



# Introduction to Elastic Load Balancing

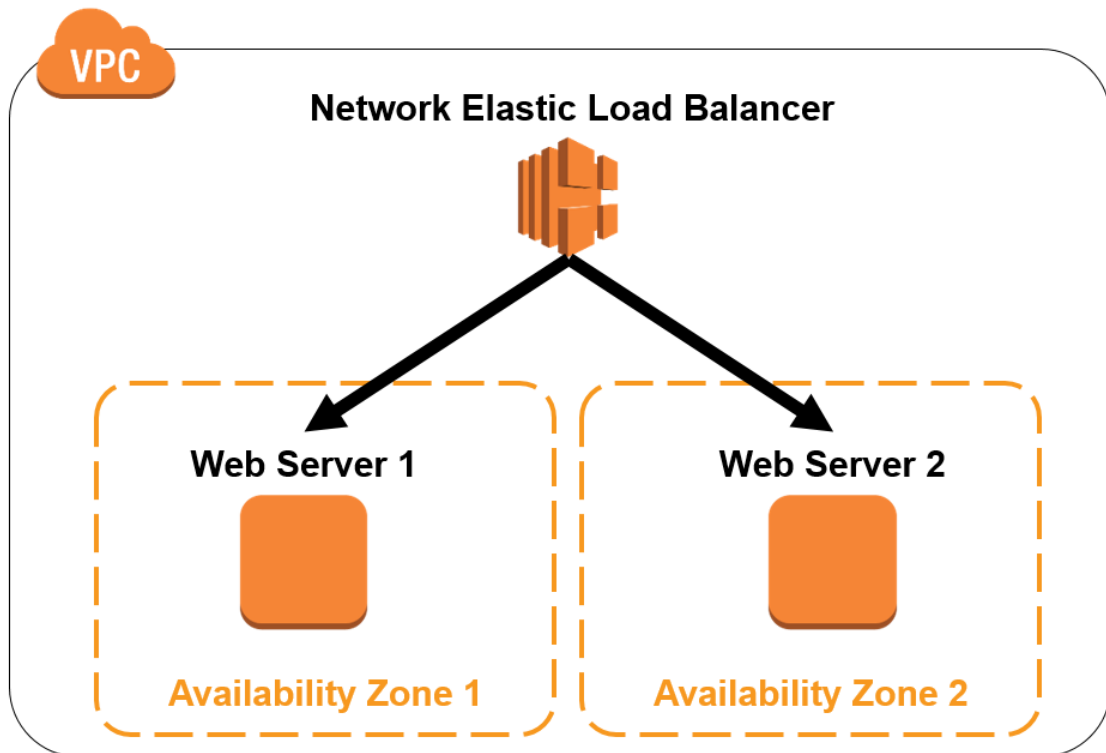
## **SPL-68 Version 2.0**

© 2018 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.

## **Overview**

This lab provides you with an introduction to Elastic Load Balancer. It covers creating and testing a Network Elastic Load Balancer. When the lab is launched you are provided with two web servers in two different Availability Zones. In the lab you will create a Network Elastic Load Balancer and use the two Web Servers as targets. You will then test the functionality of the load balancer in different scenarios.

Lab VPC



## Lab Description

In this lab you will:

- Test connectivity to two web servers that reside in two different Availability Zones
- Create a Network Load Balancer and use the two web servers as Elastic Load Balancer targets
- Test the default functionality of your load balancer
- Enable Cross-Zone load balancing and test how your load balancer behaves
- Test the behavior of your load balancer during a failure of one of your web servers
- Test the behavior of your load balancer after your web server has recovered from the failure

### Amazon Elastic Load Balancer

An Amazon Elastic Load Balancer (Amazon ELB) is a service that automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming

application traffic. Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored.

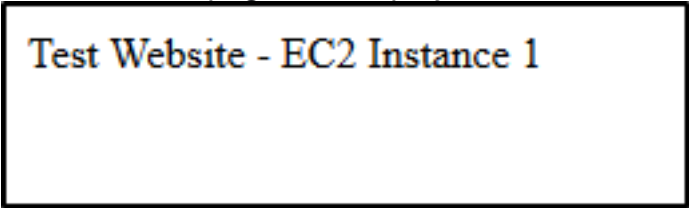
Customers can enable Elastic Load Balancing within a single Availability Zone or across multiple zones for even more consistent application performance. Elastic Load Balancing can also be used in an Amazon Virtual Private Cloud (VPC) to distribute traffic between application tiers.

## Task 1: Test Access to Your Web Servers

In this task, you will connect to each of your web servers over HTTP. This will prove that your web servers are serving a web page. These EC2 instances were created in the CloudFormation template when you started the lab.

- To the left of these instructions, copy the value of **WebServer01** to your clipboard, then:
  - Paste the value into a new browser tab
  - Press **Enter**

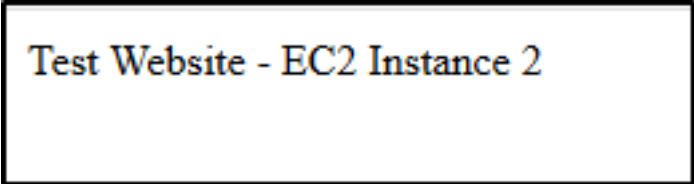
You will see a page that displays *Test Website - EC2 Instance 1*



Test Website - EC2 Instance 1

- To the left of these instructions, copy the value of **WebServer02** to your clipboard, then:
  - Paste the value into a new browser tab
  - Press **Enter**

You will see a page that displays *Test Website - EC2 Instance 2*



Test Website - EC2 Instance 2

## Task 2: Create an Elastic Load Balancer

- In the **AWS Management Console**, on the **Services** menu, click **EC2**.
- In the left navigation pane, click **Load Balancers**.
- Click **Create Load Balancer**

A load balancer serves as the single point of contact for clients. The load balancer distributes incoming traffic across multiple targets, such as Amazon EC2 instances. This increases the availability of your application.

- On the **Select load balancer type**, below **Network Load Balancer**, click **Create**

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.

- On **Step 1**, configure:
  - **Name:**
  - **VPC:** *Lab VPC*
  - Select both Availability Zones
  - Scroll to the bottom of the screen, then click **Next: Configure Routing**
- On **Step 2**, configure:
  - **Name:**
  - Expand **Advanced health check settings**

You register targets for your Network Load Balancer with a target group. By default, the load balancer sends requests to registered targets using the port and protocol that you specified for the target group.

- In **Advanced health check settings**, configure:
  - **Healthy threshold:**
  - **Interval:** *10 seconds*

The load balancer sends a health check request to each registered target every `HealthCheckIntervalSeconds` seconds, using the specified port, protocol, and ping path. It waits for the target to respond within the response timeout period. If the health checks exceed the threshold for consecutive failed responses, the load balancer takes the target out of service. When the health checks exceed the threshold for consecutive successful responses, the load balancer puts the target back in service.

Since you've set the **Healthy threshold** to 2 and the interval to *10 seconds*, it will take at least *20 seconds* for your instance to report a *healthy* status. By default, each load balancer node routes requests only to the healthy targets in its Availability Zone.

- At the bottom of the screen, click **Next: Register Targets**
- In the **Instances** section, select both EC2 instances.
- Click **Add to registered**

Make sure you click **Add to registered**

- Click **Next: Review**
- On **Step 4**

- Review your load balancer configuration
- Click **Create**
- Click **Close**
- In the **Description** tab of your load balancer, copy the DNS name to your text editor.

Your DNS name will look similar to *myELB-4e009e86b4f704cc.elb.us-west-2.amazonaws.com*

- In the left navigation pane, click **Target Groups**.

Your *MyELBTargeGroup* should be selected.

- Click the **Targets** tab.
- Wait for your EC2 instances to have a status of *healthy*.

You can click **refresh** periodically to refresh the status.

Here are the possible values for the health status of a registered target:

- **initial:** The load balancer is in the process of registering the target or performing the initial health checks on the target
- **healthy:** The target is healthy
- **unhealthy:** The target did not respond to a health check or failed the health check.
- **unused:** The target is not registered with a target group, the target group is not used in a listener rule for the load balancer, or the target is in an Availability Zone that is not enabled for the load balancer.
- **draining:** The target is deregistering and connection draining is in process

## Task 3: Test Your Load Balancer

### Test the Default Functionality

- In a new browser tab, paste the DNS name and then press **Enter**. You will see the HTML page for one of your EC2 instances.

- Refresh the page a few times.

Notice that the same EC2 instance page is displayed.

With Network Load Balancers, cross-zone load balancing is disabled by default. After you create a Network Load Balancer, you can enable or disable cross-zone load balancing at any time.

With Cross-Zone load balancing, nodes for your load balancer distribute requests from clients to registered targets. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones. When cross-zone load balancing is disabled, each load balancer node distributes traffic across the registered targets in its Availability Zone only.

## Test Cross-Zone Load Balancing Functionality

- In the **AWS Management Console**, in the left navigation pane, click **Load Balancers**.
- In the **Description** tab, scroll down to the **Attributes** section.
- Click **Edit attributes** then configure:
  - Select **Cross-Zone Load Balancing** *Enable*
  - Click **Save**
- Wait a minute or two.
- Return to the browser tab that you used to access your load balancer.
- Refresh the page a few times.

You should see that your network load balancer now directs you to both of your EC2 instances.

## Disable Cross-Zone Load Balancing Functionality

- In the **AWS Management Console**, in the left navigation pane, click **Load Balancers**.
- In the **Description** tab, scroll down to the **Attributes** section.
- Click **Edit attributes** then configure:
  - De-select **Cross-Zone Load Balancing** *Enable*
  - Click **Save**
- Wait a minute or two.
- Return to the browser tab that you used to access your load balancer.
- Refresh the page a few times.

You should see that your network load balancer is now only serving pages from one of your instances.

# Task 4: Test Your Load Balancer During a Failure

## Test Load Balancer During a Failure

- In the **AWS Management Console**, in the left navigation pane, click **Instances**.
- Select the EC2 instance that is currently serving you the web page.
- In the **Actions** menu, click **Instance State Stop**.
- Click **Yes, Stop**
- Wait a minute or two.
- Return to the browser tab that you used to access your load balancer.
- Refresh the page a few times.

You will see that that your load balancer now displays the web page for your other instance.

## Test Load Balancer After Recovering From a Failure

- In the **AWS Management Console**, in the left navigation pane, click **Instances**.
- Select the instance that you stopped earlier.
- In the **Actions** menu, click **Instance State Start**.
- Click **Yes, Start**
- Wait for a minute or two for your instance to fully start.

You can click the **refresh** button to update the status.

- In the browser tab for your Load Balancer, refresh the page.

You will see that that your load balancer now displays the web page of the instance that it originally used.