aws

用户指南

Elastic Load Balancing

Elastic Load Balancing: 用户指南

Table of Contents

什么是 Elastic Load Balancing？

Elastic Load Balancing 会自动将您的传入流量分配到一个或多个可用区域中的多个目标，例如 EC2 实例、容器和 IP 地址。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。弹性负载均衡将会扩展负载均衡器容量，以响应传入流量中的变化。

负载均衡器优势

负载均衡器跨多个计算资源 (如虚拟服务器) 分布工作负载。使用负载均衡器可提高您的应用程序的可用性和容错性。

可以根据需求变化在负载均衡器中添加和删除计算资源，而不会中断应用程序的整体请求流。

您可以配置运行状况检查，这些检查监控计算资源的运行状况，以便负载均衡器只将请求发送到正常运行的目标。此外，您可以将加密和解密的工作交给负载均衡器完成，以使您的计算资源能够专注于完成主要工作。

Elastic Load Balancing 的功能

弹性负载均衡支持多种负载均衡器类型。您可以选择最适合自己需求的负载均衡器类型。有关更多信息，请参阅弹性负载均衡功能。

有关当前一代负载均衡器的更多信息，请参阅以下文档：

- 适用于应用程序负载均衡器的用户指南
- 适用于网络负载均衡器的用户指南
- 网关负载均衡器用户指南

经典负载均衡器是 Elastic Load Balancing 的上一代负载均衡器。建议您迁移到当前一代负载均衡器。有关更多信息，请参阅迁移您的经典负载均衡器。

访问 Elastic Load Balancing

可以使用以下任意接口创建、访问和管理负载均衡器：

- AWS 管理控制台— 提供可用于访问 Elastic Load Balancing 的 Web 界面。

- AWS 命令行界面 (AWS CLI) — 为包括 Elastic Load Balancing 在内的各种 AWS 服务提供命令。在 AWS CLI Windows、macOS 和 Linux 上都支持。有关更多信息，请参阅 AWS Command Line Interface。
- AWS SDKs— 提供特定语言 APIs 并处理许多连接细节，例如计算签名、处理请求重试和错误处理。有关更多信息，请参阅 AWS SDKs。
- 查询 API — 提供您使用 HTTPS 请求调用的低级别 API 操作。使用查询 API 是访问 Elastic Load Balancing 的最直接方式。但是，查询 API 需要您的应用程序处理低级别的详细信息，例如生成哈希值以签署请求以及进行错误处理。有关更多信息，请参阅下列内容：
  - 应用程序负载均衡器、网络负载均衡器和网络负载均衡器 — API 版本 2015-12-01
  - 经典负载均衡器— API 版本 2012-06-01

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。有关更多信息，请参阅 Amazon EC2 用户指南。
- Amazon A EC2 uto Scaling — 确保即使实例出现故障，您也能运行所需数量的实例。Amazon A EC2 uto Scaling 还允许您在实例需求变化时自动增加或减少实例数量。如果通过 Elastic Load Balancing 启用 Auto Scaling，则由 Auto Scaling 启动的实例将自动注册到负载均衡器。同样，由 Auto Scaling 终止的实例将自动从负载均衡器取消注册。有关更多信息，请参阅 Amazon A EC2 uto Scaling 用户指南。
- AWS 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。
- Amazon CloudWatch — 使您能够监控负载均衡器并根据需要采取行动。有关更多信息，请参阅 Amazon CloudWatch 用户指南。
- 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。
- AWS 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 AWS 区域的多个负载均衡器之间分配流量。有关更多信息，请参阅 AWS Global Accelerator 开发人员指南。
- Route 53 — 通过将域名转换为计算机相互连接所用的数字 IP 地址，以一种可靠且经济的方式将访问者路由至网站。例如，它将`www.example.com`转换为数字 IP 地址`192.0.2.1`。 AWS 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。

- AWS WAF— 您可以 AWS 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，请参见AWS WAF 开发人员指南。

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。

Elastic Load Balancing 的工作原理

负载均衡器接受来自客户端的传入流量，并将请求路由到其在一个或多个可用区中的注册目标（例如 EC2 实例）。负载均衡器还会监控已注册目标的运行状况，并确保它只将流量路由到正常运行的目标。当负载均衡器检测到不正常目标时，它会停止将流量路由到该目标。然后，当它检测到目标再次正常时，它会恢复将流量路由到该目标。

您可通过指定一个或多个侦听器将您的负载均衡器配置为接受传入流量。侦听器是用于检查连接请求的进程。它配置了用于从客户端连接到负载均衡器的协议和端口号。同样，它配置了用于从负载均衡器连接到目标的协议和端口号。

内容

可用区与负载均衡器节点

当您为负载均衡器启用可用区时，Elastic Load Balancing 会在该可用区中创建一个负载均衡器节点。如果您在可用区中注册目标但不启用可用区，这些已注册目标将无法接收流量。当您确保每个启用的可用区均具有至少一个已注册目标时，负载均衡器将具有最高效率。

我们建议为所有负载均衡器启用多个可用区。但对于 Application Load Balancer，要求您至少启用两个或更多可用区。此配置有助于确保负载均衡器可以继续路由流量。如果一个可用区变得不可用或没有正常目标，则负载均衡器会将流量路由到其他可用区中的正常目标。

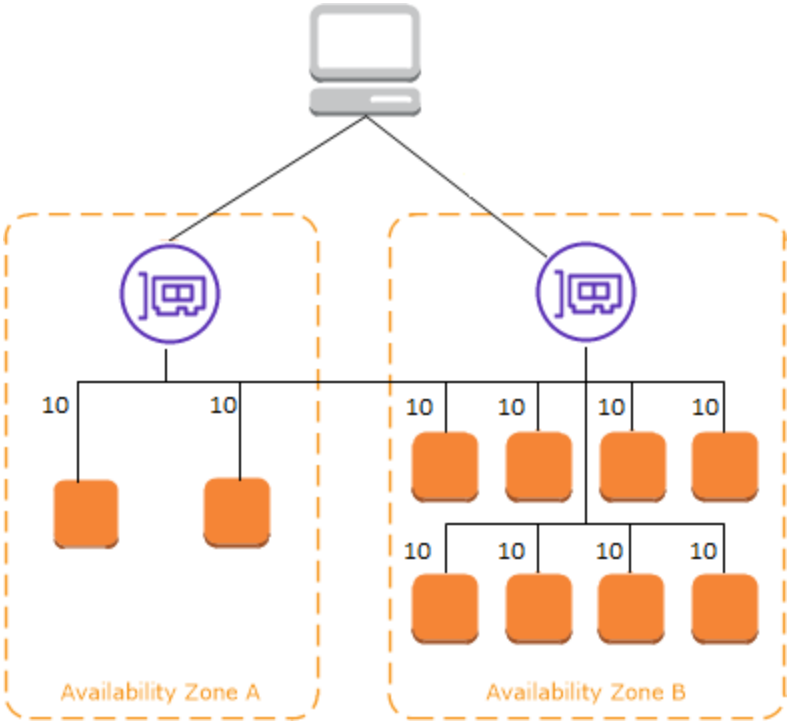
在禁用一个可用区后，该可用区中的目标将保持已注册到负载均衡器的状态。但是，即使它们保持已注册状态，负载均衡器也不会将流量路由到它们。

跨区域负载均衡

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。启用了跨区域负载均衡后，每个负载均衡器节点会在所有启用的可用区中的已注册目标之间分配流量。禁用了跨区域负载均衡后，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。

下图演示了以轮询为默认路由算法的跨可用区负载均衡效果。有 2 个已启用的可用区，其中可用区 A 中有 2 个目标，可用区 B 中有 8 个目标。客户端发送请求，Amazon Route 53 使用负载均衡器节点之一的 IP 地址响应每个请求。基于轮询路由算法，系统会分配流量，以便每个负载均衡器节点接收来自客户端 50% 的流量。每个负载均衡器节点会在其范围中的已注册目标之间分配其流量份额。

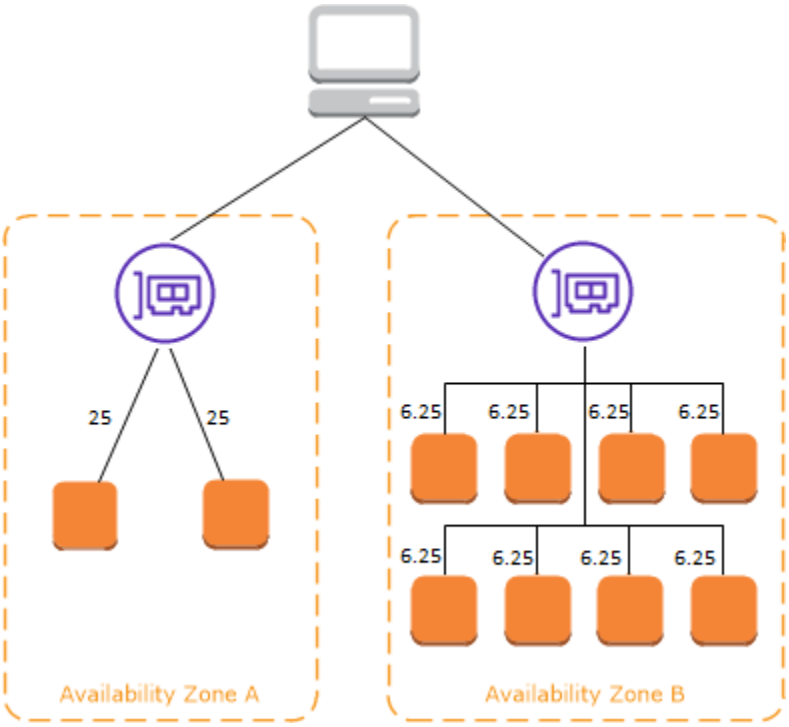
如果启用了跨区域负载均衡，则 10 个目标中的每个目标接收 10% 的流量。这是因为每个负载均衡器节点可将其 50% 的客户端流量路由到所有 10 个目标。



如果禁用了跨区域负载均衡：

- 可用区 A 中的两个目标中的每个目标接收 25% 的流量。
- 可用区 B 中的八个目标中的每个目标接收 6.25% 的流量。

这是因为每个负载均衡器节点只能将其 50% 的客户端流量路由到其可用区中的目标。



对于应用程序负载均衡器，跨可用区负载均衡始终在负载均衡器级别启用。在目标组级别，可以禁用跨可用区负载均衡。有关更多信息，请参阅《应用程序负载均衡器用户指南》中的关闭跨可用区负载均衡。

对于 Network Load Balancer 和 Gateway Load Balancer，默认情况下会禁用跨区域负载均衡。创建负载均衡器后，您随时可以启用或禁用跨区域负载均衡。有关更多信息，请参阅《网络负载均衡器用户指南》中的 Cross-zone load balancing。

在创建经典负载均衡器时，跨区域负载均衡的默认值取决于创建负载均衡器的方式。默认情况下，使用 API 或 CLI 时将禁用跨区域负载均衡。使用时 AWS 管理控制台，默认情况下会选择启用跨区域负载平衡的选项。创建经典负载均衡器后，您随时可以启用或禁用跨区域负载均衡。有关更多信息，请参阅《经典负载均衡器用户指南》中的启用跨区域负载均衡。

可用区转移

可用区转移是 Amazon 应用程序恢复控制器（ARC）中的一项功能。通过可用区转移，只需执行一次操作即可将负载均衡器资源从受损的可用区转移出去。这样，您就可以继续从 AWS 区域中的其他运行状况良好的可用区运行。

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这些资源的流量。ARC 会立即创建可用区转移。但是，可能需要很短时间（通常长达几分钟）才能完成受影响可用区中正在进行的现有连

接。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 How a zonal shift works: health checks and zonal IP addresses。

在使用可用区转移之前，请查看以下内容：

- 当您在已开启或关闭跨区域负载均衡的情况下使用网络负载均衡器时，不支持可用区转移。
- 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。
- AWS 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。

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

请求路由

在客户端将请求发送到负载均衡器之前，它会利用域名系统 (DNS) 服务器解析负载均衡器的域名。DNS 条目由 Amazon 控制，因为您的负载均衡器位于 amazonaws.com 域中。Amazon DNS 服务器会将一个或多个 IP 地址返回到客户端。这些是您的负载均衡器的负载均衡器节点的 IP 地址。对于网络负载均衡器，Elastic Load Balancing 将为您启用的每个可用区创建一个网络接口，并使用该网络接口来获取静态 IP 地址。在您创建网络负载均衡器时，可以选择将一个弹性 IP 地址关联到每个网络接口。

当流向应用程序的流量随时间变化时，Elastic Load Balancing 会扩展负载均衡器并更新 DNS 条目。DNS 条目还指定了 60 秒的 time-to-live (TTL)。这有助于确保可以快速重新映射 IP 地址以响应不断变化的流量。

客户端可以确定使用哪个 IP 地址将请求发送到负载均衡器。用于接收请求的负载均衡器节点会选择一个正常运行的已注册目标，并使用其私有 IP 地址将请求发送到该目标。

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。

路由算法

借助 Application Load Balancer，接收请求的负载均衡器节点使用以下过程：

1. 按优先级顺序评估侦听器规则以确定要应用的规则。
2. 使用为目标组配置的路由算法，从目标组中为规则操作选择目标。默认路由算法是轮询。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。

借助 Network Load Balancer，接收连接的负载均衡器节点使用以下过程：

1. 使用流哈希算法从目标组中为默认规则选择目标。它使算法基于：
   - 协议
   - 源 IP 地址和源端口
   - 目标 IP 地址和目标端口
   - TCP 序列号
2. 将每个单独的 TCP 连接在连接的有效期内路由到单个目标。来自客户端的 TCP 连接具有不同的源端口和序列号，可以路由到不同的目标。

使用网关负载均衡器时，接收连接的负载均衡器节点将使用五元组流哈希算法来选择目标设备。建立流后，同一流中的所有数据包都将始终路由到同一目标设备。负载均衡器和目标设备使用 GENEVE 协议通过端口 6081 交换流量。

借助 经典负载均衡器，接收请求的负载均衡器节点按照以下方式选择注册实例：

- 使用适用于 TCP 侦听器的轮询路由算法
- 使用适用于 HTTP 和 HTTPS 侦听器的最少未完成请求路由算法

HTTP 连接

经典负载均衡器会使用预打开连接，但 Application Load Balancer 不会使用预打开连接。经典负载均衡器和 Application Load Balancer 均使用多路复用连接。也就是说，来自多个前端连接上的多个客户端的请求可通过单一的后端连接路由到指定目标。多路复用连接可缩短延迟并减少您的应用程序上的负载。要禁止多路复用连接，请在您的 HTTP 响应中设置 `Connection: close` 标头来禁用 HTTP `keep-alive` 标头。

对于前端连接，Application Load Balancer 和经典负载均衡器支持管道化 HTTP。对于后端连接它们均不支持管道化 HTTP。

应用程序负载均衡器支持以下 HTTP 请求方法：GET、HEAD、POST、PUT、DELETE、OPTIONS 和 PATCH。

对于前端连接，Application Load Balancer 支持以下协议：HTTP/0.9、HTTP/1.0、HTTP/1.1 和 HTTP/2。HTTP/2 仅适用于 HTTPS 侦听器，使用一个 HTTP/2 连接最多可并行发送 128 个请求。应用程序负载均衡器还支持从 HTTP 升级到。 WebSockets但是，如果连接升级，Application Load Balancer 侦听器路由规则和 AWS WAF 集成将不再适用。

默认情况下，Application Load Balancer 在后端连接上使用 HTTP/1.1（负载均衡器连接到已注册的目标）。但是，您可以通过协议版本使用 HTTP/2 或 gRPC 将请求发送到目标。有关更多信息，请参阅协议版本。默认情况下，keep-alive 标头在后端连接上受支持。如果 HTTP/1.0 请求来自没有主机标头的客户端，负载均衡器会对后端连接发送的 HTTP/1.1 请求生成一个主机标头。主机标头包含负载均衡器的 DNS 名称。

对于前端连接（客户端到负载均衡器），经典负载均衡器支持以下协议：HTTP/0.9、HTTP/1.0 和 HTTP/1.1。默认情况下，它们在后端连接（已注册目标的负载均衡器）上使用 HTTP/1.1。默认情况下，keep-alive 标头在后端连接上受支持。如果 HTTP/1.0 请求来自没有主机标头的客户端，负载均衡器会对后端连接发送的 HTTP/1.1 请求生成一个主机标头。主机标头包含负载均衡器节点的 IP 地址。

HTTP 标头

Application Load Balancer 和经典负载均衡器会将 X-Forwarded-For、X-Forwarded-Proto 和 X-Forwarded-Port 标头自动添加到请求。

应用程序负载均衡器将 HTTP 主机标头中的主机名转换为小写，然后再将其发送到目标。

对于使用 HTTP/2 的前端连接，标头名称是小写的。使用 HTTP/1.1 将请求发送到目标之前，以下标头名称将转换为混合大小写：X-Forwarded-For、X-Forwarded-Proto、X-Forwarded-Port、Host、X-Amzn-Trace-Id、Upgrade 和 Connection。所有其他标头名称是小写的。

Application Load Balancer 和经典负载均衡器将响应代理返回客户端后，遵守来自传入客户端请求的连接标头。

当使用 HTTP/1.1 的应用程序负载均衡器和经典负载均衡器收到 Expect: 100-Continue 标头时，它们会立即以 HTTP/1.1 100 Continue 响应，而不会测试内容长度标头。Expect: 100-Continue 请求标头不会转发到其目标。

使用 HTTP/2 时，应用程序负载均衡器不支持来自客户端请求的 Expect: 100-Continue 标头。应用程序负载均衡器不会以 HTTP/2 100 Continue 响应，也不会将此标头转发给其目标。

HTTP 标头限制

应用程序负载均衡器的以下大小限制是无法更改的硬限制：

- 请求行：16K
- 单个标头：16K

- 整个响应标头：32 K
- 整个请求标头：64 K

负载均衡器模式

在创建负载均衡器时，您必须选择使其成为内部负载均衡器还是面向 Internet 的负载均衡器。

面向 Internet 的负载均衡器的节点具有公共 IP 地址。面向 Internet 的负载均衡器的 DNS 名称可公开解析为节点的公共 IP 地址。因此，面向 Internet 的负载均衡器可以通过 Internet 路由来自客户端的请求。

内部负载均衡器的节点只有私有 IP 地址。内部负载均衡器的 DNS 名称可公开解析为节点的私有 IP 地址。因此，内部负载均衡器可路由的请求只能来自对负载均衡器的 VPC 具有访问权限的客户端。

面向 Internet 的负载均衡器和内部负载均衡器均使用私有 IP 地址将请求路由到您的目标。因此，您的目标无需使用公有 IP 地址从内部负载均衡器或面向 Internet 的负载均衡器接收请求。

如果您的应用程序具有多个层，则可以设计一个同时使用内部负载均衡器和面向 Internet 的负载均衡器的架构。例如，如果您的应用程序使用必须连接到 Internet 的 Web 服务器，以及仅连接到 Web 服务器的应用程序服务器，则可以如此。创建一个面向 Internet 的负载均衡器并向其注册 Web 服务器。创建一个内部负载均衡器并向它注册应用程序服务器。Web 服务器从面向 Internet 的负载均衡器接收请求，并将对应用程序服务器的请求发送到内部负载均衡器。应用程序服务器从内部负载均衡器接收请求。

IP 地址类型

您为负载均衡器指定的 IP 地址类型决定了客户端如何与负载均衡器通信。

- IPv4 仅限 — 客户端使用公用地址和私有 IPv4 地址进行通信。您为负载均衡器选择的子网必须具有 IPv4 地址范围。
- Dualstack — 客户端使用公共和私有地址和地址进行通信 IPv4 。 IPv6 您为负载均衡器选择的子网必须具有 IPv4 和 IPv6 地址范围。
- 不带公共地址的 Dualstack IPv4 — 客户端使用公有和私有地址以及私有 IPv6 地址进行通信。 IPv4 您为负载均衡器选择的子网必须具有 IPv4 和 IPv6 地址范围。`internal` 负载均衡器方案不支持此选项。

下表描述了每种负载均衡器类型支持的 IP 地址类型。

负载均衡器类型	IPv4 只有	双堆栈	没有公开的双重堆栈 IPv4
应用程序负载均衡器	是	是	是
Network Load Balancer	是	是	没有
Gateway Load Balancer	是	是	没有
Classic 负载均衡器	是	没有	没有

您为目标组指定的 IP 地址类型决定了负载均衡器如何与目标通信。

- IPv4 仅限 — 负载均衡器使用私有 IPv4 地址进行通信。必须将带有 IPv4 地址的目标注册到 IPv4 目标组。
- IPv6 仅限 — 负载均衡器使用 IPv6 地址进行通信。必须将带有 IPv6 地址的目标注册到 IPv6 目标组。目标组必须与双堆栈负载均衡器一起使用。

下表描述了每个目标组协议支持的 IP 地址类型。

目标组协议	IPv4 只有	IPv6 只有	
HTTP 和 HTTPS	是	是	
TCP	是	是	
TLS	是	是	
UDP 和 TCP_UDP	是	是	

目标组协议	IPv4 只有	IPv6 只有	
GENEVE	-	-	

您的负载均衡器的网络 MTU

最大传输单位（MTU）决定了可以通过网络发送的最大数据包大小（以字节为单位）。连接的 MTU 越大，可在单个数据包中传递的数据越多。以太网帧由数据包（即您发送的实际数据）以及相关网络开销信息组成。通过互联网网关发送的流量具有 1500 的 MTU。这意味着，如果数据包超过 1500 字节，则将其分段以使用多个帧发送，或者如果在 IP 标头中设置 `Don't Fragment`，则将其丢弃。

负载均衡器节点上的 MTU 大小不可配置。Jumbo 帧 (9001 MTU) 在应用程序负载均衡器、网络负载均衡器和经典负载均衡器的负载均衡器节点中是标准的。网关负载均衡器支持 8500 MTU。有关更多信息，请参阅网关负载均衡器用户指南中的最大传输单位 (MTU)。

路径 MTU 是原始主机和接收主机之间的路径所支持的最大数据包大小。路径 MTU 发现 (PMTUD) 用于确定两台设备之间的路径 MTU。如果客户端或目标不支持巨型帧，路径 MTU 发现特别重要。

如果主机发送一个大于接收主机的 MTU 或大于路径上某台设备的 MTU 的数据包，则接收主机或设备将丢弃此数据包，然后返回以下 ICMP 消息：`Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4)`。这将指示传输主机将有效负载拆分为多个较小的数据包，并重新传输。

如果继续丢弃大于客户端或目标接口 MTU 大小的数据包，则可能是路径 MTU 发现 (PMTUD) 不起作用。为了避免这种情况，请确保路径 MTU 发现端到端工作，并且您已在客户端和目标上启用了巨型帧。有关 Path MTU 发现和启用巨型帧的更多信息，请参阅 A ma EC2 zon 用户指南中的 Path MTU 发现。

Elastic Load Balancing 入门

弹性负载均衡支持多种负载均衡器类型。您可以选择最适合自己需求的负载均衡器类型。有关更多信息，请参阅弹性负载均衡功能。

负载均衡器

- 创建应用程序负载均衡器
- 创建网络负载均衡器
- 创建网关负载均衡器

有关常见负载均衡器配置的演示，请参阅 Elastic Load Balancing 演示。

如果您有现有的经典负载均衡器，则可以迁移到 Application Load Balancer 或 Network Load Balancer。有关更多信息，请参阅 迁移您的经典负载均衡器。

Elastic Load Balancing 中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 Elastic Load Balancing 的合规计划，请参阅按合规计划划分的范围内的AWSAWS 服务服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其它因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

此文档将帮助您了解如何在使用 Elastic Load Balancing 时应用责任共担模式。其中说明了如何配置 Elastic Load Balancing 以实现您的安全性和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 Elastic Load Balancing 资源。

对于网关负载均衡器，您要负责从设备供应商那里选择和鉴定软件。您必须信任设备软件才能检查或修改来自负载均衡器的流量，负载均衡器在开放系统互连 (OSI) 模型的第 3 层（网络层）运行。被列为 Elastic Load Balancing Par tners 的设备供应商已将其设备软件集成并通过认证 AWS。您可以对该列表中的供应商提供的设备软件给予更高的信任度。但是， AWS 不保证这些供应商提供的软件的安全性或可靠性。

内容

Elastic Load Balancing 中的数据保护

分担责任模型 AWS 分担责任模型适用于 Elastic Load Balancing 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://aws.amazon.com/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API 或 AWS 服务 使用 Elastic Load Balancin AWS CLI g 或其他方式时 AWS SDKs。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

静态加密

如果您为用于 Elastic Load Balancing 访问日志的 S3 存储桶启用了使用 Amazon S3 托管加密密钥 (SSE-S3) 的服务器端加密，则 Elastic Load Balancing 会先自动加密每个访问日志文件，然后再存储到 S3 存储桶中。Elastic Load Balancing 还会在您对访问日志文件进行访问时对其进行解密。每个日志文件都使用一个唯一密钥进行加密，此密钥本身将使用定期轮换的 KMS 密钥进行加密。

传输中加密

Elastic Load Balancing 通过在负载均衡器上终止来自客户端的 HTTPS 和 TLS 流量，从而简化了构建安全 Web 应用程序的过程。负载均衡器会执行加密和解密流量的工作，而不要求每个 EC2 实例来处理 TLS 终止工作。在配置安全侦听器时，您可以指定应用程序支持的密码套件和协议版本，以及要在您的负载均衡器上安装的服务器证书。您可以使用 AWS Certificate Manager (ACM) 或 AWS Identity and Access Management (IAM) 来管理您的服务器证书。Application Load Balancer 支持 HTTPS 侦听器。Network Load Balancer 支持 TLS 侦听器。经典负载均衡器同时支持 HTTPS 和 TLS 侦听器。

适用于 Elastic Load Balancing 的 Identity and Access Management

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制可以通过身份验证（登录）和授权（具有权限）使用 Elastic Load Balancing 资源的人员。您可以使用 IAM AWS 服务 ，无需支付额外费用。

内容

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 Elastic Load Balancing 中所做的工作。

服务用户 - 如果您使用 Elastic Load Balancing 服务来完成工作，您的管理员会为您提供所需的凭证和权限。当您使用更多 Elastic Load Balancing 功能来完成工作时，您可能需要其他权限。了解如何管理访问权限有助于您向管理员请求适合的权限。

服务管理员 – 如果您在公司负责管理 Elastic Load Balancing 资源，您可能具有 Elastic Load Balancing 的完全访问权限。您有责任确定您的服务用户应访问哪些 Elastic Load Balancing 功能和资

源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。

IAM 管理员 – 如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 Elastic Load Balancing 的访问权限的详细信息。

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的如何登录您的 AWS 账户。

对于编程访问， AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务 和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。 AWS IAM Identity Center有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

IAM 用户是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户使用案例。

IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。 AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概述。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中指定主体。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《 IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

Elastic Load Balancing 如何与 IAM 一起工作

在使用 IAM 管理对 Elastic Load Balancing 的访问权限之前，您应该了解哪些 IAM 功能可与 Elastic Load Balancing 配合使用。

可与 Elastic Load Balancing 配合使用的 IAM 功能

IAM 功能	Elastic Load Balancing 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是

IAM 功能	Elastic Load Balancing 支持
ACLs	否
ABAC（策略中的标签）	是
临时凭证	是
主体权限	是
服务角色	否
服务关联角色	是

Elastic Load Balancing 基于身份的策略

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Elastic Load Balancing 内基于资源的策略

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

Elastic Load Balancing 的策略操作

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看弹性负载均衡操作的列表，请参阅《服务授权参考》中的 Actions defined by Elastic Load Balancing V2 和 Actions defined by Elastic Load Balancing V1。

Elastic Load Balancing 中的策略操作在操作前使用以下前缀：

```
elasticloadbalancing
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
    "elasticloadbalancing:action1",
    "elasticloadbalancing:action2"
]
```

您也可以使用通配符（*）指定多个操作。例如，要指定以单词 `Describe` 开头的所有操作，包括以下操作：

```
"Action": "elasticloadbalancing:Describe*"
```

有关 Elastic Load Balancing API 操作的完整列表，请参阅以下文档：

- 应用程序负载均衡器、网络负载均衡器和网关负载平衡器 — API 参考版本 2015-12-01
- 经典负载均衡器— API 参考版本 2012-06-01

Elastic Load Balancing 的策略资源

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

某些 Elastic Load Balancing API 操作支持多个资源。要在单个语句中指定多个资源，请 ARNs 用逗号分隔。

```
"Resource": [
    "resource1",
    "resource2"
]
```

要查看 Elastic Load Balancing 资源类型及其类型列表 ARNs，请参阅《服务授权参考》中的 Elastic Load Balancing V2 定义的资源和 Elastic Load Balancing V1 定义的资源。要了解用来指定各项资源的 ARN 的操作，请参阅 Actions defined by Elastic Load Balancing V2 和 Actions defined by Elastic Load Balancing V1。

Elastic Load Balancing 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Condition` 元素根据定义的条件指定语句何时执行。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看弹性负载均衡条件键的列表，请参阅《服务授权参考》中的 Condition keys for Elastic Load Balancing V2 和 Condition keys for Elastic Load Balancing V1。要了解可使用条件键的操作和资源，请参阅 Actions defined by Elastic Load Balancing V2 和 Actions defined by Elastic Load Balancing V1。

条件键

- elasticloadbalancing:ListenerProtocol 条件键
- elasticloadbalancing:SecurityPolicy 条件键
- elasticloadbalancing:Scheme 条件键
- elasticloadbalancing:SecurityGroup 条件键
- elasticloadbalancing:Subnet 条件键

- elasticloadbalancing:ResourceTag 条件键

elasticloadbalancing:ListenerProtocol 条件键

`elasticloadbalancing:ListenerProtocol` 条件键可用于定义可创建和使用的侦听器类型的条件。该策略适用于应用程序负载均衡器、网络负载均衡器和经典负载均衡器。以下操作支持此条件键：

API 版本 2015-12-01

- `CreateListener`
- `ModifyListener`

API 版本 2012-06-01

- `CreateLoadBalancer`
- `CreateLoadBalancerListeners`

以下示例策略要求用户为其应用程序负载均衡器的侦听器选择 HTTPS 协议，为其网络负载均衡器的侦听器选择 TLS 协议。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateListener",
            "elasticloadbalancing:ModifyListener"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "elasticloadbalancing:ListenerProtocol": [
                    "HTTPS",
                    "TLS"
                ]
            }
        }
```

```
    }
}
```

使用经典负载均衡器时，您可以在单个调用中指定多个侦听器。因此，您的策略必须使用多值上下文键，如以下示例所示。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateLoadBalancerListeners"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "elasticloadbalancing:ListenerProtocol": [
                        "TCP",
                        "HTTP",
                        "HTTPS"
                    ]
                }
            }
        }
    ]
}
```

elasticloadbalancing:SecurityPolicy 条件键

elasticloadbalancing:SecurityPolicy 条件键可用于定义和强制执行负载均衡器上的特定安全策略的条件。该策略适用于应用程序负载均衡器、网络负载均衡器和经典负载均衡器。以下操作支持此条件键：

API 版本 2015-12-01

- CreateListener

- ModifyListener

API 版本 2012-06-01

- CreateLoadBalancerPolicy
- SetLoadBalancerPoliciesOfListener

以下示例策略要求用户为其应用程序负载均衡器和网络负载均衡器选择指定的安全策略之一。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateListener",
            "elasticloadbalancing:ModifyListener"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "elasticloadbalancing:SecurityPolicy": [
                    "ELBSecurityPolicy-TLS13-1-2-2021-06",
                    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
                    "ELBSEcurityPolicy-TLS13-1-1-2021-06"
                ]
            }
        }
    }
}
```

elasticloadbalancing:Scheme 条件键

elasticloadbalancing:Scheme 条件键可用于定义在创建负载均衡器期间可以选择哪种方案的条件。该策略适用于应用程序负载均衡器、网络负载均衡器和经典负载均衡器。以下操作支持此条件键：

API 版本 2015-12-01

- `CreateLoadBalancer`

API 版本 2012-06-01

- `CreateLoadBalancer`

以下示例策略要求用户为其负载均衡器选择指定的方案。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "elasticloadbalancing:CreateLoadBalancer",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "elasticloadbalancing:Scheme": "internal"
            }
        }
    }
}
```

`elasticloadbalancing:SecurityGroup` 条件键

> **Important**
>
> Elastic Load Balancing 接受安全组的所有大写形式。 IDs但是，请确保使用适当的不区分大小写的条件运算符，例如 `StringEqualsIgnoreCase`。

`elasticloadbalancing:SecurityGroup` 条件键可用于定义哪些安全组可以应用于负载均衡器的条件。该策略适用于应用程序负载均衡器、网络负载均衡器和经典负载均衡器。以下操作支持此条件键：

API 版本 2015-12-01

- `CreateLoadBalancer`
- `SetSecurityGroups`

API 版本 2012-06-01

- `CreateLoadBalancer`
- `ApplySecurityGroupsToLoadBalancer`

以下示例策略要求用户为其负载均衡器选择指定的安全组之一。

```
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateLoadBalancer",
            "elasticloadbalancing:SetSecurityGroup"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEqualsIgnoreCase":{
                "elasticloadbalancing:SecurityGroup": [
                    "sg-51530134",
                    "sg-51530144",
                    "sg-51530139"
                ]
            },
        }
    }
}
```

elasticloadbalancing:Subnet 条件键

> **Important**
>
> Elastic Load Balancing 接受子网的所有大写形式。 IDs但是，请确保使用适当的不区分大小写的条件运算符，例如 `StringEqualsIgnoreCase`。

`elasticloadbalancing:Subnet` 条件键可用于定义可以创建哪些子网并将其附加到负载均衡器的条件。该策略适用于应用程序负载均衡器、网络负载均衡器、网关负载均衡器和经典负载均衡器。以下操作支持此条件键：

API 版本 2015-12-01

- `CreateLoadBalancer`
- `SetSubnets`

API 版本 2012-06-01

- `CreateLoadBalancer`
- `AttachLoadBalancerToSubnets`

以下示例策略要求用户为其负载均衡器选择指定的子网之一。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateLoadBalancer",
            "elasticloadbalancing:SetSubnets"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEqualsIgnoreCase": {
                "elasticloadbalancing:Subnet": [
                    "subnet-01234567890abcdef",
                    "subnet-01234567890abcdeg "
                ]
            }
        }
    }
}
```

elasticloadbalancing:ResourceTag 条件键

`elasticloadbalancing:ResourceTag/key`条件键特定于 Elastic Load Balancing。所有变异操作都支持此条件键。

ACLs 在 Elastic Load Balanc

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

具有 Elastic Load Balancing 的 ABAC

支持 ABAC（策略中的标签）：是

基于属性的访问权限控制（ABAC）是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`aws:RequestTag/*key-name* 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

将临时凭证与 Elastic Load Balancing 配合使用

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的临时安全凭证和使用 IAM 的。AWS 服务

Elastic Load Balancing 的跨服务主体权限

支持转发访问会话（FAS）：是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

Elastic Load Balancing 的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

Elastic Load Balancing 的服务相关角色

支持服务关联角色：是

服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理 Elastic Load Balancing 服务相关角色的详细信息，请参阅 Elastic Load Balancing 服务相关角色。

在创建过程中为资源添加标签的 Elastic Load Balancing API 权限

为使用户在创建过程中为资源添加标签，他们必须具有使用创建该资源的操作（如 `elasticloadbalancing:CreateLoadBalancer` 或 `elasticloadbalancing:CreateTargetGroup`）的权限。如果在资源创建操作中指定标签，则需要在 `elasticloadbalancing:AddTags` 操作上执行额外的授权，以验证用户是否具备为所创建资源应用标签的权限。因此，用户还必须具有使用 `elasticloadbalancing:AddTags` 操作的显式权限。

在 `elasticloadbalancing:AddTags` 操作的 IAM policy 定义中，可使用带有 `Condition` 条件键的 `elasticloadbalancing:CreateAction` 元素，为创建资源的操作授予添加标签的权限。

如下的示例演示了一个策略，其允许用户创建目标组并在创建过程中向其应用任何标签。用户无权标记任何现有资源 (他们无法直接调用 `elasticloadbalancing:AddTags` 操作)。

JSON

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:CreateTargetGroup"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
             "elasticloadbalancing:CreateAction" : "CreateTargetGroup"
         }
      }
    }
  ]
}
```

同样，下面的策略允许用户创建负载均衡器并在创建过程中应用标签。用户无权标记任何现有资源 (他们无法直接调用 `elasticloadbalancing:AddTags` 操作)。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:CreateLoadBalancer"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
            "elasticloadbalancing:AddTags"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
                "elasticloadbalancing:CreateAction" : "CreateLoadBalancer"
            }
         }
      }
   ]
}
```

仅当用户在资源创建操作中应用了标签时，系统才会评估 `elasticloadbalancing:AddTags` 操作。因此，如果未在此请求中指定任何标签，则拥有创建资源权限（假定没有标记条件）的用户无需具备使用 `elasticloadbalancing:AddTags` 操作的权限。但是，如果用户不具备使用 `elasticloadbalancing:AddTags` 操作的权限而又试图创建带标签的资源，则请求将失败。

Elastic Load Balancing 服务相关角色

Elastic Load Balancing 使用服务相关角色来获取它代表您调用其他 AWS 服务所需的权限。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色。

服务相关角色授予的权限

弹性负载均衡使用名为 AWSServiceRoleForElasticLoadBalancing 的服务相关角色来代表您调用其他 AWS 服务。

AWSServiceRoleForElasticLoadBalancing 信任 `elasticloadbalancing.amazonaws.com` 服务来代入该角色。

角色权限策略为 AWSElasticLoadBalancingServiceRolePolicy。要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 AWSElasticLoadBalancingServiceRolePolicy。

创建服务相关角色

您无需手动创建 AWSServiceRoleForElasticLoadBalancing 角色。Elastic Load Balancing 将在您创建负载均衡器或目标组时为您创建此角色。

要让 Elastic Load Balancing 用户代表您创建服务相关角色，您必须具有所需权限。有关更多信息，请参阅 IAM 用户指南中的服务相关角色权限。

编辑服务相关角色

您可以使用 IAM 编辑 AWSServiceRoleForElasticLoadBalancing 的描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色描述。

删除服务相关角色

如果不再需要使用弹性负载均衡，我们建议您删除 AWSServiceRoleForElasticLoadBalancing。

只有在删除账户中的所有负载均衡器后，才能删除此服务相关角色。 AWS 这可确保您不会无意中删除访问您的负载均衡器的权限。有关更多信息，请参阅删除 Application Load Balancer、删除 Network Load Balancer 和删除经典负载均衡器。

您可以使用 IAM 控制台、IAM CLI 或 IAM API 删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

删除 AWSServiceRoleForElasticLoadBalancing 之后，弹性负载均衡将在您创建负载均衡器时再次为您创建该角色。

AWS Elastic Load Balancing 托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于使用案例的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AWS 托管策略：AWSElasticLoadBalancingClassicServiceRolePolicy

该策略包括 Elastic Load Balancing（Classic Load Balance）代表您调用其他 AWS 服务所需的所有权限。服务相关角色已预先定义。使用预定义角色，您不必手动添加 Elastic Load Balancing 代表您完成操作所需的权限。您不能附加、分离、修改或删除此策略。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 AWSElasticLoadBalancingClassicServiceRolePolicy。

AWS 托管策略：AWSElasticLoadBalancingServiceRolePolicy

此策略包含 Elastic Load Balancing 代表您调用其他 AWS 服务所需的所有权限。服务相关角色已预先定义。使用预定义角色，您不必手动添加 Elastic Load Balancing 代表您完成操作所需的权限。您不能附加、分离、修改或删除此策略。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 AWSElasticLoadBalancingServiceRolePolicy。

AWS 托管策略：ElasticLoadBalancingFullAccess

该策略允许用户完全访问 Elastic Load Balancing 服务，并通过 AWS 管理控制台限制访问其他服务。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 ElasticLoadBalancingFullAccess。

AWS 托管策略：ElasticLoadBalancingReadOnly

此策略提供对 Elastic Load Balancing 和相关服务的只读访问权限

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 ElasticLoadBalancingReadOnly。

Elastic Load Balancing 更新 AWS 了托管策略

查看自该服务开始跟踪这些更改以来，Elastic Load Balancing AWS 托管策略更新的详细信息。

更改	描述	日期
ElasticLoadBalancingFullAccess – 对现有策略的更新	添加了在输入验证期间授予描述可用区的权限的`ec2:DescribeAvailityZones` 操作。	2026年2月23日
AWSElasticLoadBalancingServiceRolePolicy – 对现有策略的更新	添加了在输入验证期间授予描述可用区的权限的`ec2:DescribeAvailityZones` 操作。	2025 年 11 月 21 日
AWSElasticLoadBalancingServiceRolePolicy – 对现有策略的更新	增加了 `ec2:AllocateIpamPoolCidr` 操作，以授予从 IPAM 池中分配 CIDR 数据块的权限。	2025 年 2 月 17 日
ElasticLoadBalancingFullAccess – 对现有策略的更新	增加了 `arc-zonal-shift:*` 操作，以授予区可用区转移所需的权限。	2023 年 11 月 28 日

更改	描述	日期
ElasticLoadBalancingReadOnly – 对现有策略的更新	增加了下列操作，以授予区可用区转移所需的权限：`arc-zonal-shift:GetManagedResource` 、`arc-zonal-shift:ListManagedResources` 和 `arc-zonal-shift:ListZonalShifts` 。	2023 年 11 月 28 日
AWSElasticLoadBalancingServiceRolePolicy – 对现有策略的更新	增加了 `ec2:DescribeVpcPeeringConnections` 操作，以授予对等连接所需的权限。	2021 年 10 月 11 日
ElasticLoadBalancingFullAccess – 对现有策略的更新	增加了 `ec2:DescribeVpcPeeringConnections` 操作，以授予对等连接所需的权限。	2021 年 10 月 11 日
ElasticLoadBalancingFullAccess：新策略	提供对弹性负载均衡和相关服务的完全访问权限。	2018 年 9 月 20 日
ElasticLoadBalancingReadOnly：新策略	提供对 Elastic Load Balancing 和相关服务的只读访问。	2018 年 9 月 20 日
Elastic Load Balancing 开始跟踪更改	Elastic Load Balancing 开始跟踪其 AWS 托管策略的更改。	2018 年 9 月 20 日

Elastic Load Balancing 的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。有关您在使用时的合规责任的更多信息 AWS 服务，请参阅AWS 安全文档。

Elastic Load Balancing 中的故障恢复能力

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。各区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅AWS 全球基础设施。

除 AWS 全球基础设施外，Elastic Load Balancing 还提供以下功能来支持您的数据弹性：

- 在一个或多个可用区中的多个实例之间分配传入流量。
- 您可以 AWS Global Accelerator 与应用程序负载均衡器配合使用，在一个或多个区域的多个负载均衡器之间分配传入流量。 AWS 有关更多信息，请参见AWS Global Accelerator 开发人员指南。
- 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 Amazon ECS 服务配置为使用负载均衡器在集群中的服务之间分配传入流量。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南。

Elastic Load Balancing 中的基础设施安全性

作为一项托管服务，Elastic Load Balancin AWS g 受到全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 Elastic Load Balancing。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

网络隔离

虚拟私有云 (VPC) 是位于 AWS 云中您自己的逻辑隔离区域中的虚拟网络。子网是 VPC 中的 IP 地址范围。当您创建负载均衡器时，可以为负载均衡器节点指定一个或多个子网。您可以在 VPC 的子网中部署 EC2 实例并将其注册到您的负载均衡器。有关 VPC 和子网的更多信息，请参阅 Amazon VPC 用户指南。

当您在 VPC 中创建负载均衡器时，它可以面向 Internet，也可以面向内部。内部负载均衡器可路由的请求只能来自对负载均衡器的 VPC 具有访问权限的客户端。

您的负载均衡器会使用私有 IP 地址向已注册目标发送请求。因此，您的目标无需使用公有 IP 地址，即可接收来自负载均衡器的请求。

要使用私有 IP 地址从 VPC 调用 Elastic Load Balancing API，请使用 AWS PrivateLink。有关更多信息，请参阅 使用接口端点访问 Elastic Load Balancing（AWS PrivateLink）。

控制网络流量

当您使用负载均衡器时，请考虑使用以下选项来保护网络流量：

- 使用安全侦听器支持客户端和负载均衡器之间的加密通信。Application Load Balancer 支持 HTTPS 侦听器。Network Load Balancer 支持 TLS 侦听器。经典负载均衡器同时支持 HTTPS 和 TLS 侦听器。您可以从您的负载均衡器的预定义安全策略中选择，指定您的应用程序支持的密码套件和协议版本。您可以使用 AWS Certificate Manager (ACM) 或 AWS Identity and Access Management (IAM) 来管理负载均衡器上安装的服务器证书。您可以利用服务器名称指示 (SNI) 协议，使用单个安全侦听器为多个安全网站提供服务。当您将多个服务器证书与安全侦听器关联时，会自动为您的负载均衡器启用 SNI。
- 配置 Application Load Balancer 和经典负载均衡器的安全组，以仅接受来自特定客户端的流量。这些安全组必须在侦听器端口上允许来自客户端的入站流量以及流向客户端的出站流量。
- 为您的 Amazon EC2 实例配置安全组，使其仅接受来自负载均衡器的流量。这些安全组必须在侦听器端口和运行状况检查端口上允许来自负载均衡器的入站流量。
- 配置您的 Application Load Balancer，以通过身份提供商或使用公司身份安全地对用户进行身份验证。有关更多信息，请参阅使用 Application Load Balancer 对用户进行身份验证。
- 将 AWS WAF 与 Application Load Balancer 结合使用，根据 Web 访问控制列表 (Web ACL) 中的规则允许或阻止请求。

使用接口端点访问 Elastic Load Balancing（AWS PrivateLink）

您可以通过创建接口 VPC 终端节点在 Virtual Private Cloud (VPC) 与 Elastic Load Balancing API 之间建立私有连接。您可以使用此连接从 VPC 调用 Elastic Load Balancing API，而无需将互联网网关、NAT 实例或 VPN 连接附加到您的 VPC。终端节点提供了与用于创建和管理负载均衡器的 2015-12-01 版和 2012-06-01 版 Elastic Load Balancing API 的可靠、可扩展连接。

接口 VPC 终端节点由 AWS PrivateLink该功能提供支持，该功能允许您的应用程序之间 AWS 服务 使用私有 IP 地址进行通信。有关更多信息，请参阅 AWS PrivateLink。

限制

AWS PrivateLink 不支持监听器超过 50 的网络负载均衡器。

为 Elastic Load Balancing 创建接口终端节点

使用以下服务名称为 Elastic Load Balancing 创建终端节点：

```
com.amazonaws.region.elasticloadbalancing
```

有关更多信息，请参阅《AWS PrivateLink 指南》中的创建接口端点。

为 Elastic Load Balancing 创建 VPC 终端节点策略

您可以向 VPC 终端节点附加策略，以控制对 Elastic Load Balancing API 的访问。该策略指定：

- 可执行操作的主体。
- 可执行的操作。
- 可对其执行操作的资源。

以下示例显示了一个 VPC 终端节点策略，该策略拒绝所有人通过终端节点创建负载均衡器的权限。示例策略还授予所有人执行所有其他操作的权限。

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticloadbalancing:CreateLoadBalancer",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

有关更多信息，请参阅《AWS PrivateLink 指南》中的使用端点策略控制对服务的访问权限。

弹性负载均衡 API 的请求节流

Elastic Load Balancing 会根据每个区域限制其对每个 AWS 账户的 API 请求。这是为了帮助提高服务的性能和可用性。节流可确保对弹性负载均衡 API 的请求不会超过允许的最大 API 请求限制。无论您调用还是代表您调用 API 请求（例如，由应用程序或第三方应用程序），API 请求均受请求限制的约束。 AWS 管理控制台

如果超出弹性负载均衡 API 的节流限制，您将会收到 `ThrottlingException` 错误代码和 `Rate exceeded` 错误消息。

建议您做好准备，以从容地处理节流。有关更多信息，请参阅超时、重试和回退并抖动。如果您遇到严重的限制，可以联系 AWS 支持 以帮助您评估您的 API 使用情况和潜在的解决方案。每个案例都经过单独评估。 支持 可能会在系统的安全限制范围内增加限制，以保持高可用性和可预测的性能。

如何应用节流

弹性负载均衡使用令牌存储桶算法来实现 API 节流。使用此算法，您的账户拥有一个持有特定数量的令牌的存储桶。存储桶中的令牌数量代表您在任意一秒的节流限制。

弹性负载均衡提供两套 API 操作。ELB API 版本 2 支持以下类型的负载均衡器：应用程序负载均衡器、网络负载均衡器和网关负载均衡器。ELB API 版本 1 支持经典负载均衡器。每个 ELB API 版本都有自己的存储桶和令牌。

代表您调用 Elastic Load Balancing API 的服务，例如亚马逊 EC2、亚马逊 ECS、Amazon A EC2 uto Scaling，并且 AWS CloudFormation 拥有自己的账户级存储桶。这些服务不会消耗您存储桶中的令牌。

请求速率限制

使用请求速率限制时，您发出的 API 请求数量会受到节流。您发出的每个请求都会从存储桶中删除一个令牌。例如，非变异 API 操作的令牌存储桶容量为 40 个令牌。您在一秒内最多可以发出 40 个 `Describe*` 请求。如果您在一秒钟内发出的 `Describe*` 请求超过 40 个，则会被节流，同时该秒内剩余的请求将会失败。

存储桶会以设定的速率自动填充。如果存储桶的容量低于其最大容量，则每秒会将一定数量的令牌退回该存储桶，直至达到其最大容量为止。如果重填令牌到达时存储桶已满，则这些令牌将被丢弃。存储桶中的令牌数量不能超过其最大数量。例如，非变异 API 操作的存储桶容量为 40 个令牌，重填速率为每秒 10 个令牌。如果您在一秒钟内发起 40 个 `DescribeLoadBalancers` 请求，存储桶中的令牌将被

消耗至零（0）。我们会按每秒 10 个令牌的速度重填该存储桶，直至达到 40 个令牌的最大容量为止。因此，如果在此期间没有发出任何请求，一个空存储桶需要 4 秒才会达到其最大容量。

您无需等存储桶完全填满才能继续发起 API 请求。您可以在令牌添加到存储桶的同时使用这些令牌。如果您立即使用重填令牌，存储桶就不会达到最大容量。

所有弹性负载均衡 API 操作都共享同一个账户级别的节流限制。该账户级别存储桶的容量为 40 个令牌，重填速率为每秒 10 个请求令牌。

请求令牌存储桶的容量与重填速率

API 操作分为若干类别来实施请求速率限制。每个类别都有自己的限制。

类别

- 变异操作：用于创建、修改或删除资源的 API 操作。此类别通常包括所有未归类为非变异操作的 API 操作。此类操作的节流限制低于非变异 API 操作。
- 非变异操作：用于检索资源相关数据的 API 操作。此类 API 操作通常具有最高的 API 节流限制。
- 资源密集型操作：完成时间最长、消耗资源最多的变异类 API 操作。此类操作的节流限制比变异操作更低。此类操作的节流独立于其他变异操作。
- 注册操作：用于注册或注销目标的 API 操作。此类 API 操作的节流独立于其他变异操作。
- 未分类操作：此类 API 操作拥有独立的令牌存储桶容量和重填速率，即使原本属于上述其他类别亦不例外。

下表显示了各类别请求令牌存储桶的默认容量和重填速率。

类别	ELBv2 行动	ELBv1 行动	存储桶容量	重填速率（每秒）
资源密集型	`CreateLoadBalancer` ，`SetSubnets`	`CreateLoadBalancer` ，`AttachLoadBalancerToSubnets` ，`DetachLoadBalancerFromSubnets` ，`EnableAvailabilityZonesForLoadBalancer` ，	10	0.2 †

类别	ELBv2 行动	ELBv1 行动	存储桶容量	重填速率（每秒）
		DisableAvailabilityZonesForLoadBalancer		
注册	RegisterTargets，DeregisterTargets	RegisterInstancesWithLoadBalancer，DeregisterInstancesFromLoadBalancer	20	4
非变异	DescribeAccountLimits，DescribeCapacityReservation，DescribeListenerAttributes，DescribeListenerCertificates，DescribeListeners，DescribeLoadBalancerAttributes，DescribeLoadBalancers，DescribeRules，DescribeSSLPolicies，DescribeTags，DescribeTargetGroupAttributes，DescribeTargetGroups，DescribeTargetHealth	Describe*	40	10

类别	ELBv2 行动	ELBv1 行动	存储桶容量	重填速率（每秒）
变异	AddListenerCertificates，AddTags，CreateListener，CreateRule，CreateTargetGroup，DeleteListener，DeleteLoadBalancer，DeleteRule，DeleteTargetGroup，ModifyCapacityReservation，ModifyIpPools，ModifyListener，ModifyListenerAttributes，ModifyLoadBalancerAttributes，ModifyRule，ModifyTargetGroup，ModifyTargetGroupAttributes，RemoveListenerCertificates，RemoveTags，SetIpAddressType，SetRulePriorities，SetSecurityGroups	AddTags，ApplySecurityGroupsToLoadBalancer，ConfigureHealthCheck，CreateAppCookieStickinessPolicy，CreateLbCookieStickinessPolicy，CreateLoadBalancerListener，CreateLoadBalancerPolicy，Delete*，ModifyLoadBalancerAttributes，RemoveTags，SetLoadBalancer*	20	3

下表显示了未分类的请求令牌存储桶的默认容量和充值费率。 ELBv2

ELBv2 行动	存储桶容量	重填速率（每秒）
CreateTrustStore	10	0.2 †
AddTrustStoreRevocations，DeleteSharedTrustStoreAssoc	10	0.2 †

ELBv2 行动	存储桶容量	重填速率（每秒）
iation ，DeleteTrustStore ，ModifyTrustStore ，RemoveTrustStoreRevocations		
GetResourcePolicy ，GetTrustStoreCaCertificatesBundle ，GetTrustStoreRevocationContent	20	4
DescribeTrustStoreAssociations ，DescribeTrustStoreRevocations ，DescribeTrustStores	40	10

† 部分重填速率是指需要几秒钟才能生成一个完整的令牌。

监控 API 请求

您可以使用 AWS CloudTrail 来监控您的 Elastic Load Balancing API 请求。有关更多信息，请参阅 使用记录 Elastic Load Balancing 的 API 调用 AWS CloudTrail。

了解账单和使用情况报告中的弹性负载均衡代码

当您使用 Elastic Load Balancing 时，我们会在您的 AWS 账单和使用报告中包含相关代码。查看这些代码有助于您了解负载均衡器的成本和使用模式。跟踪和管理支出对于优化成本至关重要。

有关更多信息，请参阅 Elastic Load Balancing 定价。

下表列出了账单和使用情况报告中出现的弹性负载均衡代码。单位为小时或负载均衡器容量单位（LCU）。每种负载均衡器类型都有对 LCU 的具体定义。有关每种负载均衡器类型的信息，请参阅 Elastic Load Balancing 定价。 LCUs 有关账单和使用情况报告中使用的区域代码列表，请参阅 AWS Region billing codes。

应用程序负载均衡器

代码	说明	单位
region-LoadBalancerUsage	运行时间。	Hours
region-LCUUsage	二 LCUs 手的。	LCU
region-IdleProvisionedLBCapacity	已 LCUs 保留但未使用。	LCU
region-TS-LoadBalancerUsage	双向 TLS 使用信任存储的时间。	Hours
region-Outposts-LoadBalancerUsage	在 Outposts 上运行的时间。	Hours
region-Outposts-LCUUsage	在 Outposts 上 LCUs 使用的。	LCU
region-ReservedLCUUsage	LCUs 保留的。	LCU

网络负载均衡器

代码	说明	单位
region-LoadBalancerUsage	运行时间。	Hours
region-LCUUsage	二 LCUs 手的。	LCU

Gateway Load Balancer

代码	说明	单位
region-LoadBalancerUsage	运行时间。	Hours
region-LCUUsage	二 LCUs 手的。	LCU

经典负载均衡器

代码	说明	单位
region-LoadBalancerUsage	运行时间。	Hours
region-DataProcessing-Bytes	已处理的数据量。	GB
region-IdleProvisionedLBCapacity	已 LCUs 保留但未使用。	LCU

使用记录 Elastic Load Balancing 的 API 调用 AWS CloudTrail

Elastic Load Balancing 与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务所执行操作的记录。 CloudTrail 将 Elastic Load Balancing 的 API 调用捕获为事件。捕获的调用包括来自的调用 AWS 管理控制台 以及对 Elastic Load Balancing API 操作的代码调用。使用收集的信息 CloudTrail，您可以确定向 Elastic Load Balancing 发出的请求、发出请求的 IP 地址、发出请求的时间以及其他详细信息。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户凭证还是用户凭证发出的。
- 请求是否代表 IAM Identity Center 用户发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

CloudTrail 在您创建账户 AWS 账户 时在您的账户中处于活动状态，并且您自动可以访问 CloudTrail 活动历史记录。 CloudTrail 事件历史记录提供了过去 90 天中记录的管理事件的可查看、可搜索、可下载且不可变的记录。 AWS 区域有关更多信息，请参阅《AWS CloudTrail 用户指南》中的 “使用 CloudTrail 事件历史记录”。查看活动历史记录不 CloudTrail收取任何费用。

要持续记录 AWS 账户 过去 90 天内的事件，请创建跟踪或 CloudTrailLake 事件数据存储。

CloudTrail 步道

跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。使用创建的所有跟踪 AWS 管理控制台 都是多区域的。您可以通过使用 AWS CLI创建单区域或多区域跟踪。建议创建多区域跟踪，因为您可以捕获账户 AWS 区域 中的所有活动。如果您创建单区域跟踪，则只能查看跟踪的 AWS 区域中记录的事件。有关跟踪的更多信息，请参阅《AWS CloudTrail 用户指南》中的为您的 AWS 账户创建跟踪和为组织创建跟踪。

通过创建跟踪，您可以免费将正在进行的管理事件的一份副本传送到您的 Amazon S3 存储桶，但会收取 Amazon S3 存储费用。 CloudTrail 有关 CloudTrail 定价的更多信息，请参阅AWS CloudTrail 定价。有关 Amazon S3 定价的信息，请参阅 Amazon S3 定价。

CloudTrail 湖泊事件数据存储

CloudTrail L@@ ak e 允许你对自己的事件运行基于 SQL 的查询。 CloudTrail Lake 将基于行的 JSON 格式的现有事件转换为 Apache ORC 格式。ORC 是一种针对快速检索数据进行优化的列式存储格式。事件将被聚合到事件数据存储中，它是基于您通过应用高级事件选择器选择的条件的不可变的事件集合。应用于事件数据存储的选择器用于控制哪些事件持续存在并可供您查询。有关 CloudTrail Lake 的更多信息，请参阅《AWS CloudTrail 用户指南》中的 “使用 AWS CloudTrail Lake”。

CloudTrail 湖泊事件数据存储和查询会产生费用。创建事件数据存储时，您可以选择要用于事件数据存储的定价选项。定价选项决定了摄取和存储事件的成本，以及事件数据存储的默认和最长保留期。有关 CloudTrail 定价的更多信息，请参阅AWS CloudTrail 定价。

中的 Elastic Load Balancing 管理事件 CloudTrail

管理事件提供有关对中的资源执行的管理操作的信息 AWS 账户。这些也称为控制面板操作。默认情况下， CloudTrail 记录管理事件。

Elastic Load Balancing 将控制面板操作记录为管理事件。有关控制面板操作的列表，请参阅以下内容：

- 应用程序负载均衡器 — Elastic Load Balancing API 参考版本 2015-12-01
- 网络负载均衡器 — Elastic Load Balancing API 参考版本 2015-12-01
- 网关负载均衡器 — Elastic Load Balancing API 参考版本 2015-12-01
- 经典负载均衡器 — Elastic Load Balancing API 参考版本 2012-06-01

Elastic Load Balancing 事件示例

事件代表来自任何来源的单个请求，包括有关所请求的 API 操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此事件不会按任何特定顺序出现。

以下示例显示了创建负载均衡器然后使用删除负载均衡器的用户所 CloudTrail 发生的事件 AWS CLI。您可以使用 `userAgent` 元素标识 CLI。可使用 `eventName` 元素标识请求的 API 调用。有关用户 (`Alice`) 的信息可在 `userIdentity` 元素中找到。

Example示例 1： CreateLoadBalancer 来自 ELBv2 API

```
{
```

```
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-8360a9e7","subnet-b7d581c0"],
        "securityGroups": ["sg-5943793c"],
        "name": "my-load-balancer",
        "scheme": "internet-facing"
    },
    "responseElements": {
        "loadBalancers":[{
            "type": "application",
            "loadBalancerName": "my-load-balancer",
            "vpcId": "vpc-3ac0fb5f",
            "securityGroups": ["sg-5943793c"],
            "state": {"code":"provisioning"},
            "availabilityZones": [
               {"subnetId":"subnet-8360a9e7","zoneName":"us-west-2a"},
               {"subnetId":"subnet-b7d581c0","zoneName":"us-west-2b"}
            ],
            "dNSName": "my-load-balancer-1836718677.us-west-2.elb.amazonaws.com",
            "canonicalHostedZoneId": "Z2P70J7HTTTPLU",
            "createdTime": "Apr 11, 2016 5:23:50 PM",
            "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0",
            "scheme": "internet-facing"
        }]
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
```

```
    "recipientAccountId": "123456789012"
}
```

Example示例 2： DeleteLoadBalancer 来自 ELBv2 API

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0"
    },
    "responseElements": null,
    "requestID": "349598b3-000e-11e6-a82b-298133eEXAMPLE",
    "eventID": "75e81c95-4012-421f-a0cf-babdaEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

Example示例 3： CreateLoadBalancer 来自弹性负载均衡 API

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJDPLRKLG7UEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-12345678","subnet-76543210"],
        "loadBalancerName": "my-load-balancer",
        "listeners": [{
           "protocol": "HTTP",
           "loadBalancerPort": 80,
           "instanceProtocol": "HTTP",
           "instancePort": 80
        }]
    },
    "responseElements": {
        "dNSName": "my-loadbalancer-1234567890.elb.amazonaws.com"
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2012-06-01",
    "recipientAccountId": "123456789012"
}
```

Example示例 4： DeleteLoadBalancer 来自弹性负载均衡 API

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJDPLRKLG7UEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-08T12:39:25Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerName": "my-load-balancer"
    },
    "responseElements": null,
    "requestID": "f0f17bb6-b9ba-11e3-9b20-999fdEXAMPLE",
    "eventID": "4f99f0e8-5cf8-4c30-b6da-3b69fEXAMPLE"
    "eventType": "AwsApiCall",
    "apiVersion": "2012-06-01",
    "recipientAccountId": "123456789012"
}
```

有关 CloudTrail 录音内容的信息，请参阅《AWS CloudTrail 用户指南》中的CloudTrail录制内容。

迁移您的经典负载均衡器

Elastic Load Balancing 支持以下类型的负载均衡器：应用程序负载均衡器、网络负载均衡器、网关负载均衡器和经典负载均衡器。要了解每种负载均衡器类型的不同功能，请参阅弹性负载均衡产品对比。

您还可以选择将 VPC 中的现有经典负载均衡器迁移到应用程序负载均衡器或网络负载均衡器。

从经典负载均衡器迁移的好处

每种类型的负载均衡器都有其独特的特性、功能和配置。查看每种负载均衡器的优点，以帮助决定哪一种最适合您。

Application Load Balancer

使用应用程序负载均衡器而不是经典负载均衡器具有以下好处：

支持：

- 路径条件、主机条件和 HTTP 标头条件。
- 将请求从一个 URL 重定向到另一个 URL，并将请求路由到单个 EC2 实例上的多个应用程序。
- 返回自定义 HTTP 响应。
- 通过 IP 地址注册目标，并将 Lambda 函数注册为目标。包括位于负载均衡器的 VPC 之外的目标。
- 通过企业或社交身份对用户进行身份验证。
- Amazon Elastic Container Service（Amazon ECS）容器化应用程序。
- 独立监控每个服务的运行状况。

访问日志包含附加信息，并以压缩格式存储。

整体提高了负载均衡器的性能。

Network Load Balancer

使用网络负载均衡器而不是经典负载均衡器具有以下好处：

支持：

- 静态 IP 地址，允许为负载均衡器启用的每个子网分配一个弹性 IP 地址。

- 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。
- 将请求路由到单个 EC2 实例上的多个应用程序。
- Amazon Elastic Container Service（Amazon ECS）容器化应用程序。
- 独立监控每个服务的运行状况。

可以处理急剧波动的工作负载，并可以扩展到每秒处理数百万个请求。

使用迁移向导进行迁移

迁移向导使用经典负载均衡器的配置来创建等效的应用程序负载均衡器或网络负载均衡器。与其他方法相比，它减少了迁移经典负载均衡器所需的时间和精力。

> **Note**
>
> 该向导会创建一个新的负载均衡器。该向导不会将现有的经典负载均衡器转换为应用程序负载均衡器或网络负载均衡器。您必须手动将流量重定向到新创建的负载均衡器。

限制

- 新负载均衡器名称不能与同一区域内同类型的现有负载均衡器相同。
- 如果经典负载均衡器的任何标签的键中包含 `aws:` 前缀，则不会迁移这些标签。

迁移到应用程序负载均衡器时

- 如果经典负载均衡器只有一个子网，则必须指定第二个子网。
- 如果 Classic Load Balancer 的 HTTP/HTTPS 侦听器使用 TCP 运行状况检查，则运行状况检查协议将更新为 HTTP，并将路径设置为 “/”。
- 如果经典负载均衡器的 HTTPS 侦听器使用自定义或不受支持的安全策略，则迁移向导将使用新负载均衡器类型的默认安全策略。

迁移到网络负载均衡器时

- 以下实例类型将不会注册到新的目标组：C1、、、、、、、 CC1、 CC2、G1 CG1 CG2 CR1、G2 CS1、、、、M1、M2 HI1 HS1、M3、T1

- 经典负载均衡器中的某些运行状况检查设置可能无法转移到新的目标组。这些案例将在迁移向导的摘要部分中显示为更改。
- 如果经典负载均衡器具有 SSL 侦听器，则迁移向导将使用来自 SSL 侦听器的证书和安全策略创建 TLS 侦听器。

迁移向导过程

使用迁移向导迁移经典负载均衡器

1. 打开位于 https://console.aws.amazon.com/ec2/ 的 Amazon EC2 控制台。
2. 在导航窗格上的 Load Balancing（负载均衡）下，选择 Load Balancers（负载均衡器）。
3. 选择要迁移的经典负载均衡器。
4. 在负载均衡器的详细信息部分，选择启动迁移向导。
5. 选择迁移到应用程序负载均衡器或迁移到网络负载均衡器以打开迁移向导。
6. 在命名新的负载均衡器下，对于负载均衡器名称，输入新负载均衡器的名称。
7. 在命名新的目标组并查看目标下，对于目标组名称，输入新目标组的名称。
8. （可选）在目标下，您可以查看将在新目标组中注册的目标实例。
9. （可选）在查看标签下，您可以查看将应用于新负载均衡器的标签
10. 在应用程序负载均衡器的摘要或网络负载均衡器的摘要下，查看并验证迁移向导分配的配置选项。
11. 对配置摘要满意后，选择创建应用程序负载均衡器或创建网络负载均衡器以开始迁移。

使用负载均衡器复制实用程序进行迁移

该 AWS GitHub 页面上的 Elastic Load Balancing Tools 存储库中提供了负载均衡器复制工具。

资源

- Elastic Load Balancing 工具
- 经典负载均衡器到应用程序负载均衡器复制实用程序
- 经典负载均衡器到网络负载均衡器复制实用程序

手动迁移负载均衡器

以下信息提供了基于 VPC 中的现有经典负载均衡器手动创建新的 Application Load Balancer 或 Network Load Balancer 的常规说明。您可以使用 AWS 管理控制台 AWS CLI、或 AWS SDK 进行迁移。有关更多信息，请参阅 Elastic Load Balancing 入门。

在迁移过程完成后，您就可以利用新负载均衡器的功能了。

手动迁移过程

步骤 1：创建新负载均衡器

创建配置等效于经典负载均衡器的负载均衡器以进行迁移。

1. 创建具有与经典负载均衡器相同的模式（面向 Internet 或内部）、子网和安全组的新负载均衡器。
2. 使用与经典负载均衡器相同的运行状况检查设置为负载均衡器创建一个目标组。
3. 请执行下列操作之一：
   - 如果您的经典负载均衡器已附加到 Auto Scaling 组，请将目标组附加到 Auto Scaling 组。这样还可以向目标组注册 Auto Scaling 实例。
   - 向目标组注册您的 EC2 实例。
4. 创建一个或多个侦听器，每个都具有将请求转发到目标组的默认规则。如果创建 HTTPS 侦听器，则可指定您为经典负载均衡器所指定的同一证书。建议您使用默认安全策略。
5. 如果您的经典负载均衡器具有标签，请进行检查并将相关标签添加到新负载均衡器。

步骤 2：逐步将流量重定向到您的新负载均衡器

在向新负载均衡器注册您的实例后，您可以开始将流量从旧负载均衡器重定向到新负载均衡器的过程。这使您能够测试新的负载均衡器，同时将应用程序可用性风险降至最低。

逐步将流量重定向到您的新负载均衡器

1. 将新负载均衡器的 DNS 名称粘贴到已连接 Internet 的 Web 浏览器的地址栏中。如果一切正常，浏览器会显示您应用程序的默认页面。
2. 创建一个用于将域名与您的新负载均衡器关联的新 DNS 记录。如果您的 DNS 服务支持权重，则在新 DNS 记录中指定权重为 1；对于您的旧负载均衡器的现有 DNS 记录，指定权重为 9。这样可以将 10% 的流量定向到新负载均衡器，而将 90% 的流量定向到旧负载均衡器。
3. 监控您的新负载均衡器，验证它能否接收流量并将请求路由到您的实例。

> **Important**
>
> DNS 记录中的 time-to-live (TTL) 为 60 秒。这意味着，解析域名的任何 DNS 服务器在其缓存中保留记录信息的时间为 60 秒，同时更改会传播。因此，在您完成上一步后，这些 DNS 服务器仍然可以在 60 秒内将流量路由到旧负载均衡器。在传输过程中，流量可以定向到任一负载均衡器。

4. 继续更新您的 DNS 记录的权重，直到所有流量都定向到您的新负载均衡器。完成后，您可以删除旧负载均衡器的 DNS 记录。

步骤 3：更新策略、脚本和代码

如果要将经典负载均衡器迁移到 Application Load Balancer 或 Network Load Balancer，请务必执行以下操作：

- 将使用 API 版本 2012-06-01 的 IAM 策略更新为使用版本 2015-12-01。
- 更新使用`AWS/ELB`命名空间中 CloudWatch 指标的进程，以使用`AWS/ApplicationELB`或`AWS/NetworkELB`命名空间中的指标。
- 更新使用命令来使用aws elb AWS CLI 命令的脚本。aws elbv2 AWS CLI
- 更新使用`AWS::ElasticLoadBalancing::LoadBalancer`资源来使用`AWS::ElasticLoadBalancingV2`资源的 CloudFormation 模板。
- 将使用 Elastic Load Balancing API 版本 2012-06-01 的代码更新为使用版本 2015-12-01。

资源

- AWS CLI 命令参考中的 elbv2
- Elastic Load Balancing API 参考（2015 年 12 月 1 日版）
- 适用于 Elastic Load Balancing 的 Identity and Access Management
- Application Load Balancer 用户指南中的 Application Load Balancer 指标
- Network Load Balancer 用户指南中的 Network Load Balancer 指标
- 《AWS CloudFormation 用户指南》中的 AWS::ElasticLoadBalancingV2::LoadBalancer

步骤 4：删除旧负载均衡器

您可以在完成以下步骤后删除旧经典负载均衡器：

- 您已将旧负载均衡器的所有流量重定向到新负载均衡器。
- 已完成路由到旧负载均衡器的所有现有请求。

阻止用户创建经典负载均衡器

您可以创建 IAM 策略来阻止用户在您的账户中创建经典负载均衡器。

Elastic Load Balancing V2 和 Elastic Load Balancing V1 都 APIs 提供了 `CreateLoadBalancer` API 操作。创建经典负载均衡器时，您将使用 V1 API 操作，这会同时创建负载均衡器和侦听器。创建应用程序负载均衡器、网络负载均衡器或网关负载均衡器时，您将使用 V2 API 操作。V2 API 提供了一个 `CreateListener` 操作，用来在创建负载均衡器之后为其创建侦听器。

如果指定了侦听器协议，则以下策略将拒绝用户创建负载均衡器的权限。由于在创建经典负载均衡器时必须至少配置一个侦听器，因此此策略会阻止用户创建经典负载均衡器。但不会阻止用户创建其他类型的负载均衡器，因为这些负载均衡器及其侦听器的创建会使用单独的 API 操作。

```
{
    "Version": "2012-10-17",
    "Effect": "Deny",
    "Action": "elasticloadbalancing:CreateLoadBalancer",
    "Resource": [
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
    ],
    "Condition": {
        "Null": {
            "elasticloadbalancing:ListenerProtocol": false
        }
    }
}
```

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。