**Lab Steps**

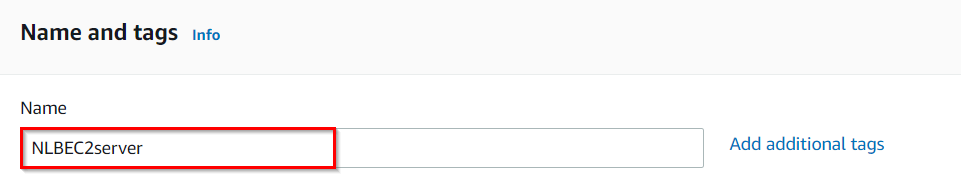
Task 1: Sign in to AWS Management Console

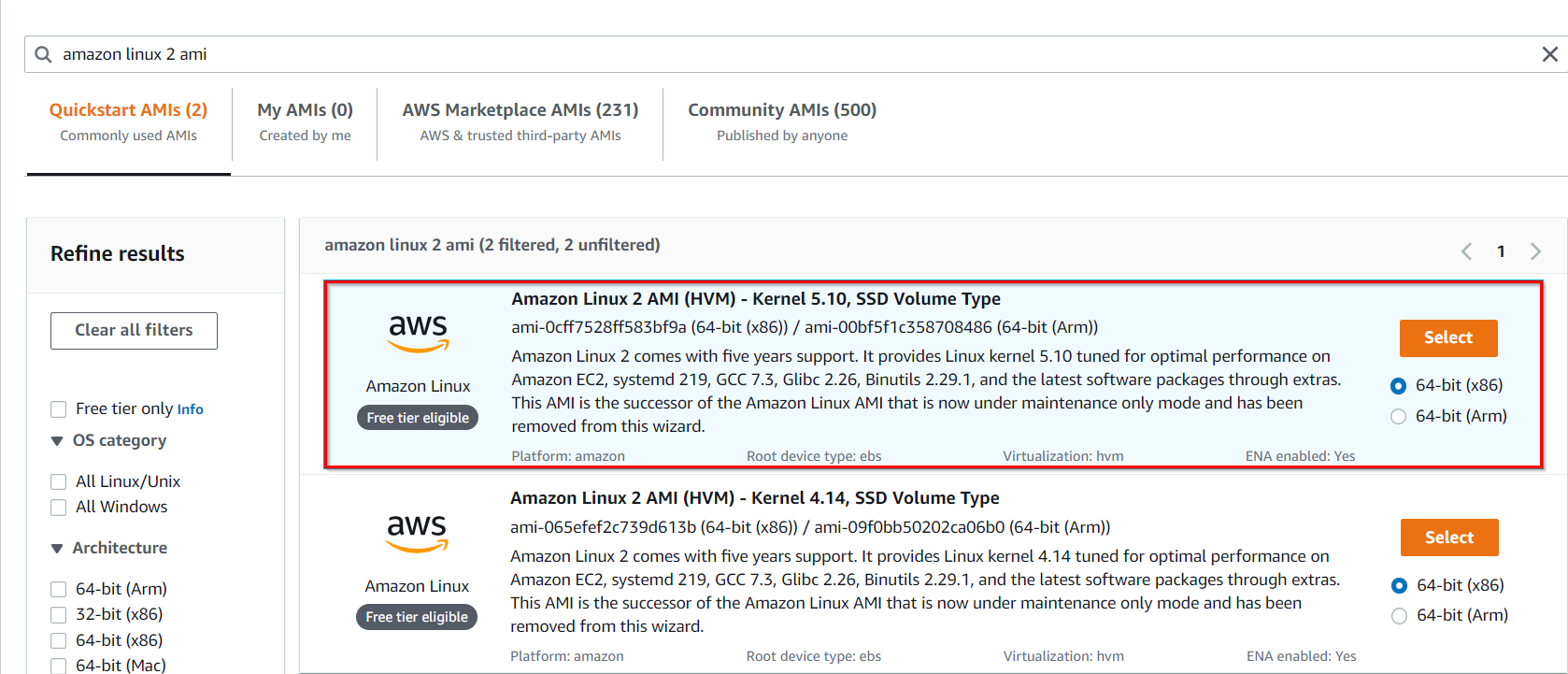
1. Click on the  button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**
4. Select Maybe later in New Home Console page pop-up

**Note :**If you face any issues, please go through [**FAQs and Troubleshooting for Labs**](https://www.whizlabs.com/labs/support-document/faqs-and-troubleshooting).

Task 2: Launching an EC2 Instance

1. Make sure you are in**US East (N. Virginia) us-east-1**Region.
2. Navigate to EC2 by clicking on the  menu in the top, then click on **EC2** in the **Compute** section.
3. Click on **Instances** from the left side bar and then click on 
4. Name : Enter ***NLBEC2server***

******

1. **For Amazon Machine Image (AMI):** Search for **Amazon Linux 2 AMI** in the search box and click on the **select** button.  
   

**Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.**

1. **For Instance Type:** select ***t2.micro***

****

1. **For Key pair:** Select **Create a new key pair** Button
   1. Key pair name: **WhizKey**
   2. Key pair type: **RSA**
   3. Private key file format: **.pem**
2. Select **Create key pair** Button.
3. In Network Settings Click on **Edit**:

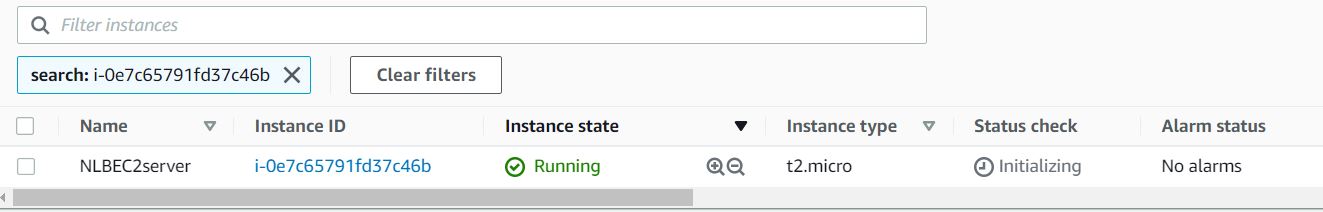
* Auto-assign public IP: **Enable**
* Select **Create new Security group**
* Security group name : Enter **NLBserver-SG**
* Description : Enter *S****ecurity Group to allow traffic to EC2***
* Check **Allow SSH from** and Select **Anywhere** from dropdown
* To add**SSH**
  + Choose Type : Select 
  + Source           : Choose **Custom** and enter ***0.0.0.0/0***
* For **HTTP**, click on **Add security group rule**.
  + Choose Type : Select **HTTP**
  + Source           : Choose **Custom** and enter ***0.0.0.0/0***
* **For  Nginx,**Click on**Add security group rule.**
  + Choose Type : Select **Custom TCP Rule**
  + Port Range    : Enter ***8080***
  + Source            : Choose **Custom** and enter ***0.0.0.0/0***

1. Click on**.**
2. Under the **User data** section, copy and paste the below code. This code installs Apache Server and also creates a web page.

|  |
| --- |
| #!/bin/bash  sudo su  yum update -y  yum install httpd -y  systemctl start httpd  systemctl enable httpd  echo “<html> <h1> Response coming from server </h1> </ html>” /var/www/html/index.html |

**Note: After pasting the user data, make sure to remove extra spacing.**

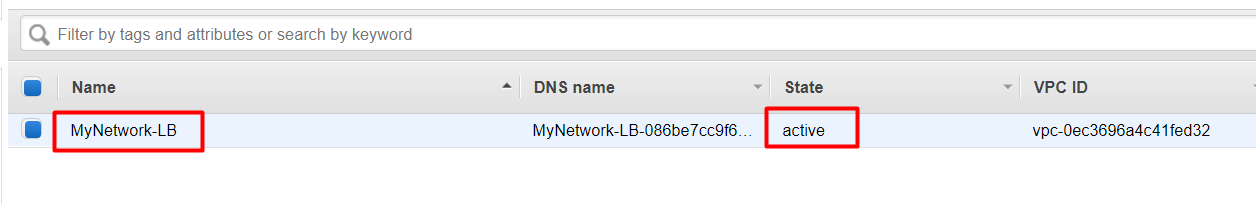
1. Keep Rest thing Default and Click on **Launch Instance** Button.
2. Select **View all Instances** to View Instance you Created
3. **Launch Status:** Your instances are now launching, Navigate to **Instances** page from left menu and wait the status of the EC2 Instance changes to running
4. Now in the EC2 dashboard, you can see the instance is running as shown below:



Task 3: Creating Network Load Balancer

1. Navigate to **Load Balancers** from the EC2 left menu bar under the sub-heading **Load balancing**.
2. Click on the .
3. **Select Load Balancer Type**: Under the **Network load balancer**, click on **Create** button.
4. The next five screens will require configuration modification from the default values provided. If a field is not mentioned, leave it as default or empty.

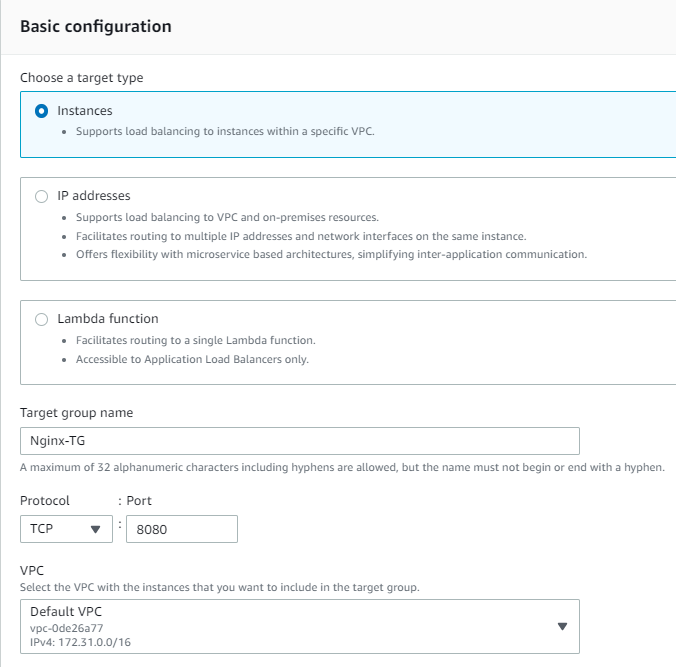
* **Configure Load Balancer:**
  + Name            : Enter ***MyNetwork-LB***
  + Scheme        : Select 
  + VPC            : Select **default VPC**
  + Mappings : **Select all available zones** using a checkbox.
* Listeners and routing:
  + Load Balancer Protocol    : Select **TCP**
  + Load Balancer Port    **:** Choose **80**
  + Default action : **Create target group**
    - Choose a target type: **Instances**
    - Target group name: Enter ***Apache-TG***
    - Protocol: Choose **TCP**
    - Port: **80**
    - Click **Next**
    - Select **NLBEC2Server** and select **Include as pending below** button
    - Click **Create target group** button
    - Ignore the warning message and click **Continue**
    - Now we can see **Apache-TG** target group created.
* Now go back to the **Load Balancers** tab, and select **Apache-TG** in **Default Action** in **Listeners and routing.**
* Click on **Create load balancer** button
* Click on **View Load Balancers**
  + After 1-2 minutes the state will change to **active** and is ready.



Task 4: Create Target Group for Nginx

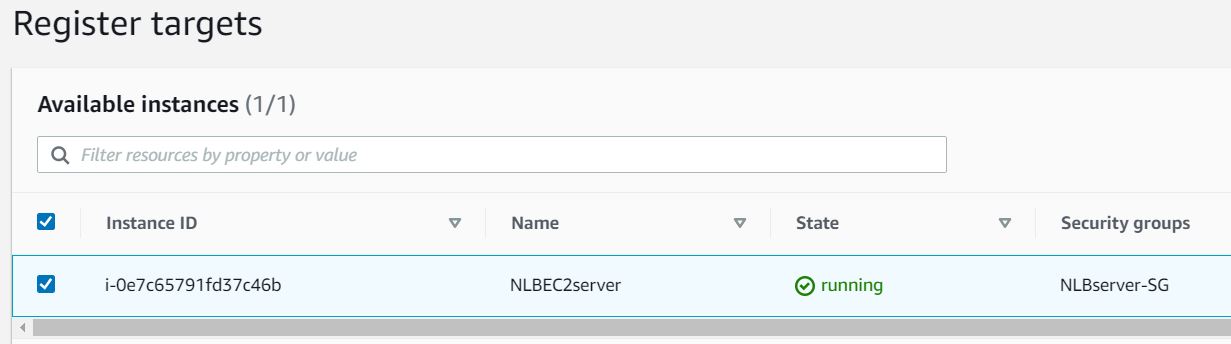
1. Navigate to  available in the left menu of EC2 Page. It will be available under the sub-heading **Load Balancing**.
2. Click on ****
3. In the Create Target Group form Basic configuration:

* Choose a target type: **Instances**
* Target group name     : Enter ***Nginx-TG***
* Protocol        : choose **TCP**
* Port            : Enter ***8080***



* Leave other settings as default. Click on the **Next** button.

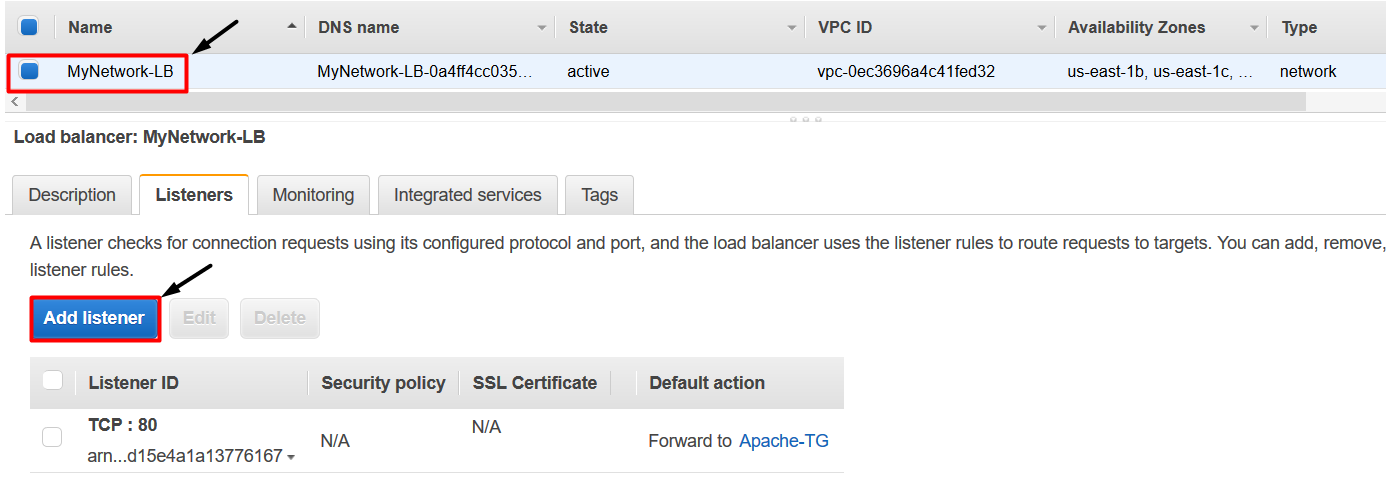
1. Select the **NLBEC2server** and select**Include as pending below**button
2. Click **Create target group**button



1. Ignore the warning message and click **Continue.**

Task 5: Creating a New Listener for Nginx on port 8080

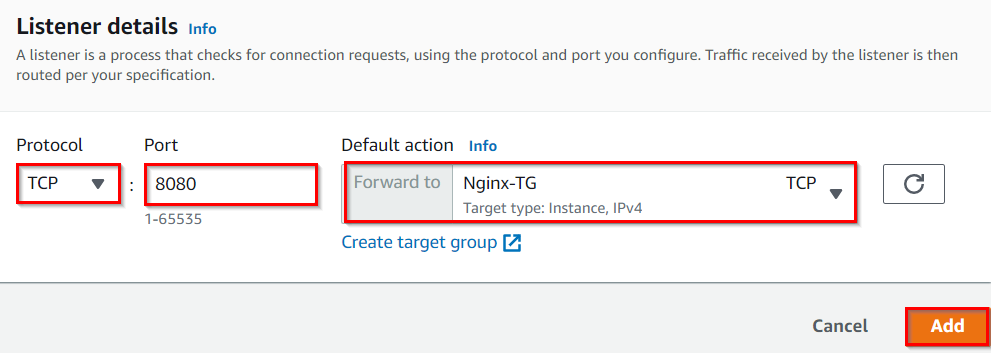
1. Now we need to add a new listener for Nginx to listen for traffic on port 8080.
2. Navigate to the Load Balancer console and select the**MyNetwork-LB.**Click on  as shown below:



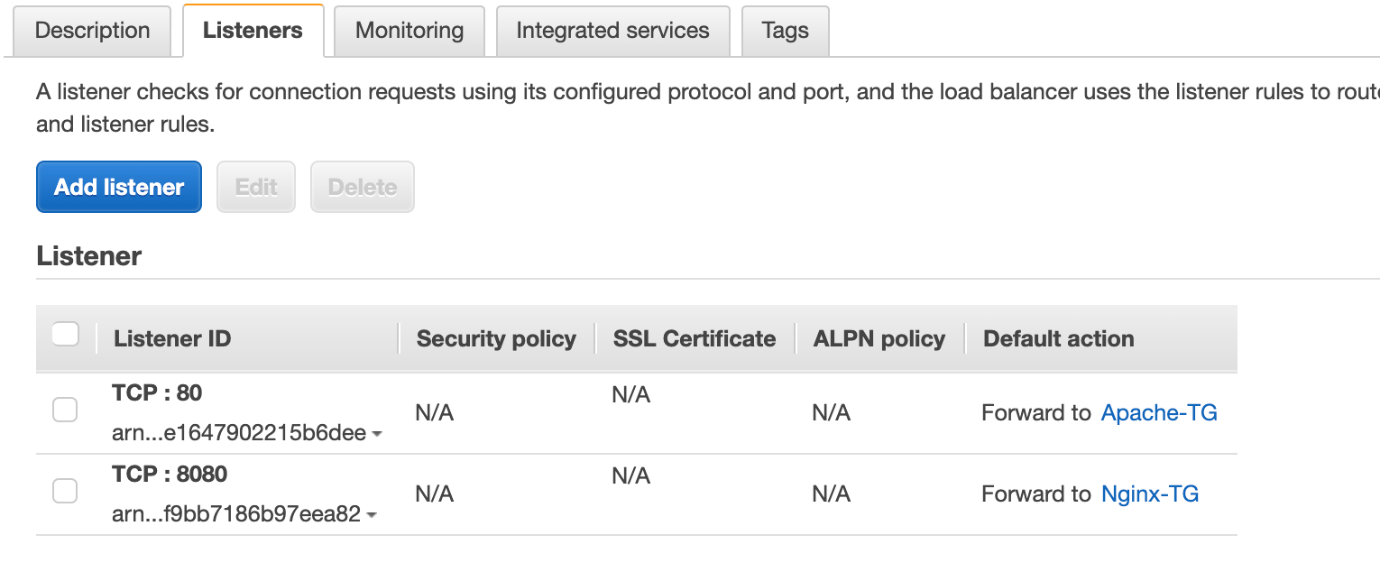
1. Now add the new listener as follows:

* Protocol    : Select **TCP**
* Port        : Enter ***8080***

1. Chose the **Nginx-TG** as shown below:



1. Click on **Add** and Click on **View Listeners**
2. Both Listeners will be shown.



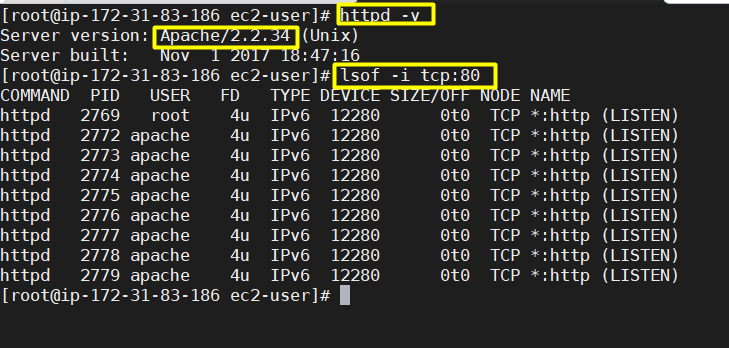
Task 6: Installing and configuring Nginx

1. SSH into the EC2 instance, following the steps in [SSH into EC2 Instance](https://www.whizlabs.com/labs/support-document/ssh-into-ec-instance).
2. Switch to the root user using the command:

sudo su

1. You will see apache is already installed and listening on port 80. To do so, run the following commands:

* To find version:
  + httpd -v
* To find HTTPD running:
  + lsof -i tcp:80



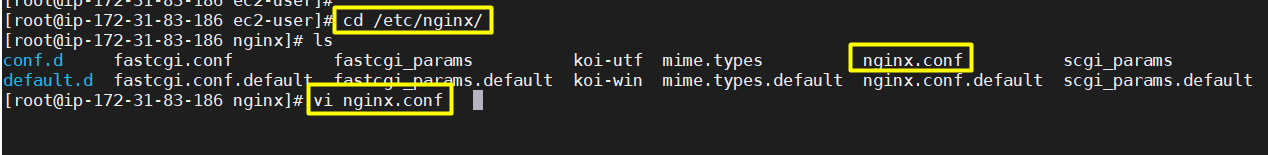
1. Now install **Nginx** using below command:

 sudo amazon-linux-extras install nginx1 -y

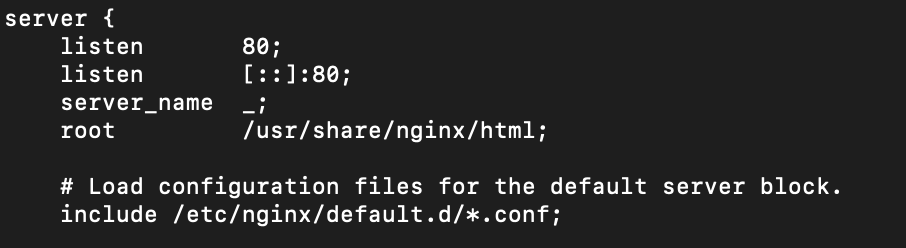
1. Now we need to change the port number for Nginx service to 8080 because HTTPD service already running on port. To do so, type the following commands:

cd  /etc/nginx

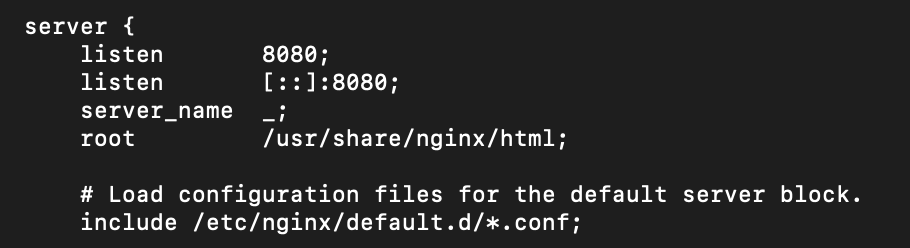
vi  nginx.conf



1. Scroll down and edit the 'listen' line to port number 8080. To edit in vi, press **i** to insert. After editing, press **esc** and give command **:wq** which saves the file and quits from vi.



1. Now change the default **port number 80 to 8080**as shown below:

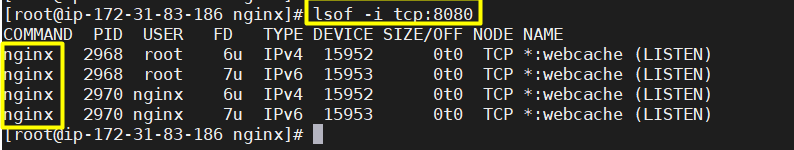


1. Start the Nginx service using below command:

* service nginx start

1. You can check the status of Nginx service using the below command:

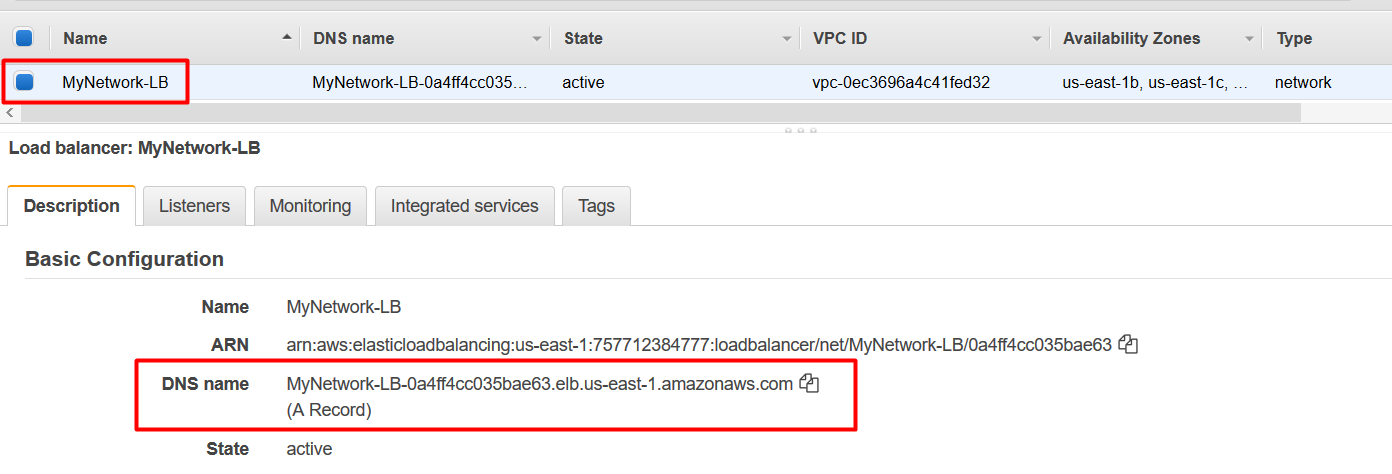
* lsof -i tcp:8080



Task 7: Testing Network Load Balancer

1. Wait 3 to 4 minutes until the**target reaches healthy status** and navigate to the Load balancer console and copy the DNS name. Enter the address in the browser.

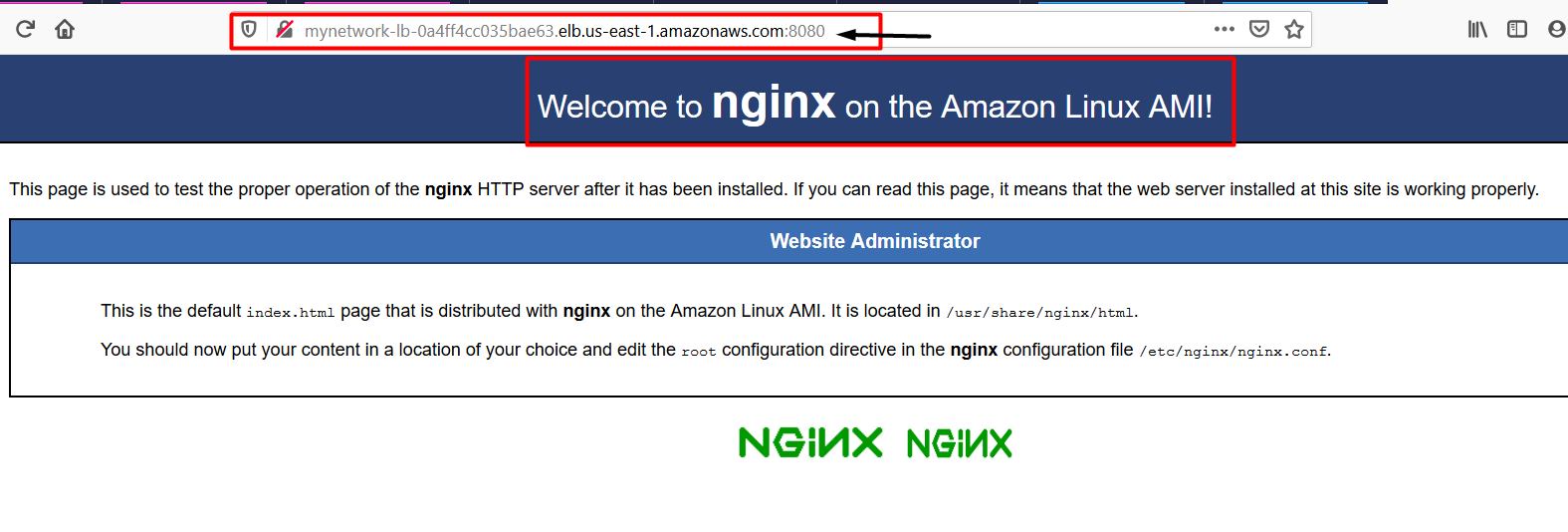
* **MyNetwork-LB-0a4ff4cc035bae63.elb.us-east-1.amazonaws.com**



1. You will see the default apache page. **By default network load balancer will route the traffic to port 80**



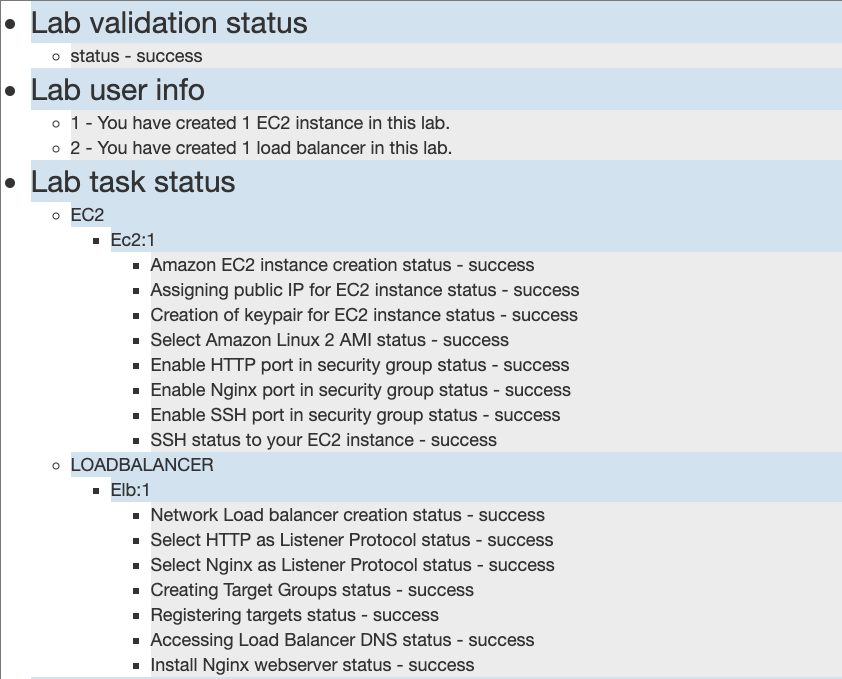
1. Now repeat the above step by appending port number **8080** at the end of load balancer endpoint (as shown below) and you will get the Nginx default page. **MyNetwork-LB-0a4ff4cc035bae63.elb.us-east-1.amazonaws.com:8080**

****

1. From the above steps we have **successfully created the Network load balancer** and routed the traffic to two different services ( Apache and Nginx ) listening on two different ports ( 80 and 8080 ).
2. We have also successfully created and tested the Network Load Balancer.

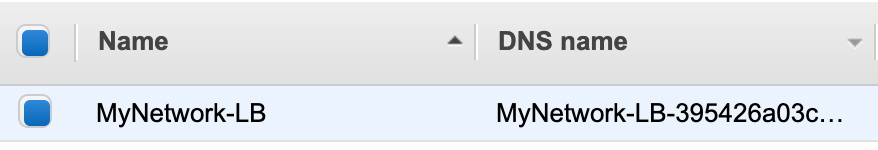
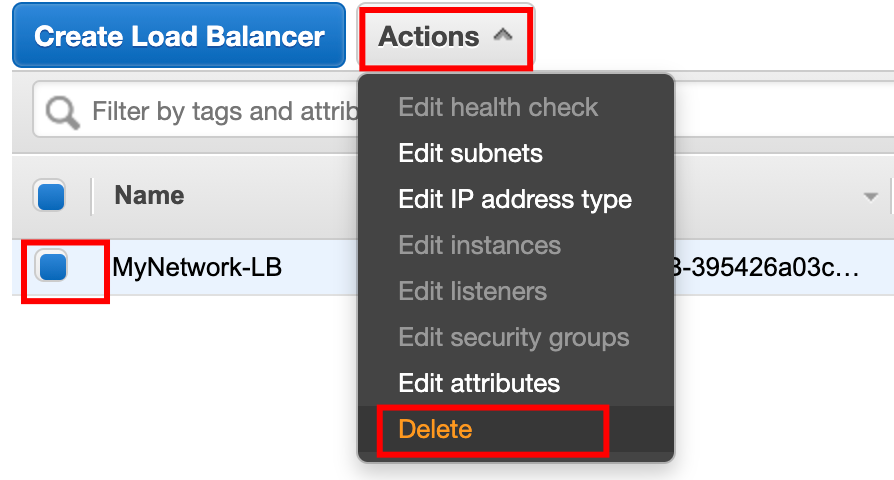
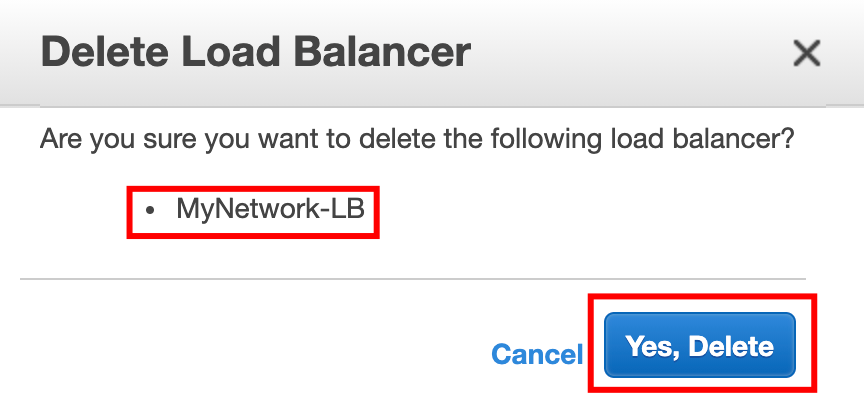
Task 8: Validation Test

1. Once the lab steps are completed, please click on the  button on the left 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 :

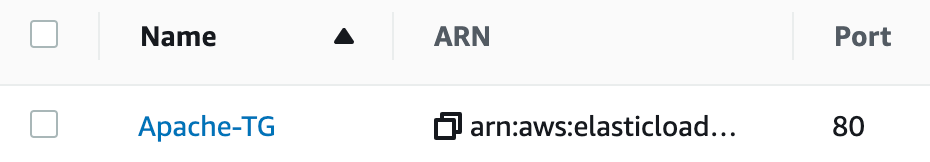
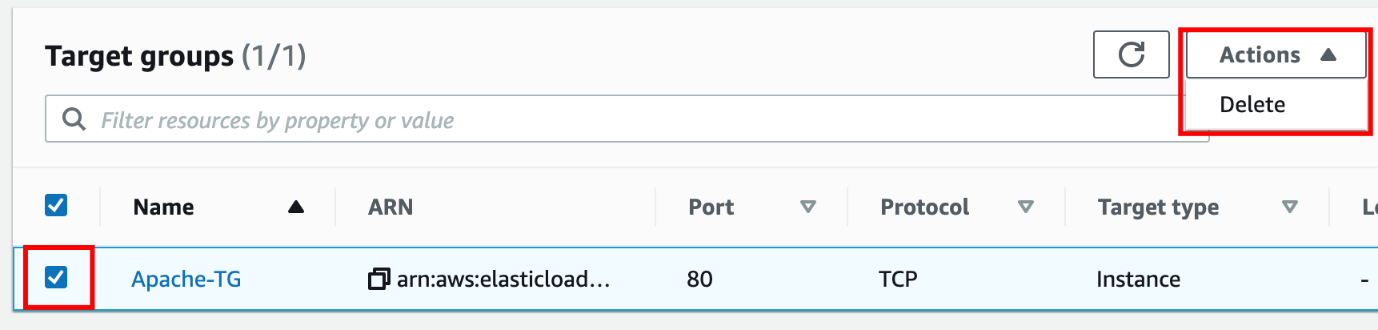
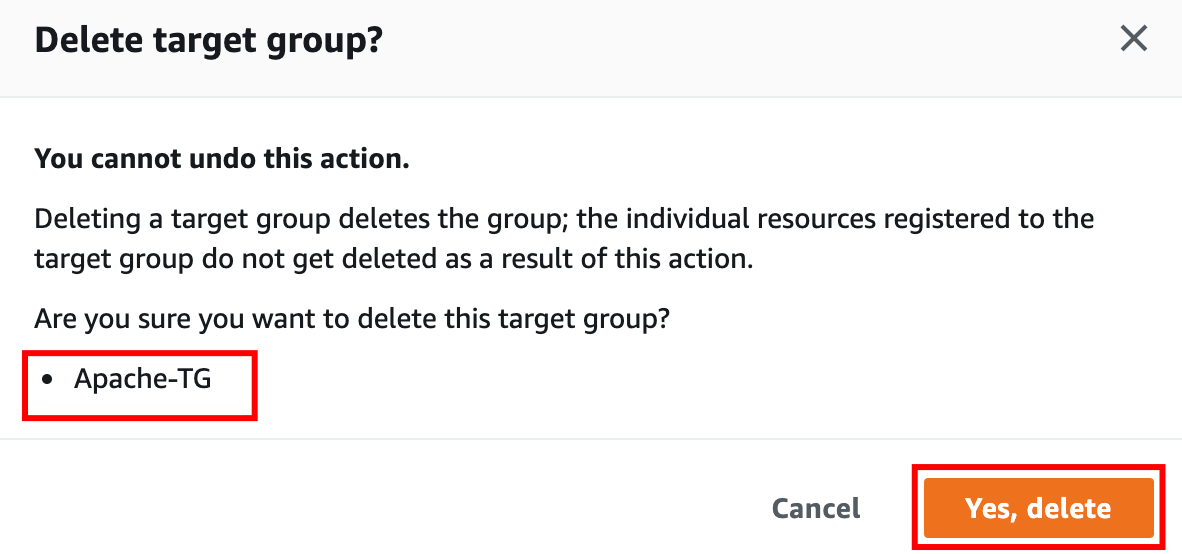


Task 9: Delete AWS Resources

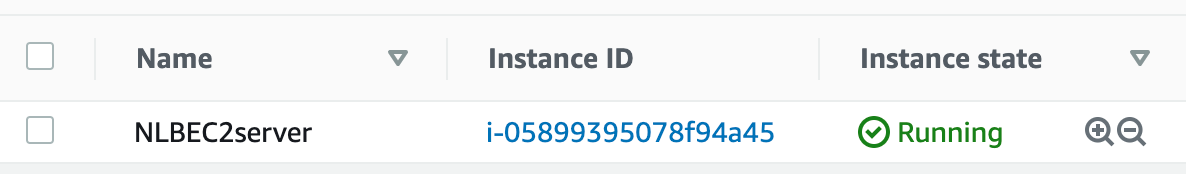
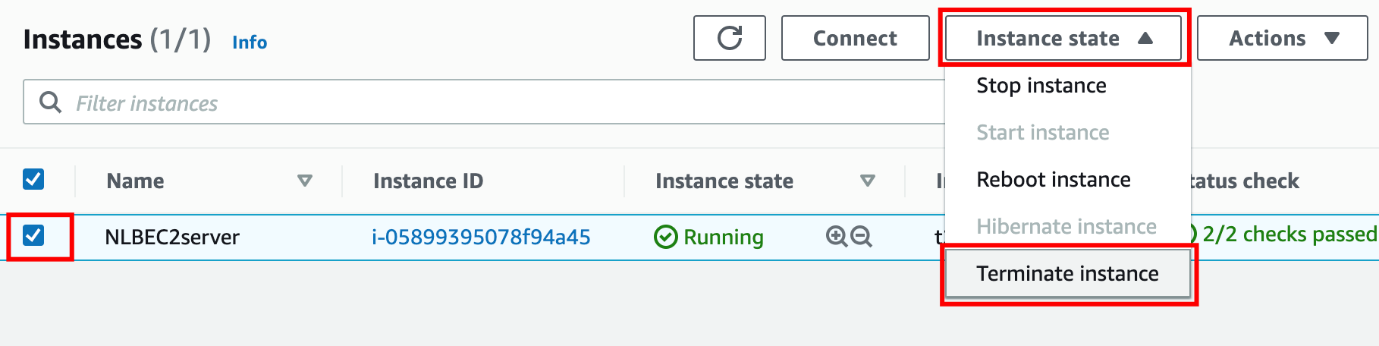
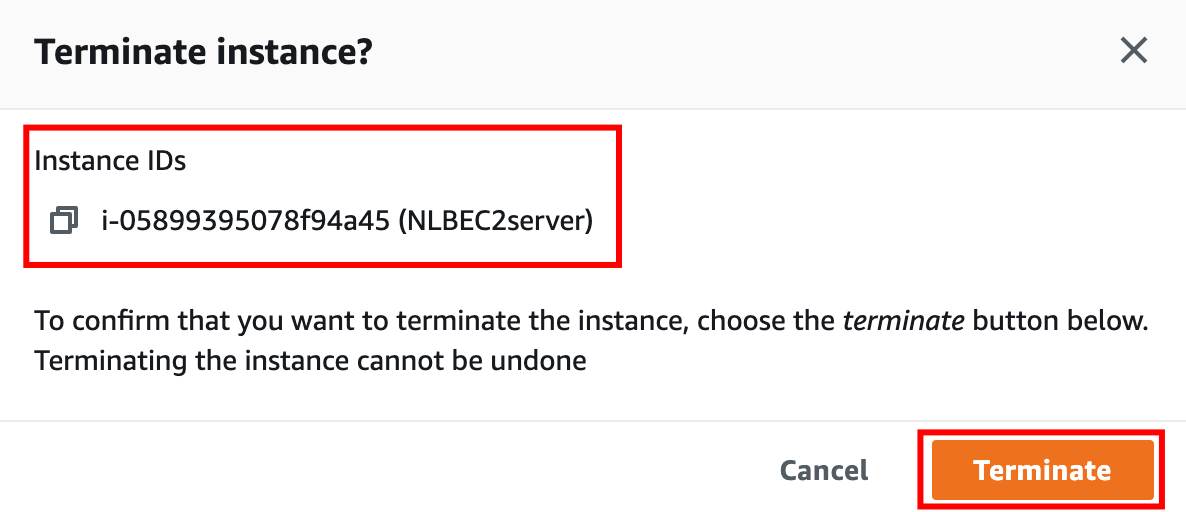
Deleting Load balancer

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

Deleting Target groups

1. In the EC2 console, navigate to in the left-side panel.
2. **Apache-TG** will be listed here.  
   
3. To delete the **target group**, need to perform the following actions:
   * **Select** the target group,
   * Click on the **Actions** button,
   * select the **Delete** option  
     
4. Confirm by clicking on the **Yes, Delete** button when a pop-up is shown.  
   
5. Web-server-TG will be deleted immediately.  
   

Terminate EC2 Instances

1. In the EC2 console, navigate to in the left-side panel.
2. EC2 Instance **NLBEC2Server** will be listed here.  
   
3. To terminate the **EC2 Instances**, need to perform the following actions:
   * **Select** the EC2 instances,
   * Click on the **Instance state** button,
   * select the **Terminate instance** option  
     
4. Confirm by clicking on the **Terminate** button when a pop-up is shown.  
   
5. **EC2 Instances** will be terminated immediately.  
   

**Completion and Conclusion**

1. You have successfully used the AWS management console to create an Amazon EC2 instance.
2. You have created a Network Load Balancer with an Apache Target group.
3. You have created an Nginx Target group and attached the EC2 server to route traffic on port 8080.
4. You tested the Network Load Balancer.

**End Lab**