# Creating an Application Load Balancer and Auto Scaling Group in AWS

Level: **Intermediate**

Amazon EC2 Amazon Web Services Elastic Load Balancing Amazon EC2 Auto Scaling

1h 29m 12s left

15 MIN

End Lab

Open Console

Validation

## Lab Credentials

**User Name**

Whiz_User_80425.64332336

**Password**

fb5d3310-99fd-4b39-b872-2a84056c8bd1

**Access Key**

AKIA4BHUHZ5XGVPAADOZ

**Secret Key**

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

## Lab Resources

No Lab Resources Found

## Support Documents

No Support Documents Found

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share

Lab Overview | Lab Steps | Lab Validation

Cloud Architect

Compute

# Lab Steps

## Task 1: Sign in to AWS Management Console

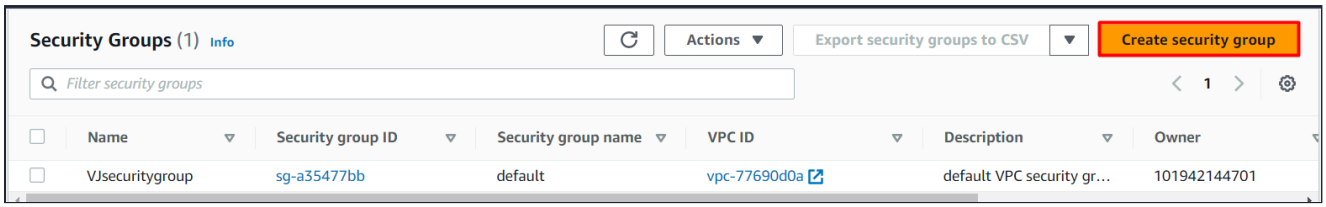
1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
    - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
    - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username**



## Task 2: Create a Security Group for the Load balancer

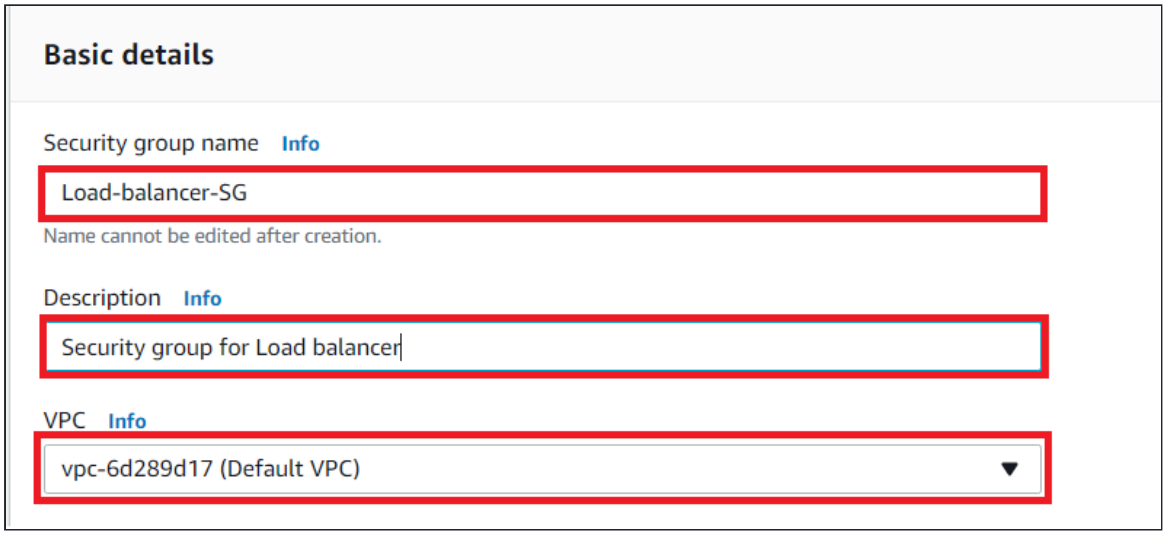
1. Make sure you are in the **N.Virginia** Region.
2. Navigate to **EC2** by clicking on the **Services** menu available under the **Compute** section.
3. On the left panel menu, select the **Security Groups** under the **Network & Security** section.

4. Click on the **Create security group** button.

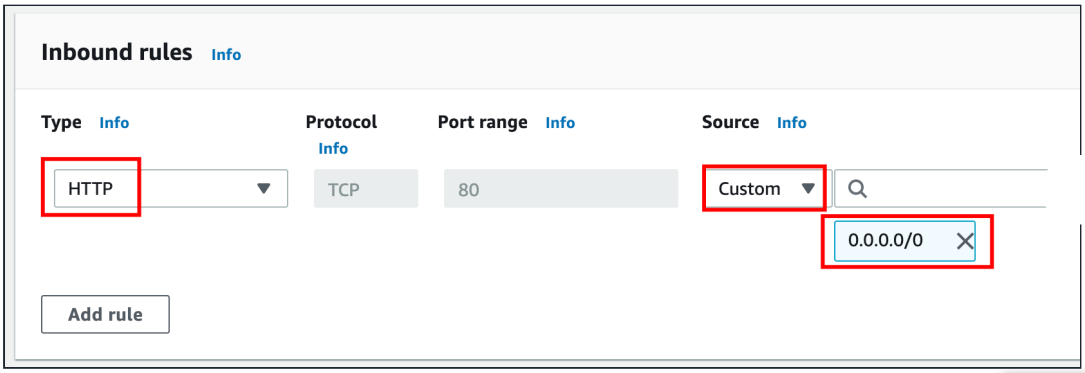



5. We are going to create a Security group for the Launch template with port 80 number enabled.

- Security group name: Enter ***Load-balancer-SG***
- Description: Enter ***Security group for Load balancer***
- VPC: Select **Default VPC**




- Click on the **Add rule** button under **Inbound rules.**
  - Type : Select **HTTP**
  - Source : Select **Custom**
  - In the textbox add **0.0.0.0/0**

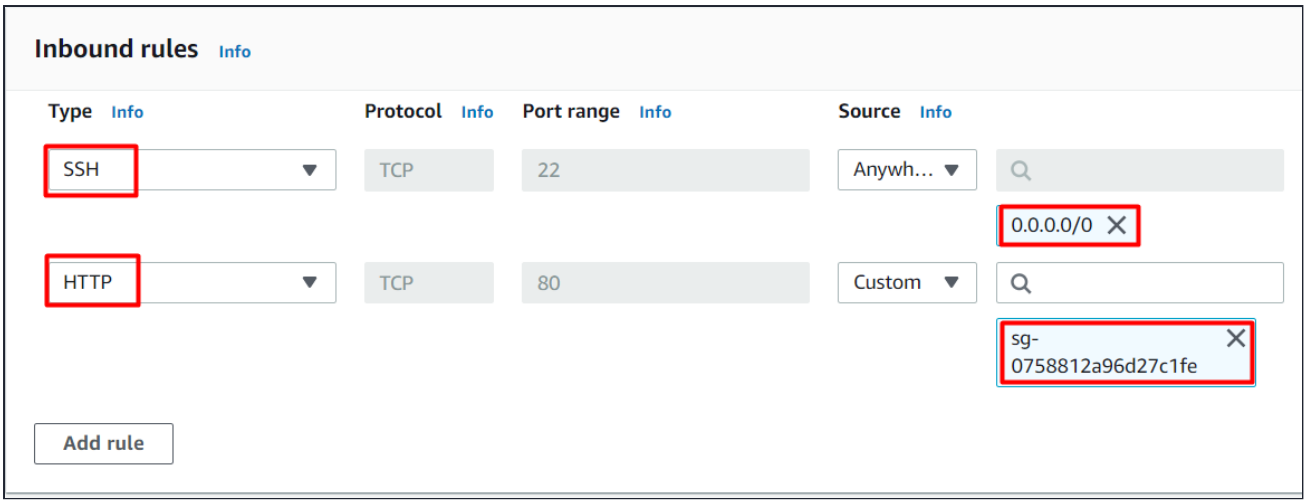



6. Leave everything as default and click on the **Create security group** button.

## Task 3: Create a Security Group for Launch template

1. Click on the **Create security group** button.
2. We are going to create a Security group for the Launch template with port 80 number enabled.
    - Security group name: Enter ***Launch-template-SG***
    - Description: Enter ***Security group for Launch template***
    - VPC: Select **Default VPC**


- Click on the **Add rule** button under **Inbound rules.**
    - Here we will add **SSH from the open internet** and **HTTP from the security group of a Load Balancer.**
    - Type : Select **SSH**
    - Source : Select **Custom**
    - In the textbox add **0.0.0.0/0**
    - Click on the **Add rule** button to add **HTTP**
    - Type : Select **HTTP**
    - Source : Select **Custom**
    - In the textbox type, **Load-balancer-SG**, select it.

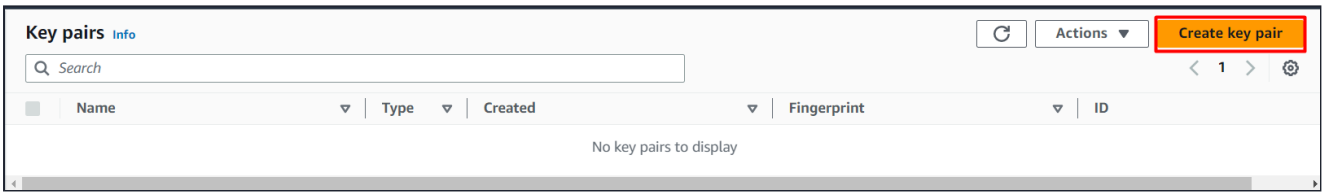



6. Leave everything as default and click on the **Create security group** button.

## Task 4: Create a Key Pair for the Launch template

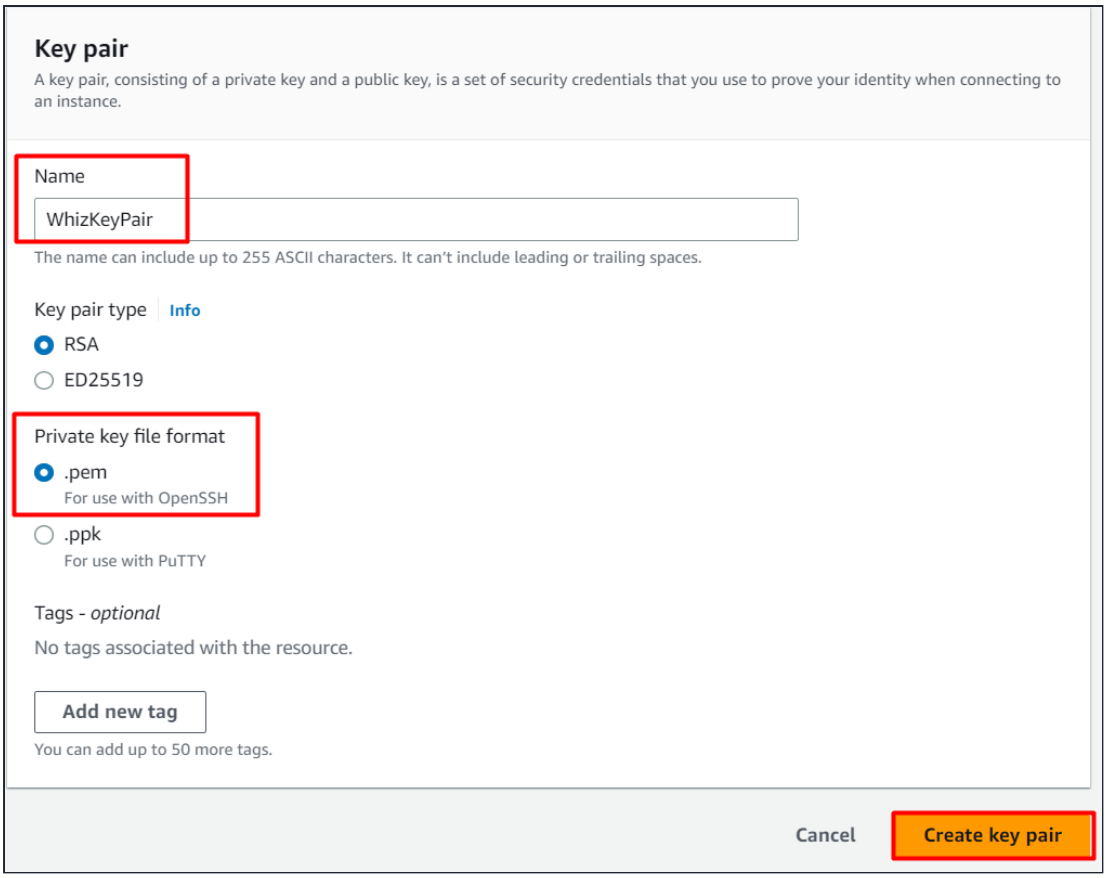
1. In the left navigation pane (scroll down) within **Network & Security**, click on the **Key Pairs.**

2. To create a new key pair, click on the **Create key pair** button.




3. Fill in the details below:

   - Name: Enter ***WhizKeyPair***
   - File format: **pem (Linux & Mac Users)** or **ppk (Windows users)**
   - Leave other options as default.
   - Click on the **Create key pair** button.

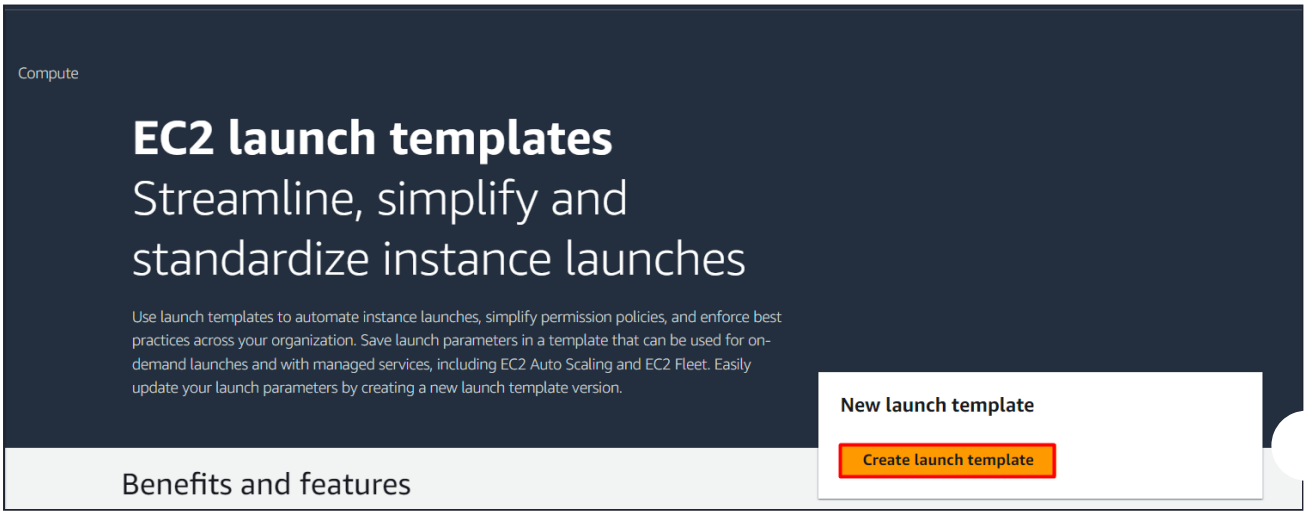



4. Key pair will be created.

## Task 5: Creating a Launch template

1. In the left navigation pane (scroll down) within **Instances**, click on the **Launch templates**

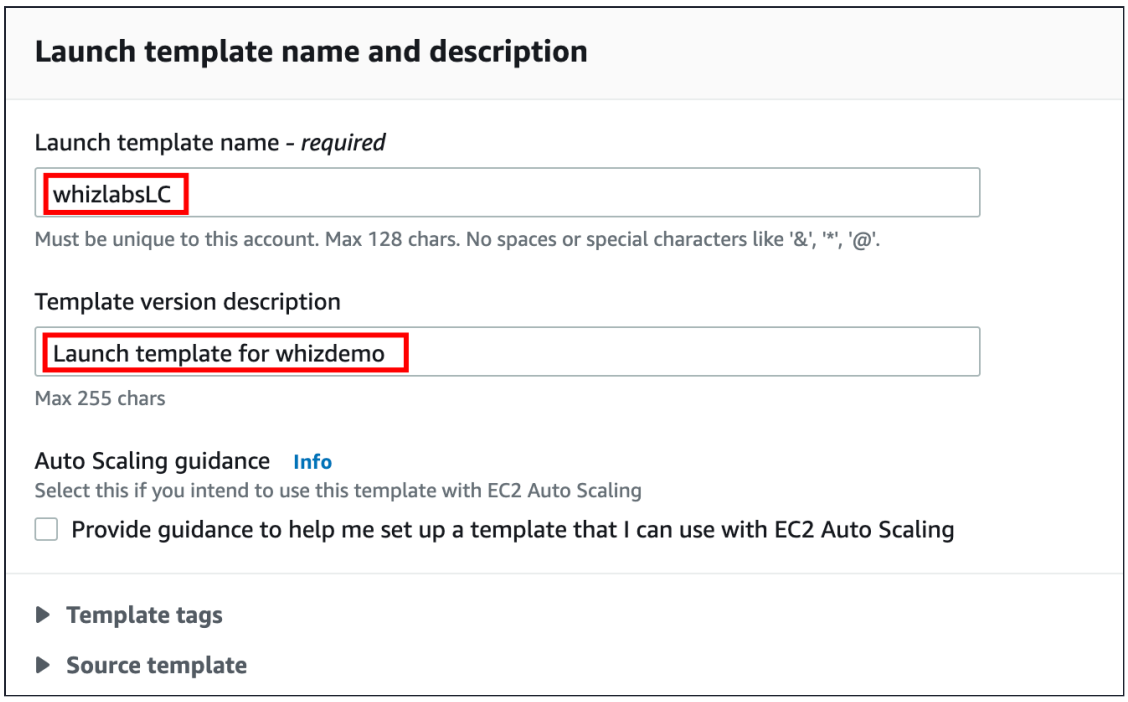
2. Click on the **Create launch template** button.




3. Under **Launch template name and description** section:

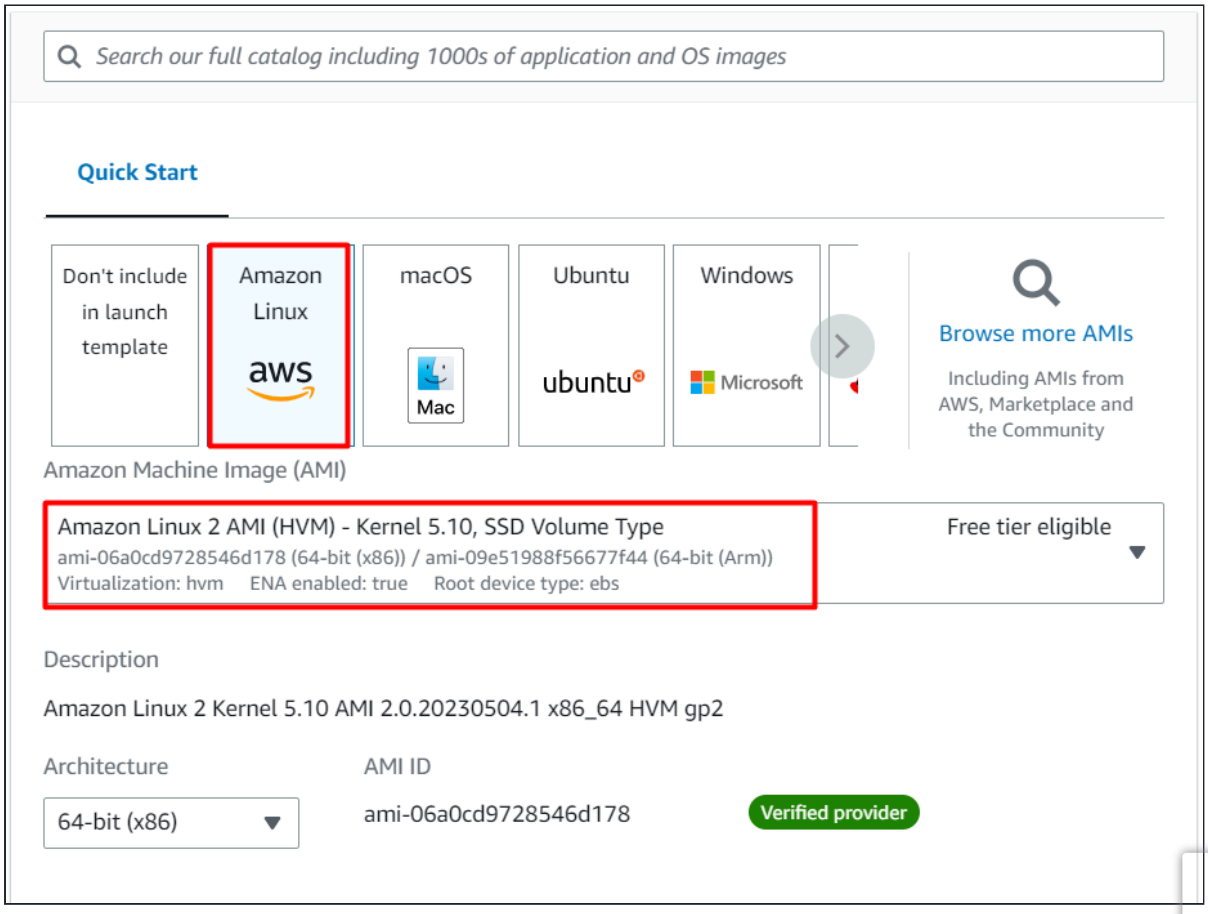
- Launch template name: Enter ***whizlabsLC***
- Template version description: Enter ***Launch template for whizdemo***

- **Leave other options as default.**

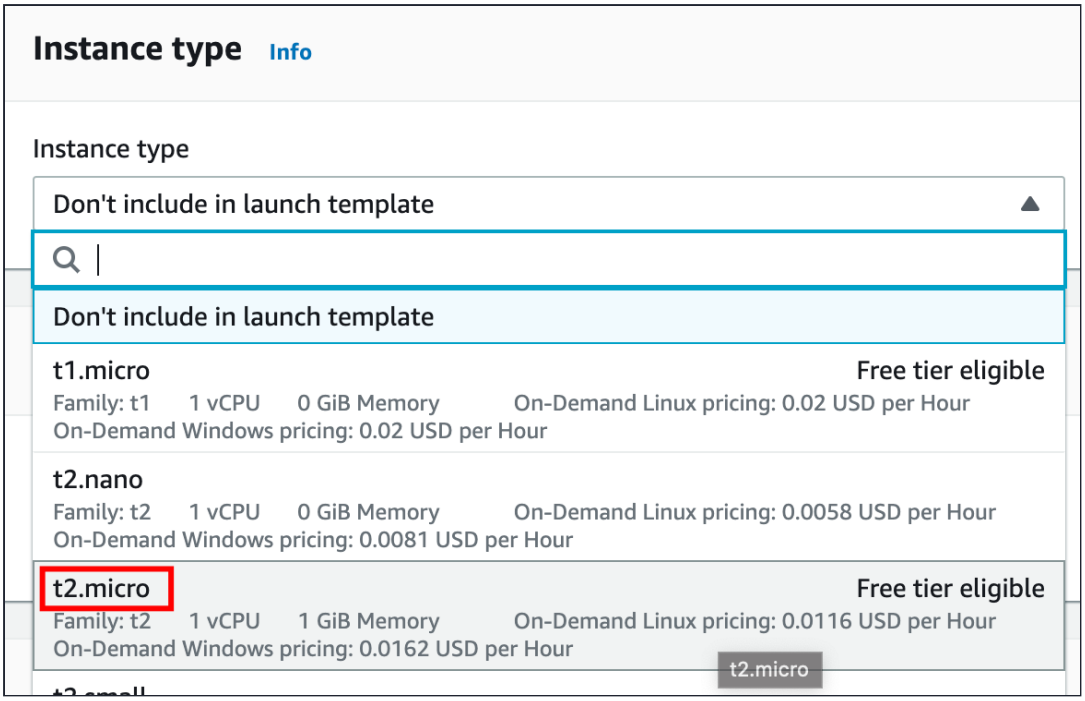



4. Under Launch template contents:

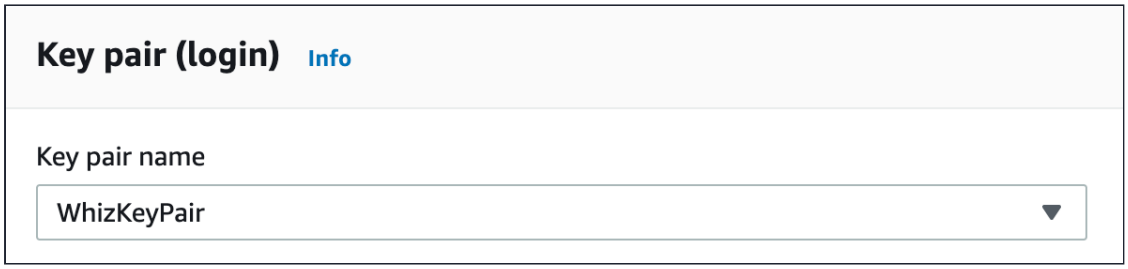
- Select **Amazon Linux** from the Quick Start
- Amazon machine image (AMI): Select **Amazon Linux 2 AMI (HVM), SSD Volume Type**



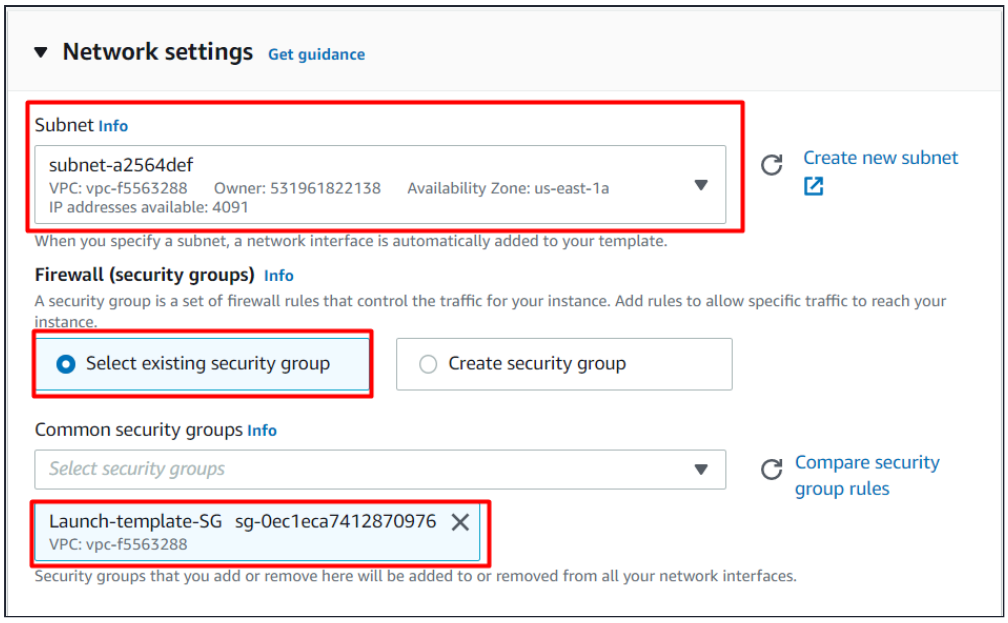
- Under Instance type:
    - Select **t2.micro** from the below list.




- Key pair (Login): Select **WhizKeyPair**




- Subnet: Choose **any subnet from the list.**
- Security groups: Select **Launch-template-SG** from the list




- Leave all other options as default.
- Expand the option of **Advanced details**, Go to the **User data**, and paste the below script.

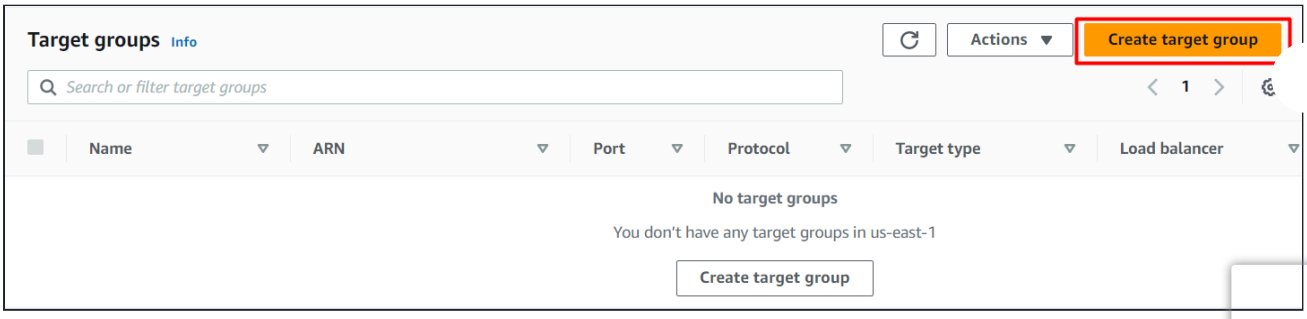
```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Hello World from $(hostname -f)" > /var/www/html/index.html
echo "Healthy" > /var/www/html/health.html
```

9. To create, click on the **Create launch template** button. Upon successful creation, it will create a Launch template.

## Task 6: Create Target group and The Load Balancer

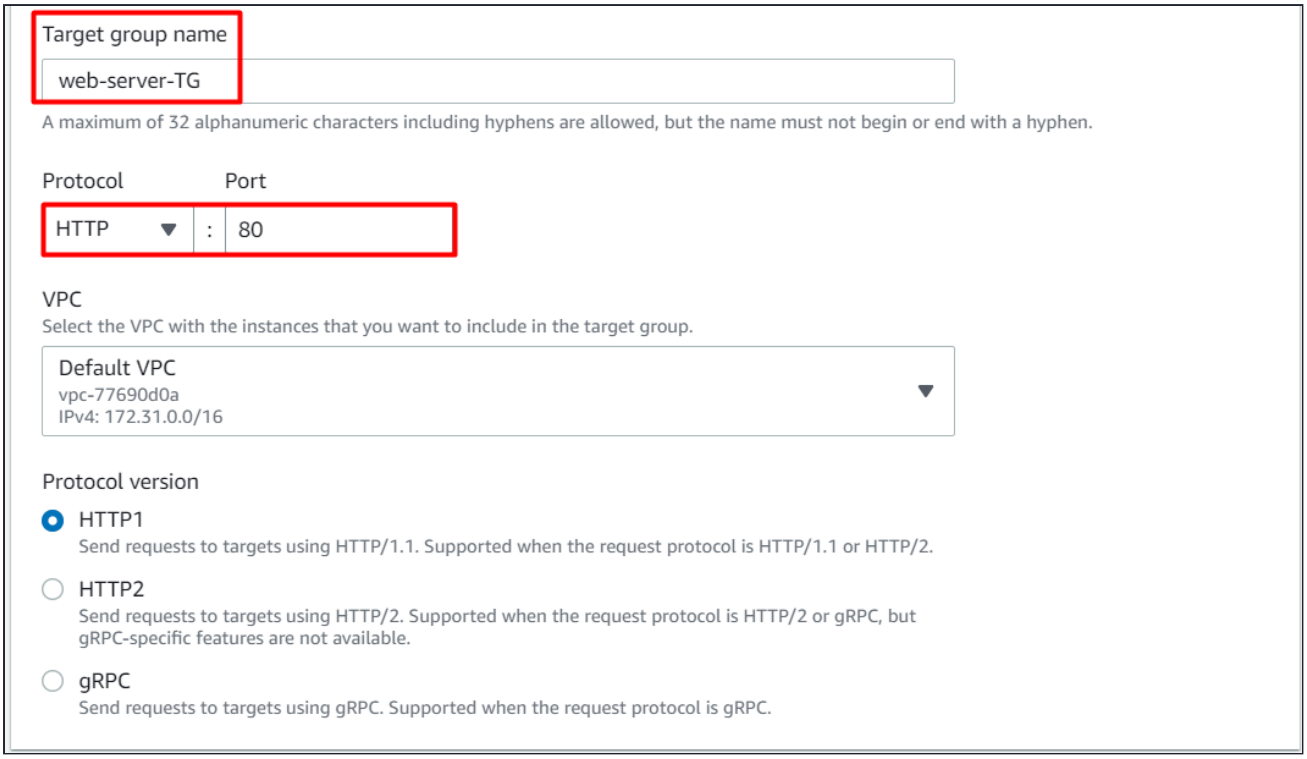
### (I) Create Target group :

1. In the EC2 console, navigate to **Target Groups** from the left navigation panel.
2. Click on **Create Target Group** button.




- **Target Type**: Select **Instances**

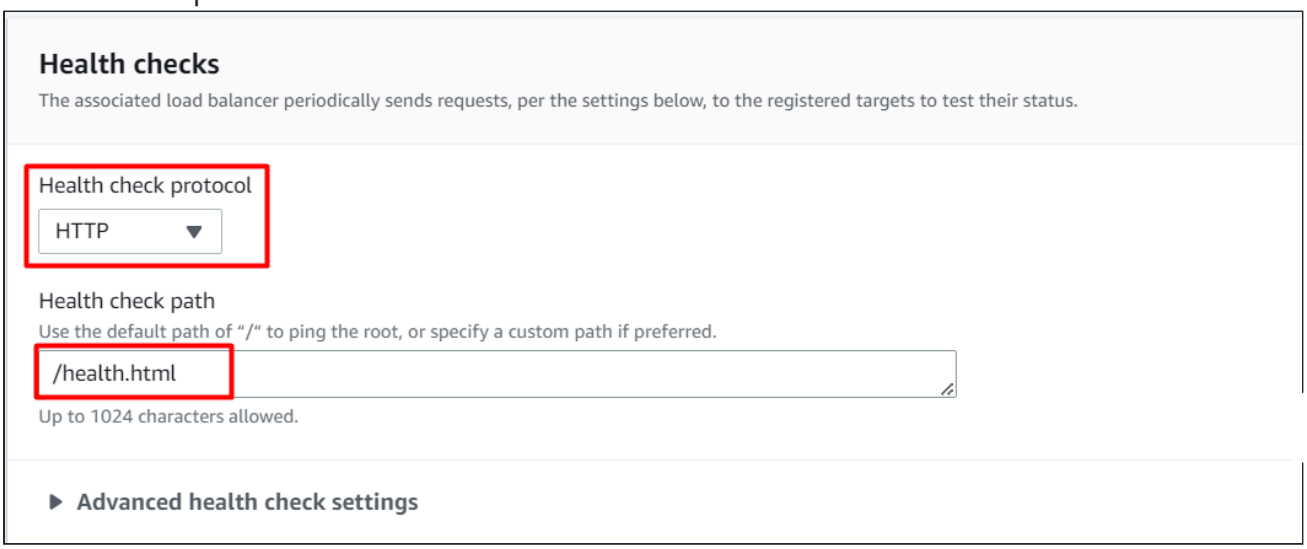
- Name: Enter **web-server-TG**
- Protocol: Choose **HTTP**
- Port : Enter **80**




- Note: The target group is used to route requests to one or more registered targets

3. Health check:

- Protocol: Select **HTTP**
- Path: Enter ***/health.html***
- Note: The load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances. These tests are called health checks.

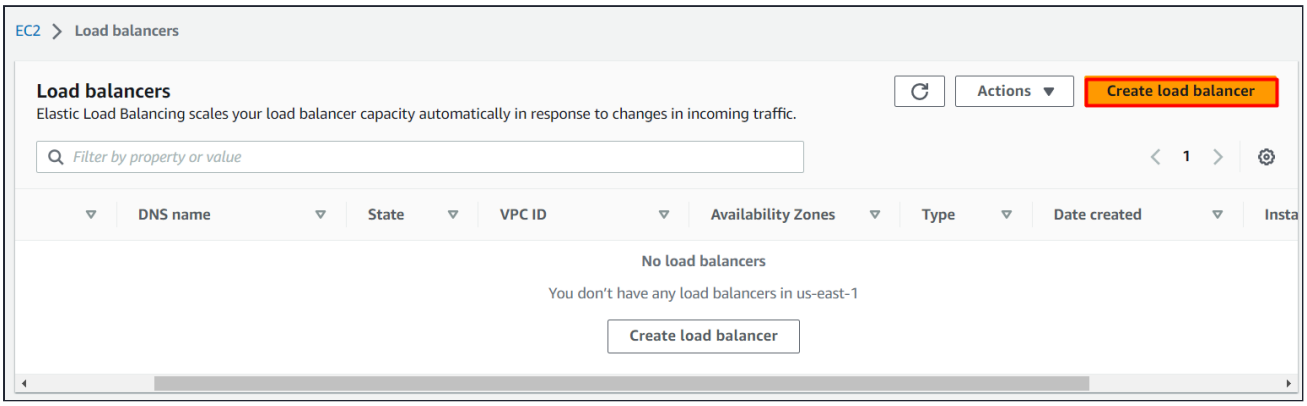



4. Click on the **Next** button.

5. Leave this page as default and click on **Create target group** button.

## (II) Create Load Balancer :

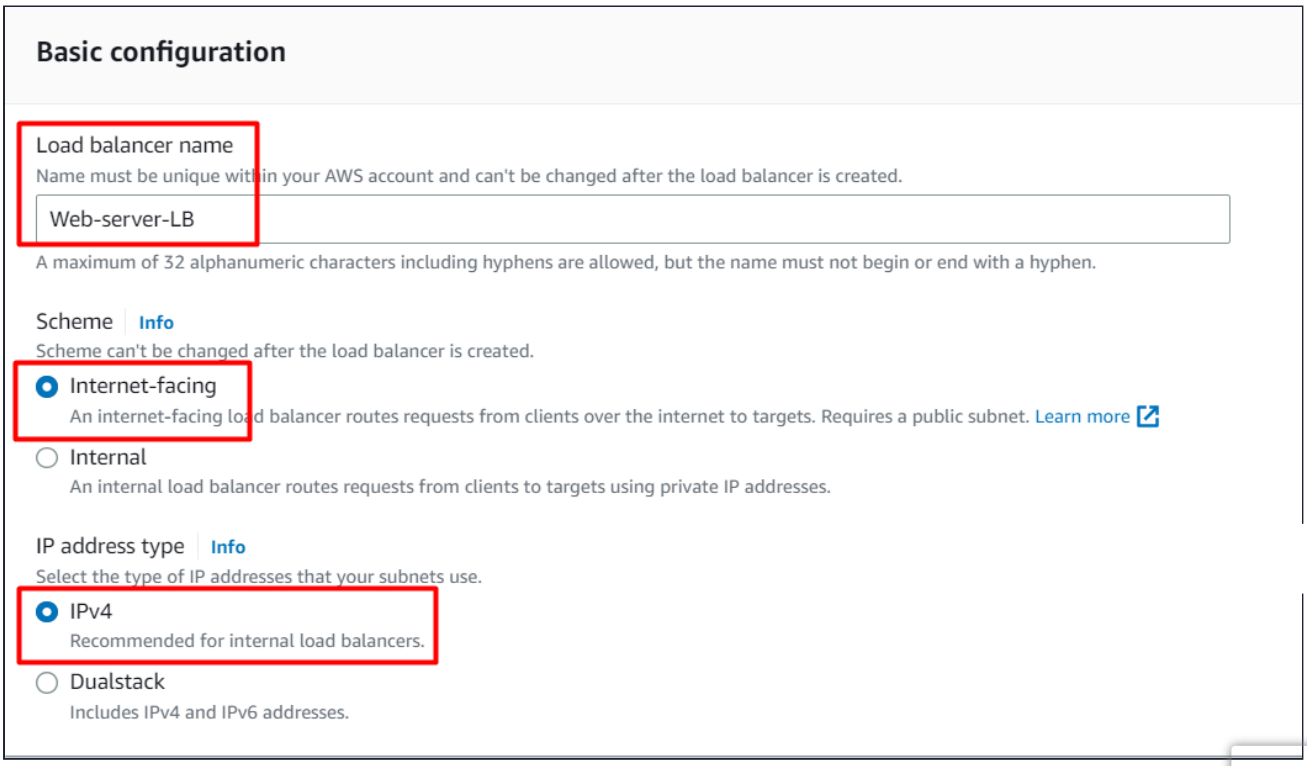
1. In the EC2 console, navigate to **Load balancers** in the left-side panel.
2. Click on **Create load balancer** button at the top-left to create a new load balancer for our web servers.




3. On the next screen, choose **Create** button under **Application Load Balancer** since we are testing the high availability of the web application.

4. In **Basic configuration:**

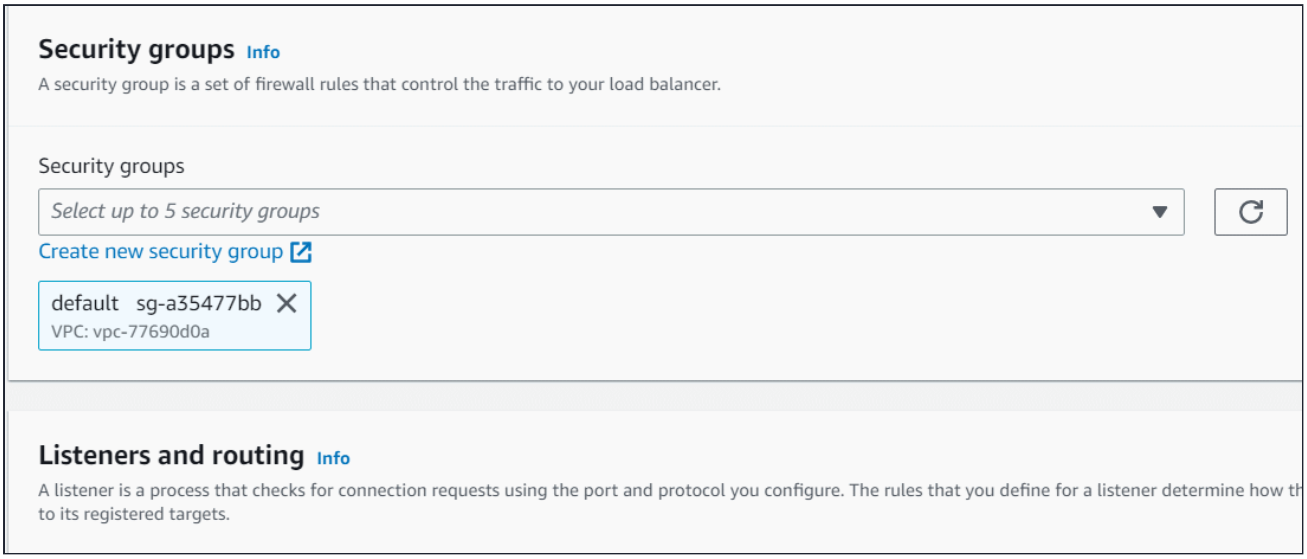
- Name: Enter **Web-server-LB**
- Scheme: Select **Internet-facing**
- IP address type: Choose **IPv4**




- **Availability Zones**

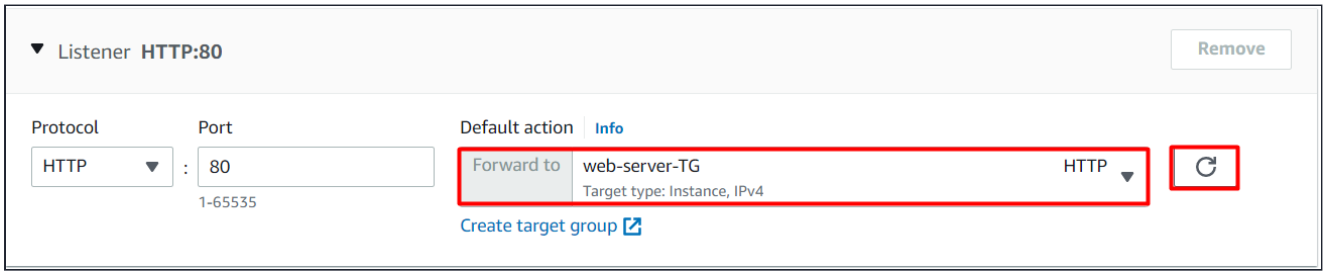
- VPC: Choose **Default**
- Availability Zones : Select **us-east-1a** and **us-east-1b.**

- **Security Groups:**
  - Remove the default one and choose **Load-balancer-SG**



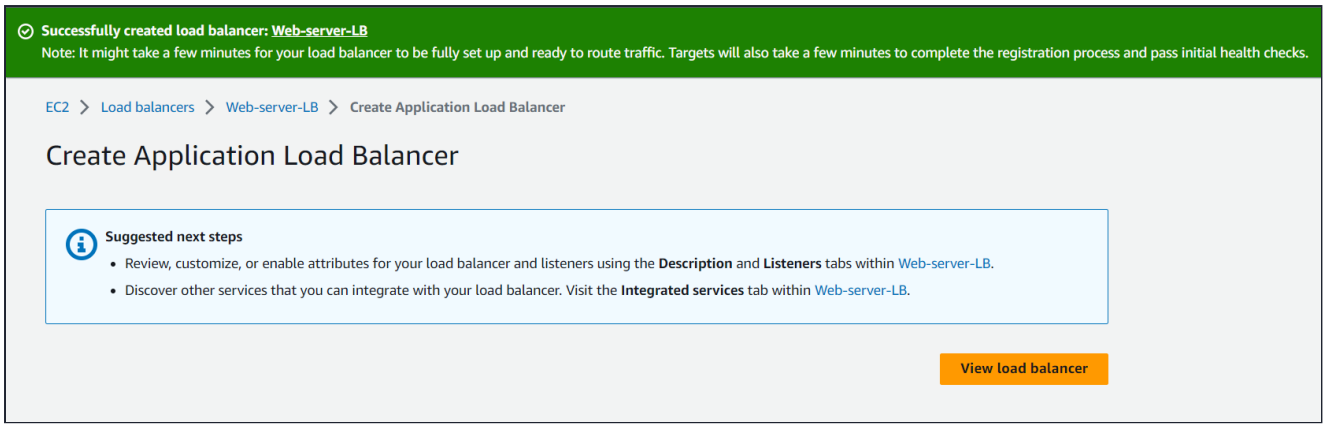

- **Note:** We must specify the availability zones in which the load balancer needs to be enabled, making it route traffic only two targets launched in those availability zones. You must include **subnets from a minimum of two Availability zones** to make our Load balancer **Highly-Available.**

5. In the listener part select the Target group that you have created earlier. if it is not visible click on refresh button.




6. Click on **Create Load Balancer** button.

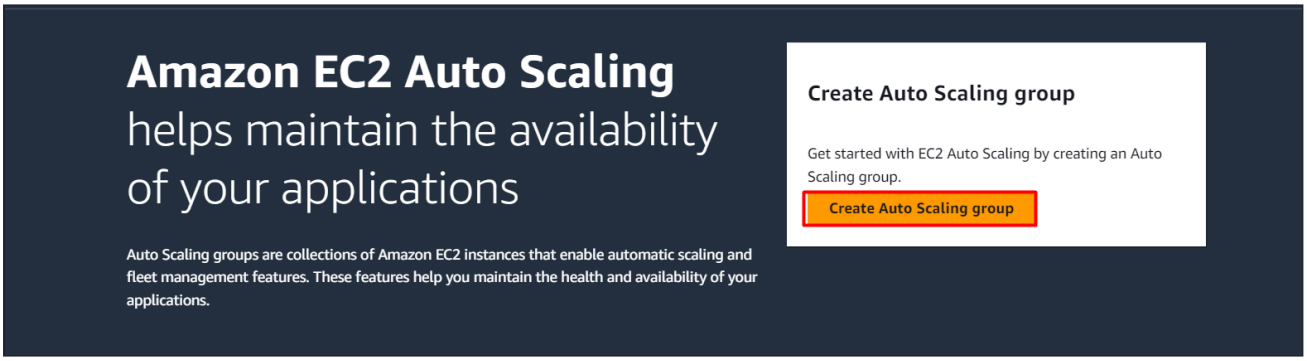
7. **You** have **successfully created** the **Application Load balancer.**



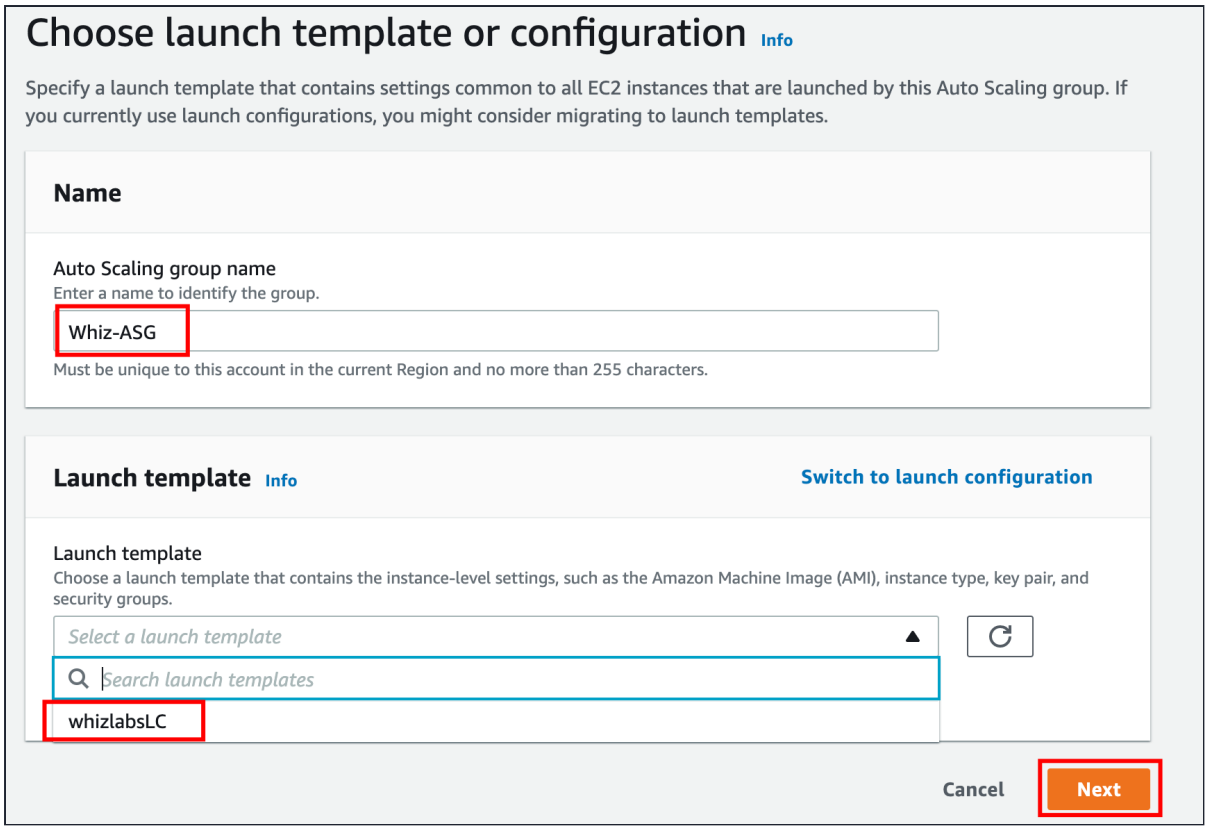

8. Wait for **2 to 3 minutes** for the load balancer to become **Active**.

## Task 7: Create an Auto Scaling Group

1. An Auto Scaling group is a scalable collection of EC2 instances. When you create an Auto Scaling group, you include information such as the subnets for the instances and the number of instances the group must maintain at all times.
2. Go to the left menu under EC2 and choose **Auto Scaling Groups** under **Auto Scaling.**
3. Click on the **Create Auto Scaling group** button.

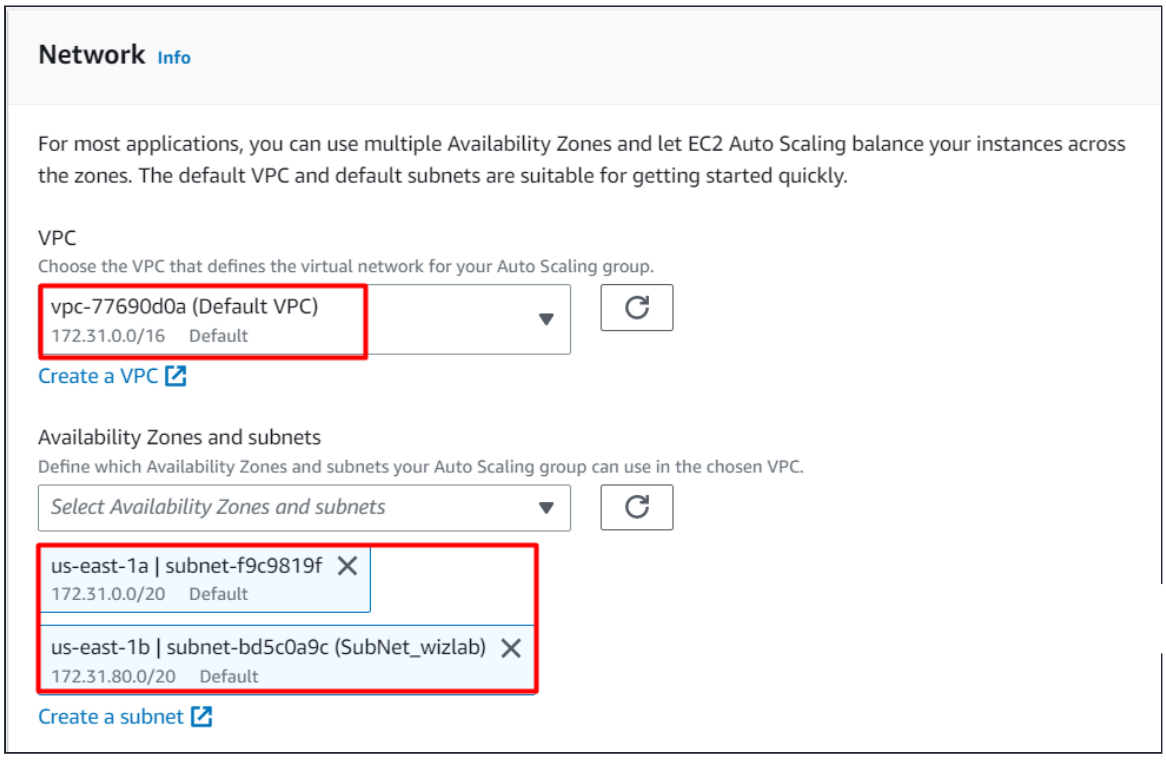



4. **Step 1 : Choose launch template or configuration**
    - Auto Scaling group name : Enter ***Whiz-ASG***
    - Select the Launch template **whizlabsLC** from the list and click on the **Next** button.




5. **Step 2: Configure settings**

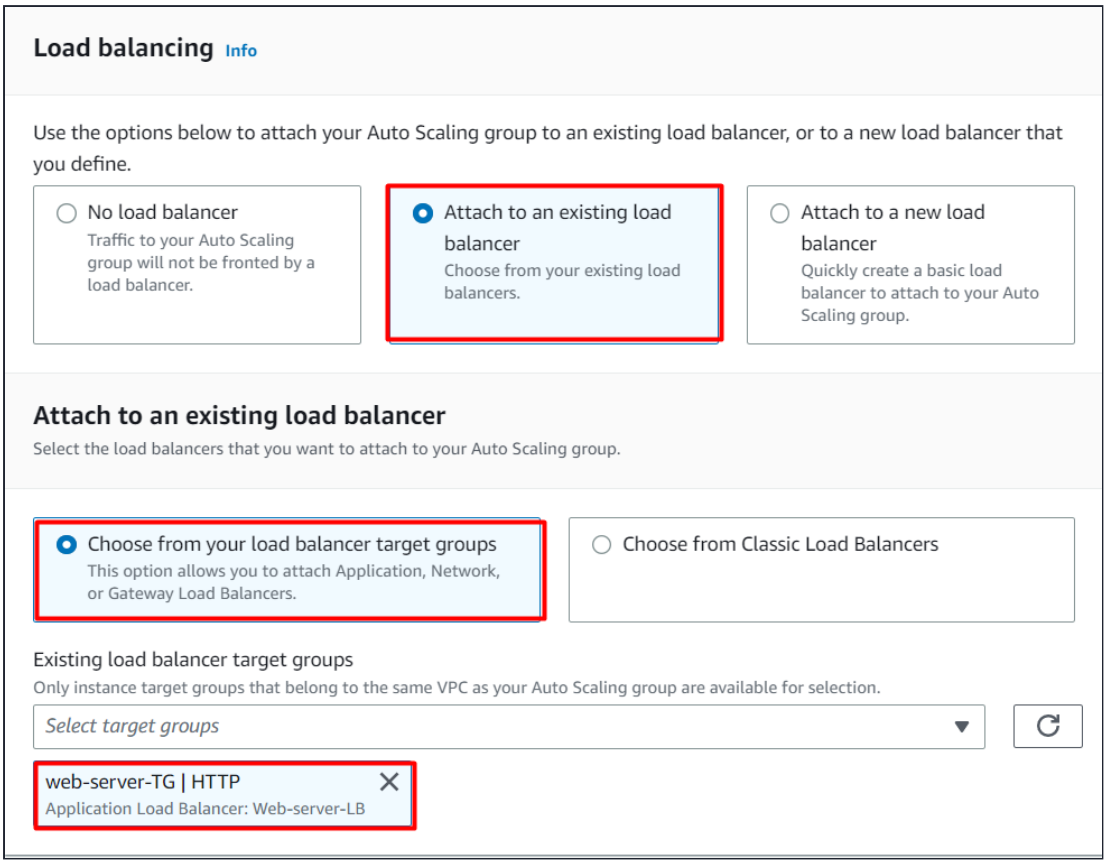
- VPC: Select the **Default VPC** from the list.
- Subnet: Select **Subnet of us-east-1a and us-east-1b**
- Click on the **Next** button.




6. **Step 3: Configure advanced options**

- Load balancing - *optional*: **Attach to an existing load balancer**

- Attach to an existing load balancer: **Choose from your load balancer target groups**
- Existing load balancer target groups: **web-server-TG**




- Health check - *optional*:
  - Health check type: EC2 (default) and **Check the** Turn on Elastic Load Balancing health checks **checkbox.**
  - Health check grace period: **60** seconds
- Click on the **Next** button.

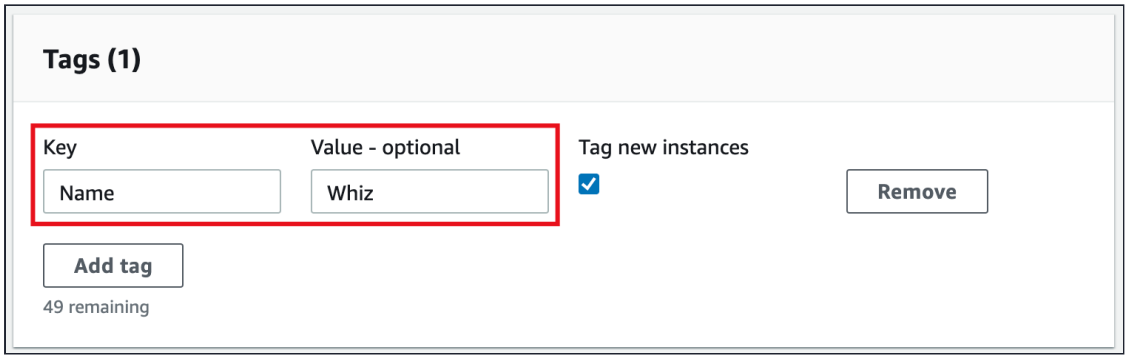
6. **Step 4: Configure group size and scaling policies**

- Under Group size - *optional*
  - Desired capacity : Enter ***1***
  - Minimum capacity : Enter ***1***
  - Maximum capacity : Enter ***4***
- **Under Scaling policies - optional**
  - Select **Target tracking scaling policy**
  - Scaling policy name: Target tracking policy

- Metric value: **Average CPU Utilization**
- Target value:Enter ***30***
- Instance need: ***60*** seconds warm up before including in metric

- Under Instance scale-in protection
  - No changes are needed, click on the **Next** button.

7. **Step 5: Add notifications**
   - No changes are needed in this page, click on the **Next** button.
8. **Step 6: Add tags**
   - Enter tags in key-value pairs to identify your auto scaling group **instances**.
     - Key: Enter ***Name***
     - Value: Enter ***Whiz***




     - Click on the **Next** button.

6. Now Review, scrolldown and click on the **Create Auto Scalling Group** button.
7. You will be redirected to the autoscaling group page, you will be able to see that one instance is launched by the autoscaling group.
8. Now go to the EC2 instances list. You will see that there are **one running instances** (which were created by your autoscaling group) You can confirm this from their tag name, which you gave at the time of creating the autoscaling group.
9. You have successfully created an autoscaling group with a policy to a minimum of 1 and a maximum of 4 instances.

## Task 8: SSH into EC2 Instance

- Please follow the steps in SSH into EC2 Instance.

## Task 9: Install the stress

1. Switch to root user:

```
sudo su
```

2. Now run the updates using the following command:

```
yum -y update
```

3. Once completed, let's install and run an stress

   - Install the required packages and libraries

```
amazon-linux-extras install epel -y
```

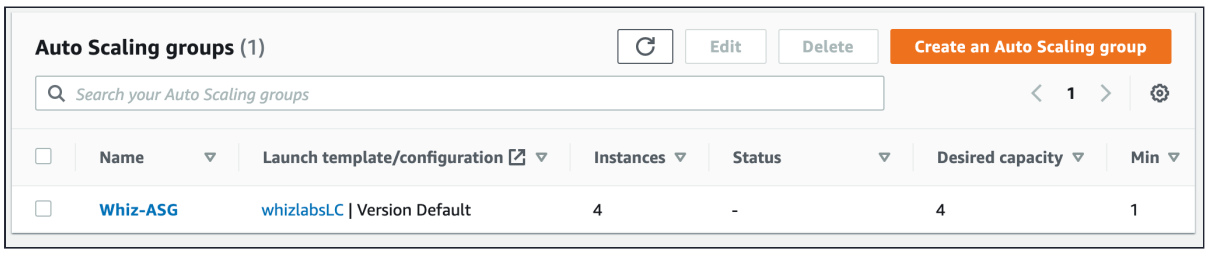
```
yum install stress -y
```

```
stress --cpu 8 --timeout 300s
```

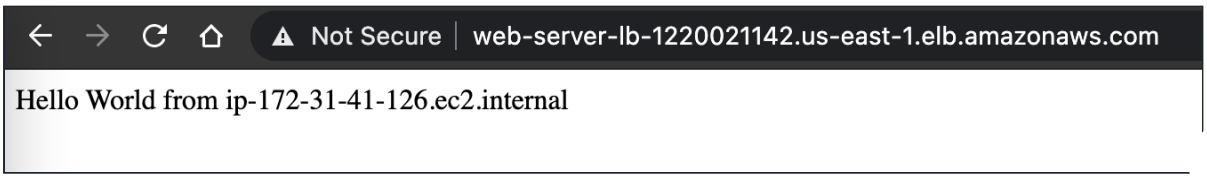
## Task 10: Test Auto Scaling Group and Elastic Load Balancer

1. For testing the auto-scaling policy, go to the EC2 instance list. You will see more new instances are getting launched.
2. That we can see in Auto scaling page also. Capacity is now incremented by 3.

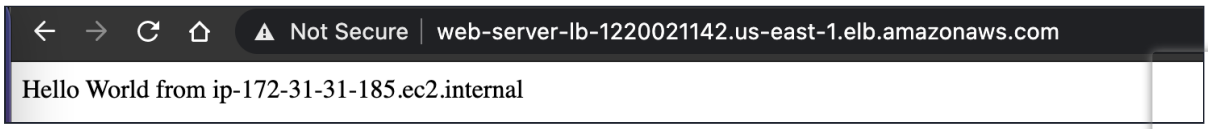
**Note: It will take 10-15 min to launch all instances**




3. Copy the DNS of your load balancer, and paste it into the browser.




4. The load balancer will now try to route to a new instance every time.

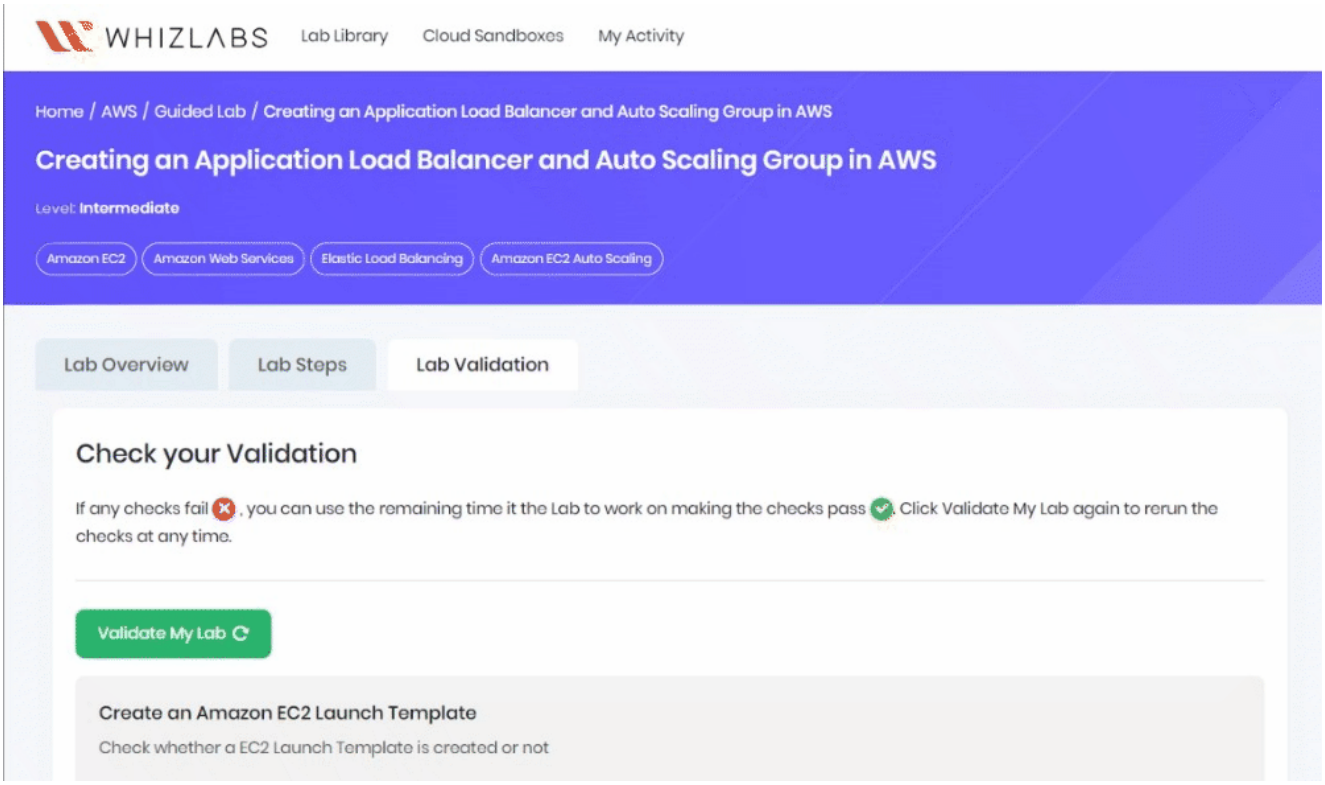


## Do you know ?

Autoscaling groups support rolling updates, which allow you to update your application without downtime. When updating the launch configuration or the Amazon Machine Image (AMI) used by the autoscaling group, it can gradually replace instances one-by-one, ensuring that your application remains available throughout the update process.

### Task 11: Validation of the Lab

1. Once the lab steps are completed, please click on the **Validation** button on the right-side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :



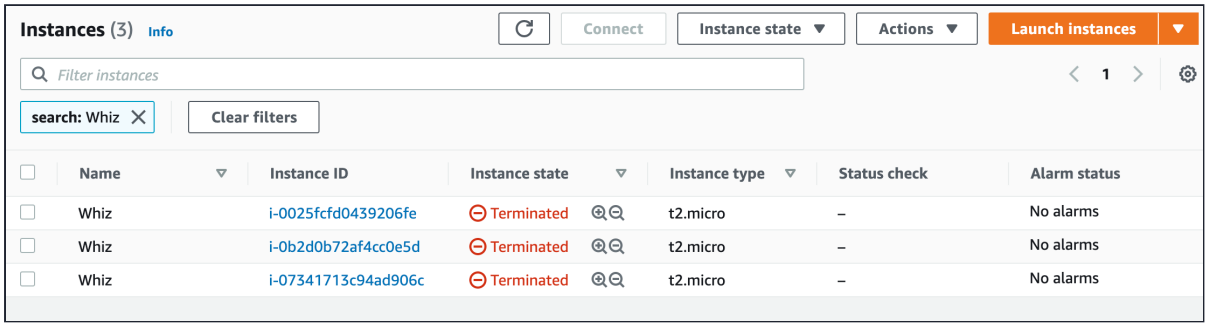

### Task 12: Delete AWS Resources

## Deleting Auto scaling group:

- In the EC2 console, navigate to **Auto scaling groups** in the left-side panel.
- **Whiz-ASG** will be listed here.
- To **delete** the Auto scaling group, need to perform the following actions:

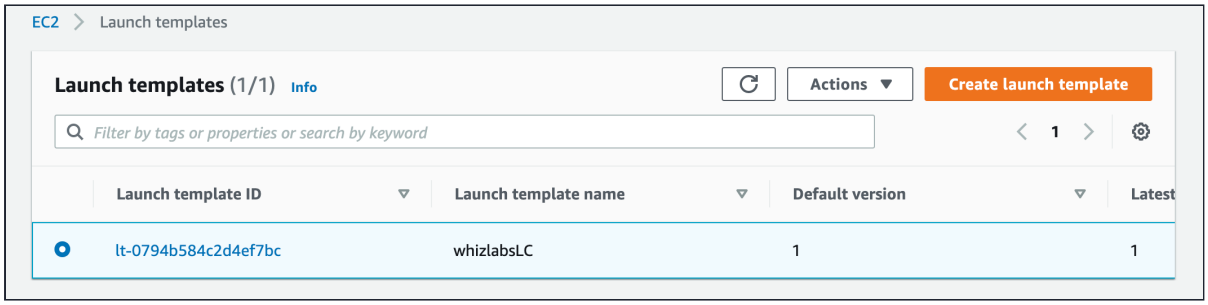
- **Select** the auto scaling group, **Whiz-ASG**
- Click on the **Actions** dropdown and then click on **Delete** ,

- Confirm by entering **delete** and click on the **Delete** button when a pop-up is shown.
- **Whiz-ASG's** status will be shown as Deleting immediately.
- It can take up to 3 minutes to delete because it will terminate the EC2 instances.
- You can confirm the termination of EC2 instances by visiting the **Instances** page.

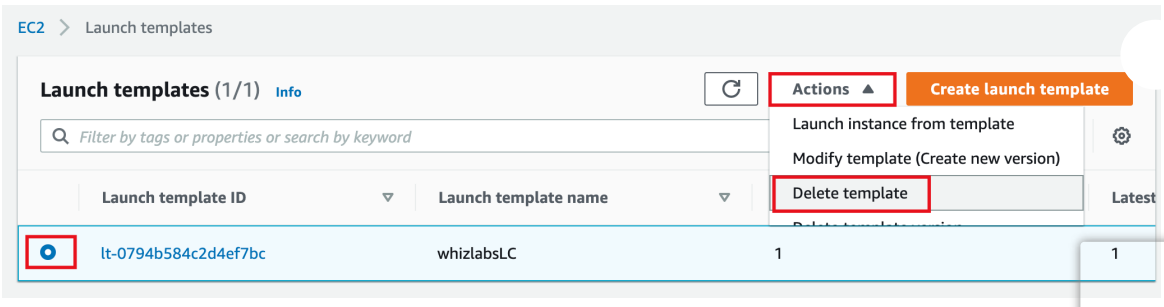



# Delete Launch templates

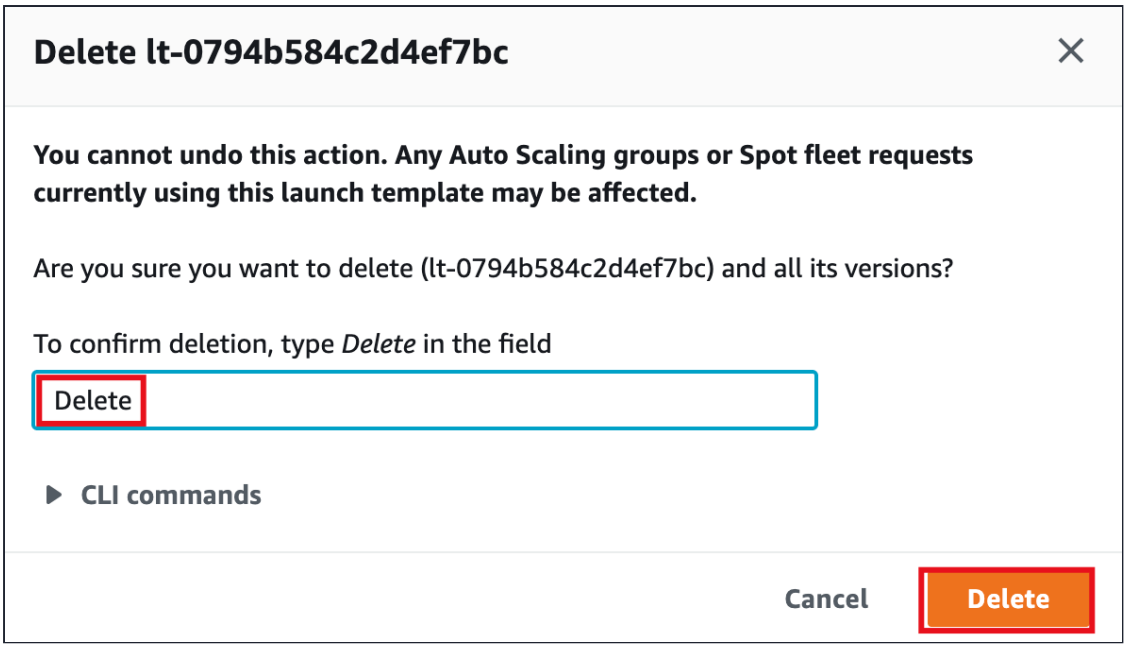
- In the EC2 console, navigate to **Launch templates** in the left-side panel.
- **WhizlabsLC** will be listed here.




- To delete the **Launch template**, need to perform the following actions:
  - **Select** the Launch template,
  - Click on the **Actions** button,
  - choose the **Delete template** option




- Confirm by **entering Delete** and clicking on the **Delete** button when a pop-up is shown.



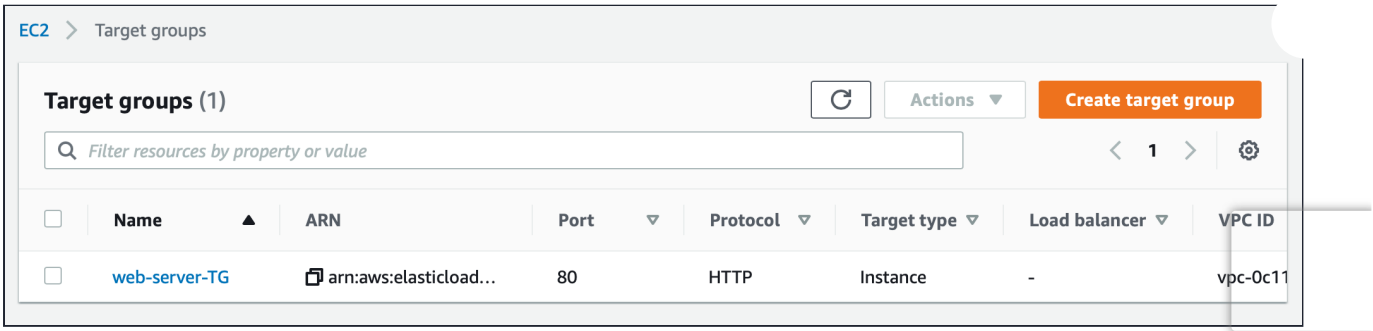

- **Whiz-ASG** will be deleted immediately.

# Delete Load balancer

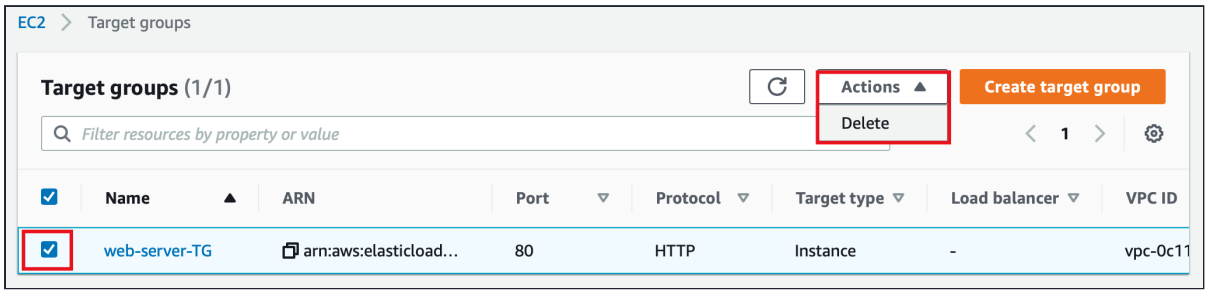
- In the EC2 console, navigate to **Load Balancers** in the left-side panel.
- **Web-server-LB** will be listed here.
- To **delete** the load balancer, need to perform the following actions:
    - **Select** the load balancer,
    - Click on the **Actions** button,
    - Select the **Delete** option.
- Confirm by clicking on the **Yes, Delete** button when a pop-up is shown.
- MyNetwork-LB will be deleted immediately.

# Delete Target group

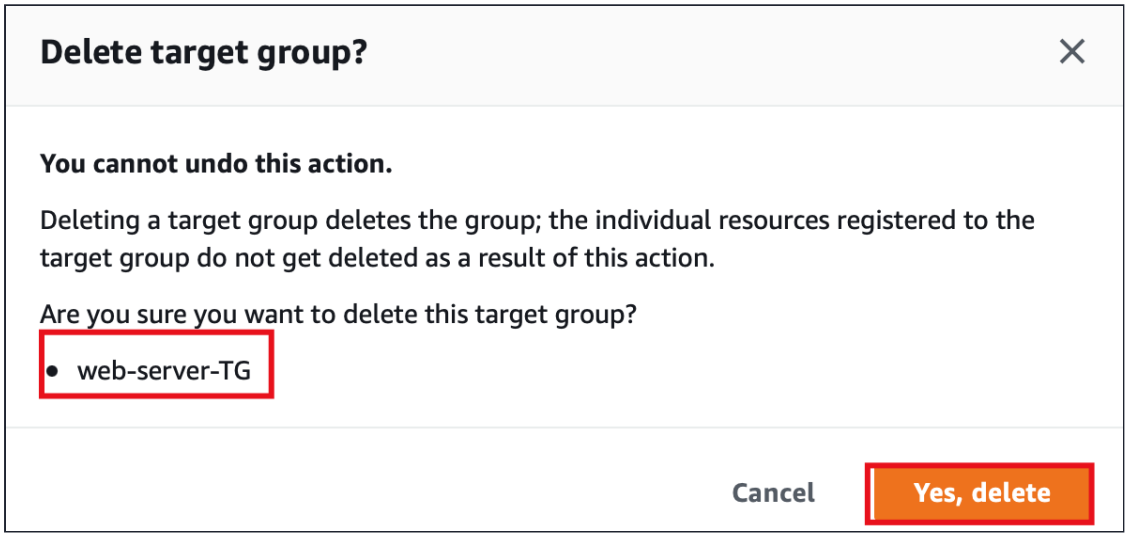
- In the EC2 console, navigate to **Target Groups** in the left-side panel.
- **web-server-TG** will be listed here.



- To **delete** the target group, need to perform the following actions:
  - **Select** the load balancer,
  - Click on the **Actions** button,
  - Select the **Delete** option.




- Confirm by clicking on the **Yes, Delete** button when a pop-up is shown.




- Target group will be deleted immediately.

# Completion and Conclusion

1. You have created a security group and key pair for the Launch template.
2. You have created a **Launch template** and **Auto Scaling**.
3. You have created an Application load balancer with Target group.
4. You have tested the Autoscaling by stressing the load on the first EC2 instance.

# End Lab

1. Sign out of the AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs dashboard.