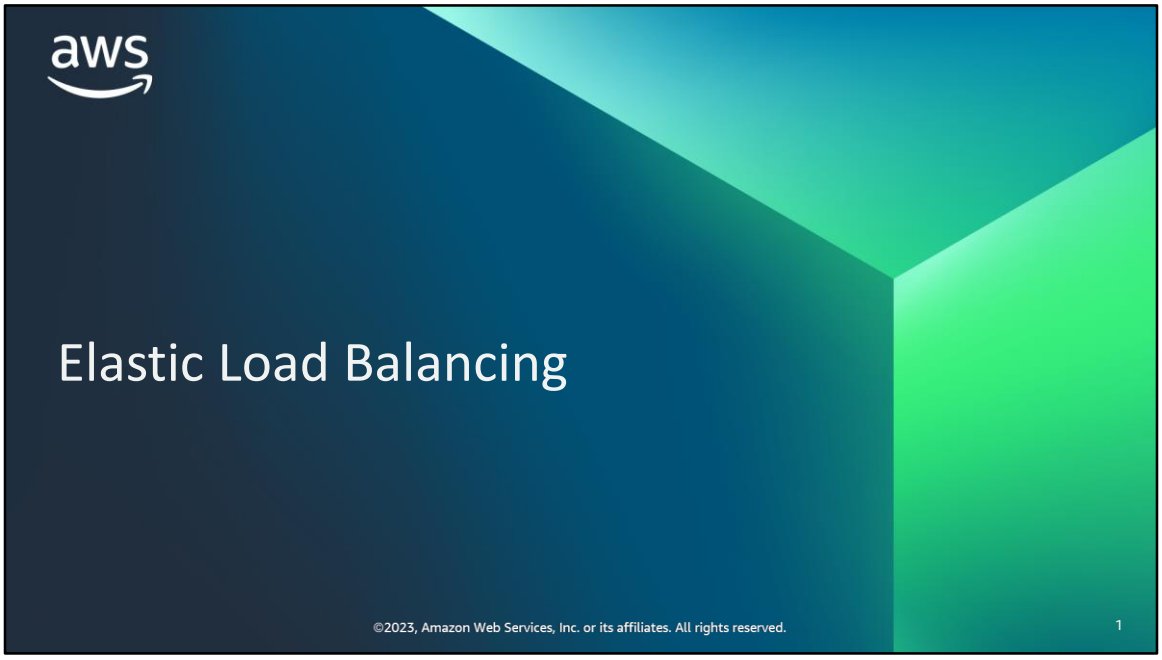
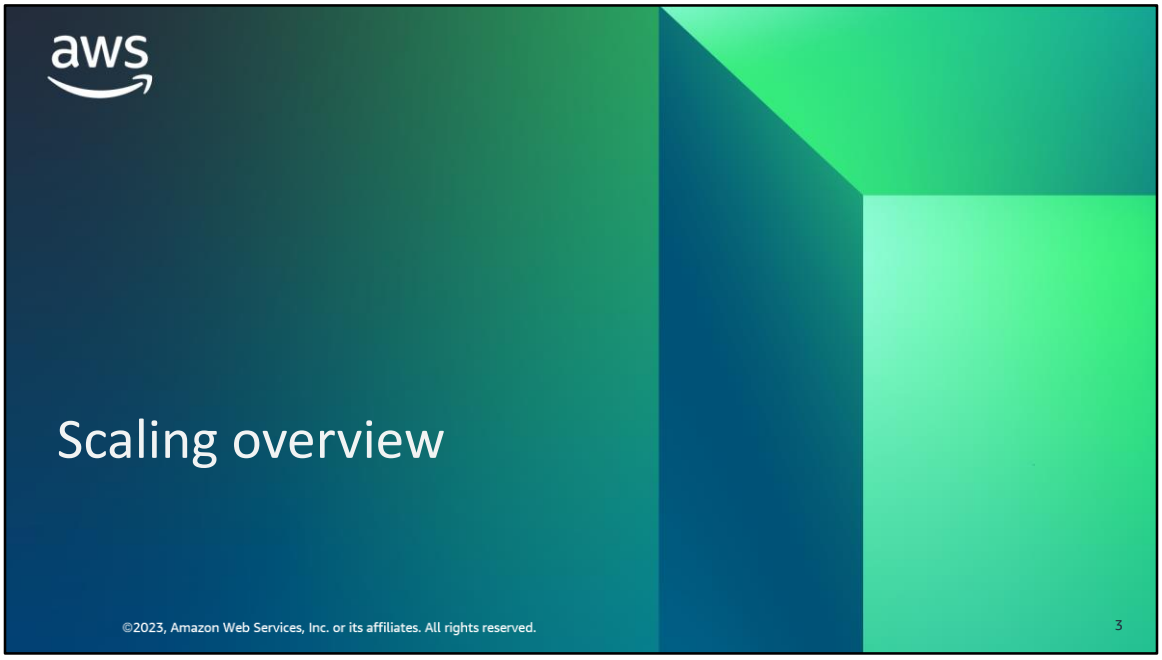


At the core of the lesson

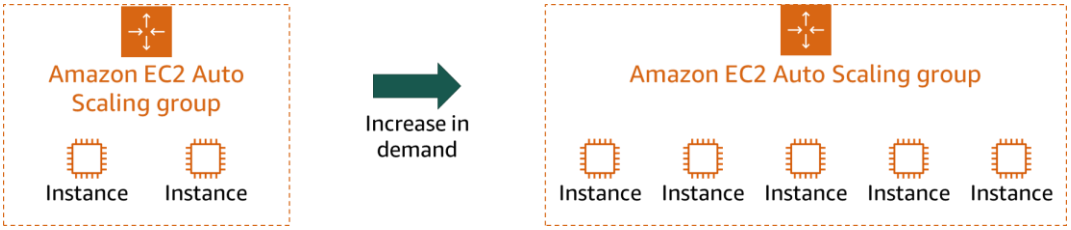
You will learn how to do the following:

- Identify the importance of scaling.
- Describe the Elastic Load Balancing (ELB) service.
- Describe the different types of ELB load balancers.



What is scaling?

- Scaling is the ability to increase or decrease compute capacity to meet fluctuating demand.
  - Scale out when demand increases.
  - Scale in when capacity needs decrease.
- You can scale manually or automatically (auto scaling).



In a traditional data center environment, the scalability of your system is bound by your hardware. Consider the example of a tax preparation business in the United States. US taxpayers must file their taxes by April 15.

- Online tax preparation companies know that they will experience a steady flow of traffic that starts near the middle of January. That traffic peaks close to the April 15 deadline.
- In a data center, anticipating this 4-month period of heavy utilization requires provisioning enough physical servers to handle the anticipated load.

However, what happens to those servers during the rest of the year? They sit idle in the data center.

In the cloud, because computing power is a programmatic resource, you can take a more flexible approach to the issue of scaling. You can do the following:

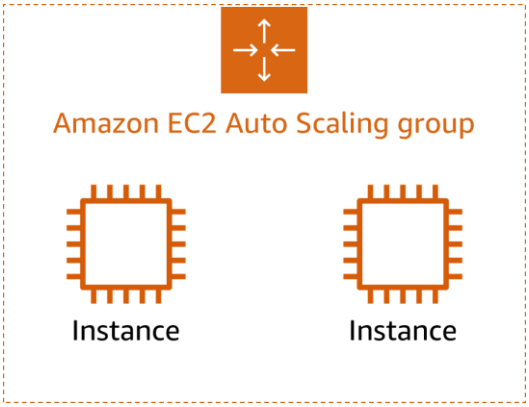
- Create new Amazon Elastic Compute Cloud (Amazon EC2) instances in advance of known peak periods in a business cycle (such as tax filing deadlines).
- Use monitoring, and programmatically scale out servers when critical resources—such as average CPU utilization across the fleet—become constrained.
- Automatically scale in the number of resources that are used during peak times when demand for the business returns to its baseline.

Amazon EC2 Auto Scaling helps ensure that you have the correct number of instances available to handle the load for your application. You can specify premade selections such as the maximum, minimum, or capacity thresholds in order to help ensure that your solution meets demand while also maintaining your limits.

In other words, you pay for the resources that you need when you need them.

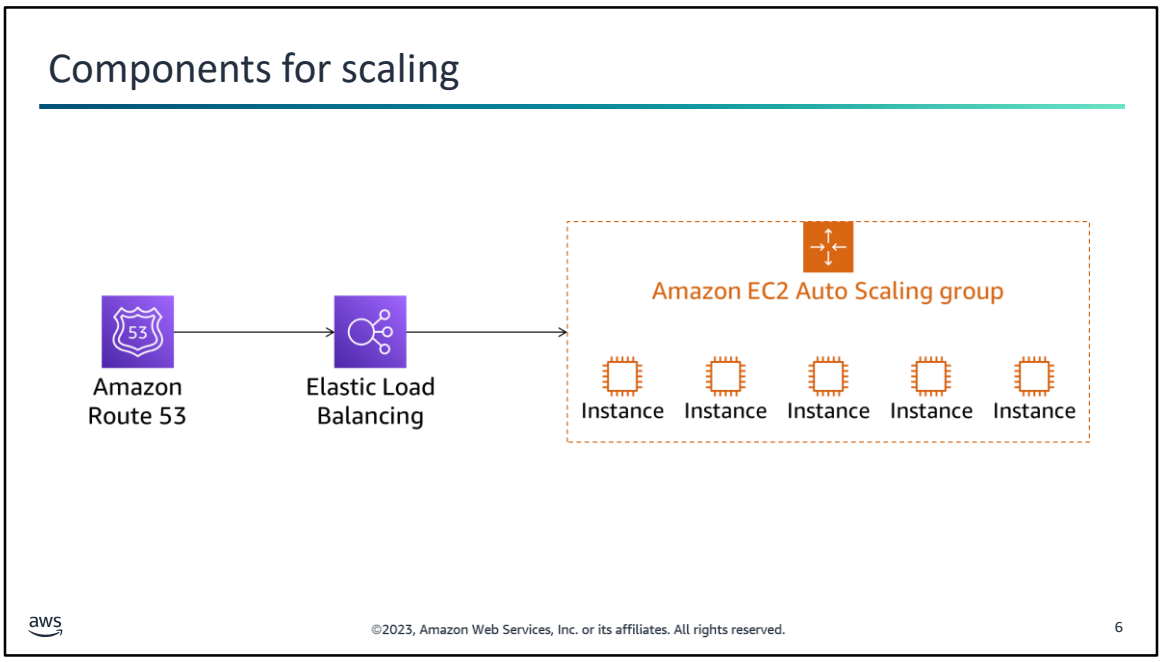
Benefits of Auto Scaling

- Fault tolerance
- High availability
- Performance
- Cost optimization



Adding Amazon EC2 Auto Scaling to your application architecture is one way to maximize the benefits of the AWS Cloud. When you use Amazon EC2 Auto Scaling, your applications gain the following benefits:

- Better fault tolerance: Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.
- Better availability: Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
- Better performance: When traffic increases, having more instances gives you the ability to distribute and share the work to maintain a good response time.
- Better cost management: Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

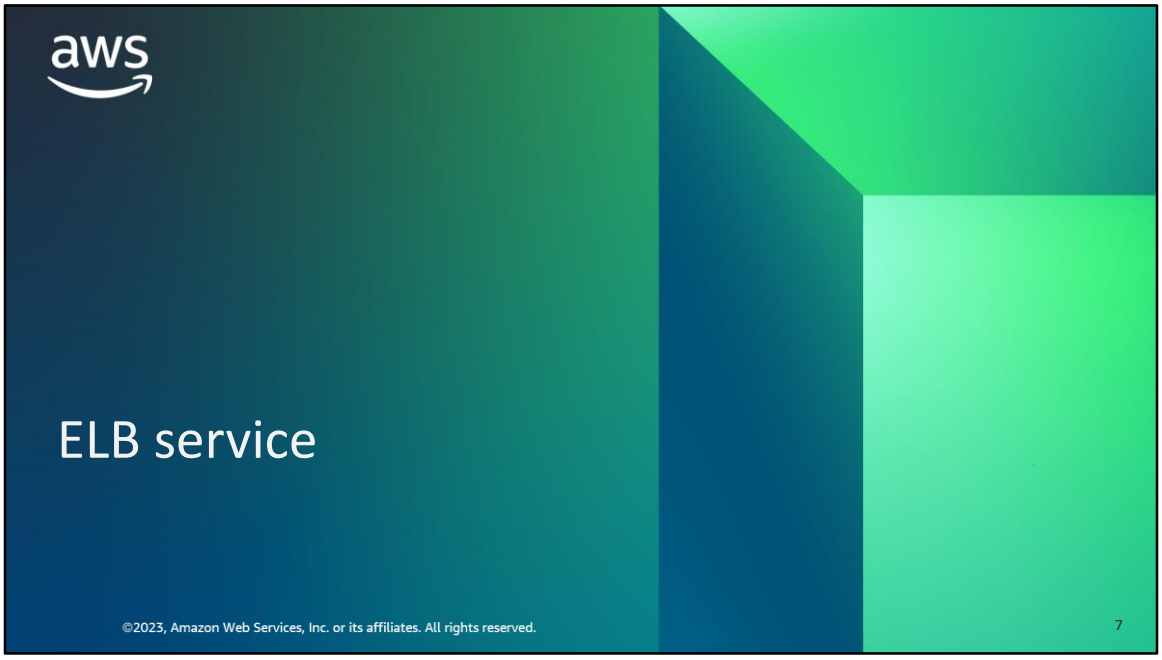


Which resources do you need to implement such a system? You can use several AWS services together to create a scalable, on-demand architecture. The components necessary for scaling are Amazon Route 53, ELB, and Amazon EC2 Auto Scaling groups.

Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses a reliable way to route users to internet applications. It translates 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. Route 53 entries are often configured to point to an ELB load balancer.

ELB automatically distributes incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses. ELB load balancers are often configured to point to Amazon EC2 Auto Scaling groups.

Each Amazon EC2 Auto Scaling group contains a collection of EC2 instances. These instances share similar characteristics and are treated as a logical grouping for the purposes of scaling and management. They help you maintain application availability and give you the ability to dynamically scale capacity up or down automatically according to conditions that you define. Any instances launched or terminated within the Auto Scaling group are automatically registered with the load balancer.



One of the services that you use to implement scaling in the AWS Cloud is ELB.

What is ELB?

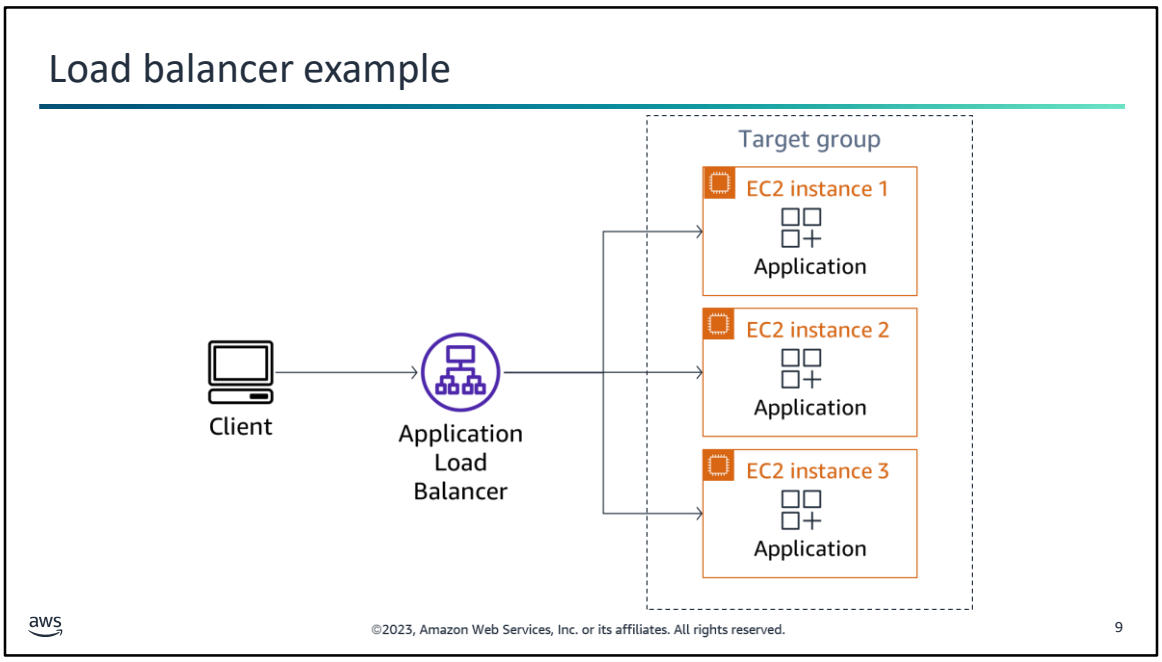
ELB

ELB does the following:

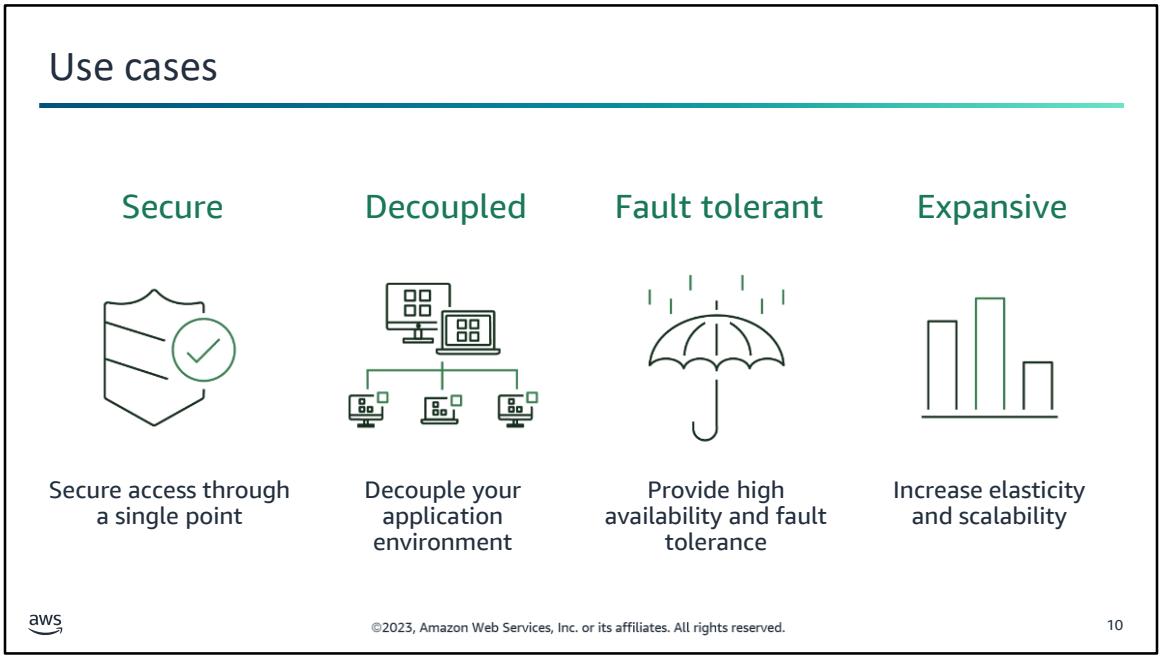
- Automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones
- Monitors the health of its registered targets and routes traffic to only the healthy targets
- Automatically scales your load balancer capacity in response to changes in incoming traffic

aws

Modern high-traffic websites must serve hundreds of thousands—if not millions—of concurrent requests from users or clients. They must return the correct text, images, video, or application data in a fast and reliable manner. To scale cost-effectively to meet these high volumes, modern computing best practices generally require adding more servers. On AWS, you scale your capacity by using ELB and Auto Scaling groups.

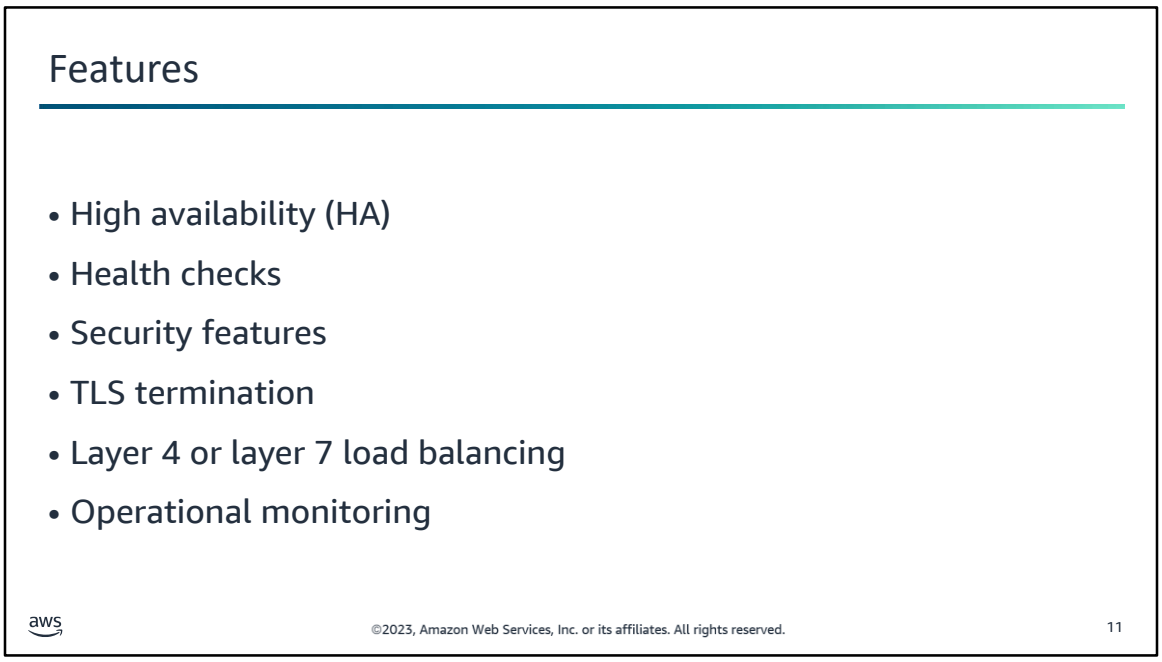


A load balancer acts as a traffic director that sits in front of your servers and routes client requests. It routes requests across all servers that can fulfill those requests in a manner that maximizes speed and capacity utilization. It helps ensure that no one server is overworked, which could degrade performance. If a single server goes down, the load balancer redirects traffic to the remaining online servers. These online servers might include EC2 instances, containers, and IP addresses. When a new server is added to the server group, the load balancer automatically starts to send requests to it.



A load balancer is used for many reasons, such as the following:

- To secure access to your web servers through a single exposed point of access
- To decouple your environment by using both public and internal load balancers
- To provide high availability and fault tolerance with the ability to distribute traffic across multiple Availability Zones
- To increase elasticity and scalability with minimal overhead



As mentioned previously, a load balancer distributes workloads across multiple compute resources, such as virtual servers. You can configure ELB load balancers in the Amazon EC2 service area on the AWS Management Console. Alternatively, you can also invoke the service through the AWS Command Line Interface (AWS CLI) or SDKs.

ELB load balancers include the following key features:

- High availability (HA): ELB load balancers can distribute traffic across multiple targets—including EC2 instances, containers, and IP addresses—in a single Availability Zone or multiple Availability Zones.
- Health checks: You can configure ELB load balancers to detect unhealthy targets, stop sending traffic to these targets, and then spread the load across the remaining healthy targets.
- Security: You can create and manage security groups that are associated with load balancers. You can also create an internal (non-internet-facing) load balancer.
- TLS termination: The load balancers include integrated certificate management and SSL decryption. Thus, you can centrally manage the SSL settings of the load balancer and offload CPU-intensive work from your applications.
- Layer 4 or layer 7 load balancing:
    - You can load balance HTTP and HTTPS applications for features that are specific to layer 7. Recall that layer 7 is the application layer in the Open Systems Interconnection (OSI) model.
    - You can also choose to use only layer 4 load balancing for applications that rely only on TCP. Recall that layer 4 is the transport layer in the OSI model.
- Operational monitoring: ELB load balancers can work with Amazon CloudWatch metrics and request tracing. You can use these resources to monitor application performance in real time.

ELB provides many of the operational monitoring metrics by default. You can use these metrics to see HTTP responses and the number of healthy and unhealthy hosts behind the load balancer. You can also filter these metrics based on the Availability Zone of the backend instances or based on the load balancer that you are using.

For health checks, you can use the load balancer to see the number of healthy and unhealthy Amazon EC2 hosts

behind the load balancer. The load balancer accomplishes this objective with an attempted connection request to the EC2 instance. To discover the availability of your EC2 instances, a load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances. These tests are called health checks.

Elastic load balancers

Types of ELB load balancers

Application	Gateway	Network	Classic
Provides advanced load balancing for traffic (HTTP, HTTPS, and Google Remote Procedure Call, gRPC)	Provides load balancing for virtual appliances and network traffic destinations (all types of IP traffic)	Provides load balancing for TCP traffic (TCP, UDP, and TLS)	Provides support and load balancing for HTTP, HTTPS, SSL/TLS, and TCP traffic on the EC2-Classic platform (previous generation)
Is used for flexible application management	Is used for virtual appliance management	Is used for extreme performance and as a static IP address for your application	Is used for existing applications that were built on the EC2-Classic platform
Operates at the application level (layer 7)	Operates as a gateway at the network level (layer 3) and as a load balancer at the transport level (layer 4)	Operates at the transport level (layer 4)	Operates at both the application level (layer 7) and the transport level (layer 4)

ELB offers four types of load balancers: Application Load Balancer, Gateway Load Balancer, Network Load Balancer, and Classic Load Balancer.

There is a key difference in how the load balancer types are configured. With Application Load Balancers, Gateway Load Balancers, and Network Load Balancers, you register targets in target groups and route traffic to the target groups. These are the AWS version 2 load balancers. With the previous version, Classic Load Balancers, you register instances with the load balancer.

The first type of load balancer is the Application Load Balancer, which functions at the application level. It supports targets with any operating system currently supported by the Amazon EC2 service. It supports a pair of open standard protocols (WebSocket and HTTP/2) and can provide additional visibility into the health of target instances and containers.

- Application Load Balancers provide advanced request routing that supports modern application architectures, including microservices and container-based applications.
- Application Load Balancers are recommended for all other Amazon Virtual Private Cloud (Amazon VPC) use cases.

Websites and mobile apps that run in containers or on EC2 instances can benefit from the use of Application Load Balancers.

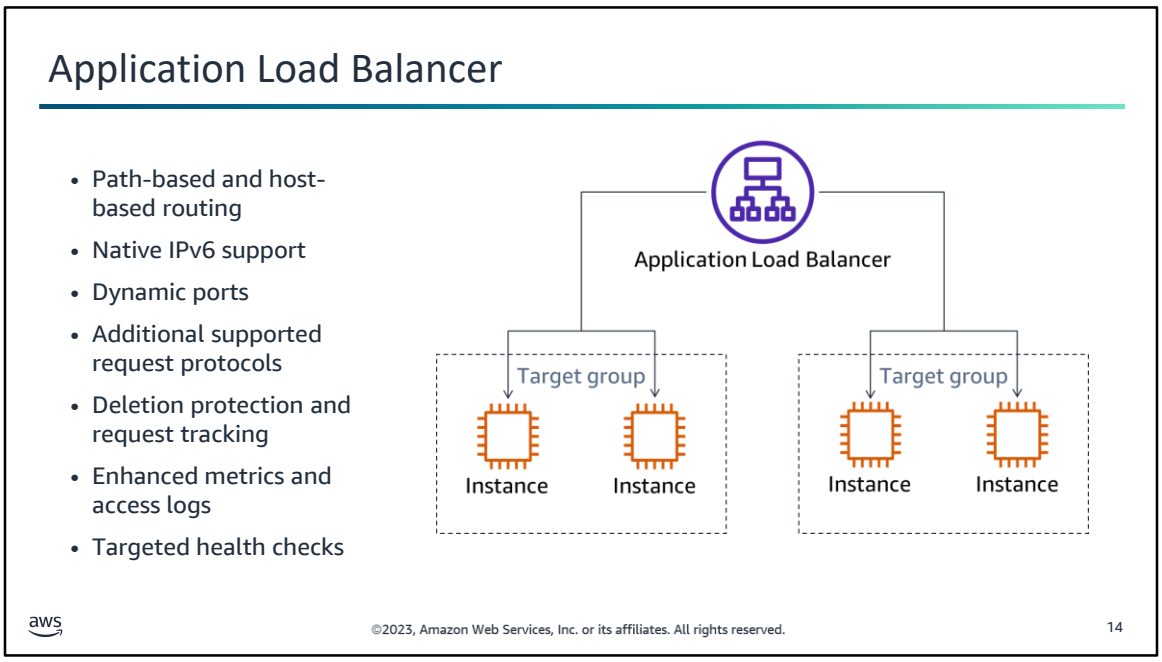
The second type of load balancer, a Gateway Load Balancer, is a detour along the traffic's route to the destination and listens to all types of IP traffic (including TCP, UDP, ICMP, GRE, ESP, and others). It provides both gateway (layer 3, network level) and load balancing (layer 4, transport level) capabilities. The Gateway Load Balancer also helps you deploy, scale, and manage your third-party virtual appliances. It gives you one gateway for distributing traffic across multiple virtual appliances while scaling them in or out based on demand.

The next type of load balancer is the Network Load Balancer, which is designed to handle tens of millions of requests per second while maintaining high throughput at low latency. Network Load Balancers work well for load balancing TCP traffic.

- You can use the same API to programmatically control both Network Load Balancers and Application Load Balancers. The API includes full programmatic control of target groups and targets.
- A Network Load Balancer works well for load balancing TCP traffic. It is optimized to handle sudden and volatile traffic patterns while using a single static IP address per Availability Zone.
- Network Load Balancers are recommended for load balancing of TCP traffic in a VPC.

Finally, the Classic Load Balancer provides basic load balancing across multiple EC2 instances. It operates at both request and connection levels. Classic Load Balancers are intended for applications that were built within the deprecated EC2-Classic platform.

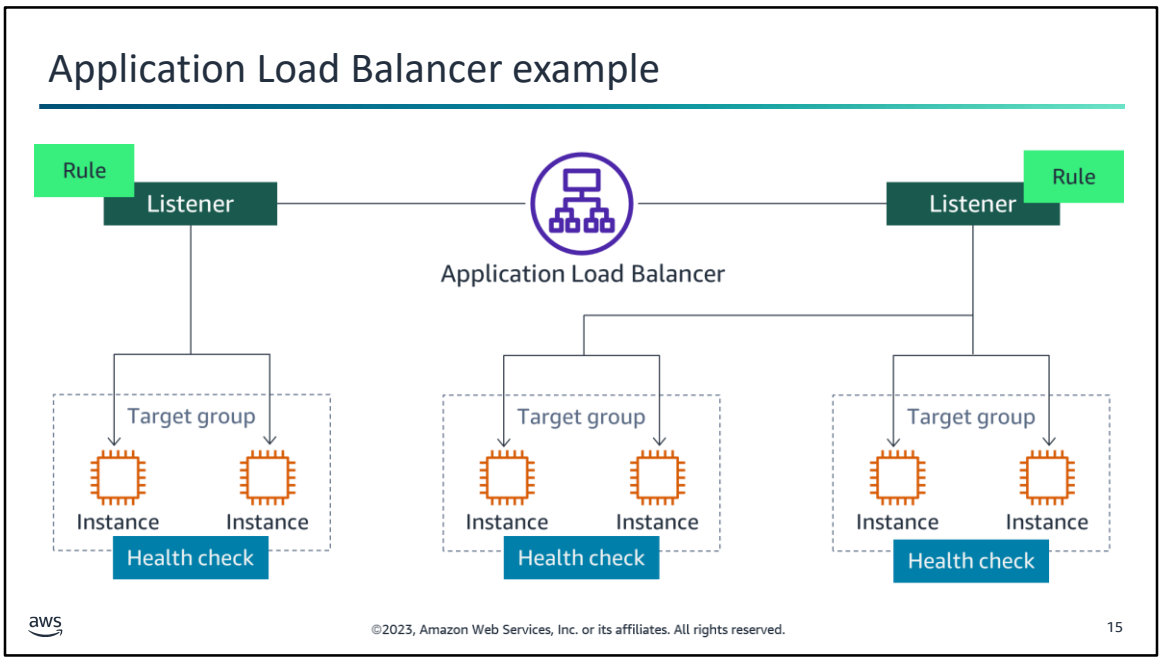
For a side-by-side comparison of the features that are available for the three load balancer types, see Elastic Load Balancing Features at https://aws.amazon.com/elasticloadbalancing/features/.



An Application Load Balancer functions at the application layer, the seventh layer of the OSI model.

The Application Load Balancer offers most of the features that the Classic Load Balancer has. Using an Application Load Balancer instead of a Classic Load Balancer has the following benefits:

- Path-based and host-based routing: With this routing, you can structure your application as smaller services and route requests to the correct service based on the content of the URL or to multiple domains by using a single load balancer.
- Native IPv6 support: Application Load Balancers support dual stack and IPv6 only.
- Dynamic ports: Application Load Balancers support dynamic host port mapping, which you can use to have multiple tasks from a single service on the same instance. You can register an instance or IP address with multiple target groups, each on a different port.
- Additional supported request protocols: Application Load Balancers send requests to targets by using HTTP/1.1. You can use the protocol version to send requests to targets by using HTTP/2 or gRPC.
- Deletion protection and request tracking: You can turn on deletion protection to prevent your Application Load Balancer from being accidentally deleted.
- Enhanced metrics and access logs: With enhanced metrics, you can monitor your load balancer and take action as needed. Access logs contain additional information and are stored in compressed format.
- Targeted health checks: Heath checks monitor the health of each service independently. Health checks are defined at the target group level, and many CloudWatch metrics are reported at the target group level. By attaching a target group to an Auto Scaling group, you can scale each service dynamically based on demand.

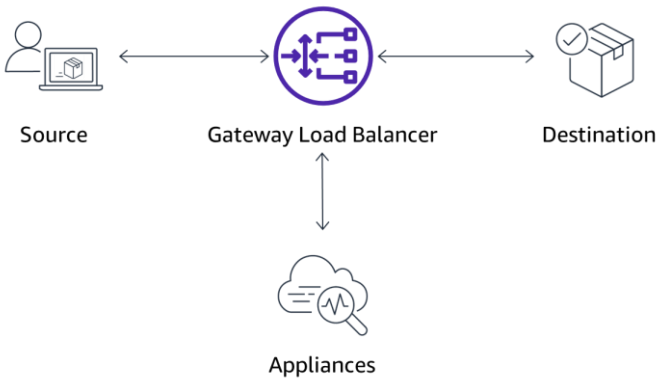


In the example, the Application Load Balancer routes requests to backend targets. When you configure listeners for the load balancer, you create rules to direct how requests that the load balancer receives are routed to the backend targets.

To add those targets to the load balancer and to configure the health check that the load balancer uses for the targets, you create target groups. You register a target to a target group and can register a target to more than one target group. In the example, the Application Load Balancer receives the user request and evaluates the listener rule that matches the request's protocol and port number. Based on the rule evaluation, the Application Load Balancer routes the request to the appropriate target group instance.

Gateway Load Balancer

- Provides layer 3 gateway and layer 4 load balancing
- Passes all layer 3 traffic through third-party virtual appliances
- Supports IP protocols
- Provides deletion protection and request tracking
- Provides enhanced metrics and access logs
- Provides targeted health checks

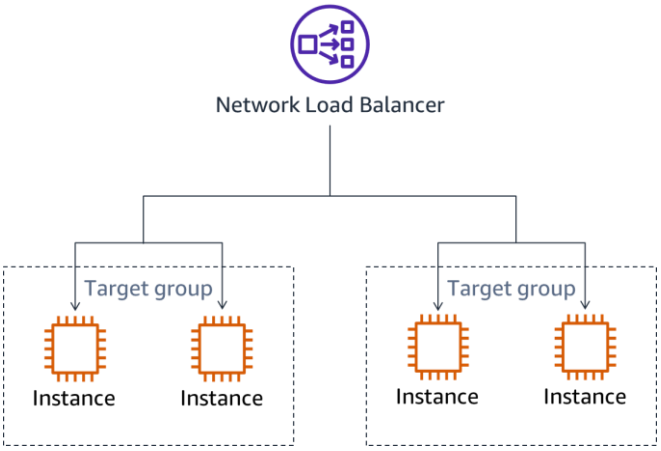


You should use a Gateway Load Balancer when deploying inline virtual appliances where network traffic is not destined for the Gateway Load Balancer itself. A Gateway Load Balancer transparently passes all layer 3 traffic through third-party virtual appliances and is invisible to the source and destination of the traffic.

In the example diagram, the source sends traffic to the destination. On its route to the destination, the network traffic arrives at the Gateway Load Balancer. The Gateway Load Balancer receives the traffic and forwards it to a healthy and available virtual appliance. The appliance then inspects the traffic and either forwards it back the Gateway Load Balancer or drops it. The Gateway Load Balancer forwards the returning traffic to a healthy and available destination. At the destination, the traffic arrives appearing unchanged. The response would follow the same path in reverse—going back out through the same appliance.

Network Load Balancer

- Sudden and volatile traffic patterns
- Single static IP address per Availability Zone
- Good option for applications that require extreme performance
- Visibility into HTTP responses
- Visibility into the number of healthy and unhealthy hosts
- Metric filtering based on Availability Zones or load balancer

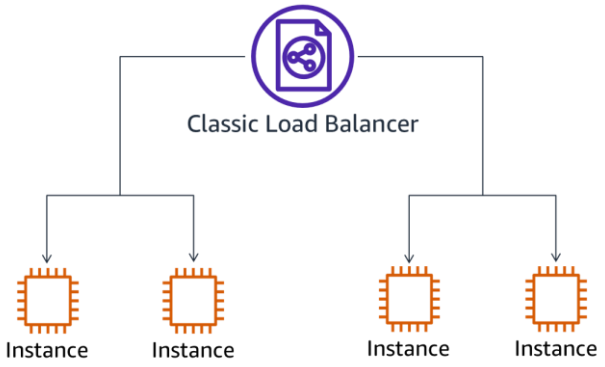


A Network Load Balancer functions at the fourth layer of the OSI model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration. Network Load Balancers are optimized to handle sudden and volatile traffic patterns while using a single static IP address per Availability Zone.

Because it handles millions of requests per second while maintaining ultra-low latencies, the Network Load Balancer works well for applications that require extreme performance.

Classic Load Balancer

- Provides access to servers through a single point
- Decouples the application environment
- Provides high availability and fault tolerance
- Increases elasticity and scalability

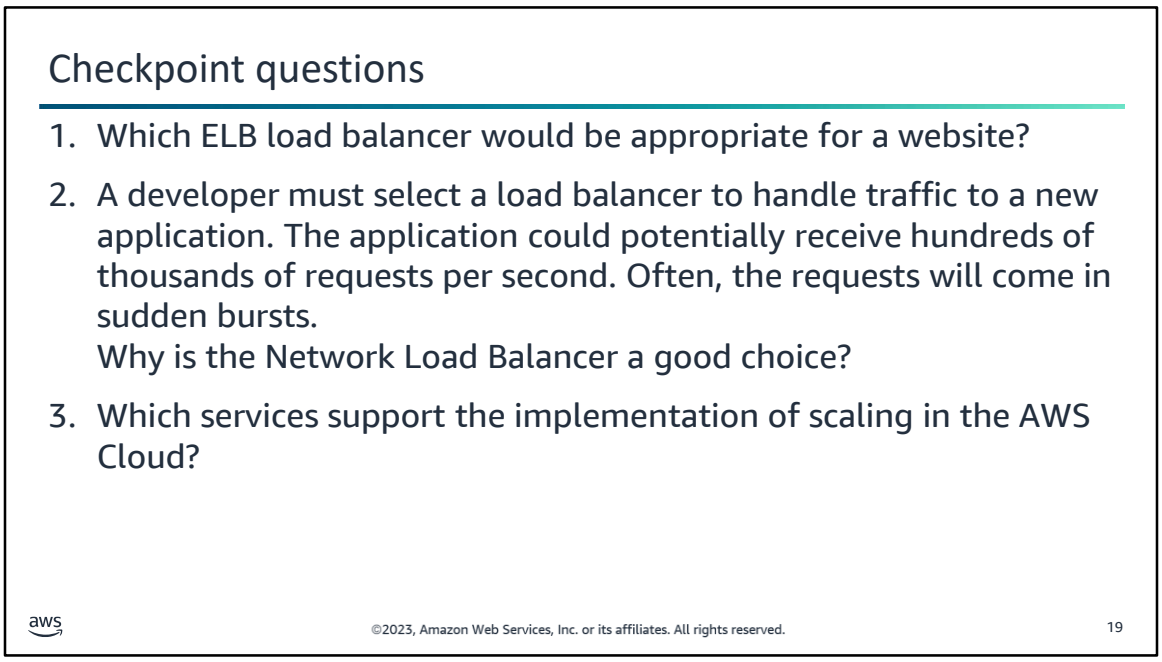


The EC2-Classic platform was introduced in the original release of Amazon EC2, and Classic Load Balancers are one of the first ELB load balancers offered on AWS. Within EC2-Classic, resources were not required to be launched within a VPC, so the overall functionality of the Classic Load Balancer is similar to the newer options. However, its configuration within the cloud environment differs due to the absence of a VPC.

Classic Load Balancers lack several of the newer features mentioned regarding the other load balancers available in version 2. However, using a Classic Load Balancer instead of an Application Load Balancer has the following benefits:

- Support for EC2-Classic
- Support for TCP and SSL listeners
- Support for sticky sessions using application-generated cookies

AWS is retiring the EC2-Classic platform, and any EC2-Classic or Classic Load Balancers that you might come across in your professional careers may be in the process of migrating to more up-to-date solutions. In addition, if an AWS account was created after December 4, 2013, it does not support EC2-Classic. Any EC2 instances in those accounts must be launched in a VPC.



The answers to the questions are as follows:

1. Which ELB load balancer would be appropriate for a website?

   Because websites use HTTP or HTTPS for communications, an Application Load Balancer would be the correct choice. You could also use the Classic Load Balancer because it also supports web protocols. However, if you must choose between Application Load Balancers and Classic Load Balancers, new website implementations should use an Application Load Balancer.

2. A developer must select a load balancer to handle traffic to a new application. The application could potentially receive hundreds of thousands of requests per second. Often, the requests will come in sudden bursts.
   Why is the Network Load Balancer a good choice?

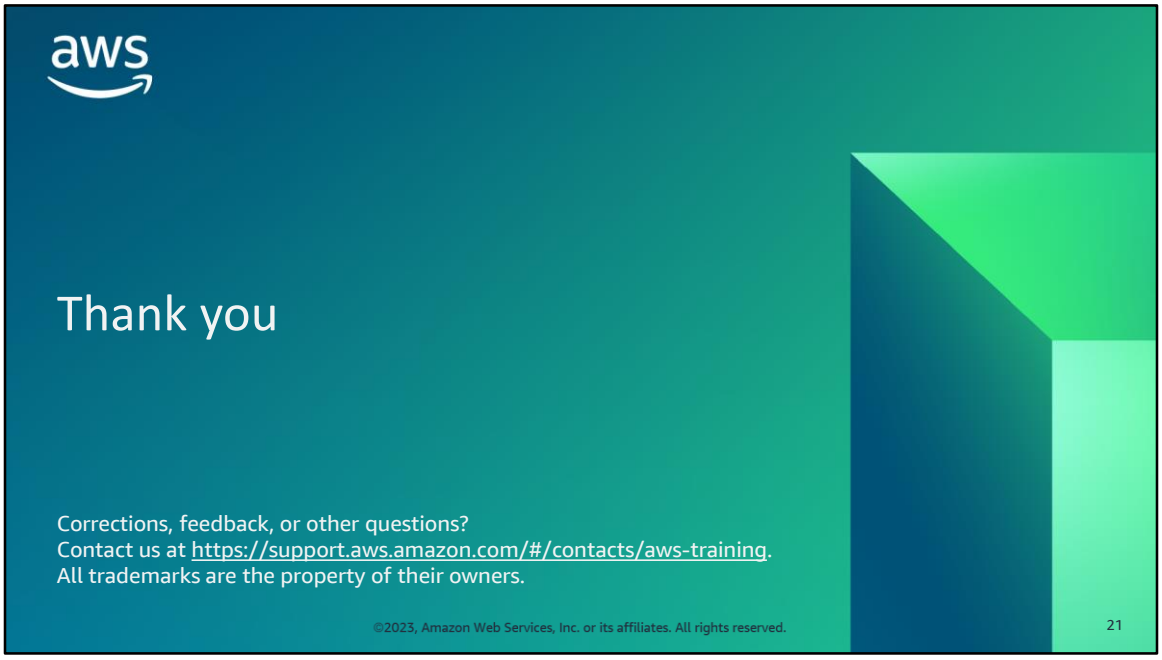
   Network Load Balancers are capable of handling millions of requests per second while maintaining ultra-low latencies. A Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.

3. Which services support the implementation of scaling in the AWS Cloud?

   Amazon Route 53, Elastic Load Balancing (ELB), and Amazon EC2 Auto Scaling

Key ideas

- ELB is a load balancing service.
- Load balancers automatically distribute incoming traffic load.
- ELB offers four types of load balancers:
  - Application Load Balancer
  - Gateway Load Balancer
  - Network Load Balancer
  - Classic Load Balancer
- ELB offers several monitoring tools.


aws
Thank you
Corrections, feedback, or other questions?
Contact us at https://support.aws.amazon.com/#/contacts/aws-training.
All trademarks are the property of their owners.
©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.
21