AWS Documentation (/index.html) » Elastic Load Balancing (/elasticloadbalancing/index.html) » Application Load Balancers (index.html) » What Is an Application Load Balancer?

What Is an Application Load Balancer?

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. This guide discusses Application Load Balancers. For more information about Network Load Balancers, see the User Guide for Network Load Balancers (https://docs.aws.amazon.com/elasticloadbalancing/latest/network/) . For more information about Classic Load Balancers, see the User Guide for Classic Load Balancers (https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/) .

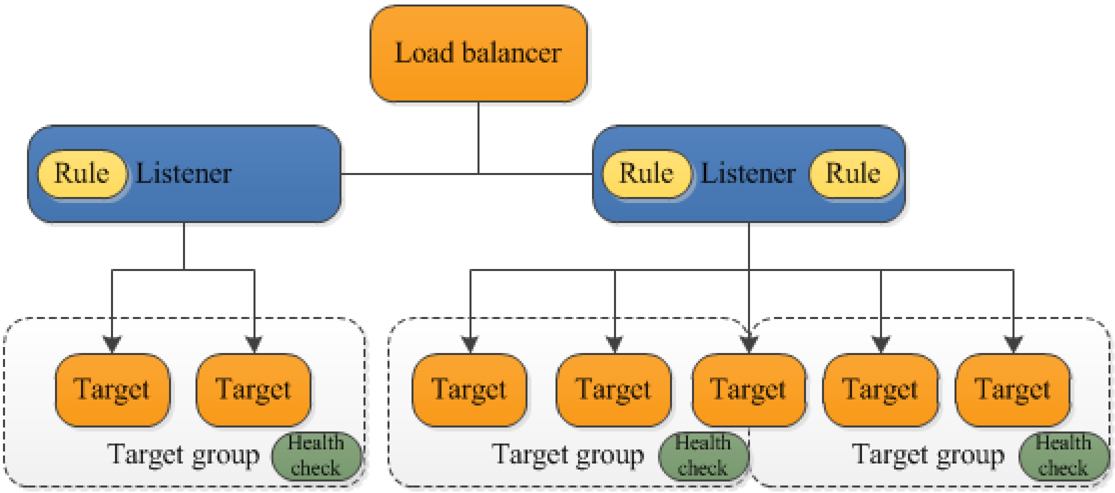
Application Load Balancer Components

A *load balancer* serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. You add one or more listeners to your load balancer.

A *listener* checks for connection requests from clients, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. Each rule consists of a priority, one or more actions, and one or more conditions. When the conditions for a rule are met, then its actions are performed. You must define a default rule for each listener, and you can optionally define additional rules.

Each *target group* routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify. You can register a target with multiple target groups. You can configure health checks on a per target group basis. Health checks are performed on all targets registered to a target group that is specified in a listener rule for your load balancer.

The following diagram illustrates the basic components. Notice that each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. One target is registered with two target groups.



For more information, see the following documentation:

- Load Balancers (application-load-balancers.html)
- Listeners (load-balancer-listeners.html)
- Target Groups (load-balancer-target-groups.html)

Application Load Balancer Overview

An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application. Elastic Load Balancing scales your load balancer as traffic to your application changes over time. Elastic Load Balancing can scale to the vast majority of workloads automatically.

You can configure health checks, which are used to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

For more information, see How Elastic Load Balancing Works (https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html) in the *Elastic Load Balancing User Guide*.

Benefits of Migrating from a Classic Load Balancer

Using an Application Load Balancer instead of a Classic Load Balancer has the following benefits:

- Support for path-based routing. You can configure rules for your listener that forward requests based on the URL in the request. This enables you to structure your application as smaller services, and route requests to the correct service based on the content of the URL.
- Support for host-based routing. You can configure rules for your listener that forward requests based on the host field in the HTTP header. This enables you to route requests to multiple domains using a single load balancer.
- Support for routing based on fields in the request, such as standard and custom HTTP headers and methods, query parameters, and source IP addresses.
- Support for routing requests to multiple applications on a single EC2 instance. You can register each instance or IP address with the same target group using multiple ports.
- Support for redirecting requests from one URL to another.
- Support for returning a custom HTTP response.
- Support for registering targets by IP address, including targets outside the VPC for the load balancer.
- Support for registering Lambda functions as targets.
- Support for the load balancer to authenticate users of your applications through their corporate or social identities before routing requests.
- Support for containerized applications. Amazon Elastic Container Service (Amazon ECS) can select an unused port when scheduling a task and register the task with a target group using this port. This enables you to make efficient use of your clusters.
- Support for monitoring the health of each service independently, as health checks are defined at the target group level and many CloudWatch metrics are reported at the target group level. Attaching a target group to an Auto Scaling group enables you to scale each service dynamically based on demand.
- Access logs contain additional information and are stored in compressed format.
- Improved load balancer performance.

For more information about the features supported by each load balancer type, see Comparison of Elastic Load Balancing Products (https://aws.amazon.com/elasticloadbalancing/details/#compare) .

How to Get Started

To create an Application Load Balancer, try one of the following tutorials:

- Getting Started with Elastic Load Balancing (https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/load-balancer-getting-started.html) in the *Elastic Load Balancing User Guide*.
- Tutorial: Use Path-Based Routing with Your Application Load Balancer (tutorial-load-balancer-routing.html)

- Tutorial: Use Microservices as Targets with Your Application Load Balancer (tutorial-target-ecs-containers.html)

Related Services

Elastic Load Balancing works with the following services to improve the availability and scalability of your applications.

- **Amazon EC2** — Virtual servers that run your applications in the cloud. You can configure your load balancer to route traffic to your EC2 instances.
- **Amazon EC2 Auto Scaling** — Ensures that you are running your desired number of instances, even if an instance fails, and enables you to automatically increase or decrease the number of instances as the demand on your instances changes. If you enable Auto Scaling with Elastic Load Balancing, instances that are launched by Auto Scaling are automatically registered with the load balancer, and instances that are terminated by Auto Scaling are automatically de-registered from the load balancer.
- **AWS Certificate Manager** — When you create an HTTPS listener, you can specify certificates provided by ACM. The load balancer uses certificates to terminate connections and decrypt requests from clients. For more information, see SSL Certificates (create-https-listener.html#https-listener-certificates) .
- **Amazon CloudWatch** — Enables you to monitor your load balancer and take action as needed. For more information, see CloudWatch Metrics for Your Application Load Balancer (load-balancer-cloudwatch-metrics.html) .
- **Amazon ECS** — Enables you to run, stop, and manage Docker containers on a cluster of EC2 instances. You can configure your load balancer to route traffic to your containers. For more information, see Service Load Balancing (https://docs.aws.amazon.com/AmazonECS/latest/developerguide/service-load-balancing.html) in the *Amazon Elastic Container Service Developer Guide*.
- **AWS Global Accelerator** — Improves the availability and performance of your application. Use an accelerator to distribute traffic across multiple load balancers in one or more AWS Regions. For more information, see the AWS Global Accelerator Developer Guide (https://docs.aws.amazon.com/global-accelerator/latest/dg/) .
- **Route 53** — Provides a reliable and cost-effective way to route visitors to websites by translating domain names (such as `www.example.com`) into the numeric IP addresses (such as `192.0.2.1`) that computers use to connect to each other. AWS assigns URLs to your resources, such as load balancers. However, you might want a URL that is easy for users to remember. For example, you can map your domain name to a load balancer.
- **AWS WAF** — You can use AWS WAF with your Application Load Balancer to allow or block requests based on the rules in a web access control list (web ACL). For more information, see Application Load Balancers and AWS WAF (application-load-balancers.html#load-balancer-waf) .

To view information about services that are integrated with your load balancer, select your load balancer in the AWS Management Console and choose the **Integrated services** tab.

Pricing

With your load balancer, you pay only for what you use. For more information, see Elastic Load Balancing Pricing (https://aws.amazon.com/elasticloadbalancing/pricing/) .