亚马逊云科技

应用程序负载均衡器

Elastic Load Balancing

Elastic Load Balancing: 应用程序负载均衡器

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 Application Load Balancer？

弹性负载均衡 在一个或多个可用区中的多个目标（如 EC2 实例、容器和 IP 地址）之间自动分配传入的流量。它会监控已注册目标的运行状况，并仅将流量传输到运行状况良好的目标。弹性负载均衡 根据传入流量随时间的变化对负载均衡器进行扩展。它可以自动扩展来处理绝大部分工作负载。

弹性负载均衡 支持以下负载均衡器：应用程序负载均衡器、网络负载均衡器、Gateway Load Balancer 和经典负载均衡器。您可以选择最适合自己需求的负载均衡器类型。本指南讨论 Application Load Balancer。有关其他负载均衡器的更多信息，请参阅网络负载均衡器用户指南、网关负载均衡器用户指南和 经典负载均衡器用户指南。

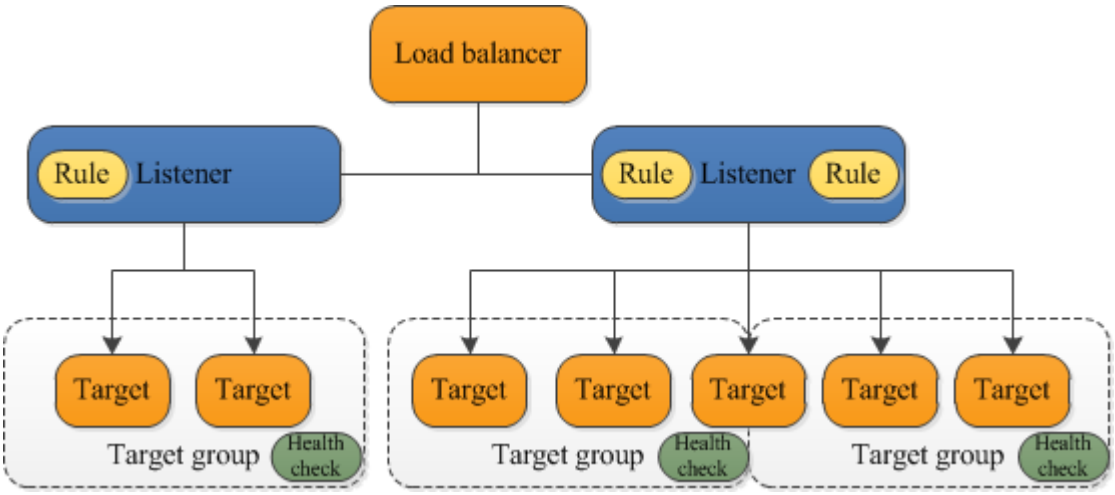
Application Load Balancer 组件

负载均衡器充当客户端的单一接触点。负载均衡器在多个可用区中的多个目标 (例如 EC2 实例) 间分配应用程序的传入流量。这将提高应用程序的可用性。可以向您的负载均衡器添加一个或多个侦听器。

侦听器使用您配置的协议和端口检查来自客户端的连接请求。您为侦听器定义的规则确定负载均衡器如何将请求路由到其已注册目标。每条规则由优先级、一个或多个操作以及一个或多个条件组成。当规则的条件满足时，将执行其操作。您必须为每个侦听器定义默认规则，并且可以选择定义其他规则。

每个目标组使用您指定的协议和端口号将请求路由到一个或多个注册目标，例如 EC2 实例。您可以向多个目标组注册一个目标。您可以对每个目标组配置运行状况检查。在注册到目标组 (它是使用负载均衡器的侦听器规则指定的) 的所有目标上，执行运行状况检查。

下图介绍基本组成部分。请注意，每个侦听器包含一个默认规则，并且一个侦听器包含将请求路由到不同目标组的另一条规则。向两个目标组注册一个目标。



有关更多信息，请参阅以下文档：

- 负载均衡器
- 侦听器
- 目标组

Application Load Balancer 概述

Application Load Balancer 在应用程序层正常工作，该层是开放系统互连 (OSI) 模型的第 7 层。负载均衡器收到请求后，将按照优先级顺序评估侦听器规则以确定应用哪个规则，然后从目标组中选择规则操作目标。可以配置侦听器规则，以根据应用程序流量的内容，将请求路由至不同的目标组。每个目标组的路由都是单独进行的，即使某个目标已在多个目标组中注册。可以配置目标组级别使用的路由算法。默认路由算法为轮询路由算法；或者，可以指定最少未完成请求路由算法。

可以根据需求变化在负载均衡器中添加和删除目标，而不会中断应用程序的整体请求流。弹性负载均衡根据传输到应用程序的流量随时间的变化对负载均衡器进行扩展。弹性负载均衡 能够自动扩展来处理绝大部分工作负载。

您可以配置运行状况检查，这些检查可用来监控注册目标的运行状况，以便负载均衡器只能将请求发送到正常运行的目标。

有关更多信息，请参阅 弹性负载均衡 用户指南中的 Elastic Load Balancing 工作原理

从经典负载均衡器迁移的好处

使用 Application Load Balancer 而不是经典负载均衡器具有以下好处：

- 支持 路径条件。对于根据请求中的 URL 转发请求的侦听器，您可以为它配置规则。这让您可以将应用程序构造为较小的服务，并根据 URL 内容将请求路由到正确的服务。
- 支持 主机条件。对于基于 HTTP 标头中主机字段转发请求的侦听器，您可以为它配置规则。这使您能够使用单个负载均衡器将请求路由到多个域。
- 支持基于请求中的字段进行路由，例如 HTTP 标头条件 和方法、查询参数和源 IP 地址。
- 支持将请求路由到单个 EC2 实例上的多个应用程序。您可以向多个目标组注册实例或 IP 地址，每个目标组都位于不同的端口。
- 支持将请求从一个 URL 重定向到另一个 URL。
- 支持返回自定义 HTTP 响应。

- 支持通过 IP 地址注册目标，包括位于负载均衡器的 VPC 之外的目标。
- 支持将 Lambda 函数注册为目标。
- 支持负载均衡器在路由请求之前使用应用程序用户的企业或社交身份对这些用户进行身份验证。
- 支持容器化的应用程序。计划任务时，Amazon Elastic Container Service (Amazon ECS) 可以选择一个未使用的端口，并可以使用此端口向目标组注册该任务。这样可以高效地使用您的群集。
- Support 支持独立监控每项服务的运行状况，因为运行状况检查是在目标组级别定义的，许多 CloudWatch 指标是在目标组级别报告的。将目标组挂载到 Auto Scaling 组的功能使您能够根据需求动态扩展每个服务。
- 访问日志包含附加信息，并以压缩格式存储。
- 已改进负载均衡器性能。

相关服务

弹性负载均衡 可与以下服务一起使用，以提高应用程序的可用性和可扩展性。

- Amazon EC2 — 在云中运行应用程序的虚拟服务器。您可以将负载均衡器配置为将流量路由到您的 EC2 实例。
- Amazon EC2 Auto Scaling - 确保运行所需数量的实例（即使实例失败也是如此），并可让您根据实例需求的变化自动增加或减少实例数。如果您使用弹性负载均衡启用 Auto Scaling，则 Auto Scaling 启动的实例将自动向目标组注册，并且 Auto Scaling 终止的实例将自动从目标组注销。
- Amazon Certificate Manager – 在创建 HTTPS 侦听器时，您必须指定由 ACM 提供的证书。负载均衡器使用证书终止连接并解密来自客户端的请求。有关更多信息，请参阅 应用程序负载均衡器的 SSL 证书。
- Amazon CloudWatch — 使您能够监控您的负载均衡器并根据需要采取行动。有关更多信息，请参阅 CloudWatch Application Load Balancer 的指标。
- Amazon ECS — 使您能够在 EC2 实例集群上运行、停止和管理 Docker 容器。您可以将负载均衡器配置为将流量路由到您的容器。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的服务负载均衡。
- Amazon Global Accelerator — 提高应用程序的可用性和性能。使用加速器在一个或多个 Amazon 区域的多个负载均衡器之间分配流量。有关更多信息，请参见Amazon Global Accelerator 开发人员指南。
- Route 53 — 通过将域名（例如）转换为计算机用于相互连接的数字 IP 地址（例如`www.example.com`），提供一种可靠且经济实惠的方式，将访问者引导到网站。`192.0.2.1` Amazon 分配 URLs 给您的资源，例如负载均衡器。不过，您可能希望使用方便用户记忆的 URL。

例如，您可以将域名映射到负载均衡器。有关更多信息，请参阅 Amazon Route 53 开发人员指南中的将流量路由到 ELB 负载均衡器。

- Amazon WAF— 您可以 Amazon WAF 与 Application Load Balancer 配合使用，根据网络访问控制列表 (Web ACL) 中的规则允许或阻止请求。有关更多信息，请参阅 Amazon WAF。

要查看有关与您的负载均衡器集成的服务的信息，请在中选择您的负载均衡器， Amazon Web Services 管理控制台 然后选择集成服务选项卡。

定价

利用负载均衡器，您可以按实际用量付费。有关更多信息，请参阅 弹性负载均衡 定价。

Application Load Balancer

负载均衡器充当客户端的单一接触点。客户端将请求发送到负载均衡器，负载均衡器再将它们发送到具体目标（例如 EC2 实例）。要配置您的负载均衡器，可以创建目标组，然后将目标注册到目标组。您还可以创建侦听器来检查来自客户端的连接请求，并创建侦听器规则以将来自客户端的请求路由到一个或多个目标组中的目标。

有关更多信息，请参阅 Elastic Load Balancing 用户指南中的 Elastic Load Balancing 工作原理

目录

- 应用程序负载均衡器的集成

您的负载均衡器的子网

创建应用程序负载均衡器时，您必须启用包含您的目标的区域。若要启用区域，请在区域中指定一个子网。弹性负载均衡在您指定的每个区域中创建一个负载均衡器节点。

注意事项

- 当您确保每个启用的区域均具有至少一个已注册目标时，负载均衡器是最有效的。
- 如果您在某区域中注册目标，但未启用该区域，这些已注册目标将无法从负载均衡器接收流量。
- 如果为负载均衡器启用多个区域，则这些区域的类型必须相同。例如，不能同时启用可用区和本地区域。
- 您可以指定已与您共享的子网。
- 弹性负载均衡会在您配置该负载均衡器的子网中创建网络接口。这些网络接口是预留的，因此即使子网的可用 IP 地址不足，负载均衡器也可以完成维护操作。此类网络接口的描述为“ELB 为子网预留的 ENI”。

应用程序负载均衡器支持以下类型的子网。

子网类型

- 可用区子网
- 本地区域子网
- Outpost 子网

可用区子网

您必须至少选择两个可用区子网。以下限制适用：

- 每个子网都必须来自不同的可用区。
- 要确保您的负载均衡器可以正确扩展，请验证负载均衡器的每个可用区子网是否都具有至少带有一个 /27 位掩码的 CIDR 数据块（例如，10.0.0.0/27）以及每个子网至少八个空闲 IP 地址。这八个 IP 地址是允许负载均衡器在需要时进行横向扩展所必需的。您的负载均衡器将使用这些 IP 地址与目标建立连接。没有它们，应用程序负载均衡器在尝试更换节点时可能会遇到困难，从而导致其进入失败状态。

注意：如果应用程序负载均衡器子网在尝试扩展时用尽可用的 IP 地址，则应用程序负载均衡器将在容量不足的情况下运行。在此期间，旧节点会继续为流量提供服务，但在尝试建立连接时，停滞的扩展尝试可能会导致 5xx 错误或超时。

本地区域子网

您可以指定本地区域子网。本地区域子网不支持以下功能：

- Lambda 函数即目标
- 双向 TLS 身份验证
- Amazon WAF 整合

Outpost 子网

您可以指定单个 Outpost 子网。以下限制适用：

- 您必须已在本地数据中心中安装并配置了 Outpost。Outpost 与其 Amazon 区域之间必须具有可靠的网络连接。有关更多信息，请参阅 Amazon Outposts 用户指南。
- 负载均衡器需要在 Outpost 上为负载均衡器节点设置两个 `large` 实例。支持的实例类型见下表。负载均衡器可根据需要进行扩展，每次可将节点大小调整一个型号（从 `large` 到`xlarge`，然后从 `xlarge` 到 `2xlarge`，然后从 `2xlarge` 到 `4xlarge`）。将节点扩展到最大实例型号后，如果您还需要额外的容量，负载均衡器会添加 `4xlarge` 实例以作为负载均衡器节点。如果您没有足够的实例容量或可用 IP 地址来扩展负载均衡器，负载均衡器将向 Amazon Health Dashboard 报告事件，并且负载均衡器状态为 `active_impaired`。
- 您可以通过实例 ID 或 IP 地址注册目标。如果您在 Amazon 区域内为前哨基地注册目标，则不会使用这些目标。
- 不支持以下功能：
  - Amazon Global Accelerator 整合
  - Lambda 函数即目标
  - 双向 TLS 身份验证
  - 粘性会话
  - 用户身份验证
  - Amazon WAF 整合
  - 目标优化器

- Health 检查日志
- 连接日志
- 容量单位预留
- JWT 验证
- 自动目标权重
- FIPS 安全策略

Application Load Balancer 可以部署在 c5/c5d m5/m5d、或 Outpos r5/r5d t 上的实例上。下表显示了负载均衡器在 Outpost 上可以使用的每个实例类型的大小和 EBS 卷：

实例类型和大小	EBS 卷 (GB)
c5/c5d	
large	50
xlarge	50
2xlarge	50
4xlarge	100
m5/m5d	
large	50
xlarge	50
2xlarge	100
4xlarge	100
r5/r5d	
large	50
xlarge	100
2xlarge	100

实例类型和大小	EBS 卷 (GB)	
4xlarge	100	

负载均衡器安全组

安全组起到防火墙的作用，可控制允许往返于负载均衡器的流量。您可以选择端口和协议以允许入站和出站流量。

与负载均衡器关联的安全组的规则必须允许侦听器和运行状况检查端口上的双向流量。当您将侦听器添加到负载均衡器或更新目标组的运行状况检查端口时，您必须检查您的安全组规则，确保它们允许新端口上的双向流量。有关更多信息，请参阅 推荐的规则。

负载均衡器状态

负载均衡器可能处于下列状态之一：

`provisioning`

正在设置负载均衡器。

`active`

负载均衡器已完全设置并准备好路由流量。

`active_impaired`

负载均衡器正在路由流量，但没有扩展所需的资源。

`failed`

负载均衡器无法设置。

负载均衡器属性

您可以通过编辑应用程序负载均衡器的属性对其进行配置。有关更多信息，请参阅 编辑负载均衡器属性。

以下是负载均衡器属性：

`access_logs.s3.enabled`

指示是否启用存储在 Amazon S3 中的访问日志。默认值为 `false`。

`access_logs.s3.bucket`

访问日志所用的 Amazon S3 存储桶的名称。如果启用访问日志，则此属性是必需的。有关更多信息，请参阅 启用访问日志。

`access_logs.s3.prefix`

Amazon S3 存储桶中位置的前缀。

`client_keep_alive.seconds`

客户端保持连接值（以秒为单位）。默认值为 3600 秒。

`deletion_protection.enabled`

指示是否启用删除保护。默认为 `false`。

`idle_timeout.timeout_seconds`

空闲超时值 (以秒为单位)。默认值为 60 秒。

`ipv6.deny_all_igw_traffic`

阻止 Internet 网关 (IGW) 访问负载均衡器，以防通过 Internet 网关意外访问内部负载均衡器。对于面向互联网的负载均衡器，它设置为 `false`；对于内部负载均衡器，它设置为 `true`。此属性不会阻止非 IGW 互联网访问（例如，通过对等互连、Transit Gateway 或 Amazon Direct Connect）。Amazon VPN

`routing.http.desync_mitigation_mode`

确定负载均衡器如何处理可能对您的应用程序构成安全风险的请求。可能的值为 `monitor`、`defensive` 和 `strictest`。默认为 `defensive`。

`routing.http.drop_invalid_header_fields.enabled`

指示具有无效标头字段的 HTTP 标头是被负载均衡器删除 (`true`) 还是路由到目标 (`false`)。默认为 `false`。Elastic Load Balancing 要求有效的 HTTP 标头名称符合正则表达式 `[-A-Za-z0-9]+`，如 HTTP 字段名注册表中所述。每个名称都由字母数字字符或连字符组成。如果您想从请求中删除不符合此模式的 HTTP 标头，请选择 `true`。

`routing.http.preserve_host_header.enabled`

指示应用程序负载均衡器是否应保留 HTTP 请求中的 `Host` 标头，并将请求发送到目标而不作任何更改。可能的值为 `true` 和 `false`。默认为 `false`。

`routing.http.x_amzn_tls_version_and_cipher_suite.enabled`

指示两个标头（`x-amzn-tls-version` 和 `x-amzn-tls-cipher-suite`）在发送到目标之前是否将被添加到客户端请求，标头中包含有关协商 TLS 版本和密码套件的信息。`x-amzn-tls-version` 标头包含有关与客户端协商的 TLS 协议版本的信息，`x-amzn-tls-cipher-suite` 标头包含有关与客户端协商的密码套件的信息。两个标头都采用 OpenSSL 格式。属性的可能值为 `true` 和 `false`。默认为 `false`。

`routing.http.xff_client_port.enabled`

指示 `X-Forwarded-For` 标头是否应保留客户端用于连接负载均衡器的源端口。可能的值为 `true` 和 `false`。默认为 `false`。

`routing.http.xff_header_processing.mode`

这让您可以在应用程序负载均衡器将请求发送到目标之前修改、保留或移除 HTTP 请求中的 `X-Forwarded-For` 标头。可能的值为 `append`、`preserve` 和 `remove`。默认为 `append`。

- 如果该值为 `append`，则应用程序负载均衡器会将客户端 IP 地址（最后一跳）添加到 HTTP 请求的 `X-Forwarded-For` 标头，然后再将请求发送到目标。
- 如果该值为 `preserve`，则应用程序负载均衡器会保留 HTTP 请求中的 `X-Forwarded-For` 标头，并将请求发送到目标而不作任何更改。
- 如果该值为 `remove`，则应用程序负载均衡器会移除 HTTP 请求中的 `X-Forwarded-For` 标头，然后再将请求发送到目标。

`routing.http2.enabled`

表示客户端是否可以使用连接到负载均衡器 HTTP/2。如果`true`是，则客户端可以使用 HTTP/2 或进行连接 HTTP/1.1。如果是`false`，则客户端必须使用进行连接 HTTP/1.1。默认值为 `true`。

`waf.fail_open.enabled`

表示如果 Amazon WAF启用了该功能的负载均衡器无法将请求转发到目标，则是否允许其将请求路由到 Amazon WAF目标。可能的值为 `true` 和 `false`。默认为 `false`。

> **Note**
>
> 引入了 `routing.http.drop_invalid_header_fields.enabled` 属性以提供 HTTP 不同步保护。添加 `routing.http.desync_mitigation_mode` 属性以为您的应用程序提供更全面的保护，使其免受 HTTP 不同步的影响。您无需同时使用这两个属性，可以选择最适合应用程序要求的属性。

IP 地址类型

您可以设置客户端可用于访问面向 Internet 和内部的负载均衡器的 IP 地址类型。

应用程序负载均衡器支持以下 IP 地址类型：

ipv4

客户端必须使用 IPv4 地址连接到负载均衡器（例如 192.0.2.1）。

dualstack

客户端可以同时使用 IPv4 地址（例如 192.0.2.1）和 IPv6 地址（例如，2001:0db8:85a3:0:0:8a2e:0370:7334）连接到负载均衡器。

dualstack-without-public-ipv4

客户端必须使用 IPv6 地址（例如，2001:0db8:85a3:0:0:8a2e:0370:7334）连接到负载均衡器。

注意事项

- 负载均衡器根据目标组的 IP 地址类型与目标进行通信。
- 当您为负载均衡器启用双堆栈模式时，Elastic Load Balancing 为负载均衡器提供 AAAA DNS 记录。使用 IPv4 地址与负载均衡器通信的客户端解析 A DNS 记录。使用 IPv6 地址与负载均衡器通信的客户端解析 AAAA DNS 记录。
- 阻止通过互联网网关对内部双堆栈负载均衡器的访问，以防意外访问互联网。但是，这并不能阻止非 IGW 互联网访问（例如，通过对等互连、Transit Gateway 或 Amazon Direct Connect）。 Amazon VPN
- 连接到身份提供者（IdP）或 Amazon Cognito 端点时，应用程序负载均衡器身份验证仅支持 IPv4。如果没有公有 IPv4 地址，负载均衡器就无法完成身份验证过程，从而导致 HTTP 500 错误。

有关更多信息，请参阅 更新应用程序负载均衡器的 IP 地址类型。

Application Load Balancer IP 地址管理

应用程序负载均衡器使用 EC2 公有 IPv4 地址池中的公有弹性 IPv4 地址。使用描述地址 CLI、API 或在控制台中查看弹性 IP (EIP) 部分时，这些 IP 地址会在您的 Amazon 账户中看到。 Amazon 每个 ALB-associated IP 地址都标有设置为 “ALB” 的 service_managed 属性。

虽然这些 IP 在您的账户中可见，但它们仍由 Application Load Balancer 服务完全管理，无法修改或发布。当不再使用时，Application Load Balancer 将 IP 释放回公有 IPv4 地址池。

CloudTrail 记录与应用程序负载均衡器的 EIP 相关的 API 调用，例如 “AllocateAddress”。这些 API 调用由服务负责人 “elasticloadbalancing.amazonaws.com” 调用。

> **Note**
>
> 注意：Application Load Balancer 分配的 IP 不计入您账户的 EIP 限制。

IPAM IP 地址池

IPAM IP 地址池是您使用 Amazon VPC IP 地址管理器（IPAM）创建的连续 IP 地址范围（或 CIDR）的集合。将 IPAM IP 地址池与应用程序负载均衡器结合使用，可让您根据路由和安全需求组织 IPv4 地址。IPAM IP 地址池允许您选择将部分或全部公有 IPv4 地址范围引入应用程序负载均衡器，并将其与应用程序 Amazon 负载均衡器一起使用。启动 EC2 实例和创建应用程序负载均衡器时，始终会优先考虑您的 IPAM IP 地址池。当您的 IP 地址不再使用时，将会立即可供再次使用。

要开始使用此功能，首先要创建一个 IPAM IP 地址池。有关更多信息，请参阅将 IP 地址带入 IPAM。

注意事项

- 不支持 IPAM IPv6 地址池。
- 内部负载均衡器或 `dualstack-without-public-ipv4` IP 地址类型不支持 IPAM IPv4 地址池。
- 如果某个负载均衡器当前正在使用 IPAM IP 地址池中的 IP 地址，则无法将其删除。
- 在过渡到其他 IPAM IP 地址池期间，现有连接将根据负载均衡器 HTTP 客户端的保持连接持续时间终止。
- IPAM IP 地址池可以在多个账户之间共享。有关更多信息，请参阅为 IPAM 配置集成选项。
- 将 IPAM IP 地址池与负载均衡器结合使用不会产生额外费用。不过可能会产生与 IPAM 相关的费用，具体取决于您使用的套餐。

  如果您的 IPAM IP 地址池中没有其他可分配的 IP 地址，Elastic Load Balancing 将改用 Amazon 托管 IPv4 地址。使用 Amazon 托管式 IPv4 地址会产生额外的费用。为避免这些成本，您可以将 IP 地址范围添加到现有的 IPAM IP 地址池中。

  有关更多信息，请参阅 Amazon VPC pricing。

负载均衡器连接

在处理请求时，负载均衡器会维护两个连接：一个与客户端的连接和一个与目标的连接。负载均衡器与客户端之间的连接也称为前端连接。负载均衡器与目标之间的连接也称为后端连接。

Cross-zone 负载平衡

对于应用程序负载均衡器，默认情况下启用跨可用区负载均衡，无法在负载均衡器级别进行更改。有关更多信息，请参阅《Elastic Load Balancing 用户指南》中的负Cross-zone 载平衡部分。

可以在目标组级别关闭跨可用区负载均衡。有关更多信息，请参阅 the section called “关闭跨可用区负载均衡”。

DNS 名称

每个 Application Load Balancer 都会收到一个默认的域名系统 (DNS) 名称，其语法如下：*name-id* .elb。 *region*.amazonaws.com。例如，my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com。

如果更喜欢使用更容易记住的 DNS 名称，则可以创建自定义域名并将其与应用程序负载均衡器的 DNS 名称相关联。在客户端使用此自定义域名进行请求时，DNS 服务器将它解析为应用程序负载均衡器的 DNS 名称。

首先，向经认可的域名注册商注册域名。接下来，使用您的 DNS 服务（如您的域注册商）创建一条 DNS 记录将请求路由到您的应用程序负载均衡器。有关更多信息，请参阅您的 DNS 服务的文档。例如，如果您将 Amazon Route 53 用作 DNS 服务，请创建一条指向应用程序负载均衡器的别名记录。有关更多信息，请参阅 Amazon Route 53 开发人员指南中的将流量路由到 ELB 负载均衡器。

应用程序负载均衡器针对每个启用的可用区都有一个 IP 地址。这些是应用程序负载均衡器节点的 IP 地址。应用程序负载均衡器的 DNS 名称解析为这些地址。例如，假设您的应用程序负载均衡器的自定义域名是 `example.applicationloadbalancer.com`。使用以下 dig 或 nslookup 命令确定应用程序负载均衡器节点的 IP 地址。

Linux 或 Mac

```
$ dig +short example.applicationloadbalancer.com
```

Windows

```
C:\> nslookup example.applicationloadbalancer.com
```

应用程序负载均衡器具有其节点的 DNS 记录。您可以使用具有以下语法的 DNS 名称来确定 Application Load Balancer 节点的 IP 地址：*az*。 *name-id* .elb。 *region*.amazonaws.com。

Linux 或 Mac

```
$ dig +short us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

Windows

```
C:\> nslookup us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

创建 Application Load Balancer

应用程序负载均衡器接收来自客户端的请求，然后将请求分发给目标组中的目标（例如 EC2 实例）。有关更多信息，请参阅《弹性负载均衡用户指南》中的弹性负载均衡的工作原理。

任务

- 先决条件
- 创建负载均衡器
- 测试负载均衡器
- 后续步骤

先决条件

- 确定应用程序将支持的可用区和 IP 地址类型。在负载均衡器 VPC 以及在每个可用区的子网。如果应用程序同时支持 IPv4 和 IPv6 流量，请确保这些子网同时具有 IPv4 和 IPv6 CIDR。在每个可用区中至少部署一个目标。有关更多信息，请参阅 the section called “您的负载均衡器的子网”。
- 确保目标实例的安全组允许来自客户端 IP 地址（如果目标通过实例 ID 指定）或负载均衡器节点（如果目标通过 IP 地址指定）的流量通过侦听器端口。有关更多信息，请参阅 推荐的规则。
- 确保目标实例的安全组允许来自负载均衡器的流量使用运行状况检查协议通过运行状况检查端口。

创建负载均衡器

在创建应用程序负载均衡器的过程中，您将创建负载均衡器、至少一个侦听器和至少一个目标组。负载均衡器在每个已启用的可用区内至少有一个运行正常的已注册目标时，即表示其做好了处理客户端请求的准备。

Console

创建应用程序负载均衡器

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择创建负载均衡器。
4. 在 Application Load Balancer 下，选择 Create（创建）。
5. 基本配置
   a. 对于Load balancer name（负载均衡器名称），输入负载均衡器的名称。名称在该区域的负载均衡器集合中必须唯一。名称最多可包含 32 个字符，并且只能包含字母数字字符和连字符。它们不能以连字符或 `internal-` 开头或结尾。应用程序负载均衡器创建后将不能更改其名称。
   b. 对于方案，选择Internet-facing或内部。面向互联网的负载均衡器将来自客户端的请求通过互联网路由到目标。内部负载均衡器使用私有 IP 地址将请求路由到目标。
   c. 对于负载均衡器 IP 地址类型，如果客户端使用 IPv4 地址与负载均衡器通信，请选择 IPv4；如果客户端同时使用 IPv4 和 IPv6 地址与负载均衡器通信，则选择双栈。如果您的客户端仅使用 IPv6 地址与负载均衡器通信，请选择不带公有 IPv4 的双堆栈。
6. 网络映射
   a. 对于 VPC，请选择您为负载均衡器准备的 VPC。对于面向互联网的负载均衡器，仅具有互联网网关的 VPC 可供选择。
   b. （可选）对于 IP 池，您可以选择使用公有 IPv4 地址的 IPAM 池。有关更多信息，请参阅 the section called “IPAM IP 地址池”。
   c. 对于可用区和子网，按以下方式为负载均衡器启用可用区：
      - 选择至少来自两个可用区的子网
      - 选择至少一个本地区域中的子网
      - 选择一个 Outpost 子网

有关更多信息，请参阅 the section called “您的负载均衡器的子网”。

对于双栈负载均衡器，请选择同时具有 IPv4 和 IPv6 CIDR 数据块的子网。

7. 安全组

我们会为负载均衡器 VPC 预选默认安全组。您可以根据需要选择其他安全组。如果您没有可满足您需求的安全组，请选择创建新的安全组，以立即创建一个。有关更多信息，请参阅《Amazon VPC 用户指南》中的创建安全组。

8. 侦听器和路由

a. 默认值是一个通过端口 80 接受 HTTP 流量的侦听器。您可保留默认侦听器设置，或者根据需要修改协议和端口。

b. 对于默认操作，选择目标组以转发流量。如果没有能满足您需求的目标组，请选择创建目标组，以立即创建一个目标组。有关更多信息，请参阅 创建目标组。

c. （可选）选择添加侦听器标签，然后输入标签键和标签值。

d. （可选）选择添加侦听器以添加其他侦听器（例如，HTTPS 侦听器）。

9. 安全侦听器设置

仅当您添加 HTTPS 侦听器时才会显示此部分。

a. 对于安全策略，请选择符合您要求的安全策略。有关更多信息，请参阅 安全策略。

b. 对于默认 SSL/TLS 证书，有以下选项可用：

- 如果您使用创建或导入了证书 Amazon Certificate Manager，请选择 From ACM，然后选择证书。
- 如果您使用 IAM 导入了一个证书，则选择从 IAM，然后选择您的证书。
- 如果您在 ACM 中没有可用的证书，但有可用于负载均衡器的证书，请选择导入证书，并提供所需的信息。否则，请选择请求新的 ACM 证书。有关更多信息，请参阅《Amazon Certificate Manager 用户指南》中的 Amazon Certificate Manager 证书 certificates。

c. （可选）选择双向认证（mTLS），然后选择一个策略以启用 ALPN。

有关更多信息，请参阅 双向 TLS 身份验证。

10. 通过服务集成进行优化

（可选）您可以将 other Amazon 与您的负载均衡器集成。有关更多信息，请参阅 负载均衡器集成。

11. 负载均衡器标签

    （可选）展开负载均衡器标签。（可选）选择添加新的标签，然后输入标签键和标签值。有关更多信息，请参阅标签。

12. 摘要

    查看配置，然后选择创建负载均衡器。在创建过程中，一些默认属性会应用于网络负载均衡器。创建网络负载均衡器后，您可以查看和编辑它们。有关更多信息，请参阅 负载均衡器属性。

Amazon CLI

创建应用程序负载均衡器

使用 create-load-balancer 命令。

以下示例创建了一个面向互联网的负载均衡器，包括两个已启用的可用区和一个安全组。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type application \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

创建内部应用程序负载均衡器

包含如下示例所示的 --scheme 选项。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type application \
    --scheme internal \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

创建双栈应用程序负载均衡器

包含如下示例所示的 `--ip-address-type` 选项。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type application \
    --ip-address-type dualstack \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

添加侦听器

使用 create-listener 命令。有关示例，请参阅 创建 HTTP 侦听器 和 创建 HTTPS 侦听器。

CloudFormation

创建应用程序负载均衡器

定义类型为 AWS::ElasticLoadBalancingV2:: LoadBalancer 的资源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      Tags:
        - Key: "department"
          Value: "123"
```

添加侦听器

定义类型为 AWS::ElasticLoadBalancingV2:: Listener 的资源。有关示例，请参阅 创建 HTTP 侦听器 和 创建 HTTPS 侦听器。

测试负载均衡器

在创建负载均衡器之后，您可验证您的 EC2 实例是否通过了初始运行状况检查。然后，您可检查负载均衡器是否正在将流量发送到您的 EC2 实例。要删除负载均衡器，请参阅 删除 Application Load Balancer。

要测试负载均衡器

1. 创建负载均衡器之后，选择关闭。
2. 在导航窗格中，选择目标组。
3. 选择新创建的目标组。
4. 选择 Targets (目标) 并验证您的实例是否已就绪。如果实例的状态为 `initial`，通常是因为该实例仍在注册过程中。此状态还可以表示实例未通过被视为正常运行所需的最少次数的运行状况检查。在至少一个实例的状态为正常后，便可测试负载均衡器。有关更多信息，请参阅 目标运行状况。
5. 在导航窗格中，选择负载均衡器。
6. 选择新创建的负载均衡器。
7. 选择 Description（描述）并复制面向互联网的负载均衡器或内部负载均衡器的 DNS 名称（例如，my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com）。

   - 对于面向互联网的负载均衡器，请将 DNS 名称粘贴到连接互联网的 Web 浏览器的地址字段中。
   - 对于内部负载均衡器，请将 DNS 名称粘贴到与 VPC 具有私有连接的 Web 浏览器的地址字段中。

   如果所有内容的配置都正确，浏览器会显示您服务器的默认页面。
8. 如果网页未显示，请参阅以下文档以获取其他配置帮助和故障排除步骤。

   - 对于与 DNS 相关的问题，请参阅《Amazon Route 53 开发人员指南》中的将流量路由到 ELB 负载均衡器。
   - 对于与负载均衡器相关的问题，请参阅 对 Application Load Balancer 进行故障排除。

后续步骤

创建负载均衡器后，您可能需要执行以下操作：

- 添加侦听器规则。
- 配置负载均衡器属性。
- 配置目标组属性。
- [HTTPS 侦听器] 将证书添加到可选证书列表。
- 配置监控功能。

更新应用程序负载均衡器的可用区

您可随时启用或禁用负载均衡器的可用区。在启用一个可用区后，负载均衡器会开始将请求路由到该可用区中的已注册目标。应用程序负载均衡器会默认开启跨可用区负载均衡，从而将请求路由到所有可用区中的所有已注册目标。关闭跨可用区负载均衡时，负载均衡器仅会将请求路由到同一可用区中的目标。有关更多信息，请参阅 Cross-zone 负载平衡。如果您确保每个启用的可用区均具有至少一个注册目标，则负载均衡器将具有最高效率。

在禁用一个可用区后，该可用区中的目标仍将注册到负载均衡器，但负载均衡器不会向这些目标路由请求。

有关更多信息，请参阅 the section called “您的负载均衡器的子网”。

Console

更新可用区

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在网络映射选项卡上，选择编辑子网。
5. 要启用可用区，请选中其复选框并选择一个子网。如果只有一个可用区，则会选择此子网。
6. 要更改已启用的可用区的子网，请从列表中选择其他子网之一。
7. 要禁用可用区，请清除其复选框。
8. 选择保存更改。

Amazon CLI

更新可用区

使用 set-subnets 命令。

```
aws elbv2 set-subnets \
    --load-balancer-arn load-balancer-arn \
    --subnets subnet-8360a9e7EXAMPLE subnet-b7d581c0EXAMPLE
```

CloudFormation

更新可用区

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref new-subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

Application Load Balancer 的安全组

应用程序负载均衡器的安全组控制允许到达和离开负载均衡器的流量。您必须确保负载均衡器可同时在侦听器端口和运行状况检查端口上与已注册目标进行通信。当您将侦听器添加到负载均衡器或更新负载均衡器所使用的目标组的运行状况检查端口来路由请求时，您必须验证与负载均衡器关联的安全组是否允许新端口上的双向流量。如果它们不允许，您可以编辑当前关联的安全组的规则或将其他安全组与负载均衡器关联。您可以选择允许的端口和协议。例如，您可以打开负载均衡器的 Internet 控制消息协议 (ICMP) 连接，以响应 Ping 请求 (但是，Ping 请求不会转发至任何实例)。

注意事项

- 为确保您的目标仅接收来自负载均衡器的流量，请限制与目标关联的安全组仅接受来自负载均衡器的流量。这可以通过在目标安全组的入口规则中将负载均衡器的安全组设置为源来实现。

- 如果您的应用程序负载均衡器是某个网络负载均衡器的目标，则该应用程序负载均衡器的安全组会使用连接跟踪来跟踪有关来自该网络负载均衡器的流量的信息。无论为 Application Load Balancer 设置的安全组规则如何，都会执行此跟踪 有关更多信息，请参阅《Amazon EC2 用户指南》中的安全组连接跟踪。
- 我们建议您允许入站 ICMP 流量以支持路径 MTU 发现。有关更多信息，请参阅《Amazon EC2 用户指南》中的路径 MTU 发现。

推荐的规则

对于将实例作为目标的面向互联网的负载均衡器，建议使用以下规则。

Inbound

Source	Port Range	Comment
0.0.0。 0/0	*listener*	在负载均衡器侦听器端口上允许所有入站流量

Outbound

Destination	Port Range	Comment
instance security group	*instance listener*	在实例侦听器端口上允许流向实例的出站流量
instance security group	*health check*	在运行状况检查端口上允许流向实例的出站流量

对于将实例作为目标的内部负载均衡器，建议使用以下规则。

Inbound

Source	Port Range	Comment
VPC CIDR	*listener*	在负载均衡器侦听器端口上允许来自 VPC CIDR 的入站流量

Outbound

Destination	Port Range	Comment
instance security group	*instance listener*	在实例侦听器端口上允许流向实例的出站流量
instance security group	*health check*	在运行状况检查端口上允许流向实例的出站流量

如果将实例作为目标的应用程序负载均衡器本身是某个网络负载均衡器的目标，建议使用以下规则。

Inbound

Source	Port Range	Comment
client IP addresses/CIDR	*alb listener*	在负载均衡器侦听器端口上允许入站客户端流量
VPC CIDR	*alb listener*	允许入站客户端流量通过 Amazon PrivateLink 负载均衡器侦听器端口
VPC CIDR	*alb listener*	允许来自 Network Load Balancer 的入栈运行状况流量

Outbound

Destination	Port Range	Comment
instance security group	*instance listener*	在实例侦听器端口上允许流向实例的出站流量
instance security group	*health check*	在运行状况检查端口上允许流向实例的出站流量

更新关联的安全组

您可以随时更新与负载均衡器关联的安全组。

Console

更新安全组

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在安全性选项卡上，选择编辑。
5. 要将一个安全组与负载均衡器关联，请选择此安全组。要删除安全组关联，请选择安全组的 X 图标。
6. 选择保存更改。

Amazon CLI

更新安全组

使用 set-security-groups 命令。

```
aws elbv2 set-security-groups \
    --load-balancer-arn load-balancer-arn \
    --security-groups sg-01dd3383691d02f42 sg-00f4e409629f1a42d
```

CloudFormation

更新安全组

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
```

```
        - !Ref mySecurityGroup
        - !Ref myNewSecurityGroup
```

更新应用程序负载均衡器的 IP 地址类型

您可以配置您的 Application Load Balancer，以便客户端可以仅使用 IPv4 地址或同时使用 IPv4 和 IPv6 地址 (dualstack) 与负载均衡器进行通信。负载均衡器根据目标组的 IP 地址类型与目标进行通信。有关更多信息，请参阅 IP 地址类型。

dualstack 要求

- 您可以在创建负载均衡器时设置 IP 地址类型并随时更新它。
- 您为负载均衡器指定的 Virtual Private Cloud (VPC) 和子网必须具有关联的 IPv6 CIDR 块。有关更多信息，请参阅 Amazon EC2 用户指南中的 IPv6 地址。
- 负载均衡器子网的路由表必须路由 IPv6 流量。
- 负载均衡器的安全组必须允许 IPv6 流量。
- 负载均衡器子网的网络 ACL 必须允许 IPv6 流量。

Console

更新 IP 地址类型

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在网络映射选项卡上，选择编辑 IP 地址类型。
5. 对于 IP 地址类型，选择 IPv4 以仅支持 IPv4 地址，选择双栈以同时支持 IPv4 和 IPv6 地址，或选择不带公有 IPv4 的双栈以仅支持 IPv6 地址。
6. 选择保存更改。

Amazon CLI

要更新 IP 地址类型

使用 set-ip-address-type 命令。

```
aws elbv2 set-ip-address-type \
    --load-balancer-arn load-balancer-arn \
    --ip-address-type dualstack
```

CloudFormation

要更新 IP 地址类型

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

更新应用程序负载均衡器的 IPAM IP 地址池

IPAM IP 地址池必须首先在 IPAM 中创建，然后才能供您的应用程序负载均衡器使用。有关更多信息，请参阅将 IP 地址带入 IPAM。

Console

更新 IPAM IP 地址池

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在网络映射选项卡上，选择编辑 IP 池。
5. 在 IP 池下，选择使用公有 IPv4 地址的 IPAM 池，然后选择一个 IPAM 池。
6. 选择保存更改。

Amazon CLI

更新 IPAM IP 地址池

使用 modify-ip-pools 命令。

```
aws elbv2 modify-ip-pools \
    --load-balancer-arn load-balancer-arn \
    --ipam-pools Ipv4IpamPoolId=ipam-pool-1234567890abcdef0
```

CloudFormation

更新 IPAM IP 地址池

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internet-facing
      IpAddressType: ipv4
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      Ipv4IpamPoolId: !Ref myIPAMPool
```

编辑应用程序负载均衡器的属性

创建应用程序负载均衡器后，您可以编辑其属性。

负载均衡器属性

- 连接空闲超时
- HTTP 客户端保持连接持续时间
- 删除保护
- 异步缓解模式

- 主机标头保留

连接空闲超时

连接空闲超时是指在负载均衡器关闭连接之前，现有客户端或目标连接可以保持不活动状态（不发送或接收任何数据）的时间段。

为确保文件上传等耗时较长的操作有时间完成，请在每个空闲超时期限过去之前发送至少 1 字节的数据，并根据需要增加空闲超时期限的长度。此外，我们建议您将应用程序的空闲超时配置为大于负载均衡器的空闲超时的值。否则，如果应用程序不正常地关闭了与负载均衡器的 TCP 连接，则负载均衡器可能会在收到数据包之前向应用程序发送请求，表明连接已关闭。如果是这种情况，则负载均衡器将向客户端发送 HTTP 502 Bad Gateway（HTTP 502 无效网关）错误。

应用程序负载均衡器不支持 HTTP/2 PING 帧。此类框架不会重置连接空闲超时。

默认情况下，Elastic Load Balancing 将负载均衡器的空闲超时值设置为 60 秒。

Console

更新空闲超时值

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在属性选项卡上，选择编辑。
5. 在流量配置下，输入连接空闲超时的值。有效范围是 1 至 4000 秒。
6. 选择保存更改。

Amazon CLI

更新空闲超时值

使用带 `idle_timeout.timeout_seconds` 属性的 modify-load-balancer-attributes 命令。有效范围为 1 至 4000 秒。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=idle_timeout.timeout_seconds,Value=120"
```

CloudFormation

更新空闲超时值

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该idle_timeout.timeout_seconds属性。有效范围为 1 至 4000 秒。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "idle_timeout.timeout_seconds"
          Value: "120"
```

HTTP 客户端保持连接持续时间

HTTP 客户端保持连接持续时间是应用程序负载均衡器与客户端保持持久 HTTP 连接的最长时间长度。在配置的 HTTP 客户端保持连接持续时间过去后，应用程序负载均衡器将接受另一个请求，然后返回一个正常关闭连接的响应。

负载均衡器发送的响应类型取决于客户端连接使用的 HTTP 版本。

- 对于使用 HTTP 1.x 连接的客户端，负载均衡器会发送包含字段 `Connection: close` 的 HTTP 标头。
- 对于使用连接的客户端 HTTP/2，负载均衡器会发送一个GOAWAY帧。

默认情况下，应用程序负载均衡器将负载均衡器的 HTTP 客户端保持连接持续时间值设置为 3600 秒（即 1 小时）。HTTP 客户端保持连接持续时间不能关闭或设置为低于最小值 60 秒，但您可以增加 HTTP 客户端保持连接持续时间，最长可达 604800 秒（即 7 天）。当最初建立与客户端的 HTTP 连接时，应用程序负载均衡器开始 HTTP 客户端保持连接持续时间。持续时间在没有流量时会继续，并且在建立新连接之前不会重置。

当使用可用区转移或可用区自动转移将负载均衡器流量从受损的可用区转移出来时，具有现有开放连接的客户端可能会继续向受损位置发出请求，直到客户端重新连接。为了支持更快的恢复，请考虑设置较低的保持连接持续时间值，以限制客户端保持连接到负载均衡器的时间。有关更多信息，请参阅《Amazon 应用程序恢复控制器（ARC）开发人员指南》中的 Limit the time that clients stay connected to your endpoints。

> **Note**
>
> 当负载均衡器将应用程序负载均衡器的 IP 地址类型切换为 `dualstack-without-public-ipv4` 时，负载均衡器会等待所有活动连接完成。要缩短切换应用程序负载均衡器的 IP 地址类型所需的时间，请考虑降低 HTTP 客户端保持连接持续时间。

应用程序负载均衡器在初始连接期间分配 HTTP 客户端保持连接持续时间值。当您更新 HTTP 客户端保持连接持续时间时，这可能会导致同时建立具有不同 HTTP 客户端保持连接持续时间值的连接。现有连接将保留其初始连接期间应用的 HTTP 客户端保持连接持续时间值。新连接将接收更新后的 HTTP 客户端保持连接持续时间值。

Console

更新客户端保持连接持续时间

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在属性选项卡上，选择编辑。
5. 在流量配置下，输入 HTTP 客户端保持连接持续时间的值。有效范围为 60 至 604800 秒。
6. 选择保存更改。

Amazon CLI

更新客户端保持连接持续时间

使用带 `client_keep_alive.seconds` 属性的 modify-load-balancer-attributes 命令。有效范围为 60 至 604800 秒。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
```

```
    --attributes "Key=client_keep_alive.seconds,Value=7200"
```

CloudFormation

更新客户端保持连接持续时间

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该client_keep_alive.seconds属性。有效范围为 60 至 604800 秒。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "client_keep_alive.seconds"
          Value: "7200"
```

删除保护

为了防止您的负载均衡器被意外删除，您可以启用删除保护。默认情况下，已为负载均衡器禁用删除保护。

如果您为负载均衡器启用删除保护，则必须先禁用删除保护，然后才能删除负载均衡器。

Console

启用或禁用删除保护

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在属性选项卡上，选择编辑。

5. 在保护部分，启用或禁用删除保护。
6. 选择保存更改。

Amazon CLI

要启用或禁用删除保护

使用带 `deletion_protection.enabled` 属性的 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=deletion_protection.enabled,Value=true"
```

CloudFormation

要启用或禁用删除保护

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该`deletion_protection.enabled`属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "deletion_protection.enabled"
          Value: "true"
```

异步缓解模式

异步缓解模式可以保护您的应用程序不受由于 HTTP 异步造成的问题的影响。负载均衡器根据每个请求的威胁级别对请求进行分类，允许安全请求，然后根据您指定的缓解模式来减轻风险。异步缓解模式

包括“监控”、“防御”和“最严格”。默认情况下采用“防御”模式，该模式可在保持应用程序可用性的同时，针对 HTTP 异步提供持久的缓解作用。您可以切换到最严格模式，确保应用程序只接收符合 RFC 7230 标准的请求。

http_desync_guardian 库会分析 HTTP 请求，防止发生 HTTP 异步攻击。有关更多信息，请参阅上的 HTTP Desync Guardian。 GitHub

分类

下面列出了这些分类。

- 合规 – 请求符合 RFC 7230 标准，不构成已知的安全威胁。
- 可接受 - 请求不符合 RFC 7230 标准，但不构成已知的安全威胁。
- 不明确 - 请求不符合 RFC 7230 标准，会带来风险，因为各个 Web 服务器和代理可能会以不同的方式处理该请求。
- 严重 - 请求会带来很高的安全风险。负载均衡器会阻止请求，向客户端提供 400 响应，并关闭客户端连接。

如果请求不符合 RFC 7230 标准，负载均衡器将递增 `DesyncMitigationMode_NonCompliant_Request_Count` 指标。有关更多信息，请参阅 Application Load Balancer 指标。

每个请求的分类都包含在负载均衡器访问日志中。如果请求不符合，则访问日志将包含分类原因代码。有关更多信息，请参阅 分类原因。

模式

下表描述 Application Load Balancer 如何根据模式和分类来处理请求。

分类。	监控模式	防御模式	最严格模式
合规	已允许	已允许	已允许
可接受	已允许	已允许	阻止
不明确	已允许	已允许[1]	阻止
严重	已允许	阻止	阻止

[1] 系统将路由请求，但关闭客户端和目标连接。如果您的负载均衡器在防御模式下收到大量不明确请求，则可能会产生额外费用。这是因为每秒增加的新连接数会影响每小时使用的负载均衡器容量单位 (LCU)。您可以使用 NewConnectionCount 指标比较负载均衡器在监控模式和防御模式下建立新连接的方式。

Console

更新异步缓解模式

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在属性选项卡上，选择编辑。
5. 在流量配置部分的数据包处理下，对于异步缓解模式，选择防御、最严格或监控。
6. 选择保存更改。

Amazon CLI

更新异步缓解模式

使用带 routing.http.desync_mitigation_mode 属性的 modify-load-balancer-attributes 命令。可能的值为 monitor、defensive 或 strictest。默认值为 defensive。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=routing.http.desync_mitigation_mode,Value=monitor"
```

CloudFormation

更新异步缓解模式

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该routing.http.desync_mitigation_mode属性。可能的值为 monitor、defensive 或 strictest。默认值为 defensive。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
```

```
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.desync_mitigation_mode"
          Value: "monitor"
```

主机标头保留

启用 Preserve host header（保留主机标头）属性时，应用程序负载均衡器会保留 HTTP 请求中的 `Host` 标头，并将请求发送到目标而不做任何修改。如果应用程序负载均衡器收到多个 `Host` 标头，它会保留所有这些标头。侦听器规则仅会应用于收到的第一个 `Host` 标头。

默认情况下，未启用 Preserve host header（保留主机标头）属性时，应用程序负载均衡器会通过以下方式修改 `Host` 标头：

未启用主机标头保留，且侦听器端口为非默认端口时：不使用默认端口（端口 80 或 443）时，如果客户端尚未附加端口号，我们会将端口号附加到主机标头中。例如，假设侦听器端口是非默认端口（例如 `8080`），HTTP 请求中具有 `Host: www.example.com` 的 `Host` 标头将被修改为 `Host: www.example.com:8080`。

未启用主机标头保留，并且侦听器端口为默认端口（端口 80 或 443）时：对于默认侦听器端口（端口 80 或 443），我们不会将端口号附加到传出的主机标头中。传入主机标头中已经存在的任何端口号都将被移除。

应用程序负载均衡器将如何根据侦听器端口来处理 HTTP 请求中的主机标头的更多示例见下表。

侦听器端口	示例请求	请求中的主机标头	已禁用主机标头保留（默认行为）	已启用主机标头保留
请求在默认 HTTP/HTTPS 监听器上发送。	`GET /index.html HTTP/1.1`	example.com	example.com	example.com

侦听器端口	示例请求	请求中的主机标头	已禁用主机标头保留（默认行为）	已启用主机标头保留
	`Host: example.com`			
在默认的 HTTP 侦听器上发送请求，并且主机标头具有端口（例如，80 或 443）。	`GET / index.html HTTP/1.1 Host: example.com:80`	example.com:80	example.com	example.com:80
请求具有绝对路径。	`GET https://dns_name/index.html HTTP/1.1 Host: example.com`	example.com	dns_name	example.com
在非默认侦听器端口（例如 8080）上发送请求	`GET / index.html HTTP/1.1 Host: example.com`	example.com	example.com:8080	example.com
在非默认侦听器上发送请求，并且主机标头具有端口（例如 8080）。	`GET / index.html HTTP/1.1 Host: example.com:8080`	example.com:8080	example.com:8080	example.com:8080

Console

启用主机标头保留

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在属性选项卡上，选择编辑。
5. 在数据包处理下，打开保留主机标头。
6. 选择保存更改。

Amazon CLI

启用主机标头保留

使用 modify-load-balancer-attributes 命令，并将 `routing.http.preserve_host_header.enabled` 属性设置为 `true`。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=routing.http.preserve_host_header.enabled,Value=true"
```

CloudFormation

启用主机标头保留

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该`routing.http.preserve_host_header.enabled`属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
```

```
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.preserve_host_header.enabled"
          Value: "true"
```

为应用程序负载均衡器添加标签

使用标签可帮助您按各种标准对负载均衡器进行分类，例如按用途、所有者或环境。

您最多可以为每个负载均衡器添加多个标签。如果您添加的标签中的键已经与负载均衡器关联，它将更新该标签的值。

当您用完标签时，可以从负载均衡器中将其删除。

限制

- 每个资源的标签数上限 – 50
- 最大密钥长度 - 127 个 Unicode 字符
- 最大值长度 - 255 个 Unicode 字符
- 标签键和值区分大小写。允许的字符包括可表示的字母、空格和数字 UTF-8，以及以下特殊字符：+-=。_ : / @。请不要使用前导空格或尾随空格。
- 请勿在标签名称或值中使用aws:前缀，因为它已保留供 Amazon 使用。您无法编辑或删除带此前缀的标签名称或值。具有此前缀的标签不计入每个资源的标签数限制。

Console

要更新负载均衡器的标签

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在标签选项卡上，选择管理标签。
5. 要添加标签，请选择添加标签，然后输入标签键和值。
6. 要更新标签，请在键或值中输入新值。
7. 要删除标签，请选择标签旁边的删除。
8. 选择保存更改。

Amazon CLI

添加 标签

使用 add-tags 命令。

```
aws elbv2 add-tags \
    --resource-arns load-balancer-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

删除标签

使用 remove-tags 命令。

```
aws elbv2 remove-tags \
    --resource-arns load-balancer-arn \
    --tag-keys project department
```

CloudFormation

添加 标签

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该Tags属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

删除 Application Load Balancer

在您的负载均衡器可用之后，您需要为保持其运行的每小时或部分小时支付费用。当您不再需要该负载均衡器时，可将其删除。当负载均衡器被删除之后，您便不再需要支付负载均衡器费用。

如果已启用删除保护，则无法删除负载均衡器。有关更多信息，请参阅 删除保护。

请注意，删除负载均衡器不会影响其注册目标。例如，您的 EC2 实例将继续运行并仍注册到其目标组。要删除目标组，请参阅删除应用程序负载均衡器目标组。

DNS 记录

如果您有一个指向负载均衡器的域的一个 DNS 记录，请将它指向新的位置并等待 DNS 更改生效，然后再删除您的负载均衡器。

- 如果此记录是存活时间（TTL）为 300 秒的 CNAME 记录，请至少等待 300 秒，然后再继续执行下一步。
- 如果此记录是 Route 53 别名（A）记录，请至少等待 60 秒。
- 如果使用 Route 53，则记录更改需要 60 秒才能传播到所有全局 Route 53 名称服务器。将此时间添加到正在更新的记录的 TTL 值。

Console

删除负载均衡器

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器，然后依次选择操作、删除负载均衡器。
4. 提示进行确认时，输入 **confirm**，然后选择删除。

Amazon CLI

删除负载均衡器

使用 delete-load-balancer 命令。

```
aws elbv2 delete-load-balancer \
    --load-balancer-arn load-balancer-arn
```

查看应用程序负载均衡器资源地图

应用程序负载均衡器资源地图以交互方式显示您的负载均衡器的架构，包括其关联的侦听器、规则、目标组和目标。资源地图还突出显示所有资源之间的关系和路由路径，生成负载均衡器配置的直观表示。

查看应用程序负载均衡器的资源地图

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 选择资源地图选项卡以显示负载均衡器的资源地图。

资源地图组件

地图视图

应用程序负载均衡器资源地图中有两个视图可用：概述和不正常目标地图。默认情况下，概述处于选中状态，并显示您的负载均衡器的所有资源。选择不正常目标地图视图将仅显示不正常目标以及与之关联的资源。

不正常目标地图视图可用于对未通过运行状况检查的目标进行故障排除。有关更多信息，请参阅 使用资源地图排查不正常目标的问题。

资源组

应用程序负载均衡器资源地图包含四个资源组，每个资源类型一个。资源组为侦听器、规则、目标组和目标。

资源磁贴

组中的每个资源都有自己的磁铁，显示有关该特定资源的详细信息。

- 将鼠标悬停在资源磁贴上会突出显示该资源与其他资源之间的关系。
- 选择资源磁贴会突出显示该资源与其他资源之间的关系，并显示有关该资源的其他详细信息。
  - 规则条件：每条规则的条件。
  - 目标组运行状况摘要：每种运行状况状态的注册目标数量。
  - 目标运行状况状态：目标的当前运行状况状态和描述。

> **Note**
>
> 您可以关闭显示资源详细信息以隐藏资源地图中的其他详细信息。

- 每个资源磁贴都包含一个链接，选中后，该链接将导航到该资源的详细信息页面。
  - 侦听器 - 选择侦听器 protocol:port。例如，`HTTP:80`
  - 规则 - 选择规则操作。例如，`Forward to target group`
  - 目标组 - 选择目标组名称。例如，`my-target-group`
  - 目标 - 选择目标 ID。例如，`i-1234567890abcdef0`

导出资源地图

选择导出后，您可以选择将应用程序负载均衡器资源地图的当前视图导出为 PDF。

应用程序负载均衡器的可用区转移

可用区转移和可用区自动转移都是 Amazon 应用程序恢复控制器（ARC）的功能。使用可用区转移时，只需一个操作即可将流量从受损的可用区转移出去。这样，您就可以继续从 Amazon Web Services 区域中的其他运行状况良好的可用区运行。

使用 zonal autoshift，您可以授权 Amazon 在活动期间代表您从可用区域转移应用程序的资源流量，以帮助缩短恢复时间。 Amazon 当内部监控显示存在可能影响客户的可用区域受损时，将启动自动换档。 Amazon 启动自动切换时，您为区域自动切换配置的资源的应用程序流量开始从可用区转移出去。

当您启动可用区转移时，负载均衡器会停止向受影响的可用区发送资源的新流量。ARC 会立即创建可用区转移。但是，可用区中正在进行的现有连接也可能需要短暂的时间才能结束，具体取决于客户端行为和连接重用情况。根据您的 DNS 设置和其他因素，现有连接可能只需几分钟即可完成，也可能需要更长时间。有关更多信息，请参阅《Amazon 应用程序恢复控制器（ARC）开发人员指南》中的 Limit the time that clients stay connected to your endpoints。

内容

- 为应用程序负载均衡器启用可用区转移
- 为应用程序负载均衡器启动可用区转移
- 为应用程序负载均衡器更新可用区转移
- 为应用程序负载均衡器取消可用区转移

在开始可用区转移之前

- 可用区转移默认处于禁用状态，并且必须在每个应用程序负载均衡器上启用。有关更多信息，请参阅为应用程序负载均衡器启用可用区转移。
- 只能为单个可用区中的特定负载均衡器启动可用区转移。无法为多个可用区启动可用区转移。
- Amazon 当多个基础设施问题影响服务时，主动从 DNS 中删除区域负载均衡器 IP 地址。在开始可用区转移之前，请务必检查当前的可用区容量。如果您的负载均衡器已关闭跨可用区负载均衡，而您使用可用区转移来删除可用区负载均衡器 IP 地址，则受可用区转移影响的可用区也会失去目标容量。

有关更多信息，请参阅《Amazon 应用程序恢复控制器（ARC）开发人员指南》中的 Best practices for zonal shifts in ARC。

Cross-zone 负载平衡

在启用跨可用区负载均衡的应用程序负载均衡器上启动可用区转移时，受影响可用区中指向目标的所有流量都将被阻止，并且可用区 IP 地址也将从 DNS 中移除。

优点：

- 可在发生可用区故障时更快恢复。
- 可在某个可用区中检测到故障时将流量转移到正常的可用区。
- 可通过模拟和识别故障来测试应用程序完整性，预防计划外停机时间。

可用区转移管理覆盖

属于应用程序负载均衡器的目标包括一个独立于 `TargetHealth` 状态的新状态 `AdministrativeOverride`。

为应用程序负载均衡器启动可用区转移后，转出的可用区内的所有目标都将视为被管理覆盖。应用程序负载均衡器将停止向被管理覆盖的目标路由新流量。现有连接将保持不变，直至其自然终止。

可能的 `AdministrativeOverride` 状态包括：

unknown

由于内部错误，无法传播状态

no_override

目标上当前没有活动的覆盖

zonal_shift_active

可用区转移在目标可用区处于活动状态

为应用程序负载均衡器启用可用区转移

可用区转移默认处于禁用状态，并且必须在每个应用程序负载均衡器上启用。这确保了您仅能使用所需的特定应用程序负载均衡器来启动可用区转移。有关更多信息，请参阅 the section called “可用区转移”。

Console

启用可用区转移

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择该应用程序负载均衡器。
4. 在属性选项卡上，选择编辑。
5. 在可用区路由配置部分，对于 ARC 可用区转移集成，请选择 启用。
6. 选择保存更改。

Amazon CLI

要启用可用区转移

使用带 `zonal_shift.config.enabled` 属性的 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=zonal_shift.config.enabled,Value=true"
```

CloudFormation

要启用可用区转移

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该zonal_shift.config.enabled属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        -Key: "zonal_shift.config.enabled"
         Value: "true"
```

为应用程序负载均衡器启动可用区转移

ARC 中的区域切换允许您暂时将受支持资源的流量从可用区移开，这样您的应用程序就可以继续在某个 Amazon 区域中的其他可用区正常运行。

先决条件

在开始之前，请确认您已为负载均衡器负载均衡器启用可用区转移。

Console

本程序说明了如何使用 Amazon EC2 控制台来启动可用区转移。有关使用 ARC 控制台启动可用区转移的步骤，请参阅《Amazon 应用程序恢复控制器（ARC）开发人员指南》中的 Starting a zonal shift。

启动可用区转移

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择该应用程序负载均衡器。
4. 在集成选项卡中展开 Amazon 应用程序恢复控制器（ARC），然后选择启动可用区转移。
5. 选择要将流量移离的可用区。
6. 选择或输入可用区转移的到期时间。可用区转移最初可以从 1 分钟设置为三天（72 小时）。

   所有可用区转移都是暂时的。您必须设置过期时间，但可以稍后更新活跃转移以设置新的过期时间。
7. 输入注释。您可以稍后更新可用区转移以编辑注释。
8. 选中该复选框以确认启动可用区转移，这会将流量移离该可用区，从而减少应用程序的容量。
9. 选择确认。

Amazon CLI

启动可用区转移

使用 Amazon 应用程序恢复控制器（ARC）的 start-zonal-shift 命令。

```
aws arc-zonal-shift start-zonal-shift \
    --resource-identifier load-balancer-arn \
    --away-from use2-az2 \
    --expires-in 2h \
    --comment "zonal shift due to scheduled maintenance"
```

为应用程序负载均衡器更新可用区转移

您可以更新可用区转移，以设置新的到期时间，也可以编辑或替换可用区转移的注释。

Console

本程序说明了如何使用 Amazon EC2 控制台来更新可用区转移。有关使用 Amazon 应用程序恢复控制器（ARC）控制台更新可用区转移的步骤，请参阅《Amazon 应用程序恢复控制器（ARC）开发人员指南》中的 Updating a zonal shift。

更新可用区转移

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择具有活跃可用区转移的应用程序负载均衡器。
4. 在集成选项卡中，展开 Amazon Application Recovery Controller（ARC），然后选择更新可用区转移。

   此操作将打开 ARC 控制台以继续更新流程。
5. （可选）对于设置可用区转移到期时间，可以选择或输入到期时间。
6. （可选）对于注释，可以选择编辑现有注释或输入新注释。
7. 选择更新。

Amazon CLI

更新可用区转移

使用 Amazon 应用程序恢复控制器（ARC）的 update-zonal-shift 命令。

```
aws arc-zonal-shift update-zonal-shift \
    --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE \
    --expires-in 1h \
    --comment "extending zonal shift for scheduled maintenance"
```

为应用程序负载均衡器取消可用区转移

在可用区转移到期之前，您可以随时取消它。你可以取消你启动的区域移动，也可以取消为区域自动移位练习跑而 Amazon 开始的区域移动。

Console

本程序说明了如何使用 Amazon EC2 控制台来取消可用区转移。有关使用 Amazon 应用程序恢复控制器（ARC）控制台取消可用区转移的步骤，请参阅《Amazon 应用程序恢复控制器（ARC）开发人员指南》中的 Canceling a zonal shift。

取消可用区转移

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择具有活跃可用区转移的应用程序负载均衡器。

4. 在集成选项卡中的 Amazon 应用程序恢复控制器（ARC）下，选择取消可用区转移。

   此操作将打开 ARC 控制台以继续取消流程。
5. 选择 Cancel zonal shift（取消可用区转移）。
6. 当系统提示进行确认时，选择 Confirm。

Amazon CLI

取消可用区转移

使用 Amazon 应用程序恢复控制器（ARC）的 cancel-zonal-shift 命令。

```
aws arc-zonal-shift cancel-zonal-shift \
    --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE
```

应用程序负载均衡器的容量预留

借助负载均衡器容量单位（LCU）预留，您可以为负载均衡器预留一个静态的最小容量。应用程序负载均衡器会自动扩展，以支持检测到的工作负载并满足容量需求。配置最小容量后，负载均衡器会根据接收到的流量继续纵向扩展或缩减容量，但也会防止容量低于配置的最小容量。

在以下情况下，可以考虑使用 LCU 预留：

- 您即将举办的活动将出现突发性异常高流量，需要确您的保负载均衡器能够在活动期间支撑突发的流量峰值。
- 您的工作负载特性导致短期内存在不可预测的流量峰值。
- 您正在配置负载均衡器，以便在特定启动时间接入或迁移服务，且需要从高容量开始运行，而非等待自动扩缩生效。
- 您正在负载均衡器之间迁移工作负载，并希望将目标配置调整为与源负载均衡器的规模相匹配。

估算您需要的容量

在确定为负载均衡器预留的容量时，我们建议您进行负载测试或分析代表预期未来流量的历史工作负载数据。使用弹性负载均衡控制台，您可以根据审核的流量估算需要预留的容量大小票。

或者，您可以使用该 CloudWatch 指标PeakLCUs来确定所需的容量级别。PeakLCUs 指标会反映流量模式中的峰值，负载均衡器必须跨所有扩展维度进行扩展以支持您的工作负载。PeakLCUs 指标与

ConsumedLCUs 指标不同，后者仅汇总流量的账单维度。建议使用 PeakLCUs 指标来确保在负载均衡器扩展期间的 LCU 预留充足。估算容量时，请使用 PeakLCUs 的每分钟 Sum。

如果您没有历史工作负载数据可供参考，也无法执行负载测试，则可以使用 LCU 预留计算器来估算所需容量。LCU 预留计算器使用基于 Amazon 观察到的历史工作负载的数据，可能无法代表您的特定工作负载。有关更多信息，请参阅 Load Balancer Capacity Unit Reservation Calculator。

LCU 预留的最小值和最大值

总预留请求必须至少为 100 LCU。最大值取决于您账户的配额。有关更多信息，请参阅 the section called “负载均衡器容量单位”。

为应用程序负载均衡器请求负载均衡器容量单位预留

在使用 LCU 预留之前，首先应检查以下信息：

- 容量是在区域层面预留的，并将在可用区之间平均分配。在启用 LCU 预留之前，请确认每个可用区中都有足够的均匀分布目标。
- LCU 预留请求遵循“先到先得”原则，具体取决于该可用区当时的可用容量。多数请求通常在几分钟内完成，但也有可能需要数小时。
- 更新现有预留时，须待先前的请求完成预调配或失败。您可以按需多次增加预留容量，但每日仅限两次“减少”操作。
- 所有预留或已配置容量在终止或取消前将持续计费。

Console

要请求 LCU 预留

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器名称。
4. 在容量选项卡中，选择编辑 LCU 预留。
5. 选择基于历史参考的估算。
6. 选择参考时段，以查看推荐的预留 LCU 级别。
7. 如果您没有历史参考工作负载，则可以选择手动估算，然后输入要预留的 LCU 数量。
8. 选择保存。

Amazon CLI

要请求 LCU 预留

使用 modify-capacity-reservation 命令。

```
aws elbv2 modify-capacity-reservation \
    --load-balancer-arn load-balancer-arn \
    --minimum-load-balancer-capacity CapacityUnits=100
```

CloudFormation

要请求 LCU 预留

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      MinimumLoadBalancerCapacity:
        CapacityUnits: 100
```

为应用程序负载均衡器更新或取消负载均衡器容量单位预留

如果负载均衡器的流量模式发生变化，您可以更新或取消负载均衡器的 LCU 预留。LCU 预留的状态必须为已预调配。

Console

更新或取消 LCU 预留

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器名称。
4. 在容量选项卡中，执行以下操作之一：
   a. 要更新 LCU 预留，请选择编辑 LCU 预留。
   b. 要取消 LCU 预留，请选择取消容量。

Amazon CLI

要取消 LCU 预订

使用 modify-capacity-reservation 命令。

```
aws elbv2 modify-capacity-reservation \
    --load-balancer-arn load-balancer-arn \
    --reset-capacity-reservation
```

监控应用程序负载均衡器的负载均衡器容量单位预留

预留状态

以下是 LCU 预留的可能状态值：

- pending - 表示该预留正在配置中。
- provisioned - 表示预留容量已准备就绪，可供使用。
- failed - 表示当前无法完成请求。
- rebalancing - 表示可用区已添加或删除，负载均衡器正在重新分配容量。

LCU 使用率

ReservedLCUs 指标每分钟报告一次。容量是以小时为单位预留的。例如，假设您的 LCU 预留量为 6,000，则 ReservedLCUs 的一小时总预留量为 6,000，一分钟的总预留量为 100。要确定您的预留 LCU 利用率，请参阅 PeakLCUs 指标。您可以设置 CloudWatch 警报，将每分钟Sum与您的预留容量值或每小时Sum的容量值进行比较ReservedLCUs，以确定您是否有足够的预留容量来满足您的需求。PeakLCUs

Console

查看 LCU 预留的状态

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器名称。
4. 在容量选项卡中，您可以查看预留状态和预留 LCU 值。

Amazon CLI

要监控 LCU 预留的状态

使用 describe-capacity-reservation 命令。

```
aws elbv2 describe-capacity-reservation \
    --load-balancer-arn load-balancer-arn
```

应用程序负载均衡器的集成

您可以通过与其他几项 Amazon 服务集成来优化 Application Load Balancer 架构，从而增强应用程序的性能、安全性和可用性。

负载均衡器集成

- Amazon 应用程序恢复控制器（ARC）
- 亚马逊 CloudFront + Amazon WAF
- Amazon Global Accelerator
- Amazon Config
- Amazon WAF

Amazon 应用程序恢复控制器（ARC）

Amazon 应用程序恢复控制器（ARC）可帮助您将负载均衡器的流量从受损的可用区转移到同一区域内的正常可用区。使用可用区转移可以降低可用区中的停电、硬件问题或软件问题对应用程序所造成影响的持续时间和严重性。

有关更多信息，请参阅 应用程序负载均衡器的可用区转移。

亚马逊 CloudFront + Amazon WAF

Amazon CloudFront 是一项网络服务，可帮助您提高所使用的应用程序的性能、可用性和安全性 Amazon。 CloudFront 充当使用应用程序负载均衡器的 Web 应用程序的分布式单一入口点。它可以实现应用程序负载均衡器的全球覆盖，使其能够从附近的边缘站点高效地为用户提供服务，为全球用户优化内容交付并减少延迟。通过在这些边缘站点自动缓存内容，可显著减少应用程序负载均衡器的负载，从而提高其性能和可扩展性。

Elastic Load Balancing 控制台中提供的一键式集成可创建具有推荐 Amazon WAF 安全保护的 CloudFront 分配，并将其关联到您的应用程序负载均衡器。这些 Amazon WAF 保护措施可在到达您的负载均衡器之前阻止常见的 Web 漏洞。您可以从控制台中负载均衡器的 “集成” 选项卡访问该 CloudFront 分配及其相应的安全控制面板。有关更多信息，请参阅《亚马逊 CloudFront 开发者指南》中的 “管理 Amazon WAF CloudFront 安全控制面板” 和 aws.amazon 上的 “ CloudFront 安全控制面板、统一 CDN 和安全体验简介”。 com/blogs。

作为安全最佳实践，请将面向 Internet 的应用程序负载均衡器的安全组配置为仅允许来自 Amazon 托管前缀列表的入站流量 CloudFront，并删除任何其他入站规则。有关更多信息，请参阅《亚马逊 CloudFront 开发者指南》中的使用 CloudFront 托管前缀列表、配置为 CloudFront 向请求添加自定义 HTTP 标头和配置 Application Load Balancer 以仅转发包含特定标头的请求 >。

> **Note**
>
> CloudFront 仅支持美国东部（弗吉尼亚北部）us-east-1 区域的 ACM 证书。如果您的 Application Load Balancer 在 us-east-1 以外的区域中配置了 ACM 证书的 HTTPS 侦听器，则需要将源连接从 HTTPS 更改 CloudFront 为 HTTP，或者在美国东部（弗吉尼亚北部）区域配置 ACM 证书并将其附加到您的分配中。 CloudFront

Amazon Global Accelerator

为优化应用程序的可用性、性能和安全性，请为负载均衡器创建加速器。加速器将 Amazon 全球网络上的流量引导到静态 IP 地址，这些地址在离客户端最近的区域中充当固定端点。 Amazon Global Accelerator 受 Shield Standard 保护，该标准可最大限度地减少应用程序停机时间和 DDoS 攻击造成的延迟。

有关更多信息，请参阅《Amazon Global Accelerator 开发人员指南》中的创建负载均衡器时添加加速器。

Amazon Config

要优化负载均衡器的监控和合规性，请进行设置 Amazon Config。 Amazon Config 提供了您 Amazon 账户中 Amazon 资源配置的详细视图。这些信息包括资源之间的相互关系以及资源的历史配置，让您可以了解资源配置和关系的历史变化。 Amazon Config 可以简化审计、合规性和问题排查。

有关更多信息，请参见Amazon Config 开发人员指南。

Amazon WAF

您可以将 Application Load Balancer Amazon WAF 与 Application Load Balancer 配合使用，根据网络访问控制列表 (Web ACL) 中的规则允许或阻止请求。

默认情况下，如果负载均衡器无法从中获得响应 Amazon WAF，则会返回 HTTP 500 错误并且不会转发请求。如果您需要负载均衡器即使无法联系目标也能将请求转发到目标 Amazon WAF，则可以启用 Amazon WAF 失效打开。

Pre-defined 网页 ACL

启用 Amazon WAF 集成后，您可以选择使用预定义的规则自动创建新的 Web ACL。预定义的 Web ACL 包括三个 Amazon 托管规则，可针对最常见的安全威胁提供保护。

- `AWSManagedRulesAmazonIpReputationList` - Amazon IP 信誉列表规则组会阻止通常与机器人或其他威胁关联的 IP 地址。有关更多信息，请参阅《Amazon WAF 开发人员指南》中的 Amazon IP reputation list managed rule group。
- `AWSManagedRulesCommonRuleSet` - 核心规则集（CRS）规则组有助于防止利用各种漏洞，包括 OWASP 出版物（如 OWASP Top 10）中描述的一些高风险和经常发生的漏洞。有关更多信息，请参阅《Amazon WAF 开发人员指南》中的 Core rule set (CRS) managed rule group。
- `AWSManagedRulesKnownBadInputsRuleSet` - 已知错误输入规则组阻止请求模式，这些模式确认无效且与漏洞攻击或发现相关联。有关更多信息，请参阅《Amazon WAF 开发人员指南》中的 Known bad inputs managed rule group。

有关更多信息，请参阅《Amazon WAF 开发人员指南》中的 Using web ACLs in Amazon WAF。

> **Note**
>
> 查看 WAF HTTP/2 流量检查行为设置，该设置控制何时 Amazon WAF 检查您的 Application Load Balancer 的 HTTP/2 请求正文。检查时间会影响安全覆盖范围以及与不同应用程序通信

模式的兼容性。要配置此设置，请导航到目标组的 “编辑目标组属性” 页面，找到 WAF HTTP/2 流量检查行为配置。

Application Load Balancer 的侦听器

侦听器是一个使用您配置的协议和端口检查连接请求的进程。在开始使用 Application Load Balancer 之前，必须添加至少一个侦听器。如果您的负载均衡器没有侦听器，则无法接收来自客户端的流量。您为侦听器定义的规则确定负载均衡器如何将请求路由到您注册的目标。例如 EC2 实例。

内容

侦听器配置

侦听器支持以下协议和端口：

- 协议：HTTP、HTTPS
- 端口：1-65535

可以使用 HTTPS 侦听器将加密和解密的工作交给负载均衡器完成，以便应用程序可以专注于其业务逻辑。如果侦听器协议为 HTTPS，您必须在侦听器上部署至少一个 SSL 服务器证书。有关更多信息，请参阅 为您的 Application Load Balancer 创建 HTTPS 侦听器。

如果必须确保目标解密 HTTPS 流量而不是负载均衡器，则可以在端口 443 上使用 TCP 侦听器创建网络负载均衡器。通过 TCP 侦听器，负载均衡器将加密流量传递到目标，而不会对其进行解密。有关网络负载均衡器的更多信息，请参阅网络负载均衡器用户指南。

WebSockets

应用程序负载均衡器为提供原生支持。 WebSockets您可以使用 HTTP HTTP/1.1 连接升级将现有连接升级为 WebSocket （ws或wss）连接。升级时，用于请求的 TCP 连接（到负载均衡器和到目标）会通过负载均衡器成为客户端与目标之间的永久 WebSocket 连接。您可以同时使用 WebSockets HTTP 和 HTTPS 侦听器。您为监听器选择的选项适用于 WebSocket 连接以及 HTTP 流量。路由到已启用目标优化器的目标组的请求不支持 Websockets。有关更多信息，请参阅《Amazon CloudFront 开发者指南》中的 WebSocket 协议工作原理。

HTTP/2

应用程序负载均衡器为 HTTP/2 HTTPS 侦听器提供原生支持。使用一个 HTTP/2 连接，您最多可以并行发送 128 个请求。您可以使用协议版本将请求发送到目标 HTTP/2。有关更多信息，请参阅 协议版本。由于 HTTP/2 使用前端连接的效率更高，因此您可能会注意到客户端与负载均衡器之间的连接较少。您不能使用的服务器推送功能。 HTTP/2

应用程序负载均衡器的双向 TLS 身份验证支持 HTTP/2 直通和验证模式。有关更多信息，请参阅 在应用程序负载均衡器中使用 TLS 进行双向身份验证。

有关更多信息，请参阅 Elastic Load Balancing 用户指南中的请求路由。

侦听器属性

应用程序负载均衡器的侦听器属性如下：

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

允许您修改 X-Amzn-Mtls-Clientcert-Serial-NumberHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

允许您修改 X-Amzn-Mtls-Clientcert-IssuerHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

允许您修改 X-Amzn-Mtls-Clientcert-SubjectHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

允许您修改 X-Amzn-Mtls-Clientcert-ValidityHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

允许您修改 X-Amzn-Mtls-Clientcert-LeafHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_mtls_clientcert.header_name`

允许您修改 X-Amzn-Mtls-ClientcertHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_tls_version.header_name`

允许您修改 X-Amzn-Tls-VersionHTTP 请求标头的标头名称。

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

允许您修改 X-Amzn-Tls-Cipher-SuiteHTTP 请求标头的标头名称。

`routing.http.response.server.enabled`

使您能够允许或移除 HTTP 响应服务器标头。

`routing.http.response.strict_transport_security.header_value`

告知浏览器只能使用 HTTPS 访问该网站，并且将来任何使用 HTTP 访问该网站的尝试都应自动转换为 HTTPS。

`routing.http.response.access_control_allow_origin.header_value`

指定要允许访问服务器的源。

`routing.http.response.access_control_allow_methods.header_value`

返回从其他源访问服务器时允许使用的 HTTP 方法。

`routing.http.response.access_control_allow_headers.header_value`

指定请求期间可以使用的标头。

`routing.http.response.access_control_allow_credentials.header_value`

指示浏览器在发出请求时是否应包含凭证（例如 Cookie 或身份验证）。

`routing.http.response.access_control_expose_headers.header_value`

返回浏览器可以向发出请求的客户端公开的标头。

`routing.http.response.access_control_max_age.header_value`

指定预检请求的结果的缓存时长（以秒为单位）。

`routing.http.response.content_security_policy.header_value`

指定浏览器为了帮助减少某些类型安全威胁的风险而强制实施的限制。

`routing.http.response.x_content_type_options.header_value`

表示是否应遵循Content-Type标题中通告的 MIME 类型，而不应更改。

`routing.http.response.x_frame_options.header_value`

指示是否允许浏览器在 frame、iframe、embed 或 object 中呈现页面。

默认操作

每个侦听器都有默认操作，也称为默认规则。默认规则无法删除，并且始终会最后执行。您可以创建其他规则。这些规则由优先级、一个或多个操作以及一个或多个条件组成。您可以随时添加或编辑规则。有关更多信息，请参阅 侦听器规则。

为您的 Application Load Balancer 创建 HTTP 侦听器

侦听器检查连接请求。您可在创建负载均衡器时定义侦听器，并可随时向负载均衡器添加侦听器。

此页面上的信息可帮助您为负载均衡器创建 HTTP 侦听器。要向您的负载均衡器添加 HTTPS 侦听器，请参阅 为您的 Application Load Balancer 创建 HTTPS 侦听器。

先决条件

- 要将转发操作添加到默认侦听器规则，您必须指定可用的目标组。有关更多信息，请参阅 为您的应用程序负载均衡器创建目标组。
- 您可以在多个侦听器中指定同一个目标组，但这些侦听器必须属于同一个负载均衡器。要将目标组与负载均衡器结合使用，您必须确认其没有被任何其他负载均衡器的侦听器使用。

添加 HTTP 侦听器

您为侦听器配置用于从客户端连接到负载均衡器的协议和端口，并为默认侦听器规则配置目标组。有关更多信息，请参阅 侦听器配置。

要添加其他侦听器规则，请参阅侦听器规则。

Console

添加 HTTP 侦听器

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择添加侦听器。
5. 对于协议，请选择 HTTP。保留默认端口或输入其他端口。
6. 对于默认操作，请选择以下路由操作之一并提供所需的信息：

   - 转发到目标组：选择一个目标组。要添加其他目标组，请选择添加目标组，然后选择一个目标组，检查相对权重并根据需要更新权重。如果在任何目标组上启用了粘性，则必须启用组级粘性。

     如果没有能满足您需求的目标组，请选择创建目标组，以立即创建一个目标组。有关更多信息，请参阅 创建目标组。
   - 重定向到 URL：在 URI 部分选项卡上分别输入每个部分，或者在完整 URL选项卡上输入完整的地址，从而输入 URL。对于状态代码，根据您的需求选择临时（HTTP 302）或永久（HTTP 301）。
   - 返回固定响应：输入要为已删除的客户端请求返回的响应代码。您也可以指定内容类型和响应正文。
7. （可选）要添加标签，请展开侦听器标签。选择添加新标签，然后输入标签键和标签值。
8. 选择添加侦听器。

Amazon CLI

创建目标组

如果您没有可以用于默认操作的目标组，请立即使用 create-target-group 命令来创建一个目标组。有关示例，请参阅 创建目标组。

创建 HTTP 侦听器

使用 create-listener 命令。以下示例会创建一个 HTTP 侦听器，其默认规则会将流量转发到指定目标组。

```
aws elbv2 create-listener \
```

```
    --load-balancer-arn load-balancer-arn \
    --protocol HTTP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

要创建转发操作以在两个目标组之间分配流量，请改用以下 `--default-actions` 选项。指定多个目标组时，您必须提供每个目标组的权重。

```
    --default-actions '[{
          "Type":"forward",
          "ForwardConfig":{
            "TargetGroups":[
              {"TargetGroupArn":"target-group-1-arn","Weight":50},
              {"TargetGroupArn":"target-group-2-arn","Weight":50}
            ]
          }
        }]'
```

CloudFormation

创建 HTTP 侦听器

定义 AWS::ElasticLoadBalancingV2:: Listener 类型的资源。以下示例会创建一个 HTTP 侦听器，其默认规则会将流量转发到指定目标组。

```
Resources:
  myHTTPlistener:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    LoadBalancerArn: !Ref myLoadBalancer
    Protocol: HTTP
    Port: 80
    DefaultActions:
      - Type: "forward"
        TargetGroupArn: !Ref myTargetGroup
```

要创建转发操作以在两个目标组之间分配流量，请使用 `ForwardConfig` 属性。指定多个目标组时，您必须提供每个目标组的权重。

```
Resources:
  myHTTPlistener:
```

```
Type: 'AWS::ElasticLoadBalancingV2::Listener'
Properties:
  LoadBalancerArn: !Ref myLoadBalancer
  Protocol: HTTP
  Port: 80
  DefaultActions:
    - Type: "forward"
      ForwardConfig:
        TargetGroups:
          - TargetGroupArn: !Ref TargetGroup1
            Weight: 50
          - TargetGroupArn: !Ref TargetGroup2
            Weight: 50
```

应用程序负载均衡器的 SSL 证书

当您为应用程序负载均衡器创建安全侦听器时，必须在负载均衡器上部署至少一个证书。负载均衡器需要 X.509 证书（SSL/TLS 服务器证书）。证书是由证书颁发机构 (CA) 颁发的数字化身份。证书包含标识信息、有效期限、公有密钥、序列号以及发布者的数字签名。

在创建用于负载均衡器的证书时，您必须指定域名。证书上的域名必须与自定义域名记录匹配，以确保我们能够验证 TLS 连接。如果不匹配，则流量不会加密。

必须为证书指定完全限定域名 (FQDN)（例如 `www.example.com`）或顶点域名（例如 `example.com`）。您还可以使用星号 (*) 作为通配符来保护同一域中的多个站点名称。请求通配符证书时，星号 (*) 必须位于域名的最左侧位置，而且只能保护一个子域级别。例如，`*.example.com` 保护 `corp.example.com` 和 `images.example.com`，但无法保护 `test.login.example.com`。另请注意，`*.example.com` 仅保护 `example.com` 的子域，而不保护裸域或顶点域 (`example.com`)。通配符名称显示在证书的 Subject（主题）字段和 Subject Alternative Name（主题替代名称）扩展中。有关公有证书的更多信息，请参阅《Amazon Certificate Manager 用户指南》中的 Request a public certificate。

我们建议您使用 Amazon Certificate Manager (ACM) 为您的负载均衡器创建证书。ACM 支持具有 2048、3072 和 4096 位密钥长度的 RSA 证书，以及所有 ECDSA 证书。ACM 与 Elastic Load Balancing 集成，以便您可以在负载均衡器上部署证书。有关更多信息，请参阅 Amazon Certificate Manager 用户指南。

或者，您可以使用 SSL/TLS 工具创建证书签名请求 (CSR)，然后获取 CA 签署的 CSR 以生成证书，然后将证书导入 ACM 或将证书上传到 Amazon Identity and Access Management (IAM)。有关将证书

导入 ACM 的信息，请参阅 Amazon Certificate Manager 用户指南中的将证书导入。有关将证书上传到 IAM 的更多信息，请参阅 IAM 用户指南中的使用服务器证书。

默认证书

创建 HTTPS 侦听器时，必须仅指定一个证书。此证书称为默认证书。创建 HTTPS 侦听器后，您可以替换默认证书。有关更多信息，请参阅 替换默认证书。

如果在证书列表中指定其他证书，则仅当客户端在不使用服务器名称指示 (SNI) 协议的情况下连接以指定主机名或证书列表中没有匹配的证书时，才使用默认证书。

如果您未指定其他证书但需要通过单一负载平衡器托管多个安全应用程序，则可以使用通配符证书或为证书的每个其他域添加使用者备用名称 (SAN)。

证书列表

创建 HTTPS 侦听器后，您可以向证书列表添加证书。如果您使用创建了侦听器 Amazon Web Services 管理控制台，则我们会为您将默认证书添加到证书列表中。否则，证书列表为空。使用证书列表可使负载均衡器在同一端口上支持多个域，并为每个域提供不同的证书。有关更多信息，请参阅将证书添加到证书列表。

负载均衡器使用支持 SNI 的智能证书选择算法。如果客户端提供的主机名与证书列表中的一个证书匹配，则负载均衡器将选择此证书。如果客户端提供的主机名与证书列表中的多个证书匹配，则负载均衡器将选择客户端可支持的最佳证书。根据以下标准，按下面的顺序选择证书：

- 公有密钥算法 (ECDSA 优先于 RSA)
- 过期时间（最好未过期）
- 哈希算法（SHA 优先于 MD5）。如果有多个 SHA 证书，则优先使用 SHA 数最大的证书。
- 密钥长度 (首选最大值)
- 有效期

负载均衡器访问日志条目指示客户端指定的主机名和向客户端提供的证书。有关更多信息，请参阅 访问日志条目。

证书续订

每个证书都有有效期限。您必须确保在有效期结束之前续订或替换负载均衡器的每个证书。这包括默认证书和证书列表中的证书。续订或替换证书不影响负载均衡器节点已收到的进行中的请求，并暂停指向

正常运行的目标的路由。续订证书之后，新的请求将使用续订后的证书。更换证书之后，新的请求将使用新证书。

您可以按如下方式管理证书续订和替换：

- 由您的负载均衡器提供 Amazon Certificate Manager 并部署在您的负载均衡器上的证书可以自动续订。ACM 会尝试在到期之前续订证书。有关更多信息，请参阅 Amazon Certificate Manager 用户指南中的托管续订。
- 如果您将证书导入 ACM，则必须监视证书的到期日期并在到期前续订。有关更多信息，请参阅 Amazon Certificate Manager 用户指南中的导入证书。
- 如果您已将证书导入 IAM 中，则必须创建一个新证书，将该新证书导入 ACM 或 IAM 中，将该新证书添加到负载均衡器，并从负载均衡器删除过期的证书。

应用程序负载均衡器的安全策略

Elastic Load Balancing 使用一个安全套接字层 (SSL) 协商配置（称为安全策略）在客户端与负载均衡器之间协商 SSL 连接。安全策略是协议和密码的组合。协议在客户端与服务器之间建立安全连接，确保在客户端与负载均衡器之间传递的所有数据都是私密数据。密码是使用加密密钥创建编码消息的加密算法。协议使用多种密码对 Internet 上的数据进行加密。在 连接协商过程中，客户端和负载均衡器会按首选项顺序提供各自支持的密码和协议的列表。默认情况下，会为安全连接选择服务器列表中与任何一个客户端的密码匹配的第一个密码。

注意事项

- HTTPS 侦听器需要有安全策略。如果您在创建侦听器时未指定安全策略，我们将使用默认安全策略。默认安全策略取决于您创建 HTTPS 侦听器的方式：
  - 控制台：默认安全策略为 `ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09`。
  - 其他方法（例如 Amazon CLI Amazon CloudFormation、和 Amazon CDK）-默认安全策略是`ELBSecurityPolicy-2016-08`。
  - 要查看负载均衡器连接请求的 TLS 协议版本（日志字段位置 5）和密钥交换（日志字段位置 13），请启用连接日志并检查相应的日志条目。有关更多信息，请参阅连接日志。
  - 以 PQ 命名的安全策略提供混合后量子密钥交换。出于兼容性考虑，它们支持经典和后量子 ML-KEM 密钥交换算法。客户端必须支持 ML-KEM 密钥交换，才能使用混合后量子 TLS 进行密钥交换。混合后量子策略支持 secp256r1mlkem768、secp384r1mlkem1024 和 X25519MLKEM768 算法。有关更多信息，请参阅Post-quantum 密码学。

  - AWS 建议实施新的基于后量子 TLS (PQ-TLS) 的安全策略`ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09`或`ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09`。该策略通过支持能够协商混合 PQ-TLS、仅限 TLS 1.3 或仅限 TLS 1.2 的客户端，从而最大限度地减少向后量子加密过渡期间的服务中断，从而最大限度地减少向后量子密码学过渡期间的服务中断。随着您的客户端应用程序开发出协商 PQ-TLS 密钥交换操作的能力，您可以逐步迁移到更严格的安全策略。
  - 名称中带有 RFC 9151 的安全策略可帮助您遵守 RFC 9151，该协议定义了美国国家安全局 (NSA) 规定的商业国家安全算法 (CNSA) 1.0 套件的 TLS 要求。为了帮助过渡，它们分为两类：强制执行完整 RFC 9151 要求的严格策略，以及支持符合 RFC 9151 和非 RFC 9151 密码以帮助逐步过渡的互操作策略（名称中包含 “INTEROP”）。 Amazon 建议从最大限度`ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07`地减少中断开始，然后随着客户支持 RFC 9151，逐渐转向更严格的策略。您可以使用 ALB 连接日志中的`tls_protocol`、`tls_cipher`和`tls_keyexchange`字段来监控客户端连接。有关 RFC 9151 的更多信息，请参阅 IETF 网站上的 RFC 9151。
- 为了满足需要禁用某些 TLS 协议版本的合规性和安全性标准，或者为了支持需要已弃用密码的旧客户端，您可以使用其中一种 `ELBSecurityPolicy-TLS-` 安全策略。要查看针对应用程序负载均衡器的请求的 TLS 协议版本，请为负载均衡器启用访问日志记录并检查相应的访问日志条目。有关更多信息，请参阅访问日志。
- 您可以分别使用您 Amazon Web Services 账户 的 IAM 中的 Elastic Load Balancing 条件密钥和服务控制策略 (SCP) 来限制用户可以使用哪些安全策略。 Amazon Organizations 有关更多信息，请参阅《Amazon Organizations 用户指南》中的服务控制策略（SCP）。
- 仅支持 TLS 1.3 的策略支持向前保密（FS）。支持 TLS 1.3 和 TLS 1.2 且仅包含 TLS_* 和 ECDHE_* 格式密码的策略也提供 FS。
- 应用程序负载均衡器支持使用 PSK (TLS 1.3) 和会话 IDs/session 票证（TLS 1.2 及更早版本）恢复 TLS。只有连接到相同的应用程序负载均衡器 IP 地址时才支持恢复。未实现 0-RTT 数据功能和 early_data 扩展。
- Application Load Balancer 不支持自定义安全策略。
- 应用程序负载均衡器仅支持目标连接的 SSL 重新协商。

后端连接

- 您可以选择用于前端连接但不能选择用于后端连接的安全策略。后端连接的安全策略取决于侦听器安全策略。如果有听众在使用：
  - RFC 9151 策略（包括任何互操作策略）-后端连接使用 `ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07`

- FIPS 后量子 TLS 策略-后端连接使用 ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09
- FIPS 策略-后端连接使用 ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04
- Post-quantum TLS 策略-后端连接使用 ELBSecurityPolicy-TLS13-1-0-PQ-2025-09
- TLS 1.3 政策-后端连接使用 ELBSecurityPolicy-TLS13-1-0-2021-06
- 其他 TLS 策略-后端连接使用 ELBSecurityPolicy-2016-08

安全策略

- 示例 describe-ssl-policies 命令
- TLS 安全策略
  - 按策略划分的协议
  - 按策略划分的密码
  - 按密码划分的策略
- FIPS 安全策略
  - 按策略划分的协议
  - 按策略划分的密码
  - 按密码划分的策略
- RFC 9151 (CNSA 1.0) 安全政策
  - 按策略划分的协议
  - 按策略划分的密码
  - 按密码划分的策略
- FS 支持的策略
  - 按策略划分的协议
  - 按策略划分的密码
  - 按密码划分的策略

示例 describe-ssl-policies 命令

您可以使用 describe-ssl-policies Amazon CLI 命令描述安全策略的协议和密码，或者找到满足您需求的策略。

以下示例描述了指定的策略。

```
aws elbv2 describe-ssl-policies \
    --names "ELBSecurityPolicy-TLS13-1-2-Res-2021-06"
```

以下示例列出了策略名称中包含指定字符串的策略。

```
aws elbv2 describe-ssl-policies \
    --query "SslPolicies[?contains(Name,'FIPS')].Name"
```

以下示例列出了支持指定协议的策略。

```
aws elbv2 describe-ssl-policies \
    --query "SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name"
```

以下示例列出了支持指定密码的策略。

```
aws elbv2 describe-ssl-policies \
    --query "SslPolicies[?Ciphers[?contains(Name,'TLS_AES_128_GCM_SHA256')]].Name"
```

以下示例列出了不支持指定密码的策略。

```
aws elbv2 describe-ssl-policies \
    --query 'SslPolicies[?length(Ciphers[?starts_with(Name,`AES128-GCM-SHA256`)]) ==
 `0`].Name'
```

TLS 安全策略

您可以使用 TLS 安全策略来满足需要禁用某些 TLS 协议版本的合规性和安全标准，或者支持需要已弃用密码的旧客户端。

仅支持 TLS 1.3 的策略支持向前保密 (FS)。支持 TLS 1.3 和 TLS 1.2 且仅包含 TLS_* 和 ECDHE_* 格式密码的策略也提供 FS。

内容

按策略划分的协议

下表描述了每个 TLS 安全策略支持的协议。

安全策略	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-2021-06	是	没有	没有	没有
ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	是	没有	没有	没有
ELBSecurityPolicy-TLS13-1-2-2021-06	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Res-2021-06	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-1-2021-06	是	是	是	没有
ELBSecurityPolicy-TLS13-1-0-2021-06	是	是	是	是
ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	是	是	是	是
ELBSecurityPolicy-TLS-1-2-Ext-2018-06	没有	是	没有	没有
ELBSecurityPolicy-TLS-1-2-2017-01	没有	是	没有	没有
ELBSecurityPolicy-TLS-1-1-2017-01	没有	是	是	没有
ELBSecurityPolicy-2016-08	没有	是	是	是

按策略划分的密码

下表描述了每个 TLS 安全策略支持的密码。

安全策略	密码
ELBSecurityPolicy-TLS13-1-3-2021-06 ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256
ELBSecurityPolicy-TLS13-1-2-2021-06 ELBSecurityPolicy-TLS13-1-2-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-TLS13-1-2-Res-2021-06 ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384
ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256

安全策略	密码
	• ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256

安全策略	密码
ELBSecurityPolicy-TLS13-1-1-2021-06	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS13-1-0-2021-06 ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS-1-2-Ext-2018-06	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS-1-2-2017-01	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256

安全策略	密码
ELBSecurityPolicy-TLS-1-1-2017-01	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-2016-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

按密码划分的策略

下表描述了支持每个密码的 TLS 安全策略。

密码名称	安全策略	密码套件
OpenSSL – TLS_AES_128_GCM_SHA256 IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	1301

密码名称	安全策略	密码套件
	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	

密码名称	安全策略	密码套件
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	1302

密码名称	安全策略	密码套件
OpenSSL – TLS_CHACHA20_POLY1305_SHA256 IANA – TLS_CHACHA20_POLY1305_SHA256	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	1303

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02b

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02f

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c023

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES128-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c027

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c009
OpenSSL — ECDHE-RSA-AES128-SHA IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c013

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02c

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c030

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c024

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES256-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c028

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c00a
OpenSSL — ECDHE-RSA-AES256-SHA IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c014

密码名称	安全策略	密码套件
OpenSSL — AES128-GCM-SHA256 IANA：TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	9c

密码名称	安全策略	密码套件
OpenSSL — AES128-SHA256 IANA：TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	3c

密码名称	安全策略	密码套件
OpenSSL — AES128-SHA IANA：TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	2f

密码名称	安全策略	密码套件
OpenSSL — AES256-GCM-SHA384 IANA：TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	9d

密码名称	安全策略	密码套件
OpenSSL — AES256-SHA256 IANA：TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	3d

密码名称	安全策略	密码套件
OpenSSL — AES256-SHA IANA：TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	35

FIPS 安全策略

联邦信息处理标准（FIPS）是美国和加拿大政府标准，其中规定了对保护敏感信息的加密模块的安全要求。要了解更多信息，请参阅 Amazon Cloud 安全性合规性页面上的美国联邦信息处理标准（FIPS）140。

所有 FIPS 策略都使用 AWS-LC FIPS 验证的加密模块。要了解更多信息，请参阅 NIST AWS-LC 加密模块验证计划网站上的加密模块页面。

> **Important**
>
> 策略 `ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04` 和 `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04` 只是为了与旧版兼容而提供。虽然他们使用 FIPS140 模块使用 FIPS 加密，但它们可能不符合 NIST 最新的 TLS 配置指南。

内容

- 按策略划分的密码
- 按密码划分的策略

按策略划分的协议

下表描述了每个 FIPS 安全策略支持的协议。

安全策略	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04	是	没有	没有	没有
ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09	是	没有	没有	没有
ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04	是	是	是	没有
ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04	是	是	是	是
ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	是	是	是	是

按策略划分的密码

下表描述了每个 FIPS 安全策略支持的密码。

安全策略	密码
ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384
ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256

安全策略	密码
	• ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全策略	密码
ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

按密码划分的策略

下表描述了支持每个密码的 FIPS 安全策略。

密码名称	安全策略	密码套件
OpenSSL – TLS_AES_128_GCM_SHA256 IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04	1301

密码名称	安全策略	密码套件
	• ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	

密码名称	安全策略	密码套件
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	1302

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02b

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02f

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c023

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES128-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c027

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c009
OpenSSL — ECDHE-RSA-AES128-SHA IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c013

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02c

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c030

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c024

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES256-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c028

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c00a
OpenSSL — ECDHE-RSA-AES256-SHA IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c014

密码名称	安全策略	密码套件
OpenSSL — AES128-GCM-SHA256 IANA：TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	9c
OpenSSL — AES128-SHA256 IANA：TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	3c

密码名称	安全策略	密码套件
OpenSSL — AES128-SHA IANA：TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	2f
OpenSSL — AES256-GCM-SHA384 IANA：TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	9d

密码名称	安全策略	密码套件
OpenSSL — AES256-SHA256 IANA：TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	3d
OpenSSL — AES256-SHA IANA：TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	35

RFC 9151 (CNSA 1.0) 安全政策

Application Load Balancer 支持可帮助您遵守 RFC 9151 的安全策略，该策略定义了美国国家安全局 (NSA) 规定的商业国家安全算法 (CNSA) 1.0 套件的 TLS 要求。RFC 9151 规定了如何使用带有 TLS 1.2 和 TLS 1.3 协议的 CNSA 套件，定义了符合政府安全标准的安全通信的加密要求。要了解有关 RFC 9151 的更多信息，请参阅 R FC 9151。

RFC 9151 策略分为两类：

- 严格的政策 — 强制执行严格的 RFC 9151 密码和签名方案要求。当您的所有客户端都支持 RFC 9151 时，请使用这些选项。
- 互操作策略 — 支持兼容 RFC 9151 和非 RFC 9151 的密码和签名方案，以帮助逐步过渡到 RFC 9151 合规性。如果您不确定是否所有客户端都支持 RFC 9151，或者您想避免在过渡期间中断客户端，请使用这些选项。所有互操作策略的策略名称中都包含 “INTEROP”。

Amazon 建议从互操作策略开始`ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07`，该策略支持可以协商经典 TLS 1.3、TLS 1.2 或严格的 RFC 9151 算法的客户端，从而最大限度地减少中断。您可以逐渐转向更严格的政策，因为您的客户可以协商严格的RFC 9151。您可以使用 ALB 连接日志中的`tls_protocol`、`tls_cipher`、和`tls_keyexchange`字段来监控客户端的连接情况。

> **Important**
>
> 当您为侦听器选择 RFC 9151 安全策略时，负载均衡器将`ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07`用于与目标和其他服务的后端连接。但是，负载均衡器无法保证或强制执行出口连接（包括与目标的连接）或客户配置的外部服务（例如第三方身份提供商或身份验证端点）的 RFC 9151 合规性。
> 您有责任确保以下几点：
>
> - 您的目标和您配置的任何外部服务都可以支持后端连接策略中的协议和密码。
> - 为了在负载均衡器和您的目标之间严格遵守 RFC 9151，您的目标必须实施符合 RFC 9151 的证书和密码。
> - 如果您的后端目标仅支持 TLS 1.0 或 TLS 1.1，则连接将失败。您必须更新目标上的协议和密码，使其与策略支持的密码保持一致。`ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07`

内容

按策略划分的协议

下表描述了每个 RFC 9151 安全策略支持的协议。

安全策略	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-RFC9151-FIPS-2023-07	是	没有	没有	没有
ELBSecurityPolicy-TLS13-1-2-RFC9151-FIPS-2023-07	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-Ext0-RFC9151-FIPS-2023-07	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07	是	是	没有	没有
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	是	是	没有	没有

按策略划分的密码

下表描述了每个 RFC 9151 安全策略支持的密码。

安全策略	密码
ELBSecurityPolicy-TLS13-1-3-RFC9151-FIPS-2023-07	• TLS_AES_256_GCM_SHA384
ELBSecurityPolicy-TLS13-1-2-RFC9151-FIPS-2023-07	• TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384

安全策略	密码
ELBSecurityPolicy-TLS13-1-2-Ext0-RFC9151-FIPS-2023-07	• TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • AES256-GCM-SHA384
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07	• TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07	• TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256

安全策略	密码
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07	• TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA

安全策略	密码
ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	• TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA

按密码划分的策略

下表描述了支持每种密码的 RFC 9151 安全策略。

密码名称	安全策略	密码套件
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-RFC9151-FIPS-2023-07	1302

密码名称	安全策略	密码套件
	• ELBSecurityPolicy-TLS13-1-2-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-Ext0-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	
OpenSSL – TLS_AES_128_GCM_SHA256 IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	1301

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-Ext0-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c02c
OpenSSL — ECDHE-RSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-Ext0-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c030
OpenSSL — AES256-GCM-SHA384 IANA：TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext0-RFC9151-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	9d

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c02b
OpenSSL — ECDHE-RSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP1-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c02f
OpenSSL — ECDHE-ECDSA-AES256-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c024
OpenSSL — ECDHE-RSA-AES256-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c028

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c023
OpenSSL — ECDHE-RSA-AES128-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP2-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c027
OpenSSL — ECDHE-ECDSA-AES256-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c00a
OpenSSL — ECDHE-RSA-AES256-SHA IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c014
OpenSSL — ECDHE-ECDSA-AES128-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c009

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES128-SHA IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP3-FIPS-2023-07 • ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	c013
OpenSSL — AES256-SHA256 IANA：TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	3d
OpenSSL — AES256-SHA IANA：TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	35
OpenSSL — AES128-GCM-SHA256 IANA：TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	9c
OpenSSL — AES128-SHA256 IANA：TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	3c
OpenSSL — AES128-SHA IANA：TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-RFC9151-INTEROP4-FIPS-2023-07	2f

FS 支持的策略

FS（向前保密）支持的安全策略通过使用唯一的随机会话密钥，提供了防止加密数据被窃听的额外保障。即使秘密的长期密钥被泄露，这也可以防止对捕获的数据进行解码。

本节中的策略支持 FS，且其名称中包含“FS”字样。但是，这些并不是唯一支持 FS 的策略。仅支持 TLS 1.3 的策略支持向前保密 (FS)。支持 TLS 1.3 和 TLS 1.2 且仅包含 TLS_* 和 ECDHE_* 格式密码的策略也提供 FS。

内容

按策略划分的协议

下表描述了每个 FS 支持的安全策略支持的协议。

安全策略	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-FS-1-2-Res-2020-10	没有	是	没有	没有
ELBSecurityPolicy-FS-1-2-Res-2019-08	没有	是	没有	没有
ELBSecurityPolicy-FS-1-2-2019-08	没有	是	没有	没有
ELBSecurityPolicy-FS-1-1-2019-08	没有	是	是	没有
ELBSecurityPolicy-FS-2018-06	没有	是	是	是

按策略划分的密码

下表描述了每个 FS 支持的安全策略支持的密码。

安全策略	密码
ELBSecurityPolicy-FS-1-2-Res-2020-10	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384

安全策略	密码
ELBSecurityPolicy-FS-1-2-Res-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-FS-1-2-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

安全策略	密码
ELBSecurityPolicy-FS-1-1-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA
ELBSecurityPolicy-FS-2018-06	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

按密码划分的策略

下表描述了支持每个密码的 FS 支持的安全策略。

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02b
OpenSSL — ECDHE-RSA-AES128-GCM-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02f
OpenSSL — ECDHE-ECDSA-AES128-SHA256 IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c023
OpenSSL — ECDHE-RSA-AES128-SHA256 IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c027
OpenSSL — ECDHE-ECDSA-AES128-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c009

密码名称	安全策略	密码套件
OpenSSL — ECDHE-RSA-AES128-SHA IANA：TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c013
OpenSSL — ECDHE-ECDSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02c
OpenSSL — ECDHE-RSA-AES256-GCM-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c030
OpenSSL — ECDHE-ECDSA-AES256-SHA384 IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c024
OpenSSL — ECDHE-RSA-AES256-SHA384 IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c028

密码名称	安全策略	密码套件
OpenSSL — ECDHE-ECDSA-AES256-SHA IANA：TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c00a
OpenSSL — ECDHE-RSA-AES256-SHA IANA：TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c014

为您的 Application Load Balancer 创建 HTTPS 侦听器

侦听器检查连接请求。您可在创建负载均衡器时定义侦听器，并可随时向负载均衡器添加侦听器。

要创建 HTTPS 侦听器，您必须在负载均衡器上部署至少一个 SSL 服务器证书。负载均衡器先使用服务器证书终止前端连接，再解密来自客户端的请求，然后将请求发送到目标。您还必须指定一个安全策略，以用于协商客户端与负载均衡器之间的安全连接。

如果需要将加密流量传输至目标且负载均衡器不对其进行解密，则可以创建一个使用端口 443 上的 TCP 侦听器的网络负载均衡器或经典负载均衡器。通过 TCP 侦听器，负载均衡器将加密流量传递到目标，而不会对其进行解密。

此页面上的信息可帮助您为负载均衡器创建 HTTPS 侦听器。要向您的负载均衡器添加 HTTP 侦听器，请参阅 为您的 Application Load Balancer 创建 HTTP 侦听器。

先决条件

- 要将转发操作添加到默认侦听器规则，您必须指定可用的目标组。有关更多信息，请参阅 为您的应用程序负载均衡器创建目标组。
- 您可以在多个侦听器中指定同一个目标组，但这些侦听器必须属于同一个负载均衡器。要将目标组与负载均衡器结合使用，您必须确认其没有被任何其他负载均衡器的侦听器使用。
- 应用程序负载均衡器不支持 ED25519 密钥。

添加 HTTPS 侦听器

您需要配置一个侦听器，以及用于从客户端连接到负载均衡器的协议和端口。有关更多信息，请参阅侦听器配置。

创建安全侦听器时，您必须指定一个安全策略和一个证书。要将证书添加到证书列表，请参阅the section called “将证书添加到证书列表”。

您必须为侦听器配置一个默认规则。您可以在创建侦听器后添加其他侦听器规则。有关更多信息，请参阅 侦听器规则。

Console

添加 HTTPS 侦听器

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择添加侦听器。
5. 对于协议，选择 HTTPS。保留默认端口或输入其他端口。
6. （可选）对于Pre-routing 操作，请选择以下操作之一：
   - 对用户进行身份验证-选择身份提供者并提供所需的信息。有关更多信息，请参阅 使用 Application Load Balancer 验证用户身份。
   - 验证令牌 — 输入 JWKS 端点、问题和任何其他声明。有关更多信息，请参阅 使用 Application Load Balancer 验证 JWT。
7. 在 “路由操作” 中，选择以下操作之一：
   - 转发到目标组：选择一个目标组。要添加其他目标组，请选择添加目标组，然后选择一个目标组，检查相对权重并根据需要更新权重。如果在任何目标组上启用了粘性，则必须启用组级粘性。

     如果没有能满足您需求的目标组，请选择创建目标组，以立即创建一个目标组。有关更多信息，请参阅 创建目标组。
   - 重定向到 URL：在 URI 部分选项卡上分别输入每个部分，或者在完整 URL选项卡上输入完整的地址，从而输入 URL。对于状态代码，根据您的需求选择临时（HTTP 302）或永久（HTTP 301）。

- 返回固定响应：输入要为已删除的客户端请求返回的响应代码。您也可以指定内容类型和响应正文。

8. 对于安全策略，我们会选择推荐的安全策略。您可以根据需要选择其他安全策略。
9. 对于默认 SSL/TLS 证书，请选择默认证书。我们还会将默认证书添加到 SNI 列表中。您可以使用下列选项之一来选择证书：

   - 来自 ACM：从证书（来自 ACM）中选择证书，这将显示来自 Amazon Certificate Manager 的可用证书。
   - 从 IAM — 从证书（来自 IAM）中选择证书，该证书会显示您导入到的证书 Amazon Identity and Access Management。
   - 导入证书：选择证书的目的地；导入到 ACM 或导入到 IAM。对于证书私钥，请复制并粘贴私钥文件 (PEM-encoded) 的内容。对于证书正文，复制并粘贴公钥证书文件的内容 (PEM-encoded)。对于证书链，请复制并粘贴证书链文件 (PEM-encoded) 的内容，除非您使用的是自签名证书，并且浏览器是否隐式接受该证书并不重要。

10. （可选）要启用双向身份验证，请在客户端证书处理下启用双向身份验证（mTLS）。

    默认模式为传递。如果选择使用信任存储进行验证：

    - 默认情况下，客户端证书已过期的连接会被拒绝。要更改此行为，请展开高级 mTLS 设置，然后在客户端证书过期下选择允许过期的客户证书。
    - 对于信任存储，请选择一个现有信任存储，也可选择新建信任存储并提供所需的信息。

11. （可选）要添加标签，请展开侦听器标签。选择添加新标签，然后输入标签键和标签值。
12. 选择添加侦听器。

Amazon CLI

创建 HTTPS 侦听器

使用 create-listener 命令。以下示例会创建一个 HTTPS 侦听器，其默认规则会将流量转发到指定目标组。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol HTTPS \
    --port 443 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn \
    --ssl-policy ELBSecurityPolicy-TLS13-1-2-2021-06 \
```

```
    --certificates certificate-arn
```

CloudFormation

创建 HTTPS 侦听器

定义 AWS::ElasticLoadBalancingV2:: Listener 类型的资源。以下示例会创建一个 HTTPS 侦听器，其默认规则会将流量转发到指定目标组。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
      Certificates:
        - CertificateArn: certificate-arn
```

为您的 Application Load Balancer 更新 HTTPS 侦听器

创建 HTTPS 侦听器后，您可以替换默认证书、更新证书列表或替换安全策略。

任务

- 替换默认证书
- 将证书添加到证书列表
- 从证书列表中删除证书
- 更新安全策略
- HTTP 标头修改

替换默认证书

您可以使用以下过程替换侦听器的默认证书。有关更多信息，请参阅 默认证书。

Console

要替换默认证书

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在 “监听器和规则” 选项卡上，选择Protocol:Port列中的文本以打开监听器的详细信息页面。
5. 在证书选项卡上，选择更改默认值。
6. 在 ACM 和 IAM 证书表中，选择新的默认证书。
7. （可选）默认情况下，我们选择将之前的默认证书添加到侦听器证书列表中。我们建议您保持此选项的选中状态，除非您当前没有用于 SNI 的侦听器证书且依赖于 TLS 会话恢复功能。
8. 选择另存为默认值。

Amazon CLI

要替换默认证书

使用 modify-listener 命令。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --certificates CertificateArn=new-default-certificate-arn
```

CloudFormation

替换默认证书

更新 AWS::ElasticLoadBalancingV2:: Listener。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
```

```
        TargetGroupArn: !Ref myTargetGroup
    SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
    Certificates:
      - CertificateArn: new-default-certificate-arn
```

将证书添加到证书列表

您可使用以下过程将证书添加到侦听器的证书列表。如果您使用创建了侦听器 Amazon Web Services 管理控制台，则我们会为您将默认证书添加到证书列表中。否则，证书列表为空。您可以将默认证书添加到证书列表，以确保此证书与 SNI 协议一起使用，即使默认证书被替换亦不例外。有关更多信息，请参阅 应用程序负载均衡器的 SSL 证书。

Console

要将证书添加到证书列表

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在 “监听器和规则” 选项卡上，选择Protocol:Port列中的文本以打开监听器的详细信息页面。
5. 选择 Certificates (证书) 选项卡。
6. 要将默认证书添加到列表，请选择将默认证书添加到列表。
7. 要将非默认证书添加大列表，请执行以下操作：

   a. 选择添加证书。
   b. 要添加已由 ACM 或 IAM 管理的证书，请选中证书对应的复选框并选择在下面以待注册的形式添加。
   c. 要添加未由 ACM 或 IAM 管理的证书，请选择导入证书，完成表格，然后选择导入。
   d. 选择添加待处理证书。

Amazon CLI

将证书添加到证书列表

使用 add-listener-certificates 命令。

```
aws elbv2 add-listener-certificates \
```

```
    --listener-arn listener-arn \
    --certificates \
        CertificateArn=certificate-arn-1 \
        CertificateArn=certificate-arn-2 \
        CertificateArn=certificate-arn-3
```

CloudFormation

要将证书添加到证书列表

定义类型为 AWS::ElasticLoadBalancingV2:: ListenerCertificate 的资源。

```
Resources:
  myCertificateList:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerCertificate'
    Properties:
      ListenerArn: !Ref myTLSListener
      Certificates:
        - CertificateArn: "certificate-arn-1"
        - CertificateArn: "certificate-arn-2"
        - CertificateArn: "certificate-arn-3"
```

从证书列表中删除证书

您可以使用以下过程从 HTTPS 侦听器的证书列表中删除证书。移除证书后，侦听器将无法再使用该证书建立连接。为确保客户端不受影响，在从列表中删除证书之前，请先将新的证书添加至列表并确认连接功能正常。

要删除 TLS 侦听器的默认证书，请参阅替换默认证书。

Console

要从证书列表中删除证书

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在 “监听器和规则” 选项卡上，选择Protocol:Port列中的文本以打开监听器的详细信息页面。
5. 在证书选项卡上，选中证书对应的复选框，然后选择删除。
6. 提示进行确认时，输入 **confirm**，然后选择删除。

Amazon CLI

从证书列表中移除证书

使用 remove-listener-certificates 命令。

```
aws elbv2 remove-listener-certificates \
    --listener-arn listener-arn \
    --certificates CertificateArn=certificate-arn
```

更新安全策略

在创建 HTTPS 侦听器时，您可以选择满足您的需求的安全策略。添加新的安全策略后，您可以将 HTTPS 侦听器更新为使用此新安全策略。Application Load Balancer 不支持自定义安全策略。有关更多信息，请参阅 应用程序负载均衡器的安全策略。

如果负载均衡器处理大量流量，则更新安全策略可能会导致中断。为降低负载均衡器处理大量流量时的中断风险，请创建额外的负载均衡器来分担流量，或请求 LCU 预留。

兼容性

- 连接到同一负载均衡器的所有安全侦听器都必须使用兼容的安全策略。要将负载均衡器的所有安全侦听器迁移到与当前使用的安全策略不兼容的安全策略，请删除除一个安全侦听器之外的所有安全侦听器，更改安全侦听器的安全策略，然后创建其他安全侦听器。
  - FIPS 后量子 TLS 策略和 FIPS 策略——兼容
  - Post-quantum TLS 策略和 FIPS 或 FIPS 后量子 TLS 策略——兼容
  - TLS 策略（非 FIPS、非后量子）和 FIPS 或 FIPS 后量子 TLS 策略——不兼容
  - TLS 策略（非 FIPS、非后量子）和后量子 TLS 策略-不兼容

Console

要更新安全策略

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在 “监听器和规则” 选项卡上，选择Protocol:Port列中的文本以打开监听器的详细信息页面。

5. 在安全选项卡上，选择编辑安全侦听器设置。
6. 在安全侦听器设置部分的安全策略下，选择新的安全策略。
7. 选择保存更改。

Amazon CLI

要更新安全策略

使用 modify-listener 命令。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --ssl-policy ELBSecurityPolicy-TLS13-1-2-Res-2021-06
```

CloudFormation

要更新安全策略

使用新的安全策略更新 AWS::ElasticLoadBalancingV2:: Listener 资源。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
      Certificates:
        - CertificateArn: certificate-arn
```

HTTP 标头修改

通过修改 HTTP 标头，您可以重命名特定负载均衡器生成的标头、插入特定的响应标头以及禁用服务器响应标头。应用程序负载均衡器的请求标头和响应标头都支持标头修改。

有关更多信息，请参阅 为应用程序负载均衡器启用 HTTP 标头修改。

Application Load Balancer 的侦听器规则

应用程序负载均衡器的侦听器规则决定了其将请求路由到目标的方式。侦听器收到请求时，将按各规则的优先级顺序（从编号最小的规则开始）评估请求。每条规则都包括要满足的条件以及满足规则条件时要执行的操作。这种灵活的路由机制让您可以实现复杂的流量分配模式，在单个负载均衡器后支持多个应用程序或微服务，以及根据应用程序的特定要求自定义请求处理。

规则基础知识

- 每条规则都由以下部分组成：优先级、操作、条件和可选的转换。
- 每个规则操作都包含类型以及执行该操作所需的信息。
- 每个规则条件都包含类型以及评估该条件所需的信息。
- 每个规则转换都包含一个要匹配的正则表达式和一个替换字符串。
- 规则条件和规则转换中使用的正则表达式不支持以下功能：lookheads、lookbehind、反向引用、原子组、所有格量词、子例程、递归和 Unicode 字符类（例如）。`\p{L}`
- 创建侦听器时，请为默认规则定义操作。默认规则不能有条件或转换。如果未满足任何其他规则的任何条件，则将执行默认规则的操作。
- 规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。不能更改默认规则的优先级。
- 每条规则必须包含以下操作之一：`forward`、`redirect` 或 `fixed-response`，并且其必须为要执行的最后一个操作。
- 除默认规则以外的每条规则可以选择包含以下条件之一：`host-header`、`http-request-method`、`path-pattern` 和 `source-ip`。此外还可以选择包含以下两个条件之一或全部：`http-header` 和 `query-string`。
- 除默认规则以外的每条规则都可以选择包含一个主机标头重写转换和一个 URL 重写转换。
- 每个条件最多可以指定三个比较字符串，每条规则最多可以指定五个比较字符串。

内容

- 删除应用程序负载均衡器的侦听器规则

侦听器规则的操作类型

操作决定了满足侦听器规则的条件时，负载均衡器将会如何处理请求。每条规则必须至少有一个操作来指定如何处理匹配的请求。每个规则操作都有一个类型和配置信息。应用程序负载均衡器支持侦听器规则的以下操作类型。

操作类型

`authenticate-cognito`

[HTTPS 侦听器] 使用 Amazon Cognito 验证用户身份。有关更多信息，请参阅 用户身份验证。

`authenticate-oidc`

[HTTPS 侦听器] 使用符合 OpenID Connect (OIDC) 条件的身份提供商验证用户身份。有关更多信息，请参阅 用户身份验证。

`fixed-response`

返回自定义 HTTP 响应。有关更多信息，请参阅 固定响应操作。

`forward`

将请求转发到指定目标组。有关更多信息，请参阅 转发操作。

`jwt-validation`

验证客户端请求中的 JWT 访问令牌。有关更多信息，请参阅 JWT 验证。

`redirect`

将请求从一个 URL 重定向到另一个。有关更多信息，请参阅 重定向操作。

操作基础知识

- 每条规则必须包含以下路由操作之一：`forward`、`redirect` 或 `fixed-response`，并且该操作必须为要执行的最后一个操作。
- HTTPS 侦听器可以包含具有用户身份验证操作和路由操作的规则。
- 包含多个操作时，将首先执行优先级最低的操作。
- 如果协议版本是 gRPC 或 HTTP/2，则唯一支持的操作是 `forward` 操作。

固定响应操作

fixed-response 操作会丢弃客户端请求并返回自定义 HTTP 响应。您可以使用此操作返回 2XX、4XX 或 5XX 响应代码和可选的消息。

采取 fixed-response 操作时，操作和重定向目标的 URL 记录在访问日志中。有关更多信息，请参阅 访问日志条目。成功 fixed-response 操作的计数在 HTTP_Fixed_Response_Count 指标中报告。有关更多信息，请参阅 Application Load Balancer 指标。

Example固定响应操作示例

您可以在创建或修改规则时指定操作。有关更多信息，请参阅 create-rule 和 modify-rule 命令。以下操作发送具有指定状态代码和消息正文的固定响应。

```
[
  {
      "Type": "fixed-response",
      "FixedResponseConfig": {
          "StatusCode": "200",
          "ContentType": "text/plain",
          "MessageBody": "Hello world"
      }
  }
]
```

转发操作

forward 操作会将请求路由至其目标组。在添加 forward 操作之前，请创建目标组并向其添加目标。有关更多信息，请参阅 创建目标组。

将流量分配到多个目标组

如果为某个 forward 操作指定多个目标组，您必须为每个目标组指定权重。每个目标组权重都是一个介于 0 到 999 之间的值。对于将侦听器规则与加权目标组匹配的请求，会根据这些目标组的权重分配给这些目标组。例如，如果指定两个目标组，每个目标组的权重为 10，则每个目标组将接收一半的请求。如果指定两个目标组，一个权重为 10，另一个权重为 20，则权重为 20 的目标组接收的请求将是另一个目标组的两倍。

如果您配置了一条在加权目标组之间分配流量的规则，并且其中一个目标组为空或全部为运行不正常的目标，则负载均衡器不会自动失效转移到有运行正常目标的目标组。

粘性会话和加权目标组

默认情况下，配置规则以便在加权目标组之间分配流量时，并不能保证支持粘性会话。为了确保支持粘性会话，请为规则启用目标组粘性。当负载均衡器首次将请求路由到加权目标组时，它会生成一个名为的 Cookie AWSALBTG ，用于对有关选定目标组的信息进行编码，对 Cookie 进行加密，并将该 Cookie 包含在对客户端的响应中。客户端应该在向负载均衡器的后续请求中包含它收到的 cookie。当负载均衡器收到与启用目标组粘性的规则匹配并且包含 Cookie 的请求时，请求将路由到 Cookie 中指定的目标组。

Application Load Balancer 不支持 URL 编码的 Cookie 值。

对于 CORS（跨源资源共享）请求，某些浏览器需要 `SameSite=None; Secure` 来启用粘性。在这种情况下，Elastic Load Balancing 会生成第二个 cookie AWSALBTGCORS，其中包含与原始粘性 cookie 相同的信息以及此`SameSite`属性。客户端会同时收到这两个 Cookie。

带有一个目标组的转发操作示例

您可以在创建或修改规则时指定操作。有关更多信息，请参阅 create-rule 和 modify-rule 命令。以下操作将请求转发到指定的目标组。

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
              }
          ]
      }
  }
]
```

具有加权目标组的转发操作示例

下面的操作将根据每个目标组的权重，将请求转发到两个指定的目标组。

```
[
  {
```

```
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
                  "Weight": 10
              },
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
                  "Weight": 20
              }
          ]
      }
  }
]
```

启用粘性的转发操作示例

如果您具有一个包含多个目标组的转发操作，并且一个或多个目标组已启用了粘性会话，则必须启用目标组粘性。

下面的操作将请求转发到两个指定的目标组，并启用了目标组粘性。对于不包含粘性 Cookie 的请求，将根据每个目标组的权重进行传输。

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
                  "Weight": 10
              },
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
                  "Weight": 20
              }
          ],
          "TargetGroupStickinessConfig": {
```

```
            "Enabled": true,
            "DurationSeconds": 1000
        }
    }
  }
]
```

重定向操作

`redirect` 操作会将客户端请求从一个 URL 重定向到另一个 URL。根据需要，您可以将重定向配置为临时 (HTTP 302) 或永久 (HTTP 301)。

URI 包括以下组成部分：

```
protocol://hostname:port/path?query
```

您必须修改以下至少一个组成部分以避免重定向循环：协议、主机名、用户名、端口或路径。未修改的任何组成部分将保留其原始值。

协议

协议（HTTP 或 HTTPS）。您可以将 HTTP 重定向到 HTTP、将 HTTP 重定向到 HTTPS 以及将 HTTPS 重定向到 HTTPS。不能将 HTTPS 重定向到 HTTP。

hostname

主机名。主机名不区分大小写，长度最多为 128 个字符，由字母数字字符、通配符（* 和？）及连字符 (-) 组成。

port (远程调试端口)

端口（1 到 65535）。

path

绝对路径，以前导“/”开头。路径区分大小写，长度最多为 128 个字符，由字母数字字符、通配符（* 和？）、&（使用 &）及以下特殊字符组成：_-.$/~"'@:+。

查询

查询参数。最大长度为 128 个字符。

您可以通过以下保留关键字，在目标 URL 中重用原始 URL 的 URI 组成部分：

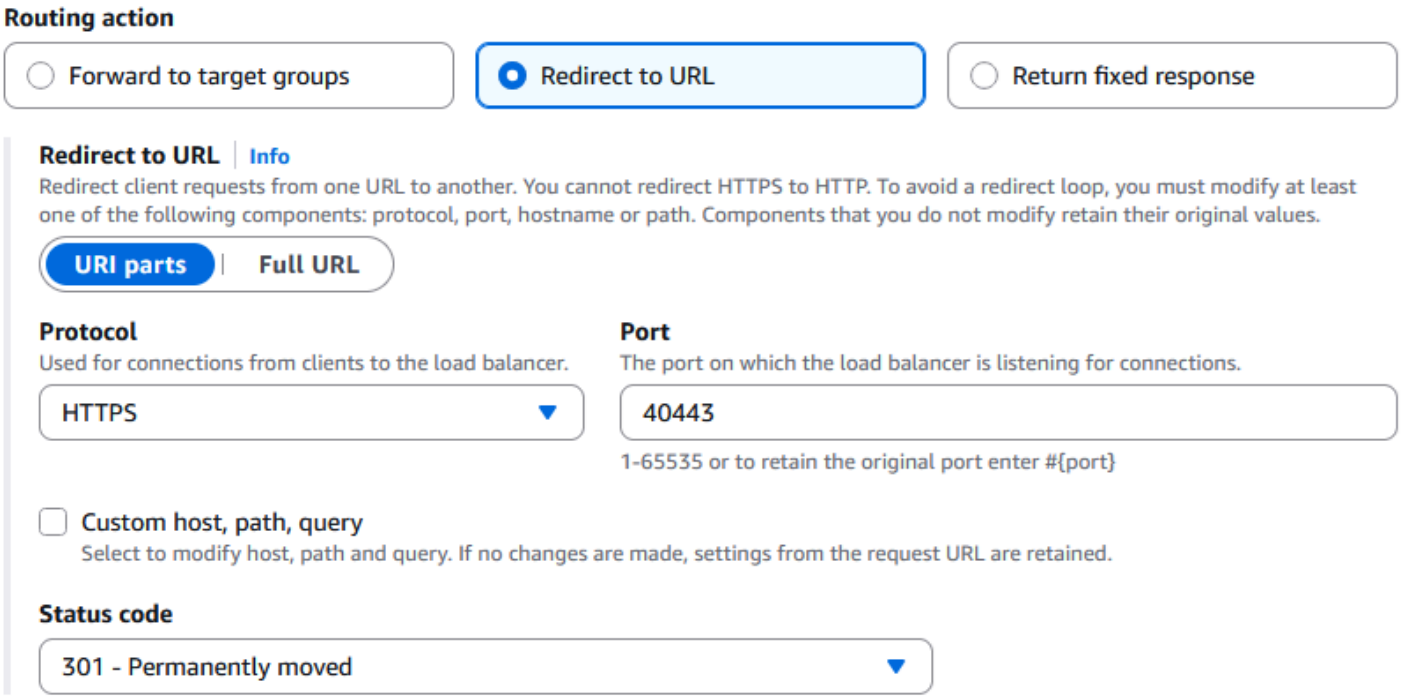
- #{protocol} – 保留协议。在协议和查询组成部分中使用。
- #{host} – 保留域。在主机名、路径和查询组成部分中使用。
- #{port} – 保留端口。在端口、路径和查询组成部分中使用。
- #{path} – 保留路径。在路径和查询组成部分中使用。
- #{query} – 保留查询参数。在查询组成部分中使用。

执行 redirect 操作时，操作记录在访问日志中。有关更多信息，请参阅 访问日志条目。成功 redirect 操作的计数在 HTTP_Redirect_Count 指标中报告。有关更多信息，请参阅 Application Load Balancer 指标。

使用控制台的重定向操作的示例

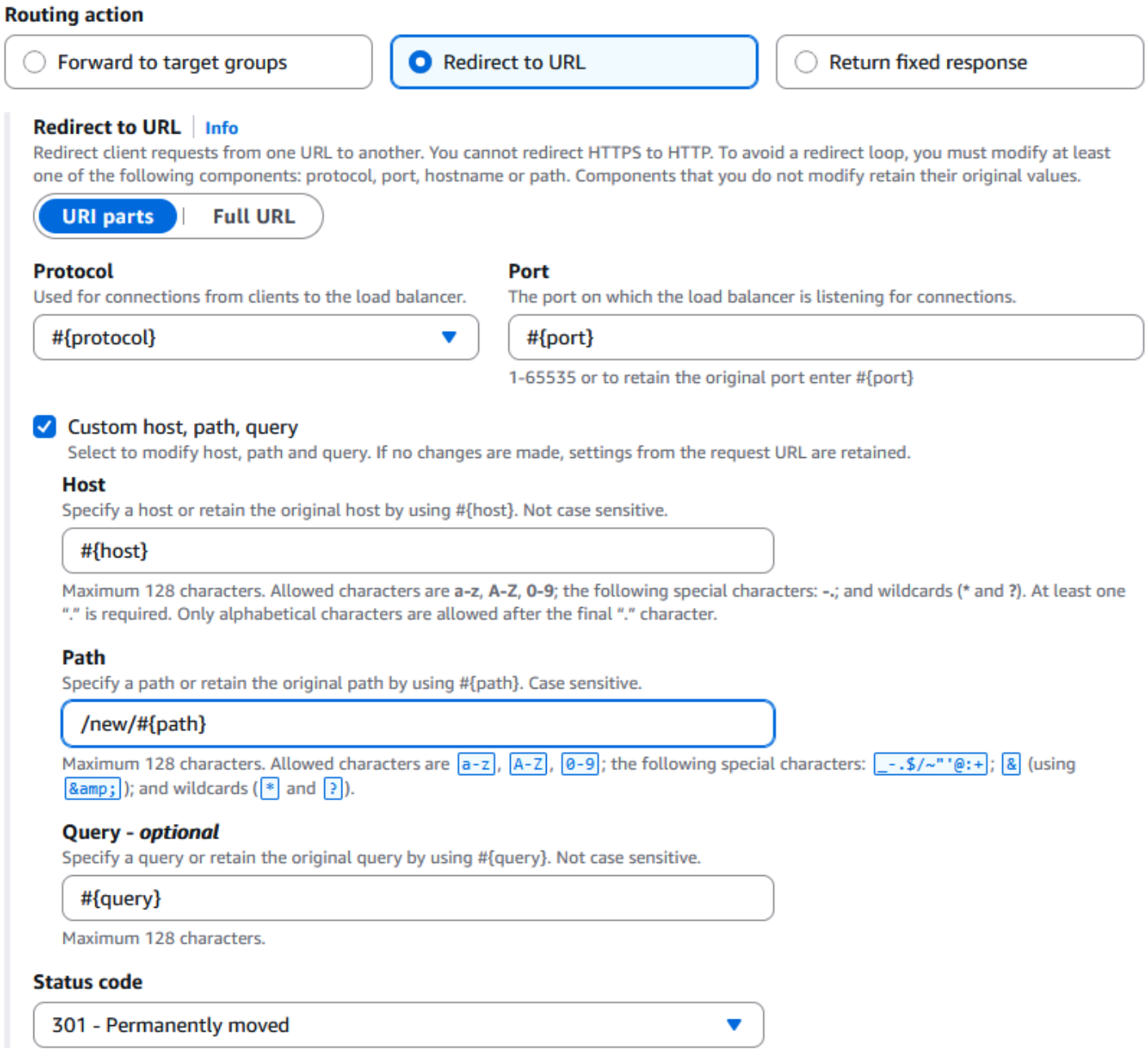
使用 HTTPS 和端口 40443 进行重定向

以下规则设置永久重定向到一个 URL，该 URL 使用 HTTPS 协议和指定的端口 (40443)，但保留原始主机名、路径和查询参数。此屏幕等同于“https://#{host}:40443/#{path}?#{query}”。



使用修改后的路径进行重定向

以下规则设置永久重定向到一个 URL，该 URL 包含原始协议、端口、主机名和查询参数，并使用 #{path} 关键字来创建修改的路径。此屏幕等同于“#{protocol}://#{host}:#{port}/new/#{path}?#{query}”。



使用重定向操作示例 Amazon CLI

使用 HTTPS 和端口 40443 进行重定向

您可以在创建或修改规则时指定操作。有关更多信息，请参阅 create-rule 和 modify-rule 命令。以下操作将 HTTP 请求重定向为端口 443 上的 HTTPS 请求，其主机名、路径和查询字符串与 HTTP 请求相同。

```
--actions '[{
    "Type": "redirect",
```

```
      "RedirectConfig": {
          "Protocol": "HTTPS",
          "Port": "443",
          "Host": "#{host}",
          "Path": "/#{path}",
          "Query": "#{query}",
          "StatusCode": "HTTP_301"
      }
  }]'
```

侦听器规则的条件类型

条件定义了要使侦听器规则生效，传入的请求必须满足的条件。如果请求满足规则的条件，则将按照规则操作指定的方式处理该请求。每个规则条件都有类型和配置信息。应用程序负载均衡器支持侦听器规则的以下条件类型。

条件类型

`host-header`

基于每个请求的主机名进行路由。有关更多信息，请参阅 主机条件。

`http-header`

基于每个请求的 HTTP 标头进行路由。有关更多信息，请参阅 HTTP 标头条件。

`http-request-method`

基于每个请求的 HTTP 请求方法路由。有关更多信息，请参阅 HTTP 请求方法条件。

`path-pattern`

基于请求中的路径模式进行路由 URLs。有关更多信息，请参阅 路径条件。

`query-string`

根据查询字符串中的 key/value 成对或值进行路由。有关更多信息，请参阅 查询字符串条件。

`source-ip`

基于每个请求的源 IP 地址进行路由。有关更多信息，请参阅 源 IP 地址条件。

条件基础知识

- 每个规则可以选择没有条件，也可包含以下条件之一：host-header、http-request-method、path-pattern 和 source-ip。每个规则还可以包含以下每个条件中的零个或多个：http-header 和 query-string。
- 使用 host-header、http-header 和 path-pattern 条件时，您可以使用值匹配，也可以使用正则表达式（regex）匹配方法。
- 您可以为每个条件指定最多三个匹配评估。例如，对于每个 http-header 条件，您最多可以指定三个字符串，以与请求中的 HTTP 标头的值进行比较。如果其中一个字符串与 HTTP 标头的值匹配，则满足条件。若要要求所有字符串都匹配，请为每个匹配评估创建一个条件。
- 您可以为每条规则指定最多五个匹配评估。例如，您可以创建一个具有五个条件的规则，其中每个条件都有一个匹配评估。
- 您可以在 http-header、host-header、path-pattern 和 query-string 条件的匹配评估中包含通配符。每条规则的通配符上限为五个。
- 规则仅应用于可见的 ASCII 字符；不包括控制字符（0x00 到 0x1f 和 0x7f）。
- 规则条件中使用的正则表达式不支持以下功能：lookheads、lookbehind、反向引用、原子组、所有格量词、子例程、递归和 Unicode 字符类（例如）。\p{L}

演示

有关演示，请参阅高级请求路由。

主机条件

您可以使用主机条件来定义基于主机标头中的主机名路由请求的规则（也称为基于主机的路由）。这使您能够使用单个负载均衡器支持多个子域和不同的顶级域。

主机名不区分大小写，长度上限为 128 个字符，并且可包含以下任何字符：

- A-Z、a-z、0-9
- - .
- *（匹配 0 个或多个字符）
- ?（完全匹配 1 个字符）

您必须包含至少一个“.”字符。在最后一个“.”字符之后只能包含字母数字字符。

主机名示例

- example.com
- test.example.com
- *.example.com

规则 *.example.com 与 test.example.com 匹配，但与 example.com 不匹配。

Example主机标头条件示例

您可以在创建或修改规则时指定条件。有关更多信息，请参阅 create-rule 和 modify-rule 命令。

Value matching

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "Values": ["*.example.com"]
      }
  }
]
```

Regex matching

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "RegexValues": ["^(.*)\\.example\\.com$"]
      }
  }
]
```

HTTP 标头条件

您可以使用 HTTP 标头条件来配置基于请求的 HTTP 标头路由请求的规则。您可以指定标准或自定义 HTTP 标头字段的名称。标头名称和匹配评估不区分大小写。比较字符串支持以下通配符：*（匹配 0 个或多个字符）和？（完全匹配 1 个字符）。标头名称不支持通配符。

启用应用程序负载均衡器属性 `routing.http.drop_invalid_header_fields` 后，将会丢弃不符合正则表达式 (`A-Z,a-z,0-9`) 的标头名称。也可以添加不符合正则表达式的标头名称。

Example HTTP 标头条件示例

您可以在创建或修改规则时指定条件。有关更多信息，请参阅 create-rule 和 modify-rule 命令。具有与指定字符串之一匹配的 User-Agent 标头的请求满足以下条件。

Value matching

```
[
  {
      "Field": "http-header",
      "HttpHeaderConfig": {
          "HttpHeaderName": "User-Agent",
          "Values": ["*Chrome*", "*Safari*"]
      }
  }
]
```

Regex matching

```
[
  {
      "Field": "http-header",
      "HttpHeaderConfig": {
          "HttpHeaderName": "User-Agent",
          "RegexValues": [".+"]
      }
  }
]
```

HTTP 请求方法条件

您可以使用 HTTP 请求方法条件来配置基于请求的 HTTP 请求方法路由请求的规则。您可以指定标准或自定义 HTTP 方法。匹配评估区分大小写。不支持通配符；因此，方法名称必须完全匹配。

我们建议您以相同的方式路由 GET 和 HEAD 请求，因为这样可以缓存对 HEAD 请求的响应。

Example HTTP 方法条件示例

您可以在创建或修改规则时指定条件。有关更多信息，请参阅 create-rule 和 modify-rule 命令。使用指定方法的请求满足以下条件。

```
[
  {
      "Field": "http-request-method",
      "HttpRequestMethodConfig": {
          "Values": ["CUSTOM-METHOD"]
      }
  }
]
```

路径条件

您可以使用路径条件来定义基于请求中的 URL 路由请求的规则（也称为基于路径的路由）。

路径模式仅应用于 URL 的路径，而不应用于其查询参数。它仅应用于可见的 ASCII 字符；不包括控制字符（0x00 到 0x1f 和 0x7f）。

仅当 URI 规范化之后才执行规则评估。

路径模式区分大小写，长度最多为 128 个字符，并且可包含以下任何字符。

- A-Z、a-z、0-9
- _ - . $ / ~ " ' @ : +
- &（使用 &）
- *（匹配 0 个或多个字符）
- ?（完全匹配 1 个字符）

如果协议版本是 gRPC，则条件可特定于程序包、服务或方法。

示例 HTTP 路径模式

- `/img/*`
- `/img/*/pics`

示例 gRPC 路径模式

- /package
- /package.service/
- /package.service/method

路径模式用于路由请求，而不是更改请求。例如，如果一个规则的路径模式为 `/img/*`，此规则会将 `/img/picture.jpg` 的请求作为 `/img/picture.jpg` 的请求转发给指定目标组。

Example路径模式条件示例

您可以在创建或修改规则时指定条件。有关更多信息，请参阅 create-rule 和 modify-rule 命令。具有包含指定字符串的 URL 的请求满足以下条件。

Value matching

```
[
  {
      "Field": "path-pattern",
      "PathPatternConfig": {
          "Values": ["/img/*"]
      }
  }
]
```

Regex matching

```
[
  {
      "Field": "path-pattern",
      "PathPatternConfig": {
          "RegexValues": ["^\\/api\\/(.*)$"]
      }
  }
]
```

查询字符串条件

您可以使用查询字符串条件来配置基于查询字符串中的 key/value 对或值路由请求的规则。匹配评估不区分大小写。支持以下通配符：*（匹配 0 个或多个字符）和 ?（完全匹配 1 个字符）。

Example查询字符串条件示例

您可以在创建或修改规则时指定条件。有关更多信息，请参阅 create-rule 和 modify-rule 命令。使用包含一 key/value 对 “version=v1” 或任何设置为 “example” 的密钥的查询字符串的请求满足以下条件。

```
[
  {
      "Field": "query-string",
      "QueryStringConfig": {
          "Values": [
            {
                "Key": "version",
                "Value": "v1"
            },
            {
                "Value": "*example*"
            }
          ]
      }
  }
]
```

源 IP 地址条件

您可以使用源 IP 地址条件来配置基于请求的源 IP 地址路由请求的规则。必须以 CIDR 格式指定 IP 地址。您可以同时使用 IPv4 和 IPv6 地址。不支持通配符。不能为源 IP 规则条件指定 `255.255.255.255/32` CIDR。

如果客户端位于代理之后，则这是代理的 IP 地址，而不是客户端的 IP 地址。

X-Forwarded-For标题中的地址不符合此条件。要在 X-Forwarded-For标题中搜索地址，请使用`http-header`条件。

Example来源 IP 条件示例

您可以在创建或修改规则时指定条件。有关更多信息，请参阅 create-rule 和 modify-rule 命令。源 IP 地址位于某个指定的 CIDR 块中的请求满足以下条件。

```
[
  {
      "Field": "source-ip",
```

```
      "SourceIpConfig": {
          "Values": ["192.0.2.0/24", "198.51.100.10/32"]
      }
  }
]
```

侦听器规则的转换

规则转换会在将入站请求路由到目标之前对其进行重写。重写请求不会改变在评估规则条件时做出的路由决定。当客户端发送的 URL 或主机标头与目标预期的标头不同时，这将非常实用。

使用规则转换将修改路径、查询字符串和主机标头的责任转移到给负载均衡器。这样就无需在应用程序代码中添加自定义修改逻辑，也无需依赖第三方代理来执行修改。

应用程序负载均衡器支持侦听器规则的以下转换。

转换

`host-header-rewrite`

重写请求中的主机标头。转换使用正则表达式来比对主机标头中的模式，然后将其替换为替换字符串。

`url-rewrite`

重写请求 URL。转换使用正则表达式来比对请求 URL 中的模式，然后将其替换为替换字符串。

转换基础知识

- 每条规则可以添加一个主机标头重写转换和一个 URL 重写转换。
- 无法向默认规则添加转换。
- 如果没有模式匹配，则会将原始请求发送到目标。
- 如果存在模式匹配但转换失败，我们将返回 HTTP 500 错误。
- 规则转换中使用的正则表达式不支持以下功能：lookheads、lookbehind、反向引用、原子组、所有格量词、子例程、递归和 Unicode 字符类（例如）。`\p{L}`

主机标头重写转换

您可以修改主机标头中指定的域名。

Example主机标头转换示例

您可以在创建或修改规则时指定转换。有关更多信息，请参阅 create-rule 和 modify-rule 命令。以下是一个主机标头转换示例。此示例会将主机标头转换为一个内部端点。

```
[
  {
      "Type": "host-header-rewrite",
      "HostHeaderRewriteConfig": {
          "Rewrites": [
              {
                  "Regex": "^mywebsite-(.+).com$",
                  "Replace": "internal.dev.$1.myweb.com"
              }
          ]
      }
  }
]
```

例如，此转换会将主机标头 `https://mywebsite-example.com/project-a` 重写为 `https://internal.dev.example.myweb.com/project-a`。

URL 重写转换

您可以修改 URL 的路径或查询字符串。通过在负载均衡器级别重写 URL，即使您的后端服务发生变化，您的前端 URLs 也可以保持用户和搜索引擎的一致性。此外还可以简化复杂的 URL 查询字符串，更遍布客户键入。

请注意，不能修改 URL 的协议或端口，只能修改路径和查询字符串。

Example URL 重写转换示例

您可以在创建或修改规则时指定转换。有关更多信息，请参阅 create-rule 和 modify-rule 命令。以下是一个 URL 重写转换示例。此示例会将目录结构转换为查询字符串。

```
[
  {
      "Type": "url-rewrite",
      "UrlRewriteConfig": {
          "Rewrites": [
              {
                  "Regex": "^/dp/([A-Za-z0-9]+)/?$",
```

```
                "Replace": "/product.php?id=$1"
            }
        ]
    }
  }
]
```

例如，此转换会将请求 URL `https://www.example.com/dp/B09G3HRMW` 重写为 `https://www.example.com/product.php?id=B09G3HRMW`。

URL 重写与 URL 重定向的区别

特征	URL 重定向	URL 重写
URL 显示	浏览器地址栏中有变化	浏览器地址栏中无变化
状态代码	使用 301（永久）或 302（临时）	状态代码无变化
Processing	浏览器端	服务器端
常见用途	域名变更、网站整合、修复损坏的链接	清理 URLs 搜索引擎优化，隐藏复杂结构，提供旧版网址映射

为应用程序负载均衡器添加侦听器规则

创建侦听器时，您会定义一条默认规则。您可以随时定义其他规则。每条规则都必须指定一个操作和一个条件，并且可以选择指定转换。有关更多信息，请参阅下列内容：

- 操作类型
- 条件类型
- 转换

Console

添加一项规则

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。

3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择协议：端口列中的文本以打开侦听器的详细信息页面。
5. 在规则选项卡上，选择添加规则。
6. （可选）要为规则指定名称，请展开名称和标签，然后输入名称。要添加其他标签，请选择添加其他标签，然后输入标签键和标签值。
7. 对于每个条件，选择添加条件，选择条件类型，然后提供所需的条件值：

   - 主机标头：选择匹配模式类型并输入主机标头。

     值匹配：最多 128 个字符。不区分大小写。允许的字符是 a-z、A-Z、0-9；以下特殊字符：-_.；以及通配符（* 和？）。您必须包含至少一个“.”字符。在最后一个“.”字符之后只能包含字母数字字符。

     正则表达式匹配：最多 128 个字符。

   - 路径：选择匹配模式类型并输入路径。

     值匹配：最多 128 个字符。区分大小写。允许的字符是 a-z、A-Z、0-9；以下特殊字符：_-.$/~"'@:+；&；以及通配符（* 和？）。

     正则表达式匹配：最多 128 个字符。

   - 查询字符串：输入键值对或不带键的值。

     最多 128 个字符。不区分大小写。允许的字符为 a-z、A-Z、0-9；以下特殊字符：_-.$/~"'@:+&()!,;=；以及通配符（* 和？）。

   - HTTP 请求方法：输入 HTTP 请求方法。

     最多 40 个字符。区分大小写。允许的字符为 A-Z，以及以下特殊字符：-_。不支持通配符。

   - HTTP 标头：选择匹配模式类型，然后输入标头名称和比较字符串。

     - HTTP 标头名称– 规则将评估包含此标头的请求以确认匹配值。

       值匹配：最多 40 个字符。不区分大小写。允许的字符是 a-z、A-Z、0-9 和以下特殊字符：*?-!#$%&'+.^_`|~。不支持通配符。

       正则表达式匹配：最多 128 个字符。

     - HTTP 标头值 – 输入要与 HTTP 标头值进行比较的字符串。

       值匹配：最多 128 个字符。不区分大小写。允许的字符为 a-z、A-Z、0-9；空格；以下特殊字符：!"#$%&'()+,./:;<=>@[]^_`{|}~-；以及通配符（* 和？）。

正则表达式匹配：最多 128 个字符。

- 源 IP – 以 CIDR 格式定义源 IP 地址。两 IPv4 者 IPv6 CIDRs 都允许。不支持通配符。

8. （可选）要添加转换，请选择添加转换，选择转换类型，然后输入要匹配的正则表达式和替换字符串。
9. （可选，仅限 HTTPS 侦听器）对于预路由操作，请选择以下操作之一：
   - 对用户进行身份验证-选择身份提供者并提供所需的信息。有关更多信息，请参阅 使用 Application Load Balancer 验证用户身份。
   - 验证令牌-输入 JWKS 端点、问题和任何其他声明。有关更多信息，请参阅 使用 Application Load Balancer 验证 JWT。
10. 对于“路由操作”，请选择以下操作之一：
    - 转发到目标组：选择一个目标组。要添加其他目标组，请选择添加目标组，然后选择一个目标组，检查相对权重并根据需要更新权重。如果在任何目标组上启用了粘性，则必须启用组级粘性。
    - 重定向到 URL：在 URI 部分选项卡上分别输入每个部分，或者在完整 URL选项卡上输入完整的地址，从而输入 URL。对于状态代码，根据您的需求选择临时（HTTP 302）或永久（HTTP 301）。
    - 返回固定响应：输入要为已删除的客户端请求返回的响应代码。您也可以指定内容类型和响应正文。
11. 选择下一步。
12. 对于优先级，输入一个介于 1 至 50000 之间的值。规则是按优先级顺序（从最低值到最高值）评估的。
13. 选择下一步。
14. 在审核和创建页面，选择创建。

Amazon CLI

添加一项规则

使用 create-rule 命令。

以下示例创建了一条具有 `forward` 操作和 `host-header` 条件的规则。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
```

```
    --priority 10 \
    --conditions "Field=host-header,Values=example.com,www.example.com" \
    --actions "Type=forward,TargetGroupArn=target-group-arn"
```

要创建转发操作以在两个目标组之间分配流量，请改用以下 --actions 选项。

```
    --actions '[{
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {"TargetGroupArn":"target-group-1-arn","Weight":50},
            {"TargetGroupArn":"target-group-2-arn","Weight":50}
          ]
        }
    }]'
```

以下示例创建了一条具有 fixed-response 操作和 source-ip 条件的规则。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
    --priority 20 \
    --conditions '[{"Field":"source-ip","SourceIpConfig":{"Values":
["192.168.1.0/24","10.0.0.0/16"]}}]' \
    --actions "Type=fixed-
response,FixedResponseConfig={StatusCode=403,ContentType=text/
plain,MessageBody='Access denied'}"
```

以下示例创建了一条具有 redirect 操作和 http-header 条件的规则。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
    --priority 30 \
    --conditions '[{"Field":"http-header","HttpHeaderConfig":
{"HttpHeaderName":"User-Agent","Values":["*Mobile*","*Android*","*iPhone*"]}}]' \
    --actions
 "Type=redirect,RedirectConfig={Host=m.example.com,StatusCode=HTTP_302}"
```

CloudFormation

添加一项规则

定义类型的资源AWS::ElasticLoadBalancingV2::ListenerRule。

以下示例创建了一条具有 forward 操作和 host-header 条件的规则。满足条件时，该规则会将流量发送到指定的目标组。

```
Resources:
    myForwardListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 10
       Conditions:
         - Field: host-header
           Values:
             - example.com
             - www.example.com
       Actions:
         - Type: forward
           TargetGroupArn: !Ref myTargetGroup
```

或者，要创建在满足条件时在两个目标组之间分配流量的转发操作，请按以下方式定义 Actions。

```
       Actions:
         - Type: forward
           ForwardConfig:
             TargetGroups:
               - TargetGroupArn: !Ref TargetGroup1
                 Weight: 50
               - TargetGroupArn: !Ref TargetGroup2
                 Weight: 50
```

以下示例创建了一条具有 fixed-response 操作和 source-ip 条件的规则。

```
Resources:
    myFixedResponseListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 20
       Conditions:
         - Field: source-ip
           SourceIpConfig:
             Values:
                - 192.168.1.0/24
```

```
                - 10.0.0.0/16
      Actions:
        - Type: fixed-response
          FixedResponseConfig:
            StatusCode: 403
            ContentType: text/plain
            MessageBody: "Access denied"
```

以下示例创建了一条具有 redirect 操作和 http-header 条件的规则。

```
Resources:
    myRedirectListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 30
       Conditions:
         - Field: http-header
           HttpHeaderConfig:
             HttpHeaderName: User-Agent
             Values:
               - "*Mobile*"
               - "*Android*"
               - "*iPhone*"
       Actions:
         - Type: redirect
           RedirectConfig:
             Host: m.example.com
             StatusCode: HTTP_302
```

编辑应用程序负载均衡器的侦听器规则

您可随时编辑侦听器规则的操作和条件。规则更新不会立即生效，因此在更新规则后的一小段短时间内，可以使用之前的规则配置来路由请求。任何正在进行的请求均会完成。

任务

- 修改默认操作
- 更新规则优先级
- 更新操作、条件和转换
- 管理规则标签

修改默认操作

默认操作会分配给名为 Default 的规则。您可以保留当前的规则类型并更改所需的信息，也可以更改规则类型并提供新的所需信息。

Console

修改默认操作

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择协议：端口列中的文本以打开侦听器的详细信息页面。
5. 在规则选项卡的侦听器规则部分中，选择默认规则。依次选择操作、编辑规则。
6. 在默认操作下，根据需要更新操作。

Amazon CLI

修改默认操作

使用 modify-listener 命令。以下示例更新了 `forward` 操作的目标组。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --default-actions Type=forward,TargetGroupArn=new-target-group-arn
```

以下示例更新了默认操作，以在两个目标组之间平均分配流量。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --default-actions '[{
      "Type":"forward",
      "ForwardConfig":{
        "TargetGroups":[
          {"TargetGroupArn":"target-group-1-arn","Weight":50},
          {"TargetGroupArn":"target-group-2-arn","Weight":50}
        ]
      }
    }]'
```

CloudFormation

修改默认操作

更新AWS::ElasticLoadBalancingV2::Listener资源。

```
Resources:
    myHTTPlistener:
      Type: 'AWS::ElasticLoadBalancingV2::Listener'
      Properties:
        LoadBalancerArn: !Ref myLoadBalancer
        Protocol: HTTP
        Port: 80
        DefaultActions:
          - Type: "forward"
            TargetGroupArn: !Ref myNewTargetGroup
```

更新规则优先级

规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。您可以随时更改非默认规则的优先级。不能更改默认规则的优先级。

Console

更新规则优先级

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择协议：端口列中的文本以打开侦听器的详细信息页面。
5. 在规则选项卡上选择该侦听器规则，然后选择操作、重新确定规则的优先级。
6. 在侦听器规则部分中，优先级列会显示当前的规则优先级。要更新规则优先级，请输入一个介于 1 至 50000 之间的值。
7. 选择保存更改。

Amazon CLI

更新规则优先级

使用 set-rule-priorities 命令。

```
aws elbv2 set-rule-priorities \
    --rule-priorities "RuleArn=listener-rule-arn,Priority=5"
```

CloudFormation

更新规则优先级

更新AWS::ElasticLoadBalancingV2::ListenerRule资源。

```
Resources:
    myListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 5
       Conditions:
         - Field: host-header
           Values:
             - example.com
             - www.example.com
       Actions:
         - Type: forward
           TargetGroupArn: !Ref myTargetGroup
```

更新操作、条件和转换

您可以更新规则的操作、条件和转换。

Console

更新规则操作、条件和转换

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择协议：端口列中的文本以打开侦听器的详细信息页面。
5. 在规则选项卡上，选择侦听器规则，然后选择操作、编辑规则。
6. 根据需要更新操作、条件和转换。有关详细步骤，请参阅添加规则。

7. 选择下一步。
8. （可选）更新优先级。
9. 选择下一步。
10. 选择保存更改。

Amazon CLI

更新规则操作、条件和转换

使用 modify-rule 命令。至少包括以下选项之一：`--actions`、`--conditions` 和 `--transforms`。

有关这些选项的示例，请参阅添加规则。

CloudFormation

更新规则操作、条件和转换

更新AWS::ElasticLoadBalancingV2::ListenerRule资源。

有关规则示例，请参阅 添加规则。

管理规则标签

标签可帮助您以不同的方式对侦听器和规则进行分类。例如，您可以按用途、所有者或环境为资源添加标签。每条规则的标签键必须无重复。如果您添加的标签中的键已经与规则关联，它将更新该标签的值。

用完标签后可以将其删除。

Console

管理规则的标签

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器的名称以打开其详细信息页面。
4. 在侦听器和规则选项卡上，选择协议：端口列中的文本以打开侦听器的详细信息页面。
5. 在规则选项卡上，选择名称标签列的文本以打开该规则的详细信息页面。

6. 在规则详细信息页面上，选择管理标签。
7. 在管理标签页面上，执行以下一项或多项操作：
   a. 要添加标签，请选择添加新标签，然后为键和值输入值。
   b. 要删除标签，请选择标签旁边的删除。
   c. 要更新标签，请为键或值输入新值。
8. 选择保存更改。

Amazon CLI

将标签添加到规则

使用 add-tags 命令。

```
aws elbv2 add-tags \
    --resource-arns listener-rule-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

移除规则的标签

使用 remove-tags 命令。

```
aws elbv2 remove-tags \
    --resource-arns listener-rule-arn \
    --tag-keys project department
```

CloudFormation

将标签添加到规则

更新AWS::ElasticLoadBalancingV2::ListenerRule资源。

```
Resources:
    myListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 10
       Conditions:
```

```
        - Field: host-header
          Values:
            - example.com
            - www.example.com
      Actions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
      Tags:
       - Key: 'project'
         Value: 'lima'
       - Key: 'department'
         Value: 'digital-media'
```

删除应用程序负载均衡器的侦听器规则

您可以随时删除侦听器的非默认规则。不能删除侦听器的默认规则。当您删除侦听器时，也会删除它的所有规则。

Console

删除规则

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选择协议：端口列中的文本以打开侦听器的详细信息页面。
5. 选择该规则。
6. 然后依次选择 Actions（操作）、Delete rule（删除规则）。
7. 提示进行确认时，输入 **confirm**，然后选择删除。

Amazon CLI

删除规则

使用 delete-rule 命令。

```
aws elbv2 delete-rule \
    --rule-arn listener-rule-arn
```

在应用程序负载均衡器中使用 TLS 进行双向身份验证

双向 TLS 身份验证是传输层安全性协议（TLS）的一种变体。传统 TLS 在服务器和客户端之间建立安全通信，其中服务器需要向其客户端提供其身份。借助双向 TLS，负载均衡器在协商 TLS 的同时协商客户端和服务器之间的双向身份验证。将双向 TLS 与应用程序负载均衡器结合使用时，可以简化身份验证管理并降低应用程序的负载。

通过使用双向 TLS，您的负载均衡器可以管理客户端身份验证，从而帮助确保只有受信任的客户端才能与您的后端应用程序通信。使用此功能时，负载均衡器使用来自第三方证书颁发机构 (CA) 的证书或使用 (PCA) Amazon 私有证书颁发机构（可选）对客户端进行身份验证，并进行吊销检查。负载均衡器会使用 HTTP 标头将客户端证书信息传递到后端，您的应用程序可以将该信息用于授权。

应用程序负载均衡器的双向 TLS 提供了以下用于验证 X.509v3 客户端证书的选项：

- 双向 TLS 传递：负载均衡器在不作验证的情况下将整个客户端证书链发送到目标。目标负责验证客户端证书链。然后通过使用客户端证书链，您可以在应用程序中实现负载均衡器身份验证和目标授权逻辑。
- 双向 TLS 验证：当负载均衡器协商 TLS 连接时，负载均衡器会为 X.509 客户端执行客户端证书身份验证。

要使用双向 TLS 传递模式，必须将侦听器配置为接受来自客户端的证书。要使用双向 TLS 进行验证，请参阅在应用程序负载均衡器上配置双向 TLS。

在应用程序负载均衡器上开始配置双向 TLS 之前

在应用程序负载均衡器上开始配置双向 TLS 之前，请注意以下事项：

配额

应用程序负载均衡器包括与您的 Amazon 账户中使用的信任存储库、CA 证书和证书吊销列表数量相关的某些限制。

有关更多信息，请参阅 Quotas for your Application Load Balancers。

证书要求

应用程序负载均衡器支持以下证书用于双向 TLS 身份验证：

- 支持的证书： X.509v3
- 支持的公钥：RSA 2K – 8K 或 ECDSA secp256r1、secp384r1、secp521r1

- 支持的签名算法：SHA256、384、512 with、384 RSA/SHA256、512 和 ,384 EC/SHA256 ,512 哈希值和 MGF1 RSASSA-PSS

CA 证书捆绑包

以下内容适用于证书颁发机构（CA）捆绑包：

- 应用程序负载均衡器批量上传每个证书颁发机构（CA）证书捆绑包。应用程序负载均衡器不支持上传单个证书。如果需要添加新证书，则必须上传证书捆绑包文件。
- 要替换 CA 证书包，请使用 ModifyTrustStoreAPI。

证书传递顺序

当您使用双向 TLS 传递时，应用程序负载均衡器会插入标头以将客户端证书链呈现给后端目标。呈现顺序从叶证书开始，以根证书结束。

会话恢复

对应用程序负载均衡器使用双向 TLS 传递或验证模式时，不支持会话恢复。

HTTP 标头

在使用双向 TLS 协商客户端连接时，应用程序负载均衡器使用 `X-Amzn-Mtls` 标头发送证书信息。有关更多信息和示例标头，请参阅 HTTP 标头和双向 TLS。

CA 证书文件

CA 证书文件必须满足以下要求：

- 证书文件必须使用 PEM（隐私增强邮件）格式。
- 证书内容必须包含在 `-----BEGIN CERTIFICATE-----` 和 `-----END CERTIFICATE-----` 边界内。
- 注释必须以 # 字符开头，并且不得包含任何 - 字符。
- 不能有任何空行。

不接受（无效）的证书示例：

```
# comments

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 01
    Signature Algorithm: ecdsa-with-SHA384
```

```
        Issuer: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
        Validity
            Not Before: Jan 11 23:57:57 2024 GMT
            Not After : Jan 10 00:57:57 2029 GMT
        Subject: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (384 bit)
                pub:
                    00:01:02:03:04:05:06:07:08
                ASN1 OID: secp384r1
                NIST CURVE: P-384
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                00:01:02:03:04:05:06:07:08
            X509v3 Subject Alternative Name:
                URI:EXAMPLE.COM
    Signature Algorithm: ecdsa-with-SHA384
         00:01:02:03:04:05:06:07:08
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

接受（有效）的证书示例：

1. 单一证书（PEM 编码）：

```
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

2. 多个证书（PEM 编码）：

```
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
# comments
-----BEGIN CERTIFICATE-----
```

```
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

HTTP 标头和双向 TLS

本节介绍在使用双向 TLS 与客户端协商连接时应用程序负载均衡器用于发送证书信息的 HTTP 标头。应用程序负载均衡器使用的特定 `X-Amzn-Mtls` 标头取决于您指定的双向 TLS 模式：传递模式或验证模式。

有关应用程序负载均衡器支持的其他 HTTP 标头的信息，请参阅 HTTP 标头和 Application Load Balancer。

传递模式的 HTTP 标头

对于传递模式下的双向 TLS，应用程序负载均衡器使用以下标头。

X-Amzn-Mtls-Clientcert

此标头包含连接中显示的整个客户端证书链的 URL-encoded PEM 格式，并带有+=/安全字符。

标头内容示例：

```
X-Amzn-Mtls-Clientcert: -----BEGIN%20CERTIFICATE-----%0AMIID<...reduced...>do0g
%3D%3D%0A-----END%20CERTIFICATE-----%0A-----BEGIN%20CERTIFICATE-----
%0AMIID1<...reduced...>3eZlyKA%3D%3D%0A-----END%20CERTIFICATE-----%0A
```

验证模式的 HTTP 标头

对于验证模式下的双向 TLS，应用程序负载均衡器使用以下标头。

X-Amzn-Mtls-Clientcert-Serial-Number

此标头包含叶证书序列号的十六进制表示形式。

标头内容示例：

```
X-Amzn-Mtls-Clientcert-Serial-Number: 03A5B1
```

X-Amzn-Mtls-Clientcert-Issuer

此标头包含颁发者可分辨名称（DN）的 RFC2253 字符串表示形式。

标头内容示例：

```
X-Amzn-Mtls-Clientcert-Issuer:
 CN=rootcamtls.com,OU=rootCA,O=mTLS,L=Seattle,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Subject

此标头包含主题可分辨名称（DN）的 RFC2253 字符串表示形式。

标头内容示例：

```
X-Amzn-Mtls-Clientcert-Subject: CN=client_.com,OU=client-3,O=mTLS,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Validity

此标头包含 ISO8601 格式的 `notBefore` 和 `notAfter` 日期。

标头内容示例：

```
X-Amzn-Mtls-Clientcert-Validity:
 NotBefore=2023-09-21T01:50:17Z;NotAfter=2024-09-20T01:50:17Z
```

X-Amzn-Mtls-Clientcert-Leaf

此标头包含树叶证书的 URL-encoded PEM 格式，并带有+=/安全字符。

标头内容示例：

```
X-Amzn-Mtls-Clientcert-Leaf: -----BEGIN%20CERTIFICATE-----%0AMIIG<...reduced...>NmrUlw
%0A-----END%20CERTIFICATE-----%0A
```

公开证书颁发机构（CA）主题名称

公开证书颁发机构（CA）主题名称有助于客户端确定在双向 TLS 身份验证期间将接受的证书，从而增强身份验证过程。

启用“公开 CA 主题名称”后，应用程序负载均衡器将根据所关联的信任存储区公开其信任的证书颁发机构（CA）主题名称列表。当客户端通过应用程序负载均衡器连接到目标时，该客户端会收到可信 CA 主题名称列表。

在 TLS 握手期间，当应用程序负载均衡器请求客户端证书时，将会在证书请求消息中包含可信 CA 区分名称（DN）列表。这有助于客户端选择与所公开 CA 主题名称相匹配的有效证书，从而简化身份验证过程并减少连接错误。

您可以在新侦听器和现有侦听器上启用“公开 CA 主题名称”。有关更多信息，请参阅 添加 HTTPS 侦听器。

应用程序负载均衡器的连接日志

Elastic Load Balancing 提供了连接日志，用于捕获有关发送到应用程序负载均衡器的请求的属性。连接日志包含客户端 IP 地址和端口、客户端证书信息、连接结果以及正在使用的 TLS 密码等信息。然后可以使用这些连接日志来查看请求模式和其他趋势。

要了解有关连接日志的更多信息，请参阅 应用程序负载均衡器的连接日志

在应用程序负载均衡器上配置双向 TLS

要使用双向 TLS 传递模式，只需要将侦听器配置为接受来自客户端的任何证书即可。当使用双向 TLS 传递时，应用程序负载均衡器会使用 HTTP 标头将整个客户端证书链发送到目标，这使您能够在应用程序中实现相应的身份验证和授权逻辑。有关更多信息，请参阅为您的应用程序负载均衡器创建 HTTPS 侦听器。

当您在验证模式下使用双向 TLS 时，当负载均衡器协商 TLS 连接时，Application Load Balancer 会为 X.509 客户端执行客户端证书身份验证。

要使用双向 TLS 验证模式，请执行以下操作：

- 创建新的信任存储资源。
- 上传您的证书颁发机构（CA）捆绑包和（可选）吊销列表。
- 将信任存储附加到配置为验证客户端证书的侦听器。

按照本节中的过程在应用程序负载均衡器上配置双向 TLS 验证模式。

任务

- 创建信任存储

- 关联信任存储
- 替换 CA 证书捆绑包
- 添加证书吊销列表
- 删除证书吊销列表
- 删除信任存储

创建信任存储

如果在创建负载均衡器或侦听器时添加了信任存储，该信任存储会自动关联到新侦听器。否则，您必须自行将其关联到侦听器。

先决条件

- 要创建信任存储，您必须拥有来自证书颁发机构（CA）的证书捆绑包。

Console

以下示例使用控制台的信任存储部分创建一个信任存储。您也可以在创建 HTTP 侦听器时创建信任存储库。

创建信任存储

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上，选择信任存储。
3. 选择创建信任存储。
4. 信任存储配置
   a. 对于信任存储，输入信任存储的名称。
   b. 对于证书颁发机构捆绑包，输入要使用的 CA 证书捆绑包的 Amazon S3 路径。
   c. （可选）使用对象版本选择 CA 证书捆绑包的早期版本。否则，将使用当前版本。
5. （可选）对于吊销，您可以选择将证书吊销列表添加到信任存储。
   a. 选择添加新 CRL，然后在 Amazon S3 中输入证书吊销列表的位置。
   b. （可选）使用对象版本选择证书吊销列表的先前版本。否则，将使用当前版本。
6. （可选）展开信任存储标签，可为您的信任存储输入最多 50 个标签。
7. 选择创建信任存储。

Amazon CLI

创建信任存储

使用 create-trust-store 命令。

```
aws elbv2 create-trust-store \
    --name my-trust-store \
    --ca-certificates-bundle-s3-bucket amzn-s3-demo-bucket \
    --ca-certificates-bundle-s3-key certificates/ca-bundle.pem
```

CloudFormation

创建信任存储

定义类型为 AWS::ElasticLoadBalancingV2:: TrustStore 的资源。

```
Resources:
  myTrustStore:
    Type: 'AWS::ElasticLoadBalancingV2::TrustStore'
    Properties:
      Name: my-trust-store
      CaCertificatesBundleS3Bucket: amzn-s3-demo-bucket
      CaCertificatesBundleS3Key: certificates/ca-bundle.pem
```

关联信任存储

创建信任存储后，您必须将其与侦听器关联，然后应用程序负载均衡器才能开始使用信任存储。每个安全侦听器只能关联一个信任存储，但一个信任存储可以关联到多个侦听器。

Console

您可以按以下过程中所示，将信任存储关联到现有侦听器。您也可以在创建 HTTPS 侦听器时关联信任存储。有关更多信息，请参阅创建 HTTPS 侦听器。

关联信任存储

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。

4. 在 “监听器和规则” 选项卡上，选择Protocol:Port列中的链接以打开安全侦听器的详细信息页面。
5. 在安全选项卡上，选择编辑安全侦听器设置。
6. 如果未启用双向 TLS，请在客户端证书处理下选择双向身份验证（mTLS），然后选择使用信任存储进行验证。
7. 对于信任存储，请选择该信任存储。
8. 选择保存更改。

Amazon CLI

关联信任存储

使用 modify-listener 命令。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --mutual-authentication "Mode=verify,TrustStoreArn=trust-store-arn"
```

CloudFormation

关联信任存储

更新 AWS::ElasticLoadBalancingV2:: Listener 资源。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
      Certificates:
        - CertificateArn: certificate-arn
      MutualAuthentication:
        - Mode: verify
```

```
        TrustStoreArn: trust-store-arn
```

替换 CA 证书捆绑包

CA 证书捆绑包是信任存储的一个必需组件。它是经过证书颁发机构验证的受信任的根证书和中间证书的集合。这些经过验证的证书确保客户端可以信任所呈现的证书由负载均衡器拥有。

一个信任存储一次只能包含一个 CA 证书捆绑包，但是您可以在创建信任存储后随时替换 CA 证书捆绑包。

Console

替换 CA 证书捆绑包

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上，选择信任存储。
3. 选择该信任存储。
4. 依次选择操作、替换 CA 捆绑包。
5. 在替换 CA 捆绑包页面的证书颁发机构捆绑包下，输入所需 CA 捆绑包的 Amazon S3 位置。
6. （可选）使用对象版本选择证书吊销列表的先前版本。否则，将使用当前版本。
7. 选择替换 CA 捆绑包。

Amazon CLI

替换 CA 证书捆绑包

使用 modify-trust-store 命令。

```
aws elbv2 modify-trust-store \
    --trust-store-arn trust-store-arn \
    --ca-certificates-bundle-s3-bucket amzn-s3-demo-bucket-new \
    --ca-certificates-bundle-s3-key certificates/new-ca-bundle-pem
```

CloudFormation

更新 CA 证书捆绑包

定义类型为 AWS::ElasticLoadBalancingV2:: TrustStore 的资源。

```
Resources:
  myTrustStore:
    Type: 'AWS::ElasticLoadBalancingV2::TrustStore'
    Properties:
      Name: my-trust-store
      CaCertificatesBundleS3Bucket: amzn-s3-demo-bucket-new
      CaCertificatesBundleS3Key: certificates/new-ca-bundle.pem
```

添加证书吊销列表

或者，您可以为信任存储创建证书吊销列表。吊销列表由证书颁发机构发布，包含已吊销证书的数据。应用程序负载均衡器仅支持 PEM 格式的证书吊销列表。

当将证书吊销列表添加到信任存储时，系统会为其提供吊销 ID。每添加一个吊销列表到信任存储，吊销 ID 都会增加，并且不能更改。

应用程序负载均衡器无法吊销证书吊销列表中具有负序列号的证书。

Console

添加吊销列表

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上，选择信任存储。
3. 选择信任存储以查看其详细信息页面。
4. 在证书吊销列表选项卡上，依次选择操作、添加吊销列表。
5. 在添加吊销列表页面上的证书吊销列表下，输入所需证书吊销列表的 Amazon S3 位置
6. （可选）使用对象版本选择证书吊销列表的先前版本。否则，将使用当前版本。
7. 选择添加吊销列表

Amazon CLI

添加吊销列表

使用 add-trust-store-revocations 命令。

```
aws elbv2 add-trust-store-revocations \
    --trust-store-arn trust-store-arn \
```

```
    --revocation-contents "S3Bucket=amzn-s3-demo-bucket,S3Key=crl/revoked-
list.crl,RevocationType=CRL"
```

CloudFormation

添加吊销列表

定义类型为 AWS::ElasticLoadBalancingV2:: TrustStoreRevocation 的资源。

```
Resources:
  myRevocationContents:
    Type: 'AWS:ElasticLoadBalancingV2::TrustStoreRevocation'
    Properties:
      TrustStoreArn: !Ref myTrustStore
      RevocationContents:
        - RevocationType: CRL
          S3Bucket: amzn-s3-demo-bucket
          S3Key: crl/revoked-list.crl
```

删除证书吊销列表

如果不再需要某个证书吊销列表，可以将其删除。删除信任存储中的某个证书吊销列表后，其吊销 ID 也奖被删除，并且在信任存储的有效期内不会重新使用。

Console

删除吊销列表

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上，选择信任存储。
3. 选择该信任存储。
4. 在证书吊销列表选项卡上，依次选择操作、删除吊销列表。
5. 当系统提示进行确认时，输入 **confirm**。
6. 选择删除。

Amazon CLI

删除吊销列表

使用 remove-trust-store-revocations 命令。

```
aws elbv2 remove-trust-store-revocations \
    --trust-store-arn trust-store-arn \
    --revocation-ids id-1 id-2 id-3
```

删除信任存储

当您不再使用某个信任存储时，可以将其删除。不能删除关联到侦听器的信任存储。

Console

删除信任存储

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上，选择信任存储。
3. 选择该信任存储。
4. 选择删除。
5. 提示进行确认时，输入 `confirm`，然后选择删除。

Amazon CLI

删除信任存储

使用 delete-trust-store 命令。

```
aws elbv2 delete-trust-store \
    --trust-store-arn trust-store-arn
```

共享应用程序负载均衡器的 Elastic Load Balancing 信任存储

Elastic Load Balancing 与 Amazon Resource Access Manager (Amazon RAM) 集成以实现信任存储共享。 Amazon RAM 是一项服务，可让您安全地在组织或组织单位 (OU) 之间 Amazon Web Services 账户 和内部共享您的 Elastic Load Balancing 信任存储资源。。如果您有多个账户，则可以创建一次信任存储，然后使用 Amazon RAM 使其可供其他账户使用。如果您的账户由管理 Amazon Organizations，则可以与组织中的所有账户共享信任存储，也可以仅与指定组织单位 (OU) 内的账户共享信任存储。

使用 Amazon RAM，您可以通过创建资源共享来共享您拥有的资源。资源共享指定要共享的资源以及与之共享资源的使用者。在此模型中 Amazon Web Services 账户 ，拥有信托商店的人（所有者）与其他 Amazon Web Services 账户 （消费者）共享。消费者可以将共享的信任存储关联到其应用程序负载均衡器侦听器，就像在自己的账户中关联信任存储一样。

信任存储所有者可以与以下对象共享信任存储：

- 具体在其组织 Amazon Web Services 账户 内部或外部 Amazon Organizations
- 其组织内部的组织单位 Amazon Organizations
- 它的整个组织都在 Amazon Organizations

内容

信任存储共享的先决条件

- 必须使用创建资源共享 Amazon Resource Access Manager。有关更多信息，请参阅《Amazon RAM 用户指南》中的 Create a resource share。
- 要共享信任商店，您必须在自己的商店中拥有该存储库 Amazon Web Services 账户。您不能共享已与您共享的信任存储。
- 要与您的组织或 Amazon Organizations中的组织单位共享信任存储，您必须启用与 Amazon Organizations的共享。有关更多信息，请参阅《Amazon RAM 用户指南》中的在 Amazon Organizations中启用资源共享。

共享信任存储的权限

信任存储所有者

- 信任存储所有者可以创建信任存储。
- 信任存储所有者可以在同一个账户中使用带有负载均衡器的信任存储。
- 信托商店所有者可以与其他 Amazon 帐户共享信任商店，或者 Amazon Organizations.

- 信任商店所有者可以取消与任何 Amazon 帐户的共享信任商店，或者 Amazon Organizations.
- 信任存储所有者不能阻止负载均衡器使用同一账户中的信任存储。
- 信任存储所有者可以列出使用共享信任存储的所有应用程序负载均衡器。
- 如果当前没有关联，则信任存储所有者可以删除信任存储。
- 信任存储所有者可以删除与共享的信任存储的关联。
- 使用共享信任存储时，信任存储所有者会收到 CloudTrail 日志。

信任存储消费者

- 信任存储消费者可以查看共享的信任存储。
- 信任存储消费者可以在同一个账户中使用信任存储创建或修改侦听器。
- 信任存储消费者可以使用共享的信任存储创建或修改侦听器。
- 信任存储消费者无法使用不再共享的信任存储来创建侦听器。
- 信任存储消费者无法修改共享的信任存储。
- 信任存储消费者在与侦听器关联时可以查看共享的信任存储 ARN。
- 信任存储使用者在使用共享信任存储创建或修改侦听器时会收到 CloudTrail 日志。

托管权限

共享信任存储时，资源共享使用托管权限来控制信任存储消费者允许的操作。您可以使用默认的托管权限 `AWSRAMPermissionElasticLoadBalancingTrustStore`（包括所有可用权限），也可以创建自己的客户托管权限。`DescribeTrustStores`、`DescribeTrustStoreRevocations` 和 `DescribeTrustStoreAssociations` 权限始终启用，不可删除。

信任存储资源共享支持下列权限：

弹性负载平衡：CreateListener

可以将共享的信任存储附加到新侦听器。

弹性负载平衡：ModifyListener

可以将共享的信任存储附加到现有侦听器。

弹性负载平衡：GetTrustStoreCaCertificatesBundle

可以下载与共享的信任存储关联的 CA 证书捆绑包。

弹性负载平衡：GetTrustStoreRevocationContent

可以下载与共享的信任存储关联的吊销文件。

弹性负载均衡：DescribeTrustStores（默认）

可以列出该账户拥有和共享的所有信任存储。

弹性负载均衡：DescribeTrustStoreRevocations（默认）

可以列出给定信任存储 ARN 的所有吊销内容。

弹性负载均衡：DescribeTrustStoreAssociations（默认）

可以列出信任存储消费者账户中与共享的信任存储关联的所有资源。

共享信任存储

要共享信任存储，您必须将它添加到资源共享。资源共享是一项 Amazon RAM 资源，可让您跨 Amazon Web Services 账户共享资源。资源共享指定了要共享的资源、共享资源的使用者，以及主体可以执行的操作。在使用 Amazon EC2 控制台共享信任存储时，必须将它添加到现有资源共享。要将信任存储添加到新的资源共享，您必须首先使用 Amazon RAM 控制台创建资源共享。

当您与其他人共享您拥有的信任存储时 Amazon Web Services 账户，您可以允许这些账户将其的 Application Load Balancer 侦听器与您账户中的信任存储区相关联。

如果您是组织中的一员， Amazon Organizations 并且启用了组织内部共享，则系统会自动授予组织中的消费者访问共享信任存储的权限。否则，消费者会收到加入资源共享的邀请，并在接受邀请后获得对共享的信任存储的访问权限。

您可以使用 Amazon EC2 控制台、 Amazon RAM 控制台或 Amazon CLI共享您拥有的信任存储。

使用 Amazon EC2 控制台共享您拥有的信任存储

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择信任存储。
3. 选择信任存储名称以查看其详细信息页面。
4. 在共享选项卡上，选择共享信任存储。
5. 在共享信任存储页面的资源共享下，选择您的信任存储将与哪些资源共享共享。
6. （可选）如果需要创建新的资源共享，请选择在 RAM 控制台中创建资源共享链接。

7. 选择共享信任存储。

要共享您拥有的信任存储，请使用 Amazon RAM 控制台

请参阅《Amazon RAM 用户指南》中的创建资源共享。

要共享您拥有的信任存储，请使用 Amazon CLI

使用 create-resource-share 命令。

停止共享信任存储

要停止共享您拥有的信任存储，必须从资源共享中将其删除。在您停止共享信任存储后，现有关联仍然存在，但是不允许与先前共享的信任存储建立新关联。当信任存储所有者或信任存储消费者删除关联时，该关联将从两个账户中删除。如果信任存储消费者希望保留资源共享，则必须要求资源共享的所有者删除该账户。

> **⚠ 删除关联**
>
> 信任商店所有者可以使用DeleteTrustStoreAssociation命令强制删除现有的信任存储关联。删除关联后，任何使用信任存储的负载均衡器侦听器都无法再验证客户端证书，并且 TLS 握手将失败。

您可以使用 Amazon EC2 控制台、 Amazon RAM 控制台或 Amazon CLI停止共享信任存储。

使用 Amazon EC2 控制台停止共享您拥有的信任存储

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择信任存储。
3. 选择信任存储名称以查看其详细信息页面。
4. 在共享选项卡的资源共享下，选择要停止共享的资源共享。
5. 选择移除 。

要停止共享您拥有的信任存储，请使用 Amazon RAM 控制台

请参阅《Amazon RAM 用户指南》中的更新资源共享。

要停止共享您拥有的信任存储，请使用 Amazon CLI

使用 disassociate-resource-share 命令。

计费和计量

共享的信任存储按照与应用程序负载均衡器的每个信任存储关联收取相同的标准信任存储费率，按小时计费。

有关更多信息（包括每个区域的具体费率），请参阅 Elastic Load Balancing 定价

使用 Application Load Balancer 验证用户身份

可以将 Application Load Balancer 配置为在用户访问应用程序时安全验证用户的身份。这使您可以将验证用户身份的工作交给负载均衡器完成，以便应用程序可以专注于其业务逻辑。

支持以下使用案例：

- 通过符合 OpenID Connect (OIDC) 条件的身份提供商 (IdP) 验证用户身份。
- 通过亚马逊 Cognito 支持的用户池通过社交 IdPs媒体（例如亚马逊、Facebook 或谷歌）对用户进行身份验证。
- 通过企业身份、使用 SAML、OpenID Connect (OIDC) 或 OAuth、通过 Amazon Cognito 支持的用户群体验证用户身份。

准备使用 OIDC-compliant IdP

如果您将 Id OIDC-compliant P 与 Application Load Balancer 配合使用，请执行以下操作：

- 使用 IdP 创建新的 OIDC 应用程序。IdP 的 DNS 必须是可公开解析的。
- 必须配置客户端 ID 和客户端密钥。
- 获取 IdP 发布的以下终端节点：授权终端节点、令牌终端节点和用户信息终端节点。可以在配置中找到此信息。
- IdP 端点证书应由可信的公共证书颁发机构颁发。
- 端点的 DNS 条目必须是可公开解析的，即使它们解析为私有 IP 地址也是如此。
- 允许在 IdP 应用程序中使用以下重定向 URL 之一（无论您的用户将使用哪种 IdP 应用程序），其中 DNS 是负载均衡器的域名，CNAME 是应用程序的 DNS 别名：

- https://*DNS*/oauth2/idpresponse
- https://*CNAME*/oauth2/idpresponse

准备使用 Amazon Cognito

可用区域

应用程序负载均衡器的 Amazon Cognito 集成现已在以下区域推出：

- 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）
- 加拿大西部（卡尔加里）
- 欧洲地区（斯德哥尔摩）
- 欧洲地区（米兰）
- 欧洲地区（法兰克福）
- 欧洲（苏黎世）
- 欧洲地区（爱尔兰）
- 欧洲地区（伦敦）
- Europe (Paris)
- 欧洲（西班牙）
- 南美洲（圣保罗）
- 亚太地区（香港）
- 亚太地区（东京）
- 亚太地区（首尔）
- 亚太地区（大阪）
- 亚太地区（孟买）
- 亚太地区（海得拉巴）

- 亚太地区（新加坡）
- 亚太地区（悉尼）
- 亚太地区（雅加达）
- 亚太地区（墨尔本）
- 中东（阿联酋）：
- 中东（巴林）
- 非洲（开普敦）
- 以色列（特拉维夫）

如果您将 Amazon Cognito 用户池与 Application Load Balancer 结合使用，请执行以下操作：

- 创建用户池。有关更多信息，请参阅 Amazon Cognito 开发人员指南中的 Amazon Cognito 用户池。
- 创建用户池客户端。必须将客户端配置为生成客户端密钥，使用代码授予流程并支持与负载均衡器所用相同的 OAuth 范围。有关更多信息，请参阅 Amazon Cognito 开发人员指南中的配置用户池应用程序客户端。
- 创建用户池域。有关更多信息，请参阅《Amazon Cognito 开发人员指南》中的 Configure a user pool domain。
- 验证请求的范围是否将返回 ID 令牌。例如，默认范围 `openid` 将返回 ID 令牌，但 `aws.cognito.signin.user.admin` 范围不返回 ID 令牌。
- 要与社交或企业 IdP 联合，请在联合身份验证部分中启用 IdP。有关更多信息，请参阅《Amazon Cognito 开发人员指南》中的 User pool sign-in with a third party identity provider。
- 允许在 Amazon Cognito 的回调 URL 字段中使用以下重定向 URL，其中 DNS 是负载均衡器的域名，CNAME 是应用程序的 DNS 别名（如果正在使用）：
  - https://*DNS*/oauth2/idpresponse
  - https://*CNAME*/oauth2/idpresponse
- 允许在 IdP 应用程序的回调 URL 中使用您的用户池域。使用 IdP 的格式。例如：
  - https://*domain-prefix*.auth。 *region*.amazoncognito。 com/saml2/idpresponse
  - https://*user-pool-domain*/saml2/idpresponse

应用程序客户端设置中的回调 URL 必须全都使用小写字母。

要使某用户能够将负载均衡器配置为使用 Amazon Cognito 验证用户身份，必须授予该用户调用 `cognito-idp:DescribeUserPoolClient` 操作的权限。

准备使用亚马逊 CloudFront

如果您在 Application Load Balancer 前面使用 CloudFront 分配，请启用以下设置：

- 转发请求标头（全部）-确保 CloudFront 不会缓存经过身份验证的请求的响应。这可避免在身份验证会话过期后从缓存提供响应。或者，为了在启用缓存时降低这种风险， CloudFront 分配的所有者可以将生存时间 (TTL) 值设置为在身份验证 Cookie 过期之前过期。
- 查询字符串转发和缓存（全部）– 确保负载均衡器能够访问使用 IdP 对用户进行身份验证所需的查询字符串参数。
- Cookie 转发（全部）-确保将所有身份验证 Cookie CloudFront 转发到负载均衡器。
- 在与亚马逊一起配置 OpenID Connect (OIDC) 身份验证时 CloudFront，请确保在整个连接路径中始终使用 HTTPS 端口 443。否则可能会因为客户端 OIDC 重定向 URL 与最初生成的 URI 端口号不匹配，从而导致身份验证失败。

配置用户身份验证

通过为一个或多个侦听器规则创建身份验证操作来配置用户身份验证。HTTPS 侦听器仅支持 `authenticate-cognito` 和 `authenticate-oidc` 操作类型。有关相应字段的描述，请参阅 Elastic Load Balancing API 参考版本 2015-12-01 AuthenticateOidcActionConfig中的AuthenticateCognitoActionConfig和。

负载均衡器会向客户端发送会话 Cookie 以保持身份验证状态。由于用户身份验证需要 HTTPS 侦听器，因此该 Cookie 始终包含 `secure` 属性。此 Cookie 包含 CORS（跨源资源共享）请求的 `SameSite=None` 属性。

对于支持多个需要独立客户端身份验证的应用程序的负载均衡器，具有身份验证操作的每个侦听器规则应具有唯一的 Cookie 名称 这可确保客户端在路由到规则中指定的目标组之前始终使用 IdP 进行身份验证。

Application Load Balancer 不支持 URL 编码的 Cookie 值。

默认情况下，`SessionTimeout` 字段设置为 7 日。如果需要更短的会话，可将会话超时配置为短至 1 秒。有关更多信息，请参阅 会话超时。

视应用程序的情况设置 `OnUnauthenticatedRequest` 字段。例如：

- 需要用户使用社交或企业身份登录的应用程序 – 这由默认选项 `authenticate` 支持。如果用户未登录，则负载均衡器会将请求重定向到 IdP 授权终端节点并且 IdP 将提示用户使用其用户界面登录。

- 为已登录用户提供个性化视图或为未登录用户提供常规视图的应用程序 – 要支持此类型的应用程序，请使用 `allow` 选项。如果用户已登录，则负载均衡器将提供用户索赔并且应用程序可以提供个性化视图。如果用户未登录，则负载均衡器将转发请求而不提供用户索赔并且应用程序可以提供常规视图。
- Single-page 每隔几秒钟加载一次的应用程序-如果您使用该deny选项，则负载均衡器会向没有身份验证信息的 AJAX 调用返回 HTTP 401 未经授权的错误。 JavaScript 但是，如果用户的身份验证信息已过期，它会将客户端重定向到 IdP 授权终端节点。

负载均衡器必须能够与 IdP 令牌终端节点 (`TokenEndpoint`) 和 IdP 用户信息终端节点 (`UserInfoEndpoint`) 通信。应用程序负载均衡器在与这些端点通信时仅支持 IPv4。如果您的 IdP 使用公有地址，请确保负载均衡器的安全组和 VPC 的网络 ACL 允许访问这些端点。使用内部负载均衡器或 IP 地址类型 `dualstack-without-public-ipv4` 时，NAT 网关可使负载均衡器与端点进行通信。有关更多信息，请参阅 Amazon VPC 用户指南中的 NAT 网关基础。

使用以下 create-rule 命令配置用户身份验证。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
    --priority 10 \
    --conditions Field=path-pattern,Values="/login" \
    --actions file://actions.json
```

以下是 `actions.json` 文件，该文件指定 `authenticate-oidc` 操作和 `forward` 操作。`AuthenticationRequestExtraParams` 允许您在身份验证期间将额外的参数传递给 IdP。请按照您的身份提供商提供的文档确定支持的字段

```
[{
    "Type": "authenticate-oidc",
    "AuthenticateOidcConfig": {
        "Issuer": "https://idp-issuer.com",
        "AuthorizationEndpoint": "https://authorization-endpoint.com",
        "TokenEndpoint": "https://token-endpoint.com",
        "UserInfoEndpoint": "https://user-info-endpoint.com",
        "ClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "ClientSecret": "12345678901234567890123456789O",
        "SessionCookieName": "my-cookie",
        "SessionTimeout": 3600,
        "Scope": "email",
        "AuthenticationRequestExtraParams": {
```

```
            "display": "page",
            "prompt": "login"
        },
        "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
},
{
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
}]
```

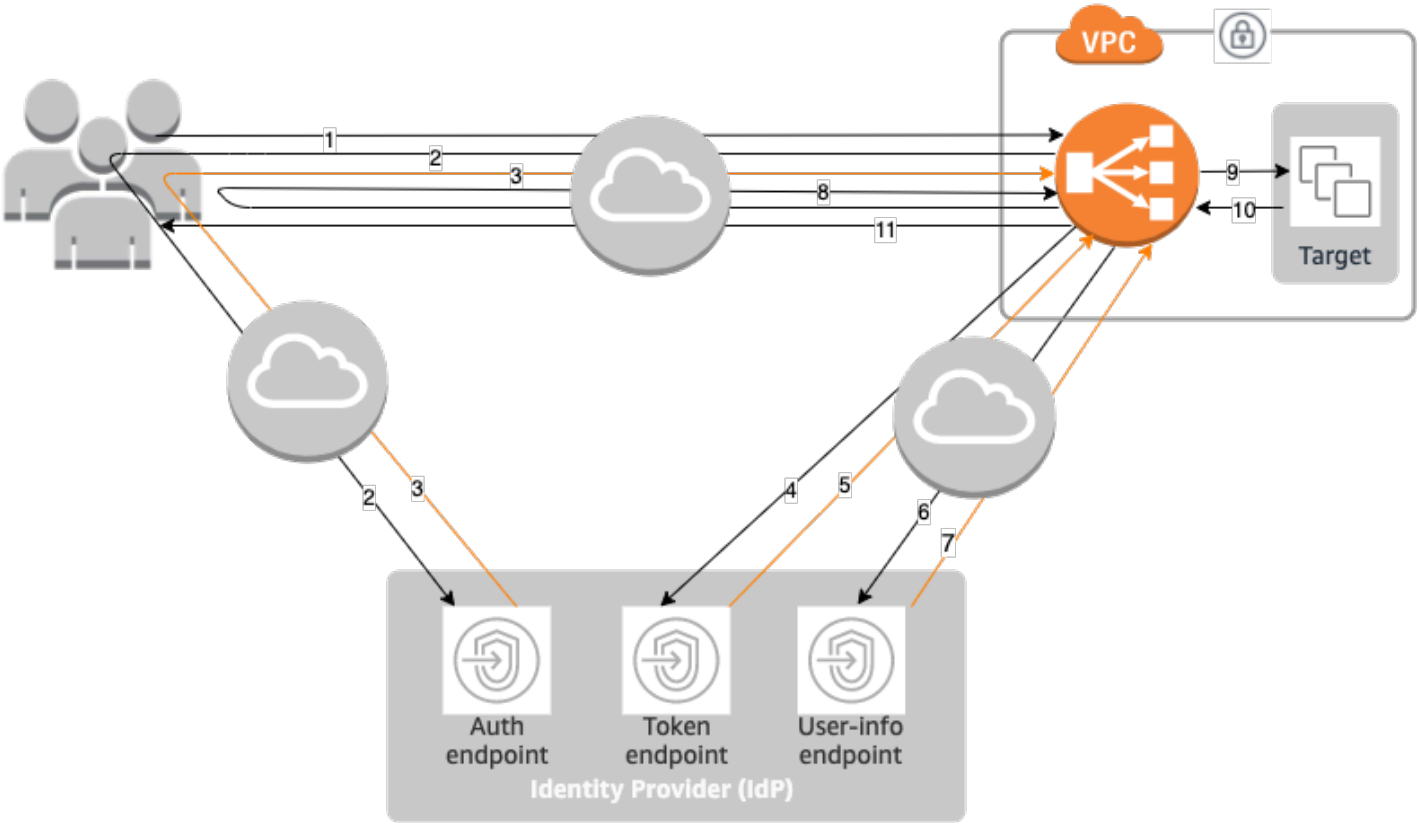
下面是指定 authenticate-cognito 操作和 forward 操作的 actions.json 文件的示例。

```
[{
    "Type": "authenticate-cognito",
    "AuthenticateCognitoConfig": {
        "UserPoolArn": "arn:aws:cognito-idp:region-code:account-id:userpool/user-pool-
id",
        "UserPoolClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "UserPoolDomain": "userPoolDomain1",
        "SessionCookieName": "my-cookie",
        "SessionTimeout": 3600,
        "Scope": "email",
        "AuthenticationRequestExtraParams": {
            "display": "page",
            "prompt": "login"
        },
        "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
},
{
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
}]
```

有关更多信息，请参阅 Application Load Balancer 的侦听器规则。

身份验证流程

下面的网络图是 Application Load Balancer 如何使用 OIDC 对用户进行身份验证的可视表示。



下面的编号项，突出显示并解释上一个网络图中显示的元素。

1. 用户向在 Application Load Balancer 后面托管的网站发送 HTTPS 请求。当满足具有身份验证操作的规则的条件时，负载均衡器将检查请求标头中的身份验证会话 Cookie。
2. 如果 Cookie 不存在，则负载均衡器会将用户重定向到 IdP 授权终端节点，以便 IdP 可对用户进行身份验证。
3. 验证用户身份之后，IdP 会使用授权代码将用户发回负载均衡器。
4. 负载均衡器会将此授权代码发送给 IdP 令牌终端节点。
5. 在收到有效的授权代码后，IdP 将向 Application Load Balancer 提供 ID 令牌和访问令牌。
6. 然后，Application Load Balancer 将访问令牌发送到用户信息终端节点。
7. 用户信息终端节点交换用户声明的访问令牌。
8. Application Load Balancer 将具有 `AWSELB` 身份验证会话 Cookie 的用户重定向到原始 URI。由于大多数浏览器将 Cookie 限制为 4K 大小，因此负载均衡器会将超出 4K 大小的 Cookie 分片为多个

Cookie。如果从 IdP 接收的用户声明和访问令牌的总大小超过 11K 字节，则负载均衡器会向客户端返回 HTTP 500 错误并递增 `ELBAuthUserClaimsSizeExceeded` 指标。

9. Application Load Balancer 验证 cookie 并将用户信息转发到 `X-AMZN-OIDC-*` HTTP 标头设置中的目标。有关更多信息，请参阅 用户申请编码和签名验证。
10. 目标向应用 Application Load Balancer 发回响应。
11. Application Load Balancer 向用户发送最终响应。

每个新请求都经历步骤 1 到 11，而后续请求则经过步骤 9 到 11。也就是说，只要 cookie 尚未过期，每个后续请求都从步骤 9 开始。

用户在 IdP 进行身份验证后，会在请求标头中添加 `AWSALBAuthNonce` cookie。这不会改变应用程序负载均衡器处理来自 IdP 的重定向请求的方式。

如果 IdP 在 ID 令牌中提供了有效的刷新令牌，则负载均衡器将保存刷新令牌并在访问令牌过期时使用刷新令牌刷新用户索赔，直至会话超时或 IdP 刷新失败。如果用户注销，刷新将失败并且负载均衡器会将用户重定向到 IdP 授权终端节点。这使负载均衡器能够在用户注销后删除会话。有关更多信息，请参阅 会话超时。

> **Note**
>
> Cookie 过期与身份验证会话到期不同。Cookie 有效期是 Cookie 的一个属性，设置为 7 天。身份验证会话的实际长度由 Application Load Balancer 上为身份验证功能配置的会话超时确定。此会话超时包含在身份验证 cookie 值中，该值也经过加密。

用户申请编码和签名验证

在负载均衡器成功验证用户身份之后，它会将从 IdP 收到的用户索赔发送给目标。负载均衡器先为用户索赔签名，以便应用程序可以验证该签名并验证索赔是负载均衡器发送的。

负载均衡器添加以下 HTTP 标头：

`x-amzn-oidc-accesstoken`

令牌终端节点中的访问令牌（明文格式）。

`x-amzn-oidc-identity`

用户信息终端节点中的主题字段 (sub)（明文格式）。

注意：此子声明是识别给定用户的最佳方法。

`x-amzn-oidc-data`

用户声明（JSON Web 令牌 (JWT) 格式）。

访问令牌和用户声明与 ID 令牌不同。访问令牌和用户声明仅允许访问服务器资源，而 ID 令牌带有的额外信息以对用户进行身份验证。应用程序负载均衡器在对用户进行身份验证时会创建一个新的访问令牌，并且仅将访问令牌和声明传递给后端，但不会传递 ID 令牌信息。

这些令牌遵循 JWT 格式，但不是 ID 令牌。JWT 格式包含 base64 URL 编码的标头、有效负载和签名，并在末尾包含填充字符。Application Load Balancer 使用 ES256（ECDSA 使用 P-256 和 SHA256）生成 JWT 签名。

JWT 标头为具有以下字段的 JSON 对象：

```
{
   "alg": "algorithm",
   "kid": "12345678-1234-1234-1234-123456789012",
   "signer": "arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/
app/load-balancer-name/load-balancer-id",
   "iss": "url",
   "client": "client-id",
   "exp": "expiration"
}
```

JWT 负载是一个 JSON 对象，该对象包含从 IdP 用户信息终端节点接收的用户索赔。

```
{
   "sub": "1234567890",
   "name": "name",
   "email": "alias@example.com",
   ...
}
```

如果希望负载均衡器加密您的用户声明，则必须将目标组配置为使用 HTTPS。此外，作为一项安全最佳实践，我们建议您将目标限制为仅接收来自应用程序负载均衡器的流量。您可以通过将目标的安全组配置为引用负载均衡器的安全组 ID 来实现此目的。

为确保安全，您必须在根据声明进行任何授权之前验证签名，并验证 JWT 标头中的 `signer` 字段是否包含预期的应用程序负载均衡器 ARN。

要获取公钥，请从 JWT 标头中获取密钥 ID 并使用它从终端节点查找公钥：

对于中国（北京），端点如下所示：

```
https://aws-elb-public-keys-prod-cn-north-1.s3.cn-north-1.amazonaws.com.cn/
```

Amazon 提供了一个库，您可以使用该库来验证由 Amazon Cognito、应用程序负载均衡器和其他 IdP 签署的 JWT。 OIDC-compatible 有关更多信息，请参阅 Amazon JWT Verify。

Timeout

会话超时

刷新令牌和会话超时将一起运行，如下所示：

- 如果会话超时短于访问令牌过期时间，则负载均衡器将遵守会话超时。如果用户与 IdP 之间有活动的会话，则可能不会提示用户重新登录。否则，会将用户重定向到登录页面。
  - 如果 IdP 会话超时长于 Application Load Balancer 会话超时，则用户无需提供凭证即可重新登录。相反，IdP 会使用新的授权代码重定向回 Application Load Balancer。授权码是一次性使用的，即使没有进行重新登录亦是如此。
  - 如果 IdP 会话超时等于或短于 Application Load Balancer 会话超时，则用户必须提供凭证才能重新登录。用户登录后，IdP 会使用新的授权代码重定向回 Application Load Balancer，然后身份验证流程的其余部分将继续进行，直到请求到达后端。
- 如果会话超时长于访问令牌过期时间并且 IdP 不支持刷新令牌，则负载均衡器会将身份验证会话一直保留到其超时，之后让用户再次登录。然后，它让用户再次登录。
- 如果会话超时长于访问令牌过期时间并且 IdP 支持刷新令牌，则负载均衡器将在每次访问令牌到期时刷新用户会话。仅当身份验证会话超时或刷新流程失败之后，负载均衡器才会让用户再次登录。

客户端登录超时

客户端必须在 15 分钟内启动并完成身份验证过程。如果客户端未能在 15 分钟限制内完成身份验证，则会收到来自负载均衡器的 HTTP 401 错误。无法更改或删除此超时。

例如，如果用户通过 Application Load Balancer 加载登录页面，则必须在 15 分钟内完成登录过程。如果用户等待并在 15 分钟超时过期后尝试登录，则负载均衡器将会返回 HTTP 401 错误。用户必须刷新页面，然后再次尝试登录。

身份验证注销

当应用程序需要注销经身份验证的用户时，应将身份验证会话 Cookie 的到期时间设置为 -1 并将客户端重定向到 IdP 注销终端节点（如果 IdP 支持一个终端节点）。为防止用户重复使用已删除的 Cookie，建议为访问令牌配置合理的短过期时间。如果客户端为负载均衡器提供了具有已过期访问令牌和非 NULL 刷新令牌的会话 Cookie，负载均衡器将联系 IdP 来确定用户是否仍处于登录状态。

客户端注销登录页不进行身份验证。这意味着此类页面不能位于需要身份验证的应用程序负载均衡器规则之后。

- 当向目标发送请求时，应用程序必须将所有身份验证 cookie 的到期时间设置为 -1。Application Load Balancer 支持最大 16K 的 Cookie，因此最多可以创建 4 个分片发送给客户端。
  - 如果 IdP 具有注销终端节点，它应该发出重定向到 IdP 注销终端节点，例如 Amazon Cognito 开发人员指南中记录的注销终端节点。
  - 如果 IdP 没有注销终端节点，请求将返回到客户端注销登录页面，并重新启动登录过程。
- 假设 IdP 具有注销终端节点，IdP 必须使访问令牌和刷新令牌过期，并将用户重定向回客户端注销登录页。
- 后续请求遵循原始身份验证流程。

使用 Application Load Balancer 验证 JWT

您可以配置 Application Load Balancer (ALB) 来验证客户端为安全服务到服务 (S2S) 或机器对机器 (M2M) 通信提供的 JSON Web 令牌 (JWT)。无论JWT是如何发布的，负载均衡器都可以在无需人为干预的情况下对其进行验证。

ALB 将验证令牌签名，并需要两个强制声明：“iss”（发行者）和 “exp”（到期）。此外，如果代币中存在，ALB 还将验证 “nbf”（不是之前）和 “iat”（当时发行）的索赔。您最多可以配置 10 个额外的索赔进行验证。这些索赔支持三种格式：

- Single-string: 单个文本值
- Space-separated 值：用空格分隔的多个值（最多 10 个值）
- String-array: 文本值数组（最多 10 个值）

如果令牌有效，则负载均衡器会将带有令牌的请求按原样转发到目标。否则，服务将拒绝该请求。

准备使用 JWT 验证

完成以下任务：

1. 向 IdP 注册您的服务，IdP 会发布客户端 ID 和客户机密钥。
2. 单独拨打 IdP 以请求访问服务。IdP 使用访问令牌进行响应。此令牌通常是由 IdP 签署的 JWT。
3. 设置 JSON 网络密钥集 (JWKS) 端点。负载均衡器在您配置的众所周知的位置获取 IdP 发布的公钥。
4. 在请求标头中包含 JWT，并在每个请求中将其转发给 Application Load Balancer。注意：仅支持 RS256 算法

JWT 验证限制

在 Application Load Balancer 中使用 JWT 验证时，JWKS（JSON Web 密钥集）端点必须满足以下要求：

- 最大响应大小：150 KB
- 最大按键数：10 把钥匙

如果您的身份提供商发出的 JWKS 响应超过上述任一限制，Application Load Balancer 将不会将请求转发到您的后端目标。

如果您的身份提供商的 JWKS 端点超过了这些限制，请考虑在您的应用程序代码中实现 JWT 验证，或者使用密钥集较小的身份提供商。

使用控制台配置 JWT 验证

1. 打开 Amazon EC2 控制台控制台，网址为https://console.aws.amazon.com/ec2/。
2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择您的 Application Load Balancer，然后选择 Li steners 选项卡。
4. 选择 HTTPS 监听器，然后选择管理规则。
5. 选择添加规则。
6. （可选）要为规则指定名称，请展开名称和标签，然后输入名称。要添加其他标签，请选择添加其他标签，然后输入标签键和标签值。
7. 在条件下，定义 1-5 个条件值

8. （可选）要添加转换，请选择添加转换，选择转换类型，然后输入要匹配的正则表达式和替换字符串。
9. 对于 “操作 Pre-routing”、“操作”，选择 “验证令牌”。
   a. 对于 JWKS 端点，请输入您的 JSON Web 密钥集端点的网址。此端点必须可公开访问，并返回用于验证 JWT 签名的公钥。
   b. 对于发行人，在您的 JWT 代币中输入 iss 索赔的预期值。
   c. （可选）要验证其他索赔，请选择其他索赔。
      i. 在索赔名称中，输入要验证的索赔名称。
      ii. 在 “格式” 中，选择应如何解释索赔值：
         1. 单字符串：声明必须与一个指定值完全匹配。
         2. 字符串数组：声明必须与数组中的一个值匹配。
         3. 空格分隔值：声明包含以空格分隔的值，这些值必须包含指定的值。
      iii. 在 “值” 中，输入索赔的预期值。
      iv. 对于其他索赔（最多 10 项索赔），请重复此操作。
10. 在 “操作”、“路由操作” 中，选择应在成功验证令牌后执行的主要操作（“转发至”、“重定向到” 或 “返回固定响应”）。
11. 根据需要配置主要操作
12. 选择保存。

使用 CLI 配置 JWT 验证

使用以下 create-rule 命令配置 JWT 验证。

创建带有验证 JWT 的操作的侦听器规则。监听器必须是 HTTPS 监听器。

> **Note**
>
> 配置 JWT 验证时，请确保您的 JWKS 端点响应大小不超过 150 KB 或包含的密钥不超过 10 个。超过这些限制的响应将阻止请求转发到您的目标。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
```

```
    --priority 10 \
    --conditions Field=path-pattern,Values="/login" \
    --actions file://actions.json
```

以下是指定jwt-validation动作和动作的actions.json文件示例。forward请按照您的身份提供商提供的文档确定支持的字段

```
--actions '[
    {
        "Type":"jwt-validation",
        "JwtValidationConfig":{
            "JwksEndpoint":"https://issuer.example.com/.well-known/jwks.json",
            "Issuer":"https://issuer.com"
        },
        "Order":1
    },
    {
        "Type":"forward",
        "TargetGroupArn":"target-group-arn",
        "Order":2
    }
]'
```

以下示例指定了要验证的额外声明。

```
--actions '[
    {
        "Type":"jwt-validation",
        "JwtValidationConfig":{
            "JwksEndpoint":"https://issuer.example.com/.well-known/jwks.json",
            "Issuer":"https://issuer.com",
            "AdditionalClaims":[
              {
                  "Format":"string-array",
                  "Name":"claim_name",
                  "Values":["value1","value2"]
              }
            ],
        },
        "Order":1
    },
    {
        "Type":"forward",
```

```
        "TargetGroupArn":"target-group-arn",
        "Order":2
    }
]'
```

有关更多信息，请参阅 the section called “侦听器规则”。

HTTP 标头和 Application Load Balancer

HTTP 请求和 HTTP 响应使用标头字段发送有关 HTTP 消息的信息。HTTP 标头会自动添加。标头字段为冒号分隔的名称值对，各个值对之间由回车符 (CR) 和换行符 (LF) 进行分隔。RFC 2616 信息标头中定义了标准 HTTP 标头字段集。此外还有应用程序广泛使用和自动添加的非标准 HTTP 标头。某些非标准 HTTP 标头具有 `X-Forwarded` 前缀。Application Load Balancer 支持以下 `X-Forwarded` 标头。

有关 HTTP 连接的更多信息，请参阅 Elastic Load Balancing 用户指南中的请求路由。

X-Forwarded 标题

- X-Forwarded-For
- X-Forwarded-Proto
- X-Forwarded-Port

X-Forwarded-For

在您使用 HTTP 或 HTTPS 负载均衡器时，`X-Forwarded-For` 请求标头可帮助您识别客户端的 IP 地址。由于负载均衡器会拦截客户端和服务器之间的流量，因此您的服务器访问日志中将仅包含负载均衡器的 IP 地址。要查看客户端的 IP 地址，请使用 `routing.http.xff_header_processing.mode` 属性。借助此属性，您可以在应用程序负载均衡器将请求发送到目标之前修改、保留或移除 HTTP 请求中的 `X-Forwarded-For` 标头。此属性的可能值为 `append`、`preserve` 和 `remove`。此属性的默认值为 `append`。

> **Important**
>
> 由于存在安全风险，应谨慎使用 `X-Forwarded-For` 标头。只有由网络内得到妥善保护的系统添加的条目才被视为可信。

处理模式

- Append
- Preserve
- 删除

Append

默认情况下，应用程序负载均衡器会在 X-Forwarded-For 请求标头中存储客户端的 IP 地址，并将该标头传递到您的服务器。如果 X-Forwarded-For 请求标头未包含在原始请求中，则负载均衡器会创建一个以客户端 IP 地址作为请求值的标头。否则，负载均衡器会将客户端 IP 地址附加到现有标头，然后将该标头传递到您的服务器。X-Forwarded-For 请求标头可能包含多个以逗号分隔的 IP 地址。

X-Forwarded-For 请求标头采用以下形式：

```
X-Forwarded-For: client-ip-address
```

下面是 IP 地址为 203.0.113.7 的客户端的 X-Forwarded-For 请求标头的示例。

```
X-Forwarded-For: 203.0.113.7
```

下面是 IPv6 地址为 X-Forwarded-For 的客户端的 2001:DB8::21f:5bff:febf:ce22:8a2e 请求标头的示例。

```
X-Forwarded-For: 2001:DB8::21f:5bff:febf:ce22:8a2e
```

在负载均衡器上启用客户端端口保留属性（routing.http.xff_client_port.enabled）后，X-Forwarded-For 请求标头包括附加到 client-ip-address 的 client-port-number（以冒号分隔）。然后标头采用以下形式：

```
IPv4 -- X-Forwarded-For: client-ip-address:client-port-number
```

```
IPv6 -- X-Forwarded-For: [client-ip-address]:client-port-number
```

对于 IPv6，请注意，当负载均衡器将 client-ip-address 附加到现有标头时，会将地址放在方括号内。

下面是 IPv4 地址为 12.34.56.78、端口号为 8080 的客户端的 X-Forwarded-For 请求标头示例。

```
X-Forwarded-For: 12.34.56.78:8080
```

下面是 IPv6 地址为 2001:db8:85a3:8d3:1319:8a2e:370:7348、端口号为 8080 的客户端的 X-Forwarded-For 请求标头示例。

```
X-Forwarded-For: [2001:db8:85a3:8d3:1319:8a2e:370:7348]:8080
```

Preserve

属性中的 preserve 模式会确保在将 HTTP 请求发送到目标之前不会以任何方式进行修改其中的 X-Forwarded-For 标头。

删除

属性中的 remove 模式会在将 HTTP 请求发送到目标之前移除其中的 X-Forwarded-For 标头。

如果您启用了客户端端口保留属性（routing.http.xff_client_port.enabled），同时还为 routing.http.xff_header_processing.mode 属性选择了 preserve 或 remove，则应用程序负载均衡器将覆盖客户端端口保留属性。它会将 X-Forwarded-For 标头保留不变，或者根据您选择的模式将其移除，然后再将请求发送到目标。

当您选择 append、preserve 或者 remove 模式时目标将收到的 X-Forwarded-For 标头示例见下表。在此例中，最后一跳的 IP 地址为 127.0.0.1。

请求描述	示例请求	append	preserve	remove
发送请求时没有 XFF 标头	GET / index.html HTTP/1.1 Host: example.com	X-Forwarded-For: 127.0.0.1	不存在	不存在
发送请求时包含一个 XFF 标头和一个客户端 IP 地址。	GET / index.html HTTP/1.1 Host: example.com X-Forward	X-Forwarded-For: 127.0.0.4, 127.0.0.1	X-Forwarded-For: 127.0.0.4	不存在

请求描述	示例请求	append	preserve	remove
	`ed-For: 127.0.0.4`			
发送请求时包含一个 XFF 标头和多个客户端 IP 地址。	`GET / index.html HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4, 127.0.0.8`	`X-Forwarded-For: 127.0.0.4, 127.0.0.8, 127.0.0.1`	`X-Forwarded-For: 127.0.0.4, 127.0.0.8`	不存在

Console

为了管理 X-Forwarded-For header

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在属性选项卡上，选择编辑。
5. 在 “流量配置” 部分的 “数据包处理” 下，为 “X-Forwarded-For 标题” 选择 “追加（默认）”、“保留” 或 “移除”。
6. 选择保存更改。

Amazon CLI

为了管理 X-Forwarded-For header

使用带 `routing.http.xff_header_processing.mode` 属性的 modify-load-balancer-attributes 命令。可能的值为 `append`、`preserve` 和 `remove`。默认为 `append`。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=routing.http.xff_header_processing.mode,Value=preserve"
```

CloudFormation

为了管理 X-Forwarded-For header

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含该routing.http.xff_header_processing.mode属性。可能的值为 append、preserve 和 remove。默认为 append。

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.xff_header_processing.mode"
          Value: "preserve"
```

X-Forwarded-Proto

X-Forwarded-Proto 请求标头可帮助您识别客户端与您的负载均衡器连接时所用的协议 (HTTP 或 HTTPS)。您的服务器访问日志仅包含在服务器和负载均衡器之间使用的协议；不含任何关于在客户端和负载均衡器之间使用的协议之信息。如需判断在客户端和负载均衡器之间使用的协议，使用 X-Forwarded-Proto 请求标题。Elastic Load Balancing 会在 X-Forwarded-Proto 请求标头中存储客户端和负载均衡器之间使用的协议，并将标头传递到您的服务器。

您的应用程序或网站可以使用存储在 X-Forwarded-Proto 请求标头中的协议来呈现重新定向至适用 URL 的响应。

X-Forwarded-Proto 请求标头采用以下形式：

```
X-Forwarded-Proto: originatingProtocol
```

以下示例包含以 HTTPS 请求形式源自客户端的请求的 X-Forwarded-Proto 请求标头：

```
X-Forwarded-Proto: https
```

X-Forwarded-Port

`X-Forwarded-Port` 请求标头可帮助您识别客户端与您的负载均衡器连接时所用的目标端口。

适用于应用程序负载均衡器的 HTTP 标头修改

应用程序负载均衡器的请求标头和响应标头都支持 HTTP 标头修改。无需更新应用程序代码，通过标头修改即可更好地控制应用程序的流量和安全性。

要启用标头修改功能，请参阅启用标头修改。

重命名 mTLS/TLS 标题

借助标头重命名功能，您可以配置由应用程序负载均衡器生成并添加到请求中的 mTLS 和 TLS 标头名称。

这种 HTTP 标头修改功能可让应用程序负载均衡器轻松支持使用特殊格式请求标头和响应标头的应用程序。

标题	说明
X-Amzn-Mtls-Clientcert-Serial-Number	确保目标能够识别和验证客户端在 TLS 握手期间提供的特定证书。
X-Amzn-Mtls-Clientcert-Issuer	通过识别颁发证书的证书颁发机构，帮助目标对客户端证书进行验证和确认身份。
X-Amzn-Mtls-Clientcert-Subject	向目标提供有关向其颁发客户端证书的实体的详细信息，有助于在 mTLS 身份验证期间进行识别、身份验证、授权和记录。
X-Amzn-Mtls-Clientcert-Validity	允许目标验证正在使用的客户端证书是否在既定有效期内，确保证书没有过期或过早使用。
X-Amzn-Mtls-Clientcert-Leaf	提供 mTLS 握手中使用的客户端证书，以方便服务器对客户端进行身份验证并验证证书链。这样可以确保连接安全且得到授权。

标题	说明
X-Amzn-Mtls-Clientcert	携带完整的客户端证书。方便目标在 mTLS 握手过程中验证证书的真实性、验证证书链并对客户端进行身份验证。
X-Amzn-TLS-Version	指示用于连接的 TLS 协议版本。它有助于确定通信的安全级别、解决连接问题并确保合规性。
X-Amzn-TLS-Cipher-Suite	指示用于保护 TLS 中连接的加密算法组合。这使服务器能够评估连接的安全性，帮助排查兼容性问题，以及确保遵守安全策略。

添加响应标头

通过使用插入标头，您可以将应用程序负载均衡器配置为在响应中添加与安全相关的标头。借助这些属性，您可以插入包括 HSTS、CORS 和 CSP 在内的各种标头。

默认情况下，这些标头为空。发生这种情况时，应用程序负载均衡器不会修改此响应标头。

启用某个响应标头时，应用程序负载均衡器会将具有所配置值的标头添加到所有响应中。如果来自目标的响应包含 HTTP 响应标头，则负载均衡器会将标头值更新为配置的值。否则，负载均衡器会将 HTTP 响应标头添加到具有所配置值的响应中。

标题	说明
Strict-Transport-Security	强制浏览器在指定的持续时间内进行 HTTPS-only 连接，这有助于防范中间人攻击、协议降级和用户错误。确保客户端和目标之间的所有通信都经过加密。
Access-Control-Allow-Origin	控制是否可以从不同的源访问目标上的资源。这有助于实现安全的跨源交互，同时防止未经授权的访问。
Access-Control-Allow-Methods	指定向目标发出跨源请求时允许的 HTTP 方法。用于控制可从不同源执行的操作。

标题	说明
Access-Control-Allow-Headers	指定跨源请求中可以包含的自定义标头或非简单标头。此标头让目标可以控制来自不同源的客户端可以发送的标头。
Access-Control-Allow-Credentials	指定客户端是否应在跨源请求中包含诸如 Cookie、HTTP 身份验证或客户端证书之类的凭证。
Access-Control-Expose-Headers	允许目标指定客户端可以在跨源请求中访问的其他响应标头。
Access-Control-Max-Age	定义浏览器可以缓存预检请求结果缓存的时长，从而减少重复预检的需要。这可以减少某些跨源请求所需的 OPTIONS 请求数量，从而有助于优化性能。
Content-Security-Policy	一种通过控制网站可以加载和执行脚本、样式、图像等资源，来防止 XSS 等代码注入攻击的安全功能。
X-Content-Type-Options	使用 no-sniff 指令来防止浏览器猜测资源的 MIME 类型，从而增强 Web 安全性。它确保浏览器仅根据声明的内容来解释内容 Content-Type
X-Frame-Options	一种通过控制是否可以将网页嵌入到框架中，从而帮助防止点击劫持攻击的标头安全机制。诸如 DENY 和 SAMEORIGIN 之类的值可以确保内容不会嵌入到恶意或不可信网站上。

禁用标头

借助禁用标头功能，您可以将应用程序负载均衡器配置为禁用响应中的 `server:awselb/2.0` 标头。这可以减少服务器特定信息的泄露，同时为应用程序提供额外的保护层。

属性名称为 routing.http.response.server.enabled。可用值为 true 或 false。默认值为 true。

限制

- 标头值可包含以下字符
  - 字母数字字符：a-z、A-Z 和 0-9
  - 特殊字符：_ :;.,\/'?!(){}[]@<>=-+*#&`|~^%
- 该属性的值大小不能超过 1K 字节。
- 弹性负载均衡会执行基本的输入验证来确认标头值是否有效。但是，验证无法确认特定的标头是否支持该值。
- 为任何属性设置空值都将导致应用程序负载均衡器还原默认行为。

为应用程序负载均衡器启用 HTTP 标头修改

默认情况下，标头修改功能处于关闭状态，必须在每个侦听器上启用。有关更多信息，请参阅 HTTP 标头修改。

Console

启用标头修改

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择该应用程序负载均衡器。
4. 在侦听器和规则选项卡上，选择协议和端口，从而打开侦听器的详细信息页面。
5. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。

   侦听器属性分为若干组。您需要选择要启用的功能。
6. [HTTPS 侦听器] 可修改 mTLS/TLS 的标头名称
   a. 展开可修改的 mTLS/TLS 标题名称。
   b. 启用要修改的请求标头并提供标头的名称。有关更多信息，请参阅 the section called “重命名 mTLS/TLS 标题”。
7. 添加响应标头

a. 展开添加响应标头。

b. 启用要添加的响应标头并提供标头的值。有关更多信息，请参阅 the section called “添加响应标头”。

8. ALB 服务器响应标头

- 启用或禁用服务器标头。

9. 选择保存更改。

Amazon CLI

启用标头修改

使用 modify-listener-attributes 命令。要查看属性列表，请参阅the section called “标头修改属性”。

```
aws elbv2 modify-listener-attributes \
    --listener-arn listener-arn \
    --attributes "Key=attribute-name,Value=attribute-value"
```

CloudFormation

启用标头修改

更新 AWS::ElasticLoadBalancingV2:: Listener 资源以包含这些属性。要查看属性列表，请参阅the section called “标头修改属性”。

```
Resources:
  myHTTPlistener:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    LoadBalancerArn: !Ref myLoadBalancer
    Protocol: HTTP
    Port: 80
    DefaultActions:
      - Type: "forward"
        TargetGroupArn: !Ref myTargetGroup
    ListenerAttributes:
      - Key: "attribute-name"
        Value: "attribute-value"
```

标头修改属性

以下为应用程序负载均衡器支持的标头修改属性。

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

修改的标题名称X-Amzn-Mtls-Clientcert-Serial-Number。

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

修改的标题名称X-Amzn-Mtls-Clientcert-Issuer。

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

修改的标题名称X-Amzn-Mtls-Clientcert-Subject。

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

修改的标题名称X-Amzn-Mtls-Clientcert-Validity。

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

修改的标题名称X-Amzn-Mtls-Clientcert-Leaf。

`routing.http.request.x_amzn_mtls_clientcert.header_name`

修改的标题名称X-Amzn-Mtls-Clientcert。

`routing.http.request.x_amzn_tls_version.header_name`

修改的标题名称X-Amzn-Tls-Version。

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

修改的标题名称X-Amzn-Tls-Cipher-Suite。

`routing.http.response.server.enabled`

指示是否允许或移除 HTTP 响应服务器标头。

`routing.http.response.strict_transport_security.header_value`

添加标Strict-Transport-Security题以告知浏览器只能使用 HTTPS 访问该网站，并且将来任何使用 HTTP 访问该网站的尝试都应自动转换为 HTTPS。

`routing.http.response.access_control_allow_origin.header_value`

添加标Access-Control-Allow-Origin头以指定允许哪些源访问服务器。

`routing.http.response.access_control_allow_methods.header_value`

添加标Access-Control-Allow-Methods头以指定从其他来源访问服务器时允许使用哪些 HTTP 方法。

`routing.http.response.access_control_allow_headers.header_value`

添加标Access-Control-Allow-Headers头以指定在跨域请求期间允许使用哪些标头。

`routing.http.response.access_control_allow_credentials.header_value`

添加标Access-Control-Allow-Credentials头以指示浏览器是否应在跨源请求中包含诸如 Cookie 或身份验证之类的凭据。

`routing.http.response.access_control_expose_headers.header_value`

添加标Access-Control-Expose-Headers头以指示浏览器可以向请求的客户端公开哪些标头。

`routing.http.response.access_control_max_age.header_value`

添加标Access-Control-Max-Age头以指定预检请求的结果可以缓存多长时间（以秒为单位）。

`routing.http.response.content_security_policy.header_value`

添加标Content-Security-Policy题以指定浏览器实施的限制，以帮助最大限度地降低某些类型安全威胁的风险。

`routing.http.response.x_content_type_options.header_value`

添加标X-Content-Type-Options题以指示是否应遵循标Content-Type题中通告的 MIME 类型，而不应更改。

`routing.http.response.x_frame_options.header_value`

添加标X-Frame-Options题以指示是否允许浏览器在框架、iframe、嵌入或对象中呈现页面。

删除 Application Load Balancer 的侦听器

在删除侦听器之前，请考虑其对应用程序的影响：

- 负载均衡器会立即停止接受侦听器上的新连接。
- 活动连接已关闭。侦听器删除后正在进行的任何请求都可能会失败。

Console

删除侦听器

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择负载均衡器。
4. 在侦听器和规则选项卡上，选中侦听器对应的复选框，然后依次选择管理侦听器、删除侦听器。
5. 提示进行确认时，输入 **confirm**，然后选择删除。

Amazon CLI

删除侦听器

使用 delete-listener 命令。

```
aws elbv2 delete-listener \
    --listener-arn listener-arn
```

Application Load Balancer 的目标组

目标组使用您指定的协议和端口号将请求路由到个别注册目标，例如 EC2 实例。您可以向多个目标组注册一个目标。您可以对每个目标组配置运行状况检查。在注册到目标组 (它是使用负载均衡器的侦听器规则指定的) 的所有目标上，执行运行状况检查。

每个目标组均用于将请求路由到一个或多个已注册的目标。在创建每个侦听器规则时，可以指定目标组和条件。满足规则条件时，流量会转发到相应的目标组。您可以为不同类型的请求创建不同的目标组。例如，为一般请求创建一个目标组，为应用程序的微服务请求创建其他目标组。您可以将每个目标组仅与一个负载均衡器一起使用。有关更多信息，请参阅 Application Load Balancer 组件。

您基于每个目标组定义负载均衡器的运行状况检查设置。每个目标组均使用默认运行状况检查设置，除非您在创建目标组时将其覆盖或稍后对其进行修改。在侦听器规则中指定一个目标组后，负载均衡器将持续监控已注册到该目标组的所有目标 (这些目标位于已为负载均衡器启用的可用区中) 的运行状况。负载均衡器将请求路由到正常运行的已注册目标。

目录

路由配置

默认情况下，负载均衡器会使用您在创建目标组时指定的协议和端口号将请求路由到其目标。此外，您可以覆盖在将目标注册到目标组时用于将流量路由到目标的端口。

目标组支持以下协议和端口：

- 协议：HTTP、HTTPS
- 端口：1-65535

当目标组配置了 HTTPS 协议或使用 HTTPS 运行状况检查时，如果任何 HTTPS 侦听器正在使用 TLS 1.3 安全策略，则将使用 ELBSecurityPolicy-TLS13-1-0-2021-06 安全策略进行目标连接。否则，将使用 ELBSecurityPolicy-2016-08 安全策略。负载均衡器将使用您在目标上安装的证书与目标建立 TLS 连接。负载均衡器不验证这些证书。因此，您可以使用自签名证书或已过期的证书。由于负载均衡器及其目标位于虚拟私有云（VPC）中，因此负载均衡器与目标之间的流量将在数据包级别进行身份验证，这样即使目标上的证书无效，它也不会面临中间人攻击或欺骗风险。离开的流量 Amazon 将没有同样的保护，可能需要采取额外的措施来进一步保护流量。

Target type

创建目标组时，指定其目标类型，此类型将确定您在向此目标组注册目标时指定的目标的类型。创建目标组后，您无法更改其目标类型。

以下是可能的目标类型：

`instance`

这些目标通过实例 ID 指定。

`ip`

目标是 IP 地址。

`lambda`

目标是 Lambda 函数。

当目标类型为 `ip` 时，您可以指定来自以下 CIDR 块之一的 IP 地址：

- 目标组的 VPC 的子网

- 10.0.0。 0/8 （ RFC 1918 ）
- 100.64.0。 0/10 (RFC 6598)
- 172.16.0。 0/12 （ RFC 1918 ）
- 192.168.0。 0/16 （ RFC 1918 ）

> **Important**
> 不能指定可公开路由的 IP 地址。

您可以使用所有支持的 CIDR 块，向目标组注册以下目标：

- 实例位于与负载均衡器 VPC 对等的 VPC（位于同一区域或不同区域）中。
- Amazon 可通过 IP 地址和端口寻址的资源（例如数据库）。
- On-premises Amazon 通过 Amazon Direct Connect 或 Site-to-Site VPN 连接链接到的资源。

> **Note**
> 对于部署在本地区域内的应用程序负载均衡器，`ip` 目标必须位于同一个本地区域中才能接收流量。
> 有关更多信息，请参阅什么是 L Amazon ocal Zones？

如果使用实例 ID 指定目标，则使用实例的主网络接口中指定的主私有 IP 地址将流量路由到实例。如果使用 IP 地址指定目标，则可以使用来自一个或多个网络接口的任何私有 IP 地址将流量路由到实例。这使一个实例上的多个应用程序可以使用同一端口。每个网络接口都可以有自己的安全组。

如果您的目标组的目标类型为 `lambda`，则可注册单个 Lambda 函数。当负载均衡器收到 Lambda 函数的请求时，它会调用 Lambda 函数。有关更多信息，请参阅 使用 Lambda 函数作为应用程序负载均衡器的目标。

您可以将 Amazon Elastic Container Service（Amazon ECS）配置为应用程序负载均衡器的目标。有关更多信息，请参阅《Amazon Elastic Container Service 用户指南》中的使用 Amazon ECS 的应用程序负载均衡器。

IP 地址类型

创建新目标组时，可以选择目标组的 IP 地址类型。此 IP 地址控制用于与目标进行通信并检查其运行状况的 IP 版本。

应用程序负载均衡器的目标组支持以下 IP 地址类型：

`ipv4`

负载均衡器使用 IPv4 与目标通信。

`ipv6`

负载均衡器使用 IPv6 与目标通信。

注意事项

- 负载均衡器根据目标组的 IP 地址类型与目标进行通信。IPv4 目标组的目标必须接受来自负载均衡器的 IPv4 流量，IPv6 目标组的目标必须接受来自负载均衡器的 IPv6 流量。
- 不能将 IPv6 目标组与 `ipv4` 负载均衡器结合使用。
- 不能将 Lambda 函数注册到 IPv6 目标组。

协议版本

默认情况下，应用程序负载均衡器使用 HTTP/1.1向目标发送请求。您可以使用协议版本通过 HTTP/2 或 gRPC 向目标发送请求。

下表汇总了请求协议和目标组协议版本组合的结果。

请求协议	协议版本	结果
HTTP/1.1	HTTP/1.1	成功
HTTP/2	HTTP/1.1	成功
gRPC	HTTP/1.1	错误
HTTP/1.1	HTTP/2	错误

请求协议	协议版本	结果
HTTP/2	HTTP/2	成功
gRPC	HTTP/2	如果目标支持 grPC，则成功
HTTP/1.1	gRPC	错误
HTTP/2	gRPC	如果 POST 请求，则成功
gRPC	gRPC	成功

gRPC 协议版本的注意事项

- 唯一支持的侦听器协议是 HTTPS。
- 监听器规则唯一支持的操作类型是 `forward`。
- 唯一支持的目标类型是 `instance` 和 `ip`。
- 负载均衡器解析 gRPC 请求并根据程序包、服务和方法将 gRPC 调用路由到相应的目标组。
- 负载均衡器支持一元、客户端流媒体、服务器端流媒体和双向流媒体。
- 您必须提供格式为 `/package.service/method` 的自定义运行状况检查方法。
- 在检查来自目标的成功响应时，必须指定 gRPC 状态代码。
- 不能将 Lambda 函数用作目标。

HTTP/2 协议版本的注意事项

- 唯一支持的侦听器协议是 HTTPS。
- 监听器规则唯一支持的操作类型是 `forward`。
- 唯一支持的目标类型是 `instance` 和 `ip`。
- 负载均衡器支持一元、客户端流媒体、服务器端流媒体和双向流媒体。每个客户端 HTTP/2 连接的最大流数为 128。

已注册目标

您的负载均衡器充当客户端的单一接触点，并跨其正常运行的已注册目标分发传入流量。您可以将每个目标注册到一个或多个目标组中。

如果对应用程序的需求增加，则可以向一个或多个目标组注册其他目标来处理需求。一旦注册过程完成并且目标通过了第一次初始运行状况检查，负载均衡器就会开始将流量路由到新注册的目标，而不管配置的阈值如何。

如果应用程序需求减少或者您需要为目标提供服务，您可以从目标组取消注册目标。取消注册目标将从目标组中删除目标，但不会影响目标。取消注册某个目标后，负载均衡器立即停止将请求路由到该目标。目标将进入 `draining` 状态，直至进行中请求完成。在您准备好目标以继续接收请求时，可以重新将目标注册到目标组。

如果要通过实例 ID 来注册目标，则可以将负载均衡器与 Auto Scaling 组一同使用。将一个目标组挂接到 Auto Scaling 组后，Auto Scaling 在启动目标时会为您向该目标组注册目标。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的将负载均衡器挂接到 Auto Scaling 组。

限制

- 不能注册同一 VPC 中其他应用程序负载均衡器的 IP 地址。如果另一个 Application Load Balancer 位于与负载均衡器 VPC 对等的 VPC 中，则可以注册其 IP 地址。
- 如果实例位于与负载均衡器 VPC 对等的 VPC（位于同一区域或不同区域）中，则不能利用实例 ID 注册这些实例。可以用 IP 地址注册这些实例。

目标优化器

可以在目标组上启用目标优化器。目标优化器允许您在目标上准确地强制执行最大数量的并发请求。它需要在目标上安装和配置的代理的帮助下运行。要启用目标优化器，请为目标组指定目标控制端口。此端口用于管理代理和负载均衡器之间的流量。目标优化器只能在创建目标组期间启用。一旦指定了目标控制端口，便无法修改。有关更多信息，请参阅 the section called “目标优化器”。

目标组属性

您可以通过编辑目标组的属性来配置它。有关更多信息，请参阅 编辑目标组属性。

如果目标组类型为 `instance` 或 `ip`，则支持以下目标组属性：

`deregistration_delay.timeout_seconds`

Elastic Load Balancing 在取消注册目标之前等待的时间量。范围为 0–3600 秒。默认值为 300 秒。

`load_balancing.algorithm.type`

路由算法决定了负载均衡器在路由请求时将如何选择目标。值为 `round_robin`、`least_outstanding_requests` 或 `weighted_random`。默认值为 `round_robin`。

`load_balancing.algorithm.anomaly_mitigation`

仅当 `load_balancing.algorithm.type` 为 `weighted_random` 时可用。指示是否启用异常缓解。该值为 `on` 或 `off`。默认为 `off`。

`load_balancing.cross_zone.enabled`

指示是否启用了跨区域负载均衡。该值为 `true`、`false` 或 `use_load_balancer_configuration`。默认为 `use_load_balancer_configuration`。

`slow_start.duration_seconds`

一个时间段 (秒)，在此期间，负载均衡器将进入目标组的流量的线性增加份额发送给新注册的目标。范围为 30–900 秒 (15 分钟)。默认值为 0 秒 (已禁用)。

`stickiness.enabled`

指示是否启用粘性会话。该值为 `true` 或 `false`。默认为 `false`。

`stickiness.app_cookie.cookie_name`

应用程序 Cookie 的名称。应用程序 Cookie 名称不能具有以下前缀：`AWSALB`、`AWSALBAPP`、或 `AWSALBTG`；这些前缀保留供负载均衡器使用。

`stickiness.app_cookie.duration_seconds`

基于应用程序的 Cookie 有效期（以秒为单位）。经过这个有效期后，Cookie 即过期。最小值为 1 秒，最大值为 7 天 (604800 秒)。默认值为 1 天 (86400 秒)。

`stickiness.lb_cookie.duration_seconds`

基于持续时间的 Cookie 有效期（以秒为单位）。经过这个有效期后，Cookie 即过期。最小值为 1 秒，最大值为 7 天 (604800 秒)。默认值为 1 天 (86400 秒)。

`stickiness.type`

粘性的类型。可能的值为 `lb_cookie` 和 `app_cookie`。

`target_group_health.dns_failover.minimum_healthy_targets.count`

必须运行状况良好的目标数量下限。如果运行正常的目标数低于此值，请在 DNS 中将该节点标记为运行不正常，从而将流量仅路由到运行正常的节点。可能的值是 `off` 或者 1 到目标数量上限之间

的整数。设置为 off 时，DNS 失效转移处于禁用状态，这意味着即使目标组中的所有目标运行都不正常，也不会从 DNS 中移除该节点。默认为 1。

target_group_health.dns_failover.minimum_healthy_targets.percentage

必须运行状况良好的目标最低百分比。如果运行状况良好的目标百分比低于此值，请在 DNS 中将节点标记为运行状况不佳，以便流量仅路由到运行状况良好的节点。可能的值为 off 或者 1 到 100 之间的整数。设置为 off 时，DNS 失效转移处于禁用状态，这意味着即使目标组中的所有目标运行都不正常，也不会从 DNS 中移除该节点。默认值为 off。

target_group_health.unhealthy_state_routing.minimum_healthy_targets.count

必须运行状况良好的目标数量下限。如果运行状况良好的目标数量低于此值，则将流量发送到所有目标（包括运行状况不佳的目标）。范围为 1 到目标数量上限。默认为 1。

target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage

必须运行状况良好的目标最低百分比。如果运行状况良好的目标百分比低于此值，则将流量发送到所有目标（包括运行状况不佳的目标）。可能的值为 off 或者 1 到 100 之间的整数。默认值为 off。

如果目标组类型为 lambda，则支持以下目标组属性：

lambda.multi_value_headers.enabled

指示在负载均衡器和 Lambda 函数之间交换的请求和响应标头是否包含值或字符串的数组。可能的值为 true 或 false。默认值为 false。有关更多信息，请参阅 Multi-value 标题。

目标组运行状况

默认情况下，只要目标组至少有一个运行状况良好的目标，就会被视为运行状况良好。如果您的实例集很大，则仅有一个运行状况良好的目标为流量提供服务是不够的。相反，您可以指定必须运行状况良好的目标数量下限或最低百分比，以及当运行状况良好的目标低于指定阈值时负载均衡器将采取哪些操作。这有助于提高您应用程序的可用性。

内容

- 为负载均衡器使用 Route 53 DNS 故障转移

运行状况不佳状态的操作

您可以为以下操作配置运行状况良好阈值：

- DNS 故障转移 — 当某区域中运行状况良好的目标低于阈值时，我们会在 DNS 中将该区域的负载均衡器节点的 IP 地址标记为运行状况不佳。因此，当客户端解析负载均衡器 DNS 名称时，流量将会仅路由到运行状况良好的区域。
- 路由故障转移 - 当某区域中运行状况良好的目标低于阈值时，负载均衡器会将流量发送到负载均衡器节点可用的所有目标（包括运行状况不佳的目标）。这增加了客户端连接成功的机会，尤其是在目标暂时未能通过运行状况检查时，并降低了运行状况良好的目标过载的风险。

要求和注意事项

- 如果您在目标组上启用目标优化器，我们建议您将目标组的运行状况检查端口设置为与 TARGET_CONTROL_DATA_ADDRESS 中的端口相同。这样可以确保如果代理运行状况不佳，目标将无法通过运行状况检查。有关更多信息，请参阅 the section called “目标优化器”。
- 不能将此功能用于目标是 Lambda 函数的目标组。如果应用程序负载均衡器是网络负载均衡器或全局加速器的目标，请不要为 DNS 故障转移配置阈值。
- 如果为某项操作指定了两种类型的阈值（计数和百分比），则负载均衡器会在违反任一阈值时执行该操作。
- 如果为这两项操作都指定了阈值，则 DNS 故障转移的阈值必须大于或等于路由故障转移的阈值，以便 DNS 故障转移会在路由故障转移时或之前发生。
- 如果您将阈值指定为百分比，我们将根据在目标组中注册的目标总数动态计算该值。
- 目标总数取决于关闭还是打开跨区域负载均衡。如果跨区域负载均衡处于关闭状态，则每个节点仅向自己区域中的目标发送流量，这意味着阈值将分别应用于每个已启用区域中的目标数量。如果跨区域负载均衡处于打开状态，则每个节点将流量发送到所有已启用区域中的所有目标，这意味着指定的阈值将应用于所有已启用区域中的目标总数。有关更多信息，请参阅 Cross-zone 负载平衡。
- 当发生 DNS 故障转移时，会影响与负载均衡器关联的所有目标组。请确保剩余区域中有足够的容量来处理这些额外流量，尤其是在跨区域负载均衡关闭的情况下。
- 通过 DNS 故障转移，我们会从负载均衡器的 DNS 主机名中删除运行状况不佳区域的 IP 地址。但是，本地客户端 DNS 缓存中可能会包含这些 IP 地址，直到 DNS 记录中的生存时间 (TTL) 到期（60 秒）。

- 使用 DNS 故障转移时，如果一个应用程序负载均衡器附加了多个目标组，并且一个可用区中有一个目标组运行不正常，同时该可用区中至少有另外一个目标组运行正常，则将会通过 DNS 运行状况检查。
- 使用 DNS 故障转移时，如果所有负载均衡器区域都被视为运行状况不佳，则负载均衡器会将流量发送到所有区域（包括运行状况不佳的区域）。
- 除了是否有足够运行状况良好的目标可能会导致 DNS 故障转移之外，还有其他因素，例如区域的运行状况。

监控

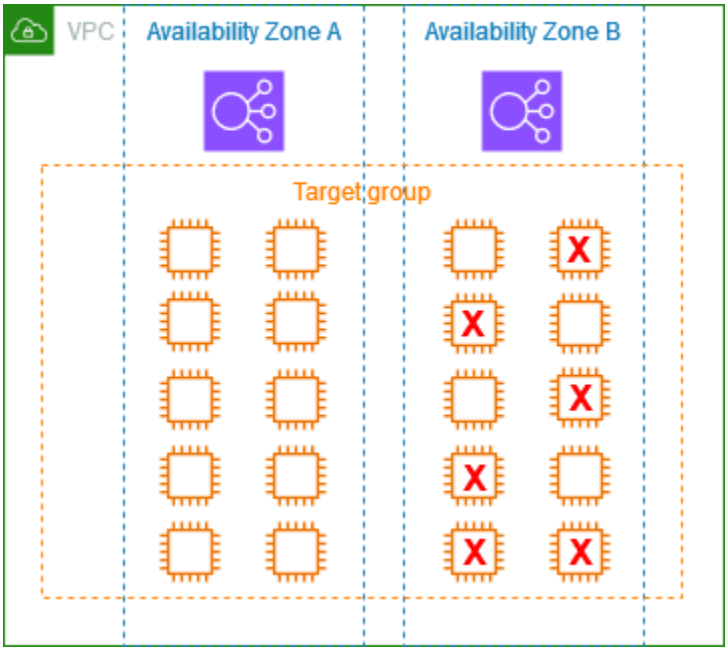
要监控目标群体的健康状况，请参阅目标群体的健康CloudWatch 指标。

示例

以下示例演示了如何应用目标组运行状况设置。

场景

- 支持 A 和 B 两个可用区的负载均衡器
- 每个可用区中包含 10 个注册目标
- 目标组具有以下目标组运行状况设置：
  - DNS 故障转移 - 50%
  - 路由故障转移 - 50%
- 可用区 B 中有六个目标失败



如果跨区域负载均衡关闭

- 每个可用区中的负载均衡器节点只能将流量发送到其可用区内的 10 个目标。
- 可用区 A 中有 10 个运行状况良好的目标，符合所需的运行状况良好的目标百分比。负载均衡器继续在 10 个运行状况良好的目标之间分配流量。
- 可用区 B 中只有 4 个运行状况良好的目标，占可用区 B 中负载均衡器节点目标的 40%。由于这低于所需的运行状况良好的目标百分比，负载均衡器会执行以下操作：
  - DNS 故障转移 - 可用区 B 在 DNS 中被标记为运行状况不佳。由于客户端无法将负载均衡器名称解析为可用区 B 中的负载均衡器节点，并且可用区 A 运行状况良好，因此客户端会向可用区 A 发送新连接。
  - 路由故障转移 - 当新连接明确发送到可用区 B 时，负载均衡器会将流量分配到可用区 B 中的所有目标（包括运行状况不佳的目标）。这样可以防止剩余运行状况良好的目标发生中断。

如果跨区域负载均衡打开

- 每个负载均衡器节点可以向两个可用区中的所有 20 个注册目标发送流量。
- 可用区 A 中有 10 个运行状况良好的目标，可用区 B 中有 4 个运行状况良好的目标，总共有 14 个运行状况良好的目标。这是两个可用区中负载均衡器节点目标的 70%，符合所需的运行状况良好的目标百分比。
- 负载均衡器将在两个可用区内 14 个运行状况良好的目标之间分配流量。

为负载均衡器使用 Route 53 DNS 故障转移

如果使用 Route 53 将 DNS 查询路由到您的负载均衡器，您也可以使用 Route 53 为您的负载均衡器配置 DNS 故障转移。在失效转移配置中，Route 53 将检查负载均衡器的目标组目标的运行状况以确定目标是否可用。如果没有已注册到负载均衡器的运行状况正常的目标，或如果负载均衡器本身运行状况不佳，则 Route 53 会将流量路由到其他可用资源，例如 Amazon S3 中运行状况正常的负载均衡器或静态网站。

例如，假设您有一个用于 `www.example.com` 的 Web 应用程序，并且您希望使用在不同区域内的两个负载均衡器之后运行的冗余实例。您希望流量主要路由到一个区域中的负载均衡器，并且您希望在发生故障期间将另一个区域中的负载均衡器用作备份。如果配置 DNS 故障转移，则可以指定您的主和辅助 (备份) 负载均衡器。如果主负载均衡器可用，则 Route 53 会将流量定向到主负载均衡器，否则会将流量定向到辅助负载均衡器。

“评估目标运行状况”的工作原理

- 在应用程序负载均衡器别名记录上将“评估目标运行状况”设置为 `Yes` 时，Route 53 将评估 `alias target` 值指定的资源的运行状况。Route 53 使用目标组运行状况检查。
- 如果附加到应用程序负载均衡器的所有目标组均运行正常，Route 53 会将别名记录标记为运行正常。如果您为目标组配置了阈值且该目标组满足其阈值要求，则视为“通过运行状况检查”。否则，只要目标组包含至少一个运行正常的目标，即视为“通过运行状况检查”。如果“通过运行状况检查”，则 Route 53 会根据您的路由策略返回记录。如果使用失效转移路由策略，则 Route 53 会返回主记录。
- 如果附加到应用程序负载均衡器的任何目标组运行不正常，则别名记录无法通过 Route 53 运行状况检查（失效时开放）。如果使用“评估目标运行状况”，则失效转移路由策略会将流量重定向到辅助资源。
- 如果附加到应用程序负载均衡器的所有目标组均为空（无目标），则 Route 53 会认为该记录运行不正常（失效时开放）。如果使用“评估目标运行状况”，则失效转移路由策略会将流量重定向到辅助资源。

有关更多信息，请参阅 Amazon 博客中的使用负载均衡器目标组运行状况阈值提高可用性和 Amazon Route 53 开发人员指南中的配置 DNS 故障转移。

为您的应用程序负载均衡器创建目标组

将目标注册到目标组。默认情况下，负载均衡器使用您为目标组指定的端口和协议将请求发送到已注册目标。在将每个目标注册到目标组时，可以覆盖此端口。

在创建目标组后，您可以添加标签。

要将流量路由到目标组中的目标，请在创建侦听器或侦听器规则时，在操作中指定目标组。有关更多信息，请参阅 Application Load Balancer 的侦听器规则。您可以在多个侦听器中指定同一个目标组，但这些侦听器必须属于同一个 Application Load Balancer。要将目标组与负载均衡器结合使用，您必须确认目标组没有被任何其他负载均衡器的侦听器使用。

您可以随时在目标组中添加或删除目标。有关更多信息，请参阅 向应用程序负载均衡器目标组注册目标。您也可以修改目标组的运行状况检查设置。有关更多信息，请参阅 更新应用程序负载均衡器目标组的运行状况检查设置。

Console

创建目标组

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的 Load Balancing (负载均衡) 下，选择 Target Groups (目标组)。
3. 选择 Create target group (创建目标组)。
4. 对于选择目标类型，若要按实例 ID 注册目标，则选择实例；要按 IP 地址注册目标，则选择 IP 地址；要将 Lambda 函数注册为目标，则选择 Lambda 函数。
5. 对于 Target group name，键入目标组的名称。此名称在每个区域的每个账户中必须唯一，最多可以有 32 个字符，只能包含字母数字字符或连字符，不得以连字符开头或结尾。
6. (可选) 对于 Protocol (协议) 和 Port (端口)，根据需要修改默认值。
7. 如果目标类型为实例或 IP 地址，则对于 IP 地址类型，请选择 IPv4 或 IPv6，否则请跳至下一步。

   请注意，仅具有选定 IP 地址类型的目标才能包括在此目标组中。创建目标组后，无法更改 IP 地址类型。
8. 对于 VPC，选择 Virtual Private Cloud (VPC)。请注意，对于 IP 地址目标类型，可供选择的 VPC 是支持上一步中所选 IP 地址类型 的 VPC。
9. （可选）对于 Protocol version (协议版本)，根据需要修改默认值。有关更多信息，请参阅 the section called “协议版本”。
10. （可选）在 Health checks (运行状况检查) 部分中，根据需要修改默认设置。有关更多信息，请参阅 the section called “运行状况检查设置”。
11. 如果目标类型为 Lambda 函数，则可以通过在 Health checks (运行状况检查) 部分中选择 Enable (启用) 来启用运行状况检查。

12. （可选）要在目标组上启用目标优化器，请指定目标控制端口。创建目标组后无法修改端口。目标优化器在目标上安装的代理的帮助下运行。有关更多信息，请参阅 the section called “目标优化器”。
13. (可选) 添加一个或多个标签，如下所示：
    a. 展开标签部分。
    b. 选择 Add tag (添加标签)。
    c. 输入标签键和标签值。
14. 选择下一步。
15. （可选）添加一个或多个目标，如下所示：
    - 如果目标类型为实例，请选择一个或多个实例，输入一个或多个端口，然后选择在下面以待注册的形式添加。

      注意：实例必须具有分配的主 IPv6 地址，才能向 IPv6 目标组注册。
    - 如果目标类型为 IP addresses (IP 地址)，请执行以下操作：
      a. 从列表中选择网络 VPC，或选择 Other private IP addresses (其他私有 IP 地址)。
      b. 手动输入 IP 地址，或使用实例详细信息查找 IP 地址。一次最多可输入 5 个 IP 地址。
      c. 输入将流量路由到指定 IP 地址的端口。
      d. 选择 Include as pending below (在下面以待注册的形式添加)。
    - 如果目标类型是 Lambda 函数，请指定单个 Lambda 函数，或者忽略此步骤并稍后指定 Lambda 函数。
16. 选择创建目标组。

Amazon CLI

创建目标组

使用 create-target-group 命令。以下示例创建了一个具有 HTTP 协议、若干按 IP 地址注册的目标、一个标签和默认运行状况检查设置的目标组。

```
aws elbv2 create-target-group \
    --name my-target-group \
    --protocol HTTP \
    --port 80 \
    --target-type ip \
```

```
    --vpc-id vpc-1234567890abcdef0 \
    --tags Key=department,Value=123
```

注册目标

使用 register-targets 命令将目标注册到目标组。有关示例，请参阅 the section called “注册目标”。

CloudFormation

创建目标组

定义类型为 AWS::ElasticLoadBalancingV2:: TargetGroup 的资源。以下示例创建了一个具有 HTTP 协议、若干按 IP 地址注册的目标、一个标签、默认运行状况检查设置和两个已注册目标的目标组。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: 10.0.50.10
          Port: 80
        - Id: 10.0.50.20
          Port: 80
```

应用程序负载均衡器目标组的运行状况检查

您的 Application Load Balancer 会定期向其注册目标发送请求以测试其状态。这些测试称为运行状况检查。

每个负载均衡器节点仅将请求路由至负载均衡器的已启用可用区中的正常目标。每个负载均衡器节点均使用每个目标注册到的目标组的运行状况检查设置来检查该目标的运行状况。在注册目标后，目标必须通过一次运行状况检查才会被视为正常。在完成每次运行状况检查后，负载均衡器节点将关闭为运行状况检查而建立的连接。

如果目标组仅包含运行状况不佳的注册目标，则负载均衡器将请求路由到所有这些目标，而不考虑这些目标的运行状况。这意味着，如果在所有已启用的可用区中，所有目标都未通过运行状况检查，则负载均衡器将在失败时开放。失败时开放的效果是根据负载均衡算法，允许传输到所有已启用的可用区中的所有目标的流量，而不考虑这些目标的运行状况。

不支持 Health 检查 WebSockets。

有关更多信息，请参阅 the section called “目标组运行状况”。

您可以使用运行状况检查日志来捕获有关对负载均衡器的注册目标进行的运行状况检查的详细信息，并将其作为日志文件存储在 Amazon S3 中。您可以使用这些运行状况检查日志来解决目标的问题。有关更多信息，请参阅 Health 检查日志。

内容

运行状况检查设置

如下表所述，您可以为目标组中的目标配置运行状况检查。表中使用的设置名称是 API 中使用的名称。负载均衡器使用指定的端口、协议和运行状况检查路径，每HealthCheckIntervalSeconds秒钟向每个注册目标发送一次运行状况检查请求。每个运行状况检查请求都是独立的，其结果在整个时间间隔内持续。目标响应所用时间不影响下一运行状况检查请求的时间间隔。如果运行状况检查超过UnhealthyThresholdCount连续失败次数，则负载均衡器会使目标停止服务。当运行状况检查超过HealthyThresholdCount连续成功率时，负载均衡器会将目标重新投入使用。

请注意，当您注销目标时，该值会减少HealthyHostCount但不会增加UnhealthyHostCount。

设置	说明
HealthCheckProtocol	对目标执行运行状况检查时负载均衡器使用的协议。对于应用程序负载均衡器，可能的协议是 HTTP 和 HTTPS。默认值为 HTTP 协议。

设置	说明
	这些协议使用 HTTP GET 方法发送运行状况检查请求
HealthCheckPort	对目标执行运行状况检查时负载均衡器使用的端口。默认设置是使用每个目标用来从负载均衡器接收流量的端口。
HealthCheckPath	目标运行状况检查的目的地。 如果协议版本是 HTTP/1.1 或 HTTP/2，请指定有效的 URI (/path?query)。默认值为 /。 如果协议版本是 gRPC，请使用格式 `/package.service/method` 指定自定义运行状况检查方法的路径。默认为 `/AWS.ALB/healthcheck` 。
HealthCheckTimeoutSeconds	以秒为单位的时间长度，在此期间内，没有来自目标的响应意味着无法通过运行状况检查。范围为 2–120 秒。默认值为 5 秒（如果目标类型为 `instance` 或 `ip`）和 30 秒（如果目标类型为 `lambda`）。
HealthCheckIntervalSeconds	各个目标的运行状况检查之间的大约时间量 (以秒为单位)。范围为 5–300 秒。默认值为 30 秒（如果目标类型为 `instance` 或 `ip`）和 35 秒（如果目标类型为 `lambda`）。
HealthyThresholdCount	将不正常目标视为正常运行之前所需的连续运行状况检查成功次数。范围为 2–10。默认值为 5。
UnhealthyThresholdCount	将目标视为不正常之前所需的连续运行状况检查失败次数。范围为 2–10。默认值为 2。

设置	说明
Matcher	检查来自目标的成功响应时要使用的代码。这些代码在控制台中称为成功代码。 如果协议版本是 HTTP/1.1 或 HTTP/2，则可能的值在 200 到 499 之间。您可以指定多个值（例如，“200,202”）或一系列值（例如，“200-299”）。默认值为 200。 如果协议版本是 gRPC，则可能的值在 0 到 99 之间。您可以指定多个值（例如，“0,1”）或一系列值（例如，“0-5”）。默认值是 12。

目标运行状况

在负载均衡器向目标发送运行状况检查请求之前，您必须将目标注册到目标组，在侦听器规则中指定其目标组，并确保已为负载均衡器启用目标的可用区。目标必须先通过初始运行状况检查，然后才能接收来自负载均衡器的请求。在目标通过初始运行状况检查后，其状态为 `Healthy`。

下表描述已注册目标的正常状态的可能值。

值	说明
`initial`	负载均衡器正处于注册目标或对目标执行初始运行状况检查的过程中。 相关原因代码：`Elb.RegistrationInProgress` | `Elb.InitialHealthChecking`
`healthy`	目标正常。 相关原因代码：无
`unhealthy`	目标未响应运行状况检查或未通过运行状况检查。

值	说明
	相关原因代码：Target.ResponseCodeMismatch |Target.Timeout |Target.FailedHealthChecks |Elb.InternalError
unused	目标未注册到目标组，侦听器规则中未使用目标组，或者目标在没有启用的可用区中，或者目标处于停止或终止状态。 相关原因代码：Target.NotRegistered |Target.NotInUse |Target.InvalidState |Target.IpUnusable
draining	目标正在取消注册，连接即将耗尽。 相关原因代码：Target.DeregistrationInProgress
unavailable	对目标组禁用运行状况检查。 相关原因代码：Target.HealthCheckDisabled

运行状况检查原因代码

如果目标的状态是 Healthy 以外的任何值，则 API 将返回问题的原因代码和描述，并且控制台将显示相同的描述。以 Elb 开头的原因代码源自负载均衡器端，以 Target 开头的原因代码源自目标端。有关运行状况检查失败的可能原因的更多信息，请参阅故障排除。

原因代码	说明
Elb.InitialHealthChecking	正在进行初始运行状况检查
Elb.InternalError	由于内部错误，运行状况检查失败
Elb.RegistrationInProgress	目标注册正在进行中

原因代码	说明
Target.Deregistrat ionInProgress	目标取消注册正在进行中
Target.FailedHealthChecks	运行状况检查失败
Target.HealthCheck Disabled	运行状况检查已禁用
Target.InvalidState	目标处于停止状态 目标处于终止状态 目标处于终止或停止状态 目标处于无效状态
Target.IpUnusable	该 IP 地址正被负载均衡器使用，因此无法用作目标
Target.NotInUse	目标组没有被配置为接收来自负载均衡器的流量 目标处于没有为负载均衡器启用的可用区
Target.NotRegistered	目标未注册到目标组
Target.ResponseCod eMismatch	运行状况检查失败，显示以下代码：[code]
Target.Timeout	请求超时

检查应用程序负载均衡器目标的运行状况

您可以检查已注册到目标组的目标的运行状况。有关运行状况检查失败问题的帮助，请参阅问题排查：已注册目标未处于可用状态。

您可以使用运行状况检查日志来捕获有关对负载均衡器的注册目标进行的运行状况检查的详细信息，并将其作为日志文件存储在 Amazon S3 中。您可以使用这些运行状况检查日志来解决目标的问题。有关更多信息，请参阅 Health 检查日志。

Console

要检查目标的运行状况

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 详细信息选项卡会显示目标总数，以及处于各运行状况的目标数。
5. 在 Targets 选项卡上，Status 列指示每个目标的状态。
6. 如果状态是 `Healthy` 以外的任何值，则状态详细信息列将包含更多信息。

接收有关运行状况不佳的目标的电子邮件通知

使用 CloudWatch 警报触发 Lambda 函数以发送有关不健康目标的详细信息。有关 step-by-step说明，请参阅以下博客文章：识别负载均衡器的运行状况不佳的目标。

Amazon CLI

要检查目标的运行状况

使用 describe-target-health 命令。此示例对输出进行筛选，以仅包括运行状况不良的目标。对于运行状况不良的目标，输出将包含原因代码。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
    --query "TargetHealthDescriptions[?TargetHealth.State!='healthy'].
[Target.Id,TargetHealth.State,TargetHealth.Reason]" \
    --output table
```

下面是示例输出。

```
-----------------------------------------------
|            DescribeTargetHealth            |
+--------------+---------+-------------------+
|  172.31.0.57 |  unused |  Target.NotInUse  |
|  172.31.0.50 |  unused |  Target.NotInUse  |
+--------------+---------+-------------------+
```

目标状态和原因代码

下表列出了每种目标状态的可能原因代码。

目标状态为 healthy

未提供原因代码。

目标状态为 initial

- `Elb.RegistrationInProgress` - 目标正处于与负载均衡器的注册流程中。
- `Elb.InitialHealthChecking` - 负载均衡器仍在向目标发送最低数量的运行状况检查，以确定其运行状况。

目标状态为 unhealthy

- `Target.ResponseCodeMismatch`：运行状况检查未返回预期的 HTTP 代码。
- `Target.Timeout`：运行状况检查请求超时。
- `Target.FailedHealthChecks`：负载均衡器在建立与目标的连接时收到错误，或目标响应格式错误。
- `Elb.InternalError`：由于内部错误，运行状况检查失败。

目标状态为 unused

- `Target.NotRegistered` - 目标未注册到目标组
- `Target.NotInUse` - 该目标组未被任何负载均衡器使用，或目标所在的可用区未启用其负载均衡器。
- `Target.InvalidState` - 目标处于停止或终止状态。
- `Target.IpUnusable` - 目标 IP 地址已保留，供负载均衡器使用。

目标状态为 draining

- `Target.DeregistrationInProgress` - 目标正处于注销过程中，且注销延迟期尚未到期。

目标状态为 unavailable

- `Target.HealthCheckDisabled`：目标组禁用了运行状况检查。

更新应用程序负载均衡器目标组的运行状况检查设置

您可以随时更新目标组的运行状况检查设置。有关运行状况检查设置的列表，请参阅 the section called “运行状况检查设置”。

Console

要更新运行状况检查设置

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Health checks 选项卡上，选择 Edit。
5. 在编辑运行状况检查设置页面上，根据需要修改设置。
6. 选择保存更改。

Amazon CLI

要更新运行状况检查设置

使用 modify-target-group 命令。以下示例更新了HealthyThresholdCount和HealthCheckTimeoutSeconds设置。

```
aws elbv2 modify-target-group \
    --target-group-arn target-group-arn \
    --healthy-threshold-count 3 \
    --health-check-timeout-seconds 20
```

CloudFormation

要更新运行状况检查设置

更新AWS::ElasticLoadBalancingV2::TargetGroup资源以包含更新的运行状况检查设置。以下示例更新了HealthyThresholdCount和HealthCheckTimeoutSeconds设置。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      HealthyThresholdCount: 3
```

```
    HealthCheckTimeoutSeconds: 20
```

编辑应用程序负载均衡器的目标组属性

为应用程序负载均衡器创建目标组后，您可以编辑其目标组属性。

目标组属性

- 取消注册延迟
- 路由算法
- 慢启动模式
- 运行状况设置
- Cross-zone 负载平衡
- 自动目标权重（ATW）
- 粘性会话
- WAF HTTP/2 流量检查行为

取消注册延迟

Elastic Load Balancing 停止将请求发送到正在取消注册的目标。默认情况下，Elastic Load Balancing 在取消注册过程完成前会等待 300 秒，这有助于完成针对目标的进行中的请求。要更改 Elastic Load Balancing 等待的时间，请更新取消注册延迟值。

取消注册的目标的初始状态为 `draining`。取消注册延迟结束后，取消注册过程完成，目标状态变为 `unused`。如果目标是 Auto Scaling 组的一部分，便可以将其终止或替换。

如果取消注册的目标没有进行中的请求且没有活动连接，则 Elastic Load Balancing 将立即完成取消注册过程，而不等待取消注册延迟结束。但是，即使目标注销已完成，目标的状态也会显示为 `draining`，直至注销延迟超时期限过期。超时期限过期后，目标转换为 `unused` 状态。

如果正在取消注册的目标在取消注册延迟结束前终止连接，客户端将收到 500 级错误响应。

Console

更新注销延迟值

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在目标注销管理窗格中，为注销延迟输入一个新值。
6. 选择保存更改。

Amazon CLI

更新注销延迟值

使用带 `deregistration_delay.timeout_seconds` 属性的 modify-target-group-attributes 命令。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=deregistration_delay.timeout_seconds,Value=60"
```

CloudFormation

更新注销延迟值

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该`deregistration_delay.timeout_seconds`属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "deregistration_delay.timeout_seconds"
          Value: "60"
```

路由算法

路由算法是在确定哪些目标将接收请求时负载均衡器使用的方法。默认情况下，使用轮询路由算法在目标组级别路由请求。还可以根据应用程序的需求使用最少未完成的请求和加权随机路由算法。一个目标组一次只能有一个活动的路由算法，但可以根据需要随时更新路由算法。

如果启用粘性会话，则所选的路由算法将用于初始目标选择。来自同一客户端的未来请求将被转发到同一目标，从而绕过所选的路由算法。如果您启用了目标优化器，则路由算法只能是循环算法。

轮询

- 轮询路由算法按顺序将请求均匀地路由到目标组中运行状况良好的目标。
- 当收到的请求复杂度相似、已注册目标的处理能力相似，或者您需要在目标之间平均分配请求时，通常使用此算法。

最少未完成请求

- 最少未完成的请求路由算法将请求路由到正在进行的请求数最少的目标。
- 当收到的请求复杂度不同、已注册目标的处理能力不同时，通常使用此算法。
- 当支持的负载均衡器使用 HTTP/2 仅支持的目标时 HTTP/1.1，它会将请求转换为多个 HTTP/1.1 请求。在此配置中，待处理请求最少的算法会将每个 HTTP/2 请求视为多个请求。
- 使用时 WebSockets，将使用未完成请求数最少算法选择目标。选择该目标后，负载均衡器将创建与该目标的连接并通过此连接发送所有消息。
- 最少未完成的请求路由算法不能与慢启动模式一起使用。

加权随机

- 加权随机路由算法以随机顺序在目标组中运行状况良好的目标之间均匀路由请求。
- 此算法支持自动目标权重（ATW）异常缓解。
- 加权随机路由算法不能与慢启动模式一起使用。
- 加权随机路由算法不能与粘性会话一起使用。

Console

更新路由算法

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在流量配置窗格中，对于负载均衡算法，选择轮询、最少未完成的请求或加权随机。
6. 选择保存更改。

Amazon CLI

更新路由算法

使用带 `load_balancing.algorithm.type` 属性的 modify-target-group-attributes 命令。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes
 "Key=load_balancing.algorithm.type,Value=least_outstanding_requests"
```

CloudFormation

更新路由算法

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含
该`load_balancing.algorithm.type`属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.algorithm.type"
          Value: "least_outstanding_requests"
```

慢启动模式

默认情况下，目标只要注册到目标组并通过了初始运行状况检查，就会开始接收其完整的请求份额。使用慢启动模式可给目标时间进行预热，然后负载均衡器向其发送完整的请求份额。

为目标组启用慢启动后，当目标组认为其目标正常时，其目标会进入慢启动模式。慢启动模式下的目标在配置的慢启动持续时间过去或目标变得不正常时退出慢启动模式。负载均衡器线性增加它可以向慢启动模式下的目标发送的请求数量。在正常目标退出慢启动模式后，负载均衡器可以向它发送完整的请求份额。

注意事项

- 为目标组启用慢启动之后，注册到目标组的正常目标不会进入慢启动模式。
- 当您为空的目标组启用慢启动，然后使用单一注册操作注册目标时，这些目标不会进入慢启动模式。仅当至少有一个正常目标未处于慢启动模式时，新注册的目标才会进入慢启动模式。
- 如果您在慢启动模式下取消注册目标，目标将退出慢启动模式。如果您再次注册同一目标，则当目标组认为该目标正常时，它将进入慢启动模式。
- 如果处于慢启动模式的目标变得不正常，则该目标将退出慢启动模式。当目标变得正常时，它将再次进入慢启动模式。
- 使用最少未完成的请求或加权随机路由算法时，无法启用慢启动模式。

Console

更新慢启动持续时间值

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在流量配置窗格中，为慢启动持续时间输入一个新值。要禁用慢启动模式，请输入 0。
6. 选择保存更改。

Amazon CLI

更新慢启动持续时间值

使用带 slow_start.duration_seconds 属性的 modify-target-group-attributes 命令。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=slow_start.duration_seconds,Value=30"
```

CloudFormation

更新慢启动持续时间值

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该slow_start.duration_seconds属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "slow_start.duration_seconds"
          Value: "30"
```

运行状况设置

默认情况下，应用程序负载均衡器会监控目标的运行状况，并将请求路由到运行正常的目标。但如果负载均衡器没有足够运行正常的目标，则会自动将流量发送到所有已注册的目标（失效时开放）。您可以修改目标组的运行状况设置，为 DNS 故障转移和路由故障转移定义阈值。有关更多信息，请参阅 the section called “目标组运行状况”。

Console

修改目标组运行状况设置

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。

4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 检查是开启还是关闭了跨区域负载均衡。根据需要更新此设置，以确保在区域出现故障时您有足够的容量来处理额外流量。
6. 展开 Target group health requirements（目标组运行状况要求）。
7. 对于 Configuration type（配置类型），我们建议您选择 Unified configuration（统一配置），它会为两个操作设置相同的阈值。
8. 对于 Healthy state requirements（运行状况良好状态要求），请执行以下操作之一：
   - 选择 Minimum healthy target count（运行状况良好的目标最低计数），然后输入介于 1 到目标组的最大目标数之间的数字。
   - 选择 Minimum healthy target percentage（运行状况良好的目标最低百分比），然后输入 1 到 100 之间的数字。
9. 选择保存更改。

Amazon CLI

修改目标组运行状况设置

使用 modify-target-group-attributes 命令。以下示例将两个运行状况不佳状态操作的运行状况良好阈值设置为 50%。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \

 "Key=target_group_health.dns_failover.minimum_healthy_targets.percentage,Value=50"
 \

 "Key=target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage,Value=5
```

CloudFormation

要修改目标组的运行状况设置

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源。以下示例将两个运行状况不佳状态操作的运行状况良好阈值设置为 50%。

```
Resources:
```

```
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "target_group_health.dns_failover.minimum_healthy_targets.percentage"
          Value: "50"
        - Key:
"target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage"
          Value: "50"
```

Cross-zone 负载平衡

负载均衡器的节点将来自客户端的请求分配给已注册目标。启用跨可用区负载均衡后，每个负载均衡器节点会在所有已注册可用区中的已注册目标之间分配流量。禁用跨可用区负载均衡后，每个负载均衡器节点会仅在其可用区中的已注册目标之间分配流量。这可能是因为可用区故障域优先于区域性故障域，从而确保运行状况良好可用区不受运行状况不佳可用区的影响，或者改善整体延迟。

对于应用程序负载均衡器，始终在负载均衡器级别启用跨可用区负载均衡，并且无法关闭。对于目标组，默认设置是使用负载均衡器设置，但您可以通过在目标组级别明确关闭跨可用区负载均衡来覆盖默认设置。

注意事项

- 当跨可用区负载均衡关闭时，不支持目标粘性。
- 当跨可用区负载均衡关闭时，不支持 Lambda 函数作为目标。
- 如果任何目标的参数 `AvailabilityZone` 设置为 `all`，则尝试通过 `ModifyTargetGroupAttributes` API 关闭跨可用区负载均衡会导致错误。
- 注册目标时，`AvailabilityZone` 参数是必需的。仅当跨可用区负载均衡关闭时，才允许特定可用区值。否则，该参数将被忽略并视为 `all`。

最佳实践

- 计划在所有可用区中为每个目标组提供足够的目标容量，以供您使用。如果无法为所有参与的可用区规划足够的容量，我们建议您继续启用跨可用区负载均衡。

- 使用多个目标组配置应用程序负载均衡器时，请确保所有目标组都参与配置区域内的相同可用区。这是为了避免在跨可用区负载均衡关闭时可用区为空，因为这会对进入空可用区的所有 HTTP 请求触发 503 错误。
- 请避免创建空子网。应用程序负载均衡器通过 DNS 公开空子网的区域 IP 地址，这会对 HTTP 请求触发 503 错误。
- 在某些情况下，关闭了跨可用区负载均衡的目标组在每个可用区都有足够的计划目标容量，但可用区中的所有目标都变得不正常。当至少有一个目标组包含所有运行状况不佳的目标时，将从 DNS 中删除负载均衡器节点的 IP 地址。在目标组拥有至少一个运行状况良好的目标后，IP 地址将恢复到 DNS。

关闭跨可用区负载均衡

您可以随时对应用程序负载均衡器目标组关闭跨可用区负载均衡。

Console

关闭跨可用区负载均衡

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在目标选择配置窗格中，选择关闭以进行Cross-zone 负载平衡。
6. 选择保存更改。

Amazon CLI

关闭跨可用区负载均衡

使用 modify-target-group-attributes 命令，并将 `load_balancing.cross_zone.enabled` 属性设置为 `false`。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=load_balancing.cross_zone.enabled,Value=false"
```

CloudFormation

关闭跨可用区负载均衡

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该load_balancing.cross_zone.enabled属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "false"
```

启用跨可用区负载均衡

您可以随时对应用程序负载均衡器目标组启用跨可用区负载均衡。目标组级别的跨可用区负载均衡设置会覆盖负载均衡器级别的设置。

Console

关闭跨可用区负载均衡

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在目标选择配置窗格中，选择开以进行Cross-zone 负载平衡。
6. 选择保存更改。

Amazon CLI

开启跨可用区负载均衡

使用 modify-target-group-attributes 命令，并将 load_balancing.cross_zone.enabled 属性设置为 true。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=load_balancing.cross_zone.enabled,Value=true"
```

CloudFormation

开启跨可用区负载均衡

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该load_balancing.cross_zone.enabled属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "true"
```

自动目标权重（ATW）

自动目标权重（ATW）持续监控运行您的应用程序的目标，检测显著性能偏差（称为异常）。ATW 能够通过实时数据异常检测来动态调整路由到目标的流量。

自动目标权重（ATW）会自动对您账户中的每个应用程序负载均衡器执行异常检测。当发现异常目标时，ATW 可以自动尝试通过减少路由的流量来稳定这些目标，这称为异常缓解。ATW 不断优化流量分配，以最大限度地提高每个目标的成功率，同时最大限度地降低目标组的失败率。

注意事项：

- 异常检测目前会监控来自您的目标的 HTTP 5xx 响应代码以及与目标的连接失败。异常检测始终处于开启状态，不能关闭。
- 使用 Lambda 作为目标时，不支持 ATW。

目录

异常检测

ATW 异常检测会监控任何与目标组中其他目标的行为存在显著偏差的目标。这些偏差称为异常，通过比较一个目标的百分比误差与目标组中其他目标的百分比误差来确定。这些错误既可能是连接错误，也可能是 HTTP 错误代码。报告值明显高于对等目标的目标则被视为异常。

异常检测要求目标组中至少有三个运行状况良好的目标。当目标注册到目标组时，必须通过运行状况检查后才能开始接收流量。目标开始接收流量后，ATW 将开始监控目标并持续发布异常检测结果。对于没有异常的目标，异常结果为 `normal`。对于存在异常的目标，异常结果为 `anomalous`。

ATW 异常检测独立于目标组运行状况检查。目标可以通过所有目标组运行状况检查，但仍会因错误率升高而被标记为异常。目标变为异常不会影响其目标组运行状况检查状态。

异常检测状态

您可以查看当前的异常检测状态。有以下可能值：

- `normal`：未检测到异常。
- `anomalous`：检测到异常。

Console

查看异常检测状态

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择目标选项卡。
5. 在已注册目标表中，您可以在异常检测结果列中查看每个目标的异常检测状态。

Amazon CLI

查看异常检测状态

使用 describe-target-health 命令。以下示例显示了指定目标组中每个目标的状态。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
    --include AnomalyDetection
```

异常缓解

ATW 异常缓解会自动将流量从异常目标路由出去，为其提供恢复的机会。

要求

ATW 的异常缓解功能仅在使用加权随机路由算法时可用。

缓解期间：

- ATW 会定期调整路由到异常目标的流量。目前，周期为每五秒一次。
- ATW 会将路由到异常目标的流量减少到执行异常缓解所需的最低量。
- 不再被检测为异常的目标将逐渐获得更多路由到它们的流量，直到它们达到与目标组中其他正常目标的同等水平。

Console

开启异常缓解功能

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在流量配置窗格中，确认为负载均衡算法选择的值是否为加权随机。

   最初选择加权随机算法时，异常检测默认处于开启状态。
6. 在异常缓解下，确保选中开启异常缓解功能。
7. 选择保存更改。

Amazon CLI

开启异常缓解功能

使用带 `load_balancing.algorithm.anomaly_mitigation` 属性的 modify-target-group-attributes 命令。

```
aws elbv2
```

缓解状态

您可以检查 ATW 是否正在对目标执行缓解。有以下可能值：

- `yes`：正在进行缓解。
- `no`：未进行缓解。

Console

查看异常缓解状态

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择目标选项卡。
5. 在已注册目标表中，您可以在有效缓解列中查看每个目标的异常缓解状态。

Amazon CLI

查看异常缓解状态

使用 describe-target-health 命令。以下示例显示了指定目标组中每个目标的状态。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
    --include AnomalyDetection
```

粘性会话

默认情况下，Application Load Balancer 会根据选定负载均衡算法将每项请求单独路由到已注册的目标。但是，您可以使用粘性会话功能（也称为会话关联），使负载均衡器能够将用户会话绑定到特定的

目标。这可确保在会话期间将来自用户的所有请求发送到同一目标中。此功能对于维护状态信息以便为客户端提供持续体验的服务器非常有用。要使用粘性会话，客户端必须支持 Cookie。

Application Load Balancer 支持基于持续时间的 Cookie 和基于应用程序的 Cookie。粘性会话会在目标组级别启用。您可以在目标组中组合使用基于持续时间的粘性、基于应用程序的粘性以及非粘性。

管理粘性会话的关键是确定负载均衡器一致地将用户请求路由到同一目标的时间长短。如果您的应用程序拥有自己的会话 Cookie，则可以使用基于应用程序的粘性，且负载均衡器会话 Cookie 将会遵循应用程序会话 Cookie 指定的持续时间。如果您的应用程序没有自己的会话 Cookie，则可以使用基于持续时间的粘性来生成具有您指定持续时间的负载均衡器会话 Cookie。

负载均衡器生成的 Cookie 内容使用轮换密钥加密。无法解密或修改负载均衡器生成的 Cookie。

对于这两种粘性类型，Application Load Balancer 将重置每次请求后生成的 Cookie 的过期期限。如果 Cookie 过期，会话将不再具有粘性，客户端应该从 Cookie 存放区删除 Cookie。

要求

- HTTP/HTTPS 负载均衡器。
- 每个可用区内至少有一个运行状况良好的实例。

注意事项

- 如果跨可用区负载均衡禁用，则不支持粘滞会话。禁用跨可用区负载均衡期间启用粘性会话的尝试将会失败。
- 对于基于应用程序的 Cookie，每个目标组必须单独指定 Cookie 名称。但是，对于基于持续时间的 Cookie，所有目标组将使用 AWSALB 作为唯一的名称。
- 如果您使用的是多层 Application Load Balancer，则可以使用基于应用程序的 Cookie 在所有层启用粘性会话。但是，如果使用基于持续时间的 Cookie，您就只能在一个层上启用粘性会话，因为 AWSALB 是唯一可用的名称。
- 如果应用程序负载均衡器同时收到 AWSALBCORS 和 AWSALB 基于持续时间的粘性 Cookie，则 AWSALBCORS 中的值将优先。
- Application-based 粘性不适用于加权目标群体。
- 如果您具有一个包含多个目标组的转发操作，并且一个或多个目标组已启用了粘性会话，则必须在目标组级别启用粘性。
- WebSocket 连接本质上是粘性的。如果客户端请求升级连接 WebSockets，则返回接受连接升级的 HTTP 101 状态码的目标就是 WebSockets连接中使用的目标。 WebSockets 升级完成后，将不使用基于 Cookie 的粘性。

- Application Load Balancer 使用 Cookie 标头中的 `Expires` 属性而不是 `Max-Age` 属性。
- Application Load Balancer 不支持 URL 编码的 Cookie 值。
- 如果应用程序负载均衡器在目标因注销而耗尽时收到新请求，则该请求将被路由到运行正常的目标。
- 如果启用了目标优化器，则不支持粘性会话。

粘性类型

- Duration-based 粘性
- Application-based 粘性

Duration-based 粘性

Duration-based stickiness 使用负载均衡器生成的 cookie () `AWSALB` 将请求路由到目标组中的同一目标。Cookie 用于将会话映射到目标。如果您的应用程序没有自己的会话 Cookie，您可以指定自己的粘性持续时间，并管理负载均衡器一致地将用户请求路由到同一目标的时间长短。

当负载均衡器第一次收到来自客户端的请求时，它会（根据选定算法）将请求路由到目标并生成名为 `AWSALB` 的 Cookie。它还会对选定目标的有关信息进行编码，加密 Cookie，并在对客户端的响应中包含 Cookie。负载平衡器生成的 Cookie 具有自身的 7 天到期时间，且不可配置。

在后续请求中，客户端应包含 `AWSALB` Cookie。当负载均衡器收到来自包含 Cookie 的客户端的请求时，它会检测到该请求并将请求路由到同一目标。如果 Cookie 存在但无法解码，或者指向某个已注销或运行不正常的目标，则负载均衡器会选择一个新目标并使用有关新目标的信息来更新 Cookie。

对于跨源资源共享（CORS）请求，某些浏览器需要 `SameSite=None; Secure` 来启用粘性。为了支持这些浏览器，负载均衡器始终会生成第二个粘性 Cookie `AWSALBCORS`（其中包含与原始粘性 Cookie 相同的信息）和 `SameSite` 属性。客户端会接收两种 cookie，包括非 CORS 请求。

Console

启用基于持续时间的粘性

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在目标选择配置下，执行以下操作：

a. 选择开启粘性。
b. 对于 Stickiness type（Stickiness 类型），请选择 Load balancer generated cookie（负载均衡器生成的 Cookie）。
c. 对于 Stickiness duration，指定一个介于 1 秒和 7 天之间的值。

6. 选择保存更改。

Amazon CLI

启用基于持续时间的粘性

使用带 `stickiness.enabled` 和 `stickiness.lb_cookie.duration_seconds` 属性的 modify-target-group-attributes 命令。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \
        "Key=stickiness.enabled,Value=true" \
        "Key=stickiness.lb_cookie.duration_seconds,Value=300"
```

CloudFormation

启用基于持续时间的粘性

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含`stickiness.enabled`和`stickiness.lb_cookie.duration_seconds`属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "stickiness.enabled"
          Value: "true"
        - Key: "stickiness.lb_cookie.duration_seconds"
          Value: "300"
```

Application-based 粘性

Application-based 粘性使您可以灵活地为客户目标粘性设置自己的标准。启用基于应用程序的粘性时，负载均衡器会根据选定算法将第一个请求路由到目标组内的目标。为启用粘性，目标应设置与负载均衡器上配置的 Cookie 匹配的自定义应用程序 Cookie。此自定义 Cookie 可包含应用程序所需的任何 Cookie 属性。

当 Application Load Balancer 收到来自目标的自定义应用程序 Cookie 时，它会自动生成新加密的应用程序 Cookie，以捕获粘性信息。此负载均衡器生成的应用程序 Cookie 可为每个启用基于应用程序的粘性的目标组捕获粘性信息。

负载均衡器生成的应用程序 Cookie 不会复制目标设置的自定义 Cookie 的属性。它自己的过期期限为 7 天，这是不可配置的。在对客户端的响应中，Application Load Balancer 仅验证在目标组级别配置的自定义 Cookie 的名称，而不验证自定义 Cookie 的值或过期属性。只要名称匹配，负载均衡器即会发送 Cookie、目标设置的自定义 Cookie 以及负载均衡器生成的应用程序 Cookie，来响应客户端。

在后续请求中，客户端必须将两个 Cookie 发送回以保持粘性。负载均衡器会解密应用程序 Cookie，并检查配置的粘性持续时间是否仍然有效。然后，它将使用 Cookie 中的信息将请求发送到目标组中的同一目标，以保持粘性。负载均衡器还会将自定义应用程序 Cookie 代理到目标，而不检查或修改它。在后续响应中，会对在负载均衡器上配置的负载均衡器生成的应用程序 Cookie 的到期期限和粘性持续时间进行重置。为了保持客户端和目标之间的粘性，Cookie 的到期期限和粘性持续时间不会消失。

如果目标失败或者目标运行状况不佳，负载均衡器会停止将请求路由到该目标，并根据选定负载均衡算法来选择新的运行状况良好的目标。现在，负载均衡器将会话视为正在“附加”到新的正常运行目标，并继续将请求路由至新的正常运行目标，即使之前失败的目标已恢复正常运行。

对于跨源资源共享 (CORS) 请求，只有当用户代理版本为 Chromium80 或更高版本时，负载均衡器才会将 `SameSite=None; Secure` 属性添加到负载均衡器生成的应用程序 Cookie 来启用粘性。

由于大多数浏览器将 Cookie 的大小限制为 4K，因此负载均衡器会将大于 4K 的应用程序 Cookie 分片为多个 Cookie。Application Load Balancer 支持最大 16K 的 Cookie，因此最多可以创建 4 个分片发送给客户端。客户端看到的应用程序 Cookie 名称以“AWSALBAPP-”开头，并包含片段编号。例如，如果 Cookie 大小为 0-4K，则客户端会看到 AWSALBAPP-0。如果 Cookie 大小为 4-8k，则客户端会看到 AWSALBAPP-0 和 AWSALBAPP-1，依此类推。

Console

启用基于应用程序的粘性

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 在目标选择配置下，执行以下操作：

   a. 选择开启粘性。

   b. 对于粘性类型，请选择 Application-based Cookie。

   c. 对于 Stickiness duration，指定一个介于 1 秒和 7 天之间的值。

   d. 对于 App cookie name（应用程序 Cookie 名称），请输入基于应用程序的 Cookie 名称。

      请勿使用 `AWSALB`、`AWSALBAPP`、或 `AWSALBTG` 作为 Cookie 名称；它们将保留以供负载均衡器使用。

6. 选择保存更改。

Amazon CLI

启用基于应用程序的粘性

使用带有以下属性的 modify-target-group-attributes 命令：

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \
        "Key=stickiness.enabled,Value=true" \
        "Key=stickiness.type,Value=app_cookie" \
        "Key=stickiness.app_cookie.cookie_name,Value=my-cookie-name" \
        "Key=stickiness.app_cookie.duration_seconds,Value=300"
```

CloudFormation

启用基于应用程序的粘性

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含以下属性：

- stickiness.enabled
- stickiness.type
- stickiness.app_cookie.cookie_name
- stickiness.app_cookie.duration_seconds

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "stickiness.enabled"
          Value: "true"
        - Key: "stickiness.type"
          Value: "app_cookie"
        - Key: "stickiness.app_cookie.cookie_name"
          Value: "my-cookie-name"
        - Key: "stickiness.app_cookie.duration_seconds"
          Value: "300"
```

手动再平衡

向上扩展时，如果目标数量大幅度增加，则可能由于粘性而导致负载分配不均。在这种情况下，您可以使用以下两种方式再平衡目标上的负载：

- 将应用程序生成的 Cookie 的有效期设置为在当前日期和时间之前。这样可以阻止客户端将 Cookie 发送到应用程序负载均衡器，后者将重启建立粘性的过程。
- 为负载均衡器基于应用程序的粘性配置设置一个较短的持续时间，例如 1 秒。这样将强制应用程序负载均衡器重新建立粘性，即使目标设置的 Cookie 尚未过期。

WAF HTTP/2 流量检查行为

默认情况下，当你将 Application Load Balancer 与集成时 Amazon WAF， Amazon WAF 会使用可用的请求数据立即检查 HTTP/2 请求。此检查模式支持双向流媒体应用程序，在这种应用程序中，服务器必须在客户端完成发送请求数据之前做出响应，从而防止在实时通信场景中出现超时。

您可以将 Application Load Balancer 配置为在 Amazon WAF 执行检查之前累积 HTTP/2 数据帧。此配置有助于防止攻击者在多个 HTTP/2 数据帧中分割恶意负载时绕过安全漏洞。此选项用于标准请求-响应应用程序，在这种应用程序中，客户端在预期响应之前传输完整的请求数据，从而确保 Amazon WAF 检查完整的请求负载。

Console

配置 WAF HTTP/2 流量检查行为

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 对于 WAF HTTP/2 流量检查行为，请选择检查模式。
6. 选择保存更改。

Amazon CLI

配置 WAF HTTP/2 流量检查行为

使用带 `waf.http2.traffic_inspection_behavior` 属性的 modify-target-group-attributes 命令。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes
 "Key=waf.http2.traffic_inspection_behavior,Value=inspect_after_sufficient_data"
```

Amazon CloudFormation

配置 WAF HTTP/2 流量检查行为

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该`waf.http2.traffic_inspection_behavior`属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      ProtocolVersion: HTTP2
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "waf.http2.traffic_inspection_behavior"
          Value: "inspect_after_sufficient_data"
```

向应用程序负载均衡器目标组注册目标

将目标注册到目标组。在创建目标组时，指定其目标类型，此类型将确定您如何注册其目标。例如，您可以注册实例 ID、IP 地址或 Lambda 函数。有关更多信息，请参阅 Application Load Balancer 的目标组。

如果对当前注册目标的需求增加，则可以注册其他目标来处理该需求。在目标准备好处理请求后，将目标注册到您的目标组。只要注册过程完成且目标通过初始运行状况检查，负载均衡器就会开始将请求路由至目标。

如果已注册目标需求减少或者您需要为目标提供服务，您可以从目标组取消注册目标。取消注册某个目标后，负载均衡器立即停止将请求路由到该目标。在目标准备好接收请求时，您可以再次将目标注册到目标组。

在取消注册目标时，负载均衡器会一直等待，直到进行中的请求完成。这称作连接耗尽。在连接耗尽期间，目标的状态为 `draining`。

取消注册通过 IP 地址注册的目标后，必须等待取消注册延迟结束，然后才可以重新注册相同的 IP 地址。

如果要通过实例 ID 来注册目标，则可以将负载均衡器与 Auto Scaling 组一同使用。将目标组挂接到 Auto Scaling 组并且该组扩展后，由 Auto Scaling 组启动的实例将自动在目标组中注册。如果您将目

标组与 Auto Scaling 组分离，则实例会自动从目标组中取消注册。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的将负载均衡器挂接到自动扩缩组。

关闭目标上的应用程序时，必须先从目标组中注销该目标，并留出时间让现有连接耗尽。您可以使用 describe-target-health CLI 命令或通过在 Amazon Web Services 管理控制台中刷新目标组视图来监控注销状态。确认目标已注销后，您可以继续停止或终止应用程序。这一顺序可防止用户在应用程序仍在处理流量过程中被终止时遇到 5XX 错误。

目标安全组

在将 EC2 实例注册为目标时，您必须确保实例的安全组允许负载均衡器在侦听器端口和运行状况检查端口上与您的实例进行通信。

推荐的规则

Inbound

Source	Port Range	Comment
load balancer security group	*instance listener*	在实例侦听器端口上允许来自负载均衡器的流量
load balancer security group	*health check*	在运行状况检查端口上允许来自负载均衡器的流量

我们还建议您允许入站 ICMP 流量以支持路径 MTU 发现。有关更多信息，请参阅《Amazon EC2 用户指南》中的路径 MTU 发现。

目标优化器

目标优化器允许您对目标组中的目标强制执行严格的并发。它需要在目标上安装和配置的代理的帮助下运行。该代理充当负载均衡器和您的应用程序之间的内联代理。您可以将代理配置为强制执行负载均衡器可以向目标发送的最大并发请求数。代理跟踪目标正在处理的请求数量。当该数字低于配置的最大值时，代理会向负载均衡器发送信号，让其知道目标已准备好处理另一个请求。

要启用目标优化器，请在创建目标组时指定目标控制端口。负载均衡器通过代理在该端口上建立控制通道，用于管理流量。此端口不同于负载均衡器发送应用程序流量的端口。在目标组中注册的目标必须在其上运行代理。

注意：目标优化器只能在创建目标组期间启用。目标控制端口创建后无法修改。

该代理以 Docker 镜像的形式提供，网址为：public.ecr.aws/aws-elb/target-optimizer/target-control-agent:latest。在运行代理容器时，您可以配置以下环境变量：

TARGET_CONTROL_DATA_ADDRESS

代理从该套接字上的负载均衡器接收应用程序流量 (IP:port)。此套接字中的端口是您为目标组配置的应用程序流量端口。默认情况下，代理可以接受纯文本连接和 TLS 连接。

TARGET_CONTROL_CONTROL_ADDRESS

代理从该套接字上的负载均衡器接收管理流量 (IP:port)。套接字中的端口是您为目标组配置的目标控制端口。

TARGET_CONTROL_DESTINATION_ADDRESS

代理代理将应用程序流量代理到此套接字 (IP:port)。你的应用程序应该在这个套接字上监听。

（可选）TARGET_CONTROL_MAX_CONCURRENCY

目标将从负载均衡器接收的最大并发请求数。它可以介于 0-1000 之间。默认为 1。

（可选）TARGET_CONTROL_TLS_CERT_PATH

代理在 TLS 握手期间向负载均衡器提供的 TLS 证书的位置。默认情况下，代理会在内存中生成自签名证书。

（可选）TARGET_CONTROL_TLS_KEY_PATH

与 TLS 握手期间代理向负载均衡器提供的 TLS 证书相对应的私钥的位置。默认情况下，代理会在内存中生成私钥。

（可选）TARGET_CONTROL_TLS_SECURITY_POLICY

您为目标组配置的 ELB 安全策略。默认值为 ELBSecurityPolicy-2016-08。

（可选）TARGET_CONTROL_PROTOCOL_VERSION

负载均衡器与代理通信所使用的协议。可能的值为 HTTP1、 HTTP2、 GRPC。默认值为 HTTP1。

（可选）RUST_LOG

代理进程的日志级别。代理软件是用 Rust 编写的。可能的值为 debug info、和 error。默认值为 info。

要修改任何环境变量的值，必须使用新值重新启动代理。您可以使用以下指标监控目标优化器：TargetControlRequestCount、、TargetControlRequestRejectCount、TargetControlAct

TargetControlWorkQueueLength、TargetControlProcessedBytes。有关更多信息，请参阅目标优化器指标。有关疑难解答信息，请参阅目标优化器疑难解答

共享子网

参与者能够在共享 VPC 中创建应用程序负载均衡器。参与者不能注册在未与他们共享的子网中运行的目标。

注册目标

每个目标组在为负载均衡器启用的每个可用区中必须至少有一个已注册目标。

您的目标组的目标类型将确定如何向该目标组注册目标。有关更多信息，请参阅 Target type。

要求和注意事项

- 当您注册实例时，实例必须处于 running 状态。
- 目标实例必须位于您为目标组指定的虚拟私有云（VPC）中。
- 当按实例 ID 为 IPv6 目标组注册目标时，必须为目标分配主 IPv6 地址。要了解更多信息，请参阅《Amazon EC2 用户指南》中的 IPv6 地址
- 在为 IPv4 目标组通过 IP 地址注册目标时，您注册的 IP 地址必须来自以下 CIDR 数据块之一：
  - 目标组的 VPC 的子网
  - 10.0.0。 0/8（RFC 1918）
  - 100.64.0。 0/10 (RFC 6598)
  - 172.16.0。 0/12（RFC 1918）
  - 192.168.0。 0/16（RFC 1918）
- 在为 IPv6 目标组按 IP 地址注册目标时，您注册的 IP 地址必须位于 VPC 的 IPv6 CIDR 数据块内，或位于已建立对等连接的 VPC 的 IPv6 CIDR 数据块内。
- 不能注册同一 VPC 中其他应用程序负载均衡器的 IP 地址。如果另一个 Application Load Balancer 位于与负载均衡器 VPC 对等的 VPC 中，则可以注册其 IP 地址。

Console

要注册目标

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。

3. 选择目标组的名称以打开其详细信息页面。
4. 选择目标选项卡。
5. 选择注册目标。
6. 如果目标组的目标类型是 `instance`，则选择可用的实例，根据需要覆盖默认端口，然后选择包含如下待处理事项。
7. 如果目标组的目标类型为 `ip`，则需为每个 IP 地址选择该网络，输入 IP 地址及端口，然后选择包含如下待处理事项。
8. 如果目标组的目标类型为 `lambda`，请选择该 Lambda 函数或输入其 ARN。有关更多信息，请参阅 使用 Lambda 函数作为目标。
9. 选择注册待处理目标。

Amazon CLI

要注册目标

使用 register-targets 命令。以下示例通过实例 ID 注册目标。由于未指定端口，负载均衡器将使用目标组端口。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

以下示例通过 IP 地址注册目标。由于未指定端口，负载均衡器将使用目标组端口。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=10.0.50.10 Id=10.0.50.20
```

以下示例将一个 Lambda 函数注册为目标。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

CloudFormation

注册目标

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含新目标。以下示例按实例 ID 注册了两个目标。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
        - Id: !GetAtt Instance2.InstanceId
          Port: 80
```

取消注册目标

如果应用程序需求减少或者您需要为目标提供服务，您可以从目标组取消注册目标。取消注册目标将从目标组中删除目标，但不会影响目标。

Console

注销目标

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在目标选项卡中，选择要删除的目标。
5. 选择注销。
6. 当系统提示您确认时，选择 Deregister (取消注册)。

Amazon CLI

注销目标

使用 deregister-targets 命令。以下示例会注销两个通过实例 ID 注册的目标。

```
aws elbv2 deregister-targets \
    --target-group-arn target-group-arn \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

使用 Lambda 函数作为应用程序负载均衡器的目标

您可以将 Lambda 函数注册为目标并将侦听器规则配置为将请求转发到 Lambda 函数的目标组。当负载均衡器将请求转发到目标组并使用 Lambda 函数作为目标时，它会调用 Lambda 函数并以 JSON 格式将请求内容传递到 Lambda 函数。

负载均衡器会直接调用 Lambda 函数，而不使用网络连接。因此，对应用程序负载均衡器安全组的出站规则没有要求。

限制

- Lambda 函数和目标组必须位于同一账户中，且位于同一区域中。
- 您可以发送到 Lambda 函数的请求正文的最大大小为 1 MB。有关相关大小限制，请参阅 HTTP 标头限制。
- Lambda 函数可以发送的响应 JSON 的最大大小为 1 MB。
- WebSockets 不支持。升级请求被拒绝，并显示 HTTP 400 代码。
- 不支持本地区域。
- 不支持自动目标权重（ATW）。

内容

有关演示，请参阅 Application Load Balancer 上的 Lambda 目标。

准备 Lambda 函数

如果您将 Lambda 函数与 Application Load Balancer 一起使用，则以下建议适用。

调用 Lambda 函数的权限

如果使用 Amazon Web Services 管理控制台创建目标组并注册 Lambda 函数，控制台会代表您将所需的权限添加到 Lambda 函数策略。否则，在创建目标组并使用注册函数后，必须使用添加权限命令授予 Elastic Load Balancing 调用您的 Lambda 函数的权限。 Amazon CLI我们建议您使用 `aws:SourceAccount` 和 `aws:SourceArn` 条件键限制对指定目标组的函数调用。有关更多信息，请参阅 IAM 用户指南中的混淆代理人问题。

```
aws lambda add-permission \
    --function-name lambda-function-arn-with-alias-name \
    --statement-id elb1 \
    --principal elasticloadbalancing.amazonaws.com \
    --action lambda:InvokeFunction \
    --source-arn target-group-arn \
    --source-account target-group-account-id
```

Lambda 函数版本控制

您可以为每个目标组注册一个 Lambda 函数。为确保您可以更改 Lambda 函数并且负载均衡器始终调用当前版本的 Lambda 函数，请在向负载均衡器注册 Lambda 函数时创建一个函数别名并在函数 ARN 中包含该别名。有关更多信息，请参阅《Amazon Lambda 开发人员指南》中的 Amazon Lambda 函数别名。

函数超时

负载均衡器会一直等待，直到您的 Lambda 函数响应或超时。建议您根据预期运行时间配置 Lambda 函数的超时。有关默认超时值以及如何更改该值的信息，请参阅配置 Lambda 函数超时。有关可配置的最大超时值的信息，请参阅 Amazon Lambda 限额。

为 Lambda 函数创建目标组

创建一个要在请求路由中使用的目标组。如果请求内容与侦听器规则匹配并且具有将该内容转发到此目标组的操作，则负载均衡器会调用已注册的 Lambda 函数。

Console

创建目标组并注册 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的 Load Balancing (负载均衡) 下，选择 Target Groups (目标组)。
3. 选择 Create target group (创建目标组)。
4. 对于选择目标类型，选择 Lambda 函数。
5. 对于目标组名称，输入目标组的名称。
6. （可选）要启用运行状况检查，请在 Health checks（运行状况检查）部分中选择 Enable（启用）。
7. （可选）展开标签。对于每个标签，请选择添加新标签，然后输入标签键和标签值。
8. 选择下一步。
9. 如果您已准备好注册该 Lambda 函数，请选中选择一个 Lambda 函数并从列表中选择该 Lambda 函数，或者选择输入 Lambda 函数 ARN 并输入该 Lambda 函数的 ARN，

   如果您还没有准备好注册该 Lambda 函数，请选择稍后注册 Lambda 函数，以后再注册目标。有关更多信息，请参阅 the section called “注册目标”。
10. 选择创建目标组。

Amazon CLI

创建目标群组类型为 lambda

使用 create-target-group 命令。

```
aws elbv2 create-target-group \
    --name my-target-group \
    --target-type lambda
```

注册 Lambda 函数

使用 register-targets 命令。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

CloudFormation

创建目标组并注册 Lambda 函数

定义类型为 AWS::ElasticLoadBalancingV2:: TargetGroup 的资源。如果您现在还没有准备好注册该 Lambda 函数，则可以省略 Targets 属性，以后再添加。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
```

从负载均衡器接收事件

负载均衡器支持通过 HTTP 和 HTTPS 进行请求的 Lambda 调用。负载均衡器采用 JSON 格式发送事件。负载均衡器将以下标头添加到每个请求：X-Amzn-Trace-Id、X-Forwarded-For、X-Forwarded-Port 和 X-Forwarded-Proto。

如果 content-encoding 标头存在，负载均衡器会对正文进行 Base64 编码并将 isBase64Encoded 设置为 true。

如果 content-encoding 标头不存在，Base64 编码取决于内容类型。对于以下类型，负载均衡器按原样发送正文并将其设置isBase64Encoded为false：text/*、 application/json application/javascript、和。 application/xml否则，负载均衡器会对正文进行 Base64 编码并将 isBase64Encoded 设置为 true。

以下是示例事件。

```
{
    "requestContext": {
        "elb": {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
group/6d0ecf831eec9f09"
```

```
        }
    },
    "httpMethod": "GET",
    "path": "/",
    "queryStringParameters": {parameters},
    "headers": {
        "accept": "text/html,application/xhtml+xml",
        "accept-language": "en-US,en;q=0.8",
        "content-type": "text/plain",
        "cookie": "cookies",
        "host": "lambda-846800462-us-east-2.elb.amazonaws.com",
        "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)",
        "x-amzn-trace-id": "Root=1-5bdb40ca-556d8b0c50dc66f0511bf520",
        "x-forwarded-for": "72.21.198.66",
        "x-forwarded-port": "443",
        "x-forwarded-proto": "https"
    },
    "isBase64Encoded": false,
    "body": "request_body"
}
```

响应负载均衡器

来自 Lambda 函数的响应必须包含 Base64 编码状态、状态代码和标头。您可以省略正文。

要在响应的正文中包含二进制内容，您必须对内容进行 Base64 编码并将 `isBase64Encoded` 设置为 `true`。负载均衡器解码内容以检索二进制内容并将该内容发送到 HTTP 响应的正文中的客户端。

负载均衡器不会遵守逐跳标头，如 `Connection` 或 `Transfer-Encoding`。您可以省略 `Content-Length` 标头，因为负载均衡器会在将响应发送到客户端之前计算它。

以下是来自基于 nodejs 的 Lambda 函数的示例响应。

```
{
    "isBase64Encoded": false,
    "statusCode": 200,
    "statusDescription": "200 OK",
    "headers": {
        "Set-cookie": "cookies",
        "Content-Type": "application/json"
    },
    "body": "Hello from Lambda (optional)"
```

```
}
```

有关与 Application Load Balancer 结合使用的 Lambda 函数模板，请参阅 github 上的 application-load-balancer-serverless-app。或者，打开 Lambda 控制台，依次选择应用程序、创建应用程序，然后从 Amazon Serverless Application Repository 中选择下列项目之一：

- ALB-Lambda-Target-UploadFiletoS3
- ALB-Lambda-Target-BinaryResponse
- ALB-Lambda-Target-WhatisMyIP

Multi-value 标题

如果来自客户端的请求或来自 Lambda 函数的响应包含具有多个值的标头或多次包含同一标头，或包含同一键具有多个值的查询参数，则可启用对多值标头语法的支持。启用多值标头后，负载均衡器和 Lambda 函数之间交换的标头和查询参数将使用数组而不是字符串。如果您没有启用多值标头语法，并且标头或查询参数具有多个值，则负载均衡器将使用其收到的最后一个值。

目录

包含多值标头的请求

用于标头和查询字符串参数的字段名称根据是否为目标组启用了多值标头而有所不同。

以下示例请求具有两个查询参数和同一个键：

```
http://www.example.com?&myKey=val1&myKey=val2
```

对于默认格式，负载均衡器将使用客户端发送的最后一个值，并使用 `queryStringParameters` 向您发送包含查询字符串参数的事件。例如：

```
"queryStringParameters": { "myKey": "val2"},
```

如果启用了多值标头，则负载平衡器将使用客户端发送的两个键值，并使用 `multiValueQueryStringParameters` 向您发送包含查询字符串参数的事件。例如：

```
"multiValueQueryStringParameters": { "myKey": ["val1", "val2"] },
```

同样，假设客户端发送标头中包含两个 Cookie 的请求：

```
"cookie": "name1=value1",
"cookie": "name2=value2",
```

对于默认格式，负载均衡器将使用客户端发送的最后一个 Cookie，并使用 `headers` 向您发送包含标头的事件。例如：

```
"headers": {
    "cookie": "name2=value2",
    ...
},
```

如果启用了多值标头，负载均衡器将使用客户端发送的两个 Cookie 并使用 `multiValueHeaders` 向您发送包含标头的事件。例如：

```
"multiValueHeaders": {
    "cookie": ["name1=value1", "name2=value2"],
    ...
},
```

如果查询参数是 URL-encoded，则负载均衡器不会对其进行解码。您必须在 Lambda 函数中对它们进行解码。

包含多值标头的响应

用于标头的字段名称根据是否为目标组启用了多值标头而有所不同。如果启用了多值标头，则必须使用 `multiValueHeaders`，否则使用 `headers`。

对于默认格式，您可以指定单个 Cookie：

```
{
  "headers": {
      "Set-cookie": "cookie-name=cookie-value;Domain=myweb.com;Secure;HttpOnly",
      "Content-Type": "application/json"
  },
}
```

如果启用了多值标头，您必须指定多个 Cookie，如下所示：

```
{
  "multiValueHeaders": {
      "Set-cookie": ["cookie-name=cookie-
value;Domain=myweb.com;Secure;HttpOnly","cookie-name=cookie-value;Expires=May 8,
 2019"],
      "Content-Type": ["application/json"]
  },
}
```

负载均衡器向客户端发送标头时所遵循的顺序可能与 Lambda 响应负载中指定的顺序不同。因此，不要指望按特定顺序返回标头。

启用多值标头

您可以对目标类型为 `lambda` 的目标组启用或禁用多值标头。

Console

启用多值标头

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Attributes（属性）选项卡上，选择 Edit（编辑）。
5. 启用多值标头。
6. 选择保存更改。

Amazon CLI

启用多值标头

使用带 `lambda.multi_value_headers.enabled` 属性的 modify-target-group-attributes 命令。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=lambda.multi_value_headers.enabled,Value=true"
```

CloudFormation

启用多值标头

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该`lambda.multi_value_headers.enabled`属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
      TargetGroupAttributes:
        - Key: "lambda.multi_value_headers.enabled"
          Value: "true"
```

启用运行状况检查

默认情况下，对类型为 `lambda` 的目标组禁用运行状况检查。您可以启用运行状况检查以通过 Amazon Route 53 实现 DNS 故障转移。Lambda 函数可以在响应运行状况检查请求之前检查下游服务的运行状况。如果来自 Lambda 函数的响应指示运行状况检查失败，则运行状况检查失败会传递到 Route 53。您可以将 Route 53 配置为故障转移到备份应用程序堆栈。

您需要支付运行状况检查的费用，就像您支付任何 Lambda 函数调用的费用一样。

以下是发送到您的 Lambda 函数的运行状况检查事件的格式。要检查事件是否为运行状况检查事件，请检查 user-agent 字段的值。运行状况检查的用户代理为 `ELB-HealthChecker/2.0`。

```
{
    "requestContext": {
        "elb": {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
group/6d0ecf831eec9f09"
```

```
        }
    },
    "httpMethod": "GET",
    "path": "/",
    "queryStringParameters": {},
    "headers": {
        "user-agent": "ELB-HealthChecker/2.0"
    },
    "body": "",
    "isBase64Encoded": false
}
```

Console

为启用运行状况检查 lambda 目标组

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Health checks 选项卡上，选择 Edit。
5. 对于运行状况检查，选择启用。
6. （可选）根据需要更新运行状况检查设置。
7. 选择保存更改。

Amazon CLI

为启用运行状况检查 lambda 目标组

使用 modify-target-group 命令。

```
aws elbv2 modify-target-group \
    --target-group-arn target-group-arn \
    --health-check-enabled
```

CloudFormation

为启用运行状况检查 lambda 目标组

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      HealthCheckEnabled: true
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
```

注册 Lambda 函数

您可以向每个目标组注册单个 Lambda 函数。要替换 Lambda 函数，建议您创建一个新的目标组，将新函数注册到该新目标组，然后将侦听器规则更新为使用新目标组。

Console

注册 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在目标选项卡中，如果未注册任何 Lambda 函数，请选择注册目标。
5. 选择该 Lambda 函数或输入其 ARN。
6. 选择注册。

Amazon CLI

注册 Lambda 函数

使用 register-targets 命令。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

CloudFormation

注册 Lambda 函数

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
```

注销 Lambda 函数

如果您不再需要向您的 Lambda 函数发送流量，则可以将其取消注册。在取消注册 Lambda 函数后，进行中的请求会失败，并显示 HTTP 5XX 错误。

要替换 Lambda 函数，建议您创建一个新的目标组，将新函数注册到该新目标组，然后将侦听器规则更新为使用新目标组。

Console

注销 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在目标选项卡上，选择目标并选择注销。
5. 当系统提示您确认时，选择 Deregister (取消注册)。

Amazon CLI

注销 Lambda 函数

使用 deregister-targets 命令。

```
aws elbv2 deregister-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

应用程序负载均衡器目标组的标签

标签有助于按各种标准 (例如，用途、所有者或环境) 对目标组进行分类。

您可以为每个目标组添加多个标签。每个目标组的标签键必须是唯一的。如果您添加的标签中的键已经与目标组关联，它将更新该标签的值。

用完标签后可以将其删除。

限制

- 每个资源的标签数上限 – 50
- 最大密钥长度 - 127 个 Unicode 字符
- 最大值长度 - 255 个 Unicode 字符
- 标签键和值区分大小写。允许的字符包括可表示的字母、空格和数字 UTF-8，以及以下特殊字符：+-=。_ : / @。请不要使用前导空格或尾随空格。
- 请勿在标签名称或值中使用aws:前缀，因为它已保留供 Amazon 使用。您无法编辑或删除带此前缀的标签名称或值。具有此前缀的标签不计入每个资源的标签数限制。

Console

要管理目标组的标签

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在标签选项卡上，选择管理标签，然后执行以下一项或多项操作：
   a. 要更新标签，请为键和值输入新值。
   b. 要添加标签，请选择添加标签，然后为键和值输入值。

c. 要删除标签，请选择标签旁边的删除。

5. 选择保存更改。

Amazon CLI

添加 标签

使用 add-tags 命令。以下示例将添加两个标签。

```
aws elbv2 add-tags \
    --resource-arns target-group-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

删除标签

使用 remove-tags 命令。以下示例将移除具有指定键的标签。

```
aws elbv2 remove-tags \
    --resource-arns target-group-arn \
    --tag-keys project department
```

CloudFormation

添加 标签

更新 AWS::ElasticLoadBalancingV2:: TargetGroup 资源以包含该Tags属性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

删除应用程序负载均衡器目标组

如果目标组未由任何侦听器规则的转发操作引用，则可以删除该目标组。删除目标组不会影响已注册到目标组的目标。如果您不再需要已注册的 EC2 实例，则可以停止或终止该实例。

Console

删除目标组

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的负载均衡下，选择目标组。
3. 选择目标组，然后依次选择操作、删除。
4. 选择删除。

Amazon CLI

删除目标组

使用 delete-target-group 命令。

```
aws elbv2 delete-target-group \
    --target-group-arn target-group-arn
```

监控 Application Load Balancer

您可使用以下功能监控负载均衡器，分析流量模式及解决与负载均衡器和目标相关的问题。

CloudWatch 指标

您可以使用 Amazon CloudWatch 以一组有序的时间序列数据（称为指标）的形式检索有关负载均衡器和目标的数据点的统计数据。您可使用这些指标来验证您的系统是否按预期运行。有关更多信息，请参阅 CloudWatch Application Load Balancer 的指标。

访问日志

您可以使用访问日志来捕获有关向负载均衡器发出的请求的详细信息，并将这些详细信息作为日志文件存储在 Amazon S3 中。您可以使用这些访问日志分析流量模式并解决与目标相关的问题。有关更多信息，请参阅 Application Load Balancer 的访问日志。

连接日志

您可以使用连接日志捕获有关发送到负载均衡器的请求的属性，并将其作为日志文件存储在 Amazon S3 中。您可以使用这些连接日志来确定客户端 IP 地址和端口、客户端证书信息、连接结果以及正在使用的 TLS 密码。然后可以使用这些连接日志来查看请求模式和其他趋势。有关更多信息，请参阅 应用程序负载均衡器的连接日志。

Health 检查日志

您可以使用运行状况检查日志来捕获有关对负载均衡器的注册目标进行的运行状况检查的详细信息，并将其作为日志文件存储在 Amazon S3 中。您可以使用这些运行状况检查日志来解决目标的问题。有关更多信息，请参阅 Health 检查日志。

请求跟踪

您可以使用来请求跟踪来跟踪 HTTP 请求。负载均衡器会为它接收的每个请求添加一个包含跟踪标识符的标头。有关更多信息，请参阅 请求 Application Load Balancer 的跟踪。

CloudTrail 日志

您可以使用 Amazon CloudTrail 捕获有关对 Elastic Load Balancing API 的调用的详细信息，并将其作为日志文件存储在 Amazon S3 中。您可以使用这些 CloudTrail 日志来确定拨打了哪些呼叫、呼叫来自哪个源 IP 地址、谁拨打了电话、何时拨打了呼叫等。有关更多信息，请参阅使用记录 Elastic Load Balancing 的 API 调用 CloudTrail。

CloudWatch Application Load Balancer 的指标

Elastic Load Balancing 将您的 CloudWatch 负载均衡器和目标的数据点发布到亚马逊。 CloudWatch 允许您以一组有序的时间序列数据（称为指标）的形式检索有关这些数据点的统计信息。可将指标视为要监控的变量，而将数据点视为该变量随时间变化的值。例如，您可以在指定时间段内监控负载均衡器的正常目标的总数。每个数据点都有相关联的时间戳和可选测量单位。

您可使用指标来验证系统是否正常运行。例如，您可以创建 CloudWatch 警报来监控指定的指标，并在该指标超出您认为可接受的范围时启动操作（例如向电子邮件地址发送通知）。

CloudWatch 仅当请求流经负载均衡器时，Elastic Load Balancing 才会向其报告指标。如果有请求流经负载均衡器，则 Elastic Load Balancing 进行测量并以 60 秒的间隔发送其指标。如果没有请求流经负载均衡器或指标无数据，则不报告指标。

应用程序负载均衡器的指标不包括运行状况检查请求。

有关更多信息，请参阅 Amazon CloudWatch 用户指南。

内容

Application Load Balancer 指标

- 负载均衡器
- LCU
- 目标
- 目标组运行状况
- Lambda 函数
- 用户身份验证
- 目标优化器

AWS/ApplicationELB 命名空间包括负载均衡器的以下指标。

指标	描述
ActiveConnectionCount	从客户端到负载均衡器以及从负载均衡器到目标的并发活动 TCP 连接的总数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
AppCookieNonStickinessCount	负载均衡器因为无法使用现有的基于应用程序的粘性会话而选择新目标的请求数。例如，该请求是来自新客户端的第一个请求，但未显示应用程序粘性 cookie，提供了粘性 cookie 但无法解密，引用了不再在目标组中注册的目标，或者粘性 cookie 格式错误或已过期。 报告标准：在目标群体上启用了 Application-based粘性。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
BYoIPUtilPercentage	IP 池的使用百分比。 报告标准：负载均衡器已启用 BYoIP。 统计数据：唯一有意义的统计数据是 Average。 Dimensions • LoadBalancer ，TargetGroup • LoadBalancer ，TargetGroup ，AvailabilityZone

指标	描述
ClientTLSNegotiationErrorCount	由因 TLS 错误而未能与负载均衡器建立会话的客户端发起的 TLS 连接数。可能的原因包括密码或协议不匹配或者客户端因无法验证服务器证书而关闭了连接。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
DesyncMitigationMode_NonCompliant_Request_Count	不符合 RFC 7230 标准的请求数。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
ExcessiveLowReputationPackets	来自已知恶意来源且超过速率限制的数据包数量。这表示如果负载均衡器处于主动阻塞模式，则会丢弃的数据包。 报告标准：有非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
GrpcRequestCount	通过 IPv4 和 IPv6 处理的 gRPC 请求数量。 报告标准：有非零值 统计数据：最有用的统计数据是 Sum. Minimum、Maximum 以及 Average 全部返回 1。 Dimensions • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup • TargetGroup • AvailabilityZone ，TargetGroup
HTTP_Fixed_Response_Count	成功的固定响应操作的次数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTP_Redirect_Count	成功的重定向操作的次数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
HTTP_Redirect_Url_Limit_Exceeded_Count	由于响应位置标头中的 URL 大于 8K 而无法完成的重定向操作数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_3XX_Count	源自负载均衡器的 HTTP 3XX 重定向代码数。该计数不包含目标生成的响应代码。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
HTTPCode_ELB_4XX_Count	源自负载均衡器的 HTTP 4XX 客户端错误代码的数量。该计数不包含目标生成的响应代码。 如果请求格式错误或不完整，则会生成客户端错误。除了负载均衡器返回 HTTP 460 错误代码的情况之外，目标不会收到这些请求。该计数不包含目标生成的任何响应代码。 报告标准：有非零值 统计数据：最有用的统计数据是 Sum. Minimum、Maximum 以及 Average 全部返回 1。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_5XX_Count	源自负载均衡器的 HTTP 5XX 服务器错误代码的数量。该计数不包含目标生成的任何响应代码。 报告标准：有非零值 统计数据：最有用的统计数据是 Sum. Minimum、Maximum 以及 Average 全部返回 1。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
HTTPCode_ELB_500_Count	源自负载均衡器的 HTTP 500 错误代码的数量。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_502_Count	源自负载均衡器的 HTTP 502 错误代码的数量。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_503_Count	源自负载均衡器的 HTTP 503 错误代码的数量。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
HTTPCode_ELB_504_Count	源自负载均衡器的 HTTP 504 错误代码的数量。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
IPv6ProcessedBytes	负载均衡器通过 IPv6 处理的总字节数。此计数包含在 Processed Bytes 中。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
IPv6RequestCount	负载均衡器收到的 IPv6 请求的数量。 报告标准：有非零值 统计数据：最有用的统计数据是 Sum. Minimum、Maximum 以及 Average 全部返回 1。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
LowReputationPacketsDropped	已知恶意源丢弃的数据包数量。当请求被资源级 DDoS 保护阻止时，就会记录此指标。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
LowReputationRequestsDenied	被拒绝并返回 HTTP 403 响应的 HTTP 请求数量。当请求被资源级 DDoS 保护阻止时，就会记录此指标。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
NewConnectionCount	从客户端到负载均衡器以及从负载均衡器到目标建立的新 TCP 连接的总数。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
NonStickyRequestCount	负载均衡器因其无法使用现有粘性会话而选择新目标的请求的数目。例如，请求是来自新客户端的第一个请求且未提供粘性 Cookie，提供了粘性 Cookie 但未指定已注册到此目标组的目标，粘性 Cookie 的格式错误或已过期，或者出现内部错误，导致负载均衡器无法读取粘性 Cookie。 报告标准：已在目标组上启用粘性。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
ProcessedBytes	负载均衡器通过 IPv4 和 IPv6（HTTP 标头和 HTTP 有效负载）处理的总字节数。此计数包括流入和流出客户端和 Lambda 函数的流量、通过 Websocket 连接的流量，以及来自身份提供者（IdP）的流量（如果启用了用户身份验证）。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
RejectedConnectionCount	由于负载均衡器达到连接数上限被拒绝的链接的数量。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
RequestCount	通过 IPv4 和 IPv6 处理的请求的数量。此指标仅在负载均衡器节点能够选择目标的请求中递增。在选择目标之前拒绝的请求不会反映在此指标中。 报告标准：在有注册目标时报告。 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • LoadBalancer ，AvailabilityZone • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup
RuleEvaluations	负载均衡器在处理请求时评估的规则数量。默认规则不计算在内。此计数包含每个请求的 10 次免费规则评估。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer

AWS/ApplicationELB 命名空间包括下列负载均衡器容量单位（LCU）指标。

指标	说明
ConsumedLCUs	负载均衡器使用的负载均衡器容量单位 (LCU) 数量。您需要为每小时使用的 LCU 数量付费。如果 LCU 预留处于活动状态，ConsumedLCUs 将在使用量低于预留容量时报告 0，并在使用量超过预留的 LCU 时报告 0。有关更多信息，请参阅 Elastic Load Balancing 定价。 报告标准：始终报告 统计数据：全部 Dimensions • LoadBalancer
PeakLCUs	负载均衡器在给定时间点使用的最大负载均衡器容量单位（LCU）数量。仅在使用 LCU 预留时适用。 报告标准：始终 统计数据：最有用的统计工具为 Sum 和 Max。 Dimensions • LoadBalancer
ReservedLCUs	按分钟报告预留容量的计费指标。任何时间段的总 ReservedLCUs 就是您要付费的 LCU 量。例如，假设您预留 500 个 LCU 一小时，则每分钟指标将为 8.33 个 LCU。有关更多信息，请参阅 监控预留。 报告标准：有非零值 统计数据：全部 Dimensions • LoadBalancer

AWS/ApplicationELB 命名空间包括目标的以下指标。

指标	说明
AnomalousHostCount	检测到异常的主机数量。 报告标准：始终报告 统计数据：唯一有意义的统计数据是 Minimum 和 Maximum。 Dimensions • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
HealthyHostCount	被视为正常运行的目标数量。 报告标准：如果有注册目标则进行报告。 统计数据：最有用的统计工具是 Average、Minimum 和 Maximum。 Dimensions • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup
HTTPCode_Target_2XX_Count ，HTTPCode_Target_3XX_Count ，HTTPCode_Target_4XX_Count ，HTTPCode_Target_5XX_Count	目标生成的 HTTP 响应代码的数量。它不包括负载均衡器生成的任何响应代码。 报告标准：如果有注册目标则进行报告。 统计数据：最有用的统计数据是 Sum. Minimum、Maximum 以及 Average 全部返回 1。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer

指标	说明
MitigatedHostCount	正在缓解的目标数量。 报告标准：始终报告 统计数据：最有用的统计工具是 Average、Minimum 和 Maximum。 Dimensions • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
RequestCountPerTarget	目标组中每个目标的平均请求数。您必须使用 TargetGroup 维度指定目标组。如果目标是 Lambda 函数，则此指标不适用。 此计数使用目标组收到的请求总数除以目标组中运行状况良好的目标的数量。如果目标组中没有运行状况良好的目标，则将其除以已注册目标的总数。 报告标准：始终报告 统计：唯一有效的统计数据是 Sum。这代表平均值，而不是总和。 Dimensions • TargetGroup • TargetGroup ，AvailabilityZone • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup

指标	说明
TargetConnectionErrorCount	负载均衡器和目标之间连接建立不成功的次数。如果目标是 Lambda 函数，则此指标不适用。运行状况检查连接失败不会增加此指标的值。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
TargetResponseTime	请求离开负载均衡器到目标开始发送响应标头所经过的时间（以秒为单位）。这与访问日志中的 target_processing_time 字段是等效的。 报告标准：有非零值 Statistics：最有用的统计工具是 Average 和 pNN.NN（百分比）。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer

指标	说明
TargetTLSNegotiationErrorCount	由未与目标建立会话的负载均衡器发起的 TLS 连接数。可能的原因包括密码或协议不匹配。如果目标是 Lambda 函数，则此指标不适用。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
UnHealthyHostCount	被视为未正常运行的目标数量。 注销某个目标将会减少 HealthyHostCount 的值，但不会增加 UnhealthyHostCount 的值。 报告标准：如果有注册目标则进行报告。 统计数据：最有用的统计工具是 Average、Minimum 和 Maximum。 Dimensions • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup
ActiveZonalShiftHostCount	因可用区转移而被视为禁用的目标数量。 报告标准：在有值时报告。 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer ，TargetGroup . • AvailabilityZone ，LoadBalancer ，TargetGroup .

AWS/ApplicationELB 命名空间包括目标组运行状况的以下指标。有关更多信息，请参阅 the section called “目标组运行状况”。

指标	说明
HealthyStateDNS	符合 DNS 运行状况良好状态要求的区域数量。 Statistics：最有用的统计工具是 Max。 Dimensions • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
HealthyStateRouting	符合路由运行状况良好状态要求的区域数量。 Statistics：最有用的统计工具是 Max。 Dimensions • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
UnhealthyRoutingRequestCount	使用路由故障转移操作（失败时开放）路由的请求数。 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
UnhealthyStateDNS	不符合 DNS 运行状况良好状态要求，因此在 DNS 中被标记为运行状况不佳的区域数量。 Statistics：最有用的统计工具是 Min。 Dimensions • LoadBalancer ，TargetGroup

指标	说明
	• AvailabilityZone ，LoadBalancer ，TargetGroup
UnhealthyStateRouting	不符合路由运行状况良好状态要求，因此负载均衡器会将流量分配到区域中的所有目标（包括运行状况不佳的目标）的区域数量。 Statistics：最有用的统计工具是 Min。 Dimensions • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup

AWS/ApplicationELB 命名空间包括已注册为目标的 Lambda 函数的以下指标。

指标	说明
LambdaInternalError	因负载均衡器或 Amazon Lambda内部出现问题而导致失败的对 Lambda 函数的请求数。要获取错误原因代码，请查看访问日志的 error_reason 字段。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • TargetGroup • TargetGroup ，LoadBalancer
LambdaTargetProcessedBytes	负载均衡器为针对 Lambda 函数的请求和来自该函数的响应处理的字节的总数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。

指标	说明
	Dimensions • LoadBalancer
LambdaUserError	因 Lambda 函数出现问题而导致失败的对 Lambda 函数的请求数。例如，负载均衡器没有调用该函数的权限，负载均衡器从格式错误或缺少必填字段的函数接收 JSON，或者请求正文或响应的大小超过了 1 MB 的最大大小。要获取错误原因代码，请查看访问日志的 error_reason 字段。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • TargetGroup • TargetGroup ，LoadBalancer

AWS/ApplicationELB 命名空间包括用户身份验证的以下指标。

指标	描述
ELBAuthError	由于身份验证操作配置错误、负载均衡器无法与 IdP 建立连接，或负载均衡器因内部错误无法完成身份验证流程，所导致的无法完成用户身份验证的次数。要获取错误原因代码，请查看访问日志的 error_reason 字段。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
ELBAuthFailure	由于 IdP 拒绝用户访问或授权代码多次使用导致的无法完成用户身份验证的次数。要获取错误原因代码，请查看访问日志的 error_reason 字段。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
ELBAuthLatency	向 IdP 查询 ID 令牌和用户信息所用的时间（毫秒）。如果这些操作中有一项或多项操作失败，这表示失败时间。 报告标准：有非零值 统计数据：所有统计数据均有意义。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
ELBAuthRefreshTokenSuccess	负载均衡器使用 IdP 提供的刷新令牌成功刷新用户声明的次数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	描述
ELBAuthSuccess	成功的身份验证操作的次数。负载均衡器从 IdP 检索用户身份声明后，验证工作流结束时此指标会递增。 报告标准：有非零值 Statistics：最有用的统计工具是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
ELBAuthUserClaimsSizeExceeded	配置的 IdP 返回大小超过 11K 字节的用户声明的次数。 报告标准：有非零值 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

AWS/ApplicationELB命名空间包括目标优化器的以下指标。

指标	说明
TargetControlRequestCount	ALB 向代理转发的请求数。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	说明
TargetControlRequestRejectCount	由于没有目标可以接收请求而被 ALB 拒绝的请求数。该指标在为零时 TargetControlWorkQueueLength 显示上升。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlActiveChannelCount	ALB 和代理之间的活动控制通道数。对于负载均衡器，这应等于代理的数量。低于预期的数字表示代理配置不正确或不可用。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlNewChannelCount	ALB 和代理之间创建的新控制通道的数量。当安装了代理的新目标成功添加到目标组时，您会看到该指标有所上升。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

指标	说明
TargetControlChannelErrorCount	ALB 和代理之间未能建立或遇到意外错误的控制通道的数量。控制通道错误将导致该代理（和目标）未收到任何应用程序流量。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlWorkQueueLength	ALB 从代理处收到的请求请求的信号数。 这些数据来自以 1 分钟为间隔拍摄的快照。 Sub-minute 不会捕获更改。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计数据：唯一有意义的统计数据是 Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlProcessedBytes	ALB 为启用目标优化器的目标组的流量处理的字节数。 报告标准：目标组上启用了目标优化器，并且存在非零值。 统计：最有意义的统计数据是Sum。 Dimensions • LoadBalancer • AvailabilityZone ，LoadBalancer

Application Load Balancer 的指标维度

要筛选 Application Load Balancer 的指标，请使用以下维度。

维度	描述
AvailabilityZone	按可用区筛选指标数据。
LoadBalancer	按负载均衡器筛选指标数据。按以下方式指定负载均衡器：app/load-balancer-name/1234567890123456（负载均衡器 ARN 的结尾部分）。
TargetGroup	按目标组筛选指标数据。按以下方式指定目标组：targetgroup/target-group-name/1234567890123456（目标组 ARN 的结尾部分）。

Application Load Balancer 指标的统计数据

CloudWatch 根据 Elastic Load Balancing 发布的指标数据点提供统计数据。统计数据是在指定的时间段内汇总的指标数据。当请求统计数据时，返回的数据流按指标名称和维度进行识别。维度是用于唯一标识指标的名称-值对。例如，您可以请求在特定可用区内启动的负载均衡器背后所有正常状态 EC2 实例的统计数据。

Minimum 和 Maximum 统计数据反映每个采样窗口中各个负载均衡器节点报告的数据点的最小值和最大值。例如，假定应用程序负载均衡器由两个负载均衡器节点组成。一个节点的 HealthyHostCount 的 Minimum 为 2，Maximum 为 10，Average 为 6，另一个节点的 HealthyHostCount 的 Minimum 为 1，Maximum 为 5，Average 为 3。因此，负载均衡器的 Minimum 为 1，Maximum 为 10，Average 大约为 4。

我们建议您监控 Minimum 统计数据中的非零值 UnHealthyHostCount，并在多个数据点为非零值时发出警报。如果您使用 Minimum，则系统将检测负载均衡器中每个节点和可用区认为目标运行不正常的情况。如果您想收到有关潜在问题的提醒，则可以设置为按 Average 或 Maximum 发出警报，我们建议客户检查此指标并调查非零值情况。要自动减少故障，请遵循有关在 Amazon EC2 Auto Scaling 或 Amazon Elastic Container Service（Amazon ECS）中使用负载均衡器运行状况检查的最佳实践。

Sum 统计数据是所有负载均衡器节点的汇总值。由于这些指标在每个周期均包含多个报告，因此 Sum 仅适用于对所有负载均衡器节点进行汇总的指标。

SampleCount 统计数据是测量的样本数。由于这些指标是基于采样间隔和事件进行收集的，因此此统计信息一般没有用。例如，对于 HealthyHostCount，SampleCount 基于每个负载均衡器节点报告的样本数，而不是运行状况正常的主机数。

百分位数指示某个值在数据集中的相对位置。您可以指定任何百分位数，最多使用两位小数 (例如 p95.45)。例如，第 95 个百分位数表示 95% 的数据低于此值，5% 的数据高于此值。百分位数通常用于隔离异常值。例如，假设某个应用程序从缓存服务大多数请求的时间是 1-2 毫秒，但如果缓存是空的，则时间需要 100-200 毫秒。最大值反映了最慢的情况，也就是大约 200 毫秒。平均值不表示数据的分布。百分位提供了一个更有意义的应用程序性能视图。通过使用第 99 个百分位数作为 Auto Scaling 触发器或 CloudWatch 警报，您可以将处理时间超过 2 毫秒的请求设定为不超过 1%。

查看您的负载均衡器的 CloudWatch 指标

您可以使用 Amazon EC2 控制台查看您的负载均衡器的 CloudWatch 指标。这些指标显示为监控图表。如果负载均衡器处于活动状态并且正在接收请求，则监控图表会显示数据点。

或者，您可以使用 CloudWatch 控制台查看负载均衡器的指标。

使用控制台查看指标

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 要查看按目标组筛选的指标，请执行以下操作：
   a. 在导航窗格中，选择 Target Groups。
   b. 选择目标组，然后选择 Monitoring 选项卡。
   c. (可选) 要按时间筛选结果，请从 Showing data for 中选择时间范围。
   d. 要获得单个指标的一个较大视图，请选择其图形。
3. 要查看按负载均衡器筛选的指标，请执行以下操作：
   a. 在导航窗格中，选择 Load Balancers。
   b. 选择您的负载均衡器，然后选择 Monitoring 选项卡。
   c. (可选) 要按时间筛选结果，请从 Showing data for 中选择时间范围。
   d. 要获得单个指标的一个较大视图，请选择其图形。

使用 CloudWatch 控制台查看指标

1. 打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。

2. 在导航窗格中，选择指标。
3. 选择 ApplicationELB 命名空间。
4. （可选）要跨所有维度查看某个指标，请在搜索字段中输入其名称。
5. (可选) 要按维度筛选，请选择下列选项之一：

   - 要仅显示为负载均衡器报告的指标，请选择 Per AppELB Metrics。要查看单个负载均衡器的指标，请在搜索字段中输入其名称。
   - 要仅显示为您的目标组报告的指标，请选择 Per AppELB, per TG Metrics。要查看单个目标组的指标，请在搜索字段中输入其名称。
   - 要仅按可用区显示为负载均衡器报告的指标，请选择 Per AppELB, per AZ Metrics。要查看单个负载均衡器的指标，请在搜索字段中输入其名称。要查看单个可用区的指标，请在搜索字段中输入其名称。
   - 要仅按可用区和目标组显示为负载均衡器报告的指标，请选择 Per AppELB, per AZ, per TG Metrics。要查看单个负载均衡器的指标，请在搜索字段中输入其名称。要查看单个目标组的指标，请在搜索字段中输入其名称。要查看单个可用区的指标，请在搜索字段中输入其名称。

要查看指标，请使用 Amazon CLI

使用以下 list-metrics 命令列出可用指标：

```
aws cloudwatch list-metrics --namespace AWS/ApplicationELB
```

要获取指标的统计数据，请使用 Amazon CLI

使用以下 get-metric- statistics 命令获取指定指标和维度的统计信息。 CloudWatch 将每个唯一的维度组合视为一个单独的指标。您无法使用未专门发布的维度组合检索统计数据。您必须指定创建指标时使用的同一维度。

```
aws cloudwatch get-metric-statistics --namespace AWS/ApplicationELB \
--metric-name UnHealthyHostCount --statistics Average  --period 3600 \
--dimensions Name=LoadBalancer,Value=app/my-load-balancer/50dc6c495c0c9188 \
Name=TargetGroup,Value=targetgroup/my-targets/73e2d6bc24d8a067 \
--start-time 2016-04-18T00:00:00Z --end-time 2016-04-21T00:00:00Z
```

下面是示例输出：

```
{
```

```
    "Datapoints": [
        {
            "Timestamp": "2016-04-18T22:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2016-04-18T04:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        ...
    ],
    "Label": "UnHealthyHostCount"
}
```

Application Load Balancer 的访问日志

Elastic Load Balancing 提供了访问日志，该访问日志可捕获有关发送到负载均衡器的请求的详细信息。每个日志都包含信息 (例如，收到请求的时间、客户端的 IP 地址、延迟、请求路径和服务器响应)。您可以使用这些访问日志分析流量模式并解决问题。

访问日志是 Elastic Load Balancing 的一项可选功能，默认情况下已禁用此功能。为负载均衡器启用访问日志之后，Elastic Load Balancing 捕获日志并将其作为压缩文件存储在您指定的 Amazon S3 存储桶中。您可以随时禁用访问日志。

您需要支付 Amazon S3 的存储费用，但无需支付 Elastic Load Balancing 用以将日志文件发送到 Amazon S3 的带宽费用。有关存储成本的更多信息，请参阅 Amazon S3 定价。

目录

访问日志文件

Elastic Load Balancing 每 5 分钟为每个负载均衡器节点发布一次日志文件。日志传输最终是一致的。负载均衡器可以传输相同时间段的多个日志。通常，如果站点具有高流量，会出现此情况。

访问日志的文件名采用以下格式：

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-address_random-string.log.gz
```

bucket

S3 存储桶的名称。

prefix

（可选）存储桶的前缀（逻辑层级结构）。您指定的前缀不得包含字符串 AWSLogs。要获取更多信息，请参阅使用前缀整理对象。

AWSLogs

我们会在您指定的存储桶名称和可选前缀后添加以 AWSLogs 开头的文件名部分。

aws-account-id

所有者的 Amazon 账户 ID。

region

负载均衡器和 S3 存储桶所在的区域。

yyyy/mm/dd

传输日志的日期。

load-balancer-id

负载均衡器的资源 ID。如果资源 ID 包含任何正斜杠 (/)，这些正斜杠将替换为句点 (.)。

end-time

日志记录间隔结束的日期和时间。例如，结束时间 20140215T2340Z 包含在 UTC 时间（即祖鲁时间）23:35 和 23:40 之间发出的请求的条目。

ip-address

处理请求的负载均衡器节点的 IP 地址。对于内部负载均衡器，这是私有 IP 地址。

random-string

系统生成的随机字符串。

以下是一个带前缀的日志文件名示例：

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-
east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

以下是一个不带前缀的日志文件名示例：

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/
us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

日志文件可以在存储桶中存储任意长时间，不过您也可以定义 Amazon S3 生命周期规则以自动存档或删除日志文件。有关更多信息，请参阅《Amazon S3 用户指南》中的对象生命周期管理。

访问日志条目

Elastic Load Balancing 会记录发送给负载均衡器的请求，包括从未到达目标的请求。例如，如果客户端发送格式错误的请求或者没有正常的目标响应请求，仍会记录请求。

每个日志条目都包含向负载均衡器发出的单个请求（如果是连接 WebSockets）的详细信息。对于 WebSockets，只有在连接关闭后才会写入条目。如果无法建立升级后的连接，则 HTTP 或 HTTPS 请求的条目相同。

> **Important**
>
> Elastic Load Balancing 将尽力记录请求。我们建议您使用访问日志来了解请求性质，而不是作为所有请求的完整描述。

内容

- 采取的操作
- 分类原因
- 错误原因代码
- 转换状态代码

语法

下表按顺序描述了访问日志条目的字段。使用空格分隔所有字段。添加新字段时，我们会将其添加到日志条目的末尾。当我们准备发布新字段时，您可能会在该字段发布之前看到一个额外的尾号“-”。务必要将日志解析配置为在最后一个记录的字段之后停止，并在我们发布新字段后更新日志解析。

字段（位置）	说明
type（1）	请求或连接的类型。可能的值如下 (忽略任何其他值)： • `http` - HTTP • `https` – HTTP over TLS • h2— HTTP/2 通过 TLS • `grpcs` – gRPC over TLS • `ws` — WebSockets • `wss`— WebSockets 通过 TLS
time（2）	负载均衡器生成对客户端的响应的时间 (采用 ISO 8601 格式)。因为 WebSockets，这是连接关闭的时间。
elb（3）	负载均衡器的资源 ID。如果您正在解析访问日志条目，请注意，资源 ID 可包含正斜杠 (/)。
client:port（4）	请求客户端的 IP 地址和端口。如果负载均衡器前面有代理，则此字段将会包含该代理的 IP 地址。
target:port（5）	处理此请求的目标的 IP 地址和端口。 如果客户端没有发送完整请求，则负载均衡器无法将请求分派到目标，并且此值设置为 -。 如果目标是 Lambda 函数，则此值设置为 -。

字段（位置）	说明
	如果请求被阻止 Amazon WAF，则此值设置为-。
request_processing_time（6）	从负载均衡器收到请求到将请求发送到目标所用的总时间（以秒为单位，精度为毫秒）。 如果负载均衡器无法将请求分派到目标，则此值设置为 -1。如果目标在空闲超时前关闭连接，或客户端发送了格式错误的请求，则会发生这种情况。 如果在达到 10 秒 TCP 连接超时之前无法与目标建立 TCP 连接，也可以将此值设置为 -1。 如果您 Amazon WAF 的 Application Load Balancer 启用或目标类型是 Lambda 函数，则将客户端发送 POST 请求所需数据所花费的时间计入。`request_processing_time`
target_processing_time（7）	从负载均衡器将请求发送到目标到该目标开始发送响应标头所用的总时间（以秒为单位，精度为毫秒）。 如果负载均衡器无法将请求分派到目标，则此值设置为 -1。如果目标在空闲超时前关闭连接，或客户端发送了格式错误的请求，则会发生这种情况。 如果注册目标在空闲超时之前未响应，则此值还可设置为 -1。 如果您 Amazon WAF 的 Application Load Balancer 未启用，则将计入客户端发送 POST 请求所需数据所花费的时间`target_processing_time` 。
response_processing_time（8）	从负载均衡器收到来自目标的响应标头到开始向客户端发送响应所用的总时间（以秒为单位，精度为毫秒）。此时间包括在负载均衡器上的排队时间以及从负载均衡器到客户端的连接获取时间。 如果负载平衡器没有接收到来自目标的响应，则此值设置为 -1。如果目标在空闲超时前关闭连接，或客户端发送了格式错误的请求，则会发生这种情况。

字段（位置）	说明
elb_status_code（9）	由负载均衡器生成的响应的状态码、固定响应规则或阻止操作的 Amazon WAF 自定义响应代码。
target_status_code（10）	来自目标的响应的状态代码。仅在已建立与目标的连接且目标已发送响应的情况下记录此值。否则，其设置为 -。
received_bytes（11）	从客户端 (申请方) 接收的请求大小 (以字节为单位)。对于 HTTP 请求，这包括标头。对于 WebSockets，这是连接时从客户端接收的总字节数。
sent_bytes（12）	发送到客户端 (申请方) 的响应的大小 (以字节为单位)。对于 HTTP 请求，这包括响应标头和正文。对于 WebSockets，这是在连接上发送给客户端的总字节数。 TCP 标头和 TLS 握手有效载荷不包括在 `sent_bytes` 中。因此`sent_bytes` 无法匹配`DataTransfer-Out-Bytes` Amazon Cost Explorer。
"request_line"（13）	来自客户端的请求行，用双引号括起来，并使用以下格式记录：HTTP 方法 + 协议：//host: port/uri + HTTP 版本。负载均衡器将保留客户端记录请求 URI 时发送的 URL。它不设置访问日志文件的内容类型。当您处理此字段时，请考虑客户端发送 URL 的方式。
"user_agent"（14）	标识发起请求的客户端的 User-Agent 字符串，用双引号括起来。该字符串包含一个或多个产品标识符 (product[/version])。如果字符串长度超过 8 KB，则将被截断。
ssl_cipher（15）	[HTTPS 侦听器] SSL 密码。如果侦听器不是 HTTPS 侦听器，则此值设置为 -。
ssl_protocol（16）	[HTTPS 侦听器] SSL 协议。如果侦听器不是 HTTPS 侦听器，则此值设置为 -。
target_group_arn（17）	目标组的 Amazon 资源名称（ARN）。
"trace_id"（18）	X-Amzn-Trace-Id标题的内容，用双引号括起来。

字段（位置）	说明
"domain_name"（19）	[HTTPS 侦听器] TLS 握手期间客户端提供的 SNI 域（用双引号括起来）。如果客户端不支持 SNI 或此域与证书不匹配且将向客户端提供默认证书，则此值将设置为 -。
"chosen_cert_arn"（20）	[HTTPS 侦听器] 向客户端提供的证书的 ARN（用双引号括起来）。如果重复使用会话，则将此值设置为 `session-reused` 。如果侦听器不是 HTTPS 侦听器，则此值设置为 -。
matched_rule_priority（21）	匹配请求的规则的优先级值。如果匹配了某个规则，则此值的范围介于 1 和 50000 之间。如果未匹配任何规则并且已执行默认操作，则此值设置为 0。如果错误发生在规则评估时，则它设置为 -1。对于任何其他错误，它设置为 -。
request_creation_time（22）	负载均衡器从客户端收到请求的时间 (采用 ISO 8601 格式)。
"actions_executed"（23）	处理请求时执行的操作（用双引号括起来）。此值是一个逗号分隔的列表，可以包含 采取的操作 中所描述的值。如果未执行任何操作（例如，针对格式错误的请求的操作），则此值设置为 - 。
"redirect_url"（24）	HTTP 响应位置标头的重定向目标的 URL，使用双引号括起。如果不执行重定向操作，则此值设置为 -。
"error_reason"（25）	错误原因代码（包含在双引号内）。如果请求失败，则这是 错误原因代码 中描述的错误代码之一。如果所采取的操作不包含身份验证操作或目标不是 Lambda 函数，则此值设置为 -。
"target:port_list"（26）	处理此请求的目标的 IP 地址和端口的列表，各个地址和端口之间用空格分隔，并且用双引号括起。当前，此列表可以包含一个项，并且与 target:port 字段匹配。 如果客户端没有发送完整请求，则负载均衡器无法将请求分派到目标，并且此值设置为 -。 如果目标是 Lambda 函数，则此值设置为 -。 如果请求被阻止 Amazon WAF，则此值设置为-。

字段（位置）	说明
"target_status_code_list"（27）	目标响应的状态代码列表，代码之间用空格分隔，并且用双引号括起来。当前，此列表可以包含一个项，并且与 target_status_code 字段匹配。 仅在已建立与目标的连接且目标已发送响应的情况下记录此值。否则，其设置为 -。
"classification"（28）	异步缓解的分类（包含在双引号内）。如果请求不符合 RFC 7230 标准，则可能的值为 “可接受”、“不明确” 和 “严重”。 如果请求符合 RFC 7230 标准，则此值设置为“-”。
"classification_reason"（29）	分类原因代码（包含在双引号内）。如果请求不符合 RFC 7230 标准，则这是 分类原因 中描述的分类代码之一。如果请求符合 RFC 7230 标准，则此值设置为“-”。
conn_trace_id（30）	连接可追溯性 ID 是一个唯一的不透明 ID，用于标识每个连接。与客户端建立连接后，来自此客户端的后续请求将在其各自的访问日志条目中包含此 ID。此 ID 充当外键，用于在连接和访问日志之间建立链接。
"transformed_host"（31）	由主机标头重写转换修改后的主机标头。如果满足以下任何条件，则该值将设为 -。 • 未应用任何转换 • 转换失败 • 转换成功，但主机标头无更改 • 没有原始的主机标头（例如， HTTP/1.0 请求）
"transformed_uri"（32）	由 URL 重写转换修改后的 URI。如果满足以下任何条件，则该值将设为 -。 • 未应用任何转换 • 转换失败 • 转换成功，但 URI 无更改
"request_transform_status"（33）	重写转换的状态。如果未应用重写转换，则该值将设为 -。否则，该值将为the section called “转换状态代码”中所述的状态值之一。

采取的操作

负载均衡器将其采取的操作存储在访问日志的 actions_executed 字段中。

- `authenticate` – 负载均衡器验证会话，验证用户身份，并将用户信息添加到规则配置所指定的请求标头中。
- `fixed-response` – 负载均衡器发出规则配置所指定的固定响应。
- `forward` – 负载均衡器将请求转发到规则配置所指定的目标。
- `redirect` – 负载均衡器将请求重定向到规则配置所指定的另一个 URL。
- `rewrite`：负载均衡器按照规则配置中的规定重写请求 URL。
- `waf` — 负载均衡器将请求转发到 Amazon WAF 以确定是否应将请求转发到目标。如果这是最终操作，则 Amazon WAF 决定应拒绝该请求。默认情况下，被拒绝的请求 Amazon WAF 将在该字段中记录为 “403”。`elb_status_code`当配置 Amazon WAF 为拒绝使用自定义响应代码的请求时，该`elb_status_code`字段将反映配置的响应代码。
- `waf-failed`— 负载均衡器尝试将请求转发到 Amazon WAF，但此过程失败。

分类原因

如果请求不符合 RFC 7230 标准，负载均衡器将在访问日志的 classification_reason 字段中存储以下代码之一。有关更多信息，请参阅 异步缓解模式。

代码	描述	分类。
`AmbiguousUri`	请求 URI 包含控制字符。	不明确
`BadContentLength`	标 Content-Length 头包含无法解析的值或不是有效数字。	严重
`BadHeader`	标头包含 Null 字符或回车符。	严重
`BadTransferEncoding`	标 Transfer-Encoding 头包含错误值。	严重
`BadUri`	请求 URI 包含 Null 字符或回车符。	严重
`BadMethod`	请求方法格式不正确。	严重

代码	描述	分类。
BadVersion	请求版本格式不正确。	严重
BothTeClPresent	该请求同时包含标 Transfer-Encoding 头和标 Content-Length 头。	不明确
DuplicateContentLength	有多个具有相同值的 Content-Length 标头。	不明确
EmptyHeader	标头是空的，或者有一行中只包含空格。	不明确
GetHeadZeroContentLength	GET 或 Content-Length HEAD 请求有一个值为 0 的标头。	可接受
MultipleContentLength	有多个 Content-Length 标题具有不同的值。	严重
MultipleTransferEncodingChunked	有多个 Transfer-Encoding：分块标题。	严重
NonCompliantHeader	标头包含非 ASCII 字符或控制字符。	可接受
NonCompliantVersion	请求版本包含错误的值。	可接受
SpaceInUri	请求 URI 包含一个未采用 URL 编码的空格。	可接受
SuspiciousHeader	有一个标题可以标准化为 Transfer-Encoding 或 Content-Length 使用常见的文本标准化技术。	不明确
SuspiciousTeClPresent	该请求同时包含 Transfer-Encoding 标头和 Content-Length 标头，其中至少有一个是可疑的。	严重

代码	描述	分类。
UndefinedContentLengthSemantics	有一个为 GET 或 Content-Length HEAD 请求定义的标头。	不明确
UndefinedTransferEncodingSemantics	有一个为 GET 或 Transfer-Encoding HEAD 请求定义的标头。	不明确

错误原因代码

如果负载均衡器无法完成身份验证操作，则负载均衡器会在访问日志的 error_reason 字段中存储以下原因代码之一。负载均衡器还会增加相应的 CloudWatch 指标。有关更多信息，请参阅 使用 Application Load Balancer 验证用户身份。

代码	描述	指标
AuthInvalidCookie	身份验证 Cookie 无效。	ELBAuthFailure
AuthInvalidGrantError	来自令牌终端节点的授权授予代码无效。	ELBAuthFailure
AuthInvalidIdToken	ID 令牌无效。	ELBAuthFailure
AuthInvalidStateParam	状态参数无效。	ELBAuthFailure
AuthInvalidTokenResponse	来自令牌终端节点的响应无效。	ELBAuthFailure
AuthInvalidUserinfoResponse	来自用户信息终端节点的响应无效。	ELBAuthFailure

代码	描述	指标
AuthMissingCodeParam	来自授权终端节点的身份验证响应缺少名为“code”的查询参数。	ELBAuthFailure
AuthMissingHostHeader	来自授权终端节点的身份验证响应缺少主机标头字段。	ELBAuthError
AuthMissingStateParam	来自授权终端节点的身份验证响应缺少名为“state”的查询参数。	ELBAuthFailure
AuthTokenEpRequestFailed	令牌终端节点存在错误响应（非 2XX）。	ELBAuthError
AuthTokenEpRequestTimeout	负载均衡器无法与令牌端点通信，或者令牌端点在 5 秒内没有响应。	ELBAuthError
AuthUnhandledException	负载均衡器遇到未处理的异常。	ELBAuthError
AuthUserinfoEpRequestFailed	IdP 用户信息终端节点存在错误响应（非 2XX）。	ELBAuthError
AuthUserinfoEpRequestTimeout	负载均衡器无法与 IdP 用户信息端点通信，或者用户信息端点在 5 秒内没有响应。	ELBAuthError
AuthUserinfoResponseSizeExceeded	IdP 返回的声明大小超过 11 K 字节。	ELBAuthUserClaimsSizeExceeded

如果负载均衡器无法完成 jwt 验证操作，则负载均衡器会在访问日志的 error_reason 字段中存储以下原因代码之一。负载均衡器还会增加相应的 CloudWatch 指标。有关更多信息，请参阅 使用 Application Load Balancer 验证 JWT。

代码	描述	指标
JWTHeaderNotPresent	请求不包含授权标头。	JWTValidationFailureCount
JWTRequestFormatInvalid	请求中的令牌格式错误或缺少必填部分（标头、有效负载或签名），标头不包含 “Bearer” 前缀，Header 包含不同的身份验证类型，例如 “Basic”，如果请求中存在多个令牌，则存在授权标头但不存在令牌	JWTValidationFailureCount
JWKSRequestTimeout	负载均衡器无法与 JWKS 端点通信，或者 JWKS 端点在 5 秒钟内没有响应。	JWTValidationFailureCount
JWKSResponseSizeExceeded	JWKS 端点返回的响应大小超过 150KB 或者 JWKS 端点返回的密钥数量超过 10。	JWTValidationFailureCount
JWKSRequestFailed	有来自 JWKS 端点的错误响应（非 2XX）。	JWTValidationFailureCount
JWKSResponseInvalid	JWKS 响应存在以下一个或多个问题： Non-JSON 格式、字符无效、JWKS 格式无效、Missing/invalid 必填的 JWKS 属性、公钥的算法不支持、公钥无法转换为解码密钥、公钥大小不是 2K。	JWTValidationFailureCount
JWTSignatureValidationError	由于任何原因都无法验证令牌签名，包括签名不匹配，令牌是使用不支持的算法签名的，JWKS 端点中不存在令牌中的 KID。	JWTValidationFailureCount
JWTClaimNotPresent	客户端请求中的 JWT 不包含验证所需的声明	JWTValidationFailureCount

代码	描述	指标
JWTClaimFormatInvalid	JWT 中声明值的格式与配置中指定的格式不匹配	JWTValidationFailureCount
JWTClaimValueInvalid	JWT 中的索赔值无效。	JWTValidationFailureCount
JWTValidationInternalError	负载均衡器在验证客户端请求中的 JWT 时遇到了意外错误。	JWTValidationFailureCount

如果对加权目标组的请求失败，则负载均衡器会在访问日志的 error_reason 字段中存储下列错误代码之一。

代码	描述
AWSALBTGCookieInvalid	与加权目标组一起使用的 AWSALBTG Cookie 无效。例如，当 Cookie 值采用 URL 编码时，负载均衡器就会返回此错误。
WeightedTargetGroupsUnhandledException	负载均衡器遇到未处理的异常。

如果对 Lambda 函数的请求失败，则负载均衡器会在访问日志的 error_reason 字段中存储下列原因代码之一。负载均衡器还会增加相应的 CloudWatch 指标。有关更多信息，请参阅 Lambda 调用操作。

代码	描述	指标
LambdaAccessDenied	负载均衡器无权调用 Lambda 函数。	LambdaUserError
LambdaBadRequest	Lambda 调用失败，因为客户端请求标头或正文不只包含字符。 UTF-8	LambdaUserError

代码	描述	指标
LambdaConnectionError	负载均衡器无法连接到 Lambda。	LambdaInternalError
LambdaConnectionTimeout	尝试连接到 Lambda 时发生超时。	LambdaInternalError
LambdaEC2AccessDeniedException	Amazon EC2 在函数初始化期间拒绝了对 Lambda 的访问。	LambdaUserError
LambdaEC2ThrottledException	Amazon EC2 在函数初始化期间限制了 Lambda。	LambdaUserError
LambdaEC2UnexpectedException	Amazon EC2 在函数初始化期间遇到意外异常。	LambdaUserError
LambdaENILimitReachedException	Lambda 无法在 Lambda 函数配置中指定的 VPC 中创建网络接口，因为超出了网络接口的限制。	LambdaUserError
LambdaInvalidResponse	来自 Lambda 函数的响应的格式不正确或缺少必填字段。	LambdaUserError
LambdaInvalidRuntimeException	指定版本的 Lambda 运行时不受支持。	LambdaUserError
LambdaInvalidSecurityGroupIDException	Lambda 函数配置中指定的安全组 ID 无效。	LambdaUserError
LambdaInvalidSubnetIDException	Lambda 函数配置中指定的子网 ID 无效。	LambdaUserError

代码	描述	指标
LambdaInvalidZipFileException	Lambda 无法解压缩指定的函数 zip 文件。	LambdaUserError
LambdaKMSAccessDeniedException	Lambda 无法解密环境变量，因为对 KMS 密钥的访问已被拒绝。检查 Lambda 函数的 KMS 权限。	LambdaUserError
LambdaKMSDisabledException	Lambda 无法解密环境变量，因为指定的 KMS 密钥已被禁用。检查 Lambda 函数的 KMS 密钥设置。	LambdaUserError
LambdaKMSInvalidStateException	Lambda 无法解密环境变量，因为 KMS 密钥的状态无效。检查 Lambda 函数的 KMS 密钥设置。	LambdaUserError
LambdaKMSNotFoundException	Lambda 无法解密环境变量，因为找不到 KMS 密钥。检查 Lambda 函数的 KMS 密钥设置。	LambdaUserError
LambdaRequestTooLarge	请求正文的大小已超过 1 MB。	LambdaUserError
LambdaResourceNotFound	找不到 Lambda 函数。	LambdaUserError
LambdaResponseTooLarge	响应的大小已超过 1 MB。	LambdaUserError
LambdaServiceException	Lambda 遇到了内部错误。	LambdaInternalError
LambdaSubnetIPAddressLimitReachedException	Lambda 无法为 Lambda 函数设置 VPC 访问权限，因为一个或多个子网没有可用的 IP 地址。	LambdaUserError

代码	描述	指标
LambdaThrottling	Lambda 函数受到限制，因为请求过多。	LambdaUserError
LambdaUnhandled	Lambda 函数遇到了未处理的异常。	LambdaUserError
LambdaUnhandledException	负载均衡器遇到未处理的异常。	LambdaInternalError
LambdaWebsocketNotSupported	WebSockets Lambda 不支持。	LambdaUserError

如果负载均衡器在向转发请求时遇到错误 Amazon WAF，它会在访问日志的 error_reason 字段中存储以下错误代码之一。

代码	说明
WAFConnectionError	负载均衡器无法连接到 Amazon WAF。
WAFConnectionTimeout	与的连接 Amazon WAF 超时。
WAFResponseReadTimeout	请求 Amazon WAF 超时。
WAFServiceError	Amazon WAF 返回了一个 5XX 错误。
WAFUnhandledException	负载均衡器遇到未处理的异常。

转换状态代码

代码	说明
TransformBufferTooSmall	重写转换失败，因为结果超出内部缓冲区的大小。尝试减少正则表达式的复杂性。
TransformCompileError	正则表达式编译失败。
TransformCompileTooBig	编译后的正则表达式太大。尝试减少正则表达式的复杂性。
TransformInvalidHost	主机标头重写转换失败，因为生成的主机无效。
TransformInvalidPath	URL 重写转换失败，因为生成的路径无效。
TransformRegexSyntaxError	正则表达式包含语法错误。
TransformReplaceError	转换替换失败。
TransformSuccess	重写转换成功完成。

示例日志条目

以下是示例日志条目。请注意，示例文本以多行形式显示，这只是为了更方便阅读。

示例 HTTP 条目

以下是 HTTP 侦听器 (端口 80 到端口 80) 的示例日志条目：

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "-" "-"
0 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.1:80" "200" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

示例 HTTPS 条目

以下是 HTTPS 侦听器 (端口 443 到端口 80) 的示例日志条目：

```
https 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.086 0.048 0.037 200 200 0 57
"GET https://www.example.com:443/ HTTP/1.1" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
 TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337281-1d84f3d73c47ec4e58577259" "www.example.com" "arn:aws:acm:us-
east-2:123456789012:certificate/12345678-1234-1234-1234-123456789012"
1 2018-07-02T22:22:48.364000Z "authenticate,forward" "-" "-" "10.0.0.1:80" "200" "-"
 "-"
TID_1234abcd5678ef90 "m.example.com" "-" "TransformSuccess"
```

示例 HTTP/2 条目

以下是 HTTP/2 直播的日志条目示例。

```
h2 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.1.252:48160 10.0.0.66:9000 0.000 0.002 0.000 200 200 5 257
"GET https://10.0.2.105:773/ HTTP/2.0" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
 TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337327-72bd00b0343d75b906739c42" "-" "-"
1 2018-07-02T22:22:48.364000Z "redirect" "https://example.com:80/" "-" "10.0.0.66:9000"
 "200" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

示例 WebSockets 条目

以下是 WebSockets 连接的日志条目示例。

```
ws 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:40914 10.0.1.192:8010 0.001 0.003 0.000 101 101 218 587
"GET http://10.0.0.30:80/ HTTP/1.1" "-" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.1.192:8010" "101" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

安全 WebSockets 入口示例

以下是安全 WebSockets连接的日志条目示例。

```
wss 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:44244 10.0.0.171:8010 0.000 0.001 0.000 101 101 218 786
"GET https://10.0.0.30:443/ HTTP/1.1" "-" ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2
arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.171:8010" "101" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

Lambda 函数的示例条目

以下是对 Lambda 函数的成功请求的示例日志条目：

```
http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "-" "-" "-" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

以下是对 Lambda 函数的失败请求的示例日志条目：

```
http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 502 - 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "LambdaInvalidResponse" "-" "-" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

配置日志传输通知

要在弹性负载均衡将日志传输到 S3 存储桶时收到通知，请使用 Amazon S3 事件通知功能。Elastic Load Balancing 使用PutObjectCreateMultipartUpload、和 POST 对象将日志传输到 Amazon S3。为确保您收到所有日志传输通知，请在配置中包含所有这些对象创建事件。

有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Amazon S3 事件通知。

处理访问日志文件

访问日志文件是压缩文件。如果您下载这些文件，则必须对其进行解压才能查看信息。

如果您的网站上有大量需求，则负载均衡器可以生成包含大量数据的日志文件 (以 GB 为单位)。您可能无法通过逐行处理来处理数量如此庞大的数据。因此，您可能必须使用提供并行处理解决方案的分析工具。例如，您可以使用以下分析工具分析和处理访问日志：

- Amazon Athena 是一种交互式查询服务，让您能够轻松使用标准 SQL 分析 Amazon S3 中的数据。有关更多信息，请参阅 Amazon Athena 用户指南中的查询 Application Load Balancer 日志。
- Loggly
- Splunk
- Sumo Logic

为 Application Load Balancer 启用访问日志

在为负载均衡器启用访问日志时，您必须指定负载均衡器将在其中存储日志的 S3 存储桶的名称。存储桶必须具有为 Elastic Load Balancing 授予写入存储桶的权限的存储桶策略。

任务

- 步骤 1：创建 S3 存储桶
- 步骤 2：将策略附加到 S3 存储桶
- 步骤 3：配置访问日志
- 步骤 4：确认存储桶权限
- 问题排查

步骤 1：创建 S3 存储桶

在启用访问日志时，您必须为访问日志指定 S3 存储桶。您可以使用现有存储桶，也可以创建专门用于访问日志的存储桶。存储桶必须满足以下要求。

要求

- 存储桶必须位于与负载均衡器相同的区域中。该存储桶和负载均衡器可由不同的账户拥有。

- 唯一支持的服务器端加密选项是 Amazon S3-managed 密钥 (SSE-S3)。有关更多信息，请参阅 Amazon S3-managed 加密密钥 (SSE-S3)。

使用 Amazon S3 控制台创建 S3 存储桶。

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择 Create bucket (创建存储桶)。
3. 在 Create a bucket (创建存储桶) 页上，执行以下操作：
   a. 对于存储桶名称，请输入存储桶的名称。此名称在 Amazon S3 内所有现有存储桶名称中必须唯一。在某些区域，可能对存储桶名称有其他限制。有关更多信息，请参阅《Amazon S3 用户指南》中的存储桶限制。
   b. 对于 Amazon 区域，选择在其中创建负载均衡器的区域。
   c. 对于默认加密，请选择 Amazon S3-managed 密钥 (SSE-S3)。
   d. 选择 创建存储桶 。

步骤 2：将策略附加到 S3 存储桶

S3 存储桶必须具有为 Elastic Load Balancing 授予将访问日志写入存储桶的权限的存储桶策略。存储桶策略是 JSON 语句的集合，这些语句以访问策略语言编写，用于为存储桶定义访问权限。每个语句都包括有关单个权限的信息并包含一系列元素。

如果您正在使用具有附加策略的现有存储桶，则可以将 Elastic Load Balancing 访问日志的语句添加到该策略。如果您这样做，则建议您评估生成的权限集，以确保它们适用于需要具有对访问日志的存储桶的访问权的用户。

存储桶策略

该策略向日志传送服务授予权限。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
```

```
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

对于 Resource，使用示例策略中显示的格式，输入访问日志所在位置的 ARN。请务必在 S3 存储桶 ARN 的资源路径中，包含该负载均衡器所在账户的账户 ID。这样可以确保只有来自指定账户的负载均衡器才能将访问日志写入 S3 存储桶。

您指定的 ARN 取决于您是否计划在步骤 3 中启用访问日志时包含前缀。

带前缀的 S3 存储桶 ARN 示例

S3 存储桶名称为 amzn-s3-demo-logging-bucket，前缀为 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

不带前缀的 S3 存储桶 ARN 示例

S3 存储桶名称为 amzn-s3-demo-logging-bucket。S3 存储桶 ARN 中没有前缀部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

传统存储桶策略

在此之前，对于 2022 年 8 月之前可用的区域，我们要求向该区域特定的弹性负载均衡账户授予权限的策略。此旧版策略仍然受到支持，但我们建议您将其替换为上述新版策略。当然如果您愿意，也可以继续使用旧版策略（此处未显示）。

安全最佳实践

- 使用完整的资源路径，包括 S3 存储桶 ARN 的账户 ID 部分。请勿在 S3 存储桶 ARN 的账户 ID 部分使用通配符（*）。

  ```
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
  ```

- 使用 aws:SourceArn 确保只有来自指定区域和账户的负载均衡器才能使用您的存储桶。

  ```
  "Condition": {
      "ArnLike": {
  ```

```
        "aws:SourceArn":
 "arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/*"
    }
}
```

- 将 aws:SourceOrgId 和 aws:SourceArn 结合使用，以确保只有来自指定组织的负载均衡器才能使用您的存储桶。

```
"Condition": {
    "StringEquals": {
        "aws:SourceOrgId": "o-1234567890"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
    }
}
```

- 如果您有 Deny 语句会阻止除显式允许的主体之外的服务主体访问，请务必将 logdelivery.elasticloadbalancing.amazonaws.com 添加到允许的服务主体列表中。例如，假设您使用了 aws:PrincipalServiceNamesList 条件，请按如下所示添加 logdelivery.elasticloadbalancing.amazonaws.com：

```
{
  "Effect": "Deny",
  "Principal": "*",
  "Condition": {
      "StringNotEqualsIfExists": {
          "aws:PrincipalServiceNamesList": [
              "logdelivery.elasticloadbalancing.amazonaws.com",
              "service.amazonaws.com"
          ]
      }
  }
}
```

  如果您使用了 NotPrincipal 元素，请按如下所示添加 logdelivery.elasticloadbalancing.amazonaws.com。请注意，我们建议您使用 aws:PrincipalServiceName 或 aws:PrincipalServiceNamesList 条件键来显式允许服务主体，而不是使用 NotPrincipal 元素。有关更多信息，请参阅 NotPrincipal。

```
{
```

```
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
       "logdelivery.elasticloadbalancing.amazonaws.com",
       "service.amazonaws.com"
    ]
  }
},
```

创建存储桶策略后，使用 Amazon S3 接口（例如 Amazon S3 控制台或 Amazon CLI 命令）将您的存储桶策略附加到 S3 存储桶。

Console

将您的存储桶策略附加到 S3 存储桶

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择存储桶的名称以打开其详细信息页面。
3. 选择 Permissions（权限），然后选择 Bucket policy（存储桶策略)、Edit（编辑）。
4. 更新存储桶策略以授予所需权限。
5. 选择保存更改。

Amazon CLI

将您的存储桶策略附加到 S3 存储桶

使用 put-bucket-policy 命令。在此示例中，存储桶策略已保存到指定的 .json 文件中。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://access-log-policy.json
```

步骤 3：配置访问日志

使用以下过程配置访问日志，以捕获请求信息并将日志文件传输到 S3 存储桶。

要求

存储桶必须满足第 1 步中所描述的要求，并且必须附加第 2 步中所描述的存储桶策略。如果包括前缀，前缀中不得包含字符串“AWSLogs”。

管理保存访问日志的 S3 存储桶

要删除您配置用于访问日志的存储桶，请确保首先禁用访问日志。否则，如果在一个不属于您的 Amazon Web Services 账户 中创建了具有相同名称和必要的存储桶策略的新存储桶，Elastic Load Balancing 会将您的负载均衡器的访问日志写入这个新存储桶。

Console

启用访问日志

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择您的负载均衡器的名称以打开其详细信息页面。
4. 在属性选项卡上，选择编辑。
5. 对于监控，打开访问日志。
6. 对于 S3 URI，输入日志文件的 S3 URI。您指定的 URI 取决于您是否使用前缀。
   - 带有前缀的 URI: s3://*amzn-s3-demo-logging-bucket*/*logging-prefix*
   - 不带前缀的 URI: s3://*amzn-s3-demo-logging-bucket*
7. 选择保存更改。

Amazon CLI

启用访问日志

使用带相关属性的 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes \
        Key=access_logs.s3.enabled,Value=true \
        Key=access_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=access_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

启用访问日志

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含相关属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "access_logs.s3.enabled"
          Value: "true"
        - Key: "access_logs.s3.bucket"
          Value: "amzn-s3-demo-logging-bucket"
        - Key: "access_logs.s3.prefix"
          Value: "logging-prefix"
```

步骤 4：确认存储桶权限

在为负载均衡器启用访问日志后，Elastic Load Balancing 将验证 S3 存储桶，并创建测试文件以确保存储桶策略指定所需权限。您可以使用 Amazon S3 控制台验证是否已创建测试文件。测试文件不是实际的访问日志文件；它不包含示例记录。

使用 Amazon S3 控制台验证您的存储桶中是否创建了测试文件

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择您指定用于访问日志的存储桶的名称。
3. 导航到测试文件 ELBAccessLogTestFile。位置取决于您是否使用前缀。
   - 带有前缀的位置：*amzn-s3-demo-logging-bucket*/*logging-prefix*/aw *123456789012* sLogs//ELBAccessLogTestFile
   - 没有前缀的位置：*amzn-s3-demo-logging-bucket*/ awsLogs//*123456789012*ELBAccessLogTestFile

问题排查

如果您遇到访问被拒绝错误，则可能的原因如下：

- 存储桶策略没有为 Elastic Load Balancing 授予将访问日志写入存储桶的权限。确认您使用的是该区域正确的存储桶策略。确认资源 ARN 使用的存储桶名称与您在启用访问日志时指定的存储桶名称相同。如果您在启用访问日志时未指定前缀，请确认资源 ARN 不包含前缀。
- 存储桶使用不支持的服务器端加密选项。存储桶必须使用 Amazon S3-managed 密钥 (SSE-S3)。

为 Application Load Balancer 禁用访问日志

您随时可为您的负载均衡器禁用访问日志。在禁用访问日志后，您的访问日志将在 S3 存储桶中保留，直至您将其删除。有关更多信息，请参阅《Amazon S3 用户指南》中的创建、配置和使用 S3 存储桶。

Console

要禁用访问日志

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择您的负载均衡器的名称以打开其详细信息页面。
4. 在属性选项卡上，选择编辑。
5. 对于监控，关闭访问日志。
6. 选择保存更改。

Amazon CLI

要禁用访问日志

使用 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=access_logs.s3.enabled,Value=false
```

应用程序负载均衡器的连接日志

Elastic Load Balancing 提供了连接日志，该连接日志可捕获有关发送到负载均衡器的请求的详细信息。每个日志都包含客户端的 IP 地址和端口、侦听器端口、使用的 TLS 密码和协议、TLS 握手延迟、连接状态和客户端证书详细信息等信息。您可以使用这些连接日志来分析请求模式并排查问题。

链接日志是 Elastic Load Balancing 的一项可选功能，默认情况下已禁用此功能。为负载均衡器启用连接日志之后，Elastic Load Balancing 捕获日志并将其作为压缩文件存储在您指定的 Amazon S3 存储桶中。您可以随时禁用连接日志。

您需要支付 Amazon S3 的存储费用，但无需支付 Elastic Load Balancing 用以将日志文件发送到 Amazon S3 的带宽费用。有关存储成本的更多信息，请参阅 Amazon S3 定价。

内容

连接日志文件

Elastic Load Balancing 每 5 分钟为每个负载均衡器节点发布一次日志文件。日志传输最终是一致的。负载均衡器可以传输相同时间段的多个日志。通常，如果站点具有高流量，会出现此情况。

连接日志的文件名采用以下格式：

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/
conn_log_aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-
address_random-string.log.gz
```

bucket

S3 存储桶的名称。

prefix

（可选）存储桶的前缀（逻辑层级结构）。您指定的前缀不得包含字符串 AWSLogs。要获取更多信息，请参阅使用前缀整理对象。

AWSLogs

我们会在您指定的存储桶名称和可选前缀后添加以 AWSLogs 开头的文件名部分。

aws-account-id

所有者的 Amazon 账户 ID。

region

负载均衡器和 S3 存储桶所在的区域。

yyyy/mm/dd

传输日志的日期。

load-balancer-id

负载均衡器的资源 ID。如果资源 ID 包含任何正斜杠 (/)，这些正斜杠将替换为句点 (.)。

end-time

日志记录间隔结束的日期和时间。例如，结束时间 20140215T2340Z 包含在 UTC 时间（即祖鲁时间）23:35 和 23:40 之间发出的请求的条目。

ip-address

处理请求的负载均衡器节点的 IP 地址。对于内部负载均衡器，这是私有 IP 地址。

random-string

系统生成的随机字符串。

以下是一个带前缀的日志文件名示例：

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/
conn_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

以下是一个不带前缀的日志文件名示例：

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/conn_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

日志文件可以在存储桶中存储任意长时间，不过您也可以定义 Amazon S3 生命周期规则以自动存档或删除日志文件。有关更多信息，请参阅《Amazon S3 用户指南》中的对象生命周期管理。

连接日志条目

每次连接尝试在连接日志文件中都有一个条目。客户端请求如何发送取决于连接是持久的还是非持久的。非持久连接只有一个请求，其会在访问日志和连接日志中创建一个条目。持久连接有多个请求，其会在访问日志中创建多个条目，并在连接日志中创建单个条目。

内容

语法

下表按顺序描述了连接日志条目的各个字段。使用空格分隔所有字段。添加新字段时，我们会将其添加到日志条目的末尾。当我们准备发布新字段时，您可能会在该字段发布之前看到一个额外的尾号“-”。务必要将日志解析配置为在最后一个记录的字段之后停止，并在我们发布新字段后更新日志解析。

字段（位置）	说明
timestamp（1）	负载均衡器成功建立连接或建立连接失败的时间（采用 ISO 8601 格式）。
client_ip（2）	请求客户端的 IP 地址。
client_port（3）	请求客户端的端口。
listener_port（4）	接收客户端请求的负载均衡器侦听器的端口。
tls_protocol（5）	[HTTPS 侦听器] 握手期间使用的 SSL/TLS 协议。-对于非 SSL/TLS 请求，此字段设置为。

字段（位置）	说明
tls_cipher（6）	[HTTPS 侦听器] 握手期间使用的 SSL/TLS 协议。-对于非 SSL/TLS 请求，此字段设置为。
tls_handshake_latency（7）	[HTTPS 侦听器] 建立成功握手所经过的总时间（以秒为单位，精确到毫秒）。在下列情况下，此字段设置为 -： • 传入的请求不是 SSL/TLS 请求。 • 未成功建立握手。
leaf_client_cert_subject（8）	[HTTPS 侦听器] 叶客户端证书的使用者名称。在下列情况下，此字段设置为 -： • 传入的请求不是 SSL/TLS 请求。 • 负载均衡器侦听器未配置为启用 mTLS。 • 服务器无法获取 le load/parse af 客户端证书。
leaf_client_cert_validity（9）	[HTTPS 侦听器] 叶客户端证书的有效期，`not-before` 和 `not-after` 采用 ISO 8601 格式。在下列情况下，此字段设置为 -： • 传入的请求不是 SSL/TLS 请求。 • 负载均衡器侦听器未配置为启用 mTLS。 • 服务器无法获取 le load/parse af 客户端证书。
leaf_client_cert_serial_number（10）	[HTTPS 侦听器] 叶客户端证书的序列号。在下列情况下，此字段设置为 -： • 传入的请求不是 SSL/TLS 请求。 • 负载均衡器侦听器未配置为启用 mTLS。 • 服务器无法获取 le load/parse af 客户端证书。
tls_verify_status（11）	[HTTPS 侦听器] 连接请求的状态。如果成功建立连接，则此值为 `Success`。连接失败时，该值为`Failed:$error_code` 。
conn_trace_id（12）	连接可追溯性 ID 是一个唯一的不透明 ID，用于标识每个连接。与客户端建立连接后，来自此客户端的后续请求将在各自的访问日志条目中包含此 ID。此 ID 充当外键，用于在连接和访问日志之间建立链接。

字段（位置）	说明
tls_keyxchange (13)	[HTTPS 侦听器] 握手期间使用的密钥交换 TLS 或。 PQ-TLS -对于非 SSL/TLS 请求，此字段设置为。

错误原因代码

如果负载均衡器无法建立连接，则负载均衡器将在连接日志中存储以下原因代码之一。

代码	说明
`ClientCertMaxChainDepthExceeded`	已超过最大客户端证书链深度
`ClientCertMaxSizeExceeded`	已超过最大客户端证书大小
`ClientCertCrlHit`	客户端证书已被 CA 吊销
`ClientCertCrlProcessingError`	CRL 处理错误
`ClientCertUntrusted`	客户端证书不可信
`ClientCertNotYetValid`	客户证书尚未生效
`ClientCertExpired`	客户端证书已过期
`ClientCertTypeUnsupported`	客户端证书类型不受支持

代码	说明
ClientCertInvalid	客户端证书无效
ClientCertPurposeInvalid	客户证书用途无效
ClientCertRejected	客户端证书被自定义服务器验证拒绝
UnmappedConnectionError	未映射的运行时连接错误
ClientCertIncompatible	客户端证书与所选监听器安全策略不兼容

示例 日志条目

以下是连接日志条目示例。请注意，示例文本以多行形式显示，这只是为了更方便阅读。

以下是与端口 443 上启用了双向 TLS 验证模式的 HTTPS 侦听器成功连接的日志条目示例。

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
 4.036
"CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Success TID_3180a73013c8ca4bac2f731159d4b0fe
```

以下是与端口 443 上启用了双向 TLS 验证模式的 HTTPS 侦听器连接失败的日志条目示例。

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
 -
"CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Failed:ClientCertUntrusted TID_1c71a68d70587445ad5127ff8b2687d7
```

处理连接日志文件

连接日志文件已被压缩。如果您使用 Amazon S3 控制台打开这些文件，则将对其进行解压缩，并且将显示信息。如果您下载这些文件，则必须对其进行解压才能查看信息。

如果您的网站上有大量需求，则负载均衡器可以生成包含大量数据的日志文件 (以 GB 为单位)。您可能无法通过逐行处理来处理数量如此庞大的数据。因此，您可能必须使用提供并行处理解决方案的分析工具。例如，您可以使用以下分析工具分析和处理连接日志：

- Amazon Athena 是一种交互式查询服务，方便使用标准 SQL 分析 Amazon S3 的数据。
- Loggly
- Splunk
- Sumo Logic

启用应用程序负载均衡器的连接日志

在为负载均衡器启用连接日志时，您必须指定负载均衡器将在其中存储日志的 S3 存储桶的名称。存储桶必须具有为 Elastic Load Balancing 授予写入存储桶的权限的存储桶策略。

任务

- 步骤 1：创建 S3 存储桶
- 步骤 2：将策略附加到 S3 存储桶
- 步骤 3：配置连接日志
- 步骤 4：确认存储桶权限
- 问题排查

步骤 1：创建 S3 存储桶

在启用连接日志时，您必须为连接日志指定 S3 存储桶。您可以使用现有存储桶，也可以创建专门用于连接日志的存储桶。存储桶必须满足以下要求。

要求

- 存储桶必须位于与负载均衡器相同的区域中。该存储桶和负载均衡器可由不同的账户拥有。
- 唯一支持的服务器端加密选项是 Amazon S3-managed 密钥 (SSE-S3)。有关更多信息，请参阅 Amazon S3-managed 加密密钥 (SSE-S3)。

使用 Amazon S3 控制台创建 S3 存储桶。

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择 Create bucket (创建存储桶)。
3. 在 Create a bucket (创建存储桶) 页上，执行以下操作：
   a. 对于存储桶名称，请输入存储桶的名称。此名称在 Amazon S3 内所有现有存储桶名称中必须唯一。在某些区域，可能对存储桶名称有其他限制。有关更多信息，请参阅《Amazon S3 用户指南》中的存储桶限制。
   b. 对于 Amazon 区域，选择在其中创建负载均衡器的区域。
   c. 对于默认加密，请选择 Amazon S3-managed 密钥 (SSE-S3)。
   d. 选择 创建存储桶 。

步骤 2：将策略附加到 S3 存储桶

S3 存储桶必须具有为 Elastic Load Balancing 授予将连接日志写入存储桶的权限的存储桶策略。存储桶策略是 JSON 语句的集合，这些语句以访问策略语言编写，用于为存储桶定义访问权限。每个语句都包括有关单个权限的信息并包含一系列元素。

如果您正在使用具有附加策略的现有存储桶，则可以将 Elastic Load Balancing 连接日志的语句添加到该策略。如果您这样做，则建议您评估生成的权限集，以确保它们适用于需要具有对连接日志的存储桶的访问权的用户。

存储桶策略

该策略向指定的日志传送服务授予权限。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
```

```
  ]
}
```

对于 Resource，使用示例策略中显示的格式，输入访问日志所在位置的 ARN。请务必在 S3 存储桶 ARN 的资源路径中，包含该负载均衡器所在账户的账户 ID。这样可以确保只有来自指定账户的负载均衡器才能将访问日志写入 S3 存储桶。

您指定的 ARN 取决于您是否计划在步骤 3 中启用访问日志时包含前缀。

带前缀的 S3 存储桶 ARN 示例

S3 存储桶名称为 amzn-s3-demo-logging-bucket，前缀为 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

不带前缀的 S3 存储桶 ARN 示例

S3 存储桶名称为 amzn-s3-demo-logging-bucket。S3 存储桶 ARN 中没有前缀部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

传统存储桶策略

在此之前，对于 2022 年 8 月之前可用的区域，我们要求向该区域特定的弹性负载均衡账户授予权限的策略。此旧版策略仍然受到支持，但我们建议您将其替换为上述新版策略。当然如果您愿意，也可以继续使用旧版策略（此处未显示）。

安全最佳实践

为了增强安全性，请使用精确的 S3 存储桶 ARN。

- 使用完整的资源路径，而不仅仅是 S3 存储桶 ARN。
- 包括 S3 存储桶 ARN 的账户 ID 部分。
- 请勿在 S3 存储桶 ARN 的账户 ID 部分使用通配符（*）。

创建存储桶策略后，使用 Amazon S3 接口（例如 Amazon S3 控制台或 Amazon CLI 命令）将您的存储桶策略附加到 S3 存储桶。

Console

将您的存储桶策略附加到 S3 存储桶

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择存储桶的名称以打开其详细信息页面。
3. 选择 Permissions（权限），然后选择 Bucket policy（存储桶策略)、Edit（编辑）。
4. 更新存储桶策略以授予所需权限。
5. 选择保存更改。

Amazon CLI

将您的存储桶策略附加到 S3 存储桶

使用 put-bucket-policy 命令。在此示例中，存储桶策略已保存到指定的 .json 文件中。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://access-log-policy.json
```

步骤 3：配置连接日志

使用以下过程配置连接日志，以捕获日志文件并将其传输到 S3 存储桶。

要求

存储桶必须满足第 1 步中所描述的要求，并且必须附加第 2 步中所描述的存储桶策略。如果要指定前缀，前缀中不得包含字符串“AWSLogs”。

管理连接日志的 S3 存储桶

要删除您配置用于连接日志的存储桶，请确保首先禁用连接日志。否则，如果在一个不属于您的 Amazon Web Services 账户 中创建了具有相同名称和必要的存储桶策略的新存储桶，Elastic Load Balancing 会将您的负载均衡器的连接日志写入这个新存储桶。

Console

启用连接日志

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。

2. 在导航窗格中，选择负载均衡器。
3. 选择您的负载均衡器的名称以打开其详细信息页面。
4. 在属性选项卡上，选择编辑。
5. 对于监控，启用连接日志。
6. 对于 S3 URI，输入日志文件的 S3 URI。您指定的 URI 取决于您是否使用前缀。
   - 带有前缀的 URI：`s3://bucket-name/prefix`
   - 不带前缀的 URI：`s3://bucket-name`
7. 选择保存更改。

Amazon CLI

启用连接日志

使用带相关属性的 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes \
        Key=connection_logs.s3.enabled,Value=true \
        Key=connection_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=connection_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

启用连接日志

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含相关属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
```

```
    SecurityGroups:
      - !Ref mySecurityGroup
    LoadBalancerAttributes:
      - Key: "connection_logs.s3.enabled"
        Value: "true"
      - Key: "connection_logs.s3.bucket"
        Value: "amzn-s3-demo-logging-bucket"
      - Key: "connection_logs.s3.prefix"
        Value: "logging-prefix"
```

步骤 4：确认存储桶权限

在为负载均衡器启用连接日志后，Elastic Load Balancing 将验证 S3 存储桶，并创建测试文件以确保存储桶策略指定所需权限。您可以使用 Amazon S3 控制台验证是否已创建测试文件。测试文件不是实际的连接日志文件；它不包含示例记录。

验证 Elastic Load Balancing 是否在 S3 存储桶中创建了测试文件

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择您指定用于连接日志的存储桶的名称。
3. 导航到测试文件 `ELBConnectionLogTestFile`。位置取决于您是否使用前缀。
   - 带有前缀的位置：*amzn-s3-demo-logging-bucket*/*prefix*/aw *123456789012* sLogs//ELBConnectionLogTestFile
   - 没有前缀的位置：*amzn-s3-demo-logging-bucket*/awsLogs//*123456789012*ELBConnectionLogTestFile

问题排查

如果您遇到访问被拒绝错误，则可能的原因如下：

- 存储桶策略没有为 Elastic Load Balancing 授予将连接日志写入存储桶的权限。确认您使用的是该区域正确的存储桶策略。确认资源 ARN 使用的存储桶名称与您在启用连接日志时指定的存储桶名称相同。如果您在启用连接日志时未指定前缀，请确认资源 ARN 不包含前缀。
- 存储桶使用不支持的服务器端加密选项。存储桶必须使用 Amazon S3-managed 密钥 (SSE-S3)。

禁用应用程序负载均衡器的连接日志

您可以随时为负载均衡器禁用连接日志。在禁用连接日志后，您的连接日志将在 S3 存储桶中保留，直至您将其删除。有关更多信息，请参阅《Amazon S3 用户指南》中的创建、配置和使用存储桶。

Console

禁用连接日志

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择您的负载均衡器的名称以打开其详细信息页面。
4. 在属性选项卡上，选择编辑。
5. 对于监控，关闭连接日志。
6. 选择保存更改。

Amazon CLI

禁用连接日志

使用 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=connection_logs.s3.enabled,Value=false
```

Health 检查日志

Elastic Load Balancing 提供运行状况检查日志，用于捕获有关注册目标运行状况检查状态的详细信息，包括运行状况检查失败时的失败原因。EC2 实例、IP 地址和 Lambda 函数目标都支持运行状况检查日志。每个日志条目都包含运行状况检查请求类型或连接、时间戳、目标地址、目标组 ID、健康状态和原因代码等信息。您可以使用这些运行状况检查日志来分析目标运行状况模式、监控运行状况变化并解决问题。

Health check 日志是一项可选功能，默认情况下处于禁用状态。为负载均衡器启用运行状况检查日志后，Elastic Load Balancing 会捕获日志并将其作为压缩文件存储在您指定的 Amazon S3 存储桶中。您可以随时禁用运行状况检查日志。

您需要支付 Amazon S3 的存储费用，但无需支付 Elastic Load Balancing 用以将日志文件发送到 Amazon S3 的带宽费用。有关存储成本的更多信息，请参阅 Amazon S3 定价。

内容

Health 检查日志文件

Elastic Load Balancing 每 5 分钟为每个负载均衡器节点发布一次日志文件。当有大量目标连接到负载均衡器或配置了较小的运行状况检查间隔（例如，每 5 秒）时，负载均衡器可以在同一时间段内传送多个日志。

运行状况检查日志的文件名使用以下格式：

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/
health_check_log_aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-
time_ip-address_random-string.log.gz
```

bucket

S3 存储桶的名称。

prefix

（可选）存储桶的前缀（逻辑层级结构）。您指定的前缀不得包含字符串 AWSLogs。要获取更多信息，请参阅使用前缀整理对象。

AWSLogs

我们会在您指定的存储桶名称和可选前缀后添加以 AWSLogs 开头的文件名部分。

aws-account-id

所有者的 Amazon 账户 ID。

region

负载均衡器和 S3 存储桶所在的区域。

yyyy/mm/dd

传输日志的日期。

load-balancer-id

负载均衡器的资源 ID。如果资源 ID 包含任何正斜杠 (/)，这些正斜杠将替换为句点 (.)。

end-time

日志记录间隔结束的日期和时间。例如，结束时间 20140215T2340Z 包含在 UTC 时间（即祖鲁时间）23:35 和 23:40 之间发出的请求的条目。

ip-address

处理请求的负载均衡器节点的 IP 地址。对于内部负载均衡器，这是私有 IP 地址。

random-string

系统生成的随机字符串。

以下是一个带前缀的日志文件名示例：

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/
health_check_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

以下是一个不带前缀的日志文件名示例：

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/health_check_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

日志文件可以在存储桶中存储任意长时间，不过您也可以定义 Amazon S3 生命周期规则以自动存档或删除日志文件。有关更多信息，请参阅《Amazon S3 用户指南》中的对象生命周期管理。

Health check 日志条目

Elastic Load Balancing 记录目标运行状况检查结果，包括该负载均衡器所有注册目标的失败原因。每个日志条目都包含对注册目标进行的单个运行状况检查结果的详细信息。

内容

语法

下表按顺序描述了运行状况检查日志条目的字段。使用空格分隔所有字段。添加新字段时，我们会将其添加到日志条目的末尾。当我们准备发布新字段时，您可能会在该字段发布之前看到一个额外的尾号“-”。务必要将日志解析配置为在最后一个记录的字段之后停止，并在我们发布新字段后更新日志解析。

字段（位置）	说明
type（1）	运行状况检查请求或连接的类型。可能的值如下 (忽略任何其他值)： • `http`--HTTP • `https`--通过 TLS 的 HTTP • `h2`-- HTTP/2 通过 TLS • `grpc`--gRPC • `lambda`--Lambda 函数
time（2）	对目标启动运行状况检查的时间戳，格式为 ISO 8601。
延迟 (3)	完成当前运行状况检查所用的总时间（以秒为单位）。
目标地址 (4)	目标的 IP 地址和端口，格式为 IP:Port。如果目标是 Lambda 函数，则为 Lambda 的 ARN。
目标群组 ID (5)	与目标关联的目标组的名称。
状态 (6)	运行状况检查的状态。`PASS`如果运行状况检查成功，则此值为。运行状况检查失败时，该值为 `FAIL`
状态码 (7)	从目标收到的运行状况检查请求的响应代码。
原因代码 (8)	如果运行状况检查失败，则失败的原因。请参阅 错误原因代码。

错误原因代码

如果目标运行状况检查失败，负载均衡器将在运行状况检查日志中记录以下原因代码之一。

代码	说明
TimedOut	由于与目标的连接尝试超时或目标在配置的运行状况检查超时时间内没有响应，因此运行状况检查失败。当目标的安全组阻止运行状况检查端口上的入站流量、目标响应缓慢或 TLS 握手未及时完成时，就会发生这种情况
ConnectionReset	Health check 失败，因为目标在返回有效响应之前重置或优雅地关闭了连接
ResponseCodeMismatch	目标对运行状况检查请求的响应的 HTTP 状态代码与配置的状态码不匹配
ResponseStringMismatch	目标返回的响应正文不包含目标组运行状况检查配置中配置的字符串
InternalError	内部负载均衡器错误
TargetError	Target 在响应运行状况检查请求时返回 5xx 错误代码
GRPCStatusHeaderEmpty	GRPC 目标响应有一个没有值的 grpc-status 标头
GRPCUnexpectedStatus	GRPC 目标以意外的 grpc 状态进行响应

> **Note**
>
> 新的TimedOut错误原因代码取代了RequestTimedOut和ConnectionTimedOut原因代码。

示例 日志条目

以下是运行状况检查日志条目的示例。请注意，示例文本以多行形式显示，这只是为了更方便阅读。

以下是成功运行状况检查的日志条目示例。

```
http 2025-10-31T12:44:59.875678Z 0.019584011 172.31.20.97:80 HCLogsTestIPs PASS 200 -
```

以下是运行状况检查失败的示例日志条目。

```
http 2025-10-31T12:44:58.901409Z 1.121980746 172.31.31.9:80 HCLogsTestIPs FAIL 502
 TargetError
```

配置日志传输通知

要在弹性负载均衡将日志传输到 S3 存储桶时收到通知，请使用 Amazon S3 事件通知功能。Elastic Load Balancing 使用PutObjectCreateMultipartUpload、和 POST 对象将日志传输到 Amazon S3。为确保您收到所有日志传输通知，请在配置中包含所有这些对象创建事件。

有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Amazon S3 事件通知。

处理运行状况检查日志文件

运行状况检查日志文件已压缩。如果您下载这些文件，则必须对其进行解压才能查看信息。

如果您的网站上有大量需求，则负载均衡器可以生成包含大量数据的日志文件 (以 GB 为单位)。您可能无法通过逐行处理来处理数量如此庞大的数据。因此，您可能必须使用提供并行处理解决方案的分析工具。例如，您可以使用以下分析工具来分析和处理运行状况检查日志：

- Amazon Athena 是一种交互式查询服务，方便使用标准 SQL 分析 Amazon S3 的数据。
- Loggly
- Splunk
- Sumo Logic

为 Application Load Balancer 启用运行状况检查日志

为负载均衡器启用运行状况检查日志时，必须指定负载均衡器将存储日志的 S3 存储桶的名称。存储桶必须具有为 Elastic Load Balancing 授予写入存储桶的权限的存储桶策略。

任务

- 步骤 1：创建 S3 存储桶

步骤 1：创建 S3 存储桶

启用运行状况检查日志时，必须为运行状况检查日志指定 S3 存储桶。您可以使用现有的存储桶，也可以创建专门用于运行状况检查日志的存储桶。存储桶必须满足以下要求。

要求

- 存储桶必须位于与负载均衡器相同的区域中。该存储桶和负载均衡器可由不同的账户拥有。
- 唯一支持的服务器端加密选项是 Amazon S3-managed 密钥 (SSE-S3)。有关更多信息，请参阅 Amazon S3-managed 加密密钥 (SSE-S3)。

使用 Amazon S3 控制台创建 S3 存储桶。

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择 Create bucket (创建存储桶)。
3. 在 Create a bucket (创建存储桶) 页上，执行以下操作：
   a. 对于存储桶名称，请输入存储桶的名称。此名称在 Amazon S3 内所有现有存储桶名称中必须唯一。在某些区域，可能对存储桶名称有其他限制。有关更多信息，请参阅《Amazon S3 用户指南》中的存储桶限制。
   b. 对于 Amazon 区域，选择在其中创建负载均衡器的区域。
   c. 对于默认加密，请选择 Amazon S3-managed 密钥 (SSE-S3)。
   d. 选择 创建存储桶 。

步骤 2：将策略附加到 S3 存储桶

您的 S3 存储桶必须具有存储桶策略，该策略可授予 Elastic Load Balancing 将运行状况检查日志写入存储桶的权限。存储桶策略是 JSON 语句的集合，这些语句以访问策略语言编写，用于为存储桶定义访问权限。每个语句都包括有关单个权限的信息并包含一系列元素。

如果您使用的是已附加策略的现有存储桶，则可以将 Elastic Load Balancing 运行状况检查日志的语句添加到策略中。如果您这样做，我们建议您评估生成的权限集，以确保它们适用于需要访问存储桶以获取运行状况检查日志的用户。

存储桶策略

该策略向指定的日志传送服务授予权限。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

对于 `Resource`，使用示例策略中显示的格式，输入访问日志所在位置的 ARN。请务必在 S3 存储桶 ARN 的资源路径中，包含该负载均衡器所在账户的账户 ID。这样可以确保只有来自指定账户的负载均衡器才能将访问日志写入 S3 存储桶。

您指定的 ARN 取决于您是否计划在步骤 3 中启用访问日志时包含前缀。

带前缀的 S3 存储桶 ARN 示例

S3 存储桶名称为 amzn-s3-demo-logging-bucket，前缀为 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

不带前缀的 S3 存储桶 ARN 示例

S3 存储桶名称为 amzn-s3-demo-logging-bucket。S3 存储桶 ARN 中没有前缀部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

传统存储桶策略

在此之前，对于 2022 年 8 月之前可用的区域，我们要求向该区域特定的弹性负载均衡账户授予权限的策略。此旧版策略仍然受到支持，但我们建议您将其替换为上述新版策略。当然如果您愿意，也可以继续使用旧版策略（此处未显示）。

安全最佳实践

为了增强安全性，请使用精确的 S3 存储桶 ARN。

- 使用完整的资源路径，而不仅仅是 S3 存储桶 ARN。
- 包括 S3 存储桶 ARN 的账户 ID 部分。
- 请勿在 S3 存储桶 ARN 的账户 ID 部分使用通配符（*）。

创建存储桶策略后，使用 Amazon S3 接口（例如 Amazon S3 控制台或 Amazon CLI 命令）将您的存储桶策略附加到 S3 存储桶。

Console

将您的存储桶策略附加到 S3 存储桶

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择存储桶的名称以打开其详细信息页面。
3. 选择 Permissions（权限），然后选择 Bucket policy（存储桶策略)、Edit（编辑）。
4. 更新存储桶策略以授予所需权限。
5. 选择保存更改。

Amazon CLI

将您的存储桶策略附加到 S3 存储桶

使用 put-bucket-policy 命令。在此示例中，存储桶策略已保存到指定的 .json 文件中。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://access-log-policy.json
```

步骤 3：配置运行状况检查日志

使用以下过程配置运行状况检查日志，以捕获日志文件并将其传送到您的 S3 存储桶。

要求

存储桶必须满足第 1 步中所描述的要求，并且必须附加第 2 步中所描述的存储桶策略。如果要指定前缀，前缀中不得包含字符串“AWSLogs”。

管理运行状况检查日志的 S3 存储桶

在删除为运行状况检查日志配置的存储桶之前，请务必禁用运行状况检查日志。否则，如果有一个新的存储桶名称和所需的存储桶策略，但是在您不拥有的存储桶中创建的，Elastic Load Balan Amazon Web Services 账户 cing 可能会将您的负载均衡器的运行状况检查日志写入这个新存储桶。

Console

启用运行状况检查日志

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择您的负载均衡器的名称以打开其详细信息页面。
4. 在属性选项卡上，选择编辑。
5. 对于监控，请打开 Healt h Check 日志。
6. 对于 S3 URI，输入日志文件的 S3 URI。您指定的 URI 取决于您是否使用前缀。
   - 带有前缀的 URI：`s3://bucket-name/prefix`
   - 不带前缀的 URI：`s3://bucket-name`
7. 选择保存更改。

Amazon CLI

启用运行状况检查日志

使用带相关属性的 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes \
```

```
        Key=health_check_logs.s3.enabled,Value=true \
        Key=health_check_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=health_check_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

启用运行状况检查日志

更新 AWS::ElasticLoadBalancingV2:: LoadBalancer 资源以包含相关属性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "health_check_logs.s3.enabled"
          Value: "true"
        - Key: "health_check_logs.s3.bucket"
          Value: "amzn-s3-demo-logging-bucket"
        - Key: "health_check_logs.s3.prefix"
          Value: "logging-prefix"
```

步骤 4：确认存储桶权限

为您的负载均衡器启用运行状况检查日志后，Elastic Load Balancing 会验证 S3 存储桶并创建一个测试文件以确保存储桶策略指定了所需的权限。您可以使用 Amazon S3 控制台验证是否已创建测试文件。测试文件不是实际的运行状况检查日志文件；它不包含示例记录。

验证 Elastic Load Balancing 是否在 S3 存储桶中创建了测试文件

1. 打开 Amazon S3 控制台，网址为 https://console.aws.amazon.com/s3/。
2. 选择您为运行状况检查日志指定的存储桶的名称。
3. 导航到测试文件 ELBHealthCheckLogTestFile。位置取决于您是否使用前缀。

- 带有前缀的位置：*amzn-s3-demo-logging-bucket*/*prefix*/aw *123456789012* sLogs//ELBHealthCheckLogTestFile
- 没有前缀的位置：*amzn-s3-demo-logging-bucket*/awsLogs//*123456789012*ELBHealthCheckLogTestFile

问题排查

如果您遇到访问被拒绝错误，则可能的原因如下：

- 存储桶策略不授予 Elastic Load Balancing 向存储桶写入运行状况检查日志的权限。确认您使用的是该区域正确的存储桶策略。确认资源 ARN 使用的存储桶名称与您在启用运行状况检查日志时指定的存储桶名称相同。如果您在启用运行状况检查日志时未指定前缀，请确认资源 ARN 不包含前缀。
- 存储桶使用不支持的服务器端加密选项。存储桶必须使用 Amazon S3-managed 密钥 (SSE-S3)。

禁用 Application Load Balancer 的运行状况检查日志

您可以随时禁用负载均衡器的运行状况检查日志。禁用运行状况检查日志后，您的运行状况检查日志将保留在您的 S3 存储桶中，直到您将其删除。有关更多信息，请参阅《Amazon S3 用户指南》中的创建、配置和使用存储桶。

Console

禁用运行状况检查日志

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格中，选择负载均衡器。
3. 选择您的负载均衡器的名称以打开其详细信息页面。
4. 在属性选项卡上，选择编辑。
5. 对于监控，请关闭 Health 检查日志。
6. 选择保存更改。

Amazon CLI

禁用运行状况检查日志

使用 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=health_check_logs.s3.enabled,Value=false
```

请求 Application Load Balancer 的跟踪

当负载均衡器收到来自客户端的请求时，它会在将请求发送到目标之前添加或更新X-Amzn-Trace-Id标头。负载均衡器和目标之间的任何服务或应用程序也可以添加或更新此标头。

您可以使用请求跟踪来跟踪 HTTP 请求 (从客户端到目标或其他服务)。如果启用访问日志，则会记录标X-Amzn-Trace-Id头的内容。有关更多信息，请参阅 Application Load Balancer 的访问日志。

语法

标X-Amzn-Trace-Id题包含以下格式的字段：

```
Field=version-time-id
```

字段

字段的名称。支持的值是 `Root` 和 `Self`。

应用程序可以出于自身目的添加任意字段。负载均衡器将保留这些字段，但不会使用它们。

版本

版本号。该值为 1。

time

新纪元时间 (用秒表示)。该值的长度为 8 位十六进制数字。

id

跟踪标识符。该值的长度为 24 位十六进制数字。

示例

如果传入的请求中不存在X-Amzn-Trace-Id标头，则负载均衡器会生成带有`Root`字段的标头并转发请求。例如：

```
X-Amzn-Trace-Id: Root=1-67891233-abcdef012345678912345678
```

如果标X-Amzn-Trace-Id头存在并且有Root字段，则负载均衡器会插入一个Self字段并转发请求。例如：

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-
abcdef012345678912345678
```

如果应用程序添加了包含一个 Root 字段和一个自定义字段的标头，则负载均衡器将保留这两个字段并插入一个 Self 字段，然后再转发该请求：

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-
abcdef012345678912345678;CalledFrom=app
```

如果标X-Amzn-Trace-Id头存在且有Self字段，则负载均衡器会更新该Self字段的值。

限制

- 负载均衡器在接收到传入的请求时将更新标头，而在接收到响应时不进行更新。
- 如果 HTTP 标头大于 7 KB，则负载均衡器会使用Root字段重写X-Amzn-Trace-Id标头。
- 使用 WebSockets，您只能在升级请求成功之前进行跟踪。

对 Application Load Balancer 进行故障排除

以下信息可帮助您解决与 Application Load Balancer 相关的问题。

问题

已注册目标未处于可用状态

如果目标进入 `InService` 状态所花费的时间超过预期，则该目标可能无法通过运行状况检查。您的目标未处于可用状态，除非通过一次运行状况检查。有关更多信息，请参阅 应用程序负载均衡器目标组的运行状况检查。

确认您的实例未通过运行状况检查，然后检查以下问题：

安全组不允许流量

> 与实例关联的安全组必须允许来自负载均衡器的使用运行状况检查端口和运行状况检查协议的流量。您可以向实例安全组添加一个规则以允许来自负载均衡器安全组的所有流量。负载均衡器的安全组也必须允许流入实例的流量。

网络访问控制列表 (ACL) 不允许流量

与实例的子网关联的网络 ACL 必须允许运行状况检查端口上的入站流量，以及临时端口 (1024-65535) 上的出站流量。与您负载均衡器节点的子网关联的网络 ACL 必须允许临时端口上的入站流量，以及运行状况检查端口和临时端口上的出站流量。

ping 路径不存在

创建运行状况检查的目标页并将其路径指定为 ping 路径。

连接超时

首先，验证您是否能使用目标的私有 IP 地址和运行状况检查协议直接从网络中连接到目标。如果您无法连接，请检查实例是否被过度使用，并将多个目标添加到目标组 (如果它太忙而无法响应)。如果您可以连接，则在运行状况检查超时期限之前，目标页可能不会响应。为运行状况检查选择更简单的目标页，或者调整运行状况检查设置。

目标未返回成功响应代码

默认情况下，成功代码为 200，但您可以选择在配置运行状况检查时指定其他成功代码。确认负载均衡器所需的成功代码，并且应用程序已配置为在成功时返回这些代码。

目标响应代码格式错误或者连接到目标时出错

验证您的应用程序是否可以对负载均衡器的运行状况检查请求做出响应。某些应用程序需要进行额外配置才能对运行状况检查做出响应，例如只有进行虚拟主机配置才能对负载均衡器发送的 HTTP 主机标头做出响应。主机标头值包含目标的私有 IP 地址，后跟不使用默认端口时的运行状况检查端口。如果目标使用默认运行状况检查端口，则主机标头值仅包含目标的私有 IP 地址。例如，如果目标的私有 IP 地址为 `10.0.0.10` 且运行状况检查端口为 `8080`，则负载均衡器在运行状况检查中发送的 HTTP 主机标头为 `Host: 10.0.0.10:8080`。如果目标的私有 IP 地址为 `10.0.0.10` 且运行状况检查端口为 `80`，则负载均衡器在运行状况检查中发送的 HTTP 主机标头为 `Host: 10.0.0.10`。可能需要进行虚拟主机配置来响应该主机，或需要使用默认配置才能成功对应用程序进行运行状况检查。运行状况检查请求具有以下属性：将 `User-Agent` 设置为 `ELB-HealthChecker/2.0`，消息标头字段的行终止符为序列 CRLF，且标头在第一个空行处终止，后跟 CRLF。

客户端无法连接到面向 Internet 的负载均衡器

如果负载均衡器未响应请求，请检查以下问题：

您的面向 Internet 的负载均衡器已连接到私有子网

您必须为负载均衡器指定公有子网。公有子网有一个指向 Virtual Private Cloud (VPC) 的 Internet 网关的路由。

安全组或网络 ACL 不允许流量

负载均衡器的安全组和负载均衡器子网的任何网络 ACLs 都必须允许来自客户端的入站流量以及通过侦听器端口流向客户端的出站流量。

负载均衡器无法接收发送到自定义域的请求

如果负载均衡器无法接收发送到自定义域的请求，请检查以下问题：

自定义域名无法解析为负载均衡器 IP 地址

- 使用命令行界面确认自定义域名解析为哪个 IP 地址。
  - Linux、macOS 或 Unix – 您可以在终端中使用 `dig` 命令。Ex.`dig example.com`
  - Windows – 您可以在命令提示符中使用 `nslookup` 命令。Ex.`nslookup example.com`
- 使用命令行界面确认负载均衡器 DNS 解析为哪个 IP 地址。
- 比较两个输出的结果。IP 地址必须匹配。

如果使用 Route 53 托管您的自定义域，请参阅《Amazon Route 53 开发人员指南》中的我的域在 Internet 上不可用。

发送到负载均衡器的 HTTPS 请求返回“NET::ERR_CERT_COMMON_NAME_INVALID”

如果 HTTPS 请求收到来自负载均衡器的 `NET::ERR_CERT_COMMON_NAME_INVALID`，请查看以下可能的原因：

- HTTPS 请求中使用的域名与 ACM 证书所关联的侦听器中指定的备用名称不匹配。
- 正在使用负载均衡器的默认 DNS 名称。无法使用默认 DNS 名称发出 HTTPS 请求，因为无法为 `*.amazonaws.com` 域请求公有证书。

负载均衡器显示的处理时间较长

负载均衡器计算处理时间的方式因配置而异。

- 如果与您 Amazon WAF 的 Application Load Balancer 关联并且客户端发送了 HTTP POST 请求，则发送 POST 请求的数据的时间将反映在负载均衡器访问日志的`request_processing_time`字段中。此行为对于 HTTP POST 请求是符合预期的。
- 如果未与您 Amazon WAF 的 Application Load Balancer 关联，并且客户端发送了 HTTP POST 请求，则发送 POST 请求的数据的时间将反映在负载均衡器访问日志的`target_processing_time`字段中。此行为对于 HTTP POST 请求是符合预期的。

负载均衡器发送响应代码 000

使用 HTTP/2 连接时，如果通过一个连接提供的请求数超过 10000，则负载均衡器将发送 GOAWAY 帧并使用 TCP FIN 关闭连接。

负载均衡器生成 HTTP 错误

以下 HTTP 错误由负载均衡器生成。负载均衡器将 HTTP 代码发送到客户端，将请求保存到访问日志并增加 `HTTPCode_ELB_4XX_Count` 或 `HTTPCode_ELB_5XX_Count` 指标。

错误

- HTTP 400：错误请求
- HTTP 401: 未授权
- HTTP 403：禁止访问
- HTTP 405：不允许的方法
- HTTP 408：请求超时
- HTTP 413：有效负载过大
- HTTP 414：URI 太长
- HTTP 460
- HTTP 463
- HTTP 464
- HTTP 500：内部服务器错误

- HTTP 501：未实现
- HTTP 502：无效网关
- HTTP 503：服务不可用
- HTTP 504：网关超时
- HTTP 505：不支持版本
- HTTP 507：存储空间不足
- HTTP 561: 未授权
- HTTP 562：JWKS 请求失败

HTTP 400：错误请求

可能的原因：

- 客户端发送的请求格式错误，不符合 HTTP 规范。
- 请求标头超过了每个请求行 16K、单个标头 16K 或整个请求标头 64K 的限制。
- 客户端在发送完整的请求正文之前关闭了连接。

HTTP 401: 未授权

您已将侦听器规则配置为对用户进行身份验证，但出现下列情况之一：

- 已将 `OnUnauthenticatedRequest` 配置为拒绝未经身份验证的用户或 IdP 拒绝访问。
- IdP 返回的声明大小超出了负载均衡器支持的最大大小。
- 客户端提交了一个不带主机标头的 HTTP/1.0 请求，并且负载均衡器未能生成重定向 URL。
- 请求的范围未返回 ID 令牌。
- 您未在客户端登录超时到期之前完成登录过程。有关更多信息，请参阅客户端登录超时。
- 由于以下原因之一，JWT 身份验证失败：
  - 请求缺少授权标头。 (JWTHeaderNotPresent)
  - 请求中的令牌格式无效。在以下情况下可能会发生这种情况：
    - Token 格式错误或缺少必填部分（标头、有效负载或签名）
    - 标题缺少 “持有者” 前缀
    - 标头包含不同的身份验证类型（例如，“Basic”）

    - 授权标头存在但缺少令牌
    - 请求中存在多个令牌 (JWTRequestFormatInvalid)
- 令牌签名验证失败。在以下情况下可能会发生这种情况：
    - 签名不匹配
    - 公钥无效或无法转换为解码密钥
    - 公钥大小不是 2K
    - 使用不支持的算法对令牌进行签名
    - 令牌中的 KID 不存在于 JWKS 端点中 () JWTSignature ValidationFailed
- JWT 缺少验证所需的索赔。 (JWTClaimNotPresent)
- JWT 中声明值的格式与指定的配置格式不匹配。 (JWTClaimFormatInvalid)

HTTP 403：禁止访问

您配置了 Amazon WAF Web 访问控制列表 (Web ACL) 来监控对您的 Application Load Balancer 的请求，该列表阻止了请求。

HTTP 405：不允许的方法

客户端使用 TRACE 方法，而 Application Load Balancer 不支持该方法。

HTTP 408：请求超时

客户端在空闲超时期到期前未发送数据。发送 TCP keep-alive 不会阻止此超时。在每个空闲超时期过去之前发送至少 1 个字节的数据。根据需要增加空闲超时期长度。

HTTP 413：有效负载过大

可能的原因：

- 目标是 Lambda 函数，请求正文超过 1 MB。
- 请求标头超过了每个请求行 16K、单个标头 16K 或整个请求标头 64K 的限制。

HTTP 414：URI 太长

请求 URL 或查询字符串参数过大。

HTTP 460

负载均衡器收到了来自客户端的请求，但客户端在空闲超时期限结束前就关闭了与负载均衡器的连接。

检查客户端超时期限是否超过负载均衡器的空闲超时期限。确保目标在客户端超时期限之前向客户端提供响应，或增加客户端超时期限以匹配负载均衡器空闲超时 (如果客户端支持这样做)。

HTTP 463

负载均衡器收到一个包含多个 IP 地址的 X-Forwarded-For 请求标头。IP 地址的上限为 30。

HTTP 464

负载均衡器收到一个与目标组协议的版本配置不兼容的传入请求协议。

可能的原因：

- 请求协议是 HTTP/1.1，而目标组协议版本是 gRPC 或 HTTP/2。
- 请求协议是 gRPC，而目标组协议版本是 HTTP/1.1。
- 请求协议是 HTTP/2，请求不是 POST，而目标组协议版本是 gRPC。

HTTP 500：内部服务器错误

可能的原因：

- 您配置了 Amazon WAF Web 访问控制列表 (Web ACL)，但在执行 Web ACL 规则时出现错误。
- 负载均衡器无法与 IdP 令牌终端节点或 IdP 用户信息终端节点进行通信。
  - 确认 IdP 的 DNS 可公开解析。
  - 验证您的负载均衡器的安全组和 VPC 的网络是否允许 ACLs 对这些终端节点进行出站访问。
  - 验证您的 VPC 可以访问 Internet。如果您有面向内部的负载均衡器，请使用 NAT 网关启用 Internet 访问。
- 从 IdP 收到的用户声明大小超过 11KB。
- IdP 令牌端点或 IdP 用户信息端点需要超过 5 秒钟才能响应。
- 负载均衡器无法与 JWKS 端点通信，或者 JWKS 端点在 5 秒钟内没有响应。
- JWKS 端点返回的响应大小超过 150KB 或者 JWKS 端点返回的密钥数量超过 10

- 目标组启用了目标优化器，但代理遇到了意外错误。请参阅the section called “对目标优化器进行故障排除”。

HTTP 501：未实现

可能的原因：

- 负载均衡器收到具有不支持的值的 Transfer-Encoding 标头。Transfer-Encoding 支持的值为 `chunked` 和 `identity`。作为替代方案，您可以使用 Content-Encoding 标头。
- websocket 请求被路由到启用了目标优化器的目标组。

HTTP 502：无效网关

可能的原因：

- 负载均衡器在尝试建立连接时从目标收到了 TCP RST。
- 负载均衡器收到来自目标的意外响应，如当尝试建立连接时，收到“ICMP Destination unreachable (Host unreachable)”。检查是否允许来自负载均衡器子网的流量流至目标端口上的目标。
- 当负载均衡器具有目标的未完成请求时，目标关闭了具有 TCP RST 或 TCP FIN 的连接。检查目标的保持活动状态持续时间是否短于负载均衡器的空闲超时值。
- 目标响应格式错误，或者包含无效的 HTTP 标头。
- 整个响应标头的目标响应标头超过了 32 K。
- 对于已取消注册的目标正在处理的请求，取消注册延迟期已结束。增加延迟期，以便较长的操作能够完成。
- 目标是 Lambda 函数，响应正文超过 1 MB。
- 目标是一个 Lambda 函数，该函数在达到其配置的超时之前未响应。
- 目标是一个返回错误的 Lambda 函数，或是受到 Lambda 服务限制的函数。
- 负载均衡器在连接到目标时遇到 SSL 握手错误。

有关更多信息，请参阅 Amazon 支持知识中心中的如何排除 Application Load Balancer HTTP 502 错误。

HTTP 503：服务不可用

可能的原因：

- 负载均衡器的目标组没有任何已注册的目标，或者已注册的目标均处于 unused 状态。
- 该请求被路由到启用了目标优化器的目标组，但由于没有目标准备好接收请求而被拒绝。请参阅the section called “对目标优化器进行故障排除”。

HTTP 504：网关超时

可能的原因：

- 负载均衡器未能在连接超时到期 (10 秒) 之前建立与目标的连接。
- 负载均衡器与目标建立了连接，但在空闲超时周期到期之前未响应。
- 子网的网络 ACL 不允许临时端口 (1024-65535) 上从目标到负载均衡器节点的流量。
- 目标返回的 content-length 标头大于整个正文。负载均衡器因等待缺少的字节而超时。
- 目标是 Lambda 函数，并且 Lambda 服务在连接超时过期之前没有响应。
- 负载均衡器在连接到目标时遇到 SSL 握手超时（10 秒）。

HTTP 505：不支持版本

负载均衡器收到一个意外的 HTTP 版本请求。例如，负载均衡器建立了 HTTP/1 连接，但收到了 HTTP/2 请求。

HTTP 507：存储空间不足

重定向 URL 过长。

HTTP 561: 未授权

已配置侦听器规则以验证用户的身份，但在验证用户身份时，IdP 返回错误代码。查看访问日志以获取相关的错误原因代码。

HTTP 562：JWKS 请求失败

负载均衡器未能从 JWKS（JSON Web 密钥集）端点收到成功且有效的响应。成功响应的状态码应在 200-299 范围内，但收到的状态码却不同。有效的回复不应存在以下问题：

- 非 JSON 格式

- 无效字符
- JWKS 格式无效
- 必填的 JWKS 属性缺失/无效
- 公钥有不支持的算法
- 无法将公钥转换为解码密钥
- 公钥大小不是 2K

目标生成 HTTP 错误

负载均衡器将有效的 HTTP 响应从目标转发到客户端，包括 HTTP 错误。目标生成的 HTTP 错误记录在 `HTTPCode_Target_4XX_Count` 和 `HTTPCode_Target_5XX_Count` 指标中。

Amazon Certificate Manager 证书不可用

决定将 HTTPS 侦听器与 Application Load Balancer 配合使用时， Amazon Certificate Manager 需要您在颁发证书之前验证域所有权。如果在安装过程中错过了此步骤，则证书将保持 `Pending Validation` 状态，直到验证后才能使用。

- 如果使用电子邮件验证，请参阅《Amazon Certificate Manager 用户指南》中的电子邮件验证。
- 如果使用 DNS 验证，请参阅《Amazon Certificate Manager 用户指南》中的 DNS 验证。

不支持多行标头

应用程序负载均衡器不支持多行标头，包括 `message/http` 媒体类型标头。如果提供了多行标头，应用程序负载均衡器会在将其传递给目标之前附加冒号字符“`:`”。

使用资源地图排查不正常目标的问题

如果您的应用程序负载均衡器目标未通过运行状况检查，则可以使用资源地图来查找不正常目标并根据失败原因代码采取措施。有关更多信息，请参阅 查看应用程序负载均衡器资源地图。

资源地图提供了两个视图：概述和不正常目标地图。默认情况下，概述处于选中状态，并显示您的负载均衡器的所有资源。选择不正常目标地图视图将仅显示与应用程序负载均衡器关联的每个目标组中的不正常目标。

> **Note**
>
> 必须启用显示资源详细信息才能查看资源地图中所有适用资源的运行状况检查摘要和错误消息。未启用时，您必须选择每个资源才能查看其详细信息。

目标组列显示每个目标组的正常目标和不正常目标的摘要。这可以帮助确定是所有目标都未通过运行状况检查，还是只有特定目标未通过运行状况检查。如果目标组中的所有目标均未通过运行状况检查，请检查目标组的配置。选择目标组名称以在新选项卡中打开其详细信息页面。

目标列显示每个目标的目标 ID 和当前运行状况检查状态。当目标运行状况不佳时，将显示运行状况检查失败原因代码。当单个目标未通过运行状况检查时，请验证目标是否有足够的资源，并确认目标上运行的应用程序是否可用。选择一个目标 ID 以将在新选项卡中打开其详细信息页面。

选择导出后，您可以选择将应用程序负载均衡器资源地图的当前视图导出为 PDF。

验证您的实例是否未通过运行状况检查，然后根据失败原因代码检查是否存在以下问题：

- 不正常：HTTP 响应不匹配
  - 验证目标上运行的应用程序是否正在向应用程序负载均衡器的运行状况检查请求发送正确的 HTTP 响应。
  - 或者，您可以更新应用程序负载均衡器的运行状况检查请求，以匹配目标上运行的应用程序的响应。
- 不正常：请求超时
  - 验证与您的目标和应用程序负载均衡器关联的安全组和网络访问控制列表（ACL）未阻止连接。
  - 验证目标是否具有足够的资源来接受来自应用程序负载均衡器的连接。
  - 验证目标上运行的任何应用程序的状态。
  - 可以在每个目标的应用程序日志中查看应用程序负载均衡器的运行状况检查响应。有关更多信息，请参阅 Health check reason codes。
- 不健康： FailedHealthChecks
  - 验证目标上运行的任何应用程序的状态。
  - 验证目标是否在侦听运行状况检查端口上的流量。

> **使用 HTTPS 侦听器时**
>
> 您可以选择用于前端连接的安全策略。用于后端连接的安全策略是根据正在使用的前端安全策略自动选择的。如果你的听众有：

- FIPS 后量子 TLS 策略-后端连接使用 ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09
- FIPS 策略-后端连接使用 ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04
- 后量子 TLS 策略-后端连接使用 ELBSecurityPolicy-TLS13-1-0-PQ-2025-09
- TLS 1.3 政策-后端连接使用 ELBSecurityPolicy-TLS13-1-0-2021-06
- 后端连接使用的所有其他 TLS 策略 ELBSecurityPolicy-2016-08

有关更多信息，请参阅安全策略。

- 验证目标是否按照安全策略指定的正确格式提供服务器证书和密钥。
- 验证目标是否支持一个或多个匹配的密码，以及应用程序负载均衡器提供的用于建立 TLS 握手的协议。

对目标优化器进行故障排除

有关详细监控，请参阅目标优化器指标

配置错误

- HTTPCode_ELB_502_Count：负载均衡器在尝试建立连接时收到了来自代理的 TCP RST。
- HTTPCode_ELB_504_Count：在空闲超时时间到期之前，负载均衡器未能与代理建立连接。
- HTTPCode_Target_5XX_Count：代理在尝试建立连接时收到了来自目标应用程序的 TCP RST。（仅在目标应用程序本身未生成此错误响应时适用。）

要修复这些问题，请确保：

- 目标上的安全组配置正确。
- 代理正在按预期配置运行。
- 目标应用程序正在运行并监听代理中配置的 TARGET_CONTROL_DESTINATION_ADDRESS。

服务不可用错误 (**HTTPCode_ELB_503_Count**)

一致的 HTTP 503 错误意味着没有足够的目标来接收 ALB 的请求。该 TargetControlRequestRejectCount 指标代表了这些被拒绝的请求。该 TargetControlWorkQueueLength 指标将降至接近零的值。要解决此问题，请考虑：

- 增加目标数量
- 将代理上的 TARGET_CONTROL_MAX_CONCURRENCY 变量设置为更大的值。

运行状况检查错误

- 如果运行状况检查端口与 TARGET_CONTROL_DATA_ADDRESS 相同，则来自ALB的运行状况检查请求将通过代理发送到目标应用程序。如果运行状况检查失败（由于 HTTP 502 或超时），请参阅 “配置错误” 部分。

Application Load Balancer 的配额

您的 Amazon 账户对每项 Amazon 服务都有默认配额（以前称为限制）。除非另有说明，否则，每个限额是区域特定的。您可以请求增加某些配额，但其他一些配额无法增加。

要查看 Application Load Balancer 的配额，请打开 Service Quotas 控制台。在导航窗格中，选择 Amazon services，然后选择 Elastic Load Balancing。您也可以使用 describe-account-limits(Amazon CLI) 命令进行 Elastic Load Balancing。

要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。如果配额在“服务配额”中尚不可用，请提交服务配额增加请求。

配额

- 负载均衡器
- 目标组
- Rules
- 信任存储
- 证书
- HTTP 标头
- 负载均衡器容量单位

负载均衡器

您的 Amazon 账户具有以下与应用程序负载均衡器相关的配额。

Name	默认值	可调整
每个区域的 Application Load Balancer 数。	50	是
每个应用程序负载负载均衡器的证书数（不包括默认证书）	25	是
每个 Application Load Balancer 的侦听器数	50	是
每个 Application Load Balancer 每个操作的目标组数	5	否

Name	默认值	可调整
每个 Application Load Balancer 的目标组数	100	否
每个 Application Load Balancer 的目标数	1000	是

目标组

以下配额适用于目标组。

Name	默认值	可调整
每个区域的目标组数	3,000 *	是
每个区域每个目标组的目标数（实例或 IP 地址）	1000	是
每个区域每个目标组的目标数（Lambda 函数）	1	否
每个目标组的负载均衡器	1	否

* 此配额由 Application Load Balancer 和 Network Load Balancer 共享。

Rules

以下配额适用于规则。

Name	默认值	可调整
每个应用程序负载均衡器的规则数（不包括默认规则）	100	是
每个规则的条件值	5	否
每个规则的条件通配符	6	否
每条规则的匹配评估数	5	否

信任存储

以下配额适用于信任存储。

Name	默认值	可调整
每个账户的信任存储数	20	是
每个负载均衡器在验证模式下使用 mTLS 的侦听器数量。	2	否

证书

以下配额适用于证书，包括广告 CA 证书名称和证书吊销列表。

Name	默认值	可调整
CA 证书大小	16 KB	否
每个信任存储的 CA 证书数	25	是
每个信任存储的 CA 证书主题大小	10000	是
最大证书链深度	4	否
每个信任存储的吊销条目	500,000	是
吊销列表文件大小	50 MB	否
每个信任存储的吊销列表	30	是
TLS 消息大小	64K	否

HTTP 标头

HTTP 标头具有以下大小限制：

Name	默认值	可调整
请求行	16K	否
单个标头	16K	否
整个响应标头	32 K	否
整个请求标头	64K	否

负载均衡器容量单位

以下配额适用于负载均衡器容量单位（LCU）。

Name	默认值	可调整
每个应用程序负载均衡器的预留应用程序负载均衡器容量单位 (LCUs)	15000	是
每个区域的预留应用程序负载均衡器容量单位（LCU）	0	是

Application Load Balancer 的文档历史记录

下表介绍了 Application Load Balancer 的版本。

变更	说明	日期
访问令牌验证	此版本增加了对负载均衡器的支持，以验证客户端为安全服务到服务 (S2S) 或机器对机器 (M2M) 通信而提供的JSON Web令牌 (JWT)。	2025 年 11 月 21 日
转换	此版本增加支持在负载均衡器将流量路由到目标之前转换传入请求的主机标头和 URL。	2025 年 10 月 15 日
访问日志和连接日志的存储桶策略	在此版本之前，您使用的存储桶策略取决于该区域在 2022 年 8 月之前还是之后可用。在此版本中，所有区域都支持新版存储桶策略。请注意，仍然支持旧版存储桶策略。	2025 年 9 月 10 日
HTTP 标头修改	此版本增加支持修改所有响应代码的 HTTP 标头。在此之前，此功能仅限于响应代码 2XX 和 3XX。	2025 年 2 月 28 日
容量单位预留	此版本增加了对为负载均衡器设置最小容量的支持。	2024 年 11 月 20 日
资源地图	此版本增加了对以可视化格式查看负载均衡器资源和关系的支持。	2024 年 3 月 8 日
一键式 WAF	此版本增加了对配置负载均衡器行为的支持，前提是它只	2024 年 2 月 6 日

	需单击一下即可集成 Amazon WAF。	
双向 TLS	此版本增加了对双向 TLS 身份验证的支持。	2023 年 11 月 26 日
自动目标权重	此版本增加了对自动目标权重算法的支持。	2023 年 11 月 26 日
FIPS 140-3 TLS 终止	此版本添加了在终止 TLS 连接时使用 FIPS 140-3 加密模块的安全策略。	2023 年 11 月 20 日
使用 IPv6 注册目标	此版本增加了对在通过 IPv6 寻址时将实例注册为目标的支持。	2023 年 10 月 2 日
支持 TLS 1.3 的安全策略	此版本增加了对 TLS 1.3 预定义安全策略的支持。	2023 年 3 月 22 日
可用区转移	此版本增加了对通过与 Amazon 应用程序恢复控制器 (ARC)的集成将流量从单个受损可用区路由出去的支持。	2022 年 11 月 28 日
关闭跨区域负载均衡	此版本增加了对关闭跨区域负载均衡的支持。	2022 年 11 月 28 日
目标组运行状况	此版本增加了对配置必须运行状况良好的目标数量下限或最低百分比以及在未达到阈值时负载均衡器采取哪些操作的支持。	2022 年 11 月 28 日
Cross-zone 负载平衡	此版本增加了对在目标组级别配置跨区域负载均衡的支持。	2022 年 11 月 17 日
IPv6 目标组	此版本支持为 Application Load Balancer 配置 IPv6 目标组。	2021 年 11 月 23 日

IPv6 内部负载均衡器	此版本支持为 Application Load Balancer 配置 IPv6 目标组。	2021 年 11 月 23 日
Amazon PrivateLink 和静态 IP 地址	此版本通过将流量直接从网络负载均衡器转发到应用程序负载均衡器，增加了对使用 Amazon PrivateLink 和公开静态 IP 地址的支持。	2021 年 9 月 27 日
保留客户端端口	此版本增加一个属性，用于保留客户端与您的负载均衡器连接时所用的源端口。	2021 年 7 月 29 日
TLS 标头	此版本添加了一个属性，用于指示在将客户端请求发送到目标之前，已将包含有关协商的 TLS 版本和密码套件的信息的 TLS 标头添加到客户端请求中。	2021 年 7 月 21 日
额外的 ACM 证书	此版本支持具有 2048、3072 和 4096 位密钥长度的 RSA 证书，以及所有 ECDSA 证书。	2021 年 7 月 14 日
Application-based 粘性	此版本添加了基于应用程序的 Cookie 来支持负载均衡器的粘性会话。	2021 年 2 月 8 日
支持 TLS 1.2 版的 FS 安全策略	此版本增加了支持 TLS 1.2 版的向前保密 (FS) 安全策略。	2020 年 11 月 24 日
WAF 失败时开放支持	如果您的负载均衡器与集成，则此版本增加了对配置负载均衡器的行为的支持 Amazon WAF。	2020 年 11 月 13 日
gRPC 和支持 HTTP/2	此版本增加了对 gRPC 工作负载和端到端的支持。 HTTP/2	2020 年 10 月 29 日

Outpost 支持	您可以在 Amazon Outposts上预置应用程序负载均衡器。	2020 年 9 月 8 日
异步缓解模式	此版本增加了对异步缓解模式的支持。	2020 年 8 月 17 日
最少未完成请求	此版本支持最少未完成请求算法。	2019 年 11 月 25 日
加权目标组	此版本增加了对使用多个目标组转发操作的支持。请求将根据您为每个目标组指定的权重分配给这些目标组。	2019 年 11 月 19 日
New attribute	此版本增加了对 routing.http.drop_invalid_header_fields.enabled 属性的支持。	2019 年 11 月 15 日
FS 的安全策略	此版本增加了对三个额外预定义向前保密安全策略的支持。	2019 年 10 月 8 日
高级请求路由	此版本增加了对侦听器规则的其他条件类型的支持。	2019 年 3 月 27 日
Lambda 函数作为目标	此版本增加了将 Lambda 函数注册为目标的支持。	2018 年 11 月 29 日
重定向操作	此版本增加了对负载均衡器的支持，以将请求重定向到其他 URL。	2018 年 7 月 25 日
Fixed-response actions	此版本增加了对负载均衡器的支持，以返回自定义 HTTP 响应。	2018 年 7 月 25 日
用于 FS 和 TLS 1.2 的安全策略	此版本支持两种额外的预定义安全策略。	2018 年 6 月 6 日

用户身份验证	此版本支持负载均衡器在路由请求之前使用应用程序用户的企业或社交身份对这些用户进行身份验证。	2018 年 5 月 30 日
Resource-level permissions	此版本支持资源级权限和标记条件键。	2018 年 5 月 10 日
慢启动模式	此版本增加了对慢启动模式的支持，这种模式会在新注册的目标预热时，逐渐增加负载均衡器向此目标发送的请求份额。	2018 年 3 月 24 日
SNI 支持	此版本增加了对服务器名称指示 (SNI) 的支持。	2017 年 10 月 10 日
IP 地址即目标	此版本增加了将 IP 地址注册为目标的支持。	2017 年 8 月 31 日
Host-based 路由	此版本支持根据主机标头中的主机名路由请求。	2017 年 4 月 5 日
TLS 1.1 和 TLS 1.2 的安全策略	此版本增加了用于 TLS 1.1 和 TLS 1.2 的安全策略。	2017 年 2 月 6 日
IPv6 支持	此版本增加了对 IPv6 地址的支持。	2017 年 1 月 25 日
请求跟踪	此版本增加了对请求跟踪的支持。	2016 年 11 月 22 日
该 TargetResponseTime 指标的百分位数支持	此版本增加了对 Amazon CloudWatch 支持的新百分位统计数据的支持。	2016 年 11 月 17 日
新负载均衡器类型	此版本的 Elastic Load Balancing 引入了 Application Load Balancer。	2016 年 8 月 11 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。