

Application Load Balancer

Load balance HTTP and HTTPS traffic with advanced request routing targeted at the delivery of modern applications.

[Get started with Elastic Load Balancing](#)

Application Load Balancer operates at the request level (layer 7), routing traffic to targets (EC2 instances, containers, IP addresses, and Lambda functions) based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times.

Features

Layer-7 Load Balancing

You can load balance HTTP/HTTPS traffic to targets - Amazon EC2 instances, microservices, and containers based on request attributes (such as X-Forwarded-For headers).

Security Features

When using Amazon Virtual Private Cloud (VPC), you can create and manage security groups associated with Elastic Load Balancing to provide additional networking and security options. You can configure an Application Load Balancer to be Internet facing or create a load balancer without public IP addresses to serve as an internal (non-internet-facing) load balancer.

ALB supports implementation of Desync protections based on the `http_desync_guardian` library. With this new feature, customer applications are protected from HTTP vulnerabilities due to Desync without making major compromises on availability and/or latency. Customers also have the capability to choose their level of tolerance to suspicious requests based on their application architecture.

Outposts Support

Application Load Balancer (ALB) supports [AWS Outposts](#), a fully managed service that extends AWS infrastructure, services, and tools to virtually any datacenter, co-location space, or on-premises facility for a truly consistent hybrid experience. Customers can provision ALBs on supported instance types and the ALB will auto scale up to the capacity available on the rack to meet varying levels of application load without manual intervention. Customers can also get notifications to help them navigate their load balancing related capacity needs. Customers can use the same AWS Console, APIs, and CLI to provision and manage ALBs on Outposts as they do today with ALBs in the Region.

HTTPS Support

An Application Load Balancer supports HTTPS termination between the clients and the load balancer. Application Load Balancers also offer management of SSL certificates through AWS Identity and Access Management (IAM) and AWS Certificate Manager for pre-defined security policies.

HTTP/2 and gRPC Support

HTTP/2 is a new version of the HyperText Transfer Protocol (HTTP) that uses a single, multiplexed connection to allow multiple requests to be sent on the same connection. It also compresses header data before sending it out in binary format and supports SSL connections to clients.

ALB can route and load balance gRPC traffic between microservices or between gRPC enabled clients and services. This allows seamless introduction of gRPC traffic management in the architectures without changing any of the underlying infrastructure on the customers' clients or services. gRPC uses HTTP/2 for transport and is becoming the protocol of choice for inter-service communications in microservices architectures. It has features like efficient binary serialization and support for numerous languages in addition to the inherent benefits of HTTP/2 like lighter network footprint, compression, and bi-directional streaming making it better than the legacy protocols like REST.

TLS Offloading

You can create an HTTPS listener, which uses encrypted connections (also known as SSL

offload). This feature enables traffic encryption between your load balancer and the clients that initiate SSL or TLS sessions. Application Load Balancer supports client TLS session termination. This enables you to offload TLS termination tasks to the load balancer, while preserving the source IP address for your back-end applications. You can choose from predefined security policies for your TLS listeners in order to meet compliance and security standards. AWS Certificate Manager (ACM) or AWS Identity and Access Management (IAM) can be used to manage your server certificates.

You can use SNI to serve multiple secure websites using a single TLS listener. If the hostname in the client matches multiple certificates, the load balancer selects the best certificate to use based on a smart selection algorithm.

Sticky Sessions

Sticky sessions are a mechanism to route requests from the same client to the same target. Application Load Balancer supports sticky sessions using load balancer generated cookies. If you enable sticky sessions, the same target receives the request and can use the cookie to recover the session context. Stickiness is defined at a target group level.

Native IPv6 Support

Application Load Balancers support native Internet Protocol version 6 (IPv6) in a VPC. This will allow clients to connect to the Application Load Balancer via IPv4 or IPv6.

Request Tracing

The Application Load Balancer injects a new custom identifier “X-Amzn-Trace-Id” HTTP header on all requests coming into the load balancer. Request tracing allows you to track a request by its unique ID as it makes its way across various services that make up the bulk of traffic for your websites and distributed applications. You can use the unique trace identifier to uncover any performance or timing issues in your application stack at the granularity of an individual request.

Redirects

Application Load Balancer can redirect an incoming request from one URL to another URL. This includes the capability to redirect HTTP requests to HTTPS requests, which allows you to meet your compliance goal of secure browsing, while being able to achieve better search ranking and SSL/TLS score for your site. You can also use redirects to send users to a different web site; for example, redirecting from an old version of an application to a new version.

Fixed Response

Application Load Balancer can control which client requests are served by your applications. This enables you to respond to incoming requests with HTTP error response codes and custom error messages from the load balancer itself, without forwarding the request to the application.

WebSockets Support

WebSockets allows a server to exchange real-time messages with end-users without the end users having to request (or poll) the server for an update. The WebSockets protocol provides bi-directional communication channels between a client and a server over a long-running TCP connection.

Server Name Indication (SNI)

Server Name Indication (SNI) is an extension to the TLS protocol by which a client indicates the hostname to connect to at the start of the TLS handshake. The load balancer can present multiple certificates through the same secure listener, which enables it to support multiple secure websites using a single secure listener. Application Load Balancers also support a smart certificate selection algorithm with SNI. If the hostname indicated by a client matches multiple certificates, the load balancer determines the best certificate to use based on multiple factors including the capabilities of the client.

IP addresses as Targets

You can load balance any application hosted in AWS or on-premises using IP addresses of the application backends as targets. This allows load balancing to an application backend hosted on any IP address and any interface on an instance. Each application hosted on the same instance can have an associated security group and use the same port. You can also use IP addresses as targets to load balance applications hosted in on-premises locations (over a Direct Connect or VPN connection), peered VPCs and EC2-Classic (using ClassicLink). The ability to load balance across AWS and on-premises resources helps you migrate-to-cloud, burst-to-cloud or failover-to-cloud.

Lambda functions as Targets

Application Load Balancers support invoking Lambda functions to serve HTTP(S) requests enabling users to access serverless applications from any HTTP client, including web browsers. You can register Lambda functions as targets for a load balancer and leverage the support for content-based routing rules to route requests to different Lambda functions. You can use an Application Load Balancer as a common HTTP endpoint for applications that use servers and serverless computing. You can build an entire website

using Lambda functions or combine EC2 instances, containers, on-premises servers and Lambda functions to build applications.

Content-based Routing

If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request such as Host field, Path URL, HTTP header, HTTP method, Query string or Source IP address.

Host-based Routing : You can route a client request based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer.

Path-based Routing : You can route a client request based on the URL path of the HTTP header.

HTTP header-based routing : You can route a client request based on the value of any standard or custom HTTP header.

HTTP method-based routing : You can route a client request based on any standard or custom HTTP method.

Query string parameter-based routing : You can route a client request based on query string or query parameters.

Source IP address CIDR-based routing : You can route a client request based on source IP address CIDR from where the request originates.

Containerized Application Support

Application Load Balancer provides enhanced container support by load balancing across multiple ports on a single Amazon EC2 instance. Deep integration with the Amazon EC2 Container Service (ECS), provides a fully-managed container offering. ECS allows you to specify a dynamic port in the ECS task definition, giving the container an unused port when it is scheduled on the EC2 instance. The ECS scheduler automatically adds the task to the load balancer using this port.

Web Application Firewall

You can now use AWS WAF to protect your web applications on your Application Load Balancers. AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.

Slow Start Mode with Load-Balancing Algorithm

Application Load Balancer supports a round-robin load-balancing algorithm. Additionally, Application Load Balancer supports a slow start mode with the round-robin algorithm that allows you to add new targets without overwhelming them with a flood of requests. With the slow start mode, targets warm up before accepting their fair share of requests based on a ramp-up period that you specify. Slow start is very useful for applications that depend on cache and need a warm-up period before being able to respond to requests with optimal performance.

User Authentication

You can offload the authentication functionality from your apps into Application Load Balancer. Application Load Balancer will securely authenticate users as they access cloud applications. Application Load Balancer is seamlessly integrated with Amazon Cognito, which allows end users to authenticate through social identity providers such as Google, Facebook, and Amazon, and through enterprise identity providers such as Microsoft Active Directory via SAML or any OpenID Connect-compliant identity provider (IdP). If you already have a custom IdP solution that is OpenID Connect-compatible, Application Load Balancer can also authenticate enterprise users by directly connecting with your identity provider.



Discover more about Elastic Load Balancing

Visit the features page.

[Learn more »](#)



Sign up for a free account

Instantly get access to the AWS Free Tier.

[Sign up »](#)



Start building in the console

Get started with Elastic Load Balancing in the AWS Console.

[Sign In »](#)

Sign In to the Console

Learn About AWS

[What Is AWS?](#)
[What Is Cloud Computing?](#)
[What Is DevOps?](#)
[What Is a Container?](#)
[What Is a Data Lake?](#)
[AWS Cloud Security](#)
[What's New](#)
[Blogs](#)
[Press Releases](#)

Help

[Contact Us](#)
[AWS Careers](#)
[File a Support Ticket](#)
[Knowledge Center](#)
[AWS Support Overview](#)
[Legal](#)

Resources for AWS

[Getting Started](#)
[Training and Certification](#)
[AWS Solutions Portfolio](#)
[Architecture Center](#)
[Product and Technical FAQs](#)
[Analyst Reports](#)
[AWS Partner Network](#)

Developers on AWS

[Developer Center](#)
[SDKs & Tools](#)
[.NET on AWS](#)
[Python on AWS](#)
[Java on AWS](#)
[PHP on AWS](#)
[Javascript on AWS](#)

[Create an AWS Account](#)



Amazon is an Equal Opportunity Employer: *Minority / Women / Disability / Veteran / Gender Identity / Sexual Orientation / Age.*

Language

عربي |

Bahasa Indonesia |

Deutsch |

English |

Español |

Français |

Italiano |

Português |

Tiếng Việt |

Türkçe |

Русский |

ไทย |

日本語 |

한국어 |

中文 (简体) |

中文 (繁體)

Privacy

|

Site Terms

|

Cookie Preferences

|

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.