AWS Solution Architect Associate Certification Training – Module 13

13. Elastic Load Balancer

Overview of Elastic Load Balancer

Elastic Load Balancing distributes incoming application or network traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses, in multiple Availability Zones. Elastic Load Balancing scales your load balancer as traffic to your application changes over time, and can scale to the vast majority of workloads automatically.

Load Balancer Benefits

A load balancer distributes workloads across multiple compute resources, such as virtual servers. Using a load balancer increases the availability and fault tolerance of your applications. You can add and remove compute resources from your load balancer as your needs change, without disrupting the overall flow of requests to your applications. You can configure health checks, which are used to monitor the health of the compute resources so that the load balancer can send requests only to the healthy ones.

Features of Elastic Load Balancing

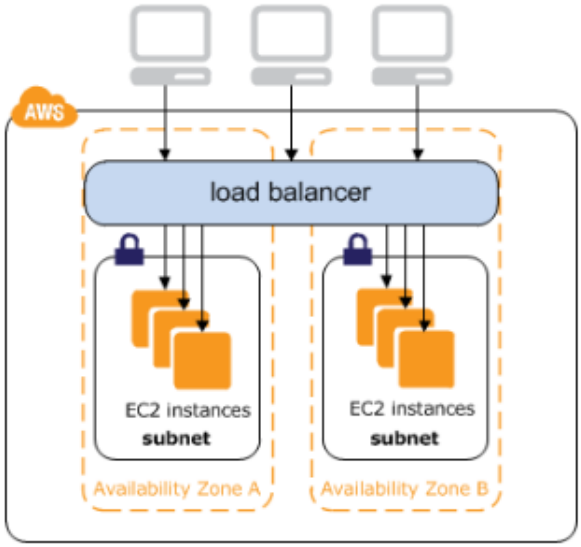
Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs.

The necessity of Elastic Load Balancer

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve greater levels of fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to distribute application traffic. Elastic Load Balancing detects unhealthy instances and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored. Customers can enable Elastic Load Balancing within a single or multiple Availability Zones for more consistent application performance. Elastic Load Balancing can also be used in an Amazon Virtual Private Cloud (“VPC”) to distribute traffic between application tiers in a virtual network that you define.

Internet Facing Load balancer

An Internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the Internet to the EC2 instances that are registered with the load balancer.



DNS setup for ELB

When your load balancer is created, it receives a public DNS name that clients can use to send requests. The DNS servers resolve the DNS name of your load balancer to the public IP addresses of the load balancer nodes for your load balancer. Each load balancer node is connected to the back-end instances using private IP addresses.

EC2-VPC: Load balancers in a VPC support IPv4 addresses only. The console displays a public DNS name with the following form

```
name-1234567890.region.elb.amazonaws.com
```

EC2-Classic: Load balancers in EC2-Classic support both IPv4 and IPv6 addresses. The console displays the following public DNS names:

The base public DNS name returns only IPv4 records. The public DNS name with the ipv6 prefix returns only IPv6 records. The public DNS name with the dualstack prefix returns both IPv4 and IPv6 records. We recommend that you enable IPv6 support by using the DNS name with the dualstack prefix to ensure that clients can access the load balancer using either IPv4 or IPv6.

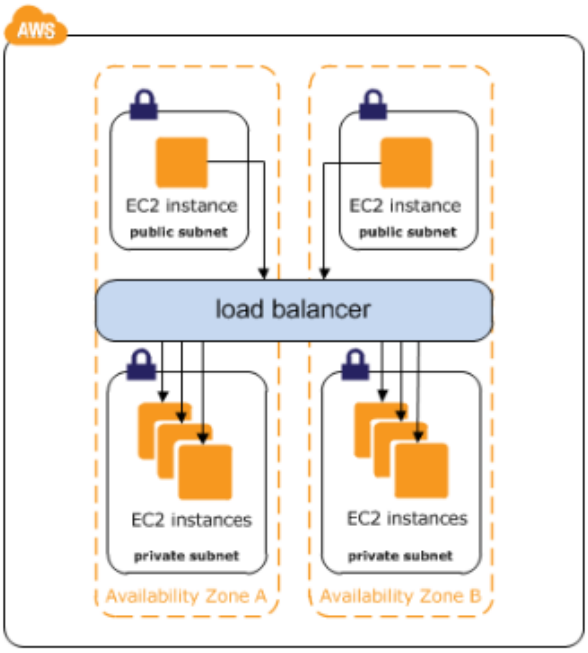
```
name-123456789.region.elb.amazonaws.com
ipv6.name-123456789.region.elb.amazonaws.com
dualstack.name-123456789.region.elb.amazonaws.com
```

Internal Facing Load balancer

When you create a load balancer in a VPC, you must choose whether to make it an internal load balancer or an Internet-facing load balancer.

The nodes of an Internet-facing load balancer have public IP addresses. The DNS name of an Internet-facing load balancer is publicly resolvable to the public IP addresses of the nodes. Therefore, Internet-facing load balancers can route requests from clients over the Internet. The nodes of an internal load balancer have only private IP addresses. The DNS name of an internal load balancer is publicly resolvable to the private IP addresses of the nodes. Therefore, internal load balancers can only route requests from clients with access to the VPC for the load balancer.

If your application has multiple tiers, for example web servers that must be connected to the Internet and database servers that are only connected to the web servers, you can design an architecture that uses both internal and Internet-facing load balancers. Create an Internet-facing load balancer and register the web servers with it. Create an internal load balancer and register the database servers with it. The web servers receive requests from the Internet-facing load balancer and send requests for the database servers to the internal load balancer. The database servers receive requests from the internal load balancer.



Public DNS Name for Your Load Balancer

When an internal load balancer is created, it receives a public DNS name with the following form:

```
internal-name-123456789.region.elb.amazonaws.com
```

The DNS servers resolve the DNS name of your load balancer to the private IP addresses of the load balancer nodes for your internal load balancer. Each load balancer node is connected to the private IP addresses of the back-end instances using elastic network interfaces. If cross-zone load balancing is enabled, each node is connected to each back-end instance, regardless of Availability Zone. Otherwise, each node is connected only to the instances that are in its Availability Zone.

Cross-Zonal Load Balancer

With *cross-zone load balancing*, each load balancer node for your Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. If cross-zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only. Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances. However, AWS still recommend that you maintain approximately equivalent numbers of instances in each enabled Availability Zone for higher fault tolerance.

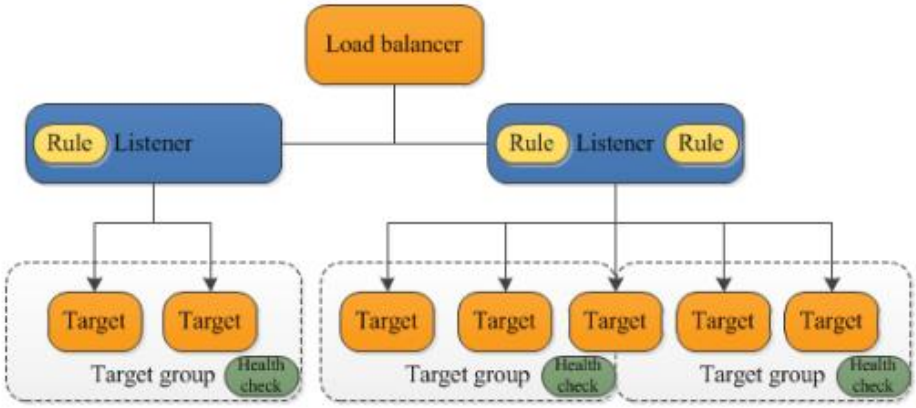
Application Load balancer (https/http)

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, and forwards requests to one or more target groups, based on the rules that you define. Each rule specifies a target group, condition, and priority. When the condition is met, the traffic is forwarded to the target group. You must define a default rule for each listener, and you can add rules that specify different target groups based on the content of the request (also known as content-based routing).

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.



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.

Network Load Balancer Overview

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (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.

When you enable an Availability Zone for the load balancer, Elastic Load Balancing creates a load balancer node in the Availability Zone. By default, each load balancer node distributes traffic across the registered targets in its Availability Zone only. If you enable cross-zone load balancing, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones.

Choosing the right Load Balancer

AWS launched Elastic Load Balancing (ELB) for AWS in the spring of 2009. Elastic Load Balancing has become a key architectural component for many AWS-powered applications. In conjunction with Auto Scaling, Elastic Load Balancing greatly simplifies the task of building applications that scale up and down while maintaining high availability.

On the Level

Per the well-known OSI model, load balancers generally run at Layer 4 (transport) or Layer 7 (application).

A Layer 4 load balancer works at the network protocol level and does not look inside of the actual network packets, remaining unaware of the specifics of HTTP and HTTPS. In other words, it balances the load without necessarily knowing a whole lot about it.

A Layer 7 load balancer is more sophisticated and more powerful. It inspects packets, has access to HTTP and HTTPS headers, and (armed with more information) can do a more intelligent job of spreading the load out to the target.

Application Load Balancing for AWS

In 2016, AWS launched a new Application Load Balancer option for ELB. This option runs at Layer 7 and supports a number of advanced features. The original option (now called a Classic Load Balancer) is still available to you and continues to offer Layer 4 and Layer 7 functionality.

Application Load Balancers support content-based routing, and supports applications that run in containers. They support a pair of industry-standard protocols (WebSocket and HTTP/2) and also provide additional visibility into the health of the target instances and containers. Web sites and mobile apps, running in containers or on EC2 instances, will benefit from the use of Application Load Balancers.

Features

- Content based routing
- Support for container based applications
- Better metrics
- Support for additional Protocols and Workloads

The Application Load Balancer supports two additional protocols: WebSocket and HTTP/2.

Elastic Load Balancing (ELB) has been an important part of AWS since 2009, when it was launched as part of a three-pack that also included Auto Scaling and Amazon CloudWatch. Since that time AWS have added many features, and also introduced the Application Load Balancer. Designed to support application-level, content-based routing to applications that run in containers, Application Load Balancers pair well with microservices, streaming, and real-time workloads.

Over the years, AWS customers have used ELB to support web sites and applications that run at almost any scale — from simple sites running on a T2 instance or two, all the way up to complex applications that run on large fleets of higher-end instances and handle massive amounts of traffic. Behind the scenes, ELB monitors traffic and automatically scales to meet demand. This process, which includes a generous buffer of headroom, has become quicker and more responsive over the years and works well even for our customers who use ELB to support live broadcasts, “flash” sales, and holidays. However, in some situations such as instantaneous fail-over between regions, or extremely spiky workloads, AWS have worked with our customers to pre-provision ELBs in anticipation of a traffic surge.

Network Load Balancer for AWS

In 2017, AWS introduced a new Network Load Balancer (NLB). It is designed to handle tens of millions of requests per second while maintaining high throughput at ultra low latency, with no effort on your part. The Network Load Balancer is API-compatible with the Application Load Balancer, including full programmatic control of Target Groups and Targets. Here are some of the most important features:

- Static IP Addresses – Each Network Load Balancer provides a single IP address for each Availability Zone in its purview. If you have targets in us-west-2a and other targets in us-west-2c, NLB will create and manage two IP addresses (one per AZ); connections to that IP address will spread traffic across the instances in all the VPC subnets in the AZ. You can also specify an existing Elastic IP for each AZ for even greater control. With full control over your IP addresses, Network Load Balancer can be used in situations where IP addresses need to be hard-coded into DNS records, customer firewall rules, and so forth.
- Zonality – The IP-per-AZ feature reduces latency with improved performance, improves availability through isolation and fault tolerance and makes the use of Network Load Balancers transparent to your client applications.
- Source Address Preservation – With Network Load Balancer, the original source IP address and source ports for the incoming connections remain unmodified.
- Long-running Connections – NLB handles connections with built-in fault tolerance, and can handle connections that are open for months or years, making them a great fit for IoT, gaming, and messaging applications.

DNS setup for ELB

Each Load Balancer receives a default Domain Name System (DNS) name. This DNS name includes the name of the AWS region in which the load balancer is created. For example, if you create a load balancer named my-loadbalancer in the US West (Oregon) region, your load balancer receives a DNS name such as my-loadbalancer-1234567890.us-west-2.elb.amazonaws.com. To access the website on your instances, you paste this DNS name into the address field of a web browser. However, this DNS name is not easy for your customers to remember and use.

If you'd prefer to use a friendly DNS name for your load balancer, such as www.example.com, instead of the default DNS name, you can create a custom domain name and associate it with the DNS name for your load balancer. When a client makes a request using this custom domain name, the DNS server resolves it to the DNS name for your load balancer.