

Introduction

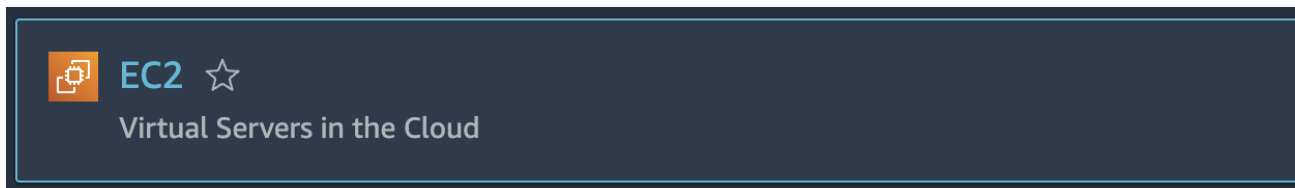
Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple Amazon EC2 instances. They enable you to achieve greater fault tolerance in your applications and seamlessly provide the correct amount of load balancing capacity needed in response to incoming application requests.

ELB detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored. Elastic Load Balancers can be enabled within a single Availability Zone or across multiple zones for greater consistent application performance.

There are [several ELB load balancers to choose from](#). The network load balancer is a network layer (layer-4) load balancer operating on TCP connections and UDP. It can scale to millions of requests per second and is a more modern alternative to the classic load balancer (also a layer-7 load balancer). With a network load balancer, backend targets are organized into *target groups* which the network load balancer distributes traffic across. You will create a network load balancer in this lab step.

Instructions

1. In the AWS Management Console search bar, enter *EC2*, and click the **EC2** result under **Services**:



2. In the left-hand menu, click **Load Balancers**:

Load Balancers

3. Click **Create Load Balancer**.

4. Take a moment to read the information for the load balancer types before clicking **Create** in the **Network Load Balancer** tile:

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

The **Create Network Load Balancer** form will open.

5. Set the following values leaving the others at their defaults:

- **Basic configuration:**
 - **Load balancer name:** *Web*
- **Network mapping:**
 - **Mappings:** Check all availability zones

Notice that the default listener is TCP port 80 which is used for serving HTTP traffic.

6. Look at the **Listeners and routing** section, and click on **Create target group** which opens a new browser tab:



Default action

Forward to *Select a target group* ▼

[Create target group](#) ↗

7. In the new target group browser tab that opened, set the following values leaving the others at their defaults:

- **Basic configuration:**
 - **Choose a target type:** Instances
 - **Target group name:** *Website*
 - **Protocol:** Select **TCP** from the drop-down menu
 - **Port:** Ensure **80** is set as the port value
- **Health checks:**
 - **Advanced health check settings:** (Click the triangle to expand the section)
 - **Interval: 10 seconds** (This will cause instances to reach a healthy state faster for this Lab, but may be too fast for certain applications)

The **target type** option allows you to specify **IP addresses** or a **Lambda function** in addition to **Instances**. Using an IP address gives you the ability to use a network load balancer with compute instances outside of AWS. You will use **Instances** within AWS for this lab.

Note: Ensure you set the protocol to TCP. Because network load balancers operate at layer 4 and aren't HTTP aware, if you set the protocol as HTTP you will be unable to use the target group with your network load balancer.

8. Scroll to the bottom and click **Next**.

On the **Register Targets** step, notice there are **No instances available**:

Available instances (0)

< 1 > ⚙

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Subnet ID
No Available instances						
0 selected						
Ports for the selected instances						
Ports for routing traffic to the selected instances (separate multiple ports with commas):						
<input type="text" value="80"/>						
<button>Include as pending below</button>						


The message is due to not creating an Auto Scaling group or launching any EC2 instances yet. That is not a problem. You will configure your Auto Scaling group to register its EC2 instances in the Network Load Balancer's target group.

9. At the bottom of the page, click **Create target group**.

10. Return to your load balancer configuration browser tab.

11. In the **Listener** section, click the refresh icon beside the **Default action** drop-down menu, and select the newly created target group from the drop-down:

Default action

Forward to	Website Target type: Instance	TCP ▲	
Create target	<input type="text" value="Q "/>		
	Website Target type: Instance	TCP	

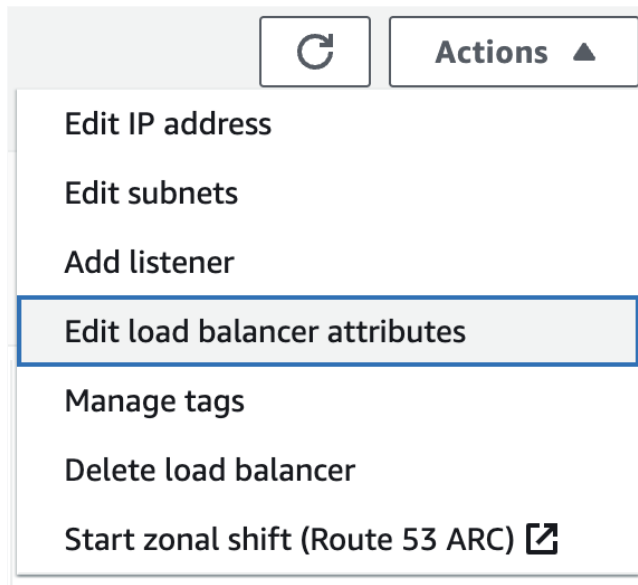
12. Scroll to the bottom and click on **Create load balancer**.

13. Wait for the **Successfully created load balancer** message to display before clicking **View load balancer**:

✔ **Successfully created load balancer: Web**

Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

14. In the load balancers table, ensure the **Web** load balancer is selected and click **Actions** > **Edit load balancer attributes**:



15. In the **Edit load balancer attributes** form, set the following value before clicking **Save changes**:

- **Cross-zone load balancing: Enable**

☒ **Cross-zone load balancing**

By default, each Network Load Balancer Elastic Network Interface (ENI) only distributes traffic across the registered targets in its Availability Zone. If you enable cross-zone load balancing, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones.

You must enable **Cross-zone load balancing** to achieve the highest level of availability. Without enabling this feature, clients could cache the DNS address of the load balancer node in one availability zone and that node would only distribute requests to instances within the availability zone. Cross-Zone Load Balancing allows every load balancer node to distribute requests across all availability zones, although for the Network Load Balancer there are data transfer charges when this feature is enabled. (There are no data charges for other types of load balancers)

16. In the left-hand menu, click **Target Groups**:



17. Click on the **Website** target group:



18. Click the **Attributes** tab:



19. On the right-hand side, click **Edit**:



20. Change the **Deregistration delay** to *30* seconds and click **Save changes**:

Deregistration delay
The time to wait for in-flight

30 seconds

0-3600

The deregistration delay specifies how long the load balancer should wait before removing an instance from the target group. The default value of 300 seconds gives connections to the instance five minutes to drain before they are forcefully closed. Depending on your application, you may be able to reduce to delay to remove instances more quickly. Thirty seconds is enough for this Lab.

Summary

In this lab step, you created a Network Load Balancer with a target group ready to service HTTP requests on port 80. This load balancer will be used as the front-end to a website. The website will run on EC2 instances that are created via an Auto Scaling group. This is a very common use case.

VALIDATION CHECKS

1 Checks

Check again 



Created Network Load Balancer

Check if the Network Load Balancer has been created

Elastic Load Balancing