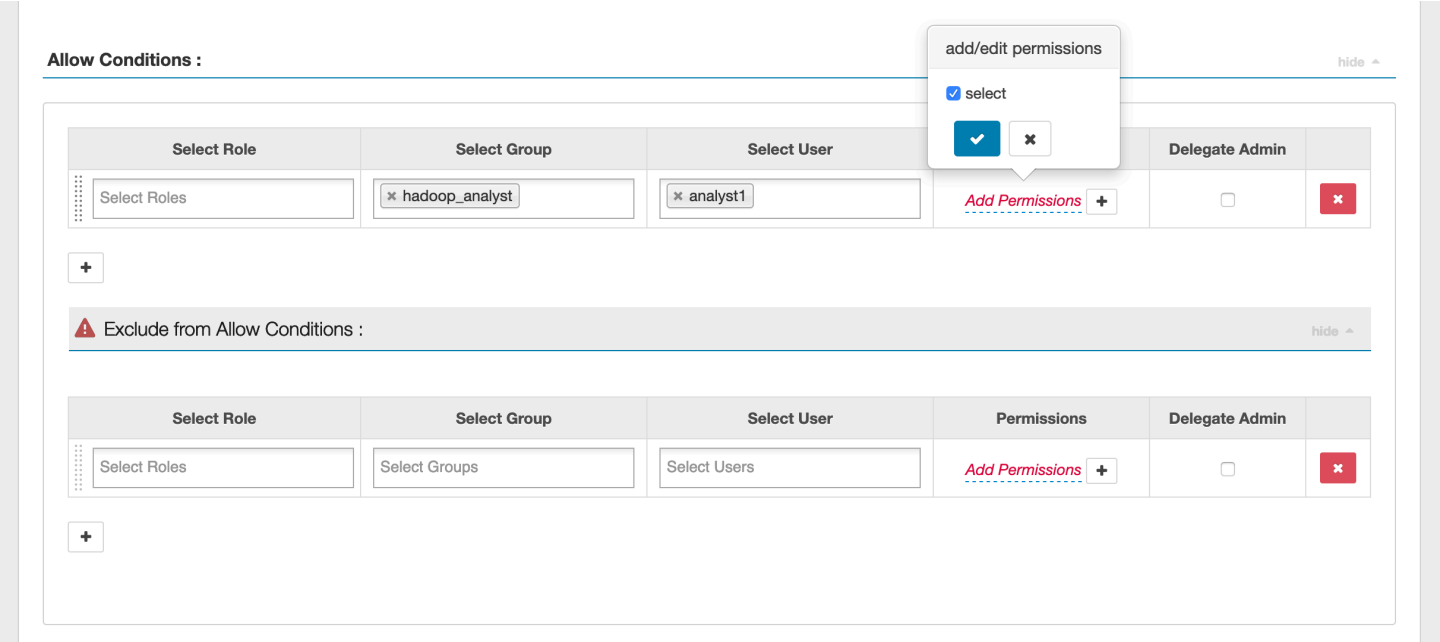
Table（表）：应用此策略的表。通配符“*”表示所有表。

EMR Spark Column (EMR Spark 列)：应用此策略的列。通配符“*”表示所有列。

Description (描述)：策略的描述。



要指定用户和组，请在下方输入用户和组以授予权限。您还可以指定 allow（允许）条件和 deny（拒绝）条件的排除项。



指定允许和拒绝条件后，单击 Save (保存)。

注意事项

EMR 集群中的每个节点都必须能够通过端口 9083 连接到主节点。

限制

以下是当前针对 Apache Spark 插件的限制：

- Record 服务器将始终连接到 Amazon EMR 集群上运行的 HMS。如果需要，将 HMS 配置为连接到远程模式。您不应该将配置值放在 Apache Spark Hive-site.xml 配置文件中。
- 使用 EMR RecordServer 无法读取 CSV 或 Avro 上使用 Spark 数据源创建的表。使用 Hive 创建和写入数据，并使用 Record 读取。
- 不支持 Delta Lake 和 Hudi 表。
- 用户必须有权访问默认数据库。这是 Apache Spark 的要求。
- Ranger Admin 服务器不支持自动完成。
- Amazon EMR 的 SparkSQL 插件不支持行筛选器或数据掩码。
- 将 ALTER TABLE 与 Spark SQL 结合使用时，分区位置必须是表位置的子目录。不支持将数据插入分区位置与表位置不同的分区中。

EMRFS S3 插件

为了更轻松地为多租户集群上的 S3 中的对象提供访问控制，EMRFS S3 插件在通过 EMRFS 访问 S3 中的数据时提供对这些数据的访问控制。您可以允许在用户和组级别访问 S3 资源。

为此，当您的应用程序尝试访问 S3 中的数据时，EMRFS 会向私有代理进程发送凭证请求，在该进程中该请求针对 Apache Ranger 插件进行身份验证和授权。如果请求获得授权，则私有代理将代入 Apache Ranger Engines 的 IAM 角色，并使用受限策略生成仅有权访问允许访问的 Ranger 策略的凭证。然后，凭证将传回 EMRFS 以访问 S3。

主题

- 支持的特征
- 安装服务配置
- 创建 EMRFS S3 策略
- EMRFS S3 策略使用说明
- 限制

支持的特征

EMRFS S3 插件提供存储级别授权。可以创建策略，以便向用户和组提供对 S3 存储桶和前缀的访问权限。授权仅针对 EMRFS 进行。

安装服务配置

安装 Trino 服务定义要求设置 Ranger Admin 服务器。要设置 Ranger Admin 服务器，请参阅 设置 Ranger 管理服务器。

按照以下步骤安装 EMRFS 服务定义。

步骤 1：SSH 进入 Apache Ranger Admin 服务器。

例如：

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

步骤 2：下载 Amazon EMR 服务定义和 Apache Ranger Admin 服务器插件。

在临时目录中，下载 Amazon EMR 服务定义。此服务定义由 Ranger 2.x 版本支持。

```
mkdir /tmp/emr-emrfs-plugin/
```

```
cd /tmp/emr-emrfs-plugin/

wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-emrfs.json
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-emr-emrfs-plugin-2.x.jar
```

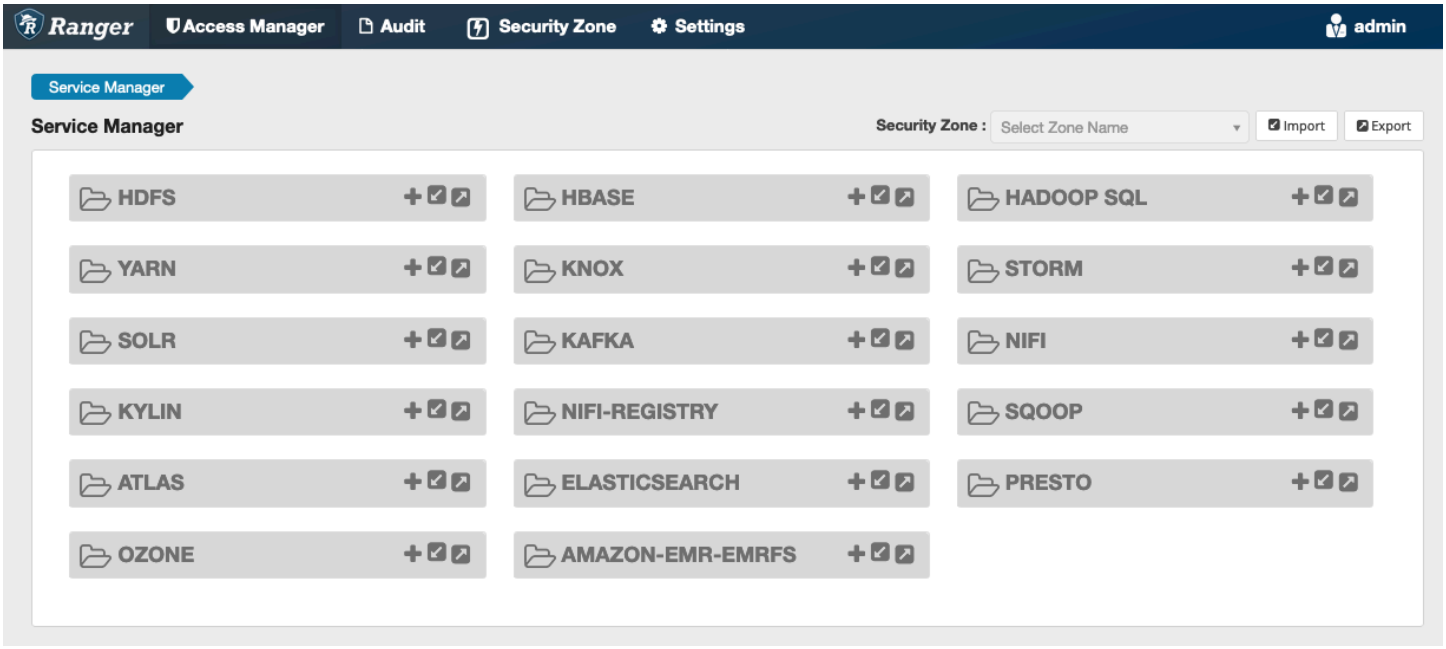
步骤 3：安装适用于 Amazon EMR 的 Apache EMRFS S3 插件。

```
export RANGER_HOME=.. # Replace this Ranger Admin's home directory eg /usr/lib/ranger/
ranger-2.0.0-admin
mkdir $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/amazon-emr-emrfs
mv ranger-emr-emrfs-plugin-2.x.jar $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-
plugins/amazon-emr-emrfs
```

步骤 4：注册 EMRFS S3 服务定义。

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_* -X
 POST -d @ranger-servicedef-amazon-emr-emrfs.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

如果此命令成功运行，您会在 Ranger Admin UI 中看到一个名为“AMAZON-EMR-S3”的新服务，如下图所示（显示 Ranger 2.0 版）。



步骤 5：创建 AMAZON-EMR-EMRFS 应用程序实例。

创建服务定义的实例。

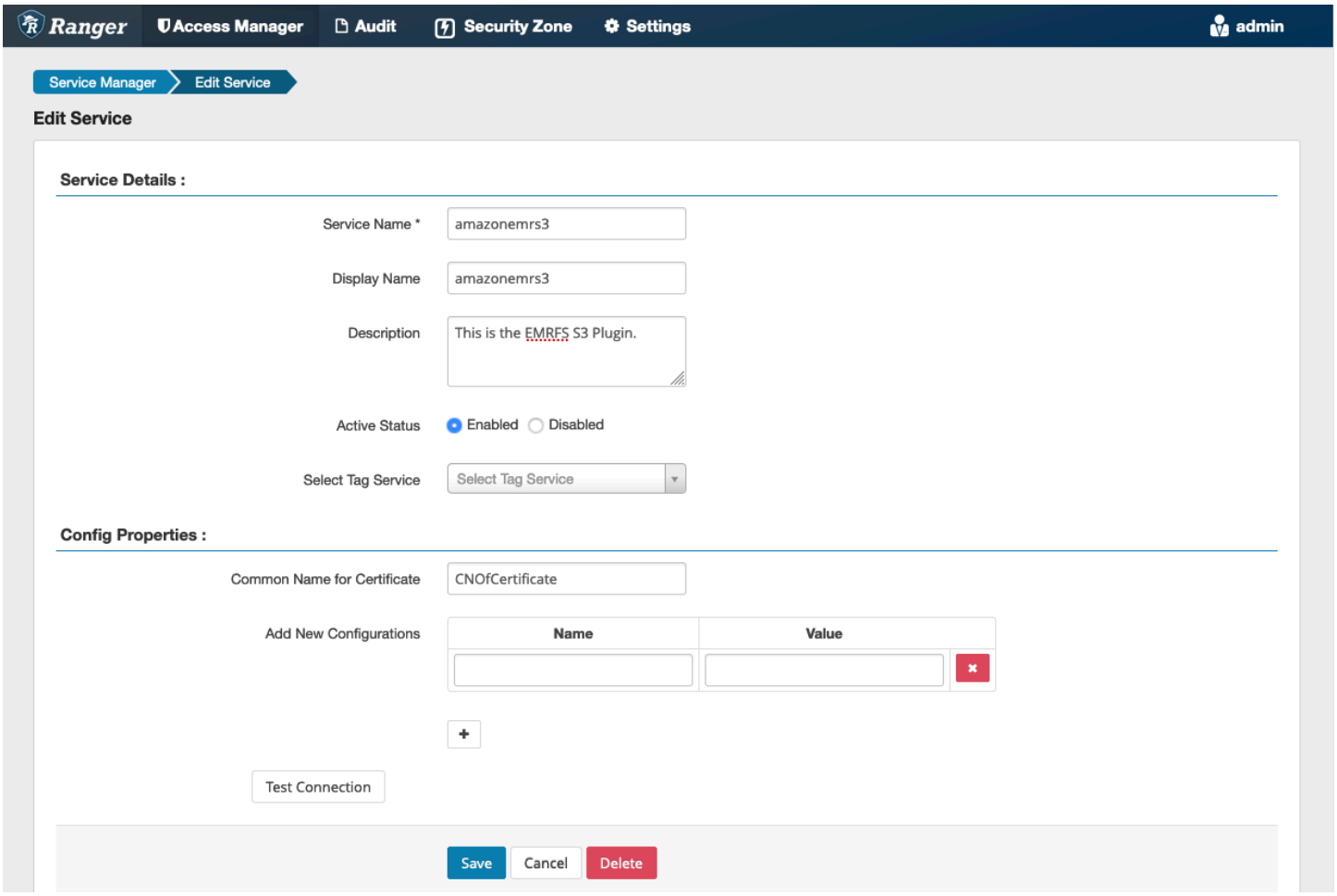
- 点击 AMAZON-EMR-EMRFS 旁边的 +。

填写以下字段：

Service Name (服务名称)（如果显示）：建议的值为 **amazonemrspark**。请记下此服务名称，创建 EMR 安全配置时将要用到。

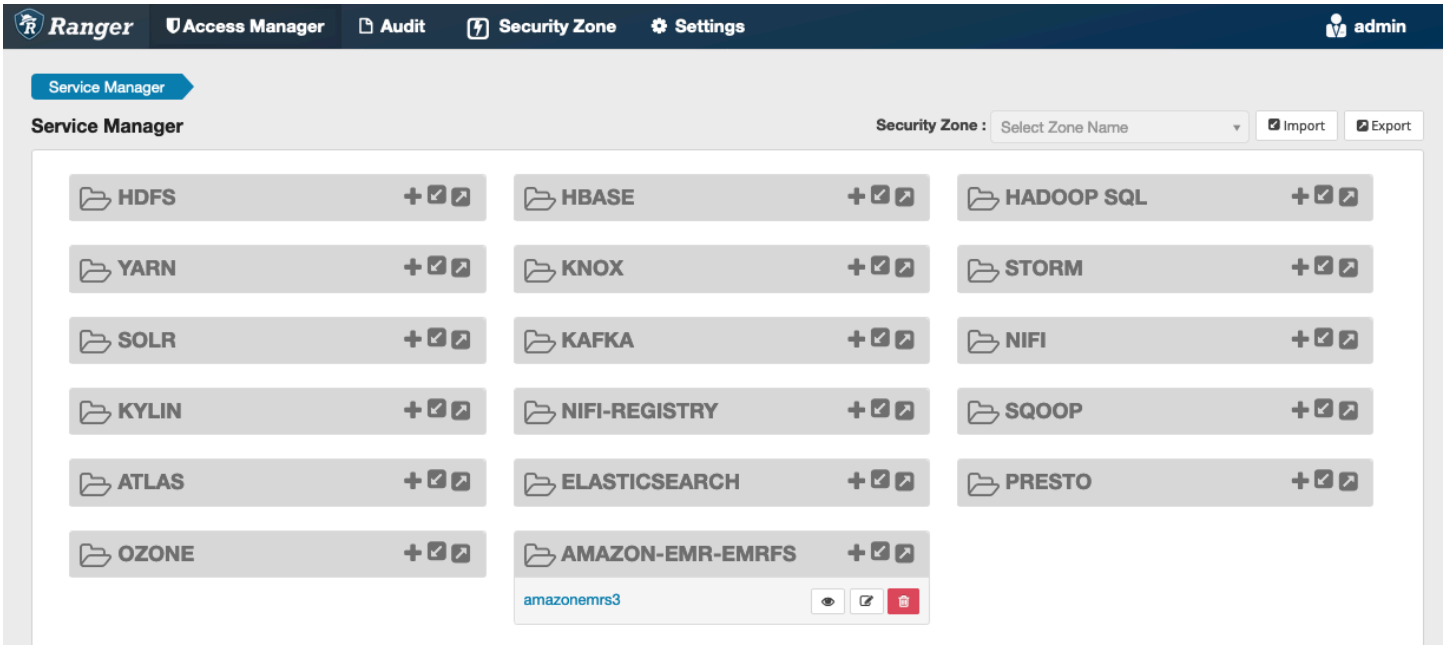
显示名称：为此服务显示的名称。建议的值为 **amazonemrspark**。

Common Name For Certificate (凭证的公用名称)：凭证中的 CN 字段，用于从客户端插件连接到管理服务器。此值必须与为插件创建的 TLS 证书中的 CN 字段匹配。



> **Note**
>
> 此插件的 TLS 凭证应该已在 Ranger Admin 服务器的信任库中注册。有关更多信息，请参阅 TLS 凭证。

创建服务时，服务管理器包含“AMAZON-EMR-EMRFS”，如下图所示。



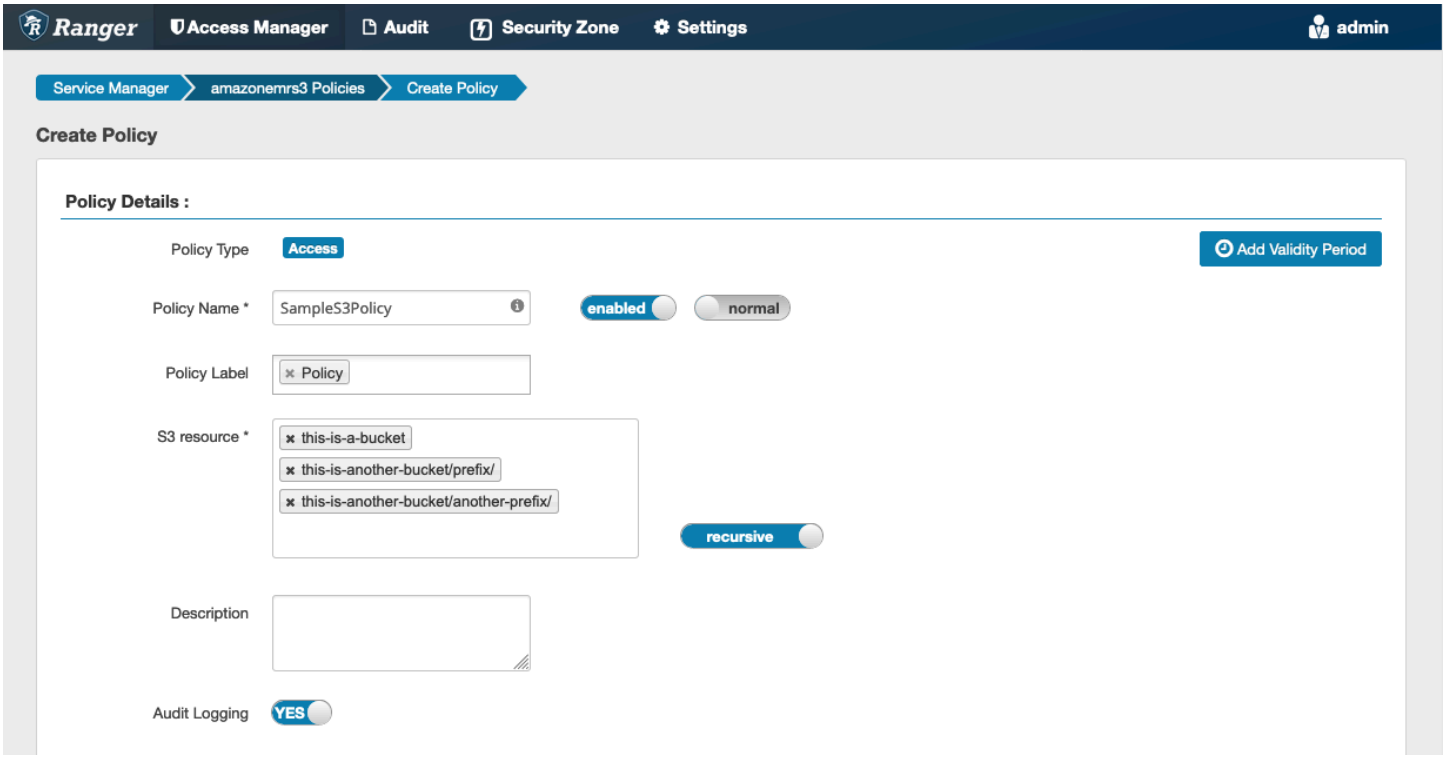
创建 EMRFS S3 策略

要创建新策略，请在服务管理器的 Create Policy（创建策略）页面填写以下字段。

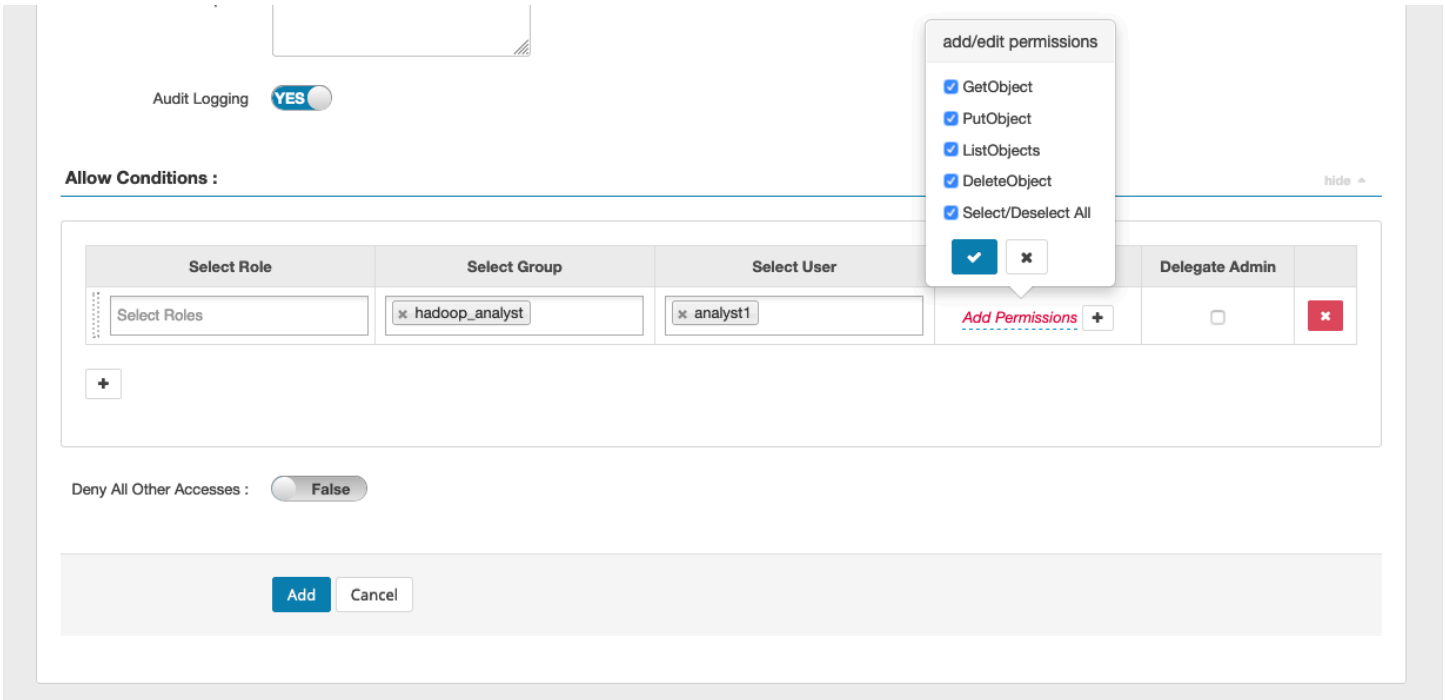
Policy Name (策略名称)：此策略的名称。

Policy Label (策略标注)：您可以在此策略上放置的标注。

S3 Resource (S3 资源)：以存储桶和可选前缀开头的资源。有关最佳实践的信息，请参阅EMRFS S3 策略使用说明。Ranger Admin 服务器中的资源不应包含 **`s3://`**、**`s3a://`** 或 **`s3n://`**。



您可以指定要授予权限的用户和组。您还可以指定 allow（允许）条件和 deny（拒绝）条件的排除项。



> **Note**
>
> 每个策略最多允许三个资源。在 EMR 集群上使用此策略时，添加三个以上的资源可能会导致错误。添加三个以上的策略将显示有关策略限制的提醒。

EMRFS S3 策略使用说明

在 Apache Renger 中创建 S3 策略时，需要谨记一些使用注意事项。

对多个 S3 对象的权限

您可以使用递归策略和通配符表达式为具有公共前缀的多个 S3 对象授予权限。递归策略为所有具有公共前缀的对象授予权限。通配符表达式选择多个前缀。它们共同为具有多个公共前缀的所有对象授予权限，如以下示例所示。

Example 使用递归策略

假设您希望获得列出 S3 存储桶中所有 Parquet 文件的权限，该存储桶的组织方式如下。

```
s3://sales-reports/americas/
    +- year=2000
    |      +- data-q1.parquet
    |      +- data-q2.parquet
    +- year=2019
    |      +- data-q1.json
    |      +- data-q2.json
    |      +- data-q3.json
    |      +- data-q4.json
    |
    +- year=2020
    |      +- data-q1.parquet
    |      +- data-q2.parquet
    |      +- data-q3.parquet
    |      +- data-q4.parquet
    |      +- annual-summary.parquet
    +- year=2021
```

首先，考虑带有前缀 `s3://sales-reports/americas/year=2000` 的 Parquet 文件。您可以通过两种方式向所有这些对象授予 GetObject 权限：

使用非递归策略：一种选择是使用两个单独的非递归策略，一个用于目录，另一个用于文件。

第一个策略向前缀 s3://sales-reports/americas/year=2020 授予权限（无尾随 /）。

```
- S3 resource = "sales-reports/americas/year=2000"
- permission = "GetObject"
- user = "analyst"
```

第二个策略使用通配符表达式授予所有带前缀 sales-reports/americas/year=2020/ 的文件的权限（请注意尾随 /）。

```
- S3 resource = "sales-reports/americas/year=2020/*"
- permission = "GetObject"
- user = "analyst"
```

使用递归策略：更方便的替代方法是使用单个递归策略并对前缀授予递归权限。

```
 - S3 resource = "sales-reports/americas/year=2020"
 - permission = "GetObject"
 - user = "analyst"
 - is recursive = "True"
```

到目前为止，只有带有 s3://sales-reports/americas/year=2000 前缀的 Parquet 文件被包括在内。您现在还可以通过引入通配符表达式，将具有不同前缀 s3://sales-reports/americas/year=2020 的 Parquet 文件包含到相同的递归策略中，如下所示。

```
 - S3 resource = "sales-reports/americas/year=20?0"
 - permission = "GetObject"
 - user = "analyst"
 - is recursive = "True"
```

PutObject 和 DeleteObject 权限的策略

为 EMRFS 上的文件编写 PutObject 和 DeleteObject 权限的策略需要特别注意，因为与 GetObject 权限不同，它们需要为前缀授予额外的递归权限。

Example PutObject 和 DeleteObject 权限的策略

例如，删除文件 annual-summary.parquet 不仅需要对实际文件的 DeleteObject 权限。

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "DeleteObject"
```

```
- user = "analyst"
```

还需要一个策略，以为其前缀授予递归 GetObject 和 PutObject 权限。

同样，修改文件 annual-summary.parquet 不仅需要对实际文件的PutObject 权限。

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "PutObject"
- user = "analyst"
```

还需要一个策略，以为其前缀授予递归 GetObject 权限。

```
- S3 resource = "sales-reports/americas/year=2020"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

策略中的通配符

有两个区域可以指定通配符。指定 S3 资源时，可以使用“*”和“?”。“*”提供针对 S3 路径的匹配，并匹配前缀后的所有内容。例如以下策略。

```
S3 resource = "sales-reports/americas/*"
```

这与以下 S3 路径匹配。

```
sales-reports/americas/year=2020/
sales-reports/americas/year=2019/
sales-reports/americas/year=2019/month=12/day=1/afile.parquet
sales-reports/americas/year=2018/month=6/day=1/afile.parquet
sales-reports/americas/year=2017/afile.parquet
```

“?”通配符匹配任意单个字符。例如，策略。

```
S3 resource = "sales-reports/americas/year=201?/"
```

这与以下 S3 路径匹配。

```
sales-reports/americas/year=2019/
sales-reports/americas/year=2018/
```

```
sales-reports/americas/year=2017/
```

用户中的通配符

为用户分配访问权限时，有两个内置通配符。第一个是为所有用户提供访问权限的“{USER}”通配符。第二个通配符是“{HOSTER}”，它提供对特定对象所有者的访问权限或直接访问权限。但是，目前尚不支持“{USER}”通配符。

限制

以下是 EMRFS S3 插件的当前限制：

- Apache Ranger 策略最多可以有三个策略。
- 对 S3 的访问必须通过 EMRFS 完成，并且可以与 Hadoop 相关的应用程序配合使用。不支持以下内容：

  – Boto3 库

  – Amazon SDK 和 Amazon CLI

  – S3A 开源连接器
- 不支持 Apache Ranger 拒绝策略。
- 目前不支持使用具有 CSE-KMS 加密的密钥在 S3 上进行操作。
- 不支持跨区域支持。
- 不支持 Apache Ranger 的安全区功能。使用安全区功能定义的访问控制限制不适用于您的 Amazon EMR 集群。
- Hadoop 用户不会生成任何审计事件，因为 Hadoop 始终访问 EC2 实例配置文件。
- 建议您禁用 Amazon EMR 一致性视图。S3 具有强一致性，因此不再需要它。有关更多信息，请参阅 Amazon S3 强一致性。
- EMRFS S3 插件会进行大量的 STS 调用。建议您对开发账户进行负载测试并监控 STS 调用量。还建议您发出 STS 请求以提高 AssumeRole 服务限制。

Trino 插件

Trino（原 PrestoSQL）是一个 SQL 查询引擎，您可以用来对 HDFS、对象存储、关系数据库和 NoSQL 数据库等数据源运行查询。它不再要求将数据迁移到中心位置，并允许您随时随地查询数据。Amazon EMR 提供了一个 Apache Ranger 插件，为 Trino 提供精细访问控制。该插件与开源 Apache Ranger Admin 服务器版本 2.0 及更高版本兼容。

主题

- 支持的特征
- 安装服务配置
- 创建 Trino 策略
- 注意事项
- 限制

支持的特征

Trino 查询引擎受精细访问控制的保护，Amazon EMR 上 Trino 的 Apache Ranger 插件支持该引擎的所有功能。这包括数据库、表、列级别的访问控制以及行筛选和数据掩码。Apache Ranger 策略可以包括对用户和组的授予策略和拒绝策略。审核事件也将提交到 CloudWatch Logs。

安装服务配置

安装 Trino 服务定义要求设置 Ranger Admin 服务器。要设置 Ranger Admin 服务器，请参阅 设置 Ranger 管理服务器。

按照以下步骤安装 Trino 服务定义。

1. SSH 进入 Apache Ranger Admin 服务器。

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

2. 卸载 Presto 服务器插件（如有）。运行以下命令。如果出现“Service not found（未找到服务）”错误，则意味着您的服务器上未安装 Presto 服务器插件。继续执行下一步骤。

```
curl -f -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin
 user_**_>_* -X DELETE -k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/
v2/api/servicedef/name/presto'
```

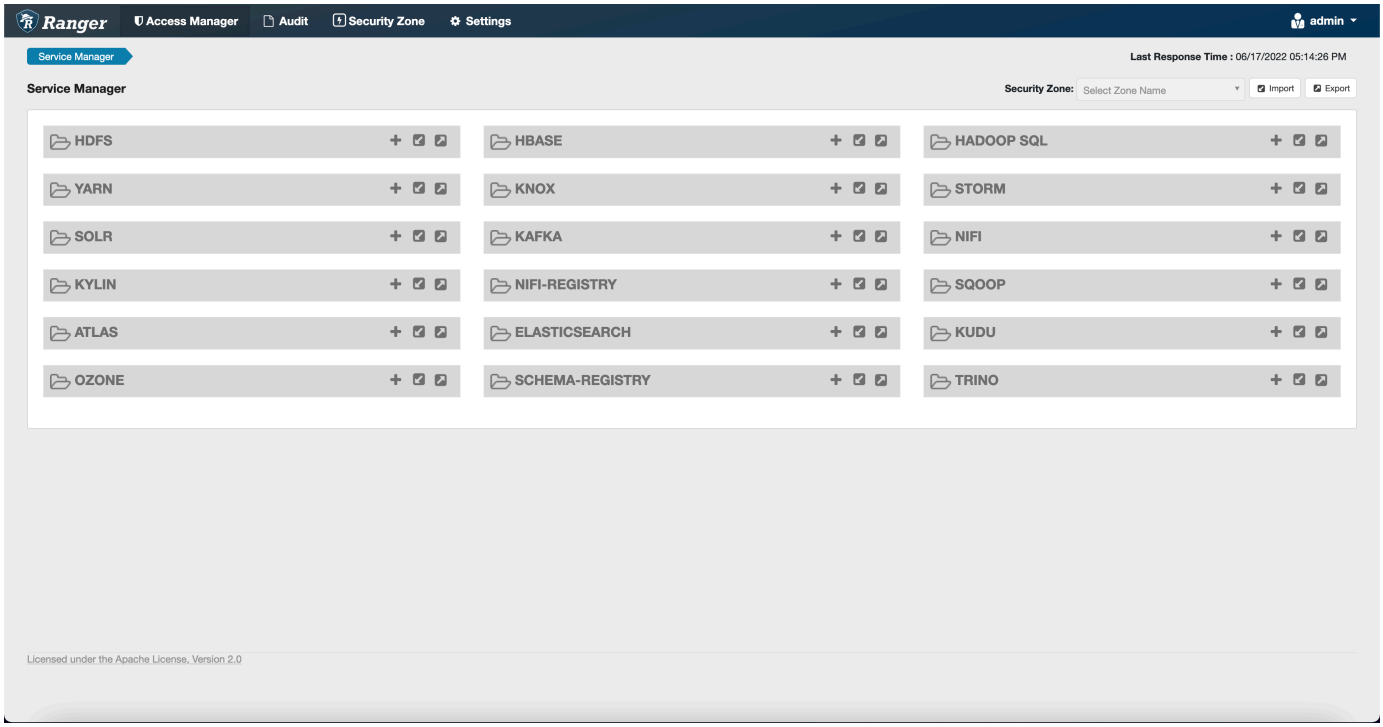
3. 下载服务定义和 Apache Ranger Admin 服务器插件。在临时目录中，下载服务定义。此服务定义由 Ranger 2.x 版本支持。

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-trino.json
```

4. 注册 Amazon EMR 的 Apache Trino 服务定义。

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_*
 -X POST -d @ranger-servicedef-amazon-emr-trino.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

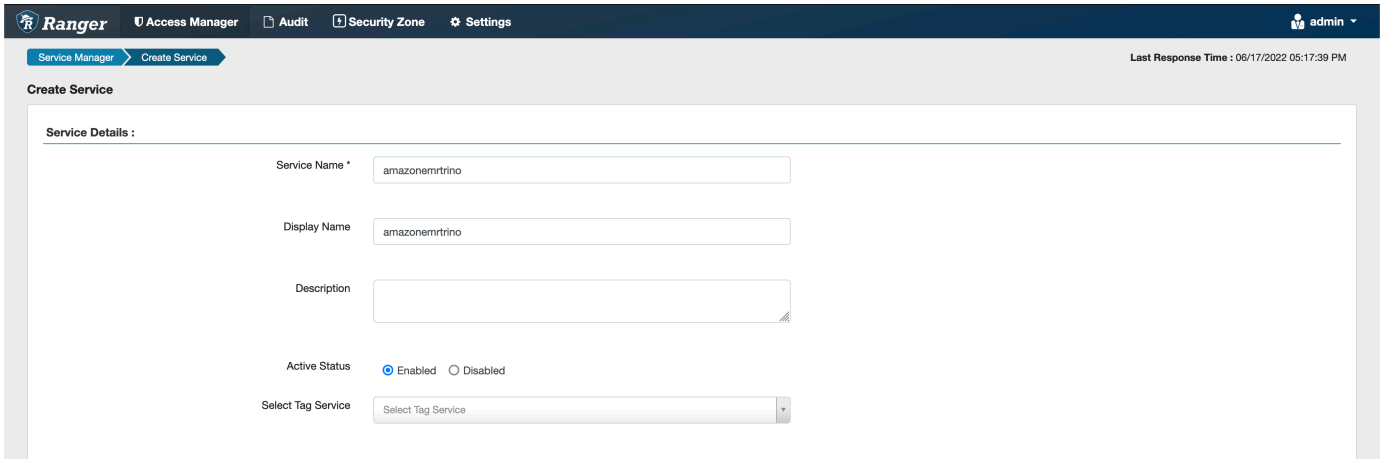
如果此命令成功运行，您会在 Ranger Admin UI 中看到一个名为 TRINO 的新服务，如下图所示。



5. 创建 TRINO 应用程序的实例，输入以下信息。

   Service Name（服务名称）：您将使用的服务名称。建议的值为 amazonemrtrino。请记下此服务名称，创建 Amazon EMR 安全配置时将要用到。

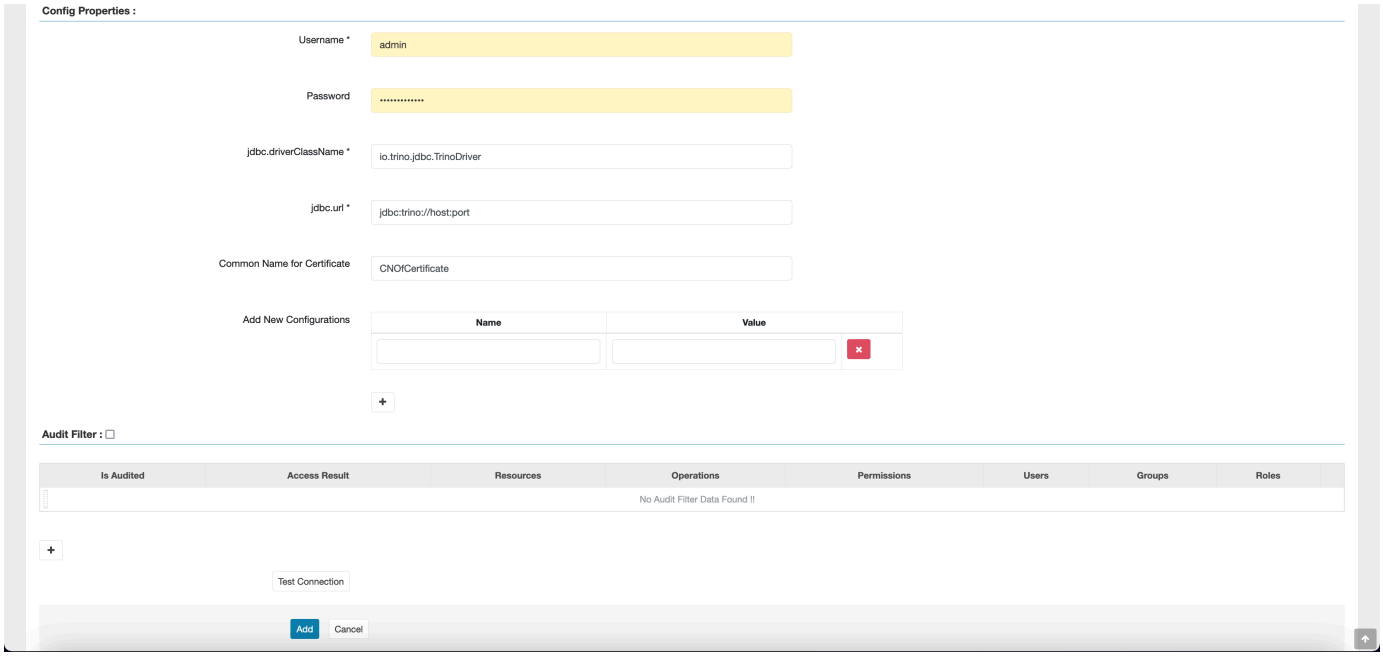
   Display Name (显示名称)：要为此实例显示的名称。建议的值为 amazonemrtrino。



jdbc.driver.ClassName：Trino 连接的 JDBC 类的类名。您可以使用默认值。

jdbc.url：连接到 Trino 协调器时要使用的 JDBC 连接字符串。

Common Name For Certificate (凭证的公用名称)：凭证中的 CN 字段，用于从客户端插件连接到管理服务器。此值必须与为插件创建的 TLS 凭证中的 CN 字段匹配。



请注意，此插件的 TLS 凭证应该已在 Ranger Admin 服务器的信任库中注册。有关更多信息，请参阅 TLS 证书。

创建 Trino 策略

创建新策略时，请填写以下字段。

Policy Name (策略名称)：此策略的名称。

Policy Label (策略标注)：您可以在此策略上放置的标注。

Catalog（目录）：应用此策略的目录。通配符“*”表示所有目录。

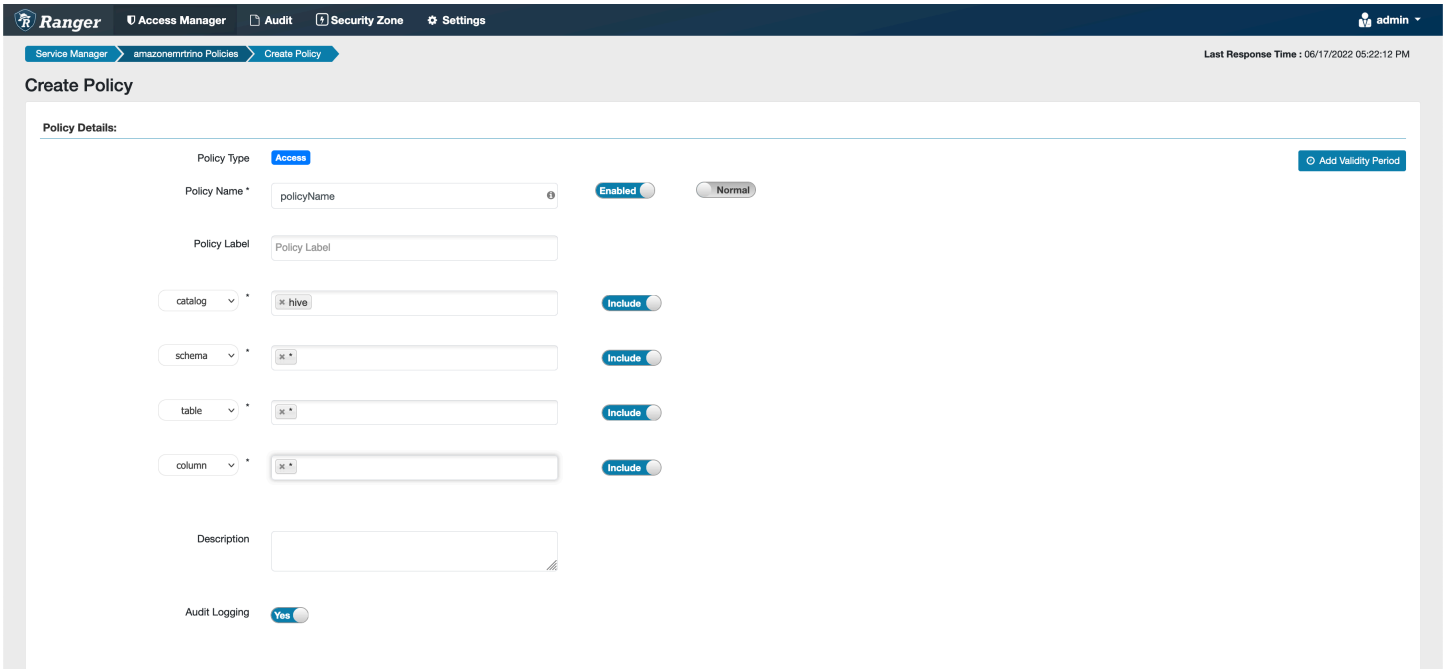
Schema（架构）：应用此策略的架构。通配符“*”表示所有架构。

Table（表）：应用此策略的表。通配符“*”表示所有表。

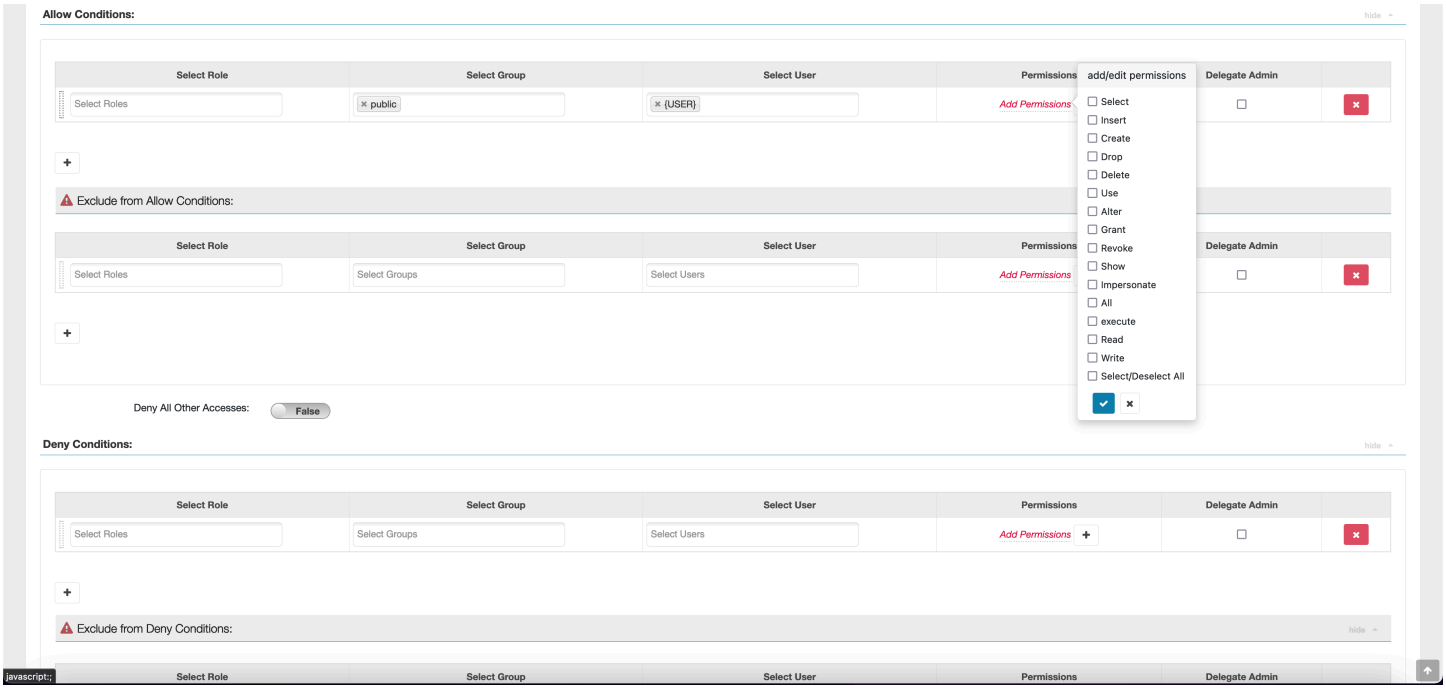
Column（列）：应用此策略的列。通配符“*”表示所有列。

Description (描述)：策略的描述。

还存在其他类型的策略，适用于 Trino User（Trino 用户）（用于用户模拟访问）、Trino System/Session Property（Trino 系统/会话属性）（用于更改引擎系统或会话属性）、Functions/Procedures（功能/过程）（用于允许功能或过程调用）和 URL（用于向数据位置上的引擎授予读/写访问权限）。



要向特定用户和组授予权限，请输入用户和组。您还可以指定 allow（允许）条件和 deny（拒绝）条件的排除项。



指定允许和拒绝条件后，选择 Save（保存）。

注意事项

在 Apache Renger 中创建 Trino 策略时，需要谨记一些使用注意事项。

Hive 元数据服务器

Hive 元数据服务器只能由受信任的引擎（特别是 Trino 引擎）访问，以防未经授权的访问。Hive 元数据服务器也可以通过集群上的所有节点访问。所需的端口 9083 为所有节点提供了主节点的访问权限。

身份验证

默认情况下，Trino 配置为使用 Amazon EMR 安全配置中配置的 Kerberos 进行身份验证。

需要传输中加密

Trino 插件要求您在 Amazon EMR 安全配置中启用传输中加密。要启用加密，请参阅 传输中加密。

限制

以下是 Trino 插件的当前限制：

- Ranger Admin 服务器不支持自动完成。

Apache Ranger 疑难解答

以下是一些与使用 Apache Ranger 相关的常见问题。

建议

- 使用单个主节点集群进行测试：单节点主集群比多节点集群配置更快，这样可以减少每次测试迭代的时间。
- 在集群上设置开发模式。启动 EMR 集群时，请将 `--additional-info"` 参数设置为：

  ```
  '{"clusterType":"development"}'
  ```

  此参数只能通过 Amazon CLI 或 Amazon SDK 设置，不能通过 Amazon EMR 控制台设置。如果设置了此标志，并且主节点无法配置，则 Amazon EMR 服务会在将集群停用之前使集群保持活动状态一段时间。这个时间有助于在集群终止前探测各种日志文件。

EMR 集群预置失败

Amazon EMR 集群启动失败的原因有多种。以下是诊断问题的几种方式。

检查 EMR 调配日志

Amazon EMR 使用 Puppet 在集群上安装和配置应用程序。查看日志将提供有关在集群的配置阶段是否存在任何错误的详细信息。如果日志被配置为推送到 S3，则可以在集群或 S3 上访问日志。

日志存储在磁盘上的 `/var/log/provision-node/apps-phase/0/{UUID}/puppet.log` 和 `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/provision-node/apps-phase/0/{UUID}/puppet.log.gz`.

常见错误消息

错误消息	原因
Puppet (err): Systemd start for emr-record-server failed! journalctl log for emr-record-server:	EMR 记录服务器启动失败。请参阅下面的 EMR 记录服务器日志。

错误消息	原因
Puppet (err): Systemd start for emr-record-server failed! journalctl log for emrsecretagent:	EMR SecretAgent 启动失败。请参阅下面的检查 SecretAgent 日志。
/Stage[main]/Ranger_plugins::Ranger_hive_plugin/Ranger_plugins::Prepare_two_way_tls[configure 2-way TLS in Hive plugin]/Exec[create keystore and truststore for Ranger Hive plugin]/returns (notice): 140408606197664:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c:707:Expecting: ANY PRIVATE KEY	Secrets Manager 中用于 Apache Ranger 插件凭证的私有 TLS 凭证格式不正确或不是私有凭证。请参阅 TLS 凭证 了解凭证格式。
/Stage[main]/Ranger_plugins::Ranger_s3_plugin/Ranger_plugins::Prepare_two_way_tls[configure 2-way TLS in Ranger s3 plugin]/Exec[create keystore and truststore for Ranger amazon-emr-s3 plugin]/returns (notice): An error occurred (AccessDeniedException) when calling the GetSecretValue operation: User: arn:aws:sts::XXXXXXXXXXX:assumed-role/EMR_EC2_DefaultRole/i-XXXXXXXXXXXX is not authorized to perform: secretsmanager:GetSecretValue on resource: arn:aws:secretsmanager:us-east-1:XXXXXXXXXX:secret:AdminServer-XXXXX	EC2 实例配置文件角色没有从 Secrets Agent 检索 TLS 凭证的权限。

检查 SecretAgent 日志

SecretAgent 日志位于 EMR 节点上的 `/emr/secretagent/log/` 中，或位于 S3 中的 `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/daemons/secretagent/` 目录中。

常见错误消息

错误消息	原因
Exception in thread "main" com.amazonaws.services.securitytoken.model.AWSSecurityTokenServiceException: User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/EMR_EC2_DefaultRole/i-XXXXXXXXXXXXXXXX is not authorized to perform: sts:AssumeRole on resource: arn:aws:iam::XXXXXXXXXXXX:role/*RangerPluginDataAccessRole* (Service: AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied; Request ID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX; Proxy: null)	上述异常意味着 EMR EC2 实例配置文件角色无权代入角色 RangerPluginDataAccessRole。请参阅用于与 Apache Ranger 本地集成的 IAM 角色。
ERROR qtp54617902-149: Web App Exception Occurred javax.ws.rs.NotAllowedException: HTTP 405 Method Not Allowed	这些错误可以忽略。

检查记录服务器日志（适用于 SparkSQL）

EMR 记录服务器日志可在 EMR 节点上的如下位置找到：/var/log/emr-record-server/，或者可以在 S3 中的以下目录中找到：s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/daemons/emr-record-server/。

常见错误消息

错误消息	原因
InstanceMetadataServiceResourceFetcher:105 - [] Fail to retrieve token com.amazonaws.SdkClientException: Failed to connect to service endpoint	EMR SecretAgent 未能启动或出现问题。检查 SecretAgent 日志中的错误和 puppet 脚本以确定是否存在任何配置错误。

查询意外失败

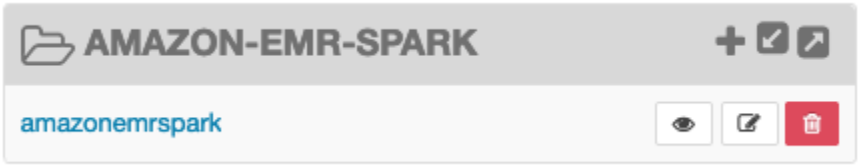
检查 Apache Ranger 插件日志（Apache Hive、EMR RecordServer、EMR SecretAgent 等日志）

此部分在与 Ranger 插件集成的所有应用程序中通用，例如 Apache Hive、EMR RecordServer 和 EMR SecretAgent。

常见错误消息

错误消息	原因
ERROR PolicyRefresher:272 - [] PolicyRefresher(serviceName=policy-repository): failed to find service. Will clean up local cache of policies (-1)	此错误消息意味着您在 EMR 安全配置中提供的服务名称与 Ranger Admin 服务器中的服务策略存储库不匹配。

如果在 Ranger Admin 服务器中，您的 AMAZON-EMR-SPARK 服务如下所示，那么您应该输入 **`amazonemrspark`** 作为服务名称。



使用安全组控制网络流量

安全组充当您的集群中 EC2 实例的虚拟防火墙，以控制入站和出站流量。每个安全组都有一组规则来控制入站流量，并有另外一组规则来控制出站流量。有关更多信息，请参阅《适用于 Linux 实例 Amazon EC2 用户指南》 中的适用于 Linux 实例的 Amazon EC2 安全组。

您将两种类型的安全组与 Amazon EMR 结合使用：Amazon EMR 管理的安全组和其它安全组。

每个集群都有与其关联的托管安全组。您可以使用 Amazon EMR 创建的默认托管式安全组，或指定自定义托管式安全组。无论哪种方式，Amazon EMR 都会向托管安全组自动添加集群在集群实例和Amazon服务之间通信所需的规则。

其它安全组是可选的。除了托管安全组以外，您还可以指定这些规则，以定制访问权限。其它安全组仅包含您定义的规则。Amazon EMR 不会对其进行修改。

Amazon EMR 在托管安全组中创建的规则允许集群在内部组件间进行通信。要允许用户和应用程序从该集群外部访问某集群，您可以在托管安全组中编辑规则和/或使用其它规则创建其它安全组。

> **Important**
>
> 在托管安全组中编辑规则可能会造成意外后果。您可能无意中阻止了集群要正常运行所需的流量，并且由于无法访问节点而导致错误。请在实施前仔细计划并测试安全组配置。

您只能在创建集群时指定安全组。它们不能在集群正在运行时被添加到集群或集群实例，但您可以从现有安全组编辑、添加和删除规则。规则在您保存后立即生效。

默认情况下，安全组是限制性的。除非添加了允许流量的规则，否则流量会被拒绝。如果有多个规则应用于相同流量和相同源，则使用最宽松的规则。例如，如果有一条规则允许来自 IP 地址 192.0.2.12/32 的 SSH，而另一条规则允许访问来自地址范围 192.0.2.0/24 的所有 TCP 流量，则优先使用允许来自包含 192.0.2.12 的地址范围的所有 TCP 流量的规则。在这种情况下，192.0.2.12 处的客户端的访问数可能会超出您的预期数量。

> **Important**
>
> 在编辑安全组规则以打开端口时要小心。请务必为运行工作负载所需的协议和端口添加仅允许来自可信和经过身份验证的客户的流量的规则。

您可以在每个区域中配置 Amazon EMR 阻止公有访问，以便如果有规则允许在未添加到例外列表的任何端口上进行公有访问时，阻止创建集群。对于 2019 年 7 月之后创建的Amazon账户，默认情况下启用 Amazon EMR 阻止公有访问。对于在 2019 年 7 月之前创建了集群的Amazon账户，默认情况下禁用 Amazon EMR 阻止公有访问。有关更多信息，请参阅使用 Amazon EMR 阻止公有访问。

主题

- 使用 Amazon EMR 管理的安全组
- 使用其它安全组
- 指定 Amazon EMR 托管的安全组和其它安全组
- 为 EMR Notebooks 指定 EC2 安全组
- 使用 Amazon EMR 阻止公有访问

> **Note**
>
> Amazon EMR 旨在使用包容性替代方案来处理可能令人反感或非包容性的行业术语，例如“主”和“从属”。我们已过渡到新的术语，以营造更具包容性的体验并促进您对服务组件的理解。
> 我们现在将“节点”描述为实例，将 Amazon EMR 实例类型描述为主实例、核心实例和任务实例。在过渡期间，您可能仍会发现对过时术语的旧引用，例如与 Amazon EMR 安全组相关的术语。

使用 Amazon EMR 管理的安全组

> **Note**
>
> Amazon EMR 旨在使用包容性替代方案来处理可能令人反感或非包容性的行业术语，例如“主”和“从属”。我们已过渡到新的术语，以营造更具包容性的体验并促进您对服务组件的理解。
> 我们现在将“节点”描述为实例，将 Amazon EMR 实例类型描述为主实例、核心实例和任务实例。在过渡期间，您可能仍会发现对过时术语的旧引用，例如与 Amazon EMR 安全组相关的术语。

不同的托管安全组与集群中的主实例以及核心和任务实例相关联。在私有子网中创建集群时，需要用于服务访问的其它托管安全组。有关网络配置的托管安全组角色的更多信息，请参阅 Amazon VPC 选项。

当您为集群指定托管安全组时，必须对所有托管安全组使用相同类型的安全组（默认或自定义）。例如，您无法为主实例指定自定义安全组，也无法为核心和任务实例指定自定义安全组。

如果您使用默认托管安全组，则无需在创建集群时指定它们。Amazon EMR 会自动使用默认托管安全组。此外，如果集群的 VPC 中尚不存在默认托管安全组，则 Amazon EMR 将会创建。如果您显式地指定默认托管安全组但它们还不存在，Amazon EMR 也会进行创建。

您可以在创建集群时，在托管安全组中编辑规则。在创建新集群时，Amazon EMR 会检查托管安全组中指定的规则，然后创建新集群需要的任何缺失的入站规则以及之前可能已添加的规则。除非另有明确说明，否则，默认 Amazon EMR 托管安全组的每个规则也会添加到您指定的自定义 Amazon EMR 托管安全组。

默认托管安全组如下所示：

- ElasticMapReduce-primary

  有关此安全组中的规则，请参阅适用于主实例（公有子网）的 Amazon EMR 托管的安全组。

- ElasticMapReduce-core

  有关此安全组中的规则，请参阅适用于核心和任务实例（公有子网）的 Amazon EMR 托管的安全组。

- ElasticMapReduce-Primary-Private

  有关此安全组中的规则，请参阅适用于主实例（私有子网）的 Amazon EMR 托管的安全组。

- ElasticMapReduce-Core-Private

  有关此安全组中的规则，请参阅适用于核心和任务实例（私有子网）的 Amazon EMR 托管的安全组。

- ElasticMapReduce-ServiceAccess

  有关此安全组中的规则，请参阅适用于服务访问（私有子网）的 Amazon EMR 托管的安全组。

适用于主实例（公有子网）的 Amazon EMR 托管的安全组

公有子网中主实例的默认托管安全组的组名称为 ElasticMapReduce-primary。它具有以下规则：如果您指定自定义托管安全组，Amazon EMR 将向您的自定义安全组添加所有相同规则。

类型	协议	端口范围	源	详细信息
入站规则				
All ICMP-IPv4 (所有 ICMP-IPv4)	所有	不适用	主实例的托管安全组的组 ID。换句话说，规则出现在相同安全组内。	这些反身规则允许来自与指定安全组关联的任何实例的入站流量。对多个集群使用默认 `ElasticMapReduce-primary` 允许这些集群的核心和任务节点通过 ICMP 或者任何 TCP 或 UDP 端口相互通信。指定自定义托管安全组以限制跨集群访问。
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		

类型	协议	端口范围	源	详细信息
All ICMP-IPV4 (所有 ICMP-IPV4)	所有	不适用	为核心和任务节点指定的托管安全组的组 ID。	这些规则允许所有入站 ICMP 流量以及来自与指定安全组关联的任何核心和任务实例的通过任何 TCP 或 UDP 端口的流量，即使这些实例位于不同的集群中。
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		
自定义	TCP	8443	各种 Amazon IP 地址范围	这些规则允许集群管理器与主节点通信。

使用旧控制台向可信源 SSH 授予主要安全组访问权限

要编辑安全组，您必须具有权限来管理集群所在 VPC 的安全组。有关更多信息，请参阅《IAM 用户指南》中的更改用户的权限和允许管理 EC2 安全组的示例策略。

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Clusters (集群)。选择您要修改的集群的 Name（名称）。
3. 在 Security and access (安全与访问) 下，选择 Security groups for Master (主节点的安全组) 链接。
4. 从列表中选择 ElasticMapReduce-master。
5. 选择 Inbound rules（入站规则）选项卡，然后选择 Edit inbound rules（编辑入站规则）。
6. 通过以下设置检查允许公有访问的入站规则。如果存在，请选择 Delete (删除) 以将其删除。

   - Type

     SSH
   - 端口

     22
   - 源

自定义 0.0.0.0/0

> **Warning**
>
> 2020 年 12 月之前，ElasticMapRemoe-Master 安全组有一个预先配置的规则，允许来自所有来源的端口 22 入站流量。创建此规则可简化与主节点的初始 SSH 客户端连接。我们强烈建议您移除此入站规则，限制流量进入可信来源。

7. 滚动到规则列表的底部并选择 Add Rule (添加规则)。
8. 对于Type (类型)，选择 SSH。

   选择 SSH，自动为 Protocol (协议) 输入 TCP，为 Port Range (端口范围) 输入 22。
9. 针对源，选择 My IP（我的 IP），自动添加您的 IP 地址作为源地址。您也可以添加一系列 Custom (自定义) 可信客户端 IP 地址，或创建针对其他客户端的其他规则。许多网络环境动态分配 IP 地址，因此您以后可能需要更新受信任客户端的 IP 地址。
10. 选择 Save（保存）。
11. 或者，从列表中选择 ElasticMapReduce-slave 并重复上述步骤，以允许对核心和任务节点执行 SSH 客户端访问。

适用于核心和任务实例（公有子网）的 Amazon EMR 托管的安全组

公有子网中核心和任务实例的默认托管安全组的组名称为ElasticMapReduce-core。默认托管安全组具有以下规则，如果您指定自定义托管安全组，则 Amazon EMR 将添加相同规则。

类型	协议	端口范围	源	详细信息
入站规则				
All ICMP-IPV4 (所有 ICMP-IPV4)	所有	不适用	核心和任务实例托管安全组的组 ID。换句话说，规则出现在相同安全组内。	这些反身规则允许来自与指定安全组关联的任何实例的入站流量。对多个集群使用默认 `ElasticMapReduce-core` 允许这些集群的核心和任务实例通过 ICMP 或者任何 TCP 或 UDP 端口相互通信。指定自定义托管安全组以限制跨集群访问。
所有 TCP	TCP	全部		

类型	协议	端口范围	源	详细信息
所有 UDP	UDP	全部		
All ICMP-IPV4 (所有 ICMP-IPV4)	所有	不适用	主实例的托管安全组的组 ID。	这些规则允许所有入站 ICMP 流量以及来自与指定安全组关联的任何主实例的通过任何 TCP 或 UDP 端口的流量，即使这些实例位于不同的集群中。
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		

适用于主实例（私有子网）的 Amazon EMR 托管的安全组

私有子网中主实例的默认托管安全组的组名称 为 ElasticMapReduce-Primary-Private。默认托管安全组具有以下规则，如果您指定自定义托管安全组，则 Amazon EMR 将添加相同规则。

类型	协议	端口范围	源	详细信息
入站规则				
All ICMP-IPv4 (所有 ICMP-IPv4)	所有	不适用	主实例的托管安全组的组 ID。换句话说，规则出现在相同安全组内。	这些反身规则允许来自与指定安全组关联且可从私有子网中进行访问的任何实例的入站流量。对多个集群使用默认 `ElasticMapReduce-Primary-Private` 允许这些集群的核心和任务节点通过 ICMP 或者任何 TCP 或 UDP 端口相互通信。指定自定义托管安全组以限制跨集群访问。
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		
All ICMP-IPV4 (所有 ICMP-IPV4)	所有	不适用	核心和任务节点的托管安全组的组 ID。	这些规则允许所有入站 ICMP 流量以及来自与指定安全组关联且从私有子网内进行访问的任何核心和任务实例的通过任何 TCP 或 UDP 端口的流量，即使这些实例位于不同的集群中。

类型	协议	端口范围	源	详细信息
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		
HTTPS (8443)	TCP	8443	私有子网中用于服务访问的托管安全组的组 ID。	此规则允许集群管理器与主节点通信。
出站规则				
所有流量	All	All	0.0.0.0/0	提供对 Internet 的出站访问。
自定义 TCP	TCP	9443	私有子网中用于服务访问的托管安全组的组 ID。	如果删除了上述“所有流量”的默认出站规则，则该规则是 Amazon EMR 5.30.0 及更高版本的最低要求。 **Note** 当您使用自定义托管安全组时，Amazon EMR 不会添加此规则。
自定义 TCP	TCP	80（ht 或 443（h ）	私有子网中用于服务访问的托管安全组的组 ID。	如果删除了上述“所有流量”的默认出站规则，则该规则是 Amazon EMR 5.30.0 及更高版本通过 https 连接到 Amazon S3 的最低要求。 **Note** 当您使用自定义托管安全组时，Amazon EMR 不会添加此规则。

适用于核心和任务实例（私有子网）的 Amazon EMR 托管的安全组

私有子网中核心和任务实例的默认托管安全组的组名称为 ElasticMapReduce-Core-Private。默认托管安全组具有以下规则，如果您指定自定义托管安全组，则 Amazon EMR 将添加相同规则。

类型	协议	端口范围	源	详细信息
入站规则				
All ICMP-IPV4 (所有 ICMP-IPV4)	所有	不适用	核心和任务实例托管安全组的组 ID。换句话说，规则出现在相同安全组内。	这些反身规则允许来自与指定安全组关联的任何实例的入站流量。对多个集群使用默认 `ElasticMapReduce-core` 允许这些集群的核心和任务实例通过 ICMP 或者任何 TCP 或 UDP 端口相互通信。指定自定义托管安全组以限制跨集群访问。
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		
All ICMP-IPV4 (所有 ICMP-IPV4)	所有	不适用	主实例的托管安全组的组 ID。	这些规则允许所有入站 ICMP 流量以及来自与指定安全组关联的任何主实例的通过任何 TCP 或 UDP 端口的流量，即使这些实例位于不同的集群中。
所有 TCP	TCP	全部		
所有 UDP	UDP	全部		
HTTPS (8443)	TCP	8443	私有子网中用于服务访问的托管安全组的组 ID。	此规则允许集群管理器与核心和任务节点通信。
出站规则				
所有流量	All	All	0.0.0.0/0	请参阅下面的编辑出站规则。
自定义 TCP	TCP	80（ht 或 443（h ）	私有子网中用于服务访问的托管安全组的组 ID。	如果删除了上述“所有流量”的默认出站规则，则该规则是 Amazon EMR 5.30.0 及更高版本通过 https 连接到 Amazon S3 的最低要求。 **Note** 当您使用自定义托管安全组时，Amazon EMR 不会添加此规则。

编辑出站规则

默认情况下，Amazon EMR 使用出站规则创建这个安全组，这些规则允许所有协议和端口上的所有出站流量。之所以选择允许所有出站流量，是因为可以在 Amazon EMR 集群上运行的各种 Amazon EMR 和客户应用程序可能需要不同的出站规则。创建默认安全组时，Amazon EMR 无法预测这些特定设置。您可以缩小安全组中的出站范围，以便仅包括那些适合您的使用案例和安全策略的规则。这个安全组至少需要以下出站规则，但是一些应用程序可能需要额外的出口。

类型	协议	端口范围	目标位置	详细信息
所有 TCP	TCP	全部	pl-*xxxxxxxx*	托管式 Amazon S3 前缀列表 com.amazonaws. *MyRegion*.s3。
所有流量	All	全部	sg-*xxxxxxxxxxxxxxxxx*	ElasticMapReduce-Core-Private 安全组的 ID
所有流量	All	全部	sg-*xxxxxxxxxxxxxxxxx*	ElasticMapReduce-Primary-Private 安全组的 ID
自定义 TCP	TCP	9443	sg-*xxxxxxxxxxxxxxxxx*	ElasticMapReduce-ServiceAccess 安全组的 ID

适用于服务访问（私有子网）的 Amazon EMR 托管的安全组

私有子网中服务访问的默认托管安全组的 Group Name (组名称) 为 ElasticMapReduce-ServiceAccess。它具有入站规则和允许流量通过 HTTPS（端口 8443、端口 9443）到私有子网中其它托管安全组的出站规则。这些规则允许集群管理器与主节点以及核心节点和任务节点进行通信。如果您使用的是自定义安全组，则需要相同的规则。

类型	协议	端口范围	源	详细信息
入站规则对于使用 Amazon EMR 版本 5.30.0 及更高版本的 Amazon EMR 集群是必需条件。				
自定义 TCP	TCP	9443	主实例的托管安全组的组 ID。	此规则允许在主实例的安全组与服务访问安全组之间进行通信。

类型	协议	端口范围	源	详细信息
所有 Amazon EMR 集群都需要出站规则				
自定义 TCP	TCP	8443	主实例的托管安全组的组 ID。	这些规则允许集群管理器与主节点以及核心节点和任务节点进行通信。
自定义 TCP	TCP	8443	核心和任务实例托管安全组的组 ID。	这些规则允许集群管理器与主节点以及核心节点和任务节点进行通信。

使用其它安全组

无论是使用默认托管安全组还是指定自定义托管安全组，您都可以使用其它安全组。其它安全组可让您灵活地定制不同的集群和从外部客户端、资源和应用程序之间的访问。

例如，请考虑以下情形。您需要多个集群相互通信，但希望仅允许某特定集群子集对主实例进行入站 SSH 访问。要执行此操作，您可以对集群使用相同的一组托管安全组。然后，您可以创建允许从可信客户端进行入站 SSH 访问的其它安全组，并且为子网中每个集群的主实例指定其它安全组。

您可以对主实例最多应用四个其它安全组，对于核心和任务实例最多应用四个，以及对于服务访问（私有子网中）最多应用四个。如有必要，您可以为主实例、核心和任务实例和服务访问指定相同的其它安全组。账户中安全组和规则的最大数量受账户限制的约束。有关更多信息，请参阅《Amazon VPC 用户指南》中的安全组限制。

指定 Amazon EMR 托管的安全组和其它安全组

您可以使用 Amazon Web Services Management Console、Amazon CLI 或 Amazon EMR API 指定安全组。如果您未指定安全组，Amazon EMR 会创建默认安全组。指定其它安全组是可选的。您可以为主实例、核心和任务实例和服务访问（仅私有子网）分配其它安全组。

New console

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

使用新控制台指定安全组

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Networking（联网）下，选择 EC2 security groups (firewall) [EC2 安全组（防火墙）] 旁边的箭头以展开此部分。在 Primary node（主节点）和 Core and task nodes（核心节点和任务节点）下，默认选择 Amazon EMR 默认托管安全组。如果您使用私有子网，则还可以选择安全组进行 Service access（服务访问）。
4. 要更改您的 Amazon EMR 托管安全组，请使用 Choose security groups（选择安全组）下拉菜单以从 Amazon EMR-managed security group（Amazon EMR 托管安全组）选项列表中选择其他选项。您有一个 Amazon EMR 托管安全组，可用于 Primary node（主节点）、Core and task nodes（核心节点和任务节点）。
5. 要添加自定义安全组，请使用相同的 Choose security groups（选择安全组）下拉菜单从 Custom security group（自定义安全组）选项列表中选择最多四个自定义安全组。对于 Primary node（主节点）和 Core and task nodes（核心节点和任务节点），最多可以有四个自定义安全组。
6. 选择适用于集群的任何其他选项。
7. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台指定安全组

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 依次选择 Create cluster (创建集群)、Go to advanced options (转到高级选项)。
3. 选择集群的选项，直到到达 Step 4: Security (步骤 4: 安全组)。
4. 选择 EC2 Security Groups (EC2 安全组) 以展开该部分。

   在 EMR managed security groups (EMR 托管的安全组) 中，默认情况下选择默认托管安全组。如果 Master（主实例）、Core & Task（核心和任务实例）或 Service Access（服务访问）（仅限私有子网）的 VPC 中不存在默认值，则 Create（创建）会显示在关联的安全组名称之前。

5. 如果使用自定义托管安全组，请从 EMR managed security groups (EMR 托管的安全组) 列表中进行选择。

   如果您选择自定义托管安全组，将显示一条消息，通知您为其它实例选择自定义安全组。您只能对集群使用自定义或默认托管安全组。
6. （可选）在 Additional security groups (其它安全组) 下，选择铅笔图标，从列表中最多选择四个安全组，然后选择 Assign security groups (分配安全组)。根据需要对每个 Master (主实例)、Core & Task (核心和任务实例) 和 Service Access (服务访问) 重复操作。
7. 选择创建集群。

使用 Amazon CLI 指定安全组

要使用 Amazon CLI 指定安全组，请使用 `create-cluster` 命令与 `--ec2-attributes` 选项的以下参数：

参数	描述
`EmrManagedPrimarySecurityGroup`	使用此参数为主实例指定自定义托管安全组。如果指定此参数，还必须指定 `EmrManagedCoreSecurityGroup` 。对于私有子网中的集群，还必须指定 `ServiceAccessSecurityGroup` 。
`EmrManagedCoreSecurityGroup`	使用此参数为核心和任务实例指定自定义托管安全组。如果指定此参数，还必须指定 `EmrManagedPrimarySecurityGroup` 。对于私有子网中的集群，还必须指定 `ServiceAccessSecurityGroup` 。
`ServiceAccessSecurityGroup`	使用此参数为服务访问指定自定义托管安全组，这仅适用于私有子网中的集群。您指定为 `ServiceAccessSecurityGroup` 的安全组不应用于任何其它目的，还应为 Amazon EMR 保留。如果指定此参数，还必须指定 `EmrManagedPrimarySecurityGroup` 。

参数	描述
AdditionalPrimarySecurityGroups	使用此参数最多为主实例指定四个其它安全组。
AdditionalCoreSecurityGroups	使用此参数最多为核心和任务实例指定四个其它安全组。

Example – 指定自定义 Amazon EMR 托管的安全组和其它安全组

以下示例为私有子网中的集群、主实例的多个其它安全组以及核心和任务实例的单个其它安全组指定自定义 Amazon EMR 托管式安全组。

> **Note**
>
> 为了便于读取，包含 Linux 行继续符 (\)。它们可以通过 Linux 命令删除或使用。对于 Windows，请将它们删除或替换为脱字号 (^) 。

```
aws emr create-cluster --name "ClusterCustomManagedAndAdditionalSGs" \
--release-label emr-emr-5.36.1 --applications Name=Hue Name=Hive \
Name=Pig --use-default-roles --ec2-attributes \
SubnetIds=subnet-xxxxxxxxxxxx,KeyName=myKey,\
ServiceAccessSecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedPrimarySecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedCoreSecurityGroup=sg-xxxxxxxxxxx,\
AdditionalPrimarySecurityGroups=['sg-xxxxxxxxxxx',\
'sg-xxxxxxxxxxx','sg-xxxxxxxxxx'],\
AdditionalCoreSecurityGroups=sg-xxxxxxxxxxx \
--instance-type m5.xlarge
```

有关更多信息，请参阅《Amazon CLI 命令参考》中的 create-cluster。

为 EMR Notebooks 指定 EC2 安全组

创建 EMR 笔记本时，使用笔记本编辑器时，将使用两个安全组来控制 EMR 笔记本和 Amazon EMR 集群之间的网络流量。默认安全组具有最小规则，其中仅允许 EMR Notebooks 服务和附加了笔记本的集群之间的网络流量。

EMR 笔记本借助 Apache Livy 通过使用 TCP 端口 18888 的代理与集群通信。利用为您的环境定制的规则创建自定义安全组时，您可以限制网络流量，以便仅使一部分笔记本在特定集群上的笔记本编辑器中运行代码。除了集群的默认安全组外，集群还使用您的自定义安全性。有关更多信息，请参阅《Amazon EMR 管理指南》中的使用安全组控制网络流量和为 EMR Notebooks 指定 EC2 安全组。

主实例的默认 EC2 安全组

除主实例的集群安全组外，主实例的默认 EC2 安全组还会与主实例关联。

组名：ElasticMapReduceEditors-Livy

规则

- 入站

  允许 TCP 端口 18888 响应来自 EMR Notebooks 默认 EC2 安全组中任何资源的请求
- 出站

  无

EMR Notebooks 的默认 EC2 安全组

EMR Notebooks 的默认 EC2 安全组会与向任意 EMR Notebooks 分配的笔记本编辑器相关联。

组名：ElasticMapReduceEditors-Editor

规则

- 入站

  无
- 出站

  允许 TCP 端口 18888 向 EMR Notebooks 默认 EC2 安全组中的任何资源发出请求。

将笔记本与 Git 存储库关联时，针对 EMR Notebooks 的自定义 EC2 安全组

要将 Git 存储库链接到笔记本，EMR 笔记本的安全组必须包含出站规则，以允许笔记本将流量路由到 Internet。为了这个目的，建议您创建一个新的安全组。更新默认 ElasticMapReduceEditors-Editor 安全组可能会向附加到此安全组的其它笔记本提供相同的出站规则。

规则

- 入站

  无

- 出站

  允许笔记本通过集群将流量路由到互联网，如以下示例所示：值 0.0.0.0/0 用于示例。您可以修改此规则以指定基于 Git 的存储库的 IP 地址。

类型	协议	端口范围	目标位置
自定义 TCP 规则	TCP	18888	SG-
HTTPS	TCP	443	0.0.0.0/0

使用 Amazon EMR 阻止公有访问

如果集群的安全配置允许来自公有 IP 地址的入站流量通过某个端口，Amazon EMR 阻止公有访问（BPA）会阻止您在公有子网中启动该集群。

> **Important**
>
> 默认情况下，阻止公有访问处于启用状态。为了增强账户保护，我们建议您将其保持启用状态。

了解阻止公有访问

您可以使用阻止公有访问账户级别配置，来 Amazon EMR 集群的公有网络访问进行集中管理。

当来自您的 Amazon Web Services 账户 的用户启动集群时，Amazon EMR 会检查该集群的安全组中的端口规则，并将其与您的入站流量规则进行比较。如果安全组具有向公有 IP 地址 IPv4 0.0.0.0/0 或 IPv6 ::/0 开放端口的入站规则，且这些端口并未指定为您账户的例外，则 Amazon EMR 不允许用户创建集群。

如果用户修改了公有子网中正在运行的集群的安全组规则，使其具有违反您账户的 BPA 配置的公有访问规则，则 Amazon EMR 将撤销新规则，前提是它有权这样做。如果 Amazon EMR 无权撤销该规则，它会在 Amazon Health 控制面板中创建一个描述违规行为的事件。要向 Amazon EMR 授予撤销规则权限，请参阅 配置 Amazon EMR 以撤销安全组规则。

默认情况下，您的 Amazon Web Services 账户 中每个 Amazon Web Services 区域 的所有集群都启用了阻止公有访问权限。BPA 适用于集群的整个生命周期，但不适用于您在私有子网中创建的集群。您可以配置 BPA 规则的例外；默认情况下，端口 22 是例外。有关设置例外的更多信息，请参阅 配置阻止公有访问。

配置阻止公有访问

您可以随时更新账户中的安全组和阻止公有访问配置。

您可以使用 Amazon Web Services Management Console、 Amazon Command Line Interface（Amazon CLI）和 Amazon EMR API 开启和关闭阻止共有访问（BPA）设置。设置按区域适用于您的账户。为了保障集群安全性，建议您使用 BPA。

New console

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

使用新控制台配置阻止公有访问

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr/。
2. 在顶部导航栏中，如果还未选中，则选中要配置的区域。
3. 在左侧导航窗格中的 EMR on EC2 下，选择 Block public access（阻止公有访问）。
4. 在 Block public access settings (阻止公有访问设置) 下，完成以下步骤。

要……	请执行此操作...
打开或关闭阻止公有访问	

要……	请执行此操作...
	选择 Edit（编辑），根据需要选择 Turn on（打开）或 Turn off（关闭），然后选择 Save（保存）。
编辑例外列表中的端口	1. 选择 Edit（编辑），然后找到 Port range exceptions（端口范围例外）部分。 2. 要将端口添加到例外列表，请选择 Add a port range (添加端口范围) 并输入新端口或端口范围。对要添加的每个端口或端口范围重复此过程。 3. 要删除端口或端口范围，请选择端口范围列表中条目旁边的 Remove（删除）。 4. 选择 Save（保存）。

Old console

使用旧控制台查看配置阻止公有访问

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/emr。
2. 在顶部导航栏中，确认要配置的区域已选中。
3. 选择 Block public access (阻止公有访问)。
4. 在 Block public access settings (阻止公有访问设置) 下，完成以下步骤。

要……	请执行此操作...
打开或关闭阻止公有访问	选择 Change（更改），根据需要选择 On（开）或 Off（关），然后选择对勾标记进行确认。

要……	请执行此操作...
编辑例外列表中的端口	1. 在 Exceptions (例外) 下，选择 Edit (编辑)。 2. 要将端口添加到例外列表，请选择 Add a port range (添加端口范围) 并输入新端口或端口范围。对要添加的每个端口或端口范围重复此过程。 3. 要删除端口或端口范围，请选择 Port ranges (端口范围) 列表中条目旁边的 x。 4. 选择 Save Changes。

Amazon CLI

使用 Amazon CLI 配置阻止公有访问

- 使用 `aws emr put-block-public-access-configuration` 命令可配置阻止公有访问，如以下示例所示。

要……	请执行此操作...
打开阻止公有访问	将 `BlockPublicSecurityGroupRules` 设置为 `true`，如以下示例所示。要启动集群，与集群关联的安全组不得具有允许公有访问的入站规则。 `aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=true`

要.....	请执行此操作...
关闭阻止公有访问	将 BlockPublicSecurityGroupRules 设置为 false，如以下示例所示。与集群关联的安全组可以具有允许在任何端口上进行公有访问的入站规则。不推荐您使用此配置。 `aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=false`
打开阻止公有访问并将端口指定为例外	以下示例将打开阻止公有访问，并将端口 22 和端口 100-101 指定为例外。这样，如果关联的安全组具有允许在端口 22、端口 100 或端口 101 上进行公有访问的入站规则，则允许创建集群。 `aws emr put-block-public-access-configuration --block-public-access-configuration '{ "BlockPublicSecurityGroupRules": true, "PermittedPublicSecurityGroupRuleRanges": [ { "MinRange": 22, "MaxRange": 22 }, { "MinRange": 100, "MaxRange": 101 } ] }'`

配置 Amazon EMR 以撤销安全组规则

Amazon EMR 需要权限才能撤销安全组规则并遵守您的阻止公有访问配置。您可以使用以下方法之一向 Amazon EMR 授予其所需的权限：

- （建议）将 AmazonEMRServicePolicy_v2 托管的策略附加到服务角色。有关更多信息，请参阅Amazon EMR 的服务角色（EMR 角色）。

- 创建允许对安全组执行 `ec2:RevokeSecurityGroupIngress` 操作的新内联策略。有关如何修改角色权限策略的更多信息，请参阅《IAM 用户指南》中的使用 IAM 控制台、Amazon API 和 Amazon CLI 修改角色权限策略。

解决阻止公有访问违规问题

如果发生阻止公有访问违规，则可以通过以下操作之一来缓解违规：

- 如果要访问集群上的 Web 界面，请使用 查看 Amazon EMR 集群上托管的 Web 界面 中所述的选项之一通过 SSH（端口 22）访问该界面。
- 要允许来自特定 IP 地址而不是公有 IP 地址的流量进入集群，请添加安全组规则。有关更多信息，请参阅《Amazon EC2 入门指南》中的向安全组添加规则。
- （不推荐）您可以将 Amazon EMR BPA 例外配置为包括所需的端口或端口范围。指定 BPA 例外时，未受保护的端口会带来风险。如果您计划指定例外，则应在不再需要该例外时立即将其删除。有关更多信息，请参阅配置阻止公有访问。

识别与安全组规则关联的集群

您可能需要识别与给定安全组规则关联的所有集群，或者找到给定集群的安全组规则。

- 如果您知道安全组，则可以在找到安全组的网络接口时识别其关联的集群。有关更多信息，请参阅 Amazon Web Services re:Post 上的 如何查找与 Amazon EC2 安全组关联的资源？。连接到这些网络接口的 Amazon EC2 实例，将使用它们所属集群的 ID 进行标记。
- 如果要查找已知集群的安全组，请按照 查看集群状态和详细信息 中的步骤操作。您可以在控制台的网络和安全面板中找到集群的安全组，也可以从 Amazon CLI 的 `Ec2InstanceAttributes` 字段中找到集群的安全组。

Amazon EMR 的合规性验证

作为多个Amazon合规性计划的一部分，第三方审计员将评估 Amazon EMR 的安全性和合规性。其中包括 SOC、PCI、FedRAMP、HIPAA 及其它。

有关特定合规性计划范围内的 Amazon 服务的列表，请参阅合规性计划范围内的 Amazon 服务。有关一般信息，请参阅 Amazon 合规性计划。

您可以使用 Amazon Artifact 下载第三方审计报告。有关更多信息，请参阅在 Amazon Artifact 中下载报告。

您在使用 Amazon EMR 时的合规性责任由您的数据的敏感性、您公司的合规性目标以及适用的法律法规决定。如果您对 Amazon EMR 的使用需遵守 HIPAA、PCI 或 FedRAMP 等标准，Amazon将提供以下有用资源：

- 安全性和合规性 Quick Start 指南 – 这些部署指南讨论了架构注意事项，并提供了在Amazon上部署基于安全性和合规性的基准环境的步骤。
- 《设计符合 HIPAA 安全性和合规性要求的架构》白皮书 – 此白皮书介绍公司如何使用Amazon创建符合 HIPAA 标准的应用程序。
- Amazon合规性资源 – 此业务手册和指南集合可能适用于您的行业和位置。
- Amazon Config – 此Amazon服务评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- Amazon Security Hub – 此 Amazon 服务提供了 Amazon 中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实践。

Amazon EMR 中的恢复功能

Amazon全球基础设施围绕Amazon区域和可用区构建。Amazon区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 Amazon 区域和可用区的更多信息，请参阅 Amazon 全球基础设施。

除了Amazon全球基础设施之外，Amazon EMR 还提供多种功能，以帮助支持您的数据恢复能力和备份需求。

- 通过 EMRFS 与 Simple Storage Service (Amazon S3) 集成
- 支持多个主节点

Amazon EMR 中的基础设施安全性

作为一项托管服务，Amazon EMR 由 Amazon Web Services：安全流程概览白皮书中所述的Amazon全球网络安全流程提供保护。

您可以使用Amazon发布的 API 调用通过网络访问 Amazon EMR。客户端必须支持传输层安全性（TLS）1.2。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 Amazon Security Token Service (Amazon STS) 生成临时安全凭证来对请求进行签名。

主题

- 使用接口 VPC 终端节点连接到 Amazon EMR

使用接口 VPC 终端节点连接到 Amazon EMR

您可以使用 Virtual Private Cloud（VPC）中的接口 VPC 终端节点（Amazon PrivateLink）直接连接到 Amazon EMR，而不是通过互联网连接。当您使用接口 VPC 终端节点时，您的 VPC 和 Amazon EMR 之间的通信完全在 Amazon 网络内进行。每个 VPC 终端节点都由您的 VPC 子网中一个或多个使用私有 IP 地址的弹性网络接口（ENI）代表。

接口 VPC 终端节点将您的 VPC 直接连接到 Amazon EMR，而无需互联网网关、NAT 设备、VPN 连接或 Amazon Direct Connect 连接。VPC 中的实例不需要公有 IP 地址便可与 Amazon EMR API 进行通信。

要通过您的 VPC 使用 Amazon EMR，您必须从位于 VPC 中的实例进行连接，或者使用 Amazon Virtual Private Network（VPN）或 Amazon Direct Connect 将您的专用网络连接到 VPC。有关 Amazon VPN 的信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的 VPN 连接。有关 Amazon Direct Connect 连接的更多信息,请参阅在《Amazon Direct Connect 用户指南》中的创建连接。

您可以创建接口 VPC 终端节点以使用Amazon控制台或 Amazon Command Line Interface（Amazon CLI）命令连接到 Amazon EMR。有关更多信息，请参阅创建接口终端节点。

在创建接口 VPC 终端节点后，如果您为终端节点启用私有 DNS 主机名，则默认 Amazon EMR 终端节点将解析为您的 VPC 终端节点。Amazon EMR 的默认服务名称终端节点采用以下格式。

```
elasticmapreduce.Region.amazonaws.com
```

如果您不启用私有 DNS 主机名，Amazon VPC 将提供一个您可以使用的 DNS 终端节点名称，格式如下。

```
VPC_Endpoint_ID.elasticmapreduce.Region.vpce.amazonaws.com
```

有关更多信息，请参阅《Amazon VPC 用户指南》 中的接口 VPC 终端节点（Amazon PrivateLink）。

Amazon EMR 支持调用您的 VPC 中的所有 API 操作。

您可以将 VPC 终端节点策略附加到 VPC 终端节点，以控制 IAM 委托人的访问权限。您还可以将安全组与 VPC 终端节点关联，以便根据网络流量的源和目标（例如 IP 地址范围）控制入站和出站访问。有关更多信息，请参阅使用 VPC 终端节点控制对服务的访问。

为 Amazon EMR 创建 VPC 终端节点策略

您可以为 Amazon EMR 的 Amazon VPC 终端节点创建一个策略，在该策略中指定以下内容：

- 可以或不能执行操作的委托人
- 可执行的操作
- 可对其执行操作的资源

有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 终端节点控制对服务的访问权限。

Example – 用于拒绝来自指定Amazon账户的所有访问的 VPC 终端节点策略

以下 VPC 终端节点策略拒绝 Amazon 账户 *123456789012* 所有使用终端节点访问资源的权限。

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example – 仅允许 VPC 访问指定的 IAM 委托人（用户）的 VPC 终端节点策略

以下 VPC 端点策略仅允许对 Amazon 账户 *123456789012* 中的用户 *lijuan* 进行完全访问。使用终端节点拒绝所有其它 IAM 委托人进行访问。

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/lijuan"
                ]
            }
        }]
}
```

Example – 允许只读 EMR 操作的 VPC 终端节点策略

以下 VPC 终端节点策略仅允许Amazon账户 *123456789012* 执行指定的 Amazon EMR 操作。

指定的操作为 Amazon EMR 提供等效的只读访问权限。针对指定账户拒绝 VPC 上的所有其它操作。所有其它帐户都被拒绝进行任何访问。有关 Amazon EMR 操作的列表，请参阅 Amazon EMR 的操作、资源和条件键。

```
{
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:DescribeEditor",
```

```
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example – 拒绝访问指定集群的 VPC 终端节点策略

以下 VPC 终端节点策略允许对所有账户和委托人进行完全访问，但拒绝Amazon账户 *123456789012* 访问在集群 ID 为 *j-A1B2CD34EF5G* 的 Amazon EMR 集群上执行的操作。仍然允许其它不支持集群资源级权限的 Amazon EMR 操作。有关 Amazon EMR 操作及其相应资源类型的列表，请参阅 Amazon EMR 的操作、资源和条件键。

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-west-2:123456789012:cluster/j-
A1B2CD34EF5G",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

管理集群

启动集群后，您可以对其进行监控和管理。Amazon EMR 提供多种可用于连接和控制集群的工具。

主题

- 连接到集群
- 向集群提交工作
- 查看和监控集群
- 使用集群扩展
- 终止集群
- 使用控制台克隆集群
- 通过 Amazon Data Pipeline 自动处理定期集群

连接到集群

运行 Amazon EMR 集群时，通常只需运行一个应用程序来分析您的数据，然后从 Amazon S3 存储桶收集输出即可。而有的时候，您可能希望在集群运行期间与主节点进行交互。例如，您可能要连接到主节点，从而运行交互式查询、检查日志文件、调试与集群有关的问题、使用在主节点上运行的应用程序（如 Ganglia）监控性能等。以下部分介绍了可用于连接到主节点的技术。

在 EMR 集群中，主节点是一种 Amazon EC2 实例，能够协调正以任务节点和核心节点的形式运行的 EC2 实例。主节点可用于与其公有 DNS 名称进行连接。默认情况下，Amazon EMR 会为主节点、核心节点和任务节点创建决定节点访问方式的安全组规则。

> **Note**
>
> 您仅可以在集群运行期间连接到主节点。集群终止时，充当主节点的 EC2 实例会终止，将不再可用。要连接到主节点，您还必须对集群进行身份验证。您可以使用 Kerberos 进行身份验证，或者在启动集群时指定 Amazon EC2 密钥对私有密钥。有关配置 Kerberos 然后进行连接的更多信息，请参阅使用 Kerberos 通过 Amazon EMR 进行身份验证。在您从控制台启动集群时，需在 Create Cluster (创建集群) 页面上的 Security and Access (安全与访问) 部分中指定 Amazon EC2 密钥对私有密钥。

默认情况下，ElasticMapReduce-master 安全组不允许入站 SSH 访问。您可能需要添加一个入站规则，以允许从您想访问的源进行 SSH 访问（TCP 端口 22）。有关修改安全组规则的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的向安全组添加规则。

> **Important**
>
> 请勿修改 ElasticMapReduce-master 安全组中的其余规则。修改这些规则可能会干扰集群的操作。

主题

- 连接之前：授权入站流量
- 使用 SSH 连接到主节点

连接之前：授权入站流量

您必须先授权来自信任客户端（例如您电脑的 IP 地址）的入站 SSH 流量（端口 22），然后才能连接到 Amazon EMR 集群。为此，请为要连接的节点编辑托管式安全组规则。例如，以下说明将向您展示如何为默认 ElasticMapReduce-master 安全组的 SSH 访问添加入站规则。

有关结合使用 Amazon EMR与安全组的更多信息，请参阅使用安全组控制网络流量。

New console

使用新控制台授予可信源 SSH 访问主要安全组的权限

要编辑安全组，您必须具有权限来管理集群所在 VPC 的安全组。有关更多信息，请参阅《IAM 用户指南》中的更改用户的权限和允许管理 EC2 安全组的示例策略。

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。集群详细信息页面随即打开。将预先选择此页面上的 Properties（属性）选项卡。
3. 在 Properties（属性）选项卡的 Networking（联网）下，选择 EC2 security groups (firewall) [EC2 安全组（防火墙）] 旁边的箭头以展开此部分。在 Primary node（主节点）下，选择安全组链接。此操作将打开 EC2 控制台。
4. 选择 Inbound rules（入站规则）选项卡，然后选择 Edit inbound rules（编辑入站规则）。
5. 通过以下设置检查允许公有访问的入站规则。如果存在，请选择 Delete (删除) 以将其删除。

- Type

  SSH
- 端口

  22
- 源

  自定义 0.0.0.0/0

> **Warning**
>
> 2020 年 12 月之前，ElasticMapRemoe-Master 安全组有一个预先配置的规则，允许来自所有来源的端口 22 入站流量。创建此规则可简化与主节点的初始 SSH 客户端连接。我们强烈建议您移除此入站规则，限制流量进入可信来源。

6. 滚动到规则列表的底部并选择 Add Rule (添加规则)。
7. 对于Type (类型)，选择 SSH。此选择将自动为 Protocol（协议）输入 TCP，为 Port Range（端口范围）输入 22。
8. 针对源，选择 My IP（我的 IP），自动添加您的 IP 地址作为源地址。您也可以添加一系列 Custom (自定义) 可信客户端 IP 地址，或创建针对其他客户端的其他规则。许多网络环境动态分配 IP 地址，因此您以后可能需要更新受信任客户端的 IP 地址。
9. 选择 Save（保存）。
10. 您可以选择返回步骤 3，选择 Core and task nodes（核心节点和任务节点），然后重复步骤 4 - 8。这会向核心节点和任务节点授予 SSH 客户端访问权限。

Old console

使用旧控制台向可信源 SSH 授予主要安全组访问权限

要编辑安全组，您必须具有权限来管理集群所在 VPC 的安全组。有关更多信息，请参阅《IAM 用户指南》中的更改用户的权限和允许管理 EC2 安全组的示例策略。

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Clusters (集群)。选择您要修改的集群的 Name（名称）。

3. 在 Security and access (安全与访问) 下，选择 Security groups for Master (主节点的安全组) 链接。
4. 从列表中选择 ElasticMapReduce-master。
5. 选择 Inbound rules（入站规则）选项卡，然后选择 Edit inbound rules（编辑入站规则）。
6. 通过以下设置检查允许公有访问的入站规则。如果存在，请选择 Delete (删除) 以将其删除。

   - Type

     SSH
   - 端口

     22
   - 源

     自定义 0.0.0.0/0

   > **Warning**
   >
   > 2020 年 12 月之前，ElasticMapRemoe-Master 安全组有一个预先配置的规则，允许来自所有来源的端口 22 入站流量。创建此规则可简化与主节点的初始 SSH 客户端连接。我们强烈建议您移除此入站规则，限制流量进入可信来源。

7. 滚动到规则列表的底部并选择 Add Rule (添加规则)。
8. 对于Type (类型)，选择 SSH。

   选择 SSH，自动为 Protocol (协议) 输入 TCP，为 Port Range (端口范围) 输入 22。
9. 针对源，选择 My IP（我的 IP），自动添加您的 IP 地址作为源地址。您也可以添加一系列 Custom (自定义) 可信客户端 IP 地址，或创建针对其他客户端的其他规则。许多网络环境动态分配 IP 地址，因此您以后可能需要更新受信任客户端的 IP 地址。
10. 选择 Save（保存）。
11. 或者，从列表中选择 ElasticMapReduce-slave 并重复上述步骤，以允许对核心和任务节点执行 SSH 客户端访问。

使用 SSH 连接到主节点

Secure Shell（SSH）是您可用于创建远程电脑安全连接的网络协议。建立连接后，请如同在远程电脑上运行那样，操作您在本地计算机上的终端。您在本地发出的命令会在远程电脑上运行，同时，远程计算机的命令输出将显示在终端窗口中。

当您将 SSH 与 Amazon 结合使用时，您将连接到 EC2 实例（这是在云端运行的虚拟服务器）。与 Amazon EMR 结合使用时，SSH 最常见的用途是连接到充当集群主节点的 EC2 实例。

使用 SSH 连接到主节点可让您监控集群并与集群交互。您可以在主节点上发出 Linux 命令、以交互方式运行应用程序（如 Hive 和 Pig）、浏览目录、读取日志文件等。您还可以在 SSH 连接中创建隧道，以查看主节点上托管的 Web 界面。有关更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

要使用 SSH 连接到主节点，您需要主节点的公有 DNS 名称。此外，与主节点关联的安全组必须具有入站规则，此规则允许来自源（其中包含发起 SSH 连接的客户端）的 SSH（TCP 端口 22）流量。您可能需要添加规则，以允许客户端通过 SSH 连接。有关修改安全组规则的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用安全组控制网络流量和 向安全组添加规则。

检索主节点的公有 DNS 名称

您可以使用 Amazon EMR 控制台和 Amazon CLI 检索主节点的公有 DNS 名称。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台检索主节点的公有 DNS 名称

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要检索公有 DNS 名称的集群。
3. 记下显示在集群详细信息页面上 Summary（摘要）部分中的 Primary node public DNS（主节点公有 DNS）值。

Old console

使用旧控制台检索主节点的公有 DNS 名称

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 在 Cluster List (集群列表) 页面上，选择您的集群的链接。
3. 记下显示在 Cluster Details (集群详细信息) 页上 Summary (摘要) 部分中 Master public DNS (主节点公有 DNS) 的值。

> **Note**
>
> 您还可以选择 SSH 链接，以获取有关创建 SSH 与主节点连接的说明。

CLI

使用 Amazon CLI 检索主节点的公有 DNS 名称

1. 要检索集群标识符，请键入以下命令。

```
aws emr list-clusters
```

输出会列出您的集群（包括集群 ID）。记下您连接的集群的集群 ID。

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "My cluster"
```

2. 要列出集群实例（包括集群的公有 DNS 名称），请键入以下命令之一。将 *j-2AL4XXXXXX5T9* 替换为上一个命令返回的集群 ID。

```
aws emr list-instances --cluster-id j-2AL4XXXXXX5T9
```

或者：

```
aws emr describe-cluster --cluster-id j-2AL4XXXXXX5T9
```

输出会列出集群实例（包括 DNS 名称和 IP 地址）。记下 PublicDnsName 值。

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040779.263,
        "CreationDateTime": 1408040515.535
    },
    "State": "RUNNING",
    "StateChangeReason": {}
},
"Ec2InstanceId": "i-e89b45e7",
"PublicDnsName": "ec2-###-##-##-###.us-west-2.compute.amazonaws.com"

"PrivateDnsName": "ip-###-##-##-###.us-west-2.compute.internal",
"PublicIpAddress": "##.###.###.##",
"Id": "ci-12XXXXXXXXFMH",
"PrivateIpAddress": "###.##.#.###"
```

有关更多信息，请参阅 Amazon CLI 中的 Amazon EMR 命令。

在 Linux、Unix 和 Mac OS X 上使用 SSH 和 Amazon EC2 私有密钥连接到主节点

要创建使用私有密钥文件进行身份验证的 SSH 连接，您需要在启动集群时指定 Amazon EC2 密钥对私有密钥。有关访问密钥对的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 密钥对。

默认情况下，您的 Linux 电脑很可能包含 SSH 客户端。例如，大多数 Linux、Unix 和 macOS 操作系统上都会安装 OpenSSH。您可以通过在命令行键入 ssh 来检查 SSH 客户端。如果您的计算机无法识别该命令，请安装 SSH 客户端以连接到主节点。OpenSSH 项目会免费提供一套完整的 SSH 工具执行文件。有关更多信息，请访问 OpenSSH 网站。

以下说明演示了如何在 Linux、Unix 和 Mac OS X 上打开与 Amazon EMR 主节点的 SSH 连接。

配置密钥对私有密钥文件权限

您必须先对 .pem 文件设置权限，以便只有密钥所有者才有权访问该文件，然后才能使用 Amazon EC2 密钥对私有密钥创建 SSH 连接。使用终端或 Amazon CLI 创建 SSH 连接时必须这样做。

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 找到您的 .pem 文件。这些说明假定文件名为 mykeypair.pem，并将其存储在当前用户的主目录中。
3. 键入以下命令以设置权限。将 *~/mykeypair.pem* 替换为您的密钥对私有密钥文件的完整路径和文件名。例如 C:/Users/<username>/.ssh/mykeypair.pem。

   ```
   chmod 400 ~/mykeypair.pem
   ```

   如果您没有对 .pem 文件设置权限，您将收到一条错误消息，指出您的密钥文件不受保护，并且密钥将被拒绝。您只需在第一次使用密钥对私有密钥文件时设置权限即可连接。

使用终端连接到主节点

1. 打开终端窗口。在 Mac OS X 上，选择 Applications (应用程序) > Utilities (实用程序) > Terminal (终端)。在其它 Linux 发布版上，终端通常位于 Applications (应用程序) > Accessories (附件) > Terminal (终端)。
2. 要建立与主节点的连接，请键入以下命令。将 *ec2-###-##-##-###.compute-1.amazonaws.com* 替换为您的集群的主节点公有 DNS 名称，并将 *~/mykeypair.pem* 替换为您的 .pem 文件的完整路径和文件名。例如 C:/Users/<username>/.ssh/mykeypair.pem。

   ```
   ssh hadoop@ec2-###-##-##-###.compute-1.amazonaws.com -i ~/mykeypair.pem
   ```

   > **Important**
   >
   > 连接到 Amazon EMR 主节点时，必须使用登录名 hadoop，否则可能会遇到类似于 Server refused our key 的错误。

3. 警告指出无法验证您正在连接的主机真实性。键入 yes 以继续。
4. 完成主节点上的工作以后，键入以下命令，以关闭 SSH 连接。

```
exit
```

如果您在使用 SSH 连接到主节点时遇到困难，请参阅排查实例的连接问题。

在 Windows 上使用 SSH 连接到主节点

Windows 用户可以使用 SSH 客户端（如 PuTTY）连接到主节点。连接到 Amazon EMR 主节点之前，应下载并安装 PuTTY 和 PuTTYgen。可以从 PuTTY 下载页面下载这些工具。

PuTTY 本身不支持 Amazon EC2 生成的密钥对私有密钥文件格式（.pem）。您可使用 PuTTYgen 将密钥文件转换为所需的 PuTTY 格式（.ppk）。您必须将密钥转换为此格式 (.ppk)，然后才能尝试使用 PuTTY 连接到主节点。

有关转换密钥的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用 PuTTYgen 转换私有密钥。

使用 PuTTY 连接到主节点

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 打开 putty.exe。也可以从 Windows 程序列表中启动 PuTTY。
3. 如有必要，在 Category (类别) 列表中，选择 Session (会话)。
4. 对于 Host Name (or IP address) [主机名 (或 IP 地址)]，键入 hadoop@*MasterPublicDNS*。例如，hadoop@*ec2-###-##-##-###.compute-1.amazonaws.com*。
5. 在 类别 (Category) 列表中，选择 Connection (连接) > SSH, Auth (SSH,身份验证)。
6. 对于 Private key file for authentication (用于身份验证的私有密钥文件)，选择 Browse (浏览)，然后选择您生成的 .ppk 文件。
7. 依次选择 Open (打开)、Yes (是) 以取消 PuTTY 安全警告。

> **Important**
>
> 登录主节点时，如果系统提示您输入用户名，请键入 hadoop。

8. 完成主节点上的工作以后，您可以关闭 PuTTY，从而关闭 SSH 连接。

> **Note**
>
> 为了防止 SSH 连接超时，您可以在 Category (类别) 列表中选择 Connection (连接)，然后选择 Enable TCP_keepalives (启用 TCP_keepalives) 选项。如果在 PuTTY 中有活动的 SSH 会话，则可以通过打开 PuTTY 标题栏的上下文（右键单击）菜单，然后选择 Change Settings (更改设置) 来更改设置。

如果您在使用 SSH 连接到主节点时遇到困难，请参阅排查实例的连接问题。

使用 Amazon CLI 连接到主节点

您可以在 Windows、Linux、Unix 和 Mac OS X 上使用 Amazon CLI 创建与主节点之间的 SSH 连接。无论使用何种平台，都需要主节点的公有 DNS 名称和 Amazon EC2 密钥对私有密钥。如果在 Linux、Unix 或 Mac OS X 上使用 Amazon CLI，则还必须对私有密钥（`.pem` 或 `.ppk`）文件设置权限（如 配置密钥对私有密钥文件权限 所示）。

使用 Amazon CLI 连接到主节点

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 要检索集群标识符，请键入：

```
aws emr list-clusters
```

输出会列出您的集群（包括集群 ID）。记下您连接的集群的集群 ID。

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "AWS CLI cluster"
```

3. 键入以下命令以打开与主节点的 SSH 连接。在下面的示例中，将 *j-2AL4XXXXXX5T9* 替换为集群 ID，将 *~/mykeypair.key* 替换为 .pem 文件（对于 Linux、Unix 和 Mac OS X）或 .ppk 文件（对于 Windows）的完整路径和文件名。例如 C:\Users\<username>\.ssh\mykeypair.pem。

```
aws emr ssh --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

4. 完成主节点上的工作后，请关闭 Amazon CLI 窗口。

有关更多信息，请参阅 Amazon CLI 中的 Amazon EMR 命令。如果您在使用 SSH 连接到主节点时遇到困难，请参阅排查实例的连接问题。

Amazon EMR 服务端口

> **Note**
>
> 以下是适用于 Amazon EMR 上的组件的接口和端口。这不是服务端口的完整列表。未列出非默认服务，例如 SSL 端口和不同类型的协议。

> **Important**
>
> 在编辑安全组规则以打开端口时要小心。请务必为运行工作负载所需的协议和端口添加仅允许来自可信和经过身份验证的客户的流量的规则。

组件	服务描述	默认情况下正在运行的服务	端口	配置键
Hadoop	HTTP KMS REST API	是	9600	hadoop.kms.http.port
HDFS	Namenode Web UI	是	9870	dfs.namenode.http-address
	Namenode RPC	是	8020	dfs.namenode.rpc-address

组件	服务描述	默认情况下正在运行的服务	端口	配置键
	数据节点 Web UI	是	9864	dfs.datanode.http.address
	用于数据传输的数据节点 HTTP	是	9866	dfs.datanode.address
	用于数据传输的数据节点 RPC	是	9867	dfs.datanode.ipc.address
Hive	HiveServer2 Thrift	是	10000	hive.server2.thrift.port
	HiveServer2 HTTP	否	10001	hive.server2.thrift.http.port
	HiveServer2 Web UI	是	10002	hive.server2.webui.port
	Hive 元存储	是	9083	hive.metastore.port / metastore.thrift.port
	WebHCat	否	50111	templeton.port
	LLAP 进程守护程序管理服务 (RPC)	否	15004	hive.llap.management.rpc.port
	用于 LLAP 进程守护程序托管的随机排序的 YARN 随机排序端口	否	15551	hive.llap.daemon.yarn.shuffle.port
	LLAP 进程守护程序 RPC	否	动态	hive.llap.daemon.rpc.port

组件	服务描述	默认情况下正在运行的服务	端口	配置键
	LLAP 进程守护程序 Web UI	否	15002	hive.llap.daemon.web.port
	LLAP 进程守护程序输出服务	否	15003	hive.llap.daemon.output.service.port
Oozie		是	11000	
Tez	Tez UI	是	8080	
YARN	随机排序	是	13562	mapreduce.shuffle.port
	Localizer RPC	是	8040	yarn.nodemanager.localizer.address
		是	8041	
	NM Webapp 地址	是	8042	yarn.nodemanager.webapp.address
	RM Web 应用程序	是	8088	yarn.resourcemanager.webapp.address
		是	8025	
	计划程序	是	8030	yarn.resourcemanager.scheduler.address

组件	服务描述	默认情况下正在运行的服务	端口	配置键
	应用程序管理器界面	是	8032	yarn.resourcemanager.address
	RM 管理界面	是	8033	yarn.resourcemanager.admin.address
	JobHistory 服务器 Web UI	是	19888	mapreduce.jobhistory.webapp.address
	JobHistory 服务器管理 Web UI	是	10033	mapreduce.jobhistor.address
	JobHistory 服务器 (RPC)	是	10020	mapreduce.jobhistory.address
	应用程序时间表服务器 (RPC)	是	10200	yarn.timeline-service.address
	应用程序时间表服务器 HTTP Web UI	是	8188	yarn.timeline-service.webapp.address
	应用程序时间表服务器 HTTPS Web UI	否	8190	yarn.timeline-service.webapp.https.address
		是	20888	
Zookeeper	客户端端口	是	2181	
		是	37301	

组件	服务描述	默认情况下正在运行的服务	端口	配置键
		是	8341	

查看 Amazon EMR 集群上托管的 Web 界面

> **Important**
>
> 可以配置自定义安全组，以允许对这些 Web 界面的入站访问。请记住，您允许入站流量通过的任何端口，都代表着潜在的安全漏洞。请仔细检查自定义安全组，以确保您最大限度地减少漏洞。有关更多信息，请参阅使用安全组控制网络流量。

安装在 EMR 集群上的 Hadoop 和其他应用程序会将用户界面发布为主节点上托管的网站。出于安全原因，在使用 Amazon EMR 托管的安全组时，这些网站仅在主节点的本地 Web 服务器上可用。因此，您需要连接到主节点才能查看 Web 界面。有关更多信息，请参阅使用 SSH 连接到主节点。Hadoop 还将用户界面发布为托管在核心节点和任务节点上的网站。这些网站也仅在节点上的本地 Web 服务器上可用。

下表列出了您可以在集群实例上查看的 Web 界面。这些 Hadoop 界面在所有集群上均可用。对于主实例界面，将 *master-public-dns-name* 替换为 Amazon EMR 控制台中集群摘要选项卡上列出的主控公有 DNS。对于核心和任务实例界面，将 *coretask-public-dns-name* 替换为为实例列出的 Public DNS name (公有 DNS 名称)。要查找某一实例的公有 DNS 名称，请在 Amazon EMR 控制台中，从列表中选择您的集群，选择硬件选项卡，选择包含要连接实例的实例组的 ID，然后记下为该实例列出的公有 DNS 名称。

界面名称	URI
Fink 历史记录服务器（EMR 版本 5.33 及更高版本）	http://*master-public-dns-name* :8082/
Ganglia	http://*master-public-dns-name* /ganglia/
Hadoop HDFS NameNode（6.x 之前的 EMR 版本）	https://*master-public-dns-name* :50470/

界面名称	URI
Hadoop HDFS NameNode	http://*master-public-dns-name* :50070/
Hadoop HDFS DataNode	http://*coretask-public-dns-name* :50075/
Hadoop HDFS NameNode（EMR 版本 6.x）	https://*master-public-dns-name* :9870/
Hadoop HDFS DataNode（6.x 之前的 EMR 版本）	https://*coretask-public-dns-name* :50475/
Hadoop HDFS DataNode（EMR 版本 6.x）	https://*coretask-public-dns-name* :9865/
HBase	http://*master-public-dns-name* :16010/
Hue	http://*master-public-dns-name* :8888/
JupyterHub	https://*master-public-dns-name* :9443/
Livy	http://*master-public-dns-name* :8998/
Spark HistoryServer	http://*master-public-dns-name* :18080/
Tez	http://*master-public-dns-name* :8080/tez-ui
YARN NodeManager	http://*coretask-public-dns-name* :8042/
YARN ResourceManager	http://*master-public-dns-name* :8088/
Zeppelin	http://*master-public-dns-name* :8890/

由于主节点上有几个特定于应用程序的界面在核心节点和任务节点上不可用，所以本文档中的说明特定于 Amazon EMR 主节点。访问核心节点和任务节点上 Web 界面的方式与访问主节点上 Web 界面的方式相同。

您可以通过多种方式访问主节点上的 Web 界面。最简单快捷的方法，就是使用 SSH 连接到主节点，并使用基于文本的浏览器 Lynx 在 SSH 客户端中查看网站。但是，Lynx 是一个基于文本的浏览器，

用户界面有限，无法显示图形。以下示例展示了如何使用 Lynx 打开 Hadoop ResourceManager 界面（使用 SSH 登录主节点时也会提供 Lynx URL）。

```
lynx http://ip-###-##-##-###.us-west-2.compute.internal:8088/
```

还有两个其他选项可用于访问提供完整浏览器功能的主节点上的 Web 界面。选择以下选项之一：

- 选项 1（推荐用于更多技术用户）：使用 SSH 客户端连接到主节点，使用本地端口转发配置 SSH 隧道，并使用互联网浏览器打开主节点上托管的 Web 界面。此方法允许您在不使用 SOCKS 代理的情况下配置 Web 界面访问。
- 选项 2（推荐给新用户）：使用 SSH 客户端连接到主节点，使用动态端口转发配置 SSH 隧道，并将互联网浏览器配置为使用附加组件（例如适用于 Firefox 的 FoxyProxy 或适用于 Chrome 的 SwitchyOmega）来管理 SOCKS 代理设置。借助此方法，您可以根据文本模式自动筛选 URL，并将代理设置限制为与主节点 DNS 名称形式匹配的域。有关如何为 Firefox 和 Google Chrome 配置 FoxyProxy 的更多信息，请参阅选项 2，第 2 部分：配置代理设置以查看主节点上托管的网站。。

> **Note**
>
> 如果您通过集群配置修改应用程序运行所在的端口，则指向该端口的超链接将不会在 Amazon EMR 控制台中更新。这是因为控制台没有读取 `server.port` 配置的功能。

使用 Amazon EMR 5.25.0 或更高版本，可以从控制台访问 Spark 历史记录服务器 UI，而无需通过 SSH 连接设置 Web 代理。有关更多信息，请参阅一键访问持久性 Spark 历史记录服务器。

主题

- 选项 1：使用本地端口转发设置到主节点的 SSH 隧道
- 选项 2，第 1 部分：使用动态端口转发设置通往主节点的 SSH 隧道
- 选项 2，第 2 部分：配置代理设置以查看主节点上托管的网站。

选项 1：使用本地端口转发设置到主节点的 SSH 隧道

要连接到主节点上的本地 Web 服务器，您需要在计算机和主节点之间创建 SSH 隧道。这也称为端口转发（port forwarding）。如果您不想使用 SOCKS 代理，可以使用本地端口转发设置通往主节点的 SSH 隧道。通过本地端口转发，您可以指定未使用的本地端口，这些端口用于将流量转发到主节点本地 Web 服务器上的特定远程端口。

使用本地端口转发设置 SSH 隧道，需要主节点的公有 DNS 名称和密钥对私有密钥文件。有关如何查找主节点公有 DNS 名称的信息，请参阅使用旧控制台检索主节点的公有 DNS 名称。有关访问密钥对的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 密钥对。有关您可能希望在主节点上查看的站点的更多信息，请参阅 查看 Amazon EMR 集群上托管的 Web 界面。

使用 OpenSSH 的本地端口转发，设置通往主节点的 SSH 隧道

在终端中使用本地端口转发设置 SSH 隧道

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 打开终端窗口。在 Mac OS X 上，选择 Applications (应用程序) > Utilities (实用程序) > Terminal (终端)。在其它 Linux 发布版上，终端通常位于 Applications (应用程序) > Accessories (附件) > Terminal (终端)。
3. 键入以下命令，以在本地计算机上打开 SSH 隧道。该示例命令通过将本地端口 8157（随机选择的未使用本地端口）上的流量，转发到主节点本地 Web 服务器上的端口 8088，来访问 ResourceManager Web 界面。

   将该命令中的 *~/mykeypair.pem* 替换为您的 .pem 文件的位置和文件名，并将 *ec2-###-##-##-###.compute-1.amazonaws.com* 替换为您的集群的主节点公有 DNS 名称。要访问不同的 Web 界面，请将 8088 替换为适当的端口号。例如，对于 Zeppelin 接口，请将 8088 替换为 8890。

   ```
   ssh -i ~/mykeypair.pem -N -L 8157:ec2-###-##-##-###.compute-1.amazonaws.com:8088 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
   ```

   -L 代表使用本地端口转发，由此，您就能指定一个本地端口，用于将数据转发到主节点本地 Web 服务器上标识的远程端口。

   在您发出此命令后，终端保持打开状态并且不返回响应。
4. 要在浏览器中打开 ResourceManager Web 界面，请在地址栏中键入 `http://localhost:8157/`。
5. 如果您已完成使用主节点上的 Web 界面，请关闭终端窗口。

选项 2，第 1 部分：使用动态端口转发设置通往主节点的 SSH 隧道

要连接到主节点上的本地 Web 服务器，您需要在计算机和主节点之间创建 SSH 隧道。这也称为端口转发（port forwarding）。如果您使用动态端口转发创建 SSH 隧道，则路由到指定未使用本地端口的

所有流量都将转发到主节点上的本地 Web 服务器。这将创建 SOCKS 代理。然后，您可以将互联网浏览器配置为使用附加组件（例如 FoxyProxy 或 SwitchyOmega）来管理您的 SOCKS 代理设置。

使用代理管理附加组件，您可以根据文本模式自动筛选 URL，并将代理设置限制为与主节点公有 DNS 名称形式匹配的域。当您在查看主节点上托管的网站和互联网上托管的网站之间进行切换时，浏览器附加组件会自动处理打开和关闭代理的操作。

在开始之前，您需要主节点的公有 DNS 名称和密钥对私有密钥文件。有关如何查找主节点公有 DNS 名称的信息，请参阅 使用旧控制台检索主节点的公有 DNS 名称。有关访问密钥对的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 Amazon EC2 密钥对。有关您可能希望在主节点上查看的站点的更多信息，请参阅 查看 Amazon EMR 集群上托管的 Web 界面。

使用 OpenSSH 的动态端口转发，设置通往主节点的 SSH 隧道

通过 OpenSSH 使用动态端口转发设置 SSH 隧道

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 打开终端窗口。在 Mac OS X 上，选择 Applications (应用程序) > Utilities (实用程序) > Terminal (终端)。在其它 Linux 发布版上，终端通常位于 Applications (应用程序) > Accessories (附件) > Terminal (终端)。
3. 键入以下命令以在本地计算机上打开 SSH 隧道。将 *~/mykeypair.pem* 替换为您的 .pem 文件的位置和文件名，将 *8157* 替换为一个未使用的本地端口号，并将 *c2-###-##-##-###.compute-1.amazonaws.com* 替换为您的集群的主节点公有 DNS 名称。

   ```
   ssh -i ~/mykeypair.pem -N -D 8157 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
   ```

   在您发出此命令后，终端保持打开状态并且不返回响应。

   > **Note**
   >
   > -D 表示使用动态端口转发，它允许您指定一个本地端口，用于将数据转发到主节点本地 Web 服务器上的所有远程端口。动态端口转发会创建一个本地 SOCKS 代理，侦听命令中指定的端口。

4. 隧道处于活动状态后，请为您的浏览器配置 SOCKS 代理。有关更多信息，请参阅选项 2，第 2 部分：配置代理设置以查看主节点上托管的网站。。
5. 如果您已完成使用主节点上的 Web 界面，请关闭终端窗口。

通过 Amazon CLI 使用动态端口转发设置 SSH 隧道

您可以在 Windows 、Linux、Unix 和 Mac OS X 上使用 Amazon CLI 创建与主节点之间的 SSH 连接。如果在 Linux、Unix 或 Mac OS X 上使用 Amazon CLI，则必须对 .pem 文件设置权限（如 配置密钥对私有密钥文件权限 所示）。如果在 Windows 上使用 Amazon CLI，则 PuTTY 必须在路径环境变量中，否则您可能会收到错误（如 OpenSSH or PuTTY not available (OpenSSH 或 PuTTY 不可用））。

通过 Amazon CLI 使用动态端口转发设置 SSH 隧道

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 创建与主节点的 SSH 连接，如 使用 Amazon CLI 连接到主节点 所示。
3. 要检索集群标识符，请键入：

```
aws emr list-clusters
```

输出会列出您的集群（包括集群 ID）。记下您连接的集群的集群 ID。

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "AWS CLI cluster"
```

4. 键入以下命令，以使用动态端口转发打开通往主节点的 SSH 隧道。在下面的示例中，将 *j-2AL4XXXXXX5T9* 替换为集群 ID，将 *~/mykeypair.key* 替换为 .pem 文件 (对于 Linux、Unix 和 Mac OS X) 或 .ppk 文件 (对于 Windows) 的位置和文件名。

```
aws emr socks --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

> **Note**
>
> socks 命令会自动配置本地端口 8157 上的动态端口转发。目前，此设置无法修改。

5. 隧道处于活动状态后，请为您的浏览器配置 SOCKS 代理。有关更多信息，请参阅选项 2，第 2 部分：配置代理设置以查看主节点上托管的网站。。
6. 如果您已完成使用主节点上的 Web 界面，请关闭 Amazon CLI 窗口。

   有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

使用 PuTTY 设置通往主节点的 SSH 隧道

Windows 用户可以使用 SSH 客户端（如 PuTTY），创建通往主节点的 SSH 隧道。连接到 Amazon EMR 主节点之前，应下载并安装 PuTTY 和 PuTTYgen。可以从 PuTTY 下载页面下载这些工具。

PuTTY 本身不支持 Amazon EC2 生成的密钥对私有密钥文件格式（`.pem`）。您可使用 PuTTYgen 将密钥文件转换为所需的 PuTTY 格式（`.ppk`）。您必须将密钥转换为此格式 (`.ppk`)，然后才能尝试使用 PuTTY 连接到主节点。

有关转换密钥的更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用 PuTTYgen 转换私有密钥。

通过 PuTTY 使用动态端口转发设置 SSH 隧道

1. 确保您已允许入站 SSH 流量。有关说明，请参阅 连接之前：授权入站流量。
2. 双击 `putty.exe` 以启动 PuTTY。也可以从 Windows 程序列表中启动 PuTTY。

> **Note**
>
> 如果您已经有与主节点之间的活动 SSH 会话，则可以通过右键单击 PuTTY 标题栏并选择 Change Settings（更改设置）来添加隧道。

3. 如有必要，在 Category (类别) 列表中，选择 Session (会话)。
4. 在 Host Name (主机名) 字段，键入 **hadoop@***MasterPublicDNS*。例如，**hadoop@***ec2-###-##-##-###.compute-1.amazonaws.com*。
5. 在 Category (类别) 列表中，展开 Connection (连接) > SSH，然后选择 Auth (身份验证)。

6. 对于 Private key file for authentication (用于身份验证的私有密钥文件)，选择 Browse (浏览)，然后选择您生成的 .ppk 文件。

> **Note**
>
> PuTTY 本身不支持 Amazon EC2 生成的密钥对私有密钥文件格式（.pem）。您可使用 PuTTYgen 将密钥文件转换为所需的 PuTTY 格式（.ppk）。您必须将密钥转换为此格式 (.ppk)，然后才能尝试使用 PuTTY 连接到主节点。

7. 在 Category (类别) 列表中，展开 Connection (连接) > SSH，然后选择 Tunnels (隧道)。
8. 在 Source port (源端口) 字段，键入 8157（未使用的本地端口），然后选择 Add (添加)。
9. 将 Destination (目标) 字段留空。
10. 选择 Dynamic (动态) 和 Auto (自动) 选项。
11. 选择 Open (打开)。
12. 选择 Yes (是) 以取消 PuTTY 安全警告。

> **Important**
>
> 登录主节点时，如果系统提示您输入用户名，请键入 hadoop。

13. 隧道处于活动状态后，请为您的浏览器配置 SOCKS 代理。有关更多信息，请参阅选项 2，第 2 部分：配置代理设置以查看主节点上托管的网站。。
14. 如果您已完成使用主节点上的 Web 界面，请关闭 PuTTY 窗口。

选项 2，第 2 部分：配置代理设置以查看主节点上托管的网站。

如果您使用带有动态端口转发的 SSH 隧道，则必须使用 SOCKS 代理管理附加组件来控制浏览器中的代理设置。使用 SOCKS 代理管理工具，您可以根据文本模式自动筛选 URL，并将代理设置限制为与主节点公有 DNS 名称形式匹配的域。当您在查看主节点上托管的网站和互联网上托管的网站之间进行切换时，浏览器附加组件会自动处理打开和关闭代理的操作。要管理代理设置，请将浏览器配置为使用附加组件（例如 FoxyProxy 或 SwitchyOmega）。

有关创建 SSH 隧道的更多信息，请参阅选项 2，第 1 部分：使用动态端口转发设置通往主节点的 SSH 隧道。有关可用 Web 界面的更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

在设置代理附加组件时，请包括以下设置：

- 使用 localhost 作为主机地址。
- 使用您选择的相同本地端口号与 选项 2，第 1 部分：使用动态端口转发设置通往主节点的 SSH 隧道 中的主节点建立 SSH 隧道。例如，端口 *8157*。此端口还必须与您在 PuTTY 或用于连接的任何其它终端仿真器中使用的端口号相匹配。
- 指定 SOCKS v5 协议。SOCKS v5 允许您有选择地设置用户授权。
- URL 模式

  以下 URL 模式应列入允许列表并使用通配符模式类型指定：

  - *ec2*.*compute*.amazonaws.com* 和 *10*.amazonaws.com* 模式与美国区域中集群的公有 DNS 名称匹配。
  - *ec2*.compute* 和 *10*.compute* 模式与所有其它区域中集群的公有 DNS 名称匹配。
  - 10.* 模式可提供对 Hadoop 中 JobTracker 日志文件的访问权限。如果此筛选条件与您的网络访问计划冲突，请更改此筛选条件。
  - *.ec2.internal* 和 *.compute.internal* 模式，分别与 `us-east-1` 区域和所有其它区域中集群的私有（内部）DNS 名称相匹配。

示例：为 Firefox 配置 FoxyProxy

以下示例演示了 Mozilla Firefox 的 FoxyProxy Standard（版本 7.5.1）配置。

FoxyProxy 提供了一套代理管理工具。该工具有便于您将代理服务器用于 URL，这些 URL 与 Amazon EMR 集群中 Amazon EC2 实例所使用的域对应模式相匹配。

使用 Mozilla Firefox 安装和配置 FoxyProxy

1. 在 Firefox 中，转到 https://addons.mozilla.org/，搜索 FoxyProxy Standard，然后按照说明将 FoxyProxy 添加到 Firefox。
2. 使用文本编辑器，根据以下示例配置创建名为 `foxyproxy-settings.json` 的 JSON 文件。

```
{
  "k20d21508277536715": {
    "active": true,
    "address": "localhost",
    "port": 8157,
    "username": "",
    "password": "",
    "type": 3,
    "proxyDNS": true,
```

```
    "title": "emr-socks-proxy",
    "color": "#0055E5",
    "index": 9007199254740991,
    "whitePatterns": [
      {
        "title": "*ec2*.*compute*.amazonaws.com*",
        "active": true,
        "pattern": "*ec2*.*compute*.amazonaws.com*",
        "importedPattern": "*ec2*.*compute*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*ec2*.compute*",
        "active": true,
        "pattern": "*ec2*.compute*",
        "importedPattern": "*ec2*.compute*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "10.*",
        "active": true,
        "pattern": "10.*",
        "importedPattern": "http://10.*",
        "type": 1,
        "protocols": 2
      },
      {
        "title": "*10*.amazonaws.com*",
        "active": true,
        "pattern": "*10*.amazonaws.com*",
        "importedPattern": "*10*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*10*.compute*",
        "active": true,
        "pattern": "*10*.compute*",
        "importedPattern": "*10*.compute*",
        "type": 1,
        "protocols": 1
      },
```

```
      {
        "title": "*.compute.internal*",
        "active": true,
        "pattern": "*.compute.internal*",
        "importedPattern": "*.compute.internal*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*.ec2.internal* ",
        "active": true,
        "pattern": "*.ec2.internal*",
        "importedPattern": "*.ec2.internal*",
        "type": 1,
        "protocols": 1
      }
    ],
    "blackPatterns": []
  },
  "logging": {
    "size": 100,
    "active": false
  },
  "mode": "patterns",
  "browserVersion": "68.12.0",
  "foxyProxyVersion": "7.5.1",
  "foxyProxyEdition": "standard"
}
```

3. 打开 Firefox Manage Your Extensions (管理您的扩展) 页面（转到 about:addons，然后选择 Extensions (扩展)）。
4. 选择 FoxyProxy Standard，然后选择更多选项按钮（看起来像省略号的按钮）。
5. 从下拉列表中选择 Options (选项)。
6. 从左侧菜单中选择 Import Settings (导入设置)。
7. 在 Import Settings (导入设置) 页面上，选择 Import Settings from FoxyProxy 6.0+ (从 FoxyProxy 6.0+ 导入设置) 下的 Import Settings (导入设置)，浏览到您创建的 `foxyproxy-settings.json` 文件所在的位置，选择该文件，然后选择 Open (打开)。
8. 当系统提示覆盖现有设置并保存新配置时，选择 OK (确定)。

示例：为 Chrome 配置 SwitchyOmega

以下示例演示了如何为 Google Chrome 设置 SwitchyOmega 扩展。SwitchyOmega 允许您配置、管理和在多个代理之间切换。

使用 Google Chrome 安装和配置 SwitchyOmega

1. 前往 https://chrome.google.com/webstore/category/extensions，搜索 Proxy SwitchyOmega，并将其添加到 Chrome。
2. 选择 New profile (新建配置文件) 并输入 `emr-socks-proxy` 作为配置文件名称。
3. 选择 PAC profile (PAC 配置文件) 然后选择 Create (创建)。代理自动配置 (PAC) 文件，可帮助您定义应转发到 Web 代理服务器的浏览器请求的允许列表。
4. 将 PAC Script（PAC 脚本）字段的内容替换为以下脚本，该脚本定义应通过您 Web 代理服务器转发的 URL。如果您在设置 SSH 隧道时指定了不同的端口号，请将 *8157* 替换为您的端口号。

```
function FindProxyForURL(url, host) {
    if (shExpMatch(url, "*ec2*.*compute*.amazonaws.com*")) return 'SOCKS5
 localhost:8157';
    if (shExpMatch(url, "*ec2*.compute*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "http://10.*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*10*.compute*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*10*.amazonaws.com*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*.compute.internal*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*ec2.internal*")) return 'SOCKS5 localhost:8157';
    return 'DIRECT';
}
```

5. 在 Actions (操作) 下，选择 Apply changes (应用更改) 以保存您的代理设置。
6. 在 Chrome 工具栏上，选择 SwitchyOmega 并 `emr-socks-proxy` 配置文件。

访问浏览器中的 Web 界面

要打开 Web 界面，请在浏览器地址栏中输入主节点或核心节点的公有 DNS 名称，后跟所选界面的端口号。以下示例显示了您要输入用来连接到 Spark HistoryServer 的 URL。

```
http://master-public-dns-name:18080/
```

有关检索节点的公有 DNS 名称的说明，请参阅检索主节点的公有 DNS 名称。有关 Web 界面 URL 的完整列表，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

向集群提交工作

此部分描述您可用于向 Amazon EMR 集群提交工作的方法。您可以通过添加步骤或以交互方式向主节点提交 Hadoop 任务的形式向集群提交工作。

向集群提交步骤时，请考虑以下步骤行为规则：

- 步骤 ID 最多可包含 256 个字符。
- 一个集群中最多可以有 256 个待处理和正在运行的步骤。
- 即使集群上运行有 256 个活动步骤，您可以采用交互方式向主节点提交任务。您可以在长时间运行的集群的生命周期内提交不限数量的步骤，但是，在给定时间只能有 256 个步骤处于 RUNNING 或 PENDING 状态。
- 对于 Amazon EMR 版本 4.8.0 和更高版本（版本 5.0.0 除外），您可以取消待处理步骤。有关更多信息，请参阅取消步骤。
- 使用 Amazon EMR 版本 5.28.0 和更高版本，可以取消待处理步骤和正在运行的步骤。您还可以选择并行运行多个步骤以提高集群利用率并节省成本。有关更多信息，请参阅并行运行多个步骤的注意事项。

> **Note**
>
> 为了获得最佳性能，我们建议您将要与 Amazon EMR 配合使用的自定义引导操作、脚本和其他文件存储在某一 Amazon S3 存储桶中，该存储桶与您的集群位于相同的 Amazon Web Services 区域 中。

主题

- 使用 Amazon EMR 管理控制台向集群添加步骤
- 使用 Amazon CLI 向集群添加步骤
- 并行运行多个步骤的注意事项
- 查看步骤
- 取消步骤

使用 Amazon EMR 管理控制台向集群添加步骤

在 Amazon Web Services Management Console 中，请按照以下操作步骤将步骤添加到集群中。有关如何为特定大数据应用程序提交步骤的详细信息，请参阅 Amazon EMR Release Guide：

- 提交自定义 JAR 步骤
- 提交 Hadoop 流式传输步骤
- 提交 Spark 步骤
- 提交 Pig 步骤
- 将命令或脚本作为步骤运行
- 将值传递到运行 Hive 脚本的步骤中

在创建集群过程中添加步骤

在 Amazon Web Services Management Console 中，您可以在创建集群时添加步骤。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

在使用新控制台创建集群时添加步骤

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Steps（步骤）下，选择 Add step（添加步骤）。在 Add Step（添加步骤）对话框的字段中输入相应值。有关格式化步骤参数的信息，请参阅 添加步骤参数。选项因步骤类型而异。要添加步骤并退出对话框，请选择添加步骤。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

在使用旧控制台创建集群时添加步骤

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/home。选择 Create cluster - Advanced Options（创建集群 - 高级选项）。
2. 在 Step 1: Software and Steps (步骤 1：软件和步骤) 页面上，对于 Steps (optional) (步骤（可选）)，选择 Run multiple steps in parallel to improve cluster utilization and save cost (并行运行多个步骤以提高集群利用率并节省成本)。并发级别的默认值为 10。您可以在 2 到 256 个步骤之间进行选择，这些步骤可以并行运行。

   > **Note**
   >
   > 只有 Amazon EMR 版本 5.28.0 和更高版本支持并行运行多个步骤。

3. 对于 After last step completes (在最后一步完成后)，选择 Cluster enters waiting state (集群进入等待状态) 或 Auto-terminate the cluster (自动终止集群)。
4. 选择 Step type (步骤类型)，然后选择 Add step (添加步骤)。
5. 在 Add Step (添加步骤) 对话框的各个字段中键入合适的值。有关格式化步骤参数的信息，请参阅 添加步骤参数。选项因步骤类型而异。如果您已启用并行运行多个步骤以提高集群利用率并节省成本，则出现故障时的操作的唯一选项是继续。接下来，选择 Add (添加)。

向正在运行的集群添加步骤

您可以使用 Amazon Web Services Management Console 向已禁用自动终止选项的集群添加步骤。

New console

使用新控制台向正在运行的集群添加步骤

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Steps（步骤）选项卡上，选择 Add step（添加步骤）。要克隆现有步骤，请选择 Actions（操作）下拉菜单，然后选择 Clone step（克隆步骤）。
4. 在 Add Step（添加步骤）对话框的字段中输入相应值。选项因步骤类型而异。要添加步骤并退出对话框，请选择 Add step（添加步骤）。

Old console

使用旧控制台向正在运行的集群添加步骤

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/home。在 Cluster List (集群列表) 页面上，选择您的集群的链接。
2. 在 Cluster Details (集群详细信息) 页面上，选择 Steps (步骤) 选项卡。
3. 在 Steps (步骤) 选项卡上，选择 Add step (添加步骤)。
4. 在 Add Step (添加步骤) 对话框的各个字段中键入相应的值，然后选择 Add (添加)。这些选项因步骤类型而异。

修改正在运行的集群中的步骤并发级别

使用 Amazon Web Services Management Console，可以修改正在运行的集群中的步骤并发级别。

> **Note**
> 您只能使用 Amazon EMR 版本 5.28.0 和更高版本并行运行多个步骤。

New console

使用新控制台，修改正在运行的集群中的步骤并发级别

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。集群必须处于运行状态才能更改其并发属性。
3. 在集群详细信息页面的 Steps（步骤）选项卡上，找到 Attributes（属性）部分。选择 Edit（编辑）以更改并发级别。请输入 1 到 256 之间的值。

Old console

使用旧控制台，修改正在运行的集群中的步骤并发级别

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/home。在 Cluster List (集群列表) 页面上，选择您的集群的链接。
2. 在 Cluster Details (集群详细信息) 页面上，选择 Steps (步骤) 选项卡。

3. 对于 Concurrency (并发)，请选择 Change (更改)。为步骤并发级别选择一个新值，然后保存。

添加步骤参数

在使用 Amazon Web Services Management Console 添加步骤到您的集群时，您可以在参数字段指定该步骤的参数。您必须使用空格、字符组成的两边字符串参数以及带引号的空格分隔参数。

Example ：参数正确

以下示例参数的格式正确，适用于Amazon Web Services Management Console，最后一个字符串参数两边用引号。

```
bash -c "aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh ."
```

您还可以为了实现可读性，将每个参数放在单独的行中，如下面的示例所示。

```
bash
-c
"aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh ."
```

Example ：参数不正确

以下示例参数的格式不正确，不适用于 Amazon Web Services Management Console。请注意，最后一个字符串参数 `aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh .` 含有空格，没有引号。

```
bash -c aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh .
```

使用 Amazon CLI 向集群添加步骤

以下过程演示如何使用 Amazon CLI 向新创建的集群和正在运行的集群添加步骤。两个示例都使用 `--steps` 子命令向集群添加步骤。

在集群创建过程中添加步骤

- 键入以下命令以创建集群并添加一个 Apache Pig 步骤。确保将 *myKey* 替换为您的 Amazon EC2 密钥对的名称。

```
aws emr create-cluster --name "Test cluster" \
```

```
--applications Name=Spark \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-groups InstanceGroupType=PRIMARY,InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]'
```

> **Note**
>
> 参数列表因步骤类型而异。

默认情况下，步骤并发级别为 1。您可以在创建集群时使用 StepConcurrencyLevel 参数设置步骤并发级别。

输出是类似于以下内容的集群标识符。

```
{
    "ClusterId": "j-2AXXXXXXGAPLF"
}
```

向正在运行的集群添加步骤

- 键入以下命令以向正在运行的集群添加步骤。使用您自己的集群 ID 替换 *j-2AXXXXXXGAPLF*。

```
aws emr add-steps --cluster-id j-2AXXXXXXGAPLF \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]'
```

输出是类似于以下内容的步骤标识符。

```
{
    "StepIds": [
 "s-Y9XXXXXXAPMD"
    ]
```

```
}
```

修改正在运行的集群中的 StepConcurrencyLevel

1. 在正在运行的集群中，您可以使用 ModifyCluster API 修改 StepConcurrencyLevel。例如，键入以下命令将 StepConcurrencyLevel 增加到 10。使用您的集群 ID 替换 *j-2AXXXXXXGAPLF*。

```
aws emr modify-cluster --cluster-id j-2AXXXXXXGAPLF --step-concurrency-level 10
```

2. 输出类似于以下内容。

```
{
"StepConcurrencyLevel": 10
}
```

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅 Amazon CLI 命令参考。

并行运行多个步骤的注意事项

- 并行运行的步骤可以按任何顺序完成，但队列中的待处理步骤要按照它们提交的顺序转换为运行状态。
- 为集群选择步骤并发级别时，必须考虑主节点实例类型是否满足用户工作负载的内存要求。主步骤执行程序进程在每个步骤的主节点上运行。与一次运行一个步骤相比，并行运行多个步骤要求主节点拥有更多的内存和更高的 CPU 利用率。
- 要对并发步骤实现复杂的计划和资源管理，您可以使用 YARN 计划功能（如 FairScheduler 或 CapacityScheduler）。例如，您可以将 FairScheduler 与 queueMaxAppsDefault 集合一起使用，以防止一次运行超过特定数量的作业。
- 步骤并发级别取决于资源管理器的配置。例如，如果 YARN 配置的并行度仅为 5，则即使 StepConcurrencyLevel 设置为 10，您也只能让五个 YARN 应用程序并行运行。有关配置资源管理器的更多信息，请参阅《Amazon EMR 版本指南》中的配置应用程序。
- 当集群的步骤并发级别大于 1 时，不能添加具有 ActionOnFailure（CONTINUE 除外）的步骤。
- 如果集群的步骤并发级别大于 1，步骤 ActionOnFailure 功能将不会激活。
- 如果集群的步骤并发级别为 1 但有多个运行步骤，则 TERMINATE_CLUSTER ActionOnFailure 可能会激活，但 CANCEL_AND_WAIT ActionOnFailure 不会。当集群步骤并发级别大于一时会出现这种边缘情况，但在运行多个步骤时会减少发生该情况。

- 使用 EMR 自动扩展可以基于 YARN 资源向上和向下扩展，以防止资源争用。有关更多信息，请参阅《Amazon EMR 管理指南》中的将自动伸缩与实例组的自定义策略结合使用。
- 当您降低步骤并发级别时，EMR 允许在减少步骤数之前完成任何正在运行的步骤。如果资源因集群运行的并发步骤太多而耗尽，我们建议手动取消所有正在运行的步骤以释放资源。

查看步骤

可以查看的步骤记录总数 (与状态无关) 是 1000。此总数包括用户提交的步骤和系统步骤。随着用户提交的步骤的状态更改为 COMPLETED 或 FAILED，可以向集群添加用户提交的其它步骤，直到达到 1000 的步骤限制。向集群添加了 1000 个步骤之后，提交更多步骤会导致删除用户提交的较旧步骤记录。这些记录不会从日志文件中删除，而是从控制台显示中删除，因此当您使用 Amazon CLI 或 API 检索集群信息时不会出现。任何时候都不会删除系统步骤记录。

可以查看的步骤信息取决于用于检索集群信息的机制。下表显示每个可用选项所返回的步骤信息。

选项	DescribeJobFlow 或 --describe --jobflow	ListSteps 或 list-steps
SDK	256 个步骤	1000 个步骤
Amazon EMR CLI	256 个步骤	NA
Amazon CLI	NA	1000 个步骤
API	256 个步骤	1000 个步骤

取消步骤

您可以通过 Amazon Web Services Management Console、Amazon CLI 或 Amazon EMR API 取消待处理和正在运行的步骤。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台取消步骤

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Steps（步骤）选项卡上，选中要取消的步骤旁边的复选框。选择 Actions（操作）下拉菜单，然后选择 Cancel steps（取消步骤）。
4. 在 Cancel the step（取消步骤）对话框中，选择取消步骤并等待退出，或者取消步骤并强制退出。然后，选择 Confirm（确认）。
5. Steps（步骤）表中的步骤状态更改为 CANCELLED。

Old console

使用旧控制台取消步骤

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 在 Cluster Details (集群详细信息) 页面上，展开 Steps (步骤)部分。
3. 对于要取消的每个步骤，请从 Steps (步骤) 列表中选择步骤。然后选择 Cancel step (取消步骤)。
4. 在 Cancel step (取消步骤) 对话框中，保留默认选项 Cancel the step and wait for it to exit (取消步骤并等待其退出)。如果要立即结束步骤而不等待任何进程完成，请选择 Cancel the step and force it to exit (取消步骤并强制其退出)。
5. 选择 Cancel step (取消步骤)。

CLI

使用 Amazon CLI 取消

- 使用 `aws emr cancel-steps` 命令指定集群和要取消的步骤。以下示例演示了用于取消两个步骤的 Amazon CLI 命令。

```
aws emr cancel-steps --cluster-id j-2QUAXXXXXXXXX \
--step-ids s-3M8DXXXXXXXXX s-3M8DXXXXXXXXX \
```

```
--step-cancellation-option SEND_INTERRUPT
```

使用 Amazon EMR 版本 5.28.0，可以在取消步骤时选择以下两个取消选项之一作为 StepCancellationOption 参数。

- SEND_INTERRUPT– 这是默认选项。当收到步骤取消请求时，EMR 会向步骤发送 SIGTERM 信号。将 SIGTERM 信号处理程序添加到您的步骤逻辑，以捕获此信号并终止子步骤进程，或等待它们完成处理。
- TERMINATE_PROCESS – 选择此选项后，EMR 会向步骤及其所有子进程发送 SIGKILL 信号，然后这些进程会立即终止。

取消步骤的注意事项

- 取消正在运行的步骤或待处理步骤将从活动步骤计数中删除该步骤。
- 假设没有更改 stepConcurrencyLevel，取消正在运行的步骤并不允许待处理步骤开始运行。
- 取消正在运行的步骤不会触发步骤 ActionOnFailure。
- 对于 EMR 5.32.0 及更高版本，SEND_INTERRUPT StepCancellationOption 会向步骤子进程发送 SIGTERM 信号。您应该注意此信号并正常进行清除和关闭。TERMINATE_PROCESS StepCancellationOption 向步骤子进程及其所有后代进程发送 SIGKILL 信号；但是，异步进程不受影响。

查看和监控集群

Amazon EMR 提供多种工具，供您收集集群信息。您可以从控制台、CLI 或者以编程的方式访问有关集群的信息。主节点上提供标准的 Hadoop Web 界面和日志文件。您还可以使用各种监控服务（如 CloudWatch 和 Ganglia）跟踪集群的性能。

从 Amazon EMR 5.25.0 开始，也可从控制台中通过 Spark 历史记录服务器的“持久性”应用程序 UI 获得应用程序历史记录。借助 Amazon EMR 6.x，持久性 YARN 时间线服务器和 Tez 用户界面也可用。这些服务是在集群外托管的，因此您可以在集群终止后的 30 天内访问应用程序历史记录，而无需 SSH 连接或 Web 代理。请参阅查看应用程序历史记录。

主题

- 查看集群状态和详细信息
- 增强型步骤调试

- 查看应用程序历史记录。
- 查看日志文件
- 查看 Amazon EC2 中的集群实例
- CloudWatch 事件和指标
- 使用 Ganglia 查看集群应用程序指标
- 在 Amazon CloudTrail 中记录 Amazon EMR API 调用

查看集群状态和详细信息

创建集群后，您可以监控其状态和获取有关其执行和可能发生的错误的详细信息 (即使在集群终止后也是如此)。Amazon EMR 会将有关已终止集群的元数据保存两个月（之后将删除）以供您参考。虽然您无法从集群历史记录中删除集群，但借助Amazon Web Services Management Console，您可使用 Filter (筛选条件) 来聚焦您关注的集群，而借助 Amazon CLI，您可使用选项与 `list-clusters` 命令来聚焦您关注的集群。

无论集群是在运行中还是已终止，您都可以访问从开始记录集群的一周内存储在集群中的应用程序历史记录。此外，在集群终止后，持久性应用程序用户界面会在集群外存储应用程序历史记录达 30 天。请参阅查看应用程序历史记录。

有关集群状态 [例如“Waiting (正在等待)”和“Running (正在运行)”] 的更多信息，请参阅了解集群的生命周期。

使用 Amazon Web Services Management Console 查看集群详细信息

https://console.aws.amazon.com/emr 中的 Clusters（集群）列表列出了您账户和 Amazon 区域中的所有集群，包括已终止的集群。该列表显示了每个集群的以下信息：名称和 ID、状态和状态详细信息、创建时间、集群已运行的已用时间，以及集群中所有 EC2 实例累计的标准实例小时数。该列表是监控您的集群状态的起点，它旨在使您能够深入了解每个集群的详细信息以进行分析和故障排除。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台查看集群信息

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要查看的集群。
3. 使用 Summary（摘要）面板查看集群配置的基本信息，例如集群状态、Amazon EMR 在集群上安装的开源应用程序以及您用于创建集群的 Amazon EMR 版本。使用“摘要”下的各个选项卡查看信息，如下表所述。

Old console

使用旧控制台查看集群信息

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 要查看集群信息摘要，请选择 Name（名称）下集群链接旁边的向下箭头。集群的行将展开以提供有关集群、硬件、步骤和引导操作的更多信息。利用此部分中的链接可深入了解具体信息。例如，单击 Steps (步骤) 下的链接可访问步骤日志文件，查看与步骤关联的 JAR，深入了解步骤的作业和任务以及访问日志文件。
3. 要深入了解集群信息，请选择 Name（名称）下的集群链接以打开集群详细信息页面。旧控制台的集群详细信息页面上提供了以下信息：

选项卡（旧控制台）	描述（旧控制台）
属性	使用此选项卡可查看集群的操作系统、集群终止和安全配置、VPC 和子网信息，以及 Amazon S3 中的日志存储位置。
Bootstrap actions (引导操作)	使用此选项卡可查看集群在启动时运行的任何引导操作的状态。引导操作用于自定义软件安装和高级配置。有关更多信息，请参阅创建引导操作以安装其它软件。

选项卡（旧控制台）	描述（旧控制台）
监控	使用此选项卡可查看集群操作的关键指标。您可以查看集群级别数据、节点级别数据以及有关 I/O 和数据存储的信息。
实例	使用此选项卡可查看有关您集群中节点的信息，包括 EC2 实例 ID、DNS 名称、EBS 卷和 IP 地址等。
步骤	使用此选项卡可查看您提交的步骤的状态和访问其日志文件。有关步骤的更多信息，请参阅向集群提交工作。
应用程序	使用此选项卡可以查看持久性的集群外 YARN 时间线服务器和 Tez UI 应用程序的详细信息。您还可以查看有关已安装应用程序、集群配置和实例组的信息。当集群正在运行时，集群上的应用程序用户界面可用。
事件	使用此选项卡可查看集群的事件日志。有关更多信息，请参阅通过 CloudWatch 监控 Amazon EMR 事件。
标签	使用此选项卡可查看您应用于集群的任何标签。

使用 Amazon CLI 查看集群详细信息

以下示例说明如何使用 Amazon CLI 检索集群详细信息。有关可用命令的更多信息，请参阅 Amazon EMR 的 Amazon CLI 命令参考。您可以使用 describe-cluster 命令查看集群级别详细信息，包括状态、硬件和软件配置、VPC 设置、引导操作、实例组等。有关集群状态的更多信息，请参阅了解集群的生命周期。以下示例演示了使用 `describe-cluster` 命令的情景，并随后提供了 list-clusters 命令的示例。

Example 查看集群状态

要使用 `describe-cluster` 命令，需要集群 ID。此示例演示使用它获取特定日期范围内创建的集群的列表，然后使用返回的集群 ID 之一列出有关各个集群状态的更多信息。

以下命令描述集群 *j-1K48XXXXXXHCB*（将替换为您的集群 ID）。

```
aws emr describe-cluster --cluster-id j-1K48XXXXXXHCB
```

您的命令的输出与以下内容类似：

```
{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1438281058.061,
                "CreationDateTime": 1438280702.498
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "EmrManagedMasterSecurityGroup": "sg-cXXXXX0",
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2KeyName": "myKey",
            "Ec2AvailabilityZone": "us-east-1c",
            "EmrManagedSlaveSecurityGroup": "sg-example"
        },
        "Name": "Development Cluster",
        "ServiceRole": "EMR_DefaultRole",
        "Tags": [],
        "TerminationProtected": false,
        "ReleaseLabel": "emr-4.0.0",
        "NormalizedInstanceHours": 16,
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1438281058.101,
                        "CreationDateTime": 1438280702.499
```

```
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "Name": "CORE",
            "InstanceGroupType": "CORE",
            "Id": "ig-2EEXAMPLEXXP",
            "Configurations": [],
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "RunningInstanceCount": 1
        },
        {
            "RequestedInstanceCount": 1,
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1438281023.879,
                    "CreationDateTime": 1438280702.499
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "Name": "MASTER",
            "InstanceGroupType": "MASTER",
            "Id": "ig-2A1234567XP",
            "Configurations": [],
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "RunningInstanceCount": 1
        }
    ],
    "Applications": [
        {
            "Version": "1.0.0",
            "Name": "Hive"
        },
        {
            "Version": "2.6.0",
            "Name": "Hadoop"
        },
```

```
            {
                "Version": "0.14.0",
                "Name": "Pig"
            },
            {
                "Version": "1.4.1",
                "Name": "Spark"
            }
        ],
        "BootstrapActions": [],
        "MasterPublicDnsName": "ec2-X-X-X-X.compute-1.amazonaws.com",
        "AutoTerminate": false,
        "Id": "j-jobFlowID",
        "Configurations": [
            {
                "Properties": {
                    "hadoop.security.groups.cache.secs": "250"
                },
                "Classification": "core-site"
            },
            {
                "Properties": {
                    "mapreduce.tasktracker.reduce.tasks.maximum": "5",
                    "mapred.tasktracker.map.tasks.maximum": "2",
                    "mapreduce.map.sort.spill.percent": "90"
                },
                "Classification": "mapred-site"
            },
            {
                "Properties": {
                    "hive.join.emit.interval": "1000",
                    "hive.merge.mapfiles": "true"
                },
                "Classification": "hive-site"
            }
        ]
    }
}
```

Example 按创建日期列出集群

要检索特定日期范围内创建的集群，请使用 `list-clusters` 命令与 `--created-after` 和 `--created-before` 参数。

以下命令将列出在 2019 年 10 月 9 日和 2019 年 10 月 12 日之间创建的所有集群。

```
aws emr list-clusters --created-after 2019-10-09T00:12:00 --created-
before 2019-10-12T00:12:00
```

Example 按状态列出集群

要按状态列出集群，使用 `list-clusters` 命令与 `--cluster-states` 参数。有效集群状态包括：STARTING、BOOTSTRAPPING、RUNNING、WAITING、TERMINATING、TERMINATED 和 TERMINATED_WITH_ERRORS。

```
aws emr list-clusters --cluster-states TERMINATED
```

您还可使用以下快捷方式参数来列出处于指定状态的所有集群：

- `--active` 筛选处于 STARTING、BOOTSTRAPPING、RUNNING、WAITING 或 TERMINATING 状态的集群。
- `--terminated` 筛选处于 TERMINATED 状态的集群。
- `--failed` 参数筛选处于 TERMINATED_WITH_ERRORS 状态的集群。

以下命令返回相同的结果。

```
aws emr list-clusters --cluster-states TERMINATED
```

```
aws emr list-clusters --terminated
```

有关集群状态的更多信息，请参阅[了解集群的生命周期]。

增强型步骤调试

如果 Amazon EMR 步骤失败，并且您通过 5.x 版或更高版本的 AMI 使用步骤 API 操作提交了工作，则 Amazon EMR 可以在某些情况下通过 API 确定并返回步骤失败的根本原因，以及相关日志文件的名称和部分应用程序堆栈跟踪。例如，可以确定以下失败：

- 常见 Hadoop 错误，例如输出目录已经存在、输入内容不存在，或应用程序用尽内存。
- Java 错误，例如用不兼容的 Java 版本编译了应用程序，或用找不到的主类运行。
- 访问在 Amazon S3 中存储的对象时发生问题。

该信息可使用 DescribeStep 和 ListSteps API 操作获得。StepSummary 的 FailureDetails 字段由这些操作返回。要访问 FailureDetails 信息，请使用 Amazon CLI、控制台或 Amazon SDK。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

Amazon EMR 新控制台不提供步骤调试。但是，您可以通过以下步骤查看集群终止详细信息。

使用新控制台查看失败详细信息

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要查看的集群。
3. 记下显示在集群详细信息页面上 Summary（摘要）部分中的 Status（状态）值。如果状态为 Terminated with errors（因错误而终止），请将鼠标悬停在文本上以查看集群失败详细信息。

Old console

使用旧控制台查看失败详细信息

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择 Cluster List (集群列表) 并选择一个集群。
3. 选择每个步骤旁边的箭头图标可查看更多详细信息。如果步骤失败而且 Amazon EMR 可以确定根本原因，则您会看到失败详细信息。

CLI

使用 Amazon CLI 查看失败详细信息

- 要使用 Amazon CLI 获取步骤的失败详细信息，请使用 `describe-step` 命令。

```
aws emr describe-step --cluster-id j-1K48XXXXXHCB --step-id s-3QM0XXXXXM1W
```

输出将类似如下：

```
{
  "Step": {
    "Status": {
      "FailureDetails": {
        "LogFile": "s3://myBucket/logs/j-1K48XXXXXHCB/steps/s-3QM0XXXXXM1W/
stderr.gz",
        "Message": "org.apache.hadoop.mapred.FileAlreadyExistsException: Output
 directory s3://myBucket/logs/beta already exists",
        "Reason": "Output directory already exists."
      },
      "Timeline": {
        "EndDateTime": 1469034209.143,
        "CreationDateTime": 1469033847.105,
        "StartDateTime": 1469034202.881
      },
      "State": "FAILED",
      "StateChangeReason": {}
    },
    "Config": {
      "Args": [
        "wordcount",
        "s3://myBucket/input/input.txt",
        "s3://myBucket/logs/beta"
      ],
      "Jar": "s3://myBucket/jars/hadoop-mapreduce-examples-2.7.2-amzn-1.jar",
      "Properties": {}
    },
    "Id": "s-3QM0XXXXXM1W",
    "ActionOnFailure": "CONTINUE",
    "Name": "ExampleJob"
  }
}
```

查看应用程序历史记录。

您可以在控制台中使用集群详细信息页面查看 Spark 历史记录服务器和 YARN 时间线服务应用程序的详细信息。Amazon EMR 应用程序历史记录可让您更轻松地对活动的任务和任务历史记录进行故障排除和分析。

New console

Applications（应用程序）选项卡的 Application user interfaces（应用程序用户界面）部分提供多个查看选项，具体取决于集群状态和您在集群上安装的应用程序。

- 集群外访问持久性应用程序用户界面：从 Amazon EMR 版本 5.25.0 开始，Spark UI 和 Spark 历史记录服务器可以使用持久性应用程序用户界面链接。对于 Amazon EMR 版本 5.30.1 及更高版本，Tez UI 和 YARN 时间线服务器也具有持久性应用程序用户界面。YARN 时间线服务器和 Tez UI 是开源应用程序，可提供适用于活动的和已终止的集群的指标。Spark 用户界面提供有关计划程序阶段和任务、RDD 大小和内存使用情况的详细信息、环境信息以及有关正在运行的执行程序的信息。持久性应用程序 UI 是在集群外运行的，因此集群信息和日志在应用程序终止后 30 天内可用。与集群上的应用程序用户界面不同，持久性应用程序 UI 不要求您通过 SSH 连接设置 Web 代理。
- 集群上的应用程序用户界面 – 可以在集群上运行的应用程序历史记录用户界面各种各样。集群上的用户界面托管在主节点上，并要求您设置到 Web 服务器的 SSH 连接。应用程序终止后，集群上的应用程序用户界面将应用程序历史记录保留一周。有关设置 SSH 隧道的更多信息和说明，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

  除了 Spark 历史记录服务器、YARN 时间线服务器和 Hive 应用程序之外，只能在集群运行时查看集群上的应用程序历史记录。

Old console

Application user interfaces (应用程序用户界面) 选项卡提供多个查看选项：

- 集群外访问持久性应用程序用户界面 – 从 Amazon EMR 版本 5.25.0 开始，Spark 可以使用持久性应用程序用户界面链接。对于 Amazon EMR 版本 5.30.1 及更高版本，Tez UI 和 YARN 时间线服务器也具有持久性应用程序用户界面。YARN 时间线服务器和 Tez UI 是开源应用程序，可提供适用于活动的和已终止的集群的指标。Spark 用户界面提供有关计划程序阶段和任务、RDD 大小和内存使用情况的详细信息、环境信息以及有关正在运行的执行程序的信息。持久性应用程序 UI 是在集群外运行的，因此集群信息和日志在应用程序终止后 30 天内可用。与集群上的应用程序用户界面不同，持久性应用程序 UI 不要求您通过 SSH 连接设置 Web 代理。
- 集群上的应用程序用户界面 – 可以在集群上运行的应用程序历史记录用户界面各种各样。集群上的用户界面托管在主节点上，并要求您设置到 Web 服务器的 SSH 连接。应用程序终止后，集群上的应用程序用户界面将应用程序历史记录保留一周。有关设置 SSH 隧道的更多信息和说明，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

除了 Spark 历史记录服务器、YARN 时间线服务器和 Hive 应用程序之外，只能在集群运行时查看集群上的应用程序历史记录。

- 高级别应用程序历史记录：使用 Amazon EMR 版本 5.8.0 到 5.36.0 以及 6.8.0 或更低版本的 6.x 发行版，您可以在 Amazon EMR 旧控制台中查看应用程序历史记录的摘要，包括阶段任务和执行程序的关键指标。自 2023 年 1 月 23 日起，Amazon EMR 将停止所有版本的高级别应用程序历史记录。如果您使用 Amazon EMR 版本 5.25.0 或更高版本，我们建议您改用持久性应用程序用户界面。

查看持久性应用程序用户界面

从 Amazon EMR 版本 5.25.0 开始，您可以使用集群的 Summary (摘要) 页面或控制台中的 Application user interfaces (应用程序用户界面) 选项卡连接到在集群外托管的持久性 Spark 历史记录服务器应用程序的详细信息。从 Amazon EMR 版本 5.30.1 开始，提供了 Tez UI 和 YARN 时间线服务器持久性应用程序界面。对持久性应用程序历史记录的一键式链接访问提供了以下好处：

- 您可以快速分析活动的作业和作业历史记录并进行故障排除，而无需通过 SSH 连接来设置 Web 代理。
- 您可以访问处于活动状态和终止状态的集群的应用程序历史记录和相关日志文件。日志在应用程序结束后的 30 天内均可用。

在 Amazon EMR 5.30.1 或 6.x 旧控制台中，在集群的 Application user interfaces（应用程序用户界面）选项卡或集群 Summary（摘要）页面上，选择 YARN timeline server（YARN 时间线服务器）、Tez UI 或 Spark history server（Spark 历史记录服务器）链接。

应用程序 UI 将在新的浏览器选项卡中打开。有关更多信息，请参阅监控和检测。

您可以通过 Spark 历史记录服务器、YARN 时间线服务器和 Tez UI 上的链接来查看 YARN 容器日志。

> **Note**
>
> 要从 Spark 历史记录服务器、YARN 时间线服务器和 Tez UI 访问 YARN 容器日志，您必须为集群启用 Amazon S3 日志记录。如果未启用日志记录，则指向 YARN 容器日志的链接将不起作用。

日志收集

要启用一键式访问持久性应用程序用户界面，Amazon EMR 需要收集两种类型的日志：

- 应用程序事件日志 会收集到 EMR 系统存储桶中。通过使用 Amazon S3 托管密钥的服务器端加密 (SSE-S3) 对事件日志进行静态加密。如果您对集群使用私有子网，请确保在私有子网的 Amazon S3 策略的资源列表中包含 “arn:aws:s3:::prod.MyRegion.appinfo.src/*”。有关更多信息，请参阅私有子网的最小 Amazon S3 策略。
- YARN 容量日志收集到您拥有的 Amazon S3 存储桶中。您必须为集群启用日志记录才能访问 YARN 容器日志。有关更多信息，请参阅配置集群日志记录和调试。

如果您出于隐私原因需要禁用此功能，则可在创建集群时使用引导脚本来停止守护程序，如以下示例所示。

```
aws emr create-cluster --name "Stop Application UI Support" --release-label emr-5.36.1
 \
--applications Name=Hadoop Name=Spark --ec2-attributes KeyName=<myEMRKeyPairName> \
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m3.xlarge
 InstanceGroupType=CORE,InstanceCount=1,InstanceType=m3.xlarge
 InstanceGroupType=TASK,InstanceCount=1,InstanceType=m3.xlarge \
--use-default-roles --bootstrap-actions Path=s3://region.elasticmapreduce/bootstrap-
actions/run-if,Args=["instance.isMaster=true","echo Stop Application UI | sudo tee /
etc/apppusher/run-apppusher; sudo systemctl stop apppusher || exit 0"]
```

运行此引导启动脚本后，Amazon EMR 不会将任何 Spark 历史记录服务器或 YARN 时间线服务器事件日志收集到 EMR 系统存储桶中。Application user interfaces (应用程序用户界面) 选项卡上没有可用的应用程序历史记录信息，并且您将不再能够从控制台访问所有应用程序用户界面。

大型 Spark 事件日志文件

在某些情况下，长时间运行的 Spark 作业（例如 Spark 流式处理）和大型作业（例如 Spark SQL 查询）可能会生成大型事件日志。使用大型事件日志，您可以快速耗尽计算实例上的磁盘空间，并在加载持久 UI 时遇到 `OutOfMemory` 错误。要避免这些问题，建议您启用 Spark 事件日志滚动和压缩功能。此功能在 Amazon EMR 版本 emr-6.1.0 及更高版本上可用。有关滚动和压缩的更多详细信息，请参阅 Spark 文档中的对滚动事件日志文件应用压缩。

要激活 Spark 事件日志滚动和压缩功能，请启用以下 Spark 配置设置。

- `spark.eventLog.rolling.enabled` – 根据大小启用事件日志滚动。默认情况下，此设置处于禁用状态。

- spark.eventLog.rolling.maxFileSize – 激活滚动时，指定事件日志文件在滚动之前的最大大小。默认值为 128 MB。
- spark.history.fs.eventLog.rolling.maxFilesToRetain – 指定要保留的非压缩事件日志文件的最大数量。默认情况下，会保留所有事件日志文件。设置为较小的数字以压缩较旧的事件日志。最小值为 1。

请注意，压缩会尝试排除具有过时事件日志文件的事件，如下所示。如果它确实丢弃了事件，则在 Spark 历史记录服务器 UI 上将看不到它们。

- 已完成作业的事件以及相关阶段或任务事件。
- 终止执行程序的事件。
- 已完成 SQL 查询的事件，以及相关作业、阶段和任务事件。

启动启用滚动和压缩的集群

1. 使用以下配置创建 spark-configuration.json 文件。

```
[
   {
     "Classification": "spark-defaults",
        "Properties": {
           "spark.eventLog.rolling.enabled": true,
           "spark.history.fs.eventLog.rolling.maxFilesToRetain": 1
        }
   }
]
```

2. 使用 Spark 滚动压缩配置创建集群，如下所示。

```
aws emr create-cluster \
--release-label emr-6.6.0 \
--instance-type m4.large \
--instance-count 2 \
--use-default-roles \
--configurations file://spark-configuration.json
```

注意事项和限制

一键式访问持久性应用程序用户界面当前具有以下限制。

- 当应用程序详细信息显示在 Spark 历史记录服务器 UI 上时，至少会有两分钟的延迟。
- 仅当应用程序的事件日志目录位于 HDFS 中时，此功能才起作用。默认情况下，Amazon EMR 将事件日志存储在 HDFS 的目录中。如果您将默认目录更改为其它文件系统（例如 Amazon S3），则此功能将不起作用。
- 此功能目前不适用于具有多个主节点的 EMR 集群或与 Amazon Lake Formation 集成的 EMR 集群。
- 要启用一键式访问持久性应用程序用户界面，您必须有权对 Amazon EMR 执行 `DescribeCluster` 操作。如果您拒绝 IAM 委托人对此操作的权限，则传播权限更改所需的时间大约为 5 分钟。
- 如果在正在运行的集群中重新配置应用程序，则将无法通过应用程序 UI 获取应用程序历史记录。
- 对于每个 Amazon Web Services 账户，活动应用程序 UI 的默认限制为 200。
- 您可以在以下区域通过该控制台访问应用程序 UI：美国东部（弗吉尼亚州北部）、美国西部（北加利福尼亚）、加拿大（中部）、欧洲地区（法兰克福、爱尔兰、伦敦、巴黎、斯德哥尔摩）、亚太地区（孟买、首尔、新加坡、悉尼和东京）、南美洲（圣保罗）、由光环新网运营的中国（北京）和由西云数据运营的中国（宁夏）。

查看高级别应用程序历史记录

> **Note**
>
> 建议您使用持久性应用程序界面来改善用户体验，应用程序历史记录可保留长达 30 天。本页所介绍的高级别应用程序历史记录在 Amazon EMR 新控制台（https://console.aws.amazon.com/emr）中不可用。有关更多信息，请参阅查看持久性应用程序用户界面。

对于 Amazon EMR 发行版 5.8.0 到 5.36.0 以及 6.8.0 或更低版本的 6.x 发行版，您可以从 Amazon EMR 旧控制台的 Application user interfaces（应用程序用户界面）选项卡查看高级别应用程序历史记录。在应用程序完成后，Amazon EMR Application user interface（应用程序用户界面）会将应用程序历史记录摘要保留七天。

注意事项和限制

当您使用旧版 Amazon EMR 控制台中的应用程序用户界面选项卡时，请注意以下限制。

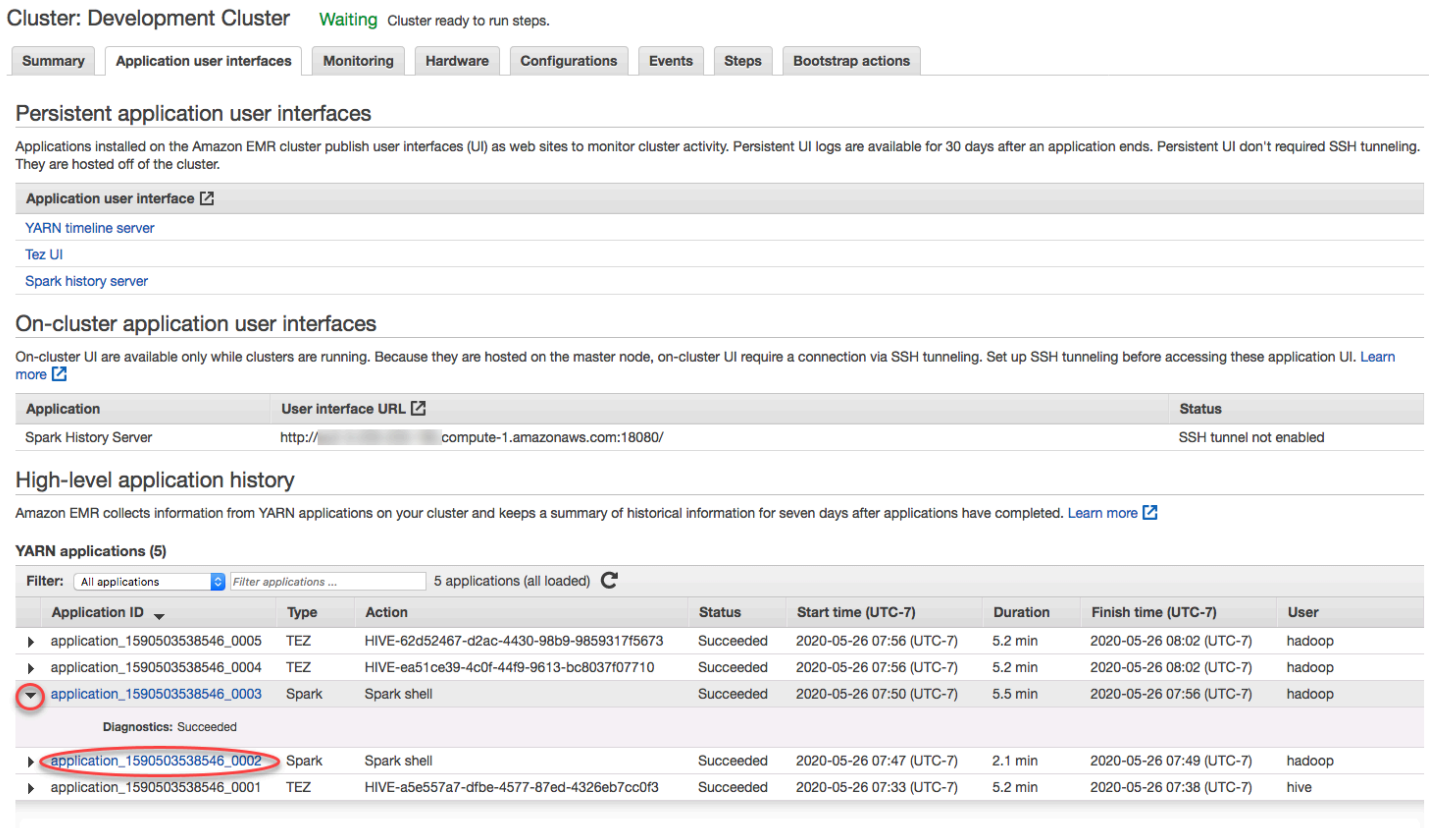
- 在使用 Amazon EMR 5.8.0 到 5.36.0 以及 6.8.0 或更低版本的 6.x 发行版时，您只能访问高级别应用程序历史记录功能。自 2023 年 1 月 23 日起，Amazon EMR 将停止所有版本的高级别应用程序历史记录。如果您使用 Amazon EMR 版本 5.25.0 或更高版本，我们建议您改用持久性应用程序用户界面。
- 高级别应用程序历史记录功能不支持 Spark Streaming 应用程序。
- 目前，具有多个主节点的 Amazon EMR 集群或与 Amazon Lake Formation 集成的 EMR 集群不支持一键式访问持久性的应用程序用户界面。

示例：查看高级别应用程序历史记录

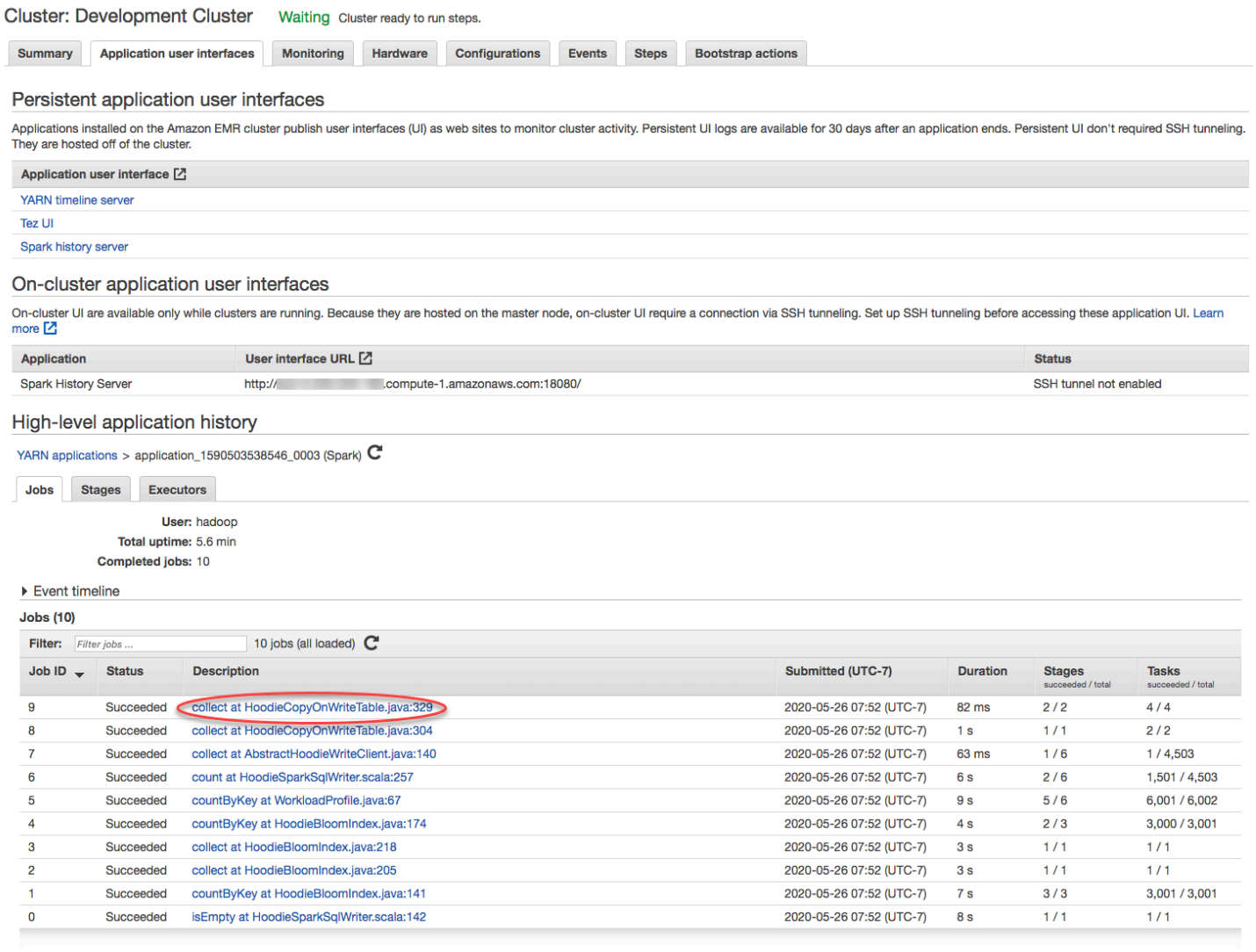
以下序列演示了如何使用旧控制台集群详细信息页面上的 Application user interfaces（应用程序用户界面）选项卡，通过 Spark 或 YARN 应用程序深入了解任务详细信息。

要查看集群详细信息，请从 Clusters (集群) 列表中选择集群 Name (名称)。要查看有关 YARN 容器日志的信息，您必须为集群启用日志记录。有关更多信息，请参阅配置集群日志记录和调试。对于 Spark 应用程序历史记录，摘要表中提供的信息只是通过 Spark 历史记录服务器 UI 提供的信息的一个子集。

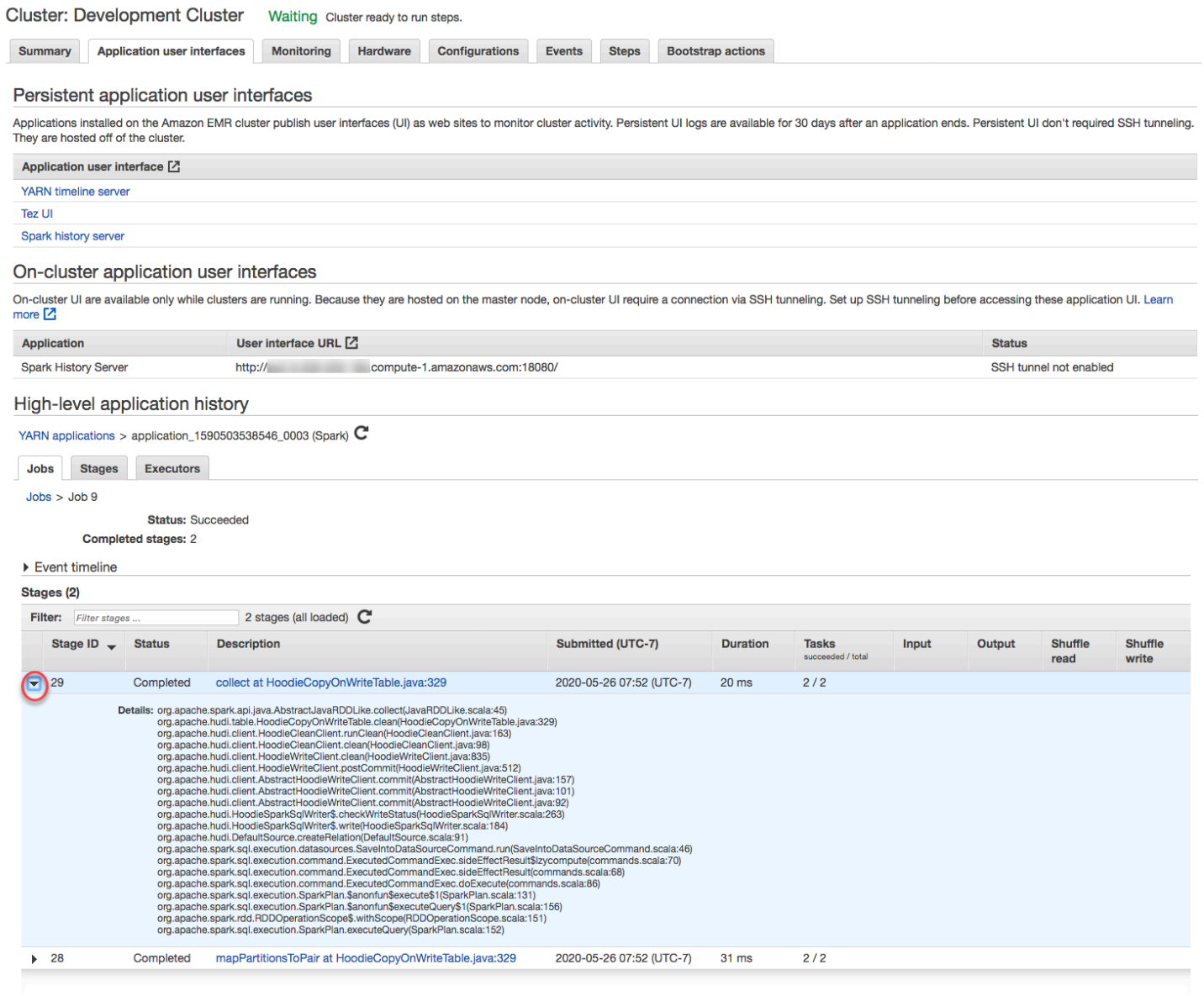
在 High-level application history (高级别应用程序历史记录) 下的 Application user interfaces (应用程序用户界面) 选项卡下，您可以展开一行以显示 Spark 应用程序的诊断摘要，或选择 Application ID (应用程序 ID) 链接以查看有关不同应用程序的详细信息。



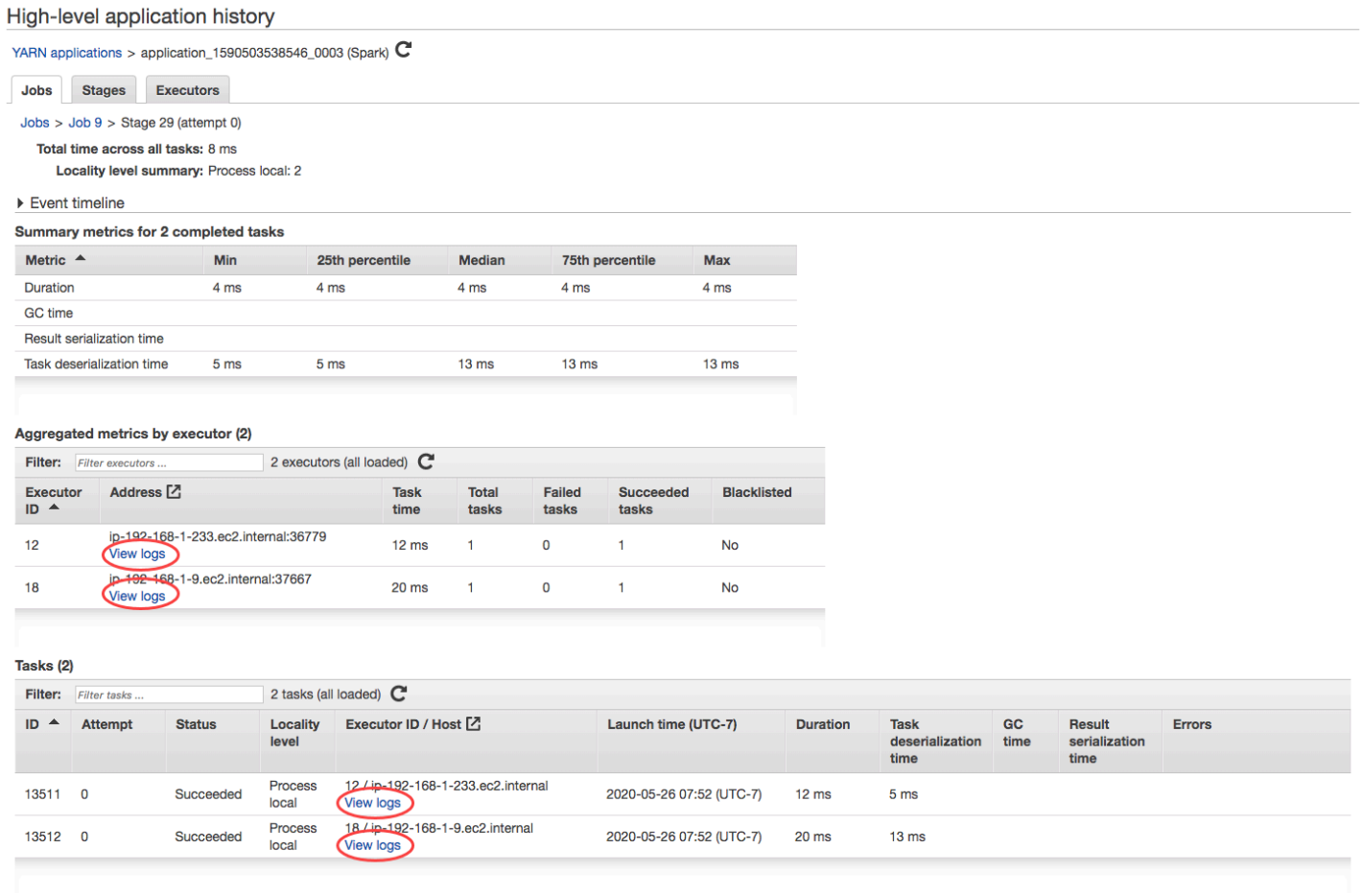
当您选择 Application ID (应用程序 ID) 链接时，UI 会更改以显示该应用程序的 YARN application (YARN 应用程序) 详细信息。在 YARN application (YARN 应用程序) 详细信息的 Jobs (任务) 选项卡中，您可以选择任务的 Description (描述) 链接以显示该任务的详细信息。



在任务详细信息页面上，您可以展开有关各个任务阶段的信息，然后选择 Description (描述) 链接以查看阶段详细信息。



在阶段详细信息页面上，您可以查看阶段任务和执行程序的关键指标。您还可以使用 View logs (查看日志) 链接查看任务和执行程序日志。



查看日志文件

Amazon EMR 和 Hadoop 都可以生成日志文件，报告集群上的状态。默认情况下，这些会写入 `/mnt/var/log/` 目录中的主节点。根据您在启动时如何配置集群，这些日志还可能归档到 Amazon S3，并可通过图形调试工具进行查看。

有多种类型的日志写入主节点。Amazon EMR 会写入步骤、引导操作和实例状态日志。Apache Hadoop 写入的日志会报告作业、任务和任务尝试的处理情况。Hadoop 还会记录守护程序的日志。有关 Hadoop 写入的日志的更多信息，请转到 http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html。

查看主节点上的日志文件

下表列出了主节点上可能显示的一些日志文件。

位置	描述
/emr/instance-controller/log/bootstrap-actions	引导操作处理期间写入的日志。
/mnt/var/log/hadoop-state-pusher	Hadoop 状态推送进程写入的日志。
/emr/instance-controller/log	实例控制器日志。
/emr/instance-state	实例状态日志。这些日志中包含有关 CPU、内存状态和节点的垃圾收集器线程的信息。
/emr/service-nanny	nanny 服务流程写入的日志。
/mnt/var/log/*application*	特定于应用程序 (如 Hadoop、Spark 或 Hive) 的日志。
/mnt/var/log/hadoop/steps/*N*	步骤日志，其中包含有关步骤处理的信息。值 *N* 指示 Amazon EMR 分配的 stepId。例如，一个集群有两个步骤：`s-1234ABCDEFGH` 和 `s-5678IJKLMNOP` 。第一步位于 `/mnt/var/log/hadoop/steps/s-1234ABCDEFGH/` 中，第二步位于 `/mnt/var/log/hadoop/steps/s-5678IJKLMNOP/` 中。 Amazon EMR 写入的步骤日志如下。 • 控制器 – 有关步骤处理的信息。如果您的步骤在加载时失败，您可以在此日志中找到堆栈跟踪。 • syslog – 描述步骤中 Hadoop 任务的执行情况。 • stderr – Hadoop 处理步骤时的标准错误通道。 • stdout – Hadoop 处理步骤时的标准输出通道。

使用 Amazon CLI 查看主节点上的日志文件。

1. 如 使用 SSH 连接到主节点 中所述使用 SSH 连接主节点。
2. 导航到您希望查看的、包含日志文件信息的目录。上表提供了一系列可用的、用户也可以搜索到的日志文件类型。以下示例演示用于导航到 ID 为 s-1234ABCDEFGH 的步骤日志的命令。

```
cd /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/
```

3. 使用您选择的文件查看器来查看日志文件。以下示例使用 Linux less 命令查看 controller 日志文件。

```
less controller
```

查看归档到 Amazon S3 的日志文件

默认情况下，使用控制台启动的 Amazon EMR 集群会自动将日志文件归档到 Amazon S3。您可以指定自己的日志路径，也可以允许控制台自动为您生成日志路径。对于使用 CLI 或 API 启动的集群，您必须手动配置 Amazon S3 日志归档。

当 Amazon EMR 配置为在 Amazon S3 中归档日志文件时，它会将文件存储在您指定的 S3 位置中，即 /*cluster-id*/ 文件夹中，其中 *cluster-id* 是集群 ID。

下表列出了 Amazon S3 上可能显示的一些日志文件。

位置	描述
/*cluster-id* /node/	节点日志，包括引导操作、实例状态和此节点的应用程序日志。每个节点的日志会存储在标有该节点的 EC2 实例标识符的文件夹中。
/*cluster-id* /node/*instance-id* /*application*	由每个应用程序或与应用程序关联的守护进程创建的日志。例如，Hive 服务器日志位于 *cluster-id* /node/*instance-id* /hive/hive-server.log 中。
/*cluster-id* /steps/*step-id*/	步骤日志，其中包含有关步骤处理的信息。*step-id* 的值表示 Amazon EMR 分配的步骤 ID。例如，一个集群有两个步骤：s-1234ABC

位置	描述
	DEFGH 和 s-5678IJKLMNOP 。第一步位于 /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/ 中，第二步位于 /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/ 中。 Amazon EMR 写入的步骤日志如下。 • 控制器 – 有关步骤处理的信息。如果您的步骤在加载时失败，您可以在此日志中找到堆栈跟踪。 • syslog – 描述步骤中 Hadoop 任务的执行情况。 • stderr – Hadoop 处理步骤时的标准错误通道。 • stdout – Hadoop 处理步骤时的标准输出通道。
/*cluster-id* /containers	应用程序容器日志。每个 YARN 应用程序的日志均存储在这些位置。
/*cluster-id* /hadoop-mapreduce/	包含有关 MapReduce 作业的配置详细信息和作业历史记录的日志。

使用 Amazon S3 控制台查看存档到 Amazon S3 的日志文件

1. 登录到 Amazon Web Services Management Console，然后通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 打开您在配置集群时指定的 S3 存储桶，以便将日志文件归档在 Amazon S3 中。
3. 导航到包含待显示信息的日志文件。上表提供了一系列可用的、用户也可以搜索到的日志文件类型。
4. 下载日志文件对象以进行查看。有关说明，请参阅下载对象。

查看调试工具中的日志文件

Amazon EMR 不会自动启用调试工具。您必须在启动集群时对此进行配置。请注意，Amazon EMR 新控制台不提供调试工具。

使用旧控制台查看集群日志

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 在 Cluster List (集群列表) 页面中，选择要查看的集群旁边的详细信息图标。

   这会打开 Cluster Details (集群详细信息) 页面。在 Steps (步骤) 部分中，每个步骤右侧的链接显示可用于该步骤的各种类型的日志。这些日志由 Amazon EMR 生成。
3. 要查看与给定步骤关联的 Hadoop 任务列表，请选择该步骤右侧的 View Jobs (查看任务) 链接。
4. 要查看与给定任务关联的 Hadoop 任务列表，请选择该任务右侧的 View Tasks (查看任务) 链接。
5. 要查看给定任务在尝试完成时运行的尝试列表，请选择任务右侧的 View Attempts (查看尝试) 链接。
6. 要查看任务尝试生成的日志，请选择任务尝试右侧的 stderr、stdout 和 syslog 链接。

在 Amazon EMR 上载日志文件到 Amazon S3 上的存储桶后，调试工具会显示日志文件的链接。因为日志文件每 5 分钟会向 Amazon S3 上载一次，所以，日志文件要花费几分钟才能在步骤完成后完成上载。

Amazon EMR 定期更新调试工具中的 Hadoop 任务、任务和任务尝试的状态。您可以单击调试窗格中的 Refresh List (刷新列表)，获取这些项目的最新状态。

查看 Amazon EC2 中的集群实例

为了帮助您管理资源，Amazon EC2 可让您将元数据以标签的形式分配给资源。每个 Amazon EC2 标签都包含一个密钥和一个值。标签可让您按各种标准 (例如用途、所有者或环境) 对 Amazon EC2 资源进行分类。

您可以根据标签搜索和筛选资源。您通过您的 Amazon 账户向资源分配的标签仅限您使用。共享该资源的其它账户无法查看您的标签。

Amazon EMR 会自动使用键值对标记其启动的每个 EC2 实例。这些键可标识该实例所属的集群和实例组。这可以轻松地筛选要显示的 EC2 实例，例如仅显示属于特定集群的实例，或者显示目前在任务实例组中运行的全部实例。如果您同时运行多个集群，或者管理大量的 EC2 实例，这将特别有用。

这些是 Amazon EMR 分配的预定义键值对：

键	Value	值定义
aws:elasticmapreduce:job-flow-id	*job-flow-identifier*	为其预置实例的集群的 ID。它以格式 `j-XXXXXXXXXXXX` 显示，最长可为 256 个字符。
aws:elasticmapreduce:instance-group-role	*group-role*	实例组的类型，输入为以下值之一：`master`、`core`、或 `task`。

您可以对 Amazon EMR 添加的标签进行查看和筛选。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的使用标签。因为 Amazon EMR 设置的标签都属于系统标签，无法编辑也无法删除，所以有关显示和筛选标签的部分与此内容是最相关的。

> **Note**
>
> Amazon EMR 在 EC2 实例的状态更新为正在运行时为其添加标签。如果 EC2 实例的预置时间和将其状态设置为正在运行的时间之间发生延迟，那么 Amazon EMR 设置的标签会在实例启动后立即显示。如果您没有看到这些标签，请等待几分钟，并刷新一下视图。

CloudWatch 事件和指标

使用事件和指标，以跟踪 Amazon EMR 集群的活动和运行状况。事件对于监控集群中的特定情况（如当集群状态从“starting (正在启动)”更改为“running (正在运行)”时）非常有用。指标对于监控特定的值（如 HDFS 在集群中使用的可用磁盘空间比例）非常有用。

有关 CloudWatch Events 的更多信息，请参阅 Amazon CloudWatch Events 用户指南。有关 CloudWatch 指标的更多信息，请参阅《Amazon CloudWatch 用户指南》中的使用 Amazon CloudWatch 指标和创建 Amazon CloudWatch 警报。

主题

- 使用 CloudWatch 指标监控 Amazon EMR 指标
- 通过 CloudWatch 监控 Amazon EMR 事件

- 响应 CloudWatch 事件

使用 CloudWatch 指标监控 Amazon EMR 指标

指标每五分钟更新一次，然后自动收集并推送到每个 Amazon EMR 集群的 CloudWatch。此时间间隔是不可配置的。CloudWatch 中报告的 Amazon EMR 指标无需任何费用。这些五分钟的数据点指标会归档 63 天，之后数据将被丢弃。

如何使用 Amazon EMR 指标？

下表显示了 Amazon EMR 报告指标的常见用途。这些是入门建议，并不全面。有关由 Amazon EMR 报告的指标的完整列表，请参阅Amazon EMR 在 CloudWatch 中的报告指标。

如何？	相关指标
跟踪我的集群进展	查看 RunningMapTasks 、Remaining MapTasks 、RunningReduceTasks 和 RemainingReduceTasks 指标。
检测处于空闲状态的集群	IsIdle 指标可跟踪某个集群 (非当前运行任务) 是否正在实时运行。您可以设置当集群闲置达到给定时长 (例如 30 分钟) 时便可引发的警报。
检测节点何时用尽存储空间	MRUnhealthyNodes 指标跟踪一个或多个核心节点或任务节点何时用尽本地磁盘存储空间并转换到 UNHEALTHY YARN 状态。例如，核心节点或任务节点在磁盘中的运行空间不足，并将无法运行任务。
检测集群何时用尽存储空间	HDFSUtilization 指标监控集群的组合 HDFS 容量，可能需要调整集群大小以添加更多核心节点。例如，HDFS 利用率较高，这可能会影响任务和集群运行状况。
检测集群何时以更少容量运行	MRLostNodes 指标跟踪一个或多个核心节点或任务节点何时无法与主节点通信。例如，主节点无法访问核心节点或任务节点。

有关更多信息，请参阅 集群因 NO_SLAVE_LEFT 和核心节点 FAILED_BY_MASTER 而终止 和 AWSSupport-AnalyzeEMRLogs。

访问 Amazon EMR 的 CloudWatch 指标

您可以使用 Amazon EMR 控制台或 CloudWatch 控制台查看 Amazon EMR 向 CloudWatch 报告的指标。您还可以使用 CloudWatch CLI 命令 `mon-get-stats` 或 CloudWatch `GetMetricStatistics` API 检索指标。有关使用 CloudWatch 查看或检索 Amazon EMR 指标的更多信息，请参阅 《Amazon CloudWatch 用户指南》。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台中查看指标

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要查看指标的集群。集群详细信息页面随即打开。
3. 在集群详细信息页面上选择 Monitoring（监控）选项卡。选择 Cluster Status（集群状态）、Node Status（节点状态）或 Inputs and outputs（输入和输出）选项中的任何一个，以加载有关集群进度和运行状况的报告。
4. 选择要查看的指标之后，您可以放大每个图表。要筛选图表的时间范围，请选择预填选项或选择 Custom（自定义）。

Old console

使用旧控制台查看指标

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/。
2. 要查看集群的指标，请选择集群以显示 Summary (摘要) 窗格。

3. 选择监控以查看有关该集群的信息。选择 Cluster Status（集群状态）、Map/Reduce（映射/化简）、Node Status（节点状态）或 IO 选项卡中的任何一个，从而加载有关集群进度和运行状况的报告。
4. 选择要查看的指标之后，您可以选取图表大小。编辑 Start (开始) 和 End (结束) 字段以按特定时间范围筛选指标。

Amazon EMR 在 CloudWatch 中的报告指标

下表列出了 Amazon EMR 在控制台中报告并推送到 CloudWatch 的指标。

Amazon EMR 指标

Amazon EMR 会将多个指标的数据发送到 CloudWatch。所有 Amazon EMR 集群会以五分钟的间隔自动发送指标。指标会存档两周。两周后，数据会被丢弃。

`AWS/ElasticMapReduce` 命名空间包括以下指标。

> **Note**
>
> Amazon EMR 从集群中提取指标。如果无法连接到集群，则在此集群再次变成可用状态之前，EMR 将不会报告任何指标。

以下指标适用于 Hadoop 2.x 版本上运行的集群。

指标	描述
集群状态	
处于空闲状态	指示集群不再执行任务，但仍处于活动状态并会产生费用。如果没有任何任务和任务处于运行状态，则此指标设置为 1；否则设置为 0。系统每隔五分钟检查一次该值，值为 1 仅表示在检查时集群处于空闲状态，并不表示它整个五分钟内都处于空闲状态。为避免误报，当多次连续 5 分钟检查获得的值均为 1 时，您应提出警报。例如，当该值在三十分钟或更长时间内都为 1 时，您应提出警报。 使用案例：监控集群性能

指标	描述
	单位：布尔值
ContainerAllocated	ResourceManager 分配的资源容器数。 使用案例：监控集群进度 单位：计数
ContainerReserved	预留的容器数。 使用案例：监控集群进度 单位：计数
ContainerPending	队列中尚未分配的容器数。 使用案例：监控集群进度 单位：计数
ContainerPendingRatio	待处理的容器与已分配的容器的比值 (ContainerPendingRatio = ContainerPending/ContainerAllocated)。如果 ContainerAllocated = 0，则 ContainerPendingRatio = ContainerPending。值 ContainerPendingRatio 表示数字，而不是百分比。此值对基于容器分配行为扩展集群资源很有用。 单位：计数
AppsCompleted	提交给 YARN 并且已完成的应用程序数。 使用案例：监控集群进度 单位：计数
AppsFailed	提交给 YARN 并且未能完成的应用程序数。 使用案例：监控集群进度，监控集群运行状况 单位：计数

指标	描述
AppsKilled	提交给 YARN 并且已终止的应用程序数。 使用案例：监控集群进度，监控集群运行状况 单位：计数
AppsPending	提交给 YARN 并且处于挂起状态的应用程序数。 使用案例：监控集群进度 单位：计数
AppsRunning	提交给 YARN 并且正在运行的应用程序数。 使用案例：监控集群进度 单位：计数
AppsSubmitted	提交给 YARN 的应用程序数。 使用案例：监控集群进度 单位：计数
节点状态	
CoreNodesRunning	处于运行状态的核心节点的数量。仅当对应的实例组存在时，才会报告此指标的数据点。 使用案例：监控集群运行状况 单位：计数
CoreNodesPending	等待分配的核心节点的数量。请求的所有核心节点可能不会立即可用；此指标报告挂起的请求。仅当对应的实例组存在时，才会报告此指标的数据点。 使用案例：监控集群运行状况 单位：计数

指标	描述
LiveDataNodes	从 Hadoop 接收任务的数据节点的百分率。 使用案例：监控集群运行状况 单位：百分比
MRTotalNodes	当前可供 MapReduce 作业使用的节点数。等效于 YARN 指标 `mapred.resourcemanager.TotalNodes` 。 使用案例：监控集群进度 单位：计数
MRActiveNodes	当前正在运行 MapReduce 任务或作业的节点数。等效于 YARN 指标 `mapred.resourcemanager.NoOfActiveNodes` 。 使用案例：监控集群进度 单位：计数
MRLostNodes	分配给 MapReduce 并且已标记为 LOST 状态的节点数。等效于 YARN 指标 `mapred.resourcemanager.NoOfLostNodes` 。 使用案例：监控集群运行状况，监控集群进度 单位：计数
MRUnhealthyNodes	当前可供 MapReduce 作业使用的标记为 UNHEALTHY 状态的节点数。等效于 YARN 指标 `mapred.resourcemanager.NoOfUnhealthyNodes` 。 使用案例：监控集群进度 单位：计数

指标	描述
MRDecommissionedNodes	分配给 MapReduce 应用程序并且已标记为 DECOMMISSIONED 状态的节点数。等效于 YARN 指标 `mapred.resourcemanager.NoOfDecommissionedNodes` 。 使用案例：监控集群运行状况，监控集群进度 单位：计数
MRRebootedNodes	可供 MapReduce 使用的已重新启动并标记为 REBOOTED 状态的节点数。等效于 YARN 指标 `mapred.resourcemanager.NoOfRebootedNodes` 。 使用案例：监控集群运行状况，监控集群进度 单位：计数
MultiMasterInstanceGroupNodesRunning	正在运行的主节点的数量。 使用案例：监控主节点故障和替换 单位：计数
MultiMasterInstanceGroupNodesRunningPercentage	正在运行的主节点超过所请求的主节点实例计数的百分比。 使用案例：监控主节点故障和替换 单位：百分比
MultiMasterInstanceGroupNodesRequested	请求的主节点数。 使用案例：监控主节点故障和替换 单位：计数
IO	

指标	描述
S3BytesWritten	写入 Amazon S3 的字节数。此指标仅聚合 MapReduce 任务，并不适用于 Amazon EMR 上的其它工作负载。 使用案例：分析集群性能，监控集群进度 单位：计数
S3BytesRead	从 Amazon S3 读取的字节数。此指标仅聚合 MapReduce 任务，并不适用于 Amazon EMR 上的其它工作负载。 使用案例：分析集群性能，监控集群进度 单位：计数
HDFSUtilization	当前使用的 HDFS 存储的百分率。 使用案例：分析集群性能 单位：百分比
HDFSBytesRead	从 HDFS 读取的字节数。此指标仅聚合 MapReduce 任务，并不适用于 Amazon EMR 上的其它工作负载。 使用案例：分析集群性能，监控集群进度 单位：计数
HDFSBytesWritten	写入 HDFS 的字节数。此指标仅聚合 MapReduce 任务，并不适用于 Amazon EMR 上的其它工作负载。 使用案例：分析集群性能，监控集群进度 单位：计数
MissingBlocks	HDFS 在其中没有副本的数据块的数量。这些数据块可能已损坏。 使用案例：监控集群运行状况 单位：计数

指标	描述
CorruptBlocks	HDFS 报告的受损数据块的数量。 使用案例：监控集群运行状况 单位：计数
TotalLoad	并发数据传输的总数。 使用案例：监控集群运行状况 单位：计数
MemoryTotalMB	集群中的总内存量。 使用案例：监控集群进度 单位：计数
MemoryReservedMB	预留内存量。 使用案例：监控集群进度 单位：计数
MemoryAvailableMB	可供分配的内存量。 使用案例：监控集群进度 单位：计数
YARNMemoryAvailablePercentage	可用于 YARN 的剩余内存的百分比 (YARNMemoryAvailablePercentage = MemoryAvailableMB / MemoryTotalMB)。此值对基于 YARN 内存使用量扩展集群资源很有用。 单位：百分比

指标	描述
MemoryAllocatedMB	分配给集群的内存量。 使用案例：监控集群进度 单位：计数
PendingDeletionBlocks	标记为进行删除的数据块数。 使用案例：监控集群进度，监控集群运行状况 单位：计数
UnderReplicatedBlocks	需要复制一次或多次的数据块数。 使用案例：监控集群进度，监控集群运行状况 单位：计数
DfsPendingReplicationBlocks	数据块复制状态：所复制的数据块、复制请求的存在时间以及不成功的复制请求。 使用案例：监控集群进度，监控集群运行状况 单位：计数
CapacityRemainingGB	剩余 HDFS 磁盘容量。 使用案例：监控集群进度，监控集群运行状况 单位：计数

Hadoop 1 指标如下：

指标	描述
集群状态	
处于空闲状态	指示集群不再执行任务，但仍处于活动状态并会产生费用。如果没有任何任务和任务处于运行状态，则此指标设置为 1；

指标	描述
	否则设置为 0。系统每隔五分钟检查一次该值，值为 1 仅表示在检查时集群处于空闲状态，并不表示它整个五分钟内都处于空闲状态。为避免误报，当多次连续 5 分钟检查获得的值均为 1 时，您应提出警报。例如，当该值在三十分钟或更长时间内都为 1 时，您应提出警报。 使用案例：监控集群性能 单位：布尔值
JobsRunning	集群中当前处于运行状态的任务数量。 使用案例：监控集群运行状况 单位：计数
JobsFailed	集群中失败的任务数量。 使用案例：监控集群运行状况 单位：计数
映射/减少	
MapTasksRunning	每个作业处于运行中的映射任务的数量。如果您安装了调度器并且有多个任务在运行，那么会生成多个图表。 使用案例：监控集群进度 单位：计数
MapTasksRemaining	每个作业的剩余映射任务的数量。如果您安装了调度器并且有多个任务在运行，那么会生成多个图表。剩余映射任务是指未处于任何以下状态的任务：运行中、已终止或已完成。 使用案例：监控集群进度 单位：计数

指标	描述
MapSlotsOpen	未使用的映射任务容量。该指标将计算为给定集群的最大映射任务数与该集群中当前运行的映射任务总数之差。 使用案例：分析集群性能 单位：计数
RemainingMapTasksPerSlot	剩余映射任务的总数与集群中可用映射插槽总数之比。 使用案例：分析集群性能 单位：比率
ReduceTasksRunning	每个作业处于运行中的缩减任务的数量。如果您安装了调度器并且有多个任务在运行，那么会生成多个图表。 使用案例：监控集群进度 单位：计数
ReduceTasksRemaining	每个作业的剩余缩减任务的数量。如果您安装了调度器并且有多个任务在运行，那么会生成多个图表。 使用案例：监控集群进度 单位：计数
ReduceSlotsOpen	未使用的缩减任务容量。该指标将计算为给定集群的最大缩减任务容量与该集群中当前运行的缩减任务数之差。 使用案例：分析集群性能 单位：计数
节点状态	

指标	描述
CoreNodesRunning	处于运行状态的核心节点的数量。仅当对应的实例组存在时，才会报告此指标的数据点。 使用案例：监控集群运行状况 单位：计数
CoreNodesPending	等待分配的核心节点的数量。请求的所有核心节点可能不会立即可用；此指标报告挂起的请求。仅当对应的实例组存在时，才会报告此指标的数据点。 使用案例：监控集群运行状况 单位：计数
LiveDataNodes	从 Hadoop 接收任务的数据节点的百分率。 使用案例：监控集群运行状况 单位：百分比
TaskNodesRunning	处于运行状态的任务节点的数量。仅当对应的实例组存在时，才会报告此指标的数据点。 使用案例：监控集群运行状况 单位：计数
TaskNodesPending	等待分配的任务节点的数量。请求的所有任务节点可能不会立即可用；此指标报告挂起的请求。仅当对应的实例组存在时，才会报告此指标的数据点。 使用案例：监控集群运行状况 单位：计数

指标	描述
LiveTaskTrackers	处于运行状态的任务跟踪程序的百分率。 使用案例：监控集群运行状况 单位：百分比
IO	
S3BytesWritten	写入 Amazon S3 的字节数。此指标仅聚合 MapReduce 任务，并不适用于 Amazon EMR 上的其它工作负载。 使用案例：分析集群性能，监控集群进度 单位：计数
S3BytesRead	从 Amazon S3 读取的字节数。此指标仅聚合 MapReduce 任务，并不适用于 Amazon EMR 上的其它工作负载。 使用案例：分析集群性能，监控集群进度 单位：计数
HDFSUtilization	当前使用的 HDFS 存储的百分率。 使用案例：分析集群性能 单位：百分比
HDFSBytesRead	从 HDFS 读取的字节数。 使用案例：分析集群性能，监控集群进度 单位：计数
HDFSBytesWritten	写入 HDFS 的字节数。 使用案例：分析集群性能，监控集群进度 单位：计数

指标	描述
MissingBlocks	HDFS 在其中没有副本的数据块的数量。这些数据块可能已损坏。 使用案例：监控集群运行状况 单位：计数
TotalLoad	集群中的所有 DataNode 报告的读取器和写入器的当前总数。 使用案例：诊断高 I/O 可能导致作业执行性能低的程度。运行 DataNode 守护程序的 Worker 节点还必须执行映射并减少任务。随着时间的推移，持续的高 TotalLoad 值可能指示高 I/O 可能是导致性能低的因素。此值的偶尔峰值属于常见情况，通常不指示问题。 单位：计数

集群容量指标

以下指标指示集群的当前容量或目标容量。仅当启用了托管扩展或自动终止时，这些指标才可用。

对于由实例集组成的集群，将在 `Units` 中测量集群容量指标。对于由实例组组成的集群，将根据托管扩展策略中使用的单位类型在 `Nodes` 或 `VCPU` 中测量集群容量指标。有关更多信息，请参阅《Amazon EMR 管理指南》中的使用 EMR 托管扩展。

指标	描述
• `TotalUnitsRequested` • `TotalNodesRequested` • `TotalVCPURequested`	集群中由托管扩展确定的单位/节点/vCPU 的目标总数。 单位：计数

指标	描述
• TotalUnitsRunning • TotalNodesRunning • TotalVCPURunning	正在运行的集群中可用单位/节点/vCPU 的当前总数。当请求集群大小调整时，将在集群中添加或删除新实例后更新此指标。 单位：计数
• CoreUnitsRequested • CoreNodesRequested • CoreVCPURequested	集群中由托管扩展确定的核心单位/节点//vCPU 的目标数量。 单位：计数
• CoreUnitsRunning • CoreNodesRunning • CoreVCPURunning	集群中正在运行的核心单位/节点/vCPU 的当前数量。 单位：计数
• TaskUnitsRequested • TaskNodesRequested • TaskVCPURequested	集群中由托管扩展确定的任务单位/节点/vCPU 的目标数量。 单位：计数

指标	描述
• TaskUnitsRunning • TaskNodesRunning • TaskVCPURunning	集群中正在运行的任务单位/节点/vCPU 的当前数量。 单位：计数

Amazon EMR 指标的维度

Amazon EMR 数据可以使用下表中的任一维度进行筛选。

维度	描述
JobFlowId	与 集群 ID 相同，它是集群的唯一标识符（以 j-XXXXXXXXXXXXX 形式表示）。您可以通过在 Amazon EMR 控制台中单击集群来找到该值。

通过 CloudWatch 监控 Amazon EMR 事件

Amazon EMR 跟踪事件并在 Amazon EMR 控制台中保存其相关信息最多七天。当集群、实例组、实例集、自动扩缩策略或步骤的状态发生变化时，Amazon EMR 会记录事件。事件捕获事件发生的日期和时间、有关受影响元素的详细信息以及其他关键数据点。

下表列出了 Amazon EMR 事件，以及事件指示的状态或状态变更、事件的严重性、事件类型、事件代码和事件消息。Amazon EMR 将事件表示为 JSON 对象并将其自动发送到事件流。当您设置规则以使用 CloudWatch Events 进行事件处理时，JSON 对象非常重要，因为规则需要在此 JSON 对象中查找匹配模式。有关更多信息，请参阅《Amazon CloudWatch Events 用户指南》中的事件和事件模式和Amazon EMR 事件。

> **Note**
>
> 为确保向您提供最相关的信息，我们会不断完善错误消息。因此，建议您不要通过解析消息中的文本来启动工作流中的后续操作。

集群启动事件

状态或状态变更	严重性	事件类型	事件代码	消息
CREATING	WARN	Amazon EMR 实例集预置	EC2 预置 – 实例容量不足	我们无法为实例集 InstanceFleetID 创建您的 Amazon EMR 集群 ClusterId (ClusterName) Amazon EC2 的实例类型 [Instancetype1, Instancetype2] 竞价容量不足，且可用区 [Instancetype3, Instancetype4] 中的实例类型 [AvailabilityZone1, AvaliabilityZone2] 的按需容量不足。有关如何应对此事件的更多信息，请查看此处的文档。
CREATING	WARN	Amazon EMR 实例组预置	EC2 预置 – 实例容量不足	我们无法为实例组 InstancegroupID 创建您的 Amazon EMR

状态或状态变更	严重性	事件类型	事件代码	消息
				集群 ClusterId （ClusterName） Amazon EC2 的实例类型 Instancetype 在可用区 AvailabilityZone 中的 [Spot or On-Demand] 容量不足。有关如何应对此事件的更多信息，请查看此处的文档。
STARTING	INFO	EMR 集群状态更改	none	已于 Time 请求 Amazon EMR 集群 ClusterId (ClusterName) ，且当前正在创建中。

状态或状态变更	严重性	事件类型	事件代码	消息
STARTING	INFO	EMR 集群状态更改	none	**Note** 仅适用于带实例集配置和 Amazon EC2 中选定的多个可用区的集群。 Amazon EMR 集群 ClusterId (ClusterName) 正在从指定可用区选项中选择的区域 (AvailabilityZoneID) 中创建。
STARTING	INFO	EMR 集群状态更改	none	Amazon EMR 集群 ClusterId (ClusterName) 于 Time 开始运行步骤。

状态或状态变更	严重性	事件类型	事件代码	消息
WAITING	INFO	EMR 集群状态更改	none	Amazon EMR 集群 ClusterId (ClusterName) 于 Time 创建，并已准备就绪。 - 或 - Amazon EMR 集群 ClusterId (ClusterName) 于 Time 完成所有待处理步骤的运行。 **Note** 处于 WAITING 状态的集群可能仍是正在处理的作业。

Note

当您的 EMR 集群在创建集群或调整集群大小操作期间遇到来自 Amazon EC2 的实例集或实例组容量不足错误时，会定期触发事件代码为 EC2 provisioning - Insufficient Instance Capacity 的事件。有关如何响应这些事件的更多信息，请参阅 响应 Amazon EMR 集群实例容量不足事件。

集群终止事件

状态或状态变更	严重性	事件类型	事件代码	消息
TERMINATED	严重性视状态变更原因而定，如下所述： • **CRITICAL** 如果集群因以下任一状态变更原因而终止：INTERNAL_ERROR 、VALIDATION_ERROR 、INSTANCE_FAILURE 、BOOTSTRAP_FAILURE 或 STEP_FAILURE 。 • **INFO** 如果集群因以下任一状态变更原因而终止：USER_REQUEST 或 ALL_STEPS_COMPLETED 。	EMR 集群状态更改	none	Amazon EMR 集群 ClusterId (ClusterName) 已于 Time 终止，原因是 StateChangeReason: Code 。
TERMINATED_WITH_ERRORS	CRITICAL	EMR 集群状态更改	none	Amazon EMR 集群 ClusterId (ClusterName) 已于 Time 因错误终止，原因是 StateChan

状态或状态变更	严重性	事件类型	事件代码	消息
				geReason: Code 。

实例集状态更改事件

> **Note**
>
> 实例集配置仅在 Amazon EMR 发行版 4.8.0 及更高版本（不包括 5.0.0 和 5.0.3）中可用。

状态或状态变更	严重性	事件类型	事件代码	消息
从 PROVISIONING 到 WAITING	INFO		none	Amazon EMR 集群 InstanceFleetID 中对实例集 ClusterId (ClusterName) 的预置已完成。预置已于 Time 开始，并且已花费 Num 分钟。实例集现在的按需容量为 Num，竞价型容量为 Num。目标按需容量为 Num，目标竞价型容量为 Num。
从 WAITING 到 RESIZING	INFO		none	对 Amazon EMR 集群 ClusterId (ClusterName) 中实例集 InstanceFleetID 的大小

状态或状态变更	严重性	事件类型	事件代码	消息
				调整已于 Time 开始。实例集的大小从按需容量 Num 调整到目标的 Num，而竞价型容量则从 Num 调整到目标的 Num。
从 RESIZING 到 WAITING	INFO		none	对 Amazon EMR 集群 ClusterId （Cluster Name） 中的实例集 InstanceFleetID 的大小调整操作已完成。大小调整已于 Time 开始，并且已花费 Num 分钟。实例集现在的按需容量为 Num，竞价型容量为 Num。目标按需容量为 Num，目标竞价型容量为 Num。

状态或状态变更	严重性	事件类型	事件代码	消息
从 RESIZING 到 WAITING	INFO		none	Amazon EMR 集群 ClusterId (Cluster Name) 中的实例集 InstanceFleetID 的大小调整操作已超时并且已停止。大小调整已于 Time 开始，并且在 Num 分钟后停止。实例集现在的按需容量为 Num，竞价型容量为 Num。目标按需容量为 Num，目标竞价型容量为 Num。
SUSPENDED	ERROR		none	由于以下原因，Amazon EMR 集群 ClusterId (ClusterName) 中的实例集 InstanceFleetID 于 Time 被捕获：ReasonDesc 。

状态或状态变更	严重性	事件类型	事件代码	消息
RESIZING	WARNING		none	对 Amazon EMR 集群 ClusterId （Cluster Name） 中的实例集 InstanceFleetID 的大小调整操作发生卡顿，原因为：ReasonDesc 。
WAITING 或 Running	INFO		none	当 Amazon EMR 在可用区 AvailabilityZone 中添加竞价型容量时，无法完成 Amazon EMR 集群 ClusterId （Cluster Name） 中的实例集 InstanceFleetID 的大小调整操作。已取消您预置额外 Spot 容量的请求。有关推荐的操作，请查看 实例和可用区灵活性的最佳实践 并重试。

状态或状态变更	严重性	事件类型	事件代码	消息
WAITING 或 Running	INFO		none	Amazon EMR 集群 InstanceFleetID 中的实例集 ClusterId (ClusterName) 的大小调整操作由 Entity 于 Time 启动。

实例集大小调整事件

事件类型	严重性	事件代码	消息
Amazon EMR 实例集大小调整	ERROR	竞价型预置超时	在可用区 AvailabilityZone 中获取竞价型容量时，无法完成 Amazon EMR 集群 ClusterId (ClusterName) 中实例集 InstanceFleetID 的大小调整操作。我们现在已经取消了您的请求并停止尝试预置任何额外的竞价型容量，并且实例集已经预置了 num 的竞价型容量。目标竞价型容量为 num。有关更多信息和建议的操作，请查看此处的文档页面，然后重试。

事件类型	严重性	事件代码	消息
Amazon EMR 实例集大小调整	ERROR	按需预置超时	在可用区 AvailabilityZone 中获取按需容量时，无法完成 Amazon EMR 集群 ClusterId (ClusterName) 中实例集 InstanceFleetID 的大小调整操作。我们现在已经取消了您的请求并停止尝试预置任何额外的按需容量，并且实例集已经预置了 num 的按需容量。目标按需容量为 num。有关更多信息和建议的操作，请查看此处的文档页面，然后重试。

事件类型	严重性	事件代码	消息
Amazon EMR 实例集大小调整	WARNING	EC2 预置 – 实例容量不足	我们无法完成 EMR 集群 ClusterId （ClusterName） 中实例集 InstanceFleetID 的大小调整操作，因为Amazon EC2 实例类型 [Instancetype1, Instancetype2] 的竞价型容量不足，且可用区 [AvailabilityZone1] 中的实例类型 [Instancetype3, Instancetype4] 的按需容量不足。实例集预置的按需容量为 num，目标按需容量为 num。预置的竞价型容量为 num，目标竞价型容量为 num。有关如何应对此事件的更多信息，请查看此处的文档。

事件类型	严重性	事件代码	消息
Amazon EMR 实例集大小调整	WARNING	竞价型预置超时 – 继续调整大小	我们仍在为实例集大小调整操作预置竞价型容量，该操作于 time 在可用区 AvailabilityZone 中 [Instance type1, Instancetype2] 的 Amazon EMR 集群 ClusterId (ClusterName) 中的实例集 ID InstanceFleetID 启动。对于在 time 启动的之前的大小调整操作，超时时间已过期，因此 Amazon EMR 在将请求的 num 实例的 num 添加到您的实例集后停止预置竞价型容量。有关更多信息，请查看此处的文档页面。

事件类型	严重性	事件代码	消息
Amazon EMR 实例集大小调整	WARNING	按需预置超时 – 继续调整大小	我们仍在为实例集大小调整操作预置按需容量，该操作于 time 在可用区 AvailabilityZone 中 [Instance type1, Instancetype2] Amazon EMR 集群 ClusterId (ClusterName) 中的实例集 ID InstanceFleetID 启动。对于在 time 启动的之前的大小调整操作，超时时间已过期，因此 Amazon EMR 在将请求的 num 实例的 num 添加到您的实例集后停止预置按需容量。有关更多信息，请查看此处的文档页面。

> **Note**
>
> 超时到期后，当 Amazon EMR 停止为实例集预置竞价型或按需容量时，就会发出预置超时事件。有关如何响应这些事件的更多信息，请参阅 响应 Amazon EMR 集群实例集调整大小超时事件。

实例组事件

事件类型	严重性	事件代码	消息
从 RESIZING 到 Running	INFO	none	对 Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 的大小调整操作已完成。它当前拥有 Num 个实例。大小调整操作已于 Time 开始，花费 Num 分钟时间完成。
从 RUNNING 到 RESIZING	INFO	none	对 Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceGroupID 的大小调整已于 Time 开始。它的实例数从 Num 个调整为 Num 个。
SUSPENDED	ERROR	none	由于以下原因，Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 于 Time 被捕获：ReasonDesc 。
RESIZING	WARNING	none	对 Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceG

事件类型	严重性	事件代码	消息
			roupID 的大小调整操作发生卡顿，原因为：ReasonDesc 。
Amazon EMR 实例组调整大小	WARNING	EC2 预置 – 实例容量不足	我们无法完成 EMR 集群 ClusterId (ClusterName) 中于 time 开始对实例组 InstanceGroupID 的调整大小操作，因为 Amazon EC2 在可用区 [AvailabilityZone1] 中的实例类型 [Instancetype] 容量 Spot/On Demand 不足。到目前为止，该实例组的运行实例计数为 num，请求的实例计数为 num。有关如何应对此事件的更多信息，请查看此处的文档。
从 RUNNING 到 RESIZING	INFO	none	Amazon EMR 集群 InstanceGroupID 中的实例组 ClusterId (ClusterName) 的大小调整由 Entity 于 Time 启动。

> **Note**
>
> 对于 Amazon EMR 5.21.0 及更高版本，您可以覆盖集群配置，并为运行的集群中的每个实例组指定额外的配置分类。要完成此操作，您可以使用 Amazon EMR 控制台、Amazon Command Line Interface（Amazon CLI）或 Amazon SDK。有关更多信息，请参阅为运行的集群中的实例组提供配置。

下表列出了重新配置操作的 Amazon EMR 事件，以及事件指示的状态或状态变更、事件的严重性和事件消息。

状态或状态变更	严重性	消息
RUNNING	INFO	Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 的重新配置由用户于 Time 启动。请求的配置版本为 Num。
从 RECONFIGURING 到 Running	INFO	对 Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 的重新配置操作已完成。重新配置已于 Time 开始，花费 Num 分钟完成。当前配置版本为 Num。
从 RUNNING 到 RECONFIGURING in	INFO	对 Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceGroupID 的重新配置已于 Time 开始。它从版本号 Num 配置为版本号 Num。
RESIZING	INFO	Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceG

状态或状态变更	严重性	消息
		roupID 的配置版本 Num 的重新配置操作在 Time 被临时阻止，因为实例组处于 State 状态。
RECONFIGURING	INFO	Amazon EMR 集群 ClusterId (ClusterName) 中实例组 Time 的实例计数 Num 的调整大小操作在 InstanceGroupID 被临时阻止，因为实例组处于 State 状态。
RECONFIGURING	WARNING	Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceGroupID 的重新配置操作于 Time 失败，失败之前经过了 Num 分钟。失败的配置版本为 Num。
RECONFIGURING	INFO	配置正在恢复到在 Time 时 Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceGroupID 的上一个成功版本号 Num。新配置版本为 Num。
从 RECONFIGURING 到 Running	INFO	配置已成功恢复到在 Time 时 Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceGroupID 的上一个成功版本号 Num。新配置版本为 Num。

状态或状态变更	严重性	消息
从 RECONFIGURING 到 SUSPENDED	CRITICAL	无法恢复到在 Time 时 Amazon EMR 集群 ClusterId (ClusterName) 中实例组 InstanceGroupID 的上一个成功版本号 Num。

自动伸缩策略事件

状态或状态变更	严重性	消息
PENDING	INFO	自动扩缩策略已于 Time 添加到 Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 。策略正在等待附加。 - 或 - Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 的自动扩缩策略已于 Time 更新。策略正在等待附加。
ATTACHED	INFO	Amazon EMR 集群 ClusterId (ClusterName) 中的实例组 InstanceGroupID 的自动扩缩策略已于 Time 附加。
DETACHED	INFO	Amazon EMR 集群 ClusterId (Cluster

状态或状态变更	严重性	消息
		Name） 中的实例组 InstanceGroupID 的自动扩缩策略已于 Time 分离。
FAILED	ERROR	Amazon EMR 集群 ClusterId （Cluster Name） 中的实例组 InstanceGroupID 的自动扩缩策略无法附加，并于 Time 失败。 - 或 - Amazon EMR 集群 ClusterId （Cluster Name） 中的实例组 InstanceGroupID 的自动扩缩策略无法分离，并于 Time 失败。

步骤事件

状态或状态变更	严重性	消息
PENDING	INFO	步骤 StepID （StepName） 已于 Time 添加到 Amazon EMR 集群 ClusterId （ClusterName），正在等待执行。
CANCEL_PENDING	WARN	Amazon EMR 集群 ClusterId （ClusterName） 中的步骤 StepID （StepName） 已于 Time 取消，正在等待取消。

状态或状态变更	严重性	消息
RUNNING	INFO	Amazon EMR 集群 ClusterId（ClusterName） 中的步骤 StepID（StepName） 已于 Time 开始运行。
COMPLETED	INFO	Amazon EMR 集群 ClusterId（ClusterName） 中的步骤 Time 已于 StepID（StepName） 完成执行。此步骤已于 Time 开始运行，花费 Num 分钟时间完成。
CANCELLED	WARN	Amazon EMR 集群 ClusterId （ClusterName） 中的集群步骤 StepID（StepName） 的取消请求于 Time 成功完成，此步骤现已取消。
FAILED	ERROR	Amazon EMR 集群 ClusterId（ClusterName） 中的步骤 StepID（StepName） 于 Time 失败。

使用 Amazon EMR 控制台查看事件

对于每个集群，您可以在详细信息窗格中查看简单的事件列表，该列表按发生顺序降序列出事件。您还可以按照事件发生顺序的降序查看区域中所有集群的所有事件。

如果您不希望用户查看区域的所有集群事件，请向附加到用户的策略添加一条语句，该语句拒绝对 "Effect": "Deny" 操作的权限 (elasticmapreduce:ViewEventsFromAllClustersInConsole)。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台查看区域中所有集群的事件

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Events（事件）。

使用新控制台查看特定集群的事件

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择集群。
3. 要查看您的所有事件，请在集群详细信息页面上选择 Events（事件）选项卡。

Old console

使用旧控制台查看区域中所有集群的事件

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/。
2. 选择 Events (事件)。

使用旧控制台查看特定集群的事件

1. 通过以下链接打开 Amazon EMR 控制台：https://console.aws.amazon.com/elasticmapreduce/。
2. 选择 Cluster List (集群列表)，选择一个集群，然后选择 View details (查看详细信息)。
3. 在集群详细信息窗格中选择 Events (事件)。

响应 CloudWatch 事件

本节介绍您可以针对 Amazon EMR 作为 CloudWatch 事件消息发出的可操作事件响应的各种方式。

主题

- 使用 CloudWatch 为 Amazon EMR 事件创建规则
- 对 CloudWatch 指标设置警报
- 响应 Amazon EMR 集群实例容量不足事件
- 响应 Amazon EMR 集群实例集调整大小超时事件

使用 CloudWatch 为 Amazon EMR 事件创建规则

Amazon EMR 自动将事件发送到 CloudWatch 事件流。您可以根据特定的模式创建匹配事件的规则，以便将事件路由到目标来执行操作，如发送电子邮件通知。针对事件的 JSON 对象匹配模式。有关 Amazon EMR 事件的详细信息，请参阅《Amazon CloudWatch Events 用户指南》中的 Amazon EMR 事件。

有关设置 CloudWatch 事件规则的信息，请参阅创建触发事件的 CloudWatch 规则。

对 CloudWatch 指标设置警报

Amazon EMR 将指标推送到 Amazon CloudWatch。作为响应，您可以使用 CloudWatch 对Amazon EMR 指标设置警报。例如，您可以在 CloudWatch 中配置警报，用于在 HDFS 使用率超过 80% 时立即向您发送电子邮件。有关详细说明，请参阅《Amazon CloudWatch 用户指南》中的创建或编辑 CloudWatch 警报。

响应 Amazon EMR 集群实例容量不足事件

概述

当所选可用区没有足够的容量来满足您的集群启动或调整大小请求时，Amazon EMR 集群会返回事件代码 `EC2 provisioning - Insufficient Instance Capacity`。如果 Amazon EMR 反复遇到容量不足异常，并且无法满足您的集群启动或集群调整大小操作的预置请求，则对于实例组和实例集该事件都会定期触发。

本页介绍在 EMR 集群发生此类事件时，如何最好地响应此类事件。

对容量不足事件的建议响应

我们建议您通过以下方式之一来应对容量不足事件：

- 等待容量恢复。容量经常变化，因此容量不足的异常可以自行恢复。只要 Amazon EC2 容量可用，您的集群就会开始或完成大小调整。
- 或者，您可以终止集群，修改实例类型配置，然后使用更新的集群配置请求创建新集群。有关更多信息，请参阅实例和可用区灵活性的最佳实践。

您还可以设置对容量不足事件的规则或自动响应，如下一节所述。

从容量不足事件中自动恢复

您可以构建自动化以响应 Amazon EMR 事件，例如带有事件代码 `EC2 provisioning - Insufficient Instance Capacity` 的事件。例如，以下 Amazon Lambda 函数终止具有使用按需型实例的实例组的 EMR 集群，然后创建一个新的 EMR 集群，其实例组包含的实例类型与原始请求不同。

以下条件会触发自动流程的发生：

- 主节点或核心节点的容量不足事件已持续超过 20 分钟。
- 集群未处于就绪或等待状态。有关 EMR 集群状态的更多信息，请参阅 了解集群的生命周期。

> **Note**
>
> 在为容量不足异常建立自动化流程时，应考虑容量不足事件是可以恢复的。容量经常发生变化，只要 Amazon EC2 容量可用，您的集群就会恢复调整大小或开始操作。

Example 响应容量不足事件的功能

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone

INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE = "EMR Instance Group Provisioning"
INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE = (
    "EC2 provisioning - Insufficient Instance Capacity"
)
ALLOWED_INSTANCE_TYPES_TO_USE = [
```

```
    "m5.xlarge",
    "c5.xlarge",
    "m5.4xlarge",
    "m5.2xlarge",
    "t3.xlarge",
]
CLUSTER_START_ACCEPTABLE_STATES = ["WAITING", "RUNNING"]
CLUSTER_START_SLA = 20

CLIENT = boto3.client("emr", region_name="us-east-1")

# checks if the incoming event is 'EMR Instance Fleet Provisioning' with eventCode 'EC2
 provisioning - Insufficient Instance Capacity'
def is_insufficient_capacity_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE
        )


# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceGroupType could be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]
    clusterCreationTime = describeClusterResponse["Cluster"]["Status"]["Timeline"][
        "CreationDateTime"
    ]
    clusterState = describeClusterResponse["Cluster"]["Status"]["State"]

    now = datetime.datetime.now()
    now = now.replace(tzinfo=timezone.utc)
    isClusterStartSlaBreached = clusterCreationTime < now - datetime.timedelta(
        minutes=CLUSTER_START_SLA
    )

    # Check if instance group receiving Insufficient capacity exception is CORE or
 PRIMARY (MASTER),
    # and it's been more than 20 minutes since cluster was created but the cluster
 state and the cluster state is not updated to RUNNING or WAITING
    if (
```

```
        (instanceGroupType == "CORE" or instanceGroupType == "MASTER")
        and isClusterStartSlaBreached
        and clusterState not in CLUSTER_START_ACCEPTABLE_STATES
    ):
        return True
    else:
        return False


# Choose item from the list except the exempt value
def choice_excluding(exempt):
    for i in ALLOWED_INSTANCE_TYPES_TO_USE:
        if i != exempt:
            return i


# Create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceGroupType cloud be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]

    # Following two lines assumes that the customer that created the cluster already
 knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # Select new instance types to include in the new createCluster request
    instanceTypeForMaster = (
        instanceTypesFromOriginalRequestMaster
        if instanceGroupType != "MASTER"
        else choice_excluding(instanceTypesFromOriginalRequestMaster)
    )
    instanceTypeForCore = (
        instanceTypesFromOriginalRequestCore
        if instanceGroupType != "CORE"
        else choice_excluding(instanceTypesFromOriginalRequestCore)
    )

    print("Starting to create cluster...")
    instances = {
        "InstanceGroups": [
            {
                "InstanceRole": "MASTER",
                "InstanceCount": 1,
```

```
                "InstanceType": instanceTypeForMaster,
                "Market": "ON_DEMAND",
                "Name": "Master",
            },
            {
                "InstanceRole": "CORE",
                "InstanceCount": 1,
                "InstanceType": instanceTypeForCore,
                "Market": "ON_DEMAND",
                "Name": "Core",
            },
        ]
    }
    response = CLIENT.run_job_flow(
        Name="Test Cluster",
        Instances=instances,
        VisibleToAllUsers=True,
        JobFlowRole="EMR_EC2_DefaultRole",
        ServiceRole="EMR_DefaultRole",
        ReleaseLabel="emr-6.10.0",
    )

    return response["JobFlowId"]


# Terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")


def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response


def lambda_handler(event, context):
    if is_insufficient_capacity_event(event):
        print(
            "Received insufficient capacity event for instanceGroup, clusterId: "
            + event["detail"]["clusterId"]
        )
```

```
        describeClusterResponse = describe_cluster(event)

        shouldTerminateCluster = is_cluster_eligible_for_termination(
            event, describeClusterResponse
        )
        if shouldTerminateCluster:
            terminate_cluster(event)

            clusterId = create_cluster(event)
            print("Created a new cluster, clusterId: " + clusterId)
        else:
            print(
                "Cluster is not eligible for termination, clusterId: "
                + event["detail"]["clusterId"]
            )

    else:
        print("Received event is not insufficient capacity event, skipping")
```

响应 Amazon EMR 集群实例集调整大小超时事件

概述

Amazon EMR 集群在对实例集集群执行调整大小操作时会发出事件。超时到期后，当 Amazon EMR 停止为实例集预置竞价型或按需容量时，就会发出预置超时事件。用户可以将超时持续时间配置为实例集调整大小规范的一部分。在对相同实例集连续调整大小的情况下，当前调整大小操作的超时到期时，Amazon EMR 会发出 `Spot provisioning timeout - continuing resize` 或 `On-Demand provisioning timeout - continuing resize` 事件。然后，它开始为队列的下一次调整大小操作预置容量。

响应实例集调整大小超时事件

我们建议您通过以下方式之一来响应预置超时事件：

- 重访调整大小规范，然后重试调整大小操作。由于容量频繁变化，只要 Amazon EC2 容量可用，您的集群就会成功调整大小。我们建议客户为需要更严格 SLA 的任务配置较低的超时持续时间值。
- 或者，您可以：
  - 根据实例和可用区灵活性的最佳实践启动具有多种实例类型的新集群，或
  - 启动具有按需容量的集群
- 对于配置超时 – 继续调整事件大小，您还可以等待调整大小操作的处理。Amazon EMR 将继续按顺序处理针对实例触发的调整大小操作，同时遵守配置的调整大小规范。

您还可以设置对该事件的规则或自动响应，如下一节所述。

自动从预置超时事件中恢复

您可以使用 Spot Provisioning timeout 事件代码构建自动化以响应 Amazon EMR 事件。例如，以下 Amazon Lambda 函数关闭了具有使用竞价型实例作为任务节点的实例集的 EMR 集群，然后创建一个新的 EMR 集群，其实例集包含的实例类型比原始请求更加多样化。在此示例中，为任务节点发出的 Spot Provisioning timeout 事件将触发 Lambda 函数的执行。

Example 响应 **Spot Provisioning timeout** 事件的示例函数

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone

SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE = "EMR Instance Fleet Resize"
SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE = (
    "Spot Provisioning timeout"
)

CLIENT = boto3.client("emr", region_name="us-east-1")

# checks if the incoming event is 'EMR Instance Fleet Resize' with eventCode 'Spot
 provisioning timeout'
def is_spot_provisioning_timeout_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE
        )


# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceFleetType could be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]
```

```
    # Check if instance fleet receiving Spot provisioning timeout event is TASK
    if (instanceFleetType == "TASK"):
        return True
    else:
        return False


# create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceFleetType cloud be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # the following two lines assumes that the customer that created the cluster
 already knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # select new instance types to include in the new createCluster request
    instanceTypesForTask = [
        "m5.xlarge",
        "m5.2xlarge",
        "m5.4xlarge",
        "m5.8xlarge",
        "m5.12xlarge"
    ]

    print("Starting to create cluster...")
    instances = {
        "InstanceFleets": [
            {
                "InstanceFleetType":"MASTER",
                "TargetOnDemandCapacity":1,
                "TargetSpotCapacity":0,
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesFromOriginalRequestMaster,
                        "WeightedCapacity":1,
                    }
                ]
            },
            {
                "InstanceFleetType":"CORE",
                "TargetOnDemandCapacity":1,
```

```
                "TargetSpotCapacity":0,
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesFromOriginalRequestCore,
                        "WeightedCapacity":1,
                    }
                ]
            },
            {
                "InstanceFleetType":"TASK",
                "TargetOnDemandCapacity":0,
                "TargetSpotCapacity":100,
                "LaunchSpecifications":{},
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesForTask[0],
                        "WeightedCapacity":1,
                    },
                    {
                        'InstanceType': instanceTypesForTask[1],
                        "WeightedCapacity":2,
                    },
                    {
                        'InstanceType': instanceTypesForTask[2],
                        "WeightedCapacity":4,
                    },
                    {
                        'InstanceType': instanceTypesForTask[3],
                        "WeightedCapacity":8,
                    },
                    {
                        'InstanceType': instanceTypesForTask[4],
                        "WeightedCapacity":12,
                    }
                ],
                "ResizeSpecifications": {
                    "SpotResizeSpecification": {
                        "TimeoutDurationMinutes": 30
                    }
                }
            }
        ]
    }
    response = CLIENT.run_job_flow(
```

```
        Name="Test Cluster",
        Instances=instances,
        VisibleToAllUsers=True,
        JobFlowRole="EMR_EC2_DefaultRole",
        ServiceRole="EMR_DefaultRole",
        ReleaseLabel="emr-6.10.0",
    )

    return response["JobFlowId"]


# terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")


def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response


def lambda_handler(event, context):
    if is_spot_provisioning_timeout_event(event):
        print(
            "Received spot provisioning timeout event for instanceFleet, clusterId: "
            + event["detail"]["clusterId"]
        )

        describeClusterResponse = describe_cluster(event)

        shouldTerminateCluster = is_cluster_eligible_for_termination(
            event, describeClusterResponse
        )
        if shouldTerminateCluster:
            terminate_cluster(event)

            clusterId = create_cluster(event)
            print("Created a new cluster, clusterId: " + clusterId)
        else:
            print(
                "Cluster is not eligible for termination, clusterId: "
                + event["detail"]["clusterId"]
```

```
            )

    else:
        print("Received event is not spot provisioning timeout event, skipping")
```

使用 Ganglia 查看集群应用程序指标

Ganglia 适用于 Amazon EMR 版本 4.2 及更高版本。Ganglia 是一个开源项目 (可扩展的分布式系统)。旨在监控集群和网格，同时尽量减少对其性能的影响。当您在集群上启用 Ganglia 时，您可以生成报告并查看整个集群的性能，还可以检查单个节点实例的性能。还配置 Ganglia 以提取和可视化 Hadoop 和 Spark 指标。有关更多信息，请参阅《Amazon EMR 版本指南》中的 Ganglia。

在 Amazon CloudTrail 中记录 Amazon EMR API 调用

Amazon EMR 与 Amazon CloudTrail 集成，后者是在 Amazon EMR 中提供用户、角色或Amazon服务所采取操作的记录的服务。CloudTrail 将 Amazon EMR 的所有 API 调用作为事件捕获。捕获调用中包括通过 Amazon EMR 控制台的调用和对 Amazon EMR API 操作的代码调用。如果您创建跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 Amazon EMR 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。使用通过 CloudTrail 收集的信息，您可以确定向 Amazon EMR 发出了什么请求、发出请求时使用的 IP 地址、何人发出的请求、请求的发出时间以及其它详细信息。

要了解有关 CloudTrail 的更多信息，请参阅《Amazon CloudTrail 用户指南》。

CloudTrail 中的 Amazon EMR 信息

在您创建 Amazon 账户时，将在该账户上启用 CloudTrail。当 Amazon EMR 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其它Amazon服务事件一同保存在事件历史记录中。您可以在 Amazon 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用 CloudTrail 事件历史记录查看事件。

要持续记录Amazon账户中的事件（包括 Amazon EMR 的事件），请创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service（Amazon S3）存储桶。预设情况下，在控制台中创建跟踪时，此跟踪应用于所有 Amazon 区域。此跟踪记录在 Amazon 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）桶。此外，您可以配置其他 Amazon 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知

- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

所有 Amazon EMR 操作都由 CloudTrail 记录，并记录在 Amazon EMR API 参考中。例如，对 RunJobFlow、ListCluster 和 DescribeCluster 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 Amazon Identity and Access Management（IAM）用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它 Amazon 服务发出。

在进程而不是用户创建集群的情况下，您可以使用 principalId 标识符，以确定与创建集群关联的用户。有关更多信息，请参阅 CloudTrail userIdentity 元素。

示例：Amazon EMR 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 RunJobFlow 操作。

```
{
 "Records": [
 {
         "eventVersion":"1.01",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"EX_PRINCIPAL_ID",
            "arn":"arn:aws:iam::123456789012:user/temporary-user-xx-7M",
            "accountId":"123456789012",
            "userName":"temporary-user-xx-7M"
         },
         "eventTime":"2018-03-31T17:59:21Z",
         "eventSource":"elasticmapreduce.amazonaws.com",
         "eventName":"RunJobFlow",
         "awsRegion":"us-west-2",
```

```
         "sourceIPAddress":"192.0.2.1",
         "userAgent":"aws-sdk-java/unknown-version Linux/xx Java_HotSpot(TM)_64-
Bit_Server_VM/xx",
         "requestParameters":{
            "tags":[
               {
                  "value":"prod",
                  "key":"domain"
               },
               {
                  "value":"us-west-2",
                  "key":"realm"
               },
               {
                  "value":"VERIFICATION",
                  "key":"executionType"
               }
            ],
            "instances":{
               "slaveInstanceType":"m5.xlarge",
               "ec2KeyName":"emr-integtest",
               "instanceCount":1,
               "masterInstanceType":"m5.xlarge",
               "keepJobFlowAliveWhenNoSteps":true,
               "terminationProtected":false
            },
            "visibleToAllUsers":false,
            "name":"MyCluster",
            "ReleaseLabel":"emr-5.16.0"
         },
         "responseElements":{
            "jobFlowId":"j-2WDJCGEG4E6AJ"
         },
         "requestID":"2f482daf-b8fe-11e3-89e7-75a3d0e071c5",
         "eventID":"b348a38d-f744-4097-8b2a-e68c9b424698"
      },
  ...additional entries
   ]
}
```

使用集群扩展

您可以自动或手动调整对 Amazon EMR 集群可用的 Amazon EC2 实例的数目，以满足各种工作负载的不同需求。要使用自动扩展，您有两个选项。您可以启用 Amazon EMR 托管扩展或创建自定义自动扩展策略。下表介绍了两个选项之间的区别。

	Amazon EMR 托管扩展	自定义自动扩展
扩展策略和规则	无需策略。Amazon EMR 通过持续评估集群指标并做出优化的扩展决策来管理自动扩缩活动。	您需要定义和管理自动伸缩策略和规则，如触发扩缩活动的特定条件、评估期、冷却时间等。
支持的 Amazon EMR 发行版	Amazon EMR 版本 5.30.0 及更高版本（Amazon EMR 版本 6.0.0 除外）	Amazon EMR 版本 4.0.0 及更高版本
支持的集群构成	实例组或实例队列	仅实例组
扩展限制配置	为整个集群配置扩展限制。	只能为每个实例组配置扩展限制。
指标评估频率	每 5 到 10 秒一次 更频繁地评估指标使 Amazon EMR 能够做出更精确的扩展决策。	您只能以五分钟的增量定义评估期。
支持的应用程序	只支持 YARN 应用程序，如 Spark、Hadoop、Hive、Flink。Amazon EMR 托管扩展不支持不基于 YARN 的应用程序，例如 Presto 或 HBase。	在定义自动扩展规则时，您可以选择支持哪些应用程序。

注意事项

- 一个 Amazon EMR 集群始终由一个或三个主节点组成。首次配置集群后，您只能扩展核心节点和任务节点，而无法扩展集群的主节点数量。

- 对于实例组，重新配置操作和大小调整操作是依次进行的，而不是同时进行的。如果您在调整实例组大小时发起重新配置，则实例组完成正在进行的大小调整后，将会立即开始重新配置。如果您在实例组重新配置时启动调整大小操作，则会按相反的顺序操作。

在 Amazon EMR 中使用托管扩展

> **Important**
>
> 我们强烈建议您使用最新的 Amazon EMR 发行版（Amazon EMR 6.14.0）来进行托管式自动扩缩。在某些早期的发行版中，您可能会遇到间歇性的应用程序故障或扩展延迟。Amazon EMR 已通过 5.x 发行版 5.30.2、5.31.1、5.32.1、5.33.1 及更高版本，以及 6.x 发行版 6.1.1、6.2.1、6.3.1 及更高版本解决了此问题。有关区域和发行版可用性的更多信息，请参阅托管式自动扩缩功能的可用性。

概述

使用 Amazon EMR 版本 5.30.0 及更高版本（Amazon EMR 6.0.0 除外），您可以启用 Amazon EMR 托管式自动扩缩功能。托管扩展让您根据工作负载自动增加或减少集群中实例或单元的数量。Amazon EMR 会持续评估集群指标，以便做出扩展决策，从而优化集群的成本和速度。托管扩展适用于由实例组或实例队列组成的集群。

托管式自动扩缩功能的可用性

- 对于亚太地区（雅加达）区域，Amazon EMR 托管式自动扩缩功能现已在 Amazon EMR 6.14.0 及更高版本中推出。
- 在下列 Amazon Web Services 区域，Amazon EMR 5.30.0 和 6.1.0 及更高版本提供了 Amazon EMR 托管式自动扩缩功能：

  美国东部（弗吉尼亚州北部和俄亥俄州）、美国西部（俄勒冈州和加利福尼亚州北部）、南美洲（圣保罗）、欧洲地区（法兰克福、爱尔兰、伦敦、米兰、巴黎和斯德哥尔摩）、加拿大（中部）、亚太地区（香港、孟买、首尔、新加坡、悉尼和东京）、中东（巴林）、非洲（开普敦）、Amazon GovCloud（美国东部）、Amazon GovCloud（美国西部）、中国（北京，由光环新网运营）和中国（宁夏，由西云数据运营）。
- Amazon EMR 托管扩展仅适用于 YARN 应用程序，如 Spark、Hadoop、Hive 和 Flink。不支持非基于 YARN 的应用程序，如 Presto 和 HBase。

托管扩展参数

您必须为托管扩展配置以下参数。该限制仅适用于核心节点和任务节点。初始配置后，无法扩展主节点。

- Minimum (最小) (`MinimumCapacityUnits`) – 集群中允许的 EC2 容量的下限。其衡量方式为通过虚拟中央处理单位 (vCPU) 核心或实例组中的实例进行衡量。其衡量方式为通过实例集单位进行衡量。
- Maximum (最大) (`MaximumCapacityUnits`) – 集群中允许的 EC2 容量的上限。其衡量方式为通过虚拟中央处理单位 (vCPU) 核心或实例组中的实例进行衡量。其衡量方式为通过实例集单位进行衡量。
- On-Demand limit (按需限制) (`MaximumOnDemandCapacityUnits`) (可选) – 集群中按需市场类型允许的 EC2 容量的上限。如果未指定此参数，则默认为 `MaximumCapacityUnits` 的值。
  - 此参数用于在按需实例和竞价型实例之间拆分容量分配。例如，如果您将最小参数设置为 2 个实例，最大参数设置为 100 个实例，按需限制设置为 10 个实例，则 Amazon EMR 托管扩展将纵向扩展到 10 个按需型实例，并将剩余容量分配给竞价型实例。有关更多信息，请参阅节点分配方案。
- Maximum core nodes (最大核心节点) (`MaximumCoreCapacityUnits`) (可选) – 集群中核心节点类型允许的 EC2 容量的上限。如果未指定此参数，则默认为 `MaximumCapacityUnits` 的值。
  - 此参数用于在核心节点和任务节点之间分配容量。例如，如果您将最小参数设置为 2 个实例，最大参数设置为 100 个实例，最大核心节点设置为 17 个实例，则 Amazon EMR 托管扩展将纵向扩展到 17 个核心节点，并将剩余的 83 个实例分配给任务节点。有关更多信息，请参阅节点分配方案。

有关托管式扩展参数的更多信息，请参阅 `ComputeLimits`。

Amazon EMR 托管式自动扩缩功能注意事项

- 仅少数 Amazon Web Services 区域和 Amazon EMR 发行版支持托管式自动扩缩功能。有关更多信息，请参阅托管式自动扩缩功能的可用性。
- 您必须为 Amazon EMR 托管扩展配置所需参数。有关更多信息，请参阅托管扩展参数。
- 要使用托管式扩展，指标收集器进程必须能够连接到公有 API 端点，以便在 API Gateway 中进行托管式扩展。如果您通过 Amazon Virtual Private Cloud 使用私有 DNS 名称，托管式扩缩将无法正常运行。为确保托管式扩展正常运行，我们建议您执行以下操作之一：
  - 从您的 Amazon VPC 中删除 API Gateway 接口 VPC 终端节点。

  - 按照我从 VPC 连接到 API Gateway API 时，为什么会收到 HTTP 403 禁止错误？中的说明禁用私有 DNS 名称设置。
  - 在您的私有子网中启动集群。有关更多信息，请参阅 私有子网 中的主题。
- 如果您的 YARN 作业在缩减过程中出现间歇性运行缓慢的情况，并且 YARN 资源管理器日志显示在此期间您的大多数节点都被列入拒绝列表，则可以调整停用超时阈值。

  将 `spark.blacklist.decommissioning.timeout` 从 1 小时减少到 1 分钟，以使节点可供其他待处理容器继续进行任务处理。

  您还应将 `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` 设置为更大的值，以确保当时间最长的“Spark 任务”仍在节点上运行时，Amazon EMR 不会强制终止该节点。当前默认值为 60 分钟，这意味着一旦节点进入停用状态，YARN 将在 60 分钟后强制终止容器。

  以下 YARN 资源管理器日志行示例显示了已添加到停用状态的节点：

```
2021-10-20 15:55:26,994 INFO
 org.apache.hadoop.YARN.server.resourcemanager.DefaultAMSProcessor
 (IPC Server handler 37 on default port 8030): blacklist are updated in
 Scheduler.blacklistAdditions: [ip-10-10-27-207.us-west-2.compute.internal,
 ip-10-10-29-216.us-west-2.compute.internal, ip-10-10-31-13.us-
west-2.compute.internal, ... , ip-10-10-30-77.us-west-2.compute.internal],
 blacklistRemovals: []
```

  查看 details on how Amazon EMR integrates with YARN deny listing during decommissioning of nodes（有关 Amazon EMR 如何在节点停用期间与 YARN 拒绝名单集成的详细信息）、拒绝列出的节点以及配置节点停用行为的更多信息。
- 过度使用 EBS 卷可能会导致托管扩展问题。我们建议您将 EBS 卷的利用率保持在 90% 以下。有关更多信息，请参阅实例存储。
- Amazon CloudWatch 指标对于 Amazon EMR 托管扩展的运行至关重要。我们建议您密切监控 Amazon CloudWatch 指标，以确保数据不会丢失。有关如何配置 CloudWatch 告警以检测缺失指标的更多信息，请参阅使用 Amazon CloudWatch 告警。
- 在未安装 Presto 的 5.30.0 和 5.30.1 的集群上进行托管扩展操作可能会导致应用程序故障或导致统一的实例组或实例集处于 `ARRESTED` 状态，尤其是在缩减操作之后快速执行扩展操作时。

  解决方法是即使您的任务不需要 Presto，也可以在使用 Amazon EMR 发行版 5.30.0 和 5.30.1 创建集群时，将 Presto 选为要安装的应用程序。

- 在为 Amazon EMR 托管扩展设置最大核心节点和按需限制时，请考虑实例组和实例集之间的差异。每个实例组包含相同的实例类型和相同的实例购买选项：按需或 Spot。对于每个实例集，您可以指定最多 5 个实例类型，这些类型可预配置为按需实例和竞价型实例。有关更多信息，请参阅使用集或统一实例组创建集群、集选项和 节点分配方案。
- 对于 Amazon EMR 5.30.0 及更高版本，如果您移除主安全组默认的允许所有出站规则 0.0.0.0/，则必须添加一条规则，以允许与您的安全组建立出站 TCP 连接，从而在端口 9443 上访问服务。您的服务访问安全组应允许来自主安全组端口 9443 上的入站 TCP 流量。有关配置安全组的更多信息，请参阅适用于主实例（私有子网）的 Amazon EMR 托管安全组。
- 托管扩展不支持 YARN 节点标签功能。避免在具有托管扩展的集群上使用节点标签。例如，不允许执行程序仅在任务节点上运行。当您在 Amazon EMR 集群中使用节点标签时，可能会发现集群没有向上扩展，这可能会导致应用程序运行缓慢。
- 您可以使用 Amazon CloudFormation 配置 Amazon EMR 托管扩展。有关更多信息，请参阅《Amazon CloudFormation 用户指南》中的 AWS::EMR::Cluster。

功能历史记录

此表列出了对 Amazon EMR 托管扩展功能的更新。

发行日期	能力	Amazon EMR 版本
2023 年 10 月 10 日	托管式自动扩缩功能已在 ap-southeast-3 亚太地区（雅加达）区域开放。	6.14.0 及更高版本
2023 年 7 月 28 日	增强了托管扩展，以便在 Amazon EMR 在纵向扩展当前实例组的过程中遇到延迟时，可以在纵向扩展时切换到不同的任务实例组。	5.34.0 及更高版本，6.4.0 及更高版本
2023 年 6 月 16 日	增强了托管扩展，以了解运行应用程序主节点的节点，这样这些节点就不会被缩减。有关更多信息，请参阅了解节点分配策略和方案。	5.34.0 及更高版本，6.4.0 及更高版本

发行日期	能力	Amazon EMR 版本
2022 年 3 月 21 日	添加了在缩减集群时使用的 Spark 随机排序数据感知。对于启用了 Apache Spark 和托管式扩展功能的 Amazon EMR 集群，Amazon EMR 会持续监控 Spark 执行程序和中间随机排序数据位置。利用这些信息，Amazon EMR 只能缩减不包含积极使用的随机排序数据的未充分利用的实例。这可以防止重新计算丢失的随机排序数据，从而有助于降低成本和提高任务性能。有关更多信息，请参阅 Spark Programming Guide（Spark 编程指南）。	5.34.0 及更高版本，6.4.0 及更高版本

为 Amazon EMR 配置托管扩展

以下部分说明了如何使用 Amazon Web Services Management Console、Amazon SDK for Java 或 Amazon Command Line Interface 启动使用托管扩展的 EMR 集群。

主题

- 使用Amazon Web Services Management Console配置托管扩展
- 使用Amazon CLI配置托管扩展
- 使用 Amazon SDK for Java 配置托管扩展

使用Amazon Web Services Management Console配置托管扩展

您可以在创建集群时使用 Amazon EMR 控制台配置托管式扩缩，也可更改正在运行的集群的托管式扩缩策略。

New console

在通过新控制台创建集群时配置托管扩展

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 选择 Amazon EMR 发行版 emr-5.30.0 或更高版本（emr-6.0.0 版本除外）。
4. 在 Cluster scaling and provisioning option（集群扩展和预置选项）下，选择 Use EMR-managed scaling（使用 EMR 托管扩展）。指定 Minimum（最小）和 Maximum（最大）实例数、Maximum core node（最大核心节点）实例数和 Maximum On-Demand（最大按需型）实例数。
5. 选择适用于集群的任何其他选项。
6. 要启动集群，选择 Create cluster（创建集群）。

使用新控制台在现有集群上配置托管扩展

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。
3. 在集群详细信息页面的 Instances（实例）选项卡上，找到 Instance group settings（实例组设置）部分。选择 Edit cluster scaling（编辑集群扩展），为 Minimum（最小）和 Maximum（最大）实例数及 On-Demand limit（按需限制）指定新值。

Old console

在旧控制台上创建集群时，您可以使用快速选项或高级集群配置选项来配置托管扩展。您还可以通过修改 Summary（摘要）或 Hardware（硬件）页面上的 Managed Scaling（托管式扩缩）设置，从而为正在运行的集群创建或修改托管式扩缩策略。

在通过旧控制台创建集群时使用快速选项配置托管扩展

1. 打开 Amazon EMR 控制台，选择 Create cluster (创建集群)，然后打开 Create Cluster - Quick options (创建集群 – 快速选项)。

2. 在 Cluster scaling and provisioning option（集群扩展和预置选项）旁边的 Hardware configuration（硬件配置）部分中，选中复选框以启用 scale cluster nodes based on workload（根据工作负载扩展集群节点）。
3. 在 Core and task units (核心和任务单位) 下，指定核心实例和任务实例的 Minimum (最小) 和 Maximum (最大) 数量。

在通过旧控制台创建集群时使用高级选项配置托管扩展

1. 在 Amazon EMR 控制台中，依次选择 Create cluster (创建集群)、Go to advanced options (转至高级选项)、Step 1: Software and Steps (步骤 1: 软件和步骤)，然后转至 Step 2: Hardware Configuration (步骤 2: 硬件配置)。
2. 在 Cluster composition (集群构成) 部分中，选择 Instance fleets (实例队列) 或 Uniform instance groups (统一实例组)。
3. 在 Cluster scaling and provisioning option（集群扩展和预置选项）下，选择 Enable cluster scaling（启用集群扩展）。然后，选择 Use EMR managed scaling (使用 EMR 托管扩展)。在 Core and task units（核心和任务单位）下，指定实例或实例集单位的 Minimum（最小）和 Maximum（最大）数量、On-Demand limit（按需限制）和 Maximum Core Node（最大核心节点）计数。

   对于由实例组组成的集群，如果要为每个实例组定义自定义自动扩展策略，您还可以选择 Create a custom automatic scaling policy (创建自定义自动扩展策略)。有关更多信息，请参阅将自动伸缩与实例组的自定义策略结合使用 。

使用旧控制台修改现有集群上的托管扩展

1. 打开 Amazon EMR 控制台，从集群列表中选择集群，然后选择 Hardware (硬件) 选项卡。
2. 在 Cluster scaling and provisioning option（集群扩展和预置选项）部分中，为 Amazon EMR 托管扩展选择 Edit（编辑）。
3. 在 Cluster scaling and provisioning option（集群扩展和预置选项）部分中，为 Minimum（最小）和 Maximum（最大）实例数及 On-Demand limit（按需限制）指定新值。

使用Amazon CLI配置托管扩展

在创建集群时，可以使用 Amazon EMR 的 Amazon CLI 命令配置托管扩展。您可以使用速记语法 (可在相关命令中指定内联 JSON 配置)。也可以引用包含配置 JSON 的文件。您也可以将托管扩展策略应

用于现有集群，并删除以前应用的托管扩展策略。此外，您可以从正在运行的集群中检索扩展策略配置的详细信息。

在集群启动期间启用托管扩展

您可以在集群启动期间启用托管扩展，如以下示例所示。

```
aws emr create-cluster \
 --service-role EMR_DefaultRole \
 --release-label emr-5.36.1 \
 --name EMR_Managed_Scaling_Enabled_Cluster \
 --applications Name=Spark Name=Hbase \
 --ec2-attributes KeyName=keyName,InstanceProfile=EMR_EC2_DefaultRole \
 --instance-groups InstanceType=m4.xlarge,InstanceGroupType=MASTER,InstanceCount=1
 InstanceType=m4.xlarge,InstanceGroupType=CORE,InstanceCount=2 \
 --region us-east-1 \
 --managed-scaling-policy
 ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

在使用 `create-cluster` 时，还可以使用 --managed-scaling-policy 选项指定托管式策略配置。

将托管扩展策略应用于现有集群

您可以将托管扩展策略应用于现有集群，如以下示例所示。

```
aws emr put-managed-scaling-policy
--cluster-id j-123456
--managed-scaling-policy ComputeLimits='{MinimumCapacityUnits=1,
MaximumCapacityUnits=10,  MaximumOnDemandCapacityUnits=10, UnitType=Instances}'
```

也可以使用 `aws emr put-managed-scaling-policy` 命令将托管扩展策略应用于现有集群。以下示例使用对 JSON 文件 `managedscaleconfig.json` 的引用，该文件指定托管扩展策略配置。

```
aws emr put-managed-scaling-policy --cluster-id j-123456 --managed-scaling-policy
 file:///./managedscaleconfig.json
```

以下示例显示 `managedscaleconfig.json` 文件的内容，该文件定义托管扩展策略。

```
{
    "ComputeLimits": {
        "UnitType": "Instances",
        "MinimumCapacityUnits": 1,
```

```
        "MaximumCapacityUnits": 10,
        "MaximumOnDemandCapacityUnits": 10
    }
}
```

检索托管扩展策略配置

GetManagedScalingPolicy 命令检索策略配置。例如，以下命令检索集群 ID 为 j-123456 的集群的配置。

```
aws emr get-managed-scaling-policy --cluster-id j-123456
```

该命令生成以下示例输出。

```
{
   "ManagedScalingPolicy": {
      "ComputeLimits": {
         "MinimumCapacityUnits": 1,
         "MaximumOnDemandCapacityUnits": 10,
         "MaximumCapacityUnits": 10,
         "UnitType": "Instances"
      }
   }
}
```

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

删除托管扩展策略

RemoveManagedScalingPolicy 命令可删除策略配置。例如，以下命令删除集群 ID 为 j-123456 的集群的配置。

```
aws emr remove-managed-scaling-policy --cluster-id j-123456
```

使用 Amazon SDK for Java 配置托管扩展

以下程序摘要说明如何使用 Amazon SDK for Java 配置托管扩展：

```
package com.amazonaws.emr.sample;

import java.util.ArrayList;
```

```
import java.util.List;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.Application;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimits;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimitsUnitType;
import com.amazonaws.services.elasticmapreduce.model.InstanceGroupConfig;
import com.amazonaws.services.elasticmapreduce.model.JobFlowInstancesConfig;
import com.amazonaws.services.elasticmapreduce.model.ManagedScalingPolicy;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowRequest;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowResult;

public class CreateClusterWithManagedScalingWithIG {

 public static void main(String[] args) {
  AWSCredentials credentialsFromProfile = getCreadentials("AWS-Profile-Name-Here");

  /**
   * Create an Amazon EMR client with the credentials and region specified in order to
 create the cluster
   */
  AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
   .withCredentials(new AWSStaticCredentialsProvider(credentialsFromProfile))
   .withRegion(Regions.US_EAST_1)
   .build();

  /**
   * Create Instance Groups - Primary, Core, Task
   */
  InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
    .withInstanceCount(1)
    .withInstanceRole("MASTER")
    .withInstanceType("m4.large")
    .withMarket("ON_DEMAND");

  InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
   .withInstanceCount(4)
   .withInstanceRole("CORE")
```

```
    .withInstanceType("m4.large")
    .withMarket("ON_DEMAND");

  InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
    .withInstanceCount(5)
    .withInstanceRole("TASK")
    .withInstanceType("m4.large")
    .withMarket("ON_DEMAND");

  List<InstanceGroupConfig> igConfigs = new ArrayList<>();
  igConfigs.add(instanceGroupConfigMaster);
  igConfigs.add(instanceGroupConfigCore);
  igConfigs.add(instanceGroupConfigTask);

        /**
         *  specify applications to be installed and configured when Amazon EMR creates
the cluster
         */
  Application hive = new Application().withName("Hive");
  Application spark = new Application().withName("Spark");
  Application ganglia = new Application().withName("Ganglia");
  Application zeppelin = new Application().withName("Zeppelin");

  /**
   * Managed Scaling Configuration -
         * Using UnitType=Instances for clusters composed of instance groups
   *
         * Other options are:
         * UnitType = VCPU ( for clusters composed of instance groups)
         * UnitType = InstanceFleetUnits ( for clusters composed of instance fleets)
         **/
  ComputeLimits computeLimits = new ComputeLimits()
    .withMinimumCapacityUnits(1)
    .withMaximumCapacityUnits(20)
    .withUnitType(ComputeLimitsUnitType.Instances);

  ManagedScalingPolicy managedScalingPolicy = new ManagedScalingPolicy();
  managedScalingPolicy.setComputeLimits(computeLimits);

  // create the cluster with a managed scaling policy
  RunJobFlowRequest request = new RunJobFlowRequest()
          .withName("EMR_Managed_Scaling_TestCluster")
          .withReleaseLabel("emr-5.36.1")          // Specifies the version label for
the Amazon EMR release; we recommend the latest release
```

```
          .withApplications(hive,spark,ganglia,zeppelin)
          .withLogUri("s3://path/to/my/emr/logs")  // A URI in S3 for log files is
 required when debugging is enabled.
          .withServiceRole("EMR_DefaultRole")      // If you use a custom IAM service
 role, replace the default role with the custom role.
          .withJobFlowRole("EMR_EC2_DefaultRole")  // If you use a custom Amazon EMR
 role for EC2 instance profile, replace the default role with the custom Amazon EMR
 role.
          .withInstances(new JobFlowInstancesConfig().withInstanceGroups(igConfigs)
              .withEc2SubnetId("subnet-123456789012345")
              .withEc2KeyName("my-ec2-key-name")
              .withKeepJobFlowAliveWhenNoSteps(true))
          .withManagedScalingPolicy(managedScalingPolicy);
    RunJobFlowResult result = emr.runJobFlow(request);

    System.out.println("The cluster ID is " + result.toString());
 }

 public static AWSCredentials getCredentials(String profileName) {
  // specifies any named profile in .aws/credentials as the credentials provider
  try {
   return new ProfileCredentialsProvider("AWS-Profile-Name-Here")
     .getCredentials();
        } catch (Exception e) {
            throw new AmazonClientException(
                    "Cannot load credentials from .aws/credentials file. " +
                    "Make sure that the credentials file exists and that the profile
 name is defined within it.",
                    e);
        }
 }

 public CreateClusterWithManagedScalingWithIG() { }
}
```

了解节点分配策略和方案

本部分概述了可用于 Amazon EMR 托管扩展的节点分配策略和常见扩展方案。

节点分配策略

Amazon EMR 托管扩展基于以下纵向扩展和缩减策略分配核心节点和任务节点：

纵向扩展策略

- Amazon EMR 托管扩展首先为核心节点添加容量，然后为任务节点添加容量，直到达到所允许的最大容量或直到实现所需的纵向扩展目标容量。
- 当 Amazon EMR 在纵向扩展当前实例组的过程中遇到延迟时，使用托管式扩展的集群会自动切换到不同的任务实例组。
- 如果设置了 `MaximumCoreCapacityUnits` 参数，则 Amazon EMR 会扩展核心节点，直到核心单位达到所允许的最大限制。所有剩余容量都添加到任务节点。
- 如果设置了 `MaximumOnDemandCapacityUnits` 参数，则 Amazon EMR 使用按需型实例扩展集群，直到按需型单位达到所允许的最大限制。使用竞价型实例添加所有剩余容量。
- 如果同时设置了 `MaximumCoreCapacityUnits` 和 `MaximumOnDemandCapacityUnits` 参数，Amazon EMR 在扩展期间会考虑这两个限制。

  例如，如果 `MaximumCoreCapacityUnits` 小于 `MaximumOnDemandCapacityUnits`，Amazon EMR 首先扩展核心节点，直到达到核心容量限制。对于剩余容量，Amazon EMR 首先使用按需型实例扩展任务节点，直到达到按需型限制，然后对任务节点使用竞价型实例。

缩减策略

- Amazon EMR 5.34.0 及更高版本和 Amazon EMR 6.4.0 及更高版本支持可感知 Spark 随机排序数据（Spark 在分区之间重新分配以执行特定操作的数据）的托管式自动扩缩功能。有关随机排序操作的更多信息，请参阅 Spark 编程指南。托管式扩展仅缩减未充分利用且不包含积极使用的随机排序数据的实例。这种智能扩展可防止随机排序数据意外丢失，从而避免重新尝试任务和重新计算中间数据。
- Amazon EMR 托管扩展首先删除任务节点，然后删除核心节点，直到实现所需的目标缩减容量。集群永远不会缩减到低于托管式扩展策略中的最小限制。
- 在每个节点类型（核心节点或任务节点）中，Amazon EMR 托管扩展首先删除竞价型实例，然后删除按需型实例。
- 对于使用 Amazon EMR 5.x 版本 5.34.0 及更高版本和 6.x 版本 6.4.0 及更高版本启动的集群，Amazon EMR 托管式自动扩缩功能不会缩减正在运行适用于 Apache Spark 的 `ApplicationMaster` 的节点。这样可以最大限度地减少任务失败和重试次数，这有助于提高作业性能并降低成本。要确认集群中哪些节点正在运行 `ApplicationMaster`，请访问 Spark 历史记录服务器，然后在 Spark 应用程序 ID 的执行程序选项卡下筛选驱动程序。

如果集群没有任何负载，Amazon EMR 将取消在之前评估中添加的新实例，并执行缩减操作 如果集群负载过重，Amazon EMR 会取消移除实例，并执行纵向扩展操作。

节点分配注意事项

我们建议您对核心节点使用按需型购买选项，以避免在竞价型实例回收时丢失 HDFS 数据。当更多竞价型实例添加到任务节点时，您可以使用任务节点的竞价型购买选项来降低成本并加快任务执行速度。

节点分配方案

您可以通过设置不同组合的最大、最小、按需限制和最大核心节点参数，根据您的需求创建各种扩展方案。

方案 1: 仅扩展核心节点

要仅扩展核心节点，托管式扩展参数必须满足以下要求：

- 按需限制等于最大边界。
- 最大核心节点等于最大边界。

当未指定按需限制和最大核心节点参数时，这两个参数都默认为最大边界。

以下示例仅演示了扩展核心节点的方案。

集群初始状态	扩展参数	扩展行为
实例组 核心：1 个按需型 任务：1 个按需型和 1 个竞价型	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 20	使用按需型在核心节点上扩展 1 到 20 个实例或实例集单元。在任务节点上没有扩展。
实例机群 核心：1 个按需型 任务：1 个按需型和 1 个竞价型	UnitType: InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20	

集群初始状态	扩展参数	扩展行为
	MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 20	

方案 2：仅扩展任务节点

要仅扩展任务节点，托管扩展参数必须满足以下要求：

- 最大核心节点必须等于最小边界。

以下示例仅演示了扩展任务节点的方案。

集群初始状态	扩展参数	扩展行为
实例组 核心：2 个按需型 任务：1 个竞价型	UnitType: Instances MinimumCapacityUnits : 2 MaximumCapacityUnits : 20 MaximumCoreCapacityUnits : 2	保持核心节点稳定在 2 个，并且仅在 0 到 18 个实例或实例集单位之间扩展任务节点。最小边界和最大边界之间的容量仅添加到任务节点。
实例机群 核心：2 个按需型 任务：1 个竞价型	UnitType：InstanceFleetUnits MinimumCapacityUnits : 2 MaximumCapacityUnits : 20 MaximumCoreCapacityUnits : 2	

方案 3：集群中仅有按需型实例

要仅拥有按需型实例，您的集群和托管扩展参数必须满足以下要求：

- 按需限制等于最大边界。

当未指定按需限制时，参数值默认为最大边界。默认值表示 Amazon EMR 仅扩展按需型实例。

如果最大核心节点小于最大边界，则可以使用最大核心节点参数来分配核心节点和任务节点之间的容量。

要在由实例组组成的集群中启用此方案，集群中的所有节点组必须在初始配置期间使用按需市场类型。

以下示例演示了在整个集群中使用按需型实例的方案。

集群初始状态	扩展参数	扩展行为
实例组 核心：1 个按需型 任务：1 个按需	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 12	使用按需类型在核心节点上扩展 1 到 12 个实例或实例集单位。在任务节点上使用按需扩展剩余容量。不使用竞价型实例进行扩展。
实例机群 核心：1 个按需型 任务：1 个按需	UnitType：InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 12	

方案 4: 集群中只有竞价型实例

要仅拥有竞价型实例，托管扩展参数必须满足以下要求：

- 按需限制设置为 0。

如果最大核心节点小于最大边界，则可以使用最大核心节点参数来分配核心节点和任务节点之间的容量。

要在由实例组组成的集群中启用此方案，核心实例组必须在初始配置期间使用竞价型购买选项。如果任务实例组中没有竞价型实例，则 Amazon EMR 托管扩展会在需要时使用竞价型实例创建任务组。

以下示例演示了在整个集群中使用竞价型实例的方案。

集群初始状态	扩展参数	扩展行为
实例组 核心：1 个 Spot 任务：1 个竞价型	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 0	使用 Spot 在核心节点上扩展 1 到 20 个实例或实例集单位。不使用按需类型进行扩展。
实例机群 核心：1 个 Spot 任务：1 个竞价型	UnitType：InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 0	

方案 5：在核心节点上扩展按需型实例，在任务节点上扩展 Spot 实例

要在核心节点上扩展按需型实例和在任务节点上扩展 Spot 实例，托管扩展参数必须满足以下要求：

- 按需限制必须等于最大核心节点。
- 按需限制和最大核心节点必须小于最大边界。

要在由实例组组成的集群中启用此方案，核心节点组必须使用按需购买选项。

以下示例演示了在核心节点上扩展按需型实例和在任务节点上扩展竞价型实例的方案。

集群初始状态	扩展参数	扩展行为
实例组 核心：1 个按需型 任务：1 个按需型和 1 个竞价型	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 7 MaximumCoreCapacityUnits : 7	由于任务节点上已有 1 个按需单位且按需最大限制为 7，请在核心节点上纵向扩展到 6 个按需单位。。然后在任务节点上纵向扩展到 13 个竞价型单位。
实例机群 核心：1 个按需型 任务：1 个按需型和 1 个竞价型	UnitType：InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 7 MaximumCoreCapacityUnits : 7	

了解托管扩展指标

为集群启用托管扩展时，Amazon EMR 以一分钟的精细程度发布高分辨率数据指标。您可以使用 Amazon EMR 控制台或 Amazon CloudWatch 控制台查看由托管扩展控制的每次大小调整启动和完成的相关事件。CloudWatch 指标对于 Amazon EMR 托管扩展的运行至关重要。我们建议您密切监控 CloudWatch 指标，以确保数据不会丢失。有关如何配置 CloudWatch 告警以检测缺失指标的更多信息，请参阅使用 Amazon CloudWatch 告警。有关使用 Amazon EMR 的 CloudWatch Events 详细信息，请参阅 监控 CloudWatch Events。

以下指标指示集群的当前容量或目标容量。仅当启用了托管扩展时，这些指标才可用。对于由实例集组成的集群，将在 Units 中测量集群容量指标。对于由实例组组成的集群，将根据托管扩展策略中使用的单位类型在 Nodes 或 vCPU 中测量集群容量指标。

指标	描述
• TotalUnitsRequested • TotalNodesRequested • TotalVCPURequested	集群中由托管扩展确定的单位/节点/vCPU 的目标总数。 单位：计数
• TotalUnitsRunning • TotalNodesRunning • TotalVCPURunning	正在运行的集群中可用单位/节点/vCPU 的当前总数。当请求集群大小调整时，将在集群中添加或删除新实例后更新此指标。 单位：计数
• CoreUnitsRequested • CoreNodesRequested • CoreVCPURequested	集群中由托管扩展确定的核心单位/节点//vCPU 的目标数量。 单位：计数
• CoreUnitsRunning • CoreNodesRunning • CoreVCPURunning	集群中正在运行的核心单位/节点/vCPU 的当前数量。 单位：计数
• TaskUnitsRequested • TaskNodesRequested • TaskVCPURequested	集群中由托管扩展确定的任务单位/节点/vCPU 的目标数量。 单位：计数

指标	描述
• TaskUnitsRunning • TaskNodesRunning • TaskVCPURunning	集群中正在运行的任务单位/节点/vCPU 的当前数量。 单位：计数

以下指标指示集群和应用程序的使用状态。这些指标可用于所有 Amazon EMR 功能，但在为集群启用托管扩展时，将以更高的分辨率和一分钟的精细程度发布数据。您可以将以下指标与上表中的集群容量指标相关联，以了解托管扩展决策。

指标	描述
AppsCompleted	提交给 YARN 并且已完成的应用程序数。 使用案例：监控集群进度 单位：计数
AppsPending	提交给 YARN 并且处于挂起状态的应用程序数。 使用案例：监控集群进度 单位：计数
AppsRunning	提交给 YARN 并且正在运行的应用程序数。 使用案例：监控集群进度 单位：计数
ContainerAllocated	ResourceManager 分配的资源容器数。 使用案例：监控集群进度

指标	描述
	单位：计数
ContainerPending	队列中尚未分配的容器数。 使用案例：监控集群进度 单位：计数
ContainerPendingRatio	待处理的容器与已分配的容器的比值 (ContainerPendingRatio = ContainerPending/ContainerAllocated)。如果 ContainerAllocated = 0，则 ContainerPendingRatio = ContainerPending。值 ContainerPendingRatio 表示数字，而不是百分比。此值对基于容器分配行为扩展集群资源很有用。 单位：计数
HDFSUtilization	当前使用的 HDFS 存储的百分率。 使用案例：分析集群性能 单位：百分比
IsIdle	指示集群不再执行任务，但仍处于活动状态并会产生费用。如果没有任何任务和任务处于运行状态，则此指标设置为 1；否则设置为 0。系统每隔五分钟检查一次该值，值为 1 仅表示在检查时集群处于空闲状态，并不表示它整个五分钟内都处于空闲状态。为避免误报，当多次连续五分钟检查获得的值均为 1 时，您应提出警报。例如，当该值在三十分钟或更长时间内都为 1 时，您应提出警报。 使用案例：监控集群性能 单位：布尔值

指标	描述
MemoryAvailableMB	可供分配的内存量。 使用案例：监控集群进度 单位：计数
MRActiveNodes	当前正在运行 MapReduce 任务或作业的节点数。等效于 YARN 指标 mapred.resourcemanager.NoOfActiveNodes 。 使用案例：监控集群进度 单位：计数
YARNMemoryAvailablePercentage	可用于 YARN 的剩余内存的百分比 (YARNMemoryAvailablePercentage = MemoryAvailableMB / MemoryTotalMB)。此值对基于 YARN 内存使用量扩展集群资源很有用。 单位：百分比

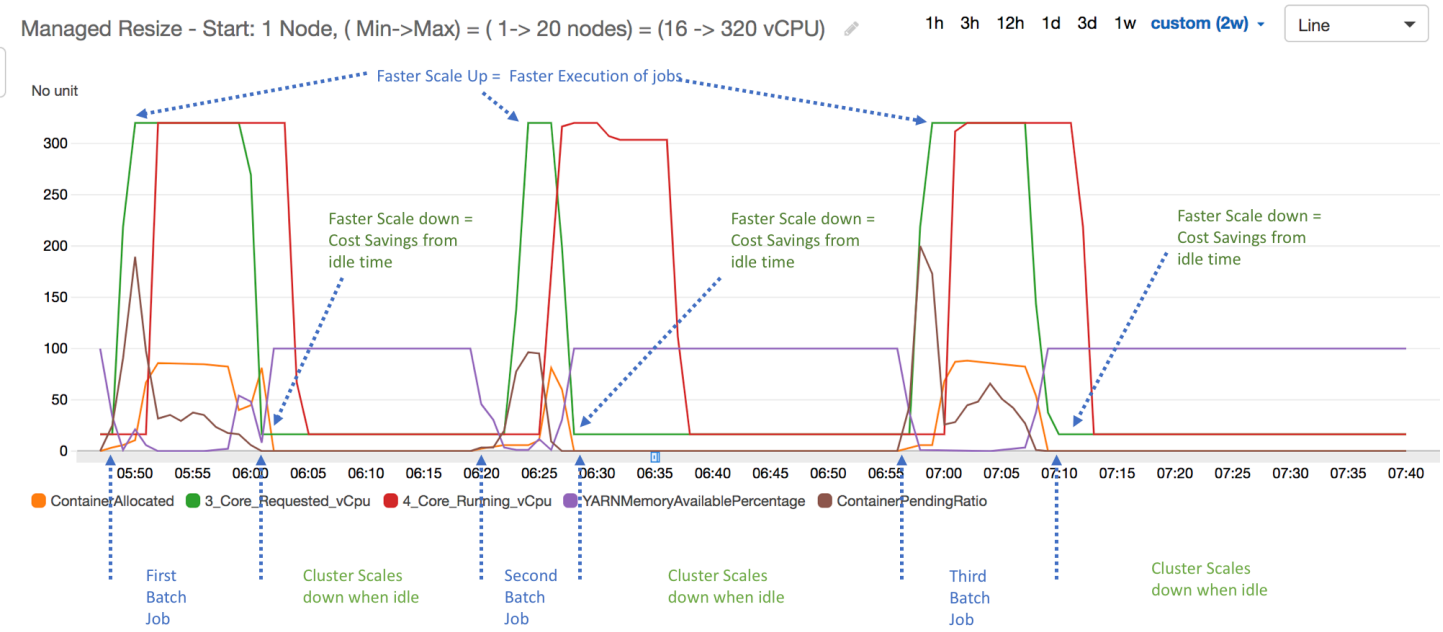
绘制托管扩展指标的图表

您可以绘制指标图表，以便直观地显示集群的工作负载模式和 Amazon EMR 托管扩展做出的相应扩展决策，如以下步骤所示。

在 CloudWatch 控制台中绘制托管扩展指标的图表

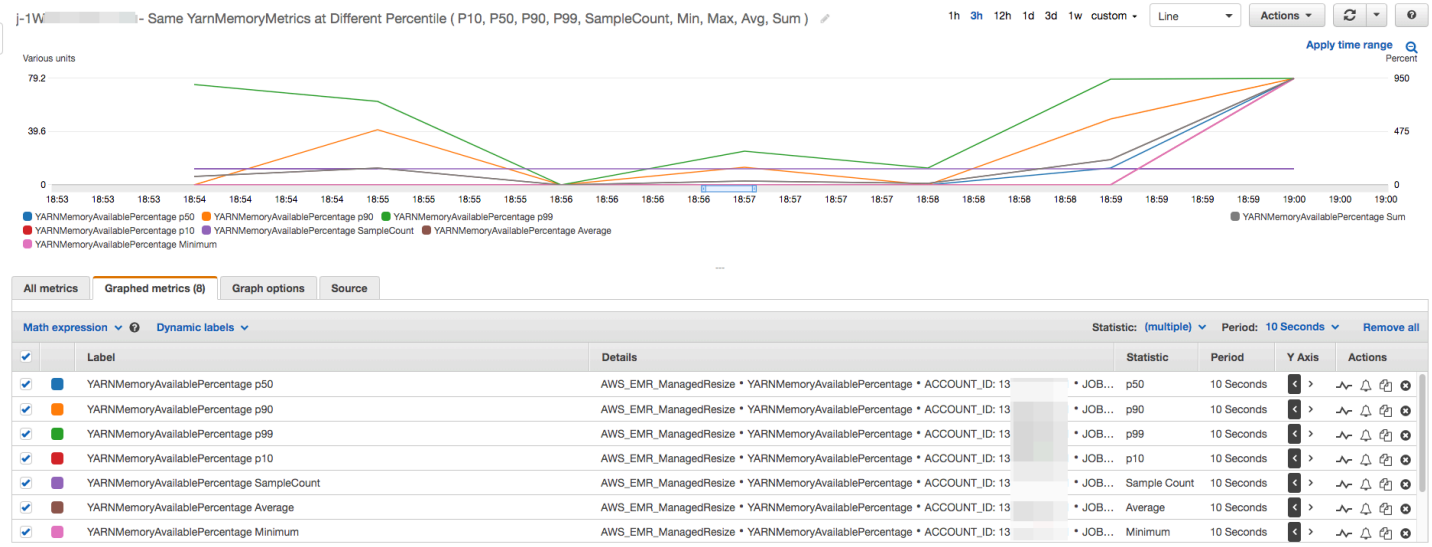
1. 打开 CloudWatch 控制台。
2. 在导航窗格中，选择 Amazon EMR。您可以搜索要监控的集群的集群标识符。
3. 向下滚动到图形的指标。打开指标显示图形。
4. 要为一个或多个指标绘制图表，请选中每个指标旁边的复选框。

以下示例介绍了集群的 Amazon EMR 托管扩展活动。该图形显示三个自动缩减期，这些时段可在工作负载活动性较低时节省成本。



所有集群容量和使用指标均以一分钟的间隔发布。其它统计信息也与每个一分钟数据相关联，这样您就可以绘制各种函数，如 Percentiles、Min、Max、Sum 、Average、SampleCount。

例如，下图以不同的百分位数 P10、P50、P90、P99 绘制同一 YARNMemoryAvailablePercentage 指标以及 Sum、Average、Min、SampleCount。



将自动伸缩与实例组的自定义策略结合使用

利用 Amazon EMR 发行版 4.0 及更高版本中具有自定义策略的自动扩展功能，您可以基于在扩缩策略中指定的 CloudWatch 指标和其他参数，以编程方式横向扩展和横向缩减核心节点和任务节点。实例组配置可使用具有自定义策略的自动扩展，在您使用实例队列时，自动扩展不可用。有关实例组和实例队列的更多信息，请参阅使用实例集或统一实例组创建集群。

> **Note**
>
> 要在 Amazon EMR 中使用具有自定义策略功能的自动伸缩，您必须在创建集群时为 `VisibleToAllUsers` 参数设置 `true`。有关更多信息，请参阅 SetVisibleToAllUsers。

扩展策略是实例组配置的一部分。您可以在初始配置实例组的过程中指定策略，或者通过修改现有集群中的实例组 (甚至在实例组处于活动状态时) 来指定策略。集群中的每个实例组（主实例组除外）均可拥有自己的扩展策略，该策略包含横向扩展和横向缩减规则。可单独配置横向扩展规则和缩减规则，每种规则具有不同的参数。

可使用 Amazon Web Services Management Console、Amazon CLI 或 Amazon EMR API 配置扩展策略。在使用 Amazon CLI 或 Amazon EMR API 时，应以 JSON 格式指定扩展策略。此外，在使用 Amazon CLI 或 Amazon EMR API 时，您可以指定自定义 CloudWatch 指标。自定义指标不可用于使用 Amazon Web Services Management Console 的选项。当您最初使用控制台创建扩展策略时，将预配置适用于许多应用程序的默认策略以帮助您入门。您可以删除或修改默认规则。

即使您可利用自动扩展功能实时调整 EMR 集群容量，您也应考虑基准工作负载需求并规划节点和实例组配置。有关更多信息，请参阅集群配置指南。

> **Note**
>
> 对于大多数工作负载，需要同时设置横向扩展规则和缩减规则才能优化资源使用率。单独设置其中一种规则意味着，您需要在扩展活动后手动调整实例计数的大小。换句话说，这将通过手动重置来设置“单向”自动横向扩展规则或缩减规则。

为自动伸缩功能创建 IAM 角色

Amazon EMR 中的自动伸缩功能需要一个 IAM 角色，此角色有权在触发扩展活动时添加和终止实例。使用适当的角色策略和信任策略配置的默认角色 `EMR_AutoScaling_DefaultRole`

可用于实现此目的。当您首次使用 Amazon Web Services Management Console 创建具有扩展策略的集群时，Amazon EMR 将创建默认角色并附加默认托管策略 `AmazonElasticMapReduceforAutoScalingRole` 来授予权限。

当使用 Amazon CLI 创建具有自动扩展策略的集群时，您必须先确保拥有默认 IAM 角色，或确保拥有自定义 IAM 角色且已附加能够提供适当权限的策略。要创建默认角色，您可以在创建集群前运行 `create-default-roles` 命令。然后，您可以在创建集群时指定 `--auto-scaling-role EMR_AutoScaling_DefaultRole` 选项。或者，您也可以创建一个自定义自动扩展角色，然后在创建集群时指定该角色，例如 `--auto-scaling-role` *`MyEMRAutoScalingRole`*。如果您创建了一个适用于 Amazon EMR 的自定义自动伸缩角色，建议您根据托管式策略制定自定义角色的权限策略。有关更多信息，请参阅为 Amazon EMR 配置 IAM 服务角色对Amazon服务和资源的权限。

了解自动伸缩角色

当横向扩展规则触发实例组的扩展活动时，将根据您的规则向实例组添加 Amazon EC2 实例。只要 Amazon EC2 实例进入 `InService` 状态，应用程序（如 Apache Spark、Apache Hive 和 Presto）就可使用新节点。您也可以设置用于终止实例和删除节点的缩减规则。有关自动扩展 Amazon EC2 实例生命周期的更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 Auto Scaling 生命周期。

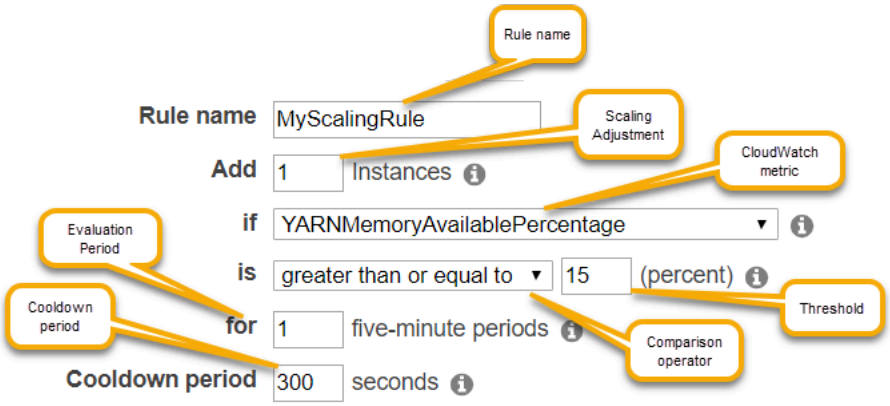
您可以配置集群终止 Amazon EC2 实例的方式，可以选择在用于计费的 Amazon EC2 实例小时边界终止或在任务完成时终止。此设置同时应用于自动扩展和手动调整大小操作。有关此配置的更多信息，请参阅集群缩减。

策略中每个规则的以下参数可确定自动扩展行为。

> **Note**
>
> 此处列出的参数基于Amazon Web Services Management Console上的 Amazon EMR。您使用 Amazon CLI 或 Amazon EMR API 时，系统会提供其它高级配置选项。有关高级选项的详细信息，请参阅《Amazon EMR API 参考》中的 SimpleScalingPolicyConfiguration。

- 最大实例数和最小实例数。Maximum instances (最大实例数) 约束指定了实例组可包含的 Amazon EC2 实例的最大数目，并且此约束应用于所有横向扩展规则。同样，Minimum instances (最小实例数) 约束指定了最小 Amazon EC2 实例数，并且此约束应用于所有横向缩减规则。
- Rule name (规则名称)，此参数在策略中必须是唯一的。
- scaling adjustment (扩展调整)，此参数确定在规则触发的扩展活动期间要添加的 EC2 实例的数目（适用于横向扩展规则）或要终止的 EC2 实例的数目（适用于缩减规则）。

- CloudWatch metric (CloudWatch 指标)，针对警报条件监视此参数。
- comparison operator (比较运算符，用于比较 CloudWatch 指标和 Threshold (阈值) 值并确定触发条件。
- evaluation period (评估周期)，以 5 分钟为增量，在触发扩展活动之前，CloudWatch 指标在这段时间内必须符合触发条件。
- Cooldown period (冷却时间)（以秒为单位），用于确定由规则启动的一个扩展活动和下一个扩展活动开始之间必须经过的时间量，而不管触发扩展活动的规则如何。当实例组完成扩缩活动并达到其扩展后状态时，冷却时间使得可能会触发后续扩展活动的 CloudWatch 指标实现稳定状态。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的 Auto Scaling 冷却时间。



注意事项和限制

- Amazon CloudWatch 指标对于运行 Amazon EMR 自动扩展至关重要。我们建议您密切监控 Amazon CloudWatch 指标，以确保数据不会丢失。有关如何配置 Amazon CloudWatch 告警以检测缺失指标的更多信息，请参阅 使用 Amazon CloudWatch 告警。
- 过度使用 EBS 卷可能会导致托管扩展问题。我们建议您密切监控 EBS 卷的使用情况，以确保 EBS 卷的利用率低于 90%。请参阅 实例存储 以了解有关指定额外 EBS 卷的信息。
- Amazon EMR 发行版 5.18 到 5.28 中的自定义策略自动扩展可能会遇到由于 Amazon CloudWatch 指标中间歇性丢失数据而导致的扩展失败。我们建议您使用最新版本的 Amazon EMR 来改进自动扩展。如果您希望使用 5.18 到 5.28 之间的 Amazon EMR 发行版，也可以联系 Amazon Support 获取补丁。

使用Amazon Web Services Management Console配置自动伸缩

创建集群时，可以使用高级集群配置选项为实例组配置扩展策略。此外，您也可以通过在现有集群的 Hardware (硬件) 设置中修改实例组，为使用中的实例组创建或修改扩展策略。

> **Note**
>
> Amazon EMR 新控制台（https://console.aws.amazon.com/emr）使用托管扩展，而不是自动扩展。要使用自动扩展，请确保您已登录旧控制台，网址为：https://console.aws.amazon.com/elasticmapreduce。

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 如果您正在创建集群，请在 Amazon EMR 控制台中，依次选择 Create Cluster (创建集群)、Go to advanced options (转到高级选项)、Step 1: Software and Steps (步骤 1: 软件和步骤)，然后转至 Step 2: Hardware Configuration (步骤 2: 硬件配置)。

   – 或 -

   如果您修改的是正在运行的集群中的实例组，请从集群列表中选择您的集群，然后展开 Hardware (硬件) 部分。
3. 在 Cluster scaling and provisioning option（集群扩展和预置选项）部分中，选择 Enable cluster scaling（启用集群扩展）。然后，选择 Create a custom automatic scaling policy (创建自定义自动扩展策略)。

   在 Custom automatic scaling policies (自定义自动扩展策略) 表中，单击要配置实例组所在行中显示的铅笔图标。Auto Scaling 规则屏幕将打开。
4. 键入最大实例数，这是您希望实例组在横向扩展后包含的最大实例数；并键入最小实例数，这是您希望实例组在缩减后包含的最小实例数。
5. 单击铅笔图标以编辑规则参数，单击 X 以从策略中删除规则，然后单击添加规则以添加其它规则。
6. 选择本主题前面介绍的规则参数。有关 Amazon EMR 的可用 CloudWatch 指标说明，请参阅《Amazon CloudWatch 用户指南》中的 Amazon EMR 指标和维度。

使用Amazon CLI配置自动伸缩

在创建集群时以及在创建实例组时，可以使用适用于 Amazon EMR 的 Amazon CLI 命令配置自动伸缩。您可以使用速记语法 (可在相关命令中指定内联 JSON 配置)。也可以引用包含配置 JSON 的文件。您也可以将一个自动扩展策略应用于现有实例组并删除以前应用的自动扩展策略。此外，您可以从正在运行的集群中检索扩展策略配置的详细信息。

> **Important**
>
> 在创建具有自动伸缩策略的集群时，您必须使用 `--auto-scaling-role` *`MyAutoScalingRole`* 命令来指定用于自动伸缩的 IAM 角色。默认角色为 *`EMR_AutoScaling_DefaultRole`*，可使用 `create-default-roles` 命令创建它。此角色只能在创建集群时添加且无法添加到现有集群。

有关配置弹性伸缩策略时可用参数的详细描述，请参阅《Amazon EMR API 参考》中的 PutAutoScalingPolicy。

创建具有已应用于实例组的自动伸缩策略的集群

您可在 `aws emr create-cluster` 命令的 `--instance-groups` 选项内指定自动扩展配置。以下示例说明了一个 create-cluster 命令，其中以内联方式提供核心实例组的自动扩展策略。此命令会创建一个与默认横向扩展策略等效的扩展配置，该策略会在使用 Amazon EMR Amazon Web Services Management Console 创建自动扩展策略时显示。为简洁起见，不显示缩减策略。建议不要创建没有向内扩展规则的向外扩展规则。

```
aws emr create-cluster --release-label emr-5.2.0 --service-role
 EMR_DefaultRole --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole
 --auto-scaling-role EMR_AutoScaling_DefaultRole  --instance-groups
 Name=MyMasterIG,InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1
 'Name=MyCoreIG,InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,AutoScalingPolicy
scale-out,Description=Replicates the default scale-out rule in the
 console.,Action={SimpleScalingPolicyConfiguration={AdjustmentType=CHANGE_IN_CAPACITY,ScalingAd
ElasticMapReduce,Period=300,Statistic=AVERAGE,Threshold=15,Unit=PERCENT,Dimensions=[{Key=JobFlo
```

以下命令说明了如何使用命令行将自动扩展策略定义作为实例组配置文件（名为 *`instancegroupconfig.json`*）的一部分提供。

```
aws emr create-cluster --release-label emr-5.2.0 --service-role EMR_DefaultRole --ec2-
attributes InstanceProfile=EMR_EC2_DefaultRole --instance-groups file://your/path/to/
instancegroupconfig.json --auto-scaling-role EMR_AutoScaling_DefaultRole
```

具有配置文件的内容，如下所示：

```
[
```

```
{
  "InstanceCount": 1,
  "Name": "MyMasterIG",
  "InstanceGroupType": "MASTER",
  "InstanceType": "m5.xlarge"
},
{
  "InstanceCount": 2,
  "Name": "MyCoreIG",
  "InstanceGroupType": "CORE",
  "InstanceType": "m5.xlarge",
  "AutoScalingPolicy":
    {
     "Constraints":
      {
       "MinCapacity": 2,
       "MaxCapacity": 10
      },
     "Rules":
     [
      {
       "Name": "Default-scale-out",
       "Description": "Replicates the default scale-out rule in the console for YARN
 memory.",
       "Action":{
        "SimpleScalingPolicyConfiguration":{
          "AdjustmentType": "CHANGE_IN_CAPACITY",
          "ScalingAdjustment": 1,
          "CoolDown": 300
        }
       },
       "Trigger":{
        "CloudWatchAlarmDefinition":{
          "ComparisonOperator": "LESS_THAN",
          "EvaluationPeriods": 1,
          "MetricName": "YARNMemoryAvailablePercentage",
          "Namespace": "AWS/ElasticMapReduce",
          "Period": 300,
          "Threshold": 15,
          "Statistic": "AVERAGE",
          "Unit": "PERCENT",
          "Dimensions":[
             {
               "Key" : "JobFlowId",
```

```
                "Value" : "${emr.clusterId}"
              }
            ]
          }
        }
      }
    ]
  }
}
]
```

将具有自动伸缩策略的实例组添加到集群

您可以将 `--instance-groups` 选项与 `add-instance-groups` 命令结合使用来指定扩展策略配置，所采用的方式与使用 `create-cluster` 时的方式相同。以下示例将 JSON 文件 *instancegroupconfig.json* 引用与实例组配置结合使用。

```
aws emr add-instance-groups --cluster-id j-1EKZ3TYEVF1S2 --instance-groups file://your/path/to/instancegroupconfig.json
```

向现有实例组应用自动伸缩策略或修改已应用的策略

使用 `aws emr put-auto-scaling-policy` 命令将自动扩展策略应用于现有实例组。该实例组必须是使用自动伸缩 IAM 角色的集群的一部分。以下示例使用对指定自动扩展策略配置的 JSON 文件 *autoscaleconfig.json* 的引用。

```
aws emr put-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-id ig-3PLUZBA6WLS07 --auto-scaling-policy file://your/path/to/autoscaleconfig.json
```

下面显示了 `autoscaleconfig.json` 文件的内容，此文件定义上一示例中显示的相同的横向扩展规则。

```
{
          "Constraints": {
                  "MaxCapacity": 10,
                  "MinCapacity": 2
          },
          "Rules": [{
                  "Action": {
                          "SimpleScalingPolicyConfiguration": {
                                  "AdjustmentType": "CHANGE_IN_CAPACITY",
                                  "CoolDown": 300,
```

```
                                "ScalingAdjustment": 1
                        }
                },
                "Description": "Replicates the default scale-out rule in the console
for YARN memory",
                "Name": "Default-scale-out",
                "Trigger": {
                        "CloudWatchAlarmDefinition": {
                                "ComparisonOperator": "LESS_THAN",
                                "Dimensions": [{
                                        "Key": "JobFlowID",
                                        "Value": "${emr.clusterID}"
                                }],
                                "EvaluationPeriods": 1,
                                "MetricName": "YARNMemoryAvailablePercentage",
                                "Namespace": "AWS/ElasticMapReduce",
                                "Period": 300,
                                "Statistic": "AVERAGE",
                                "Threshold": 15,
                                "Unit": "PERCENT"
                        }
                }
        }]
 }
```

从实例组中删除自动伸缩策略

```
aws emr remove-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07
```

检索自动伸缩策略配置

describe-cluster 命令检索 InstanceGroup 块中的策略配置。例如，以下命令检索集群 ID 为 j-1CWOHP4PI30VJ 的集群的配置。

```
aws emr describe-cluster --cluster-id j-1CWOHP4PI30VJ
```

该命令生成以下示例输出。

```
{
    "Cluster": {
        "Configurations": [],
        "Id": "j-1CWOHP4PI30VJ",
        "NormalizedInstanceHours": 48,
        "Name": "Auto Scaling Cluster",
        "ReleaseLabel": "emr-5.2.0",
        "ServiceRole": "EMR_DefaultRole",
        "AutoTerminate": false,
        "TerminationProtected": true,
        "MasterPublicDnsName": "ec2-54-167-31-38.compute-1.amazonaws.com",
        "LogUri": "s3n://aws-logs-232939870606-us-east-1/elasticmapreduce/",
        "Ec2InstanceAttributes": {
            "Ec2KeyName": "performance",
            "AdditionalMasterSecurityGroups": [],
            "AdditionalSlaveSecurityGroups": [],
            "EmrManagedSlaveSecurityGroup": "sg-09fc9362",
            "Ec2AvailabilityZone": "us-east-1d",
            "EmrManagedMasterSecurityGroup": "sg-0bfc9360",
            "IamInstanceProfile": "EMR_EC2_DefaultRole"
        },
        "Applications": [
            {
                "Name": "Hadoop",
                "Version": "2.7.3"
            }
        ],
        "InstanceGroups": [
            {
                "AutoScalingPolicy": {
                    "Status": {
                        "State": "ATTACHED",
                        "StateChangeReason": {
                            "Message": ""
                        }
                    },
                    "Constraints": {
                        "MaxCapacity": 10,
                        "MinCapacity": 2
                    },
                    "Rules": [
                        {
                            "Name": "Default-scale-out",
                            "Trigger": {
```

```
                        "CloudWatchAlarmDefinition": {
                            "MetricName": "YARNMemoryAvailablePercentage",
                            "Unit": "PERCENT",
                            "Namespace": "AWS/ElasticMapReduce",
                            "Threshold": 15,
                            "Dimensions": [
                                {
                                    "Key": "JobFlowId",
                                    "Value": "j-1CWOHP4PI30VJ"
                                }
                            ],
                            "EvaluationPeriods": 1,
                            "Period": 300,
                            "ComparisonOperator": "LESS_THAN",
                            "Statistic": "AVERAGE"
                        }
                    },
                    "Description": "",
                    "Action": {
                        "SimpleScalingPolicyConfiguration": {
                            "CoolDown": 300,
                            "AdjustmentType": "CHANGE_IN_CAPACITY",
                            "ScalingAdjustment": 1
                        }
                    }
                },
                {
                    "Name": "Default-scale-in",
                    "Trigger": {
                        "CloudWatchAlarmDefinition": {
                            "MetricName": "YARNMemoryAvailablePercentage",
                            "Unit": "PERCENT",
                            "Namespace": "AWS/ElasticMapReduce",
                            "Threshold": 75,
                            "Dimensions": [
                                {
                                    "Key": "JobFlowId",
                                    "Value": "j-1CWOHP4PI30VJ"
                                }
                            ],
                            "EvaluationPeriods": 1,
                            "Period": 300,
                            "ComparisonOperator": "GREATER_THAN",
                            "Statistic": "AVERAGE"
```

```
                            }
                        },
                        "Description": "",
                        "Action": {
                            "SimpleScalingPolicyConfiguration": {
                                "CoolDown": 300,
                                "AdjustmentType": "CHANGE_IN_CAPACITY",
                                "ScalingAdjustment": -1
                            }
                        }
                    }
                ]
            },
            "Configurations": [],
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "Name": "Core - 2",
            "ShrinkPolicy": {},
            "Status": {
                "Timeline": {
                    "CreationDateTime": 1479413437.342,
                    "ReadyDateTime": 1479413864.615
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "RunningInstanceCount": 2,
            "Id": "ig-3M16XBE8C3PH1",
            "InstanceGroupType": "CORE",
            "RequestedInstanceCount": 2,
            "EbsBlockDevices": []
        },
        {
            "Configurations": [],
            "Id": "ig-OP62I28NSE8M",
            "InstanceGroupType": "MASTER",
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "Name": "Master - 1",
            "ShrinkPolicy": {},
            "EbsBlockDevices": [],
            "RequestedInstanceCount": 1,
```

```
                "Status": {
                    "Timeline": {
                        "CreationDateTime": 1479413437.342,
                        "ReadyDateTime": 1479413752.088
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "RunningInstanceCount": 1
            }
        ],
        "AutoScalingRole": "EMR_AutoScaling_DefaultRole",
        "Tags": [],
        "BootstrapActions": [],
        "Status": {
            "Timeline": {
                "CreationDateTime": 1479413437.339,
                "ReadyDateTime": 1479413863.666
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Cluster ready after last step completed."
            }
        }
    }
}
```

手动调整正在运行的集群的大小

您可以使用 Amazon Web Services Management Console、Amazon CLI 或 Amazon EMR API，在正在运行的集群中的核心实例组和任务实例组以及实例集中添加和删除实例。如果集群使用的是实例组，您可显式更改实例计数。如果集群使用的是实例队列，您可更改按需实例和竞价型实例的目标单位数。然后，实例队列可添加和删除实例以满足新目标。有关更多信息，请参阅实例集选项。一旦实例可用，应用程序就可使用新预置的 Amazon EC2 实例来托管节点。删除实例后，Amazon EMR 会以不中断任务并防止数据丢失的方式关闭任务。有关更多信息，请参阅在任务完成时终止。

使用控制台调整集群大小

您可以使用 Amazon EMR 控制台调整正在运行的集群的大小。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台更改现有集群的实例数

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择要更新的集群。集群必须正在运行；您无法调整预置集群或已终止集群的大小。
3. 在集群详细信息页面的 Instances（实例）选项卡上，查看 Instance groups（实例组）面板。
4. 要调整现有实例组的大小，请选择要调整大小的核心实例组或任务实例组旁边的单选按钮，然后选择 Resize instance group（调整实例组大小）。为实例组指定新的实例数，然后选择 Resize（调整大小）。

   > **Note**
   >
   > 如果您选择减少正在运行的实例组大小，Amazon EMR 将智能地选择要从实例组中删除的实例，将数据丢失的风险降至最低。要对调整大小操作进行更精细地控制，可以依次选择实例组的 ID、要删除的实例，然后使用 Terminate（终止）选项。有关智能缩减行为的更多信息，请参阅 集群缩减。

5. 如果要取消调整大小操作，您可以选择状态为 Resizing（调整大小）的实例组的单选按钮，然后从列表操作中选择 Stop resize（停止调整大小）。
6. 要向集群添加一个或多个任务实例组以应对不断增加的工作负载，请从操作列表中选择 Add task instance group（添加任务实例组）。选择 Amazon EC2 实例类型，输入任务组的实例数，然后选择 Add task instance group（添加任务实例组）以返回到集群的 Instance groups（实例组）面板。

Old console

使用旧控制台更改现有集群的实例数

1. 从 Cluster List (集群列表) 页面中，选择要调整大小的集群。
2. 在 Cluster Details (集群详细信息) 页面上，选择 Hardware (硬件)。
3. 如果您的集群使用的是实例组，请选择要调整其大小的实例组的 Instance count (实例数量) 列中的 Resize (调整大小)，键入新的实例数量，然后选中绿色复选标记。

   –或–

   如果您的集群使用的是实例集，请在 Provisioned capacity (预置容量) 列中选择 Resize (调整大小)，为 On-Demand units (按需单位数量) 和 Spot units (Spot 单位数量) 键入新值，然后选择 Resize (调整大小)。

当您对节点数进行更改时，实例组的 Status (状态) 会更新。当您请求的更改完成后，Status (状态) 为 Running (正在运行)。

使用 Amazon CLI 调整集群大小

您可以使用 Amazon CLI 调整正在运行的集群的大小。您可以增加或减少任务节点数，并且可以增加正在运行的集群中的核心节点数。还可以使用 Amazon CLI 或 API 关闭核心实例组中的实例。应小心地执行此操作。关闭核心实例组中的实例存在数据丢失的风险，且该实例不会被自动替换。

除了调整核心组和任务组的大小之外，还可使用 Amazon CLI 向正在运行的集群添加一个或多个任务实例组。

通过使用 Amazon CLI 更改实例数来调整集群大小

可以使用带 `InstanceCount` 参数的 Amazon CLI `modify-instance-groups` 子命令向核心组或任务组添加实例，并且可以从任务组删除实例。要向核心或任务组添加实例，请增加 `InstanceCount`。要减少任务组中的实例数，请减少 `InstanceCount`。将任务组的实例计数更改为 0 会删除所有实例，而不删除实例组。

- 要将任务实例组中的实例数从 3 增加到 4，请键入以下命令，将 *`ig-31JXXXXXXBT0`* 替换为实例组 ID。

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-31JXXXXXXBT0,InstanceCount=4
```

要检索 InstanceGroupId，请使用 describe-cluster 子命令。输出是一个名为 Cluster 的 JSON 对象，其中包含每个实例组的 ID。要使用此命令，您需要集群 ID（可以使用 aws emr list-clusters 命令或控制台进行检索）。要检索实例组 ID，请键入以下命令，将 *j-2AXXXXXXGAPLF* 替换为集群 ID。

```
aws emr describe-cluster --cluster-id j-2AXXXXXXGAPLF
```

借助 Amazon CLI，您还可以通过 --modify-instance-groups 子命令终止核心实例组中的实例。

> **Warning**
>
> 指定 EC2InstanceIdsToTerminate 时，请务必谨慎。无论实例上运行的应用程序的状态如何，实例都将立即终止，并且不会自动替换实例。无论集群的 Scale down behavior (缩减行为) 配置如何，都会出现此情况。通过这种方式终止实例将面临数据丢失以及出现不可预测的集群行为的风险。

要终止特定实例，您需要使用实例组 ID（由 aws emr describe-cluster --cluster-id 子命令返回）和实例 ID（由 aws emr list-instances --cluster-id 子命令返回），键入以下命令，将 *ig-6RXXXXXX07SA* 替换为实例组 ID，并将 *i-f9XXXXf2* 替换为实例 ID。

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-6RXXXXXX07SA,EC2InstanceIdsToTerminate=i-f9XXXXf2
```

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

通过使用 Amazon CLI 添加任务实例组来调整集群大小

借助 Amazon CLI，您可以通过 --add-instance-groups 子命令向集群添加 1-48 个任务实例组。任务实例组只能添加到包含主实例组和核心实例组的集群。借助 Amazon CLI，每次使用 --add-instance-groups 子命令时您可以添加多达五个任务实例组。

1. 要向集群添加单个任务实例组，请键入以下命令，将 *j-JXBXXXXXX37R* 替换为集群 ID。

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-groups
 InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
```

2. 要向集群添加多个任务实例组，请键入以下命令，将 *j-JXBXXXXXX37R* 替换为集群 ID。可以在单个命令中添加多达 5 个任务实例组。

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-
groups InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
 InstanceCount=10,InstanceGroupType=task,InstanceType=m5.xlarge
```

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

中断调整

通过使用 Amazon EMR 4.1.0 版或更高版本，您可以在现有大小调整操作执行期间下达新的大小调整指示。此外，您可以停止之前提交的调整请求，或提交新请求以覆盖之前的请求，而不必等待其完成。您还可以从控制台或使用 `ModifyInstanceGroups` API 调用停止现有的大小调整操作，并将当前计数作为目标集群数。

下面的屏幕截图显示了正在调整但可通过选择 Stop (停止) 来停止的任务实例组。

使用 Amazon CLI 中断大小调整操作

您可以在 Amazon CLI 中使用 `modify-instance-groups` 子命令来停止大小调整操作。假定您的实例组包含 6 个实例，您需要将实例增加到 10 个。但随后您决定撤消此请求：

- 初始请求：

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-myInstanceGroupId,InstanceCount=10
```

第二个请求 (停止第一个请求)：

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-myInstanceGroupId,InstanceCount=6
```

> **Note**
>
> 此种处理是异步进行的，因此，在执行后续请求前，您可能会看到实例计数相对于之前的 API 请求有所变化。对于缩减请求，如果节点上有正在进行的工作，则在节点完成其工作前，实例组不会缩减。

“暂停”状态

如果尝试启动新的集群节点时遇到太多错误，实例组就会进入“暂停”状态。例如，如果新节点在执行引导操作时出现故障，实例组会进入 SUSPENDED (暂停) 状态，而不是持续地预置新节点。解决了基础问题后，您可以重置集群实例组上所需数量的节点，然后实例组继续开始分配节点。修改实例组会促使 Amazon EMR 尝试再次预置节点。没有重启或终止任何运行节点。

在 Amazon CLI 中，`list-instances` 子命令返回所有实例及其状态（与 `describe-cluster` 子命令执行的操作一样）。如果 Amazon EMR 检测到实例组故障，则组的状态会更改为 SUSPENDED。

使用 Amazon CLI 重置处于 SUSPENDED（暂停）状态的集群

键入带有 `describe-cluster` 参数的 `--cluster-id` 子命令可查看集群中实例的状态。

- 要查看有关集群中所有实例和实例组的信息，请键入以下命令，将 *j-3KVXXXXXXY7UG* 替换为集群 ID。

  ```
  aws emr describe-cluster --cluster-id j-3KVXXXXXXY7UG
  ```

  输出会显示有关实例组和实例状态的信息：

  ```
  {
      "Cluster": {
          "Status": {
              "Timeline": {
                  "ReadyDateTime": 1413187781.245,
                  "CreationDateTime": 1413187405.356
              },
              "State": "WAITING",
              "StateChangeReason": {
                  "Message": "Waiting after step completed"
              }
          },
          "Ec2InstanceAttributes": {
  ```

```
            "Ec2AvailabilityZone": "us-west-2b"
        },
        "Name": "Development Cluster",
        "Tags": [],
        "TerminationProtected": false,
        "RunningAmiVersion": "3.2.1",
        "NormalizedInstanceHours": 16,
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1413187775.749,
                        "CreationDateTime": 1413187405.357
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "MASTER",
                "InstanceGroupType": "MASTER",
                "InstanceType": "m5.xlarge",
                "Id": "ig-3ETXXXXXXFYV8",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1413187781.301,
                        "CreationDateTime": 1413187405.357
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "CORE",
                "InstanceGroupType": "CORE",
                "InstanceType": "m5.xlarge",
                "Id": "ig-3SUXXXXXQ9ZM",
                "Market": "ON_DEMAND",
```

```
                "RunningInstanceCount": 1
            }
...
}
```

要查看有关特定实例组的信息，请键入带有 `list-instances` 和 `--cluster-id` 参数的 `--instance-group-types` 子命令。您可以查看主实例组、核心实例组和任务实例组的信息。

```
aws emr list-instances --cluster-id j-3KVXXXXXXY7UG --instance-group-types "CORE"
```

使用带有 `modify-instance-groups` 参数的 `--instance-groups` 子命令可重置 SUSPENDED 状态下的集群。实例组 ID 通过 `describe-cluster` 子命令返回。

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-3SUXXXXXXQ9ZM,InstanceCount=3
```

缩减集群大小时的注意事项

如果您选择缩减正在运行的集群的大小，则请考虑以下 Amazon EMR 行为和最佳实践：

- 为了减少对正在进行的作业的影响，Amazon EMR 会智能地选择要删除的实例。有关集群缩减行为更多信息，请参阅《Amazon EMR 管理指南》中的 在任务完成时终止。
- 当您缩减集群的大小时，Amazon EMR 会将数据从其删除的实例复制到剩余的实例。确保组中剩余的实例中有足够的存储容量用于存储此数据。
- Amazon EMR 尝试停用组中实例上的 HDFS。缩减集群大小之前，建议您尽量减少 HDFS 写入 I/O。
- 要在缩减集群大小时进行最精细的控制，您可以在控制台中查看集群并导航到 Instances（实例）选项卡。选择要调整大小的实例组的 ID。然后对要删除的特定实例使用 Terminate（终止）选项。

为预调配容量配置超时

当您使用实例集时，您可以配置预调配超时。预调配超时指示 Amazon EMR 在集群启动或集群扩展操作期间集群超过指定的时间阈值时停止预调配实例容量。以下主题涵盖如何为集群启动和集群纵向扩展操作配置预调配超时。

主题

- 在 Amazon EMR 中为集群启动预调配超时
- 在 Amazon EMR 中自定义集群大小调整的预调配超时期限

在 Amazon EMR 中为集群启动预调配超时

您可以为集群中的每个实例集定义预调配竞价型实例的超时期限。如果 Amazon EMR 无法预调配 Spot 容量，您可以改为选择终止集群或预调配按需容量。如果超时期限在集群大小调整过程中结束，则 Amazon EMR 将取消未预调配的 Spot 请求。未预调配的竞价型实例不会转移到按需容量。

> **Note**
>
> 您无法在旧控制台中自定义预调配超时期限。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

执行以下步骤，自定义使用 Amazon EMR 控制台启动集群的预调配超时期限。

New console

在通过新控制台创建集群时配置预调配超时

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在创建集群页面上，导航到集群配置并选择实例集。
4. 在集群扩展和预调配选项下，为您的核心和任务实例集指定 Spot 大小。
5. 在 Spot 超时配置下，选择在 Spot 超时后终止集群或在 Spot 超时后切换到按需。然后，指定预调配竞价型实例的超时期限。默认值为 1 小时。
6. 选择适用于集群的任何其他选项。
7. 要启动具有已配置超时的集群，选择创建集群。

Amazon CLI

要使用 **`create-cluster`** 命令指定预调配超时

```
aws emr create-cluster \
```

```
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSp
{"OnDemandSpecification":{"AllocationStrategy":"lowest-
price"}},"InstanceTypeConfigs":[{"WeightedCapacity":1,"EbsConfiguration":
{"EbsBlockDeviceConfigs":[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemand
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecifi
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification"
{"AllocationStrategy":"lowest-price"}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemand
 - 2"}]'
```

在 Amazon EMR 中自定义集群大小调整的预调配超时期限

您可以为集群中的每个实例集定义预调配竞价型实例的超时期限。如果 Amazon EMR 无法预调配 Spot 容量，则它将取消调整大小请求并停止尝试预调配额外的 Spot 容量。创建集群时，您可以配置超时。对于正在运行的集群，您可以添加或更新超时。

超时期限到期后，Amazon EMR 会自动将事件发送到 Amazon CloudWatch Events 流。使用 CloudWatch，您可以根据特定的模式创建匹配事件的规则，以便将事件路由到目标来执行操作。例如，您可以将规则配置为发送电子邮件通知。有关如何创建规则的更多信息，请参阅使用 CloudWatch 为 Amazon EMR 事件创建规则。有关不同事件详细信息的更多信息，请参阅实例集状态更改事件。

为集群大小调整预调配超时的示例

使用 Amazon CLI 命令为调整大小指定预调配超时

以下示例使用 `create-cluster` 命令为调整大小添加预调配超时。

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
```

```
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTyp
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}},"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 2"}]'
```

以下示例使用 modify-instance-fleet 命令为调整大小添加预调配超时。

```
aws emr modify-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet '{"InstanceFleetId":"if-XXXXXXXXXXX","ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":60}}}' \
--region us-east-1
```

以下示例使用 add-instance-fleet-command 为调整大小添加预调配超时。

```
aws emr add-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet
 '{"InstanceFleetType":"TASK","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTypeCo
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":35}}}' \
--region us-east-1
```

使用 Amazon CLI 为调整大小和启动指定预调配超时

以下示例使用 create-cluster 命令为调整大小和启动添加预调配超时。

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSpeci
{"OnDemandSpecification":{"AllocationStrategy":"lowest-price"}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}},"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 2"}]'
```

调整预调配超时大小的注意事项

在为实例集配置集群预调配超时时，请考虑以下行为。

- 您可以为竞价型实例和按需实例配置预调配超时。最小预调配超时为 5 分钟。最大预调配超时为 7 天。
- 您只能为使用实例集的 EMR 集群配置预调配超时。您必须分别配置每个核心和任务实例集。
- 创建集群时，您可以配置预调配超时。您可以为正在运行的集群添加超时或更新现有超时。
- 如果您提交多个调整大小操作，则 Amazon EMR 会跟踪每个调整大小操作的预调配超时。例如，将集群的预调配超时设置为 *60* 分钟。然后，在时间 *T1* 时提交调整大小操作 *R1*。在时间 *T2* 时提交第二次调整大小操作 *R2*。R1 的预调配超时在 *T1 + 60 #*#时过期。R2 的预调配超时在 *T2 + 60 #*#时过期。
- 如果您在超时到期之前提交新的纵向扩展大小调整操作，则 Amazon EMR 将继续尝试为您的 EMR 集群预调配容量。

集群缩减

> **Note**
>
> 自 Amazon EMR 发行版 5.10.0 起，不再支持缩减行为选项。因为 Amazon EC2 中引入了按秒计费，Amazon EMR 集群的默认缩减行为现在在任务完成时终止。

对于 Amazon EMR 发行版 5.1.0 到 5.9.1，有两种缩减行为选项：在用于 Amazon EC2 计费的实例小时边界终止，或者在任务完成时终止。从 Amazon EMR 发行版 5.10.0 开始，在实例小时边界处终止的设置已弃用，因为在 Amazon EC2 中引入了按秒计费。我们不建议在提供了此选项的版本中指定在实例小时边界终止。

> **Warning**
>
> 如果您使用 Amazon CLI 发出 `modify-instance-groups` 与 `EC2InstanceIdsToTerminate`，则这些实例将立即终止，而不考虑这些设置，且无论实例上运行的应用程序的状态如何。通过这种方式终止实例将面临数据丢失以及出现不可预测的集群行为的风险。

当指定在任务完成后终止时，Amazon EMR 首先将拒绝列出来自节点的任务，并耗尽这些任务，然后再终止 Amazon EC2 实例。对于指定的任一行为，Amazon EMR 都不会终止核心实例组中的 Amazon EC2 实例（如果这会导致 HDFS 受损）。

在任务完成时终止

Amazon EMR 让您能够在不影响工作负载的情况下缩减集群。在向下调整操作期间，Amazon EMR 能够正常停止核心节点和任务节点上的 YARN、HDFS 及其它守护进程，而不会丢失数据或中断任务。Amazon EMR 仅缩减已完成所分配的工作并处于空闲状态的实例组大小。对于 YARN NodeManager 正常停止，您可以手动调整节点等待停止的时间。

使用 `YARN-site` 配置分类中的属性来设置此时间。要使用 Amazon EMR 发行版 5.12.0 及更高版本，请指定 `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` 属性。要使用早期的 Amazon EMR 发行版，请指定 `YARN.resourcemanager.decommissioning.timeout` 属性。

如果停止超时过后仍有容器或 YARN 应用程序在运行，则系统会强制停止此节点，且由 YARN 在其它节点上重新计划受影响的容器。默认值为 3600 秒 (1 小时)。您可以将此超时设为任意大的值，

以强制自然缩减操作等待更长时间。有关更多信息，请参阅 Apache Hadoop 文档中的 Graceful Decommission of YARN nodes（正常停止 YARN 节点）。

任务节点组

Amazon EMR 将智能地选择没有针对任何步骤或应用程序运行的任务的实例，并首先将这些实例从集群中删除。如果集群中的所有实例均处于使用状态，Amazon EMR 会等待实例上的任务完成，然后将其从集群中删除。默认等待时间为 1 小时。该值可通过 `YARN.resourcemanager.decommissioning.timeout` 设置进行更改。Amazon EMR 将动态使用此新设置。您可以将其设置为任意大数字，以确保 Amazon EMR 在缩减集群大小的同时不会终止任何任务。

核心节点组

在核心节点上，必须将 YARN NodeManager 和 HDFS DataNode 进程守护程序都停止才能缩减实例组。对于 YARN，自然缩减可确保标记为停止的节点仅在没有挂起或未完成的容器或应用程序时才转换到 `DECOMMISSIONED` 状态。如果开始淘汰时，节点上没有任何正在运行的容器，淘汰会立即完成。

对于 HDFS，自然缩减可确保 HDFS 的目标容量大到足以容纳所有现有数据块。如果目标容量不够大，则只停止部分核心实例，以便剩余节点能够处理驻留在 HDFS 中的当前数据。您应确保额外的 HDFS 容量以允许进一步执行淘汰。在尝试减少实例组之前，您还应该尽量减少写入 I/O。过多的写入 I/O 可能会延迟调整大小操作的完成。

另一个限制是默认复制因素：`/etc/hadoop/conf/hdfs-site` 内的 `dfs.replication`。创建集群时，Amazon EMR 会根据集群中的实例数来配置此值：1-3 个实例为 1；4-9 个实例的集群为 2；10 个以上实例的集群为 3。

> **Warning**
>
> 1. 如果单个节点出现故障，则在少于四个节点的集群上将 `dfs.replication` 设置为 1 可能会导致 HDFS 数据丢失。建议您使用具有至少四个核心节点的集群来处理生产工作负载。
> 2. Amazon EMR 不允许集群扩展 `dfs.replication` 下方的核心节点。例如，如果是 `dfs.replication = 2`，则最小核心节点数为 2。
> 3. 当您使用托管扩缩、自动扩缩或选择手动调整集群大小时，建议您将设置 `dfs.replication` 为 2 或更高。

自然缩减不允许您将核心节点减少到低于 HDFS 复制因子。这是为了允许 HDFS 因副本不足而关闭文件。要避开此限制，请降低重复因子并重启 NameNode 进程守护程序。

配置 Amazon EMR 缩减行为

> **Note**
>
> Amazon EMR 发行版 5.10.0 及更高版本不再支持在实例小时终止缩减行为选项。以下缩减行为选项仅在 Amazon EMR 控制台发行版 5.1.0 到 5.9.1 中显示。

在创建集群时，您可以使用Amazon Web Services Management Console、Amazon CLI 或 Amazon EMR API 配置缩减行为。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的信息。

New console

使用新控制台配置缩减行为

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群），然后选择 Create cluster（创建集群）。
3. 在 Cluster scaling and provisioning option（集群扩展和预置选项）部分中，找到 Cluster termination（集群终止）并选择手动终止集群或者让 Amazon EMR 在指定的空闲时间后终止集群。您也可以选择打开针对漏洞或错误的终止保护。
4. 选择适用于集群的任何其他选项。
5. 要启动集群，选择 Create cluster（创建集群）。

Old console

使用旧控制台配置缩减行为

1. 打开 Amazon EMR 控制台（https://console.amazonaws.cn/elasticmapreduce）。
2. 选择创建集群。转到 Advanced options（高级选项），然后在 Step 1: Software and Steps（步骤 1：软件和步骤）和 Step 2: Hardware（步骤 2：硬件）中选择配置设置。

3. 在 Step 3: General Cluster Settings（步骤 3：常规集群设置）中，选择首选的缩减行为。完成剩余配置并创建集群。

Amazon CLI

使用 Amazon CLI 配置缩减行为

- 使用 `--scale-down-behavior` 选项指定 `TERMINATE_AT_INSTANCE_HOUR` 或 `TERMINATE_AT_TASK_COMPLETION`。

终止集群

此部分介绍终止集群的方法。有关启用终止保护和自动终止集群的信息，请参阅控制集群终止。您可以在 STARTING、RUNNING 或 WAITING 状态下终止集群。WAITING 状态下的集群必须终止，否则会无限期地运行，给您的账户产生费用。您可以终止无法离开 STARTING 状态或不能完成步骤的集群。

如果要终止已开启终止保护的集群，则必须首先禁用终止保护，然后才能终止集群。可以使用控制台或 Amazon CLI 来终止集群，或使用 `TerminateJobFlows` API 以编程方式终止集群。

根据集群配置，集群完全终止并释放已分配的资源 (如 EC2 实例) 可能需要 5 至 20 分钟。

> **Note**
>
> 您无法重新启动已终止的集群，但可以克隆已终止的集群，将其配置重用于新集群。有关更多信息，请参阅使用控制台克隆集群。

> **Important**
>
> Amazon EMR 使用 Amazon EMR 服务角色 和 `AWSServiceRoleForEMRCleanup` 角色来清理账户中不再使用的集群资源，例如 Amazon EC2 实例。您必须在角色策略包含删除或终止资源所需的操作。否则，Amazon EMR 将无法执行这些清理操作，并且集群上剩余的未使用资源可能会产生费用。

使用控制台终止集群

您可以使用 Amazon EMR 控制台终止一个或多个集群。在控制台中终止集群的步骤根据是否打开了终止保护而有所不同。要终止已保护的集群，您必须首先禁用终止保护。

New console

使用新控制台终止集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 选择 Clusters（集群），然后选择要终止的集群。
3. 在 Actions（操作）下拉菜单下，选择 Terminate cluster（终止集群）以打开 Terminate cluster（终止集群）提示。
4. 出现提示时，选择 Terminate（终止）。根据集群配置，终止可能需要 5 至 10 分钟。有关如何终止 Amazon EMR 集群的更多信息，请参阅 终止集群。

Old console

使用旧控制台，在终止保护关闭的情况下终止集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择要终止的集群。您可以选择多个集群，然后同时终止它们。
3. 选择 Terminate (终止)。
4. 系统提示时，选择 Terminate (终止)。

   Amazon EMR 终止集群中的实例并停止保存日志数据。

使用旧控制台，在终止保护打开的情况下终止集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 在 Cluster List (集群列表) 页面中，选择要终止的集群。您可以选择多个集群，然后同时终止它们。
3. 选择 Terminate (终止)。

4. 系统提示时，选择 Change (更改) 以关闭终止保护。如果您选择了多个集群，请选择 Turn off all (关闭所有) 以同时禁用所有集群的终止保护。
5. 在 Terminate clusters (终止集群) 对话框中，对于 Termination Protection (终止保护)，选择 Off (关闭)，然后单击复选标记进行确认。
6. 单击 Terminate (终止)。

   Amazon EMR 终止集群中的实例并停止保存日志数据。

使用 Amazon CLI 终止集群

使用 Amazon CLI 终止不受保护的集群

要使用 Amazon CLI 终止不受保护的集群，请使用带 --cluster-ids 参数的 `terminate-clusters` 子命令。

- 键入以下命令以终止单个集群，将 *j-3KVXXXXXXX7UG* 替换为您的集群 ID。

  ```
  aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG
  ```

  要终止多个集群，请键入以下命令，并将 *j-3KVXXXXXXX7UG* 和 *j-WJ2XXXXXX8EU* 替换为您的集群 ID。

  ```
  aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG j-WJ2XXXXXX8EU
  ```

  有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

使用 Amazon CLI 终止受保护的集群

要使用 Amazon CLI 终止受保护的集群，首先请使用带 `--no-termination-protected` 参数的 `modify-cluster-attributes` 子命令禁用终止保护。然后使用带 `terminate-clusters` 参数的 `--cluster-ids` 子命令终止该集群。

1. 键入以下命令以禁用终止保护，将 *j-3KVTXXXXXX7UG* 替换为您的集群 ID。

   ```
   aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
   ```

2. 要终止集群，请键入以下命令并将 *j-3KVXXXXXXX7UG* 替换为您的集群 ID。

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG
```

要终止多个集群，请键入以下命令，并将 *j-3KVXXXXXXX7UG* 和 *j-WJ2XXXXXX8EU* 替换为您的集群 ID。

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG j-WJ2XXXXXX8EU
```

有关在 Amazon CLI 中使用 Amazon EMR 命令的更多信息，请参阅https://docs.amazonaws.cn/cli/latest/reference/emr。

使用 API 终止集群

`TerminateJobFlows` 操作结束步骤处理，从 Amazon EC2 向 Amazon S3（如果已配置）上载任何日志数据，并终止 Hadoop 集群。如果您在 `KeepJobAliveWhenNoSteps` 请求中将 `False` 设置为 `RunJobFlows`，集群也可以自动终止。

您可以使用此操作终止单个集群或按照集群 ID 终止一系列集群。

有关 `TerminateJobFlows` 所独有的输入参数的更多信息，请参阅 TerminateJobFlows。有关该请求中通用参数的更多信息，请参见通用请求参数。

使用控制台克隆集群

您可以使用 Amazon EMR 控制台克隆集群，从而复制原始集群的配置，用作新集群的基础。

> **Note**
>
> 我们重新设计了 Amazon EMR 控制台，以便其易于使用。您可以在新控制台中克隆使用横向缩减的集群，但如果您想对其进行手动扩展或使用托管扩展，则只能创建新集群。请参阅 控制台中的新增功能，以了解有关新旧控制台体验差异的更多信息。

New console

使用新控制台克隆集群

1. 登录 Amazon Web Services Management Console 并打开 Amazon EMR 控制台，网址为 https://console.aws.amazon.com/emr。
2. 在左侧导航窗格中的 EMR on EC2 下，选择 Clusters（集群）。
3. 从集群列表中克隆集群

   a. 使用搜索和筛选条件选项，在列表视图中查找要克隆的集群。

   b. 选中要克隆的集群所在行左侧的复选框。

   c. Clone（克隆）选项将在列表视图顶部显示。选择 Clone（克隆）以启动克隆过程。如果集群已配置步骤，请选择 Include steps（包含步骤），如果要克隆这些步骤以及其他集群配置，请选择 Continue（继续）。

   d. 查看从克隆集群复制的新集群设置。请根据需要调整设置。如果对新集群的配置满意，请选择 Create cluster（创建集群）以启动新集群。
4. 从集群详细信息页面克隆集群

   a. 要导航到要克隆的集群的详细信息页面，请从集群列表视图中选择其 Cluster ID（集群 ID）。

   b. 在集群详细信息页面顶部，从 Actions（操作）菜单中选择 Clone cluster（克隆集群）以启动克隆过程。如果集群已配置步骤，请选择 Include steps（包含步骤），如果要克隆这些步骤以及其他集群配置，请选择 Continue（继续）。

   c. 查看从克隆集群复制的新集群设置。请根据需要调整设置。如果对新集群的配置满意，请选择 Create cluster（创建集群）以启动新集群。

Old console

使用旧控制台克隆集群

1. 导航到 Amazon EMR 新控制台，然后从侧面导航栏中选择切换到旧控制台。有关切换到旧控制台后预期情况的更多信息，请参阅 Using the old console。
2. 选择创建集群。
3. 从 Cluster List (集群列表) 页面，单击要克隆的集群。
4. 在 Cluster Details (集群详细信息) 页面顶部，单击 Clone (克隆)。

在对话框中，选择 Yes (是) 以在克隆的集群中包括原始集群中的步骤。选择 No (否) 以克隆原始集群的配置而不包括任何步骤。

> **Note**
>
> 对于使用 AMI 3.1.1 及更高版本 (Hadoop 2.x) 或 AMI 2.4.8 及更高版本 (Hadoop 1.x) 创建的集群，如果克隆集群并包括步骤，则所有系统步骤 (如配置 Hive) 都会随用户提交的步骤一起克隆 (总共最多 1000 个)。无法克隆不再出现在控制台步骤历史记录中的任何较旧步骤。对于较早的 AMI，只能克隆 256 个步骤 (包括系统步骤)。有关更多信息，请参阅向集群提交工作。

5. Create Cluster (创建集群) 页面显示原始集群的配置副本。检查配置，进行任何必要的更改，然后单击 Create Cluster (创建集群)。

通过 Amazon Data Pipeline 自动处理定期集群

Amazon Data Pipeline 服务可以自动处理数据的移动和转换。您可以使用它安排将输入数据移入 Amazon S3 的时间，以及安排启动集群处理这些数据的时间。例如，在拥有 Web 服务器记录流量日志的情况下可以考虑此项服务。如果您每周需要运行一次集群分析流量数据，可以使用 Amazon Data Pipeline 安排这些集群。Amazon Data Pipeline 是一种数据驱动型工作流，因此，一个任务（启动集群）可以依赖另一个任务（将输入数据移入 Amazon S3）。此外，它还拥有强健的重试功能。

有关 Amazon Data Pipeline 的更多信息，请参阅《Amazon Data Pipeline 开发人员指南》，尤其是有关 Amazon EMR 的教程：

- 教程: 启动 Amazon EMR 任务流
- 入门: 使用 Amazon Data Pipeline、Amazon EMR 和 Hive 处理 Web 日志
- 教程：使用 Amazon Data Pipeline进行 Amazon DynamoDB 导入和导出

集群问题排查

EMR 集群的运行生态系统十分复杂，包括开源软件、自定义应用程序代码和 Amazon Web Services。当以上任何环节出现问题时，集群都可能失败，或者花费比您预期更长的时间才能完成。以下主题可帮助您识别集群的问题以及如何修复它们。

主题

- 提供了哪些可用于故障排查的工具？
- 查看并重新启动 Amazon EMR 和应用程序进程（进程守护程序）
- Amazon EMR 中的常见错误
- 失败集群的故障排查
- 速度很慢的集群的故障排查
- 对 Lake Formation 集群进行故障排查

在开发新的 Hadoop 应用程序时，我们建议您启用调试并处理一小部分但具有代表性的数据子集以测试应用程序。您可能还希望逐步运行应用程序以分别测试每个步骤。有关更多信息，请参阅配置集群日志记录和调试和步骤 5：分步测试集群。

提供了哪些可用于故障排查的工具？

要识别和修复集群错误，您可以使用本页上描述的工具。启动集群时，您可能需要初始化一些工具。默认情况下，每个集群都可以使用其他工具。

主题

- 查看 EMR 集群详细信息
- 查看 EMR 集群错误详细信息
- 运行脚本和配置 Amazon EMR 流程
- 查看日志文件
- 监控 EMR 集群的性能

查看 EMR 集群详细信息

您可以使用 Amazon Web Services Management Console、Amazon CLI 或 EMR API 检索有关 EMR 集群和作业执行的详细信息。有关使用 Amazon Web Services Management Console和 Amazon CLI 的更多信息，请参阅查看集群状态和详细信息。

Amazon EMR 控制台详细信息窗格

在 Amazon EMR 控制台上的集群列表中，您可查看有关您的账户和 Amazon Web Services 区域 中每个集群的状态的概要信息。此列表显示您过去两个月启动的所有活动和已终止的集群。从 Clusters (集群) 列表中，您可选择集群的 Name (名称) 以查看集群详细信息。此信息已进行分类，可轻松导航。

集群详细信息页面中提供的应用程序用户界面对排查集群的问题较为有用。它提供了 YARN 应用程序的状态，对于部分应用程序（如 Spark 应用程序），您可深入了解各种指标和方面（如作业、阶段和执行程序）。有关更多信息，请参阅查看应用程序历史记录。。此功能仅在 Amazon EMR 5.8.0 及更高版本中提供。

Amazon EMR 命令行界面

您可以使用 `--describe` 参数从 Amazon CLI 中查找有关某一集群的详细信息。

Amazon EMR API

您可以使用 `DescribeJobFlows` 操作从 API 中查找有关某一集群的详细信息。

查看 EMR 集群错误详细信息

当 EMR 集群因错误而终止时，`DescribeCluster` 和 `ListClusters` API 会返回错误代码和错误消息。对于选定的集群错误，`ErrorDetail` 数据数组可以帮助您排查失败原因。

有关包含 `ErrorDetail` 数据的错误代码列表，请参阅 包含 ErrorDetail 信息的错误代码。

> **Note**
>
> 为确保向您提供最相关的信息，我们会不断完善错误消息。我们不建议您解析 `ErrorMessage` 中的文本，因为此文本可能会发生变化。

运行脚本和配置 Amazon EMR 流程

作为故障排除过程的一部分，您可能会发现在集群上运行自定义脚本或查看和配置集群进程很有帮助。

查看并重启应用程序进程

查看集群上正在运行的进程以诊断潜在问题可能会很有帮助。您可以通过连接到集群的主节点 (master node) 停止并重新启动集群进程。有关更多信息，请参阅查看并重新启动 Amazon EMR 和应用程序进程（进程守护程序）。

在不连接 SSH 的情况下运行命令和脚本

要在集群上运行命令或脚本，可以使用`command-runner.jar`或者`script-runner.jar`工具，而无需与主节点 (master node) 建立 SSH 连接。有关更多信息，请参阅在 Amazon EMR 集群上运行命令和脚本。

查看日志文件

Amazon EMR 和 Hadoop 都会在集群运行时生成日志文件。根据在启动集群时指定的配置，您可以从多种不同工具中访问这些日志文件。有关更多信息，请参阅配置集群日志记录和调试。

主节点 上的日志文件

每个集群都会将日志文件发布到主节点上的 /mnt/var/log/ 目录。仅在集群处于运行状态时才能获取这些日志文件。

归档到 Amazon S3 的日志文件

如果您启动集群并指定 Amazon S3 日志路径，则集群会每隔 5 分钟将存储在主节点上 /mnt/var/log/ 中的日志文件复制到 Amazon S3。这样可确保即使在终止集群后也可以访问日志文件。因为文件的存档间隔是 5 分钟，所以可能无法获取突然终止集群的最后几分钟的状况信息。

监控 EMR 集群的性能

Amazon EMR 提供了多种用于监控集群性能的工具。

Hadoop Web 界面

每个集群都会将一组 Web 界面 (包含有关集群的信息) 发布到主节点上。您可以使用 SSH 隧道连接主节点上的这些网页，从而对它们进行访问。有关更多信息，请参阅查看 Amazon EMR 集群上托管的 Web 界面。

CloudWatch metrics（CloudWatch 指标）

每个集群都会向 CloudWatch 报告指标。CloudWatch 是一种跟踪指标的 Web 服务，您可以使用此服务对这些指标设置警报。有关更多信息，请参阅使用 CloudWatch 指标监控 Amazon EMR 指标。

查看并重新启动 Amazon EMR 和应用程序进程（进程守护程序）

当您排除集群故障时，您可能想要列出正在运行的进程。您可能还想停止或重新启动进程。例如，在更改配置之后或在分析日志文件和错误消息从而发现特定进程的问题之后，重启程序。

在集群上运行的进程有两种类型：Amazon EMR 进程（例如实例控制器和日志推送程序），以及与集群上安装的应用程序关联的进程（例如 hadoop-hdfs-namenode 和 hadoop-yarn-resourcemanager）。

要在集群上直接使用进程，您首先要连接到主节点 (master node)。有关更多信息，请参阅连接到集群。

查看正在运行的进程

用来查看集群上正在运行进程的方法取决于您使用的 Amazon EMR 版本。

EMR 5.30.0 and 6.0.0 and later

Example ：列出所有运行中的进程

以下示例使用`systemctl`并指定`--type`查看所有进程。

```
systemctl --type=service
```

Example ：列出特定的流程

以下示例列出含有`hadoop`名称的所有进程。

```
systemctl --type=service | grep -i hadoop
```

输出示例：

```
hadoop-hdfs-namenode.service           loaded active running Hadoop namenode
hadoop-httpfs.service                  loaded active running Hadoop httpfs
hadoop-kms.service                     loaded active running Hadoop kms
hadoop-mapreduce-historyserver.service loaded active running Hadoop historyserver
hadoop-state-pusher.service            loaded active running Daemon process that
processes and serves EMR metrics data.
hadoop-yarn-proxyserver.service        loaded active running Hadoop proxyserver
hadoop-yarn-resourcemanager.service    loaded active running Hadoop resourcemanager
hadoop-yarn-timelineserver.service     loaded active running Hadoop timelineserver
```

Example ：查看特定进程的详细状态报告

以下示例显示了hadoop-hdfs-namenode服务的详细状态报告。

```
sudo systemctl status hadoop-hdfs-namenode
```

输出示例：

```
hadoop-hdfs-namenode.service - Hadoop namenode
   Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Wed 2021-08-18 21:01:46 UTC; 26min ago
 Main PID: 9733 (java)
    Tasks: 0
   Memory: 1.1M
   CGroup: /system.slice/hadoop-hdfs-namenode.service
           # 9733 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server -
XX:OnOutOfMemoryError=kill -9 %p ...

Aug 18 21:01:37 ip-172-31-20-123 systemd[1]: Starting Hadoop namenode...
Aug 18 21:01:37 ip-172-31-20-123 su[9715]: (to hdfs) root on none
Aug 18 21:01:37 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: starting namenode,
 logging to /var/log/hadoop-hdfs/ha...out
Aug 18 21:01:46 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: Started Hadoop
 namenode:[  OK  ]
Aug 18 21:01:46 ip-172-31-20-123 systemd[1]: Started Hadoop namenode.
Hint: Some lines were ellipsized, use -l to show in full.
```

EMR 4.x - 5.29.0

Example ：列出所有运行中的进程

以下示例列出了所有正在运行的进程。

```
initctl list
```

EMR 2.x - 3.x

Example ：列出所有运行中的进程

以下示例列出了所有正在运行的进程。

```
ls /etc/init.d/
```

停止并重新启动进程

在确定哪些进程正在运行后，您可以根据需要将其停止然后重新启动。

EMR 5.30.0 and 6.0.0 and later

Example：停止进程

以下示例停止hadoop-hdfs-namenode进程。

```
sudo systemctl stop hadoop-hdfs-namenode
```

您可以查询status以验证该进程是否已停止。

```
sudo systemctl status hadoop-hdfs-namenode
```

输出示例：

```
hadoop-hdfs-namenode.service - Hadoop namenode
  Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
  Active: failed (Result: exit-code) since Wed 2021-08-18 21:37:50 UTC; 8s ago
Main PID: 9733 (code=exited, status=143)
```

Example：启动进程

以下示例启动hadoop-hdfs-namenode进程。

```
sudo systemctl start hadoop-hdfs-namenode
```

您可以查询状态以确认该进程是否正在运行。

```
sudo systemctl status hadoop-hdfs-namenode
```

输出示例：

```
hadoop-hdfs-namenode.service - Hadoop namenode
   Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Wed 2021-08-18 21:38:24 UTC; 2s ago
```

```
 Process: 13748 ExecStart=/etc/init.d/hadoop-hdfs-namenode start (code=exited,
status=0/SUCCESS)
Main PID: 13800 (java)
   Tasks: 0
  Memory: 1.1M
  CGroup: /system.slice/hadoop-hdfs-namenode.service
          # 13800 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server
-XX:OnOutOfMemoryError=kill -9 %p...
```

EMR 4.x - 5.29.0

Example : 停止正在运行的进程

以下示例停止hadoop-hdfs-namenode服务。

```
sudo stop hadoop-hdfs-namenode
```

Example ：重新启动已停止的进程

以下示例重新启动hadoop-hdfs-namenode服务。您必须使用start命令而不是restart。

```
sudo start hadoop-hdfs-namenode
```

Example ：查看进程状态

以下获取hadoop-hdfs-namenode的状态。您可以使用status命令以确认该过程是否已停止或启动。

```
sudo status hadoop-hdfs-namenode
```

EMR 2.x - 3.x

Example ：停止应用程序进程

以下示例停止hadoop-hdfs-namenode服务，该服务与集群上安装的 Amazon EMR 版本相关联。

```
sudo /etc/init.d/hadoop-hdfs-namenode stop
```

Example ：重新启动应用程序进程

以下示例命令重新启动hadoop-hdfs-namenode进程：

```
sudo /etc/init.d/hadoop-hdfs-namenode start
```

Example：停止 Amazon EMR 过程

以下示例停止与集群上的 Amazon EMR 版本无关的进程，例如：实例控制器。

```
sudo /sbin/stop instance-controller
```

Example ：重新启动 Amazon EMR 进程

以下示例重新启动与集群上的 Amazon EMR 版本无关的进程，例如：实例控制器。

```
sudo /sbin/start instance-controller
```

> **Note**
>
> `/sbin/start, stop` 和 `restart` 命令是 `/sbin/intictl` 的符号链接。有关 `initctl` 的更多信息，请在命令提示中键入 `man initctl`，以查看 initctl man 页面。

Amazon EMR 中的常见错误

有时，集群会出现故障或数据处理速度很慢。以下各节列出了一些常见的集群问题以及如何修复这些问题的建议。

主题

- 包含 ErrorDetail 信息的错误代码
- 资源错误
- 输入与输出错误
- 权限错误
- Hive 集群错误
- VPC 错误
- 流式集群错误
- 自定义 JAR 集群错误
- Amazon GovCloud (US-West) 错误

- 查找缺少的集群

包含 ErrorDetail 信息的错误代码

当 EMR 集群因错误而终止时，`DescribeCluster` 和 `ListClusters` API 会返回错误代码和错误消息。对于部分集群错误，`ErrorDetail` 数据数组可以帮助您排查失败原因。

包含 `ErrorDetail` 数组的错误提供了以下详细信息：

`ErrorCode`

可用于编程访问的唯一错误代码。

`ErrorData`

键值对中的标识符列表，可用于编程或手动查找。有关错误代码包含的 `ErrorData` 值的描述，请参阅错误代码的疑难解答页面。

`ErrorMessage`

错误描述以及指向 Amazon EMR 文档中更多信息的链接。

> **Note**
>
> 我们不建议您解析 `ErrorMessage` 中的文本，因为此文本可能会发生变化。

按类别划分的错误代码

- 引导失败错误代码
- 内部错误代码
- 验证失败错误代码

引导失败错误代码

以下各节提供引导失败错误代码的问题排查信息。

主题

- BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE
- BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY

- BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY

BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE

概述

当集群因 BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE 错误而终止时，主实例中的引导操作已失败。有关引导操作的更多信息，请参阅 创建引导操作以安装其它软件。

解析

要解决此错误，请查看 API 错误中返回的详细信息、修改您的引导操作脚本，然后使用更新的引导操作创建新集群。

要排查失败的 EMR 集群问题，请参阅 `DescribeCluster` 和 `ListClusters` API 返回的 `ErrorDetail` 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。`ErrorDetail` 中的 `ErrorData` 数组返回此错误代码的以下信息：

`primary-instance-id`

引导操作失败的主实例的 ID。

`bootstrap-action`

失败的引导操作的序号。`bootstrap-action` 值为 `1` 的脚本是在实例上运行的第一个引导操作。

`return-code`

失败的引导操作返回码。

`amazon-s3-path`

失败的引导操作的 Amazon S3 位置。

`public-doc`

错误代码文档的公有 URL。

完成步骤

执行以下步骤以确定和修复引导操作错误的根本原因。然后启动新集群。

1. 查看 Amazon S3 中的引导操作日志文件，确定失败的根本原因。要了解有关如何查看 Amazon EMR 日志的更多信息，请参阅 查看日志文件。

2. 如果您在创建实例时开启了集群日志，请参阅 stdout 日志以了解更多信息。您可以在以下 Amazon S3 位置找到引导操作的 stdout 日志：

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

   有关集群的更多信息，请参阅 配置集群日志记录和调试。
3. 要确定引导操作失败，请查看 stdout 日志中的异常以及 ErrorData 中的return-code 值。
4. 使用上一步中的发现来修改您的引导操作，使其避免异常或在异常发生时可以正常地处理异常。
5. 使用更新的引导操作启动新集群。

BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY

概述

当主实例无法从您指定的 Amazon S3 位置下载引导操作脚本时，集群终止并出现 BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY 错误。典型的原因包括：

- 引导操作脚本文件不在指定的 Amazon S3 位置。
- 集群上的 Amazon EC2 实例的服务角色（也称为 Amazon EMR 的 EC2 实例配置文件）无权访问引导操作脚本所在的 Amazon S3 存储桶。有关服务角色的更多信息，请参阅集群 EC2 实例（EC2 实例配置文件）的服务角色。

有关引导操作的更多信息，请参阅 创建引导操作以安装其它软件。

解析

要解决此错误，请确保您的主实例具有对引导操作脚本的适当访问权限。

要排查失败的 EMR 集群问题，请参阅 DescribeCluster 和 ListClusters API 返回的 ErrorDetail 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。ErrorDetail 中的 ErrorData 数组返回此错误代码的以下信息：

primary-instance-id

引导操作失败的主实例的 ID。

bootstrap-action

失败的引导操作的序号。bootstrap-action 值为 1 的脚本是在实例上运行的第一个引导操作。

amazon-s3-path

失败的引导操作的 Amazon S3 位置。

public-doc

错误代码文档的公有 URL。

完成步骤

执行以下步骤以确定和修复引导操作错误的根本原因。然后启动新集群。

故障排除步骤

1. 使用 ErrorData 数组中的 amazon-s3-path 值在 Amazon S3 中查找相关的引导操作脚本。
2. 如果您在创建实例时开启了集群日志，请参阅 stdout 日志以了解更多信息。您可以在以下 Amazon S3 位置找到引导操作的 stdout 日志：

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

   有关集群的更多信息，请参阅 配置集群日志记录和调试。
3. 要确定引导操作失败，请查看 stdout 日志中的异常以及 ErrorData 中的return-code 值。
4. 使用上一步中的发现来修改您的引导操作，使其避免异常或在异常发生时可以正常地处理异常。
5. 使用更新的引导操作启动新集群。

BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY

概述

该 BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY 错误表示主实例找不到该实例刚刚从指定的 Amazon S3 存储桶下载的引导操作脚本。

解析

要解决此错误，请确保您的主实例具有对引导操作脚本的适当访问权限。

要排查失败的 EMR 集群问题，请参阅 DescribeCluster 和 ListClusters API 返回的 ErrorDetail 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。ErrorDetail 中的 ErrorData 数组返回此错误代码的以下信息：

primary-instance-id

引导操作失败的主实例的 ID。

bootstrap-action

失败的引导操作的序号。bootstrap-action 值为 1 的脚本是在实例上运行的第一个引导操作。

amazon-s3-path

失败的引导操作的 Amazon S3 位置。

public-doc

错误代码文档的公有 URL。

完成步骤

执行以下步骤以确定和修复引导操作错误的根本原因。然后启动新集群。

1. 要在 Amazon S3 中查找相关的引导操作脚本，请使用 ErrorData 数组中的 amazon-s3-path 值。
2. 查看 Amazon S3 中的引导操作日志文件，确定失败的根本原因。要了解有关如何查看 Amazon EMR 日志的更多信息，请参阅 查看日志文件。

   > **Note**
   >
   > 如果您没有为集群开启日志，则必须使用相同的配置和引导操作创建一个新集群。要确保集群日志已开启，请参阅 配置集群日志记录和调试。

3. 查看 stdout 日志以了解您的引导操作，并确认没有自定义流程可以删除主实例上 /emr/instance-controller/lib/bootstrap-actions 文件夹中的文件。您可以在以下 Amazon S3 位置找到引导操作的 stdout 日志：

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

4. 使用更新的引导操作启动新集群。

内部错误代码

以下各节提供内部错误代码的问题排查信息。

主题

- INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ
- INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY
- INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY

INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ

概述

当所选可用区没有足够的容量来满足您的 Amazon EC2 实例类型请求时，集群会终止并出现 `INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ` 错误。为集群选择的子网决定可用区。有关 Amazon EMR 的子网的更多信息，请参阅 配置联网。

解析

要解决此错误，请修改您的实例类型配置并使用更新后的请求创建一个新集群。

要排查失败的 EMR 集群问题，请参阅 `DescribeCluster` 和 `ListClusters` API 返回的 `ErrorDetail` 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。`ErrorDetail` 中的 `ErrorData` 数组返回此错误代码的以下信息：

`instance-type`

容量不足的实例类型。

`availability-zone`

您的子网解析到的可用区。

`public-doc`

错误代码文档的公有 URL。

完成步骤

执行以下步骤以确定和修复集群配置错误的根本原因。

- 查看集群配置的最佳实践。请参阅《Amazon EMR Management Guide》中的 集群配置的最佳实践。
- 排查启动问题并检查您的配置。请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的排查启动问题。

- 使用更新的集群配置启动新集群。

INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY

概述

当 Amazon EMR 由于实例不可用等于或低于您的最高竞价型价格而无法满足您对主节点的竞价型实例请求时，集群会终止并出现 INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY 错误，。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的竞价型实例。

解析

要解决此错误，请为集群指定在目标价格范围内的实例类型，或者提高相同实例类型的价格限制。

要排查失败的 EMR 集群问题，请参阅 `DescribeCluster` 和 `ListClusters` API 返回的 `ErrorDetail` 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。`ErrorDetail` 中的 `ErrorData` 数组返回此错误代码的以下信息：

`primary-instance-id`

失败集群的主实例的 ID。

`instance-type`

容量不足的实例类型。

`availability-zone`

您的子网所在的可用区。

`public-doc`

错误代码文档的公有 URL。

完成步骤

执行以下步骤排查集群配置策略的问题，然后启动新集群：

1. 查看 Amazon EC2 竞价型实例的最佳实践，并查看您的集群配置策略。有关更多信息，请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的 EC2竞价型的最佳实践和 集群配置的最佳实践。
2. 请修改您的实例类型配置或可用区，并使用更新后的请求创建一个新集群。

3. 如果问题仍然存在，请为主实例使用按需容量。

INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY

概述

当没有足够的容量来满足主节点的竞价型实例请求时，集群会终止并出现 INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY 错误。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的竞价型实例。

解析

要解决此错误，请为集群指定在目标价格范围内的实例类型，或者提高相同实例类型的价格限制。

要排查失败的 EMR 集群问题，请参阅 DescribeCluster 和 ListClusters API 返回的 ErrorDetail 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。ErrorDetail 中的 ErrorData 数组返回此错误代码的以下信息：

primary-instance-id

失败集群的主实例的 ID。

instance-type

容量不足的实例类型。

availability-zone

您的子网解析到的可用区。

public-doc

错误代码文档的公有 URL。

完成步骤

执行以下步骤排查集群配置策略的问题，然后启动新集群：

1. 查看 Amazon EC2 竞价型实例的最佳实践，并查看您的集群配置策略。有关更多信息，请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的 EC2竞价型的最佳实践和 集群配置的最佳实践。
2. 请修改您的实例类型配置，并使用更新后的请求创建一个新集群。

3. 如果问题仍然存在，请为主实例使用按需容量。

验证失败错误代码

以下各节提供验证失败错误代码的问题排查信息。

主题

- VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC
- VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC
- VALIDATION_ERROR_INVALID_SSH_KEY_NAME
- VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC

概述

当您的集群和您为集群引用的子网属于不同的虚拟私有云（VPC）时，集群将终止并出现 `VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC` 错误。您可以使用 Amazon EMR 在 VPC 中跨子网启动具有实例集配置的集群。有关实例集的更多信息，请参阅《Amazon EMR Management Guide》中的 配置实例集。

解析

要解决此错误，请使用与集群属于同一 VPC 的子网。

要排查失败的 EMR 集群问题，请参阅 `DescribeCluster` 和 `ListClusters` API 返回的 `ErrorDetail` 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。`ErrorDetail` 中的 `ErrorData` 数组返回此错误代码的以下信息：

`vpc`

对于每个 子网:VPC 对，返回该子网所属的 VPC ID。

`subnet`

对于每个 子网:VPC 对，返回该子网的 ID。

`public-doc`

错误代码文档的公有 URL。

完成步骤

执行以下步骤来识别和修复错误：

1. 查看 ErrorData 数组中列出的子网 ID，并确认它们属于您要启动 EMR 集群的 VPC。
2. 修改您的子网配置。您可以使用以下方法之一查找 VPC 中所有可用的公有和私有子网。
   - 导航到 Amazon VPC 控制台。选择子网，然后列出驻留在您集群 Amazon Web Services 区域中的所有子网。要仅查找公有子网或私有子网，请应用自动分配公有 IPv4 地址筛选条件。要在您的集群使用的 VPC 中查找和选择子网，请使用按 VPC 筛选选项。有关如何创建子网的更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的创建子网。
   - 使用 Amazon CLI 查找您的集群使用的 VPC 中所有可用的公有和私有子网。有关更多信息，请参阅 describe-subnets API。要在 VPC 中创建新的子网，请参阅 create-subnet API。
3. 启动一个新集群，其子网与集群属于同一 VPC。

VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC

概述

当您的集群和您分配给集群的安全组属于不同的虚拟私有云（VPC）时，集群将终止并出现 VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC 错误。有关安全组的更多信息，请参阅 指定 Amazon EMR 托管的安全组和其它安全组 和 使用安全组控制网络流量。

解析

要解决此错误，请使用与集群属于同一 VPC 的安全组。

要排查失败的 EMR 集群问题，请参阅 DescribeCluster 和 ListClusters API 返回的 ErrorDetail 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。ErrorDetail 中的 ErrorData 数组返回此错误代码的以下信息：

vpc

对于每个安全组:VPC 对，返回该安全组所属的 VPC 的 ID。

security-group

对于每个安全组:VPC 对，返回该安全组的 ID。

public-doc

错误代码文档的公有 URL。

完成步骤

执行以下步骤来识别和修复错误：

1. 查看 ErrorData 数组中列出的安全组 ID，并确认它们属于您要启动 EMR 集群中的 VPC。
2. 导航到 Amazon VPC 控制台。选择安全组以列出所选区域内的所有安全组。在与您的集群相同的 VPC 中找到安全组，然后修改您的安全组配置。
3. 启动一个新集群，其安全组与集群属于同一 VPC。

VALIDATION_ERROR_INVALID_SSH_KEY_NAME

概述

当您使用无法通过 SSH 连接到主实例的 Amazon EC2 密钥对时，集群会终止并出现 VALIDATION_ERROR_INVALID_SSH_KEY_NAME 错误。密钥对名称可能不正确，或者请求的 Amazon Web Services 区域 中可能不存在密钥对。有关密钥对的更多信息，请参阅《Amazon EC2 用户指南（适用于 Linux 实例）》中的 Amazon EC2 密钥对和 Linux 实例。

解析

要解决此错误，请使用有效的 SSH 密钥对名称创建一个新集群。

要排查失败的 EMR 集群问题，请参阅 DescribeCluster 和 ListClusters API 返回的 ErrorDetail 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。ErrorDetail 中的 ErrorData 数组返回此错误代码的以下信息：

ssh-key

您在创建集群时提供的 SSH 密钥对名称。

public-doc

错误代码文档的公有 URL。

完成步骤

执行以下步骤来识别和修复错误：

1. 检查您的 *keypair*.pem 文件，并确认它是否与您在 Amazon EMR 控制台中看到的 SSH 密钥的名称相匹配。

2. 导航到 Amazon EC2 控制台。确认您使用的 SSH 密钥名称在您使用的集群的 Amazon Web Services 区域 中可用。您可以在 Amazon Web Services Management Console 顶部的账户 ID 旁边找到您的 Amazon Web Services 区域。
3. 使用有效 SSH 密钥名称启动新集群。

VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

概述

如果您的集群的 Amazon Web Services 区域 和可用区不支持一个或多个实例组的指定实例类型，则集群会终止并出现 `VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED` 错误。Amazon EMR 可能支持区域内一个可用区中的某个实例类型，但不支持另一个可用区中的实例类型。您为集群选择的子网决定了区域内的可用区。如需了解 Amazon EMR 支持的实例类型和区域，请参阅 支持的实例类型。

解析

要解决此错误，请在您请求集群的区域和可用区中指定 Amazon EMR 支持的集群实例类型。

要排查失败的 EMR 集群问题，请参阅 `DescribeCluster` 和 `ListClusters` API 返回的 `ErrorDetail` 信息。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。`ErrorDetail` 中的 `ErrorData` 数组返回此错误代码的以下信息：

`instance-types`

不支持的实例类型列表。

`availability-zones`

您的子网解析到的可用区列表。

`public-doc`

错误代码文档的公有 URL。

完成步骤

执行以下步骤来识别和修复错误：

1. 使用 Amazon CLI 检索可用区中的可用实例类型。为此，您可以使用 `ec2 describe-instance-type-offerings` 命令按位置（Amazon Web Services 区域 或可用区）筛选可用的实例类型。例如，以下命令返回指定可用区 *`us-east-2a`* 中提供的实例类型。

```
aws ec2 describe-instance-type-offerings --location-type "availability-
zone" --filters Name=location,Values=us-east-2a --region us-east-2 --query
 "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

要了解有关如何发现可用实例类型的更多信息，请参阅查找 Amazon EC2 实例类型。

2. 确定与集群的相同区域和可用区中可用的实例类型后，请选择以下解决方案之一继续：

   a. 在您所选的实例类型可用且受 Amazon EMR 支持的可用区中，创建新集群并为该集群选择子网。

   b. 在与集群失败的相同区域和 Amazon EC2 子网中创建一个新集群，但使用在该位置受 Amazon EMR 支持的实例类型。

如需了解 Amazon EMR 支持的实例类型和区域，请参阅 支持的实例类型。要比较实例类型的功能，请参阅 Amazon EC2 实例类型。

资源错误

以下错误通常是由集群上受限制的资源导致的。

主题

- 集群因 NO_SLAVE_LEFT 和核心节点 FAILED_BY_MASTER 而终止
- 无法复制块，只能设法复制到零节点。
- EC2 QUOTA EXCEEDED
- 太多获取失败
- 文件只能复制到 0 个节点而不是 1 个节点
- 拒绝列出的节点
- 节流错误
- 实例类型不受支持
- EC2 容量不足

集群因 NO_SLAVE_LEFT 和核心节点 FAILED_BY_MASTER 而终止

通常，发生这一现象是因为禁用了终止保护，而且所有核心节点都超过了 `yarn-site` 配置分类（对应于 `yarn-site.xml` 文件）中最大利用率阈值指定的磁盘存储容量。默认情况下，该值为 90%。当核心节点的磁盘利用率超出此利用率阈值时，则 YARN NodeManager 运行状况服务会将节点报告为

UNHEALTHY。如果此节点处于此状态，Amazon EMR 会将此节点列入拒绝名单且不会向其分配 YARN 容器。如果此节点保持不正常状态 45 分钟，则 Amazon EMR 会把要终止的相关 Amazon EC2 实例标记为 FAILED_BY_MASTER。当与核心节点关联的所有 Amazon EC2 实例都被标记为终止时，集群将因没有资源用于执行任务而以 NO_SLAVE_LEFT 状态终止。

如果一个核心节点超过磁盘利用率，则可能会导致连锁反应。如果单个节点因 HDFS 而超过磁盘利用率阈值，那么其它节点很可能同样会接近阈值。第一个节点超过磁盘利用率阈值时，Amazon EMR 会将其列入拒绝名单。这会增加剩余节点的磁盘利用率负担，因为节点开始在自身之间复制其在已列入拒绝名单的节点上丢失的 HDFS 数据。每个节点随后都会以相同的方式进入 UNHEALTHY 状态，并且集群最终会终止。

最佳实践和建议

为集群硬件配置充足的存储容量

创建集群时，请确保有足够的核心节点并且每个节点都为 HDFS 提供足够的实例存储空间和 EBS 存储卷。有关更多信息，请参阅计算集群的必需的 HDFS 容量。您也可以手动或使用 Auto Scaling 功能，将核心实例添加到现有实例组。新实例与实例组中的其它实例具有相同的存储配置。有关更多信息，请参阅使用集群扩展。

启用终止保护

启用终止保护。这样一来，如果核心节点被列入拒绝名单，您可以使用 SSH 连接到关联的 Amazon EC2 实例进行故障排查并恢复数据。如果启用终止保护，请注意，Amazon EMR 不会将该 Amazon EC2 实例替换为新实例。有关更多信息，请参阅使用终止保护。

为 MRUnhealthyNodes CloudWatch 指标创建警报

此指标会报告其中报告了 UNHEALTHY 状态的节点的数量。此指标等同于 YARN 指标 mapred.resourcemanager.NoOfUnhealthyNodes。您可以为此警报设置通知，在节点维持不正常状态到达 45 分钟计时之前提醒您。有关更多信息，请参阅使用 CloudWatch 指标监控 Amazon EMR 指标。

使用 yarn-site 调整设置

下面的设置可以根据您的应用程序要求进行调整。例如，您可能需要通过提高 yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage 的值，来提高节点报告为 UNHEALTHY 的磁盘利用率阈值。

在使用 yarn-site 配置分类创建集群时，您可以设置这些值。有关更多信息，请参阅《Amazon EMR 版本指南》中的配置应用程序。您还可以使用 SSH 连接到与核心节点关联的 Amazon EC2 实

例，然后使用文本编辑器在 /etc/hadoop/conf.empty/yarn-site.xml 添加值。完成更改后，您必须按照以下方式重新启动 hadoop-yarn-nodemanager。

> **Important**
>
> 当您重启 NodeManager 服务时，活动的 YARN 容器会被终止，除非在您创建集群时使用 yarn-site 配置分类将 yarn.nodemanager.recovery.enabled 设为 true。您还必须使用 yarn.nodemanager.recovery.dir 属性指定存储容器状态的目录。

```
sudo /sbin/stop hadoop-yarn-nodemanager
sudo /sbin/start hadoop-yarn-nodemanager
```

有关当前 yarn-site 属性和默认值的更多信息，请参阅 Apache Hadoop 文档中的 YARN 默认设置。

属性	默认值	描述
yarn.nodemanager.disk-health-checker.interval-ms	120000	磁盘运行状况检查程序运行的频率（以秒为单位）。
yarn.nodemanager.disk-health-checker.min-healthy-disks	0.25	NodeManager 的运行状况必须正常才能启动新容器的最小磁盘数量。这对应于 yarn.nodemanager.local-dirs（默认情况下，Amazon EMR 中的 /mnt/yarn ）和 yarn.nodemanager.log-dirs（默认情况下，为 /var/log/hadoop-yarn/containers ，与 Amazon EMR 中的 mnt/var/log/hadoop-yarn/containers 符号链接）。
yarn.nodemanager.disk-health-checker.max-disk-utilizat	90.0	磁盘标记为故障后允许的磁盘空间利用率的最大百分比。值范围是 0.0 到 100.0。如果值大于或等于 100，则

属性	默认值	描述
ion-per-disk-perce ntage		NodeManager 会检查整个磁盘。这适用于 yarn-node manager.local-dirs 和 yarn.nodemanager.l og-dirs 。
yarn.nodemanager.d isk-health-checker .min-free-space-pe r-disk-mb	0	磁盘上可供使用的最小空间。这适用于 yarn-node manager.local-dirs 和 yarn.nodemanager.l og-dirs 。

无法复制块，只能设法复制到零节点。

错误：“Cannot replicate block, only managed to replicate to zero nodes.” 通常在集群没有足够的 HDFS 存储时发生。当您在集群中生成的数据多于 HDFS 中可以存储的数据时，会发生此错误。只有在集群运行时您才会看到此错误，因为当作业结束时，它会释放所使用的 HDFS 空间。

向集群提供的 HDFS 空间量取决于用作核心节点的 Amazon EC2 实例的数量和类型。任务节点不用于 HDFS 存储。每个 Amazon EC2 实例上的所有磁盘空间（包括连接的 EBS 存储卷）均可供 HDFS 使用。有关每种 EC2 实例类型的本地存储量的更多信息，请参阅适用于《Linux 实例的 Amazon EC2 用户指南》中的实例类型和系列。

另一个会影响可用 HDFS 空间量的因素是重复因子，即为冗余数据存储在 HDFS 中的每个数据块的副本数量。重复因子会随着集群中的节点数量而增加：对于拥有 10 个或更多节点的集群，每个数据块有 3 个副本；对于拥有 4 个至 9 个节点的集群，每个数据块有 2 个副本；对于拥有 3 个或更少节点的集群，每个数据块有 1 个副本 (无冗余)。用 HDFS 的总可用空间除以重复因子。有些情况下，例如将节点数量从 9 增加到 10 时，重复因子的增大实际上会导致 HDFS 的可用空间量下降。

例如，拥有 10 个 m1.large 类型核心节点的集群会向 HDFS 提供 2833 GB 的空间 ((10 节点 X 每个节点 850 GB) /重复因子 3)。

如果您的集群占用空间超过了为 HDFS 提供的空间，您可以向集群添加额外核心节点；或使用数据压缩创建更多 HDFS 空间。如果您的集群可以停止和重启，您可以考虑使用更大 Amazon EC2 实例类型的核心节点。您还可以考虑调整重复因子。然而，应当注意，减小重复因子会降低 HDFS 数据的冗余程度以及集群从丢失或损坏的 HDFS 数据块中恢复的能力。

EC2 QUOTA EXCEEDED

如果您收到一条 EC2 QUOTA EXCEEDED 消息，可能存在多种原因。根据配置差异，以前的集群可能需要 5-20 分钟的时间才能终止并释放分配的资源。如果您在尝试启动集群时收到 EC2 QUOTA EXCEEDED 错误，原因可能是最近终止的集群中的资源尚未释放。此消息也可能是由于将实例组或实例队列的大小调整为大于账户的当前实例配额的目标大小。这可能通过自动扩展手动或自动发生。

请考虑使用以下选项来解决此问题：

- 按照《Amazon Web Services 一般参考》中的 Amazon Service Quotas 中的说明请求提高服务限额。对于某些 API，设置 CloudWatch 事件可能比增加限制更好。有关更多信息，请参阅 何时在 CloudWatch 中设置 EMR 事件。
- 如果一个或多个正在运行的集群未达到容量，请为正在运行的集群调整实例组的大小或降低实例队列上的目标容量。
- 创建具有较少 EC2 实例或更少目标容量的集群。

太多获取失败

步骤或任务尝试日志中的“Too many fetch-failures”或“Error reading task output”错误消息表示，正在运行的任务取决于另一任务的输出。当 reduce 任务在队列中等待执行和需要一个或多个映射任务的输出以及系统尚未提供输出时，通常会发生此错误。

有几个原因会导致输出可能不可用：

- 前提任务仍在处理中。这通常是一个映射任务。
- 如果数据存储在不同的实例上，那么数据不可用的原因可能是网络连接较差。
- 如果 HDFS 用于检索输出，那么 HDFS 可能会出现问题。

此错误的最常见原因是之前的任务仍在处理中。如果在 reduce 任务最初尝试运行时发生了此错误，那么极可能是上述原因造成的。您可以通过检查返回此错误的集群步骤的 syslog 日志，确定情况是否如此。如果 syslog 显示了 map-reduce 任务的执行进度，这表明在还有尚未完成的映射任务的同时 reduce 阶段已经开始。

要在日志中寻找的是映射进度百分比，此值会上升到 100% 然后回落到更低值。当映射百分比是 100% 时，这并不意味着所有的映射任务都已完成。它仅仅意味着 Hadoop 正在执行所有映射任务。如果此值降到 100% 以下，则意味着映射任务已失败，然后 Hadoop 可能会根据配置试图重新计划任务。如果日志中映射百分比保持在 100%，请查看 CloudWatch 指标，特别是 RunningMapTasks，以检查映射任务是否仍在处理中。您还可以使用主节点上的 Hadoop Web 界面找到此信息。

如果您看到此问题，那么可以尝试以下几件事情：

- 使 reduce 阶段在开始前等待更长时间。您可以通过将 Hadoop 配置设置 mapred.reduce.slowstart.completed.maps 更改为更长时间，实现上述目的。有关更多信息，请参阅创建引导操作以安装其它软件。
- 使 Reducer 计数与集群的总 Reducer 容量相符。您可以通过调整作业的 Hadoop 配置设置 mapred.reduce.tasks，实现上述目的。
- 使用组合器类代码，将需要提取的输出量降至最低。
- 进行检查，以确保 Amazon EC2 服务不存在影响集群网络性能的问题。您可以使用 Service Health Dashboard 实现上述目的。
- 查看集群中实例的 CPU 和内存资源，确保数据处理不会淹没节点的资源。有关更多信息，请参阅配置集群硬件和联网。
- 检查 Amazon EMR 集群中使用的 Amazon Machine Image（AMI）的版本。如果版本是 2.3.0 到 2.4.4 (含)。请更新为最高版本。指定范围中的 AMI 版本所用的 Jetty 版本可能无法从映射阶段提供输出。当 Reducer 无法从映射阶段获取输出时，会发生提取错误。

  Jetty 是一种开源 HTTP 服务器，用于 Hadoop 集群中的机器间通信。

文件只能复制到 0 个节点而不是 1 个节点

当将文件写入 HDFS 时，会复制到多个核心节点。当您看到此错误时，就意味着 NameNode 守护程序中没有将数据写入 HDFS 的可用 DataNode 实例。换句话说，没有发生数据块复制。此错误可能是由多种问题导致的：

- HDFS 文件系统可能已经用尽了空间。这是最有可能的原因。
- 运行作业时，DataNode 实例可能并不可用。
- 可能已经阻止 DataNode 实例与主节点进行通信。
- 核心实例组中的实例可能并不可用。
- 权限可能会缺失。例如，JobTracker 守护程序可能不具有创建作业跟踪程序信息的权限。
- DataNode 实例的预留空间设置可能并不足够。通过检查 dfs.datanode.du.reserved 配置设置，确定情况是否如此。

要检查此问题是否是由 HDFS 用尽磁盘空间导致的，请查看 CloudWatch 中的 `HDFSUtilization` 指标。如果此值过高，您可以将其它核心节点添加到此集群。如果有一个您认为可能会用尽 HDFS 磁盘空间的集群，那么您可以在 CloudWatch 中设置一个警报，以便当 `HDFSUtilization` 的值超出某一

水平时提醒您。有关更多信息，请参阅手动调整正在运行的集群的大小和使用 CloudWatch 指标监控 Amazon EMR 指标。

如果问题不是 HDFS 用尽了空间，请检查 DataNode 日志、NameNode 日志和网络连接，以找出可能阻止 HDFS 复制数据的其它问题。有关更多信息，请参阅 查看日志文件。

拒绝列出的节点

NodeManager 守护程序负责在核心节点和任务节点上启动和管理容器。该容器由主节点上运行的 ResourceManager 守护程序分配给 NodeManager 守护程序。ResourceManager 通过检测信号监控 NodeManager 节点。

在下列情况下，ResourceManager 守护程序会将 NodeManager 节点列入黑名单，将此节点从可用于处理任务的节点库中删除：

- 如果 NodeManager 在过去的 10 分钟（60 万毫秒）内尚未向 ResourceManager 守护进程发送检测信号。可以使用 `yarn.nm.liveness-monitor.expiry-interval-ms` 配置设置来配置此时间段。有关更改 Yarn 配置设置的更多信息，请参阅《Amazon EMR 版本指南》中的配置应用程序。
- NodeManager 检查由 `yarn.nodemanager.local-dirs` 和 `yarn.nodemanager.log-dirs` 确定的磁盘的运行状况。该检查包括权限和可用磁盘空间 (< 90%)。如果磁盘未通过检查，则 NodeManager 将停止使用该特定磁盘，但仍将节点状态报告为正常。如果大量磁盘未通过检查，则向 ResourceManager 报告的节点状态为不正常，并且不会向节点分配新容器。

主应用程序也可将 NodeManager 节点列入拒绝名单（如果该节点包含 3 个以上的失败任务）。您可以使用 `mapreduce.job.maxtaskfailures.per.tracker` 配置参数将此值更改为更高的值。您可以更改的其它配置设置可控制将任务标记为失败之前的尝试次数：用于映射任务的 `mapreduce.map.max.attempts` 和用于缩减任务的 `mapreduce.reduce.maxattempts`。有关更改配置设置的更多信息，请参阅《Amazon EMR 版本指南》中的配置应用程序。

节流错误

当 Amazon EMR 因另一项服务限制活动而无法完成请求时，会出现“Throttled from *Amazon EC2* while launching cluster”和“Failed to provision instances due to throttling from *Amazon EC2*”两项错误。Amazon EC2 是限制错误的最常见来源，但其它服务可能是限制错误的原因。Amazon服务限制适用于每个区域以提高性能，节流错误表明您已超出该区域中账户的服务限制。

可能的原因

Amazon EC2 节流错误的最常见来源是启动了大量的集群实例，因此超出了 EC2 实例的服务限制。集群实例可能会因以下原因启动：

- 创建了新集群。
- 手动调整了集群。有关更多信息，请参阅手动调整正在运行的集群的大小。
- 作为自动扩展规则的结果，集群中的实例组添加实例（向外扩展）。有关更多信息，请参阅了解自动伸缩角色。
- 集群中的实例队列添加实例以满足增加的目标容量。有关更多信息，请参阅配置实例集。

向 Amazon EC2 发出的 API 请求的频率或类型也可能导致节流错误。有关 Amazon EC2 如何限制 API 请求的更多信息，请参阅《Amazon EC2 API 参考》中的查询 API 请求率。

Solutions

考虑以下解决方案：

- 按照《Amazon Web Services 一般参考》中的 Amazon Service Quotas 中的说明请求提高服务限额。对于某些 API，设置 CloudWatch 事件可能比增加限制更好。有关更多信息，请参阅 何时在 CloudWatch 中设置 EMR 事件。
- 如果您的集群以相同的时间表启动（例如，在整点开始时），请考虑错开启动时间。
- 如果您有针对峰值需求而定义大小的集群，并且您定期拥有实例容量，请考虑指定自动扩展以按需添加和删除实例。通过这种方式，可以更高效地使用实例，并且根据需求配置文件，可以跨账户在给定时间请求更少的实例。有关更多信息，请参阅将自动伸缩与实例组的自定义策略结合使用 。

实例类型不受支持

如果您创建了一个集群，并且它失败并显示错误消息“The requested instance type *InstanceType* is not supported in the requested Availability Zone”，则表示您创建了集群并在创建集群的区域和可用区域中为一个或多个实例组指定了不被 Amazon EMR 支持的实例类型。Amazon EMR 可能在区域内的一个可用区域中支持某个实例类型，而在另一个可用区中则不支持。为集群选择的子网决定区域内的可用区。

解决方案

使用 Amazon CLI 确定可用区中的可用实例类型

- 使用带 `--dry-run` 选项的 `ec2 run-instances` 命令。在以下示例中，将 *`m5.xlarge`* 替换为您要使用的实例类型，将 *`ami-035be7bafff33b6b6`* 替换为与该实例类型关联的 AMI，将 *`subnet-12ab3c45`* 替换为您要查询的可用区中的子网。

```
aws ec2 run-instances --instance-type m5.xlarge --dry-run --image-
id ami-035be7bafff33b6b6 --subnet-id subnet-12ab3c45
```

有关查找 AMI ID 的说明，请参阅查找 Linux AMI。您可以使用 describe-subnets 命令查找子网 ID。

要了解有关如何发现可用实例类型的更多信息，请参阅查找 Amazon EC2 实例类型。

确定可用的实例类型后，可以执行以下任一操作：

- 在同一个区域和 EC2 子网中创建集群，并选择与初始选择具有相似功能的其它实例类型。有关受支持实例类型的列表，请参阅支持的实例类型。要比较 EC2 实例类型的功能，请参阅 Amazon EC2 实例类型。
- 在提供实例类型可用且受 Amazon EMR 支持的可用区中为集群选择子网。

EC2 容量不足

当您尝试在没有更多指定 EC2 实例类型的可用区中创建集群或向集群添加实例时，会发生“EC2 is out of capacity for *InstanceType*”的错误。为集群选择的子网决定可用区。

要创建集群，请执行以下操作之一：

- 指定具有相似功能的不同实例类型
- 在另一个区域中创建集群
- 在可用区中选择您想要的实例类型可用的子网。

要将实例添加到正在运行的集群中，请执行以下操作之一：

- 修改实例组配置或实例集配置，以添加具有类似功能的可用实例类型。有关受支持实例类型的列表，请参阅支持的实例类型。要比较 EC2 实例类型的功能，请参阅 Amazon EC2 实例类型。
- 终止集群并在提供实例类型的区域和可用区中重新创建集群。

输入与输出错误

以下是集群输入与输出操作中的常见错误。

主题

- 您的 Amazon Simple Storage Service（Amazon S3）路径中是否至少有三个斜杠？
- 您是否正在尝试以递归的方式遍历输入目录？
- 您的输出目录是否已经存在？
- 您是否尝试指定一个使用 HTTP URL 的资源？
- 您是否在使用无效名称格式引用 Amazon S3 存储桶？
- 您在向 Amazon S3 装载数据或者从中移出数据时是否遇到问题？

您的 Amazon Simple Storage Service（Amazon S3）路径中是否至少有三个斜杠？

在指定 Amazon S3 存储桶时，必须在 URL 的末尾加上终止斜杠。举例来说，不要以下列格式引用存储桶“s3n://*DOC-EXAMPLE-BUCKET1*”，而应使用“s3n://*DOC-EXAMPLE-BUCKET1*/”，否则 Hadoop 会在大多数情况下导致集群失败。

您是否正在尝试以递归的方式遍历输入目录？

Hadoop 不会以递归的方式搜索文件的输入目录。如果您拥有 /corpus/01/01.txt、/corpus/01/02.txt、/corpus/02/01.txt 等目录结构，且指定 /corpus/ 作为集群的输入参数，则 Hadoop 将找不到任何输入文件，因为 /corpus/ 目录为空，且 Hadoop 不会查看子目录的内容。同样地，Hadoop 不会以递归的方式查看 Amazon S3 存储桶的子目录。

输入文件必须直接放入您指定的输入目录或者 Amazon S3 存储桶中，而不是子目录中。

您的输出目录是否已经存在？

如果您指定的输出路径已经存在，Hadoop 会在大多数情况下导致集群失败。这意味着如果您运行集群一次，并使用完全相同的参数再运行一次，那么第一次正常很可能运行，但后面的运行就再也不会成功；因为在第一次运行后，会出现输出路径，这导致所有后续的运行失败。

您是否尝试指定一个使用 HTTP URL 的资源？

Hadoop 不接受使用 http:// 前缀指定的资源位置。您不能使用 HTTP URL 引用资源。例如，以 http://mysite/myjar.jar 作为 JAR 参数进行传递会导致集群失败。

您是否在使用无效名称格式引用 Amazon S3 存储桶？

如果您尝试在 Amazon EMR 中使用诸如“*DOC-EXAMPLE-BUCKET1*.1”的存储桶名称，那么集群会失败，因为 Amazon EMR 要求存储桶名称必须是有效的 RFC 2396 主机名称，名称不能以数字结尾。

另外，根据 Hadoop 的要求，与 Amazon EMR 一起使用的 Amazon S3 存储桶名称必须仅包含小写字母、数字、英文句点（.）和连字符（-）。有关如何格式化 Amazon S3 存储桶名称的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的存储桶限制。

您在向 Amazon S3 装载数据或者从中移出数据时是否遇到问题？

Amazon S3 是最常见的 Amazon EMR 输入与输出源。常见的错误是将 Amazon S3 当做典型的文件系统进行处理。Amazon S3 和文件系统之间存在区别，在运行集群时必须予以考虑。

- 如果 Amazon S3 中出现内部错误，您的应用程序必须顺利地予以处理，并重试该操作。
- 如果调用 Amazon S3 的返回时间太长，那么，您的应用程序可能必须降低调用 Amazon S3 的频率。
- 列出 Amazon S3 存储桶中的所有对象是一种成本很高的调用。您的应用程序应该将此类操作的次数降到最低程度。

您可以使用多种方式提高您的集群与 Amazon S3 的交互。

- 使用 Amazon EMR 的最新发行版启动您的集群。
- 使用 S3DistCp 将对象移入和移出 Amazon S3。S3DistCp 会实施错误处理、重试和回退，以满足 Amazon S3 的要求。有关更多信息，请参阅使用 S3DistCp 的分布式复制。
- 设计应用程序时应考虑最终一致性这一因素。在集群运行时将 HDFS 用于中间数据存储和 Amazon S3，以便仅输入原始数据与输出最终结果。
- 如果您的集群每秒向 Amazon S3 提交不少于 200 个事务，请联系支持人员，以每秒提交更多的事务为前提准备存储桶，并考虑使用 Amazon S3 性能提示与和技巧中所述的密钥分区策略。
- 设置 Hadoop 配置设置 io.file.buffer.size 为 65536。这可让 Hadoop 在 Amazon S3 对象中搜索时花费较少的时间。
- 如果您的集群遇到 Amazon S3 并发问题，请考虑禁用 Hadoop 的推测执行功能。当您排除速度很慢的集群故障时，这也非常有用。您可以通过将 `mapreduce.map.speculative` 和 `mapreduce.reduce.speculative` 属性设置为 `false` 来执行此操作。在您启动集群时，您可以使用 `mapred-env` 配置分类来设置这些值。有关更多信息，请参阅 Amazon EMR 发行版指南中的配置应用程序。
- 如果您运行的是 Hive 集群，请参阅 您是否有关于在 Amazon S3 与 Hive 之间加载数据的问题？。

有关 Amazon S3 权限的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Amazon S3 错误最佳实践。

权限错误

使用权限或凭证时常出现以下错误。

主题

- 您将正确凭证传递给了 SSH 吗？
- 如果您在使用 IAM，您是否有正确的 Amazon EC2 策略设置？

您将正确凭证传递给了 SSH 吗？

如果您不能使用 SSH 连接主节点，最有可能是您的安全凭证出了问题。

首先，检查并确保包含您 SSH 密钥的 .pem 文件有正确的权限。您可以使用 chmod 更改 .pem 文件上的权限，如以下示例所示，其中，要把 mykey.pem 替换为自己的 .pem 文件的名称。

```
chmod og-rwx mykey.pem
```

第二种可能性是，您没使用创建集群时指定的密钥对。如果您创建了多个密钥对，这种情况很容易出现。检查 Amazon EMR 控制台中的集群详细信息（或使用 CLI 中的 `--describe` 选项），并确保创建集群时指定的密钥对的名称无误。

验证您在使用正确的密钥对且 .pem 文件上的权限正确设置后，您可以使用以下命令通过 SSH 连接主节点，其中，要把 mykey.pem 替换为您个人 .pem 文件的名称，把 `hadoop@ec2-01-001-001-1.compute-1.amazonaws.com` 替换为主节点的公有 DNS 名称（通过 CLI 中的 `--describe` 选项或通过 Amazon EMR 控制台实现）。

> **Important**
>
> 在连接到 Amazon EMR 集群节点时，必须使用登录名 `hadoop`，否则可能发生类似 `Server refused our key` 的错误。

```
ssh -i mykey.pem hadoop@ec2-01-001-001-1.compute-1.amazonaws.com
```

有关更多信息，请参阅使用 SSH 连接到主节点。

如果您在使用 IAM，您是否有正确的 Amazon EC2 策略设置？

因为 Amazon EMR 使用 EC2 实例作为节点，Amazon EMR 的用户也需要设置某些 Amazon EC2 策略，以便 Amazon EMR 能代表用户管理那些实例。如果您没有设置所需的权限，则 Amazon EMR 返回以下错误："账户没有调用 EC2 的权限。"

有关您的 IAM 账户需要设置以运行 Amazon EMR 的 Amazon EC2 策略的更多信息，请参阅Amazon EMR 如何与 IAM 配合使用。

Hive 集群错误

您通常可以在 `syslog` 文件中找出 Hive 错误的原因，可以从 Steps (步骤) 窗格中链接到此文件。如果您无法在此文件中确定问题，那么可以检查 Hadoop 任务尝试错误消息。在 Task Attempts (任务尝试) 窗格上链接到此问题。

Hive 集群常会出现以下错误。

主题

- 您使用的是否是最新版本的 Hive？
- 您是否在 Hive 脚本中遇到了语法错误？
- 作业是否在交互运行时失败？
- 您是否有关于在 Amazon S3 与 Hive 之间加载数据的问题？

您使用的是否是最新版本的 Hive？

最新版本的 Hive 具有所有当前修补程序和缺陷修复，可以解决您的问题。

您是否在 Hive 脚本中遇到了语法错误？

如果步骤失败，请查看运行 Hive 脚本步骤日志的 `stdout` 文件。如果错误不在此文件中，请查看失败任务尝试的任务尝试日志的 `syslog` 文件。有关更多信息，请参阅查看日志文件。

作业是否在交互运行时失败？

如果您在主节点上交互运行 Hive 并且集群失败，那么请查看失败任务尝试的任务尝试日志中的 `syslog` 条目。有关更多信息，请参阅查看日志文件。

您是否有关于在 Amazon S3 与 Hive 之间加载数据的问题？

如果您有关于访问 Amazon S3 中数据的问题，那么请先检查 您在向 Amazon S3 装载数据或者从中移出数据时是否遇到问题？ 中列出的可能原因。如果那些问题都不是原因，请考虑以下 Hive 特定的选项。

- 确保您使用的是最新版本的 Hive，它具有可解决您的问题的所有当前修补程序和错误修复。有关更多信息，请参阅 Apache Hive。
- 使用 `INSERT OVERWRITE` 要求列出 Amazon S3 存储桶或文件夹的内容。这是一项成本高昂的操作。如果可能的话，请手动修剪路径，而不是让 Hive 列出并删除现有对象。
- 如果您使用 5.0 之前的 Amazon EMR 发行版，则可以使用 HiveQL 中的以下命令，在集群上本地预缓存 Amazon S3 列表操作的结果：

```
set hive.optimize.s3.query=true;
```

- 在可能的情况下，使用静态分割。
- 在 Hive 和 Amazon EMR 的一些版本中，使用 ALTER TABLES 有可能会失败，因为此表的存储位置可能与 Hive 的期望位置不同。解决方法是在 `/home/hadoop/conf/core-site.xml` 中添加或更新以下内容：

```
<property>
    <name>fs.s3n.endpoint</name>
    <value>s3.amazonaws.com</value>
</property>
```

VPC 错误

Amazon EMR 的 VPC 配置常会出现以下错误。

主题

- 无效子网配置
- 缺少 DHCP 选项集
- 权限错误
- 导致 START_FAILED 的错误
- 集群 Terminated with errors 和 NameNode 无法启动

无效子网配置

在 Cluster Details（集群详细信息）页面的 Status（状态）字段中，您会看到与以下内容相似的错误：

```
The subnet configuration was invalid: Cannot find route to InternetGateway
in main RouteTable rtb-id for vpc vpc-id.
```

要解决此问题，您必须创建 Internet 网关并将其连接到您的 VPC。有关更多信息，请参阅在您的 VPC 中添加互联网网关。

或者，验证您是否已经为 VPC 配置启用 DNS 解析并启用启用 DNS 主机名支持。有关更多信息，请参阅将 DNS 与您的 VPC 一起使用。

缺少 DHCP 选项集

您可以看到集群系统日志 (syslog) 的步骤失败信息包含与以下内容相似的错误：

```
ERROR org.apache.hadoop.security.UserGroupInformation
(main): PriviledgedActionException as:hadoop (auth:SIMPLE)
cause:java.io.IOException:
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

或

```
ERROR org.apache.hadoop.streaming.StreamJob (main): Error Launching job :
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

要解决此问题，必须配置 VPC，使其包含将参数设置为以下值的 DHCP 选项集：

> **Note**
>
> 如果您使用 Amazon GovCloud (US-West) 区域，请将 domain-name 设置为 **us-gov-west-1.compute.internal**，而不是以下示例中使用的值。

- domain-name = **ec2.internal**

  如果您所在的区域是美国东部（弗吉尼亚北部），请使用 **ec2.internal**。对于其它区域，请使用 *region-name*.**compute.internal**。例如，在 us-west-2 中，请使用 domain-name=**us-west-2.compute.internal**。

- domain-name-servers = **AmazonProvidedDNS**

有关更多信息，请参阅 DHCP 选项集。

权限错误

stderr 日志中记录的某个步骤失败表示 Amazon S3 资源没有适当的权限。这是一个 403 错误，内容如下所示：

```
Exception in thread "main" com.amazonaws.services.s3.model.AmazonS3Exception: Access
 Denied (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request
 ID: REQUEST_ID
```

如果 ActionOnFailure 设置为 TERMINATE_JOB_FLOW，则将导致集群以 SHUTDOWN_COMPLETED_WITH_ERRORS 状态终止。

可通过几种方法解决此问题，包括：

- 如果您正在 VPC 中使用 Amazon S3 存储桶策略，请确保通过创建一个 VPC 终端节点并在创建该终端节点时在 Policy (策略) 选项下选择 Allow all (全部允许) 来授予对所有存储桶的访问权。
- 确保与 S3 资源关联的任何策略都包括您从中启动集群的 VPC。
- 尝试从集群中运行以下命令来验证您是否能访问存储桶

  ```
  hadoop fs -copyToLocal s3://path-to-bucket /tmp/
  ```

- 通过在集群上的 log4j.logger.org.apache.http.wire 文件中将 DEBUG 参数设置为 /home/hadoop/conf/log4j.properties，可以获得更具体的调试信息。在尝试从集群访问存储桶后，可以查看 stderr 日志文件。该日志文件将提供更多详细信息：

  ```
  Access denied for getting the prefix for bucket - us-west-2.elasticmapreduce with
   path samples/wordcount/input/
  15/03/25 23:46:20 DEBUG http.wire: >> "GET /?prefix=samples%2Fwordcount%2Finput
  %2F&delimiter=%2F&max-keys=1 HTTP/1.1[\r][\n]"
  15/03/25 23:46:20 DEBUG http.wire: >> "Host: us-
  west-2.elasticmapreduce.s3.amazonaws.com[\r][\n]"
  ```

导致 START_FAILED 的错误

在 AMI 3.7.0 之前，对于指定了主机名的 VPC，Amazon EMR 会按如下方式在子网的内部主机名与自定义域地址之间建立映射：ip-*X.X.X.X.customdomain.com*.tld。例如，如果主机名为 ip-10.0.0.10 且 VPC 已将域名选项设置为 customdomain.com，则通过 Amazon EMR 映射所生成的主机名将为 ip-10.0.1.0.customdomain.com。这会在 /etc/hosts 中添加一个条目以将主机名解析为 10.0.0.10。此行为已随 AMI 3.7.0 更改，现在 Amazon EMR 已完全遵循 VPC 的 DHCP 配置。以前，客户还可使用引导操作来指定主机名映射。

如果您想保留此行为，则必须提供 DNS，并为自定义域转发所需的解析设置。

集群 Terminated with errors 和 NameNode 无法启动

在使用自定义 DNS 域名的 VPC 中启动 EMR 集群时，您的集群可能会启动失败并在控制台中显示以下错误消息：

```
Terminated with errors  On the master instance(instance-id), bootstrap action 1
 returned a  non-zero return code
```

此故障是由 NameNode 无法启动导致的。这会导致 NameNode 日志中出现以下错误，其 Amazon S3 URI 使用以下格式：s3://*mybucket*/*logs*/*cluster-id*/daemons/*master instance-id*/hadoop-hadoop-namenode-*master node hostname*.log.gz

```
2015-07-23 20:17:06,266 WARN
      org.apache.hadoop.hdfs.server.namenode.FSNamesystem (main): Encountered
 exception
      loading fsimage  java.io.IOException: NameNode is not formatted.
      at

 org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:212)
           at

 org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:1020)
           at

 org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:739)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:537)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:596)
```

```
      at  org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:765)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:749)
      at

 org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1441)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1507)
```

这是由一个潜在问题造成的：由于同时使用了 Amazon 提供的 DNS 服务器和用户提供的自定义 DNS 服务器，在 VPC 中启动 EMR 集群时，EC2 实例可能拥有多组完全限定域名。如果用户提供的 DNS 服务器未为用于指定 EMR 集群中节点的任何 A 记录提供任何指针 (PTR) 记录，则这种配置方式会导致集群启动失败。解决方案是为每个 A 记录 (在 VPC 的任意子网中启动 EC2 实例时创建) 添加 1 条 PTR 记录。

流式集群错误

您通常可以在 `syslog` 文件中找出流式错误的原因。在 Steps (步骤) 窗格上链接到此问题。

流式集群常会出现以下错误。

主题

- 数据是否以错误的格式发送到映射器？
- 您的脚本是否超时？
- 是否正在传递无效的流式参数？
- 您的脚本退出时是否有错误？

数据是否以错误的格式发送到映射器？

要检查是否出现了这种错误，请在任务尝试日志的失败任务尝试的 `syslog` 文件中查看错误消息。有关更多信息，请参阅查看日志文件。

您的脚本是否超时？

映射器或 Reducer 脚本的默认超时值为 600 秒。如果脚本需要的时间长于此时长，那么任务尝试将会失败。通过在任务尝试日志中检查失败任务尝试的 `syslog` 文件，可以检验是否出现了这种情况。有关更多信息，请参阅查看日志文件。

通过为 mapred.task.timeout 配置设置设定新值，您可以更改时间限值。此设置会指定毫秒数，经过此时段后，Amazon EMR 将终止尚未读取输入、写入输出或更新其状态字符串的任务。您可以通过传递其它流式参数 -jobconf mapred.task.timeout=800000 更新此值。

是否正在传递无效的流式参数？

Hadoop 流式处理仅支持以下参数。如果您传递的参数不同于以下列出的内容，集群将会失败。

```
-blockAutoGenerateCacheFiles
-cacheArchive
-cacheFile
-cmdenv
-combiner
-debug
-input
-inputformat
-inputreader
-jobconf
-mapper
-numReduceTasks
-output
-outputformat
-partitioner
-reducer
-verbose
```

此外，Hadoop 流式处理只能识别使用 Java 语法传入的参数，即前面加一个连字符的参数。如果您传入的参数前面加了两个连字符，集群将会失败。

您的脚本退出时是否有错误？

如果映射器或 Reducer 脚本退出时出现错误，您可以在失败任务尝试的任务尝试日志的 stderr 文件中找到此错误。有关更多信息，请参阅查看日志文件。

自定义 JAR 集群错误

自定义 JAR 集群常会出现以下错误。

主题

- 创建作业前，您的 JAR 抛出异常吗？
- 您的 JAR 在映射任务内部抛出错误吗？

创建作业前，您的 JAR 抛出异常吗？

如果在创建 Hadoop 作业期间您的自定义 JAR 的主程序抛出异常，查看的最佳位置是步骤日志的 `syslog` 文件。有关更多信息，请参阅查看日志文件。

您的 JAR 在映射任务内部抛出错误吗？

如果在处理输入数据期间您的自定义 JAR 和映射器抛出异常，查看的最佳位置是任务尝试日志的 `syslog` 文件。有关更多信息，请参阅查看日志文件。

Amazon GovCloud (US-West) 错误

Amazon GovCloud (US-West) 区域的安全、配置和默认设置与其它区域有所不同。因此，在采用更普遍的问题排查建议之前，请先使用以下核对清单排查特定于 Amazon GovCloud (US-West) 区域的 Amazon EMR 问题。

- 验证 IAM 角色配置是否正确。有关更多信息，请参阅为 Amazon EMR 配置 IAM 服务角色对 Amazon服务和资源的权限。
- 确保您的 VPC 配置已正确配置了 DNS 解决方案/主机名支持、Internet 网关和 DHCP 选项设置参数。有关更多信息，请参阅VPC 错误。

如果这些步骤都不能解决问题，请继续执行排查常见 Amazon EMR 错误的步骤。有关更多信息，请参阅Amazon EMR 中的常见错误。

查找缺少的集群

如果控制台列表或 `ListClusters` API 中缺少您的集群，请检查以下内容：

- 确认自完成之日起集群的运行期限小于两个月。Amazon EMR 会免费保存已完成集群的相关元数据信息两个月。您无法从控制台中删除已完成的集群，相反，Amazon EMR 会在两个月后自动清除已完成的集群。
- 确认您具有查看集群的角色权限。
- 确认您所查看的是与集群驻留所在的同一 Amazon Web Services 区域。

失败集群的故障排查

本节向您介绍对已经失败的集群进行故障排除的过程。这意味着集群终止，出现了错误代码。

> **Note**
>
> 当 EMR 集群因错误而终止时，`DescribeCluster` 和 `ListClusters` API 会返回错误代码和错误消息。对于某些集群错误，`ErrorDetail` 数据数组还可以帮助您排查失败的问题。有关更多信息，请参阅包含 ErrorDetail 信息的错误代码。

如果集群成功运行，但花了很长时间才返回结果，请参阅 速度很慢的集群的故障排查。

主题

- 步骤 1：收集有关此问题的数据
- 步骤 2：检查环境
- 步骤 3：查看最后一次状态更改
- 步骤 4：检查日志文件
- 步骤 5：分步测试集群

步骤 1：收集有关此问题的数据

对集群进行故障排查的第一步是收集有关出现问题的信息以及集群的当前状态和配置。此信息将在以下步骤中用于确认或排查问题的可能原因。

定义问题

首先要对这个问题作出明确的定义。问自己一些问题：

- 我预计发生什么？ 实际发生了什么？
- 首次出现此问题是什么时候？ 自那以后有多久发生一次？
- 配置或运行集群的方式是否有任何变化？

集群详细信息

以下集群详细信息有助于追踪问题。有关如何收集此信息的更多信息，请参阅查看集群状态和详细信息。

- 集群的标识符。（也称为工作流标识符。）
- Amazon Web Services 区域 和在其中启动集群的可用区。
- 集群的状态，包括上次状态更改的详细信息。
- 为主节点、核心节点和任务节点指定的 EC2 实例的类型和数量。

步骤 2：检查环境

Amazon EMR 作为 Web 服务的生态系统的一部分和开源软件运行。影响那些依赖项的事情会影响 Amazon EMR 的性能。

主题

- 检查服务中断
- 检查使用限制
- 检查发布版
- 检查 Amazon VPC 子网配置

检查服务中断

Amazon EMR 在内部使用多种 Amazon Web Services。它在 Amazon EC2 上运行虚拟服务器，在 Amazon S3 上存储数据和脚本，并向 CloudWatch 报告指标。中断这些服务的事件很少见。但一旦发生，即可能会导致 Amazon EMR 出现问题。

进一步操作之前，请检查 Service Health Dashboard。检查您启动集群的区域，以查看这些服务中是否存在中断事件。

检查使用限制

如果您正在启动一个大型集群，同时启动了大量集群，或者您是与其他用户共享 Amazon Web Services 账户的用户，则集群可能会因为您超出 Amazon 服务限制而失败。

Amazon EC2 将在单个Amazon区域上运行的虚拟服务器实例数量限制为 20 个按需或预留实例。如果您启动的集群包含超过 20 个节点，或者启动的集群会导致您 Amazon Web Services 账户上的活动 EC2 实例总数超过 20 个，则该集群将无法启动所需的全部 EC2 实例并且可能会失败。出现这种情况时，Amazon EMR 会返回 `EC2 QUOTA EXCEEDED` 错误。您可以通过提交增加 Amazon EC2 实例限制申请，请求Amazon增加您可以在您的账户上运行的 EC2 实例的数量。

可能导致您超出使用限制的另一件事是集群终止与释放其所有资源之间的延迟。根据配置差异，以前的集群可能需要 5-20 分钟的时间才能完全终止并释放分配的资源。如果您在尝试启动集群时收到 EC2 QUOTA EXCEEDED 错误，原因可能是最近终止的集群中的资源尚未释放。在这种情况下，您可以请求增加您的 Amazon EC2 配额，也可以等待 20 分钟并重新启动集群。

Amazon S3 将在账户上创建的存储桶数量限制为 100。如果您的集群创建的新存储桶超过此限制，则存储桶创建将失败，并可能导致集群失败。

检查发布版

将用于启动集群的发行版标注与最新的 Amazon EMR 版本进行比较。Amazon EMR 的每个版本都包含改进，例如新的应用程序、功能、补丁和错误修复。影响集群的问题可能已经在最新的发布版中得到修复。如果可能，请使用最新版本的重新运行集群。

检查 Amazon VPC 子网配置

如果您的集群是在 Amazon VPC 子网中启动的，则需要按照配置联网中所述配置子网。此外，请检查您启动集群的子网是否有足够的空闲弹性 IP 地址，以便为集群中的每个节点分配一个。

步骤 3：查看最后一次状态更改

最后一次状态更改提供关于集群最后一次更改状态时所发生的事情的信息。这常常包含一些信息，告诉您当集群把状态更改为 FAILED 时出错的地方。例如，如果您启动了流集群并指定了已经在 Amazon S3 中的输出位置，集群将会失败，同时提示最后一次状态更改为“流输出目录已经存在”。

您可以通过查看集群的详细信息窗格从控制台查找最后一次状态更改值，可以使用 `list-steps` 或 `describe-cluster` 参数从 CLI 查看或使用 `DescribeCluster` 和 `ListSteps` 操作从 API 查看。有关更多信息，请参阅查看集群状态和详细信息。

步骤 4：检查日志文件

下一步是检查日志文件，以找到错误代码或集群遇到的问题的其它指示。有关可用日志文件、查找日志文件的位置以及如何查看这些日志文件的信息，请参阅查看日志文件。

可能需要进行一些调查工作才能确定发生了什么。Hadoop 在集群中的各个节点上运行任务尝试中的任务工作。Amazon EMR 可以启动推测性任务尝试，终止未首先完成的其它任务尝试。这会生成重要的活动，并记录到控制器，stderr 和 syslog 日志文件中。此外，多个任务尝试同时运行，但一个日志文件只能线性显示结果。

首先检查引导操作日志，查看集群启动期间的错误或意外配置更改。从那里查看步骤日志，以确定作为错误步骤的一部分启动的 Hadoop 任务。检查 Hadoop 任务日志以确定失败的任务尝试。任务尝试日志将包含导致任务尝试失败的详细信息。

以下部分说明如何使用各种日志文件识别集群中的错误。

检查引导操作日志

引导操作会在集群启动时运行脚本。它们通常用于在集群上安装附加软件或更改默认值的配置设置。检查这些日志可以深入了解集群设置期间发生的错误以及可能影响性能的配置设置更改。

检查步骤日志

步骤日志有四种类型。

- 控制器 – 包含由 Amazon EMR（Amazon EMR）生成的文件，这些文件由尝试运行您的步骤时遇到的错误引起。如果您的步骤在加载时失败，您可以在此日志中找到堆栈跟踪。此处通常描述加载或访问您的应用程序的错误，以及缺少映射器文件错误。
- stderr 包含处理步骤时发生的错误消息。此处通常介绍应用程序加载错误。此日志有时包含堆栈跟踪。
- stdout – 包含由映射器和折叠器（reducer）可执行文件生成的状态。此处通常介绍应用程序加载错误。此日志有时包含应用程序错误消息。
- syslog – 包含来自非 Amazon 软件（例如 Apache 和 Hadoop）的日志。流式传输错误通常在此处进行描述。

检查 stderr 是否有明显的错误。如果 stderr 显示一个简短的错误列表，则该步骤会快速停止并引发错误。这通常由集群中运行的映射器和折叠器（reducer）应用程序中的错误引起。

检查控制器和 syslog 的最后一行是否有错误或故障通知。注意有关失败任务的任何通知，特别是显示“Job Failed (任务失败)”的时候。

检查任务尝试日志

如果在前面对步骤日志的分析中发现了一个或多个失败的任务，请调查相应任务尝试的日志，以获得更详细的错误信息。

步骤 5：分步测试集群

您尝试跟踪错误源时的一项有用的技术是重启集群并逐一向集群提交步骤。这可以让您检查每个步骤的结果然后才处理下一个步骤，让您有机会更正并重新运行失败的步骤。这也有一个优势，即您仅需上传一次输入数据。

分步测试集群

1. 启动新集群，启用保持活动状态和终止保护两种功能。保持活动状态可在集群处理了所有待执行的步骤后保持集群运行。终止保护可以防止集群在出现错误时关闭。有关更多信息，请参阅将集群配置为在步骤执行后继续或终止和使用终止保护。
2. 向集群提交一个步骤。有关更多信息，请参阅向集群提交工作。
3. 当该步骤完成处理后，检查步骤日志文件中是否有错误。有关更多信息，请参阅步骤 4：检查日志文件。查找这些日志文件最快的方法是连接到主节点并查看那里的日志文件。直到该步骤运行一些时间结束或失败，步骤日志文件才会出现。
4. 如果该步骤成功无误，运行下一个步骤。如果有错误，调查日志文件中的错误。如果是您的代码中的错误，更正并重新运行该步骤。继续操作直到所有步骤均运行无误。
5. 当您完成了集群调试想终止集群时，您必须手动终止该集群。这么做是必要的，因为集群启动时启用了终止保护。有关更多信息，请参阅使用终止保护。

速度很慢的集群的故障排查

本节向您介绍对仍在运行但花费很长时间才能返回结果的集群进行故障排除的过程。有关当集群已终止且出现错误代码时要执行的操作的详细信息，请参见 失败集群的故障排查

Amazon EMR 可让您指定集群中实例的数量和类型。这些规范是对完成您的数据处理所需速度造成影响的主要手段。您可能考虑的一件事情是重新运行该集群，这次要指定拥有更多资源的 EC2 实例或指定集群中更多的实例。有关更多信息，请参阅配置集群硬件和联网。

以下主题向您介绍确定造成集群缓慢的其它原因的过程。

主题

- 步骤 1：收集有关此问题的数据
- 步骤 2：检查环境
- 步骤 3：检查日志文件
- 步骤 4：检查集群和实例运行状况

- 步骤 5：检查已暂停的组
- 步骤 6：审查配置设置
- 步骤 7：检查输入数据

步骤 1：收集有关此问题的数据

对集群进行故障排查的第一步是收集有关出现问题的信息以及集群的当前状态和配置。此信息将在以下步骤中用于确认或排查问题的可能原因。

定义问题

首先要对这个问题作出明确的定义。问自己一些问题：

- 我预计发生什么？ 实际发生了什么？
- 首次出现此问题是什么时候？ 自那以后有多久发生一次？
- 配置或运行集群的方式是否有任何变化？

集群详细信息

以下集群详细信息有助于追踪问题。有关如何收集此信息的更多信息，请参阅查看集群状态和详细信息。

- 集群的标识符。（也称为工作流标识符。）
- Amazon Web Services 区域 和在其中启动集群的可用区。
- 集群的状态，包括上次状态更改的详细信息。
- 为主节点、核心节点和任务节点指定的 EC2 实例的类型和数量。

步骤 2：检查环境

主题

- 检查服务中断
- 检查使用限制
- 检查 Amazon VPC 子网配置
- 重启集群

检查服务中断

Amazon EMR 在内部使用多种 Amazon Web Services。它在 Amazon EC2 上运行虚拟服务器，在 Amazon S3 上存储数据和脚本，并向 CloudWatch 报告指标。中断这些服务的事件很少见。但一旦发生，即可能会导致 Amazon EMR 出现问题。

进一步操作之前，请检查 Service Health Dashboard。检查您启动集群的区域，以查看这些服务中是否存在中断事件。

检查使用限制

如果您正在启动一个大型集群，同时启动了大量集群，或者您是与其他用户共享 Amazon Web Services 账户的用户，则集群可能会因为您超出 Amazon 服务限制而失败。

Amazon EC2 将在单个Amazon区域上运行的虚拟服务器实例数量限制为 20 个按需或预留实例。如果您启动的集群包含超过 20 个节点，或者启动的集群会导致您 Amazon Web Services 账户上的活动 EC2 实例总数超过 20 个，则该集群将无法启动所需的全部 EC2 实例并且可能会失败。出现这种情况时，Amazon EMR 会返回 `EC2 QUOTA EXCEEDED` 错误。您可以通过提交增加 Amazon EC2 实例限制申请，请求Amazon增加您可以在您的账户上运行的 EC2 实例的数量。

可能导致您超出使用限制的另一件事是集群终止与释放其所有资源之间的延迟。根据配置差异，以前的集群可能需要 5-20 分钟的时间才能完全终止并释放分配的资源。如果您在尝试启动集群时收到 `EC2 QUOTA EXCEEDED` 错误，原因可能是最近终止的集群中的资源尚未释放。在这种情况下，您可以请求增加您的 Amazon EC2 配额，也可以等待 20 分钟并重新启动集群。

Amazon S3 将在账户上创建的存储桶数量限制为 100。如果您的集群创建的新存储桶超过此限制，则存储桶创建将失败，并可能导致集群失败。

检查 Amazon VPC 子网配置

如果您的集群是在 Amazon VPC 子网中启动的，则需要按照配置联网中所述配置子网。此外，请检查您启动集群的子网是否有足够的空闲弹性 IP 地址，以便为集群中的每个节点分配一个。

重启集群

处理减速可能是由瞬变条件造成的。请考虑终止并重启该集群，看看性能是否有所改进。

步骤 3：检查日志文件

下一步是检查日志文件，以找到错误代码或集群遇到的问题的其它指示。有关可用日志文件、查找日志文件的位置以及如何查看这些日志文件的信息，请参阅查看日志文件。

可能需要进行一些调查工作才能确定发生了什么。Hadoop 在集群中的各个节点上运行任务尝试中的任务工作。Amazon EMR 可以启动推测性任务尝试，终止未首先完成的其它任务尝试。这会生成重要的活动，并记录到控制器，stderr 和 syslog 日志文件中。此外，多个任务尝试同时运行，但一个日志文件只能线性显示结果。

首先检查引导操作日志，查看集群启动期间的错误或意外配置更改。从那里查看步骤日志，以确定作为错误步骤的一部分启动的 Hadoop 任务。检查 Hadoop 任务日志以确定失败的任务尝试。任务尝试日志将包含导致任务尝试失败的详细信息。

以下部分说明如何使用各种日志文件识别集群中的错误。

检查引导操作日志

引导操作会在集群启动时运行脚本。它们通常用于在集群上安装附加软件或更改默认值的配置设置。检查这些日志可以深入了解集群设置期间发生的错误以及可能影响性能的配置设置更改。

检查步骤日志

步骤日志有四种类型。

- 控制器 – 包含由 Amazon EMR（Amazon EMR）生成的文件，这些文件由尝试运行您的步骤时遇到的错误引起。如果您的步骤在加载时失败，您可以在此日志中找到堆栈跟踪。此处通常描述加载或访问您的应用程序的错误，以及缺少映射器文件错误。
- stderr 包含处理步骤时发生的错误消息。此处通常介绍应用程序加载错误。此日志有时包含堆栈跟踪。
- stdout – 包含由映射器和折叠器（reducer）可执行文件生成的状态。此处通常介绍应用程序加载错误。此日志有时包含应用程序错误消息。
- syslog – 包含来自非 Amazon 软件（例如 Apache 和 Hadoop）的日志。流式传输错误通常在此处进行描述。

检查 stderr 是否有明显的错误。如果 stderr 显示一个简短的错误列表，则该步骤会快速停止并引发错误。这通常由集群中运行的映射器和折叠器（reducer）应用程序中的错误引起。

检查控制器和 syslog 的最后一行是否有错误或故障通知。注意有关失败任务的任何通知，特别是显示“Job Failed (任务失败)”的时候。

检查任务尝试日志

如果在前面对步骤日志的分析中发现了一个或多个失败的任务，请调查相应任务尝试的日志，以获得更详细的错误信息。

检查 Hadoop 守护进程日志

在极少数情况下，Hadoop 本身可能会失败。要查看是否是这种情况，您必须查看 Hadoop 日志。它们位于每个节点的 `/var/log/hadoop/` 处。

您可以使用 JobTracker 日志将失败的任务尝试映射到运行它的节点。知道与任务尝试关联的节点后，您即可以检查托管该节点的 EC2 实例的运行状况，以查看是否存在任何问题（例如 CPU 或内存不足）。

步骤 4：检查集群和实例运行状况

Amazon EMR 集群由正在 Amazon EC2 实例上运行的节点组成。如果那些实例受到资源约束 (如耗尽 CPU 或内存)。遇到网络连接问题，或遭到终止，集群处理的速度会受到影响。

集群中最多有三种类型的节点：

- 主节点 – 管理集群。如果主节点遇到性能问题，整个集群都会受到影响。
- 核心节点 – 处理 map-reduce 任务并维护 Hadoop 分布式文件系统（HDFS）。如果这些节点之一遇到了性能问题，就会降低 HDFS 操作和 map-reduce 处理的速度。您可以向集群添加其它核心节点以改进性能，但无法移除核心节点。有关更多信息，请参阅手动调整正在运行的集群的大小。
- 任务节点 – 处理 map-reduce 任务。这些纯粹是计算资源，并不存储数据。您可以向集群添加任务节点，提高性能速度，或移除不需要的任务节点。有关更多信息，请参阅手动调整正在运行的集群的大小。

当您查看集群的运行状况时，应该既查看整个集群的性能，又查看各个实例的性能。您可以使用以下多种工具：

使用 CloudWatch 检查集群运行状况

每个 Amazon EMR 集群都会向 CloudWatch 报告指标。这些指标提供集群的概括性性能信息，如总负载、HDFS 使用率、正在运行的任务、其余任务、损坏的数据块等。查看 CloudWatch 指标可以让您了解集群进展的概况，深入剖析造成处理减速的原因。除了使用 CloudWatch 分析现有性能问题，您还可以设置警报，如果未来出现性能问题，CloudWatch 就会发出警报。有关更多信息，请参阅使用 CloudWatch 指标监控 Amazon EMR 指标。

检查任务状态和 HDFS 运行状况

使用集群详细信息页面上的 Application user interfaces (应用程序用户界面) 选项卡可查看 YARN 应用程序详细信息。对于某些应用程序，您可以深入了解更多详细信息和直接访问日志。这对 Spark 应用程序特别有用。有关更多信息，请参阅查看应用程序历史记录。。

Hadoop 向您提供可用于查看信息的一系列 Web 界面。有关如何访问这些 Web 界面的详细信息，请参见 查看 Amazon EMR 集群上托管的 Web 界面。

- JobTracker – 提供关于集群正在处理的任务的进度信息。您可以使用此界面确定作业陷入困境的时间。
- HDFS NameNode – 提供有关每个节点上 HDFS 利用率和可用空间百分比的信息。您可以使用此界面确定 HDFS 受到资源约束和需要其它容量的时间。
- TaskTracker – 提供关于集群正在处理的任务任务的信息。您可以使用此界面确定任务陷入困境的时间。

使用 Amazon EC2 查看实例运行状况

另一种查找集群中实例状态信息的方法是使用 Amazon EC2 控制台。因为集群中的每个节点在 EC2 实例上运行，您可以使用 Amazon EC2 提供的工具检查它们的状态。有关更多信息，请参阅查看 Amazon EC2 中的集群实例。

步骤 5：检查已暂停的组

如果尝试启动节点时遇到太多错误，实例组就会进入暂停状态。例如，如果新节点在执行引导操作时反复失败，实例组将在一段时间后进入 SUSPENDED 状态，而不是继续尝试供应新节点。

在以下情况中，节点可能无法出现：

- Hadoop 或该集群不知是何原因损坏，不接受新节点进入集群
- 新节点的引导操作失败
- 该节点无法正常工作，未能签入 Hadoop

如果实例组处于 SUSPENDED 状态，集群处于 WAITING 状态，您可以添加一个集群步骤，以重置所需核心节点和任务节点的数量。添加该步骤可恢复集群的处理，使实例组重回 RUNNING 状态。

有关如何重置暂停状态下的集群的详细信息，请参阅“暂停”状态。

步骤 6：审查配置设置

配置设置指定集群如何运行的详细信息，例如，重试任务的次数和排序可用的内存大小。当使用 Amazon EMR 启动集群时，除了标准 Hadoop 配置设置外，还有 Amazon EMR 特定的设置。这些配置设置存储在集群的主节点上。您可以检查配置设置，以确保集群有所需的资源高效运行。

Amazon EMR 定义用于启动集群的默认 Hadoop 配置设置。这些值基于 AMI 和您为该集群指定的实例类型。您可以使用引导操作或在作业执行参数中指定新值，从而修改配置设置的默认值。有关更多信息，请参阅创建引导操作以安装其它软件。为了确定引导操作是否更改了配置设置，请检查引导操作日志。

Amazon EMR 记录用于执行每次任务的 Hadoop 设置。日志数据存储在主节点 `/mnt/var/log/hadoop/history/` 目录下名为 `job_`*`job-id`*`_conf.xml` 的文件中，其中，将 *`job-id`* 替换为作业标识符。如果您已启用了日志归档，此数据复制到 Amazon S3 中的 `logs/`*`date`*`/`*`jobflow-id`*`/jobs` 文件夹，其中，*`date`* 是该任务运行的日期，*`jobflow-id`* 是该集群的标识符。

以下 Hadoop 作业配置设置对于调查性能问题尤其有用。有关 Hadoop 配置设置及它们如何影响 Hadoop 操作的详细信息，请转到 http://hadoop.apache.org/docs/。

> **Warning**
>
> 1. 如果单个节点出现故障，则在少于四个节点的集群上将 `dfs.replication` 设置为 1 可能会导致 HDFS 数据丢失。建议您使用具有至少四个核心节点的集群来处理生产工作负载。
> 2. Amazon EMR 不允许集群扩展 `dfs.replication` 下方的核心节点。例如，如果是 `dfs.replication = 2`，则最小核心节点数为 2。
> 3. 当您使用托管扩缩、自动扩缩或选择手动调整集群大小时，建议您将设置 `dfs.replication` 为 2 或更高。

配置设置	描述
dfs.replication	为了形成类似 RAID 的环境而接收复制的单个数据块 (类似于硬盘数据块) 的 HDFS 节点的数量。确定包含该数据块副本的 HDFS 节点的数量。
io.sort.mb	排序可用的总内存大小。此值应该是 10x io.sort.factor。此设置也可用于计算任务节点所用的总内存大小，计算方法

配置设置	描述
	是 io.sort.mb 与 mapred.tasktracker.ap.tasks.maximum 相乘。
io.sort.spill.percent	供排序期间使用，在该点将开始使用磁盘，因为分配的排序内存已满。
mapred.child.java.opts	已淘汰。使用 mapred.map.child.java.opts 和 mapred.reduce.child.java.opts。启动 JVM 在其内部执行任务时 TaskTracker 使用的 Java 选项。通用参数是“-Xmx”，设置最大的内存大小。
mapred.map.child.java.opts	启动 JVM 在其内部执行映射任务时 TaskTracker 使用的 Java 选项。通用参数是“-Xmx”，设置最大的内存堆大小。
mapred.map.tasks.speculative.execution	确定同一任务的映射任务尝试是否可以并行启动。
mapred.reduce.tasks.speculative.execution	确定同一任务的 reduce 任务尝试是否可以并行启动。
mapred.map.max.attempts	可以尝试映射任务的最大次数。如果所有尝试都失败，那么该映射任务可标记为“失败”。
mapred.reduce.child.java.opts	启动 JVM 在其内部执行减少任务时 TaskTracker 使用的 Java 选项。通用参数是“-Xmx”，设置最大的内存堆大小。
mapred.reduce.max.attempts	可以尝试 reduce 任务的最大次数。如果所有尝试都失败，那么该映射任务可标记为“失败”。
mapred.reduce.slowstart.completed.maps	尝试 reduce 任务前应该完成的映射任务量。不等待足够长的时间可能会造成在尝试中出现“Too many fetch-failure”错误。
mapred.reuse.jvm.num.tasks	一个任务在单个 JVM 中运行。指定可以重新使用相同 JVM 的任务数。
mapred.tasktracker.map.tasks.maximum	映射期间每个任务节点可并行执行的任务的最大数量。

配置设置	描述
mapred.tasktracker.reduce.tasks.maximum	化简期间每个任务节点可并行执行的任务的最大数量。

如果您的集群任务需要使用大量内存，您可以通过在每个核心节点使用较少的任务以及降低作业跟踪程序堆大小来增强性能。

步骤 7：检查输入数据

请查看您的输入数据。它是否在键值之间均匀分配？ 如果您的数据严重偏向一个或几个键值，那么可能将处理负载映射到少量节点，而其它节点则闲置。工作的不均衡分配可能会导致处理速度较慢。

不平衡数据集的示例是，依据按字母顺序排列的词运行集群，但有一个数据集仅包含以字母“a”开始的词。当工作被映射时，以“a”开始的节点处理值会过量，而以其它字母开始的节点处理词会处于闲置状态。

对 Lake Formation 集群进行故障排查

本部分向您介绍在将 Amazon EMR 与 Amazon Lake Formation 结合使用时对常见问题进行故障排查的过程。

不允许访问数据湖

您必须明确选择在 Amazon EMR 集群上进行数据筛选，然后才能分析和处理数据湖中的数据。当数据访问失败时，您将在笔记本条目的输出中看到一条通用 `Access is not allowed` 消息。

要选择并允许在 Amazon EMR 上进行数据筛选，请参阅《Amazon Lake Formation 开发人员指南》中的允许在 Amazon EMR 上进行数据筛选以获取说明。

会话过期

EMR Notebooks 和 Zeppelin 的会话超时由 Lake Formation 的 IAM 角色的 `Maximum CLI/API session duration` 设置控制。此设置的默认值为 1 小时。如果出现会话超时，在尝试返回 Spark SQL 命令时，您将在笔记本条目的输出中看到下列消息。

```
Error 401    HTTP ERROR: 401 Problem accessing /sessions/2/statements.
Reason:  JWT token included in request failed validation.
```

```
Powered by Jetty:// 9.3.24.v20180605
 org.springframework.web.client.HttpClientErrorException: 401 JWT token included in
 request failed validation…
```

要验证您的会话，请刷新页面。系统将提示您使用 IdP 重新进行身份验证并将重定向回笔记本。重新进行身份验证后，您可以继续运行查询。

用户在所请求表上没有权限

试图访问您没有访问权限的表时，在尝试运行 Spark SQL 命令时，您将在笔记本条目的输出中看到下列异常：

```
org.apache.spark.sql.AnalysisException:
 org.apache.hadoop.hive.ql.metadata.HiveException: Unable to fetch table table.
Resource does not exist or requester is not authorized to access requested
 permissions.
(Service: AWSGlue; Status Code: 400; Error Code: AccessDeniedException; Request ID: …
```

要访问该表，您必须通过在 Lake Formation 中更新与此表关联的权限，向用户授予访问权限。

查询与 Lake Formation 共享的跨账户数据

使用 Amazon EMR 从其他账户访问与您共享的数据时，某些 Spark 库会尝试调用 `Glue:GetUserDefinedFunctions` API 操作。由于 Amazon RAM 托管权限的版本 1 和 2 不支持此操作，您会收到以下错误消息：

```
"ERROR: User: arn:aws:sts::012345678901:assumed-role/my-
spark-role/i-06ab8c2b59299508a is not authorized to perform:
glue:GetUserDefinedFunctions on resource: arn:exampleCatalogResource
because no resource-based policy allows the glue:GetUserDefinedFunctions
action"
```

要解决此错误，创建资源共享的数据湖管理员必须更新附加到资源共享的 Amazon RAM 托管权限。托管权限的版本 3 Amazon RAM 允许主体执行 `glue:GetUserDefinedFunctions` 操作。

如果您创建了新的资源共享，Lake Formation 会默认应用最新版 Amazon RAM 托管权限，您无需执行任何操作。要为现有资源共享启用跨账户数据访问，您需要将 Amazon RAM 托管权限更新到版本 3。

您可以在 Amazon RAM 中查看分配给与您共享的资源的 Amazon RAM 权限。版本 3 中包含以下权限：

```
Databases
  AWSRAMPermissionGlueDatabaseReadWriteForCatalog
  AWSRAMPermissionGlueDatabaseReadWrite

Tables
  AWSRAMPermissionGlueTableReadWriteForCatalog
  AWSRAMPermissionGlueTableReadWriteForDatabase

AllTables
  AWSRAMPermissionGlueAllTablesReadWriteForCatalog
  AWSRAMPermissionGlueAllTablesReadWriteForDatabase
```

更新现有资源共享的 Amazon RAM 托管权限版本

您（数据湖管理员）可以按照《Amazon RAM 用户指南》中的说明将 Amazon RAM 托管权限更新到较新版本，也可以撤消资源类型的所有现有权限然后重新授予。如果您撤消权限，则 Amazon RAM 会删除与资源类型关联的 Amazon RAM 资源共享。如果您重新授予权限，Amazon RAM 会创建新的资源共享并附加最新版本的 Amazon RAM 托管权限。

插入、创建和变更表

不支持插入、创建和变更受 Lake Formation 策略保护的数据库中的表。如果执行这些操作，在尝试运行 Spark SQL 命令时，您会在笔记本条目的输出中看到下列异常。

```
java.io.IOException:
 com.amazon.ws.emr.hadoop.fs.shaded.com.amazonaws.services.s3.model.AmazonS3Exception:
           Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied; Request ID: …
```

有关更多信息，请参阅将 Amazon EMR 与 Amazon Lake Formation 集成的限制。

编写用于启动和管理集群的应用程序

主题

- 端到端 Amazon EMR Java 源代码示例
- API 调用的常见概念
- 使用 SDK 调用 Amazon EMR API
- 管理 Amazon EMR Service Quotas

通过调用 Amazon SDK 之一的包装函数，您可以访问 Amazon EMR API 提供的功能。Amazon 开发工具包提供特定语言的函数，这些函数包装 Web 服务的 API 并简化与 Web 服务的连接，从而为您处理许多连接详细信息。有关使用 SDK 之一调用 Amazon EMR 的更多信息，请参阅使用 SDK 调用 Amazon EMR API。

> **Important**
>
> Amazon EMR 的最大请求速度是每十秒钟一个请求。

端到端 Amazon EMR Java 源代码示例

开发人员可使用自定义 Java 代码调用 Amazon EMR API，以便执行与使用 Amazon EMR 控制台或 CLI 可能执行的相同操作。本节提供了完整的操作步骤，用以安装 Amazon Toolkit for Eclipse 和运行用于将步骤添加到 Amazon EMR 集群中的功能完备的 Java 源代码示例。

> **Note**
>
> 此示例的重点是 Java，但 Amazon EMR 还利用 Amazon EMR SDK 集合支持多种编程语言。有关更多信息，请参阅使用 SDK 调用 Amazon EMR API。

此 Java 源代码示例演示了如何使用 Amazon EMR API 执行以下任务：

- 检索Amazon凭证并将它们发送到 Amazon EMR 以进行 API 调用
- 配置新的自定义步骤和新的预定义步骤
- 将新步骤添加到现有 Amazon EMR 集群

- 从正在运行的集群检索集群步骤 ID

> **Note**
>
> 此示例演示了如何将步骤添加到现有集群，因此要求在您的账户上具有活动集群。

在您开始之前，请安装与您的计算机平台相匹配的 Eclipse IDE for Java EE Developers 版本。有关更多信息，请转到 Eclipse 下载。

下一步，安装适用于 Eclipse 的数据库开发插件。

安装数据库开发 Eclipse 插件

1. 打开 Eclipse IDE。
2. 依次选择 Help (帮助)、Install New Software (安装新软件)。
3. 在 Work with: (使用:) 字段中，键入 **`http://download.eclipse.org/releases/kepler`**，或者键入与您的 Eclipse IDE 版本号相匹配的路径。
4. 在项目列表中，依次选择 Database Development (数据库开发)、Finish (完成)。
5. 在提示时重新启动 Eclipse。

接下来，安装 Toolkit for Eclipse，以便提供实用的预配置源代码项目模板。

安装 Toolkit for Eclipse

1. 打开 Eclipse IDE。
2. 依次选择 Help (帮助)、Install New Software (安装新软件)。
3. 在 Work with: (使用:) 字段中，输入 **`https://aws.amazon.com/eclipse`**。
4. 在项目列表中，依次选择 Amazon Toolkit for Eclipse 和 Finish (完成)。
5. 在提示时重新启动 Eclipse。

下一步，创建一个新的 Amazon Java 项目，然后运行示例 Java 源代码。

创建新的 Amazon Java 项目

1. 打开 Eclipse IDE。

2. 依次选择 File (文件)、New (新建) 和 Other (其它)。
3. 在 Select a wizard (选择向导) 对话框中，依次选择 Amazon Java Project (亚马逊云科技 Java 项目)、Next (下一步)。
4. 在 New Amazon Java Project (新建亚马逊云科技 Java 项目) 对话框的 **Project name:** 字段中，输入新项目的名称，如 **EMR-sample-code**。
5. 选择 Configure Amazon accounts... (配置亚马逊云科技账户...)，输入您的公有和私有访问密钥，然后选择 Finish (完成)。有关创建访问密钥的更多信息，请参阅《亚马逊云科技一般参考》中的如何获取安全凭证？。

> **Note**
>
> 您不应直接在代码中嵌入访问密钥。使用 Amazon EMR SDK，可以在已知位置放置访问密钥，这样就不必保留在代码中。

6. 在新的 Java 项目中，右键单击 src 文件夹，然后选择 New (新建) 和 Class (类)。
7. 在 Java Class (Java 类) 对话框的 Name (名称) 字段中，输入新类的名称，例如 **main**。
8. 在 Which method stubs would you like to create? (您想创建哪些方法存根?) 部分中，依次选择 public static void main(String[] args) 和 Finish (完成)。
9. 在您的新类中输入 Java 源代码，然后添加示例中针对这些类和方法的相应 import 语句。为方便起见，以下显示了完整的源代码清单。

> **Note**
>
> 在以下示例代码中，将示例集群 ID (JobFlowId) *j-xxxxxxxxxxxx* 替换为您的账户中的有效集群 ID，可以在Amazon Web Services Management Console中找到该 ID 或使用以下 Amazon CLI 命令找到该 ID：
>
> ```
> aws emr list-clusters --active | grep "Id"
> ```
>
> 此外，将示例 Amazon S3 路径 *s3://path/to/my/jarfolder* 替换为您的 JAR 的有效路径。最后，将示例类名称 *com.my.Main1* 替换为您的 JAR 中的类的正确名称 (如果适用) 。

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
```

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

 public static void main(String[] args) {
  AWSCredentials credentials_profile = null;
  try {
   credentials_profile = new
 ProfileCredentialsProvider("default").getCredentials();
       } catch (Exception e) {
           throw new AmazonClientException(
                   "Cannot load credentials from .aws/credentials file. " +
                   "Make sure that the credentials file exists and the profile
 name is specified within it.",
                   e);
       }

  AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
   .withCredentials(new AWSStaticCredentialsProvider(credentials_profile))
   .withRegion(Regions.US_WEST_1)
   .build();

  // Run a bash script using a predefined step in the StepFactory helper class
     StepFactory stepFactory = new StepFactory();
     StepConfig runBashScript = new StepConfig()
       .withName("Run a bash script")
       .withHadoopJarStep(stepFactory.newScriptRunnerStep("s3://jeffgoll/emr-
scripts/create_users.sh"))
       .withActionOnFailure("CONTINUE");

     // Run a custom jar file as a step
     HadoopJarStepConfig hadoopConfig1 = new HadoopJarStepConfig()
        .withJar("s3://path/to/my/jarfolder") // replace with the location of the
 jar to run as a step
        .withMainClass("com.my.Main1") // optional main class, this can be omitted
 if jar above has a manifest
        .withArgs("--verbose"); // optional list of arguments to pass to the jar
     StepConfig myCustomJarStep = new StepConfig("RunHadoopJar", hadoopConfig1);
```

```
     AddJobFlowStepsResult result = emr.addJobFlowSteps(new
 AddJobFlowStepsRequest()
    .withJobFlowId("j-xxxxxxxxxxxx") // replace with cluster id to run the steps
    .withSteps(runBashScript,myCustomJarStep));

          System.out.println(result.getStepIds());

 }
}
```

10. 依次选择 Run (运行)、Run As (运行方式) 和 Java Application (Java 应用程序)。
11. 如果示例正确运行，则在 Eclipse IDE 控制台窗口中会显示这些新步骤的 ID 列表。正确的输出类似于：

```
[s-39BLQZRJB2E5E, s-1L6A4ZU2SAURC]
```

API 调用的常见概念

主题

- Amazon EMR 的端点
- 在 Amazon EMR 中指定集群参数
- Amazon EMR 中的可用区
- 如何在 Amazon EMR 集群中使用其它文件和库

如果编写调用 Amazon EMR API 的应用程序，在调用 SDK 的包装函数之一时需要了解几个概念。

Amazon EMR 的端点

终端节点是作为 Web 服务入口点的 URL。每个 Web 服务请求都必须包含一个终端节点。端点指定了创建、描述或终止集群所在的Amazon区域。其形式是 `elasticmapreduce.`*`regionname`*`.amazonaws.com`。如果您指定了通用端点（`elasticmapreduce.amazonaws.com`），Amazon EMR 会将您的请求转至默认区域的端点。对于 2013 年 3 月 8 日或之后创建的账户而言，默认区域是 us-west-2；对于较早的账户而言，默认区域是 us-east-1。

有关 Amazon EMR 的区域和端点的更多信息，请参阅《Amazon Web Services 一般参考》 中的 Regions and endpoints。

在 Amazon EMR 中指定集群参数

Instances 参数可让您配置 EC2 实例的类型和数量，以创建用于处理数据的节点。Hadoop 将数据的处理分布到多个集群节点。主节点负责跟踪核心和任务节点的运行状况并轮询节点的作业结果状态。核心和任务节点执行实际的数据处理。如果有一个单节点集群，则该节点同时用作主节点和核心节点。

KeepJobAlive 请求中的 RunJobFlow 参数用于确定在集群步骤执行完毕时是否终止该集群。当您知道该集群按预期运行时，将该值设置为 False。当您在集群执行暂停期间排除任务流程故障和添加步骤时，将该值设置为 True。这样可以减少向 Amazon Simple Storage Service（Amazon S3）上载结果的时间和费用，只需在修改重新启动此集群的步骤后重复该流程即可。

如果 KeepJobAlive 为 true，在成功地让集群完成其工作后，您必须发送一个 TerminateJobFlows 请求，或让集群继续运行并生成Amazon费用。

有关 RunJobFlow 所独有的参数的更多信息，请参见 RunJobFlow。有关该请求中通用参数的更多信息，请参见通用请求参数。

Amazon EMR 中的可用区

Amazon EMR 将 EC2 实例作为处理集群的节点。这些 EC2 实例的位置由可用区和区域组成。区域分散存在，位于独立的地理区域。可用区是在一个区域内的不同位置，与其它可用区的故障隔离。每个可用区向同一区域中的其它可用区提供低成本、低延迟的网络连接。有关 Amazon EMR 的区域和端点的列表，请参阅《Amazon Web Services 一般参考》 中的 Regions and endpoints。

AvailabilityZone 参数指定集群的一般位置。此参数是可选的，一般而言，我们不鼓励使用该参数。未指定 AvailabilityZone 时，Amazon EMR 为集群自动挑选最佳的 AvailabilityZone 值。如果想让您的实例和其它运行中的现有实例处于同一位置，而您的集群需要从那些实例中读取数据或写入数据，您可能会发现这个参数很有用。有关更多信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》。

如何在 Amazon EMR 集群中使用其它文件和库

有些时候，您可能喜欢通过您的映射器或 Reducer 应用程序使用其它文件或自定义库。例如，您可能喜欢使用把 PDF 文件转换成纯文本的库。

Hadoop 流式处理时缓存文件供映射器或 Reducer 使用

- 在 JAR args 字段中，添加以下参数：

```
-cacheFile s3://bucket/path_to_executable#local_path
```

文件 `local_path` 在映射器的工作目录中，映射器可以引用这个文件。

使用 SDK 调用 Amazon EMR API

主题

- 使用 Amazon SDK for Java 创建 Amazon EMR 集群

Amazon SDK 提供的函数包装 API 并关注许多连接详细信息，如计算签名、处理请求重试和错误处理。软件开发工具包也包含示例代码、教程和其它资源，以便帮助您开始编写调用 Amazon 的应用程序。调用软件开发工具包中的包装函数能大大简化编写 Amazon 应用程序的过程。

有关如何下载和使用 Amazon SDK 的更多信息，请参阅用于亚马逊云科技的工具下的 SDK。

使用 Amazon SDK for Java 创建 Amazon EMR 集群

Amazon SDK for Java 提供具有 Amazon EMR 功能的三个包：

- com.amazonaws.services.elasticmapreduce
- com.amazonaws.services.elasticmapreduce.model
- com.amazonaws.services.elasticmapreduce.util

有关这些包的更多信息，请参阅 Amazon SDK for Java API 参考。

以下示例说明了 SDK 如何使用 Amazon EMR 简化编程过程。下面的代码示例使用 `StepFactory` 对象（用于创建通用 Amazon EMR 步骤类型的帮助程序类）创建一个启用了调试的交互式 Hive 集群。

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

 public static void main(String[] args) {
```

```
 AWSCredentialsProvider profile = null;
 try {
  credentials_profile = new ProfileCredentialsProvider("default"); // specifies any
named profile in .aws/credentials as the credentials provider
       } catch (Exception e) {
           throw new AmazonClientException(
                   "Cannot load credentials from .aws/credentials file. " +
                   "Make sure that the credentials file exists and that the profile
name is defined within it.",
                   e);
       }

 // create an EMR client using the credentials and region specified in order to create
the cluster
 AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
  .withCredentials(credentials_profile)
  .withRegion(Regions.US_WEST_1)
  .build();

       // create a step to enable debugging in the AWS Management Console
 StepFactory stepFactory = new StepFactory();
 StepConfig enabledebugging = new StepConfig()
     .withName("Enable debugging")
     .withActionOnFailure("TERMINATE_JOB_FLOW")
     .withHadoopJarStep(stepFactory.newEnableDebuggingStep());

       // specify applications to be installed and configured when EMR creates the
cluster
 Application hive = new Application().withName("Hive");
 Application spark = new Application().withName("Spark");
 Application ganglia = new Application().withName("Ganglia");
 Application zeppelin = new Application().withName("Zeppelin");

 // create the cluster
 RunJobFlowRequest request = new RunJobFlowRequest()
         .withName("MyClusterCreatedFromJava")
         .withReleaseLabel("emr-5.20.0") // specifies the EMR release version label,
we recommend the latest release
         .withSteps(enabledebugging)
         .withApplications(hive,spark,ganglia,zeppelin)
         .withLogUri("s3://path/to/my/emr/logs") // a URI in S3 for log files is
required when debugging is enabled
         .withServiceRole("EMR_DefaultRole") // replace the default with a custom IAM
service role if one is used
```

```
          .withJobFlowRole("EMR_EC2_DefaultRole") // replace the default with a custom
 EMR role for the EC2 instance profile if one is used
          .withInstances(new JobFlowInstancesConfig()
              .withEc2SubnetId("subnet-12ab34c56")
              .withEc2KeyName("myEc2Key")
              .withInstanceCount(3)
              .withKeepJobFlowAliveWhenNoSteps(true)
              .withMasterInstanceType("m4.large")
              .withSlaveInstanceType("m4.large"));

    RunJobFlowResult result = emr.runJobFlow(request);
    System.out.println("The cluster ID is " + result.toString());

 }

}
```

至少，您必须传递分别对应于 EMR_DefaultRole 和 EMR_EC2_DefaultRole 的服务角色和任务流角色。可以通过为同一个账户调用此 Amazon CLI 命令来实现此目的。首先，查看这两个角色是否已存在：

```
aws iam list-roles | grep EMR
```

将显示实例配置文件 (EMR_EC2_DefaultRole) 和服务角色 (EMR_DefaultRole) (如果存在) ：

```
"RoleName": "EMR_DefaultRole",
            "Arn": "arn:aws:iam::AccountID:role/EMR_DefaultRole"
            "RoleName": "EMR_EC2_DefaultRole",
            "Arn": "arn:aws:iam::AccountID:role/EMR_EC2_DefaultRole"
```

如果默认角色不存在，则可以使用以下命令创建它们：

```
aws emr create-default-roles
```

管理 Amazon EMR Service Quotas

主题

- 什么是 Amazon EMR Service Quotas
- 如何管理 Amazon EMR Service Quotas

- 何时在 CloudWatch 中设置 EMR 事件

本节中的主题描述了 EMR Service Quotas（以前称为 Service Limits）、如何在Amazon Web Services Management Console中管理它们，以及何时使用 CloudWatch Events 而不是 Service Quotas 来监控集群和触发操作是有利的。

什么是 Amazon EMR Service Quotas

您的Amazon账户对于每项Amazon服务都具有默认的服务配额（以前称为限制）。EMR 服务有两种类型的限制：

- 资源限制 – 您可以使用 EMR 创建 EC2 资源。但是，这些 EC2 资源受服务配额限制。这一类别的资源限制如下：
  - 可以同时运行的最大活动集群数。
  - 每个实例组的最大活动实例数。
- API 限制 – 使用 EMR API 时，有两种限制：
  - 突增限制 – 这是您一次可以进行的 API 调用的最大数量。例如，您每秒可以发出的最大 AddInstanceFleet API 请求数默认设置为 5 次调用/秒。这意味着 AddInstanceFleet API 的突增限制为 5 次调用/秒，或者在任何给定时间，您最多可以进行 5 次 AddInstanceFleet API 调用。但是，使用突增限制后，您的后续调用将受到速率限制的限制。
  - 速率限制 – 这是 API 突增容量的补给速率。例如，AddInstanceFleet 调用的补给速率默认设置为 0.5 次调用/秒。这意味着在达到突增限制后，您必须等待至少 2 秒（0.5 次调用/秒 X 2 秒 = 1 次调用）才能进行 API 调用。如果在此之前进行调用，则会受 EMR Web 服务的节流。在任何时候，您只能发出与突增容量相同数量的调用，而不会受到节流。每多等待一秒，您的突增容量就会增加 0.5 次调用，直到达到最大限制 5，即突增限制。

如何管理 Amazon EMR Service Quotas

Service Quotas 是一项Amazon功能，您可以使用该功能从中央位置使用Amazon Web Services Management Console、API 或 CLI 查看和管理您的 Amazon EMR 服务配额（也称限制）。要了解有关查看配额和请求增加的更多信息，请参阅《Amazon Web Services 一般参考》 中的 Amazon service quotas。

对于某些 API，设置 CloudWatch 事件可能比增加服务配额更好。您还可以通过使用 CloudWatch 在达到服务配额之前主动设置警报和触发增加请求来节省时间。有关更多信息，请参阅 何时在 CloudWatch 中设置 EMR 事件。

何时在 CloudWatch 中设置 EMR 事件

对于某些轮询 API，例如 DescribeCluster、DescribeStep 和 ListClusters，设置 CanWatch 事件可以缩短对更改的响应时间并释放您的服务配额。例如，如果您将 Lambda 函数设置为在集群状态更改时运行（例如步骤完成或集群终止时），则可以使用该触发器启动工作流中的下一个操作，而不是等待下一次轮询。否则，如果您有专用的 Amazon EC2 实例或 Lambda 函数不断轮询 EMR API 以进行更改，您不仅会浪费计算资源，还可能会达到您的服务配额。

以下是您可以通过迁移到事件驱动架构而受益的几个案例。

案例 1：使用 DescribeCluster API 调用轮询 EMR 以完成步骤

Example 使用 DescribeCluster API 调用轮询 EMR 以完成步骤

一种常见模式是将步骤提交到正在运行的集群并轮询 Amazon EMR 以获取有关该步骤的状态，通常使用 DescribeCluster 或 DescribeStep API。此任务也可以通过挂接到 Amazon EMR 步骤状态更改事件以最小延迟完成。

此事件在其负载中包含以下信息。

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Step Status Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:53:09Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "ERROR",
    "actionOnFailure": "CONTINUE",
    "stepId": "s-ZYXWVUTSRQPON",
    "name": "CustomJAR",
    "clusterId": "j-123456789ABCD",
    "state": "FAILED",
    "message": "Step s-ZYXWVUTSRQPON (CustomJAR) in Amazon EMR cluster j-123456789ABCD
 (Development Cluster) failed at 2016-12-16 20:53 UTC."
  }
}
```

在细节贴图中，Lambda 函数可以解析“state”、“stepId”或“clusterId”以查找相关信息。

案例 2：轮询 EMR 获取可用集群以运行工作流

Example 轮询 EMR 获取可用集群以运行工作流

运行多个集群的客户的一种模式是在集群可用时立即在集群上运行工作流。如果有许多集群正在运行，并且需要在等待的集群上执行工作流，那么模式可能是使用 DescribeCluster 或 ListClusters API 调用轮询 EMR 以获取可用集群。减少知道集群何时已准备好执行某个步骤的延迟的另一种方法，是处理 Amazon EMR 集群状态更改事件。

此事件在其负载中包含以下信息。

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "WAITING",
    "message": "Amazon EMR cluster j-123456789ABCD ..."
  }
}
```

对于此事件，可以设置 Lambda 函数以便在集群状态更改为“WAITING (正在等待)”时立即将等待工作流发送到集群。

案例 3：轮询 EMR 以终止集群

Example 轮询 EMR 以终止集群

运行多个 EMR 集群的客户的一种常见模式是轮询 Amazon EMR 来查找已终止的集群，以便不再向其发送工作。您可以使用 DescribeCluster 和 ListClusters API 调用或使用 Amazon EMR 集群状态更改事件来实施此模式。

集群终止时，发出的事件类似于以下示例。

```
{
  "version": "0",
  "id": "1234abb0-f87e-1234-b7b6-000000123456",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T21:00:23Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user
 request\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "TERMINATED",
    "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has
 terminated at 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
  }
}
```

负载的“详细信息”部分包括可对其执行操作的 clusterId 和状态。

Amazon 术语表

有关最新的 Amazon 术语，请参阅《Amazon Web Services 词汇表参考》中的 Amazon 词汇表。