



Elastic Load Balancing (ELB) Load Balancers and Listeners

At the core of the lesson

You will learn how to do the following:

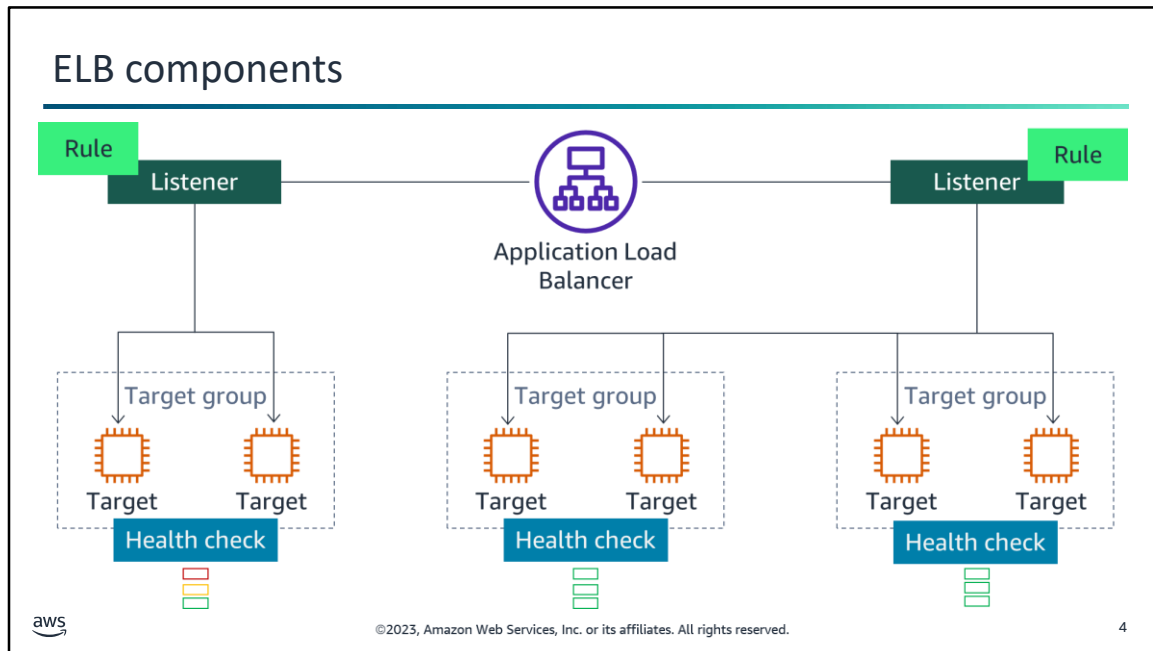
- Review the components of an architecture that uses a load balancer.
- Define the purpose of listeners and target groups.
- Describe how to create and configure a load balancer by using the AWS Command Line Interface (AWS CLI).



Recall the following information from the ELB presentation:

A *load balancer* works as the single point of contact for clients and serves as a traffic flag in front of your servers. It distributes the incoming application traffic across multiple targets, such as Amazon Elastic Compute Cloud (Amazon EC2) instances. The load balancer will maximize speed by monitoring capacity performance and the health status of targets that are located in multiple Availability Zones.

Before you can use a chosen load balancer and benefit from its features, you must add listeners and register your targets (or target groups).



To describe the components of Elastic Load Balancing (ELB), this example shows an Application Load Balancer for a website. The load balancer serves as the single point of contact for all clients. It distributes incoming application traffic across multiple targets.

Load balancers can have more than one listener. This example shows two listeners:

- Each listener checks for connection requests from clients, by using the protocol and port that were configured.
- The listener forwards requests to one or more target groups, based on the defined rules.

Rules are attached to each listener, and 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.
- This configuration is also known as content-based routing.

Each target group routes requests to one or more registered targets, such as EC2 instances, by using the protocol and port number that you specify:

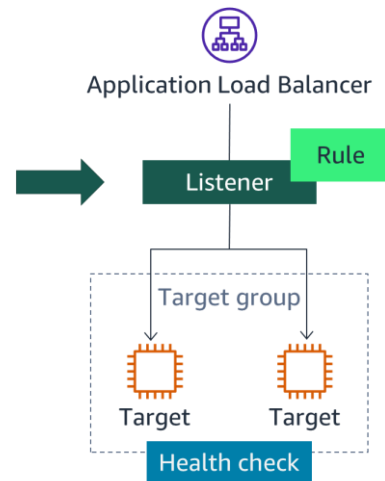
- You can register a target with multiple target groups.
- You can configure health checks for each target group.

Health checks, which are shown as attached to each target group, are performed on all targets that are registered to a target group that is specified in a listener rule for your load balancer.

Notice that each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. As the diagram implies, you can register a target with multiple target groups.

Listeners

- A listener is a process that defines the port and protocol that the load balancer listens on.
- Each load balancer needs at least one listener to accept traffic.
- Up to 50 listeners can be created on a load balancer.
- Routing rules are defined on listeners.

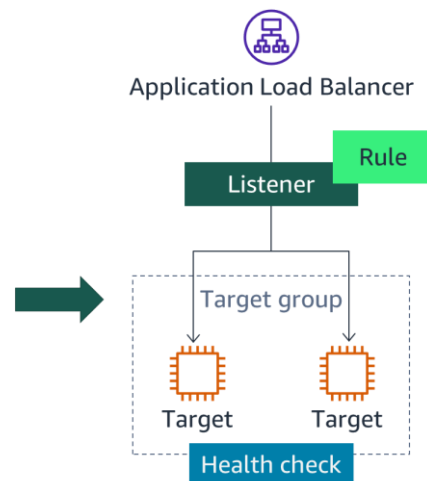


Before you can use an Application Load Balancer, you must add one or more listeners. A *listener* is a process that checks for connection requests from a client to an instance by using the protocol and port that you specify.

The rules that you define for the listener also determine how the load balancer routes traffic from connecting clients to one or more targets or target groups.

Target groups

- A target group contains registered targets that provide support to resources such as the following:
 - Amazon Elastic Compute Cloud (Amazon EC2) instances
 - Amazon Elastic Container Service (Amazon ECS) container instances
- A single target can have multiple target group registrations.



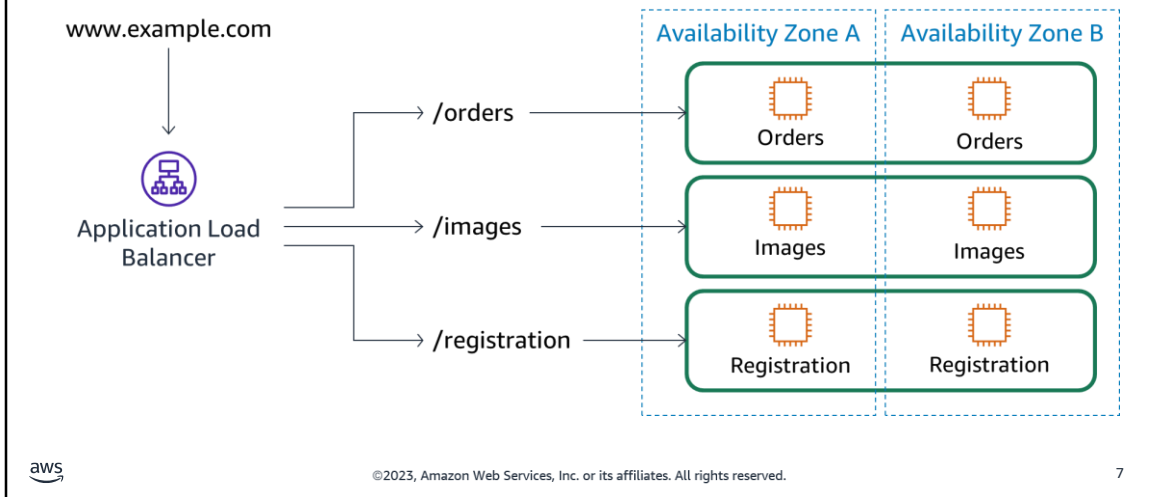
As mentioned previously, a load balancer serves as the single point of contact for clients. When you create each listener rule, you specify a *target group* and conditions. Targets are organized into target groups. Each target group is used to route requests to one or more registered targets, such as EC2 instances or Amazon Elastic Container Service (Amazon ECS) container instances.

The load balancer can distribute incoming application traffic across multiple targets.

You can also configure health checks for each target group. When the rule for your load balancer specifies a target group, health checks are performed on all targets that are registered to that target group.

For more information, see “What is an Application Load Balancer?” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>.

Application Load Balancer example



This diagram shows a single Application Load Balancer that is configured with three rules. The design illustrates that traffic is routed based on the URL that the client requests. Clients that request the `www.example.com/orders` page are routed to one set of EC2 instances. Clients that request the `/images` page are routed to other EC2 instances. Requests to the registration area are similarly routed to a different set of instances.



Create a load balancer

Methods to create a load balancer



1. AWS Management Console



2. AWS Command Line Interface
(AWS CLI)



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9

After you select the type of load balancer that best suits your needs, you can choose to create it with the AWS Management Console or the AWS Command Line Interface (AWS CLI).

At the following links, you can find AWS documentation with instructions on how to create each type of load balancer by using either method:

Application Load Balancer

To create an Application Load Balancer by using the console, see “Getting Started with Application Load Balancers” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/application-load-balancer-getting-started.html>.

To create an Application Load Balancer by using the AWS CLI, see “Tutorial: Create an Application Load Balancer Using the AWS CLI” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/tutorial-application-load-balancer-cli.html>.

Network Load Balancer

To create a Network Load Balancer by using the console, see “Getting Started with Network Load Balancers” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/network/network-load-balancer-getting-started.html>.

To create a Network Load Balancer by using the AWS CLI, see “Tutorial: Create a Network Load Balancer Using the AWS CLI” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/network/network-load-balancer-cli.html>.

Gateway Load Balancer

To create a Gateway Load Balancer by using the console, see “Getting Started with Gateway Load Balancers” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/getting-started.html>.

To create a Gateway Load Balancer by using the AWS CLI, see “Getting Started with Gateway Load Balancers

Using the AWS CLI” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/getting-started-cli.html>.

Classic Load Balancer

AWS retired the EC2-Classic network. Accounts created after December 4, 2013, are virtual private cloud (VPC) only and cannot create this type of load balancer.

Process to create a load balancer

1. Create a load balancer by using the AWS CLI.
2. Create a target group for the load balancer.
3. Register EC2 instances to the target group.
4. Create a listener for the load balancer.
5. Verify the health of the registered targets.



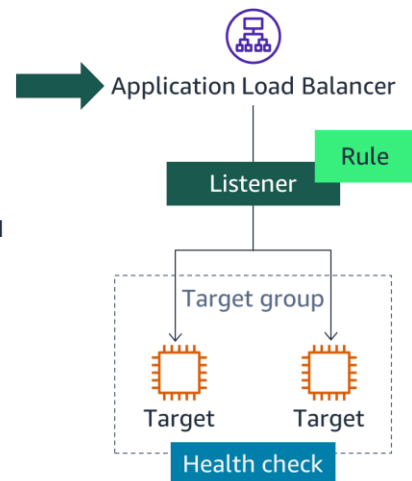
There are five steps to create a load balancer. Over the next several slides, you will move through each step of creating an Application Load Balancer by using the AWS CLI.

For demonstrations of common load balancer configurations, see AWS Elastic Load Balancer Demos at <https://exampleloadbalancer.com>.

****For Accessibility:** Diagram of process to create a load balancer. The steps are as follows: 1. Create a load balancer by using the AWS CLI. 2. Create a target group for the load balancer. 3. Register EC2 instances to the target group. 4. Create a listener for the load balancer. 5. Verify the health of the registered targets. **End Description.**

Create an Application Load Balancer

1. Use the `create-load-balancer` command to create a load balancer. You must specify two subnets that are not from the same Availability Zone.



First, complete the following steps:

1. To verify that you are running a version of the AWS CLI that supports Application Load Balancers, run the following command: `aws elbv2 help`

If an error message indicates that `elbv2` is not a valid choice, update your version of the AWS CLI. For more information, see “Getting started with the AWS CLI” in the *AWS Command Line Interface User Guide* at <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html>.

2. Launch your EC2 instances in a virtual private cloud (VPC). Ensure that the security groups for these instances allow access on the listener port and the health check port. For more information, see “Target Security Groups” at <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/target-group-register-targets.html#target-security-groups>.

To complete this step, run the `create-load-balancer` command to create an Application Load Balancer.

For more information about the command, see “`create-load-balancer`” in the *AWS CLI Command Reference* at <https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-load-balancer.html>.

Example: Create an Application Load Balancer

Command

```
aws elbv2 create-load-balancer \
--name my-load-balancer \
--subnets subnet-12345678 subnet-23456789 \
--security-groups sg-12345678
```

Result

```
{
  "LoadBalancers": [{
    "LoadBalancerArn":
    "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/1234567890123456",
    "DNSName": "my-load-balancer-1234567890123456.us-east-1.elb.amazonaws.com",
    "CanonicalHostedZoneId": "Z35SXDOTRQ7X7K",
    "CreatedTime": "2023-01-10T16:33:59.670Z",
    "LoadBalancerName": "my-load-balancer", ...
  ]
}
```



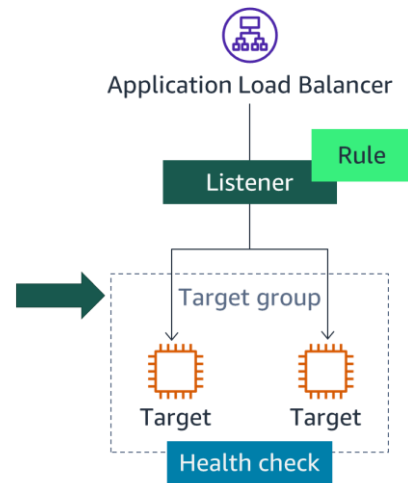
The result from issuing the command is similar to the example on the slide.

Note that the output includes the Amazon Resource Name (ARN) of the newly created load balancer, similar to the following format:

arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/1234567890123456

Create a target group

2. Use the `create-target-group` command to create a target group. Specify the same virtual private cloud (VPC) that you used for your EC2 instances.



To complete the second step, run the `create-target-group` command to create a target group.

For more information about the command, see “`create-target-group`” in the AWS CLI Command Reference at <https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-target-group.html>.

You can create IPv4 and IPv6 target groups to associate with load balancers. The target group's IP address type determines the IP version that the load balancer will use to communicate with and check the health of your backend targets. IPv4 target groups support IP and instance type targets. IPv6 targets only support IP targets.

Example: Create a target group

Command

```
aws elbv2 create-target-group \  
--name my-targets \  
--protocol HTTP \  
--port 80 \  
--vpc-id vpc-1234567890123 \  
--ip-address-type ipv4
```

Result

```
{  
  "TargetGroups": [{  
    "TargetGroupArn":  
    "arn:aws:elasticloadbalancing:us-east-  
1:123456789012:targetgroup/my-  
targets/1234567890123456",  
    "TargetGroupName": "my-targets",  
    "Protocol": "HTTP",  
    "Port": 80,  
    "VpcId": "vpc-1234567890123",  
    "HealthCheckProtocol": "HTTP",  
    "HealthCheckPort": "traffic-port",  
    "HealthCheckIntervalSeconds": 30,  
    "HealthCheckTimeoutSeconds": 5,  
    "HealthyThresholdCount": 5,  
    "UnhealthyThresholdCount": 2,  
    "HealthCheckPath": "/",  
    "Matcher": {  
      "HttpCode": "200"...
```



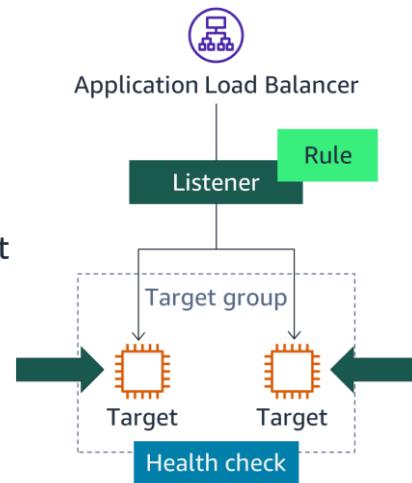
The result from issuing the command is similar to the example on the slide.

The output includes the ARN of the target group, similar to the following format:

arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/my-targets/1234567890123456

Register EC2 instances

3. Use the `register-targets` command to register your instances with your target group.

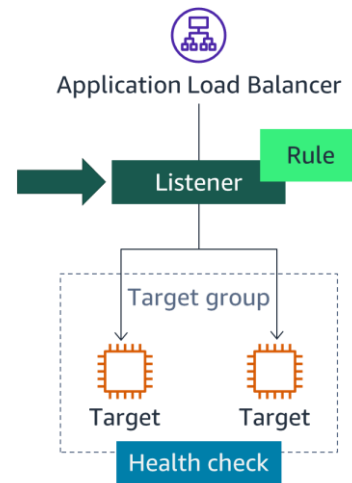


To complete the third step, run the `register-targets` command to register your instances with your target group. For the `targetgroup-arn` parameter, enter the ARN of the target group that you created in the previous step. Note that this command does not have output.

For more information about the command, see “`register-targets`” in the *AWS CLI Command Reference* at <https://docs.aws.amazon.com/cli/latest/reference/elbv2/register-targets.html>.

Create a listener

4. Use the `create-listener` command to create a listener for your load balancer.



To complete the fourth step, use the `create-listener` command to create a listener for your load balancer with a default rule that forwards requests to your target group.

For more information about the command, see “`create-listener`” in the *AWS CLI Command Reference* at <https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-listener.html>.

Example: Create a listener

Command

```
aws elbv2 create-listener \
--load-balancer-arn loadbalancer-arn \
--protocol HTTP \
--port 80 \
--default-actions
Type=forward,TargetGroupArn=targetgroup-arn
```

Result

```
{
  "Listeners": [
    {
      "ListenerArn":
"arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/1234567890123456/14f6f490e5f8bdbf",
      "LoadBalancerArn": "LoadBalancerArn":
"arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/1234567890123456",
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "Type": "forward",
          "TargetGroupArn":
"arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/my-targets/9546a5cf349b582c "
        }
      ]
    }
  ]
}
```

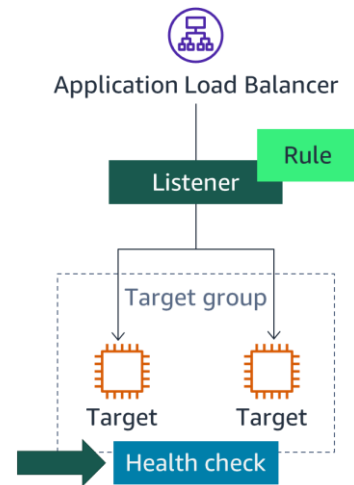


The result from issuing the command is similar to the example on the slide.

The output includes the ARN of the listener, similar to the following format: `arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/1234567890123456/14f6f490e5f8bdbf`

Verify the health of the targets

- Optionally, verify the health of the registered targets for your target group by using the `describe-target-health` command.



This final step is optional. To verify the health of the target instances, run the `describe-target-health` command.

For more information about the command, see “`describe-target-health`” in the *AWS CLI Command Reference* at <https://docs.aws.amazon.com/cli/latest/reference/elbv2/describe-target-health.html>.

Example: Verify the health of the targets

Command

```
aws elbv2 describe-target-health \  
--target-group-arn targetgroup-arn
```

Result

```
{  
  "TargetHealthDescriptions": [  
    {  
      "HealthCheckPort": "80",  
      "Target": {  
        "Id": "i-12345678",  
        "Port": 80  
      },  
      "TargetHealth": {  
        "State": "healthy"  
      }  
    },  
    {  
      "HealthCheckPort": "80",  
      "Target": {  
        "Id": "i-23456789",  
        "Port": 80  
      },  
      "TargetHealth": {  
        "State": "healthy" ...  
      }  
    }  
  ]  
}
```



The result from issuing the command is similar to the example on the slide.

Note that the status of each instance in this example is *healthy*.

Checkpoint questions

1. A website is running slowly. It has many URLs, including `example.com/login`, `example.com/products`, and `example.com/orders`. The website is currently hosted on an EC2 instance. How could you use a load balancer to improve the performance of the website?
2. Do health checks apply to the load balancer or to the target servers?



The answers to the questions are as follows:

1. A website is running slowly. It has many URLs, including `example.com/login`, `example.com/products`, and `example.com/orders`. The website is currently hosted on an EC2 instance. How could you use a load balancer to improve the performance of the website?
Each URL could be handled by a different group of EC2 servers. Rules on an Application Load Balancer could then be used to route traffic to the different groups of EC2 servers. If a specific URL needs to scale to meet demand, only the group of servers for that URL would need to scale instead of the entire footprint. This could reduce costs.
2. Do health checks apply to the load balancer or to the target servers?
Health checks are used to monitor target servers that load balancers route traffic to. If a health check fails, the load balancer can deregister the unhealthy server.

Key ideas



- Each listener checks for connection requests from clients and then forwards requests to one or more target groups, based on the defined rules.
- Each rule specifies a target group, condition, and priority.



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.