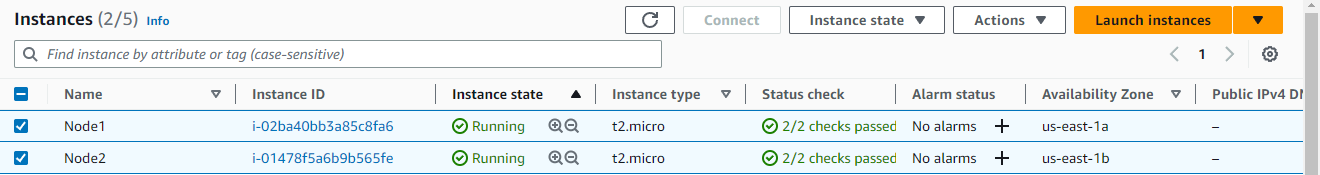
**Project C: 3**

# Run web Application on multi-node architecture with HA and failover

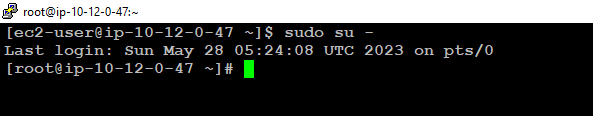
# **To create two EC2 instance in different availability zone:**

1. Node1 (us-east-1a)
2. Node2 (us-east-1b)

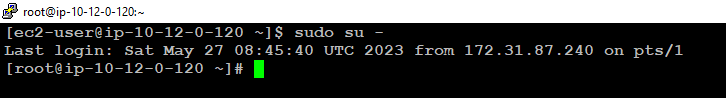


## -To connect both EC2 instance using putty software:

**Node1:**



**Node2:**



# **To Install Apache service on both Instance:**

## **-**Start apache and check status of an apache service on both Node:

$sudo su -

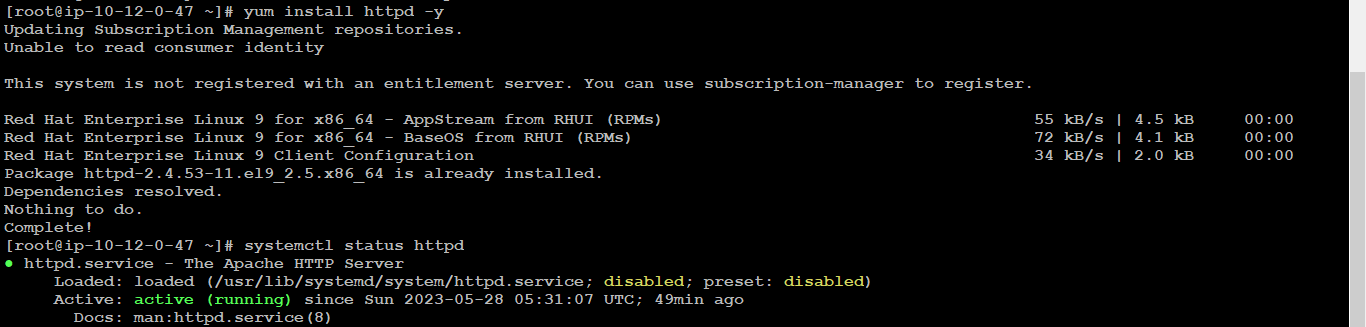
#yum update -y

#yum install httpd -y

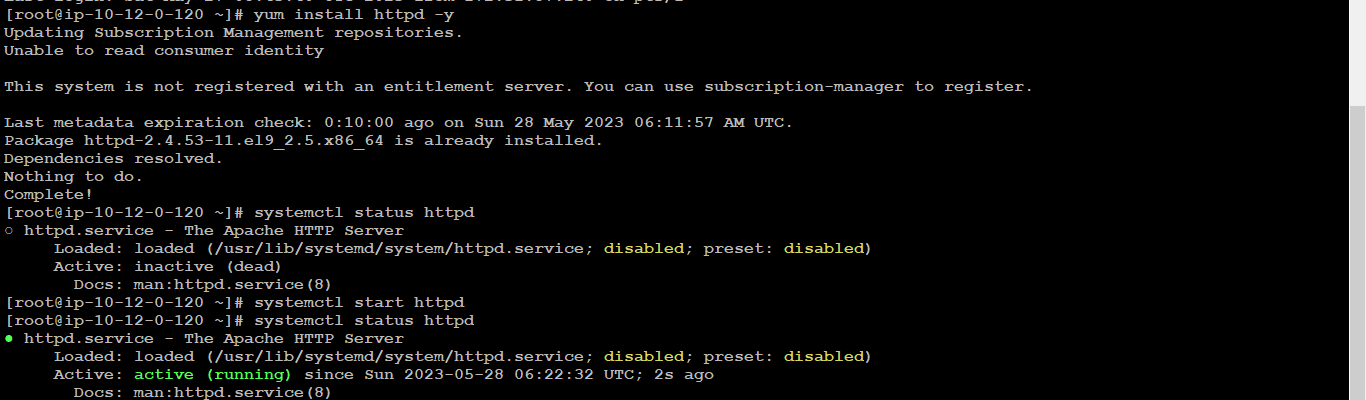
#systemctl start httpd

#systemctl status httpd

**Node1:**



**Node2:**



# **To copy the html file to document root on both instance:**

## -Document root:

#cd /var/ww/html/

## -Create two directory /var/www/html on Node1 and Node2 server:

#cd /var/www/html/

#mkdir rinfotech rinfoservice

#ls –ltr

#cd rinfotech

#vim index.html

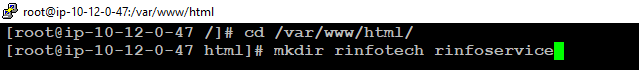
#cat index.html

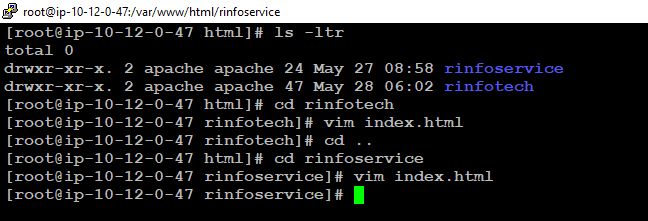
#cd rinfoservice

#vim index.html

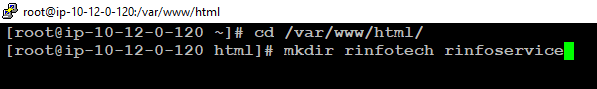
#cat index.html

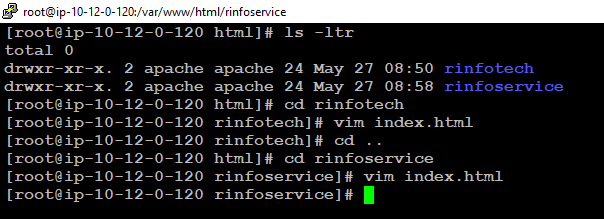
**Node1:**





**Node2:**



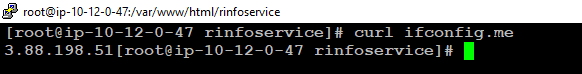


# **-To verify the both nodes, websites accessible or not:**

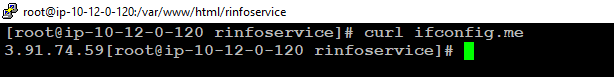
## -Find the public ip using the following command:

#curl ifconfig.me

**Node1:**

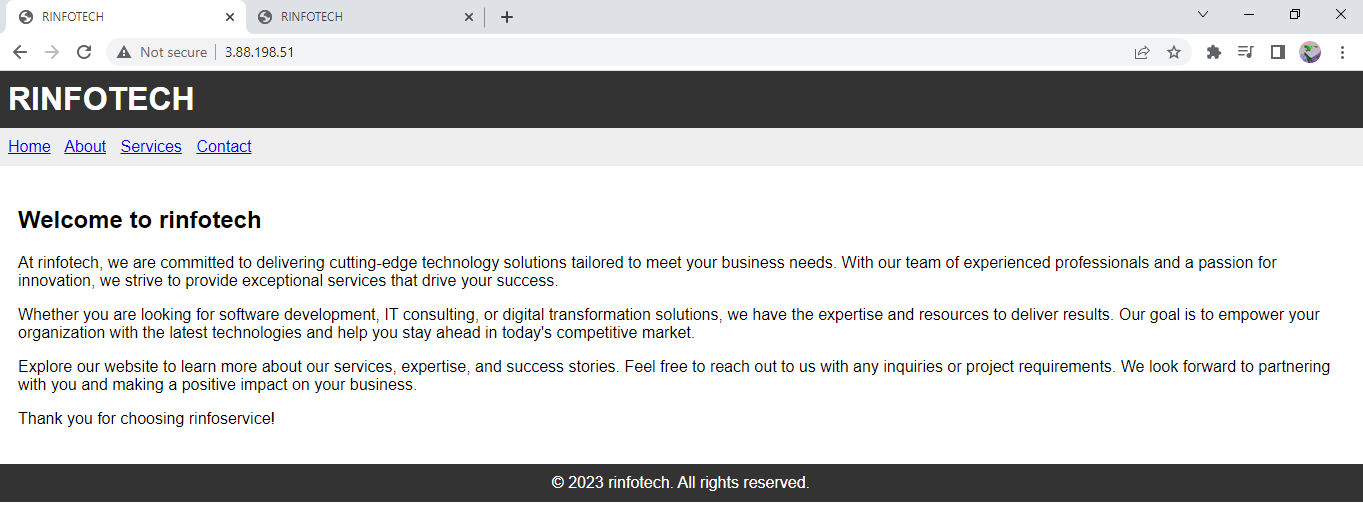


**Node2:**

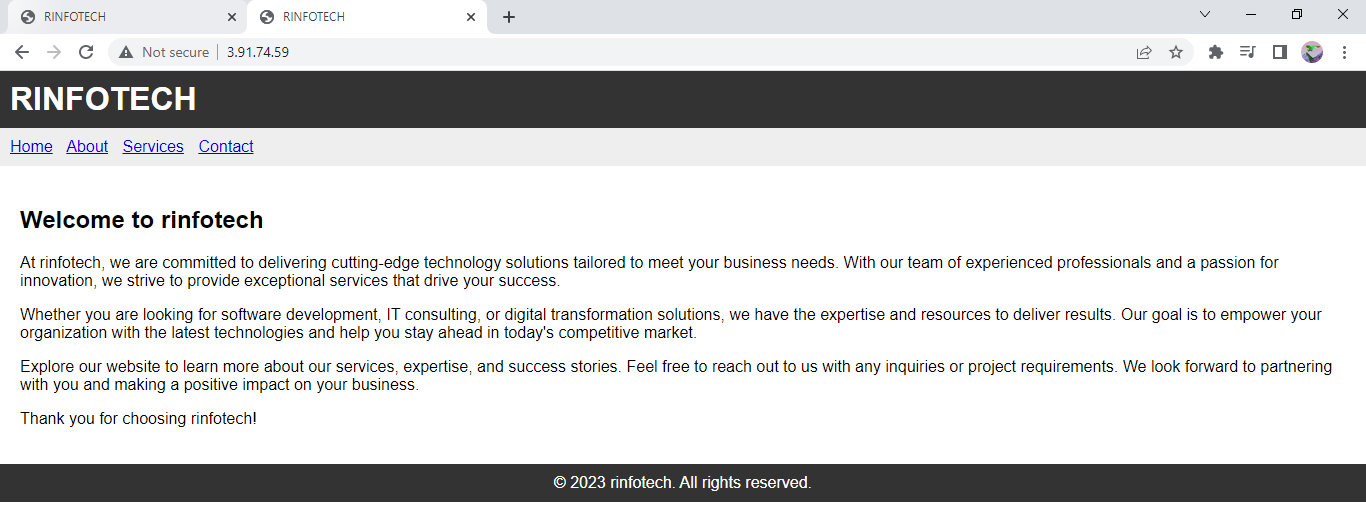


## -Go to web browser and enter the public IP

**Node1:**



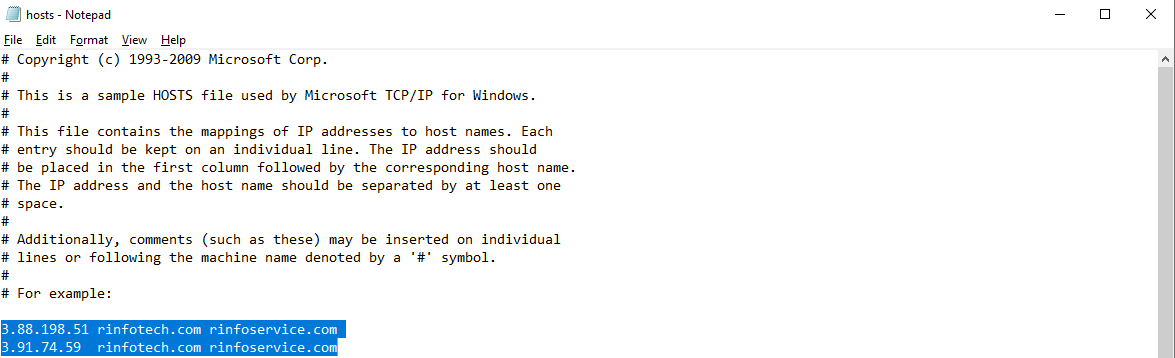
**Node2:**



# **To merge the two websites in single point on both nodes:**

**Win\_10:**

## -By using the local DNS host file:



# **To create Virtual host configuration file for each domain on Node1&2 server:**

**Node1&2:**

## -Create virtual host in conf.d file

#cd /etc/httpd/conf.d

#ls –ltr

#vi virtualhost.conf

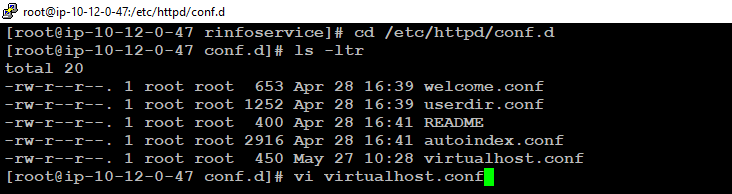
<Directory /var/www/html/infotech>  
        Require all granted  
        AllowOverride None  
</Directory>

<VirtualHost \*:80>  
        DocumentRoot /var/www/html/infotech/  
        ServerName [infotech.com](http://abc123.com/)  
          
</VirtualHost>

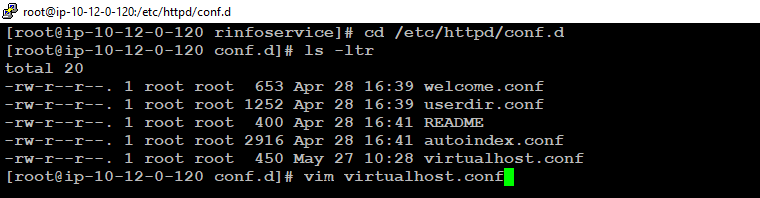
<Directory /var/www/html/infoservice>  
        Require all granted  
        AllowOverride None  
</Directory>

<VirtualHost \*:80>  
        DocumentRoot /var/www/html/infoservice/  
        ServerName [infoservice.com](http://xyz123.com/)  
          
</VirtualHost>

**Node1:**



**Node2:**



# **Assign ownership of the directories to the Apache user (usually 'apache') and grant appropriate permissions on node1&2 server:**

## -Assign ownership and permissions:

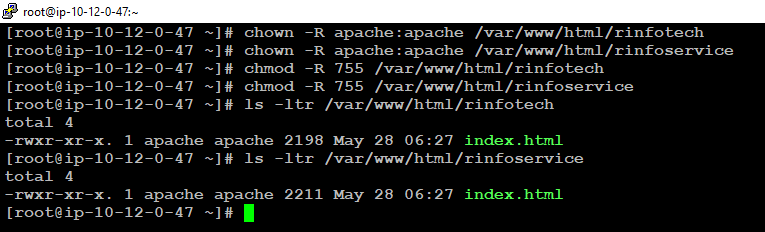
#chown -R apache:apache /var/www/html/rinfotech

#chown -R apache:apache /var/www/html/rinfoservice

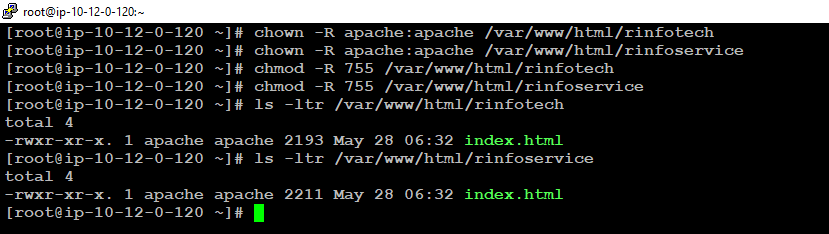
#chmod -R 755 /var/www/html/rinfotech

#chmod -R 755 /var/www/html/rinfoservice

**Node1:**



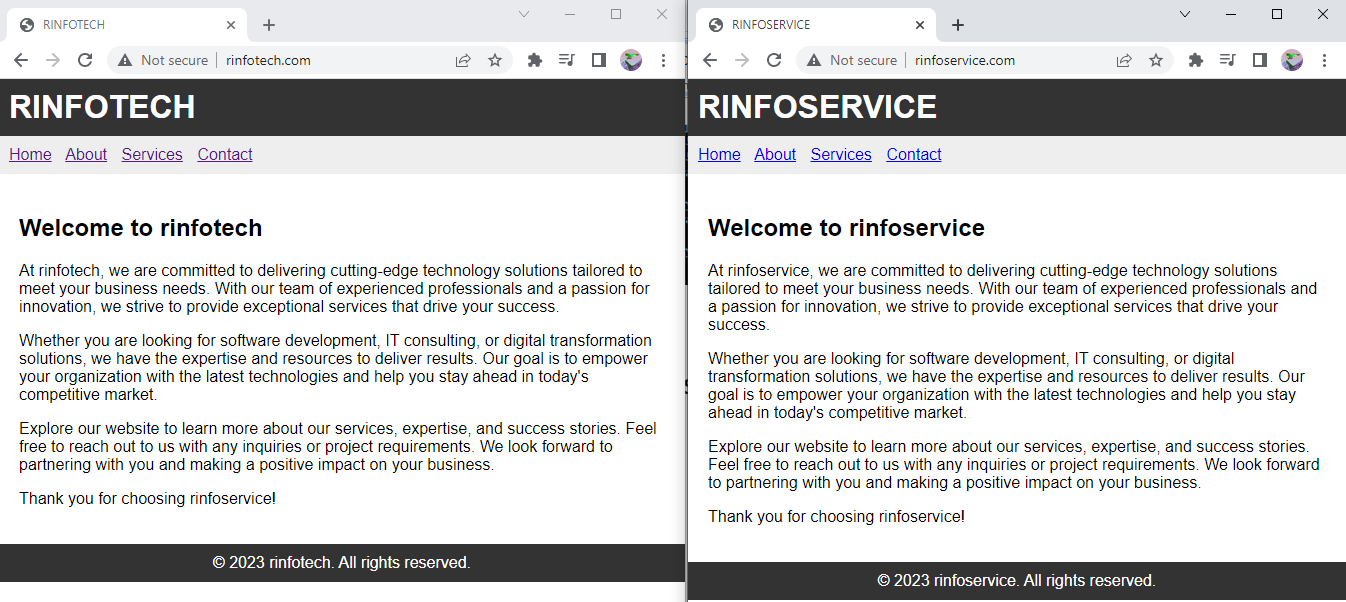
**Node2:**



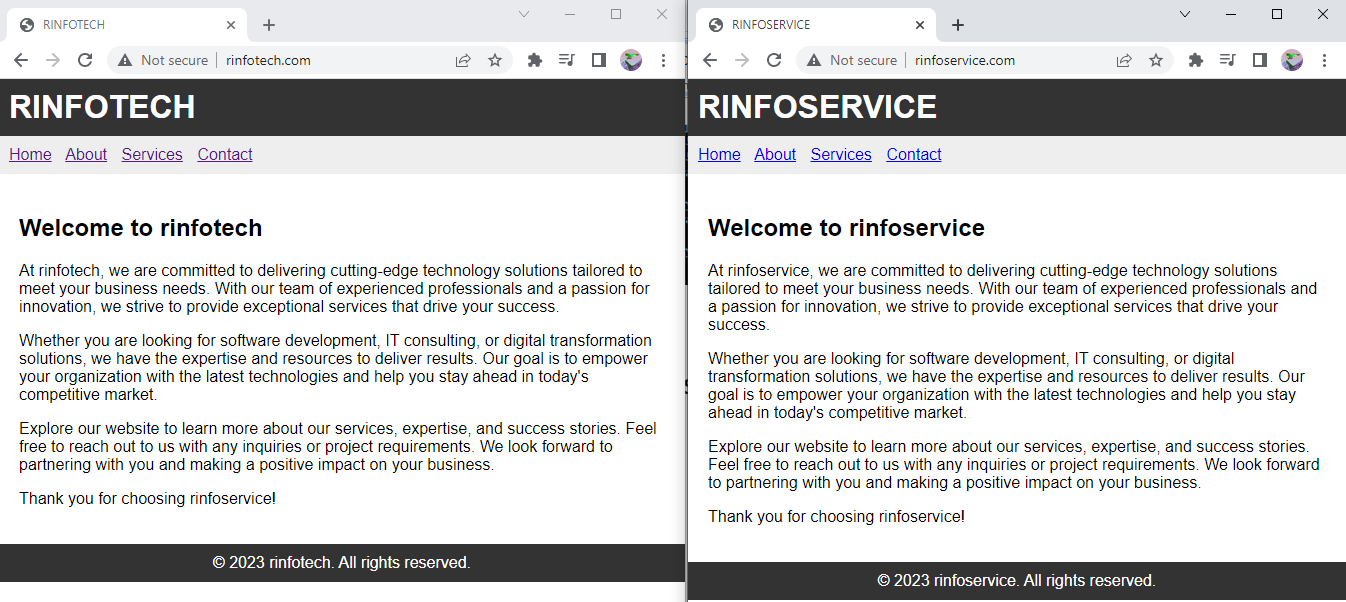
# **To verify the two websites accessible on node1&2 using local DNS name:**

## -Validate the websites:

**Node1:**



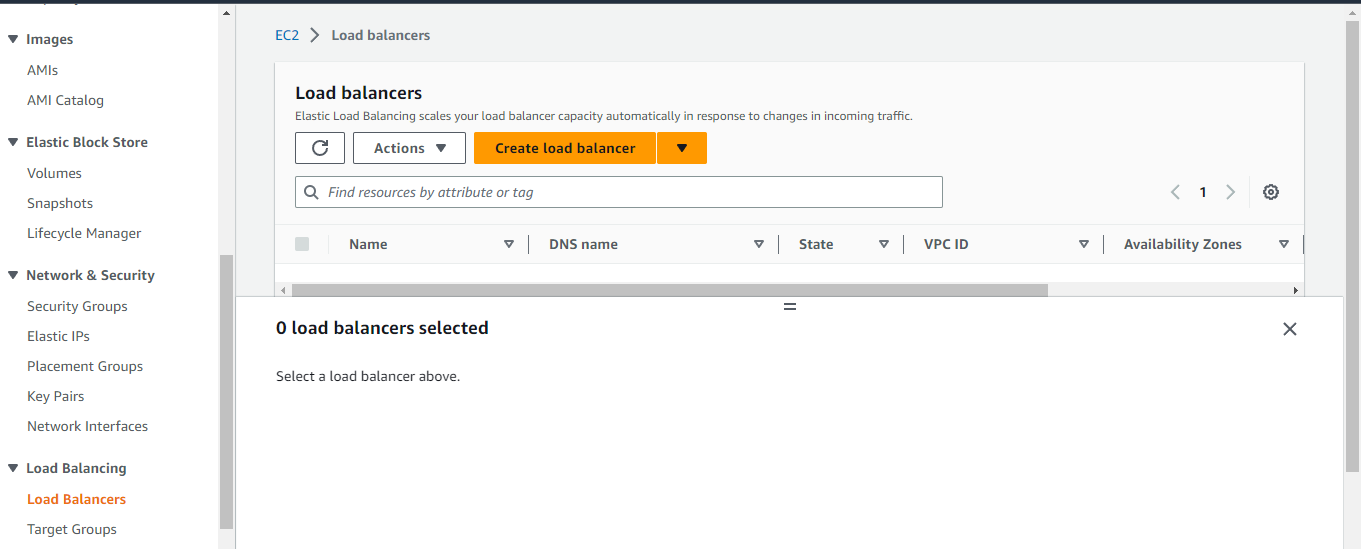
**Node2:**



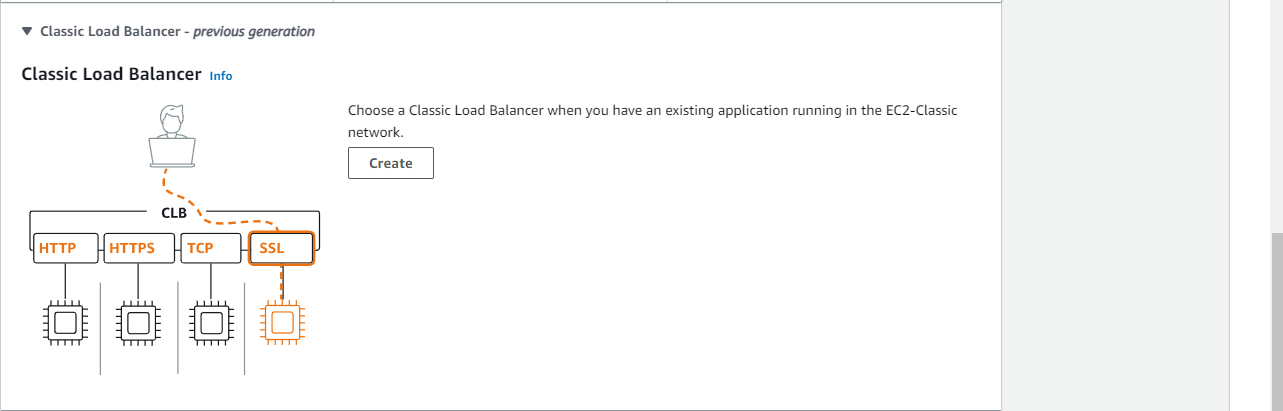
# **To Create Classic load balancer for two nodes:**

->Click on "Load Balancers" in the navigation pane.

## -Click on the "Create Load Balancer" button.



## -Select "Classic Load Balancer" from the options: Create



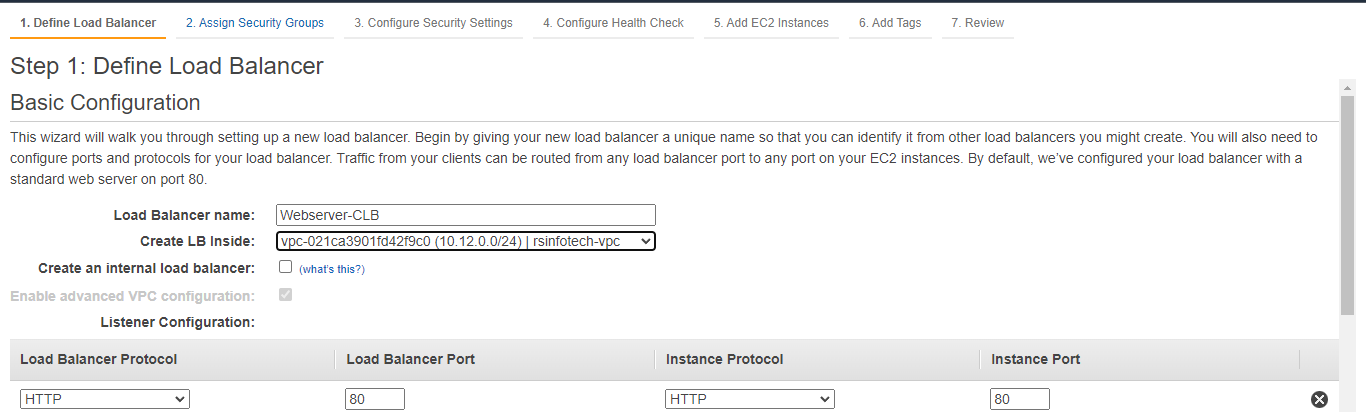
## -Configure the load balancer settings:

Give your load balancer a name: Webserver-CLB

Select the appropriate VPC (Virtual Private Cloud) for your setup.

Select "Internet-facing" or "Internal" based on your requirements.

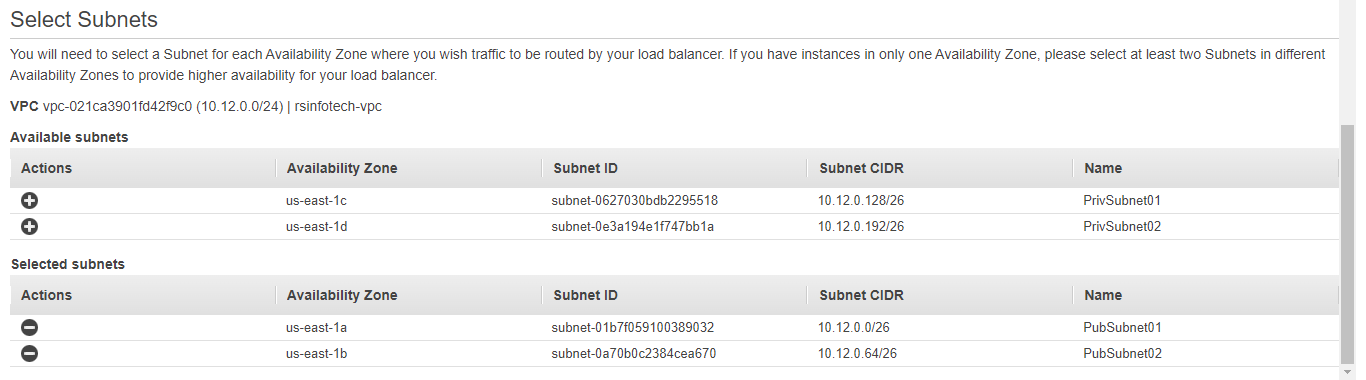
Specify the listeners for your load balancer (e.g., HTTP on port 80).



## -Configure the availability zones:

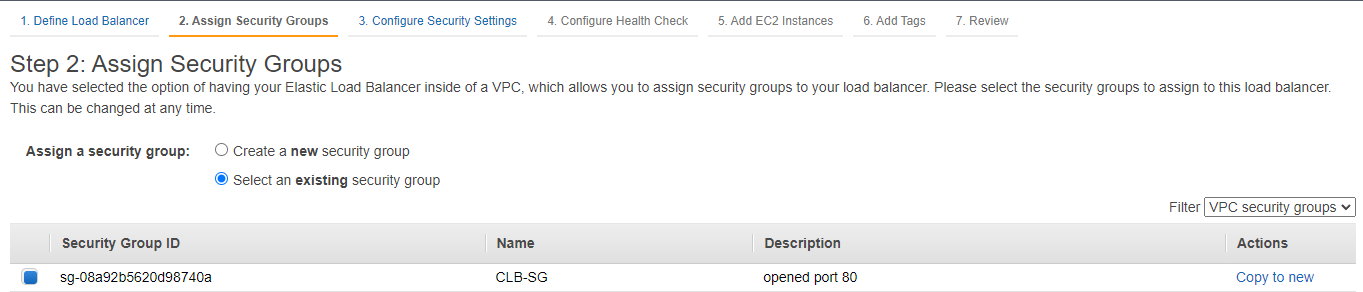
Choose the availability zones where your application instances are located.

Enable or disable cross-zone load balancing based on your needs.



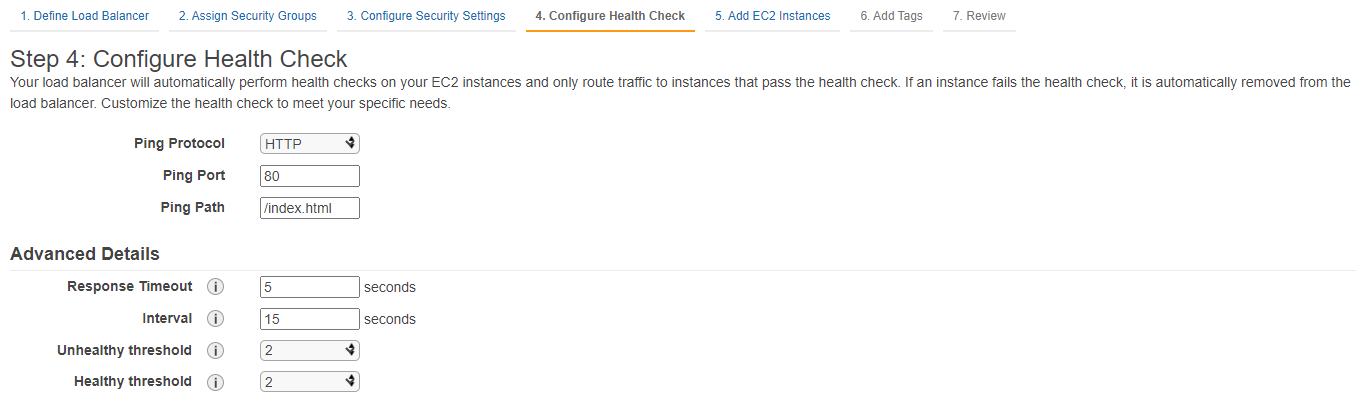
## -Configure security settings:

Choose an existing security group or create a new one to control inbound traffic to your load balancer.



## -Configure health checks:

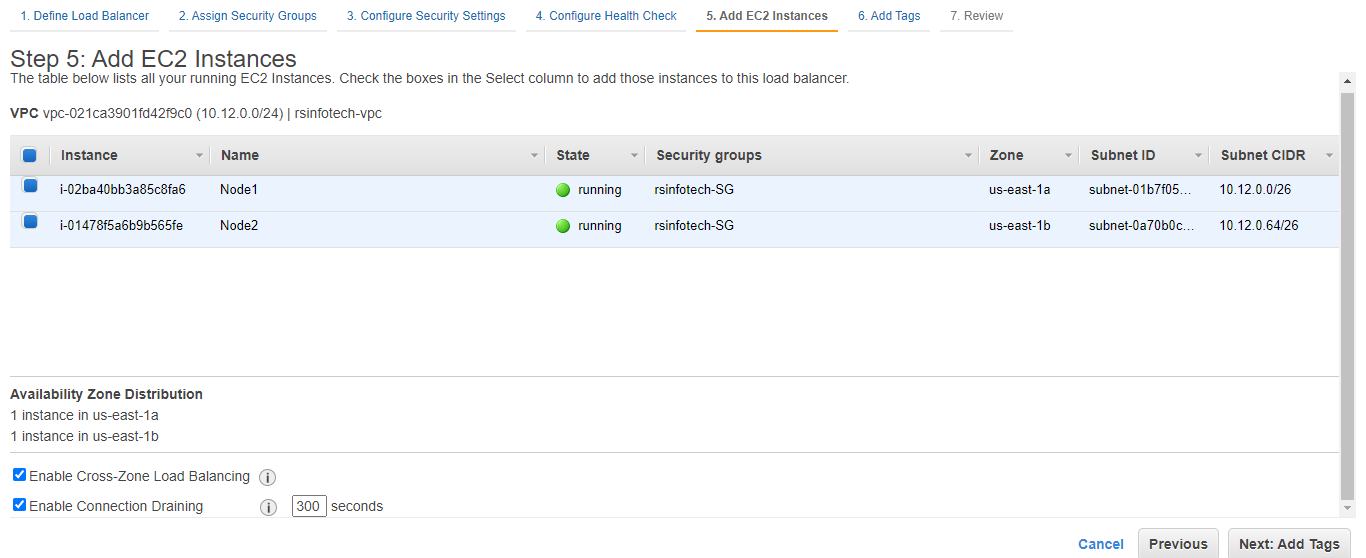
Define the protocol, port, and endpoint for health checks to determine the status of your application instances.



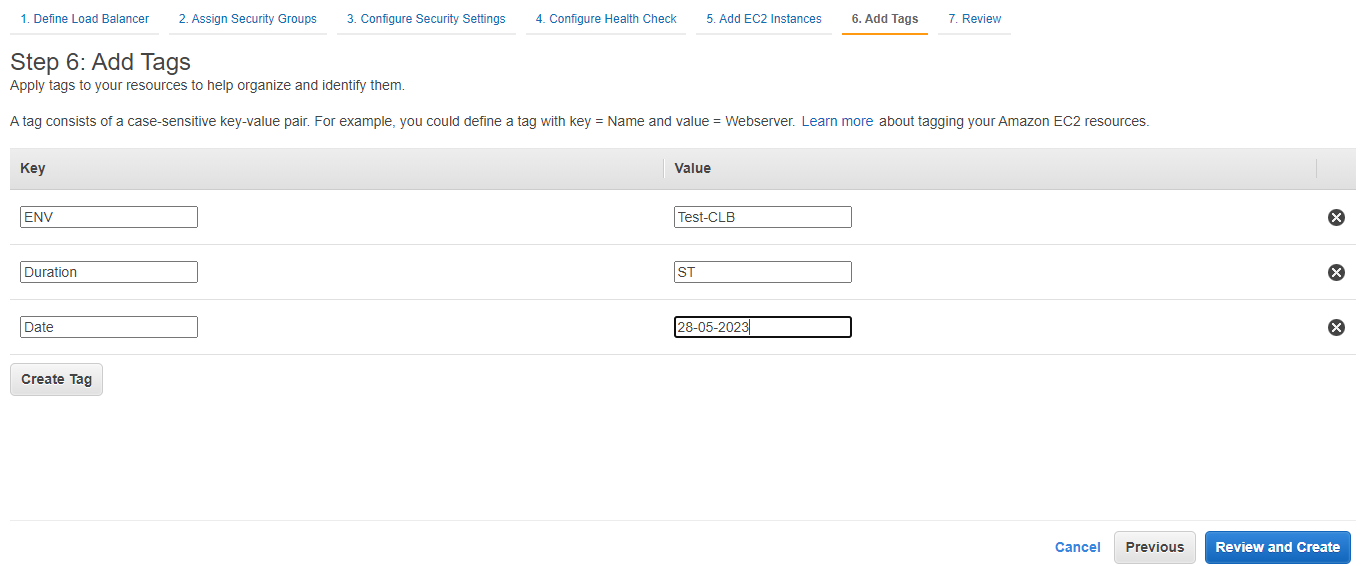
## -Add EC2 application instances:

Select the instances that you want to register with the load balancer.

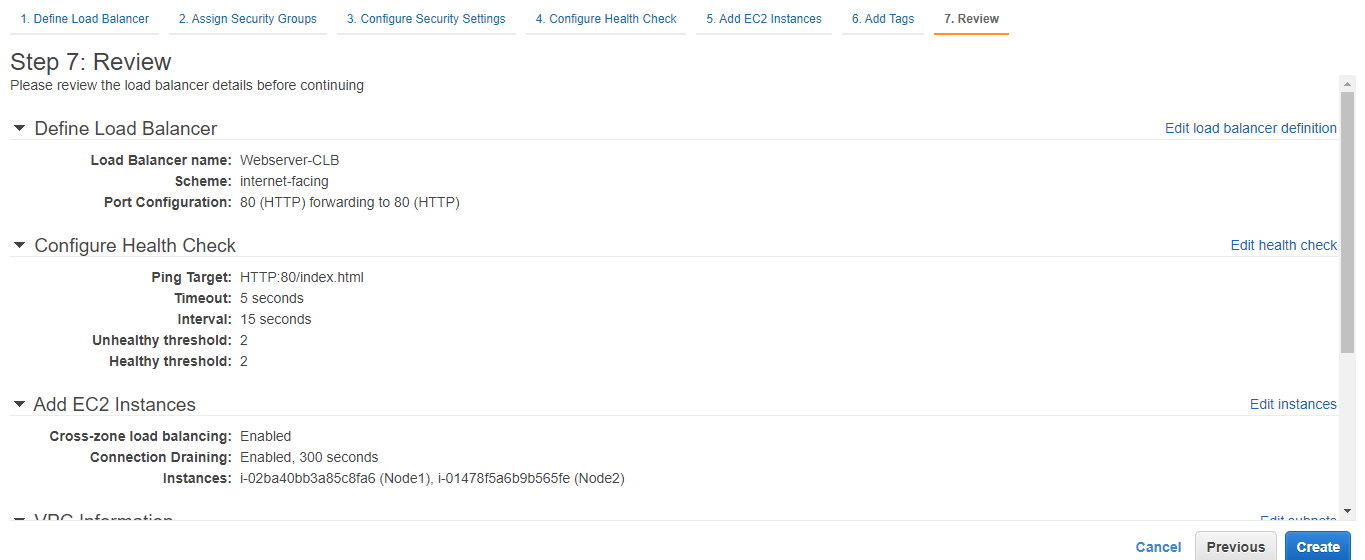
Specify the ports on which your application listens.



## -Add Tags for CLB:

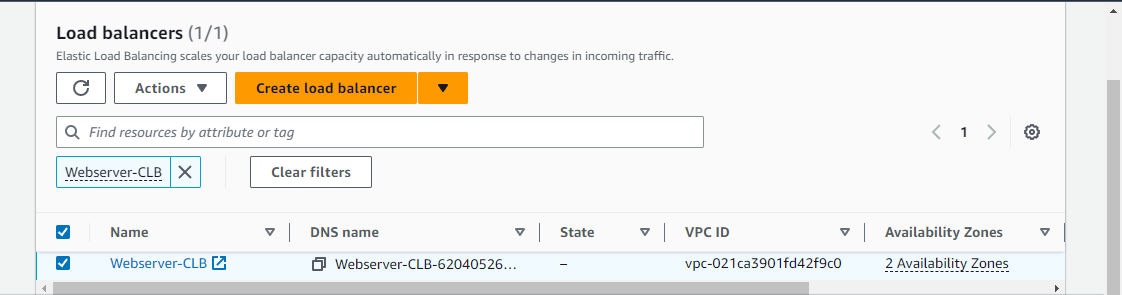


## -Review the load balancer configuration and click on "Create" to create the load balancer.



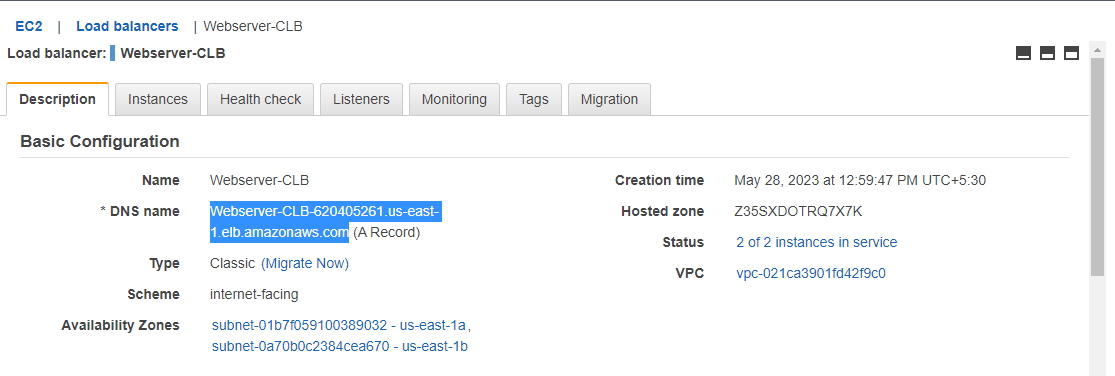
## -Successfully created load balancer

Wait for the load balancer to be provisioned. Once it is ready, you will see its status as "Active."



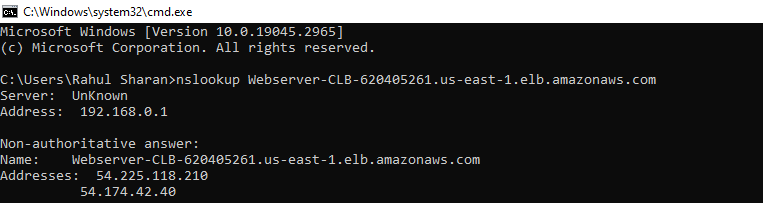
## -Obtain the DNS name of the load balancer from the EC2 dashboard.

-LB Dashboard-> Description->DNS Name

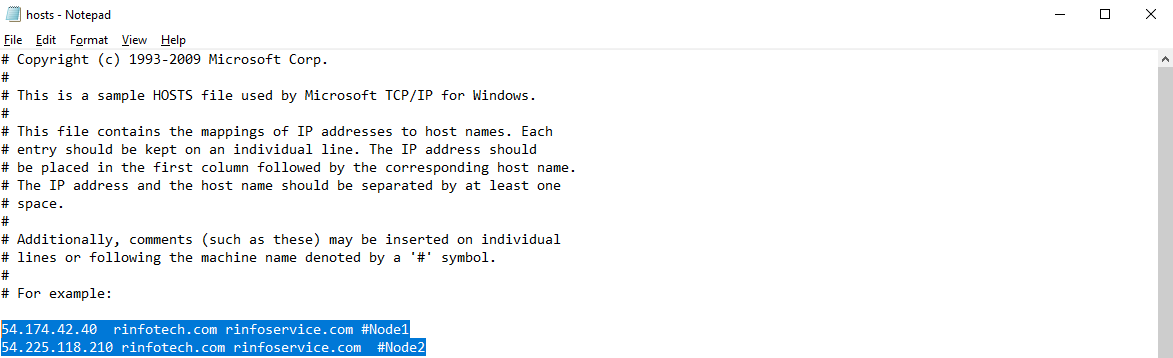


## -Check the LB Ips using DNS name:

Copy the DNS name and go local system win\_10

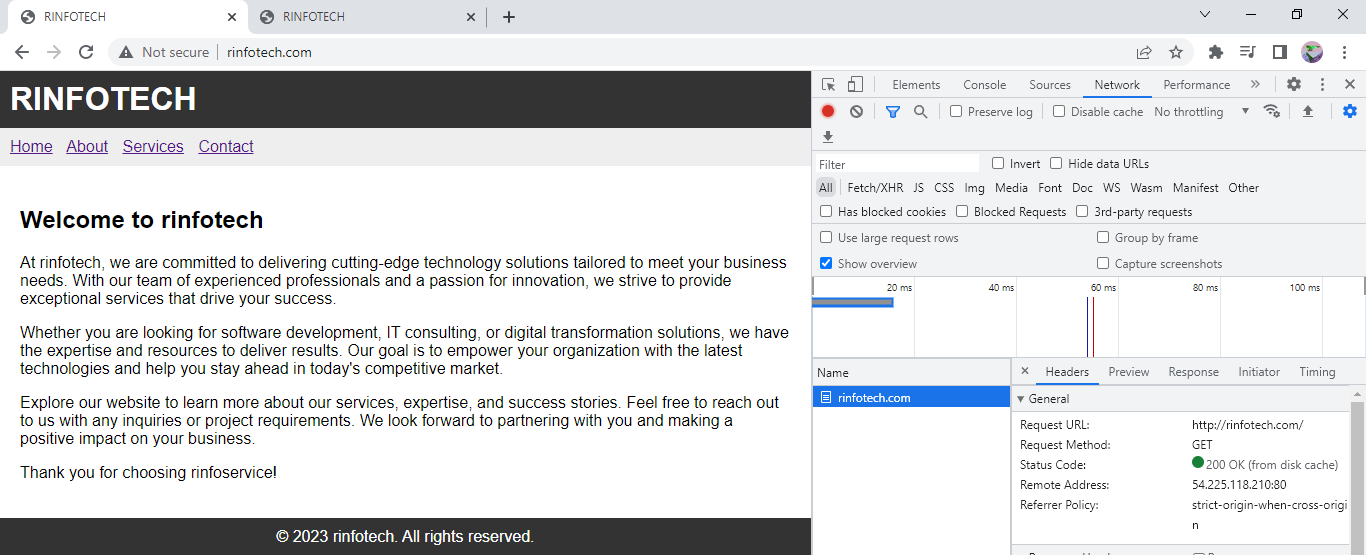


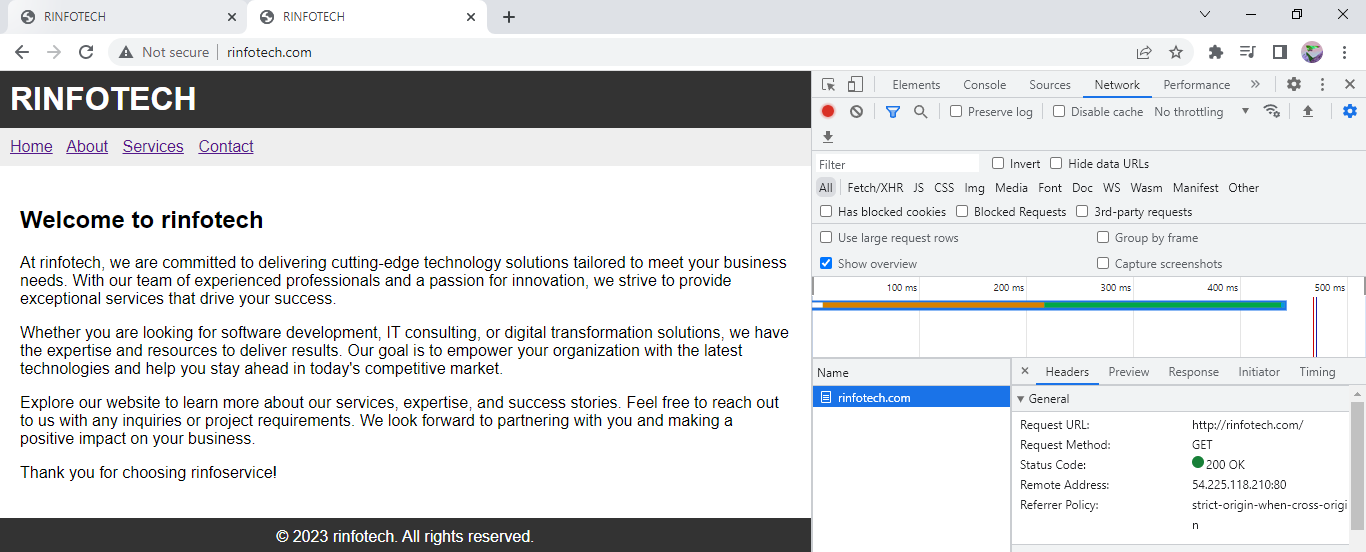
## -Update the host file:



# **To verify the traffic on web browser on node1&2 using local DNS name:**

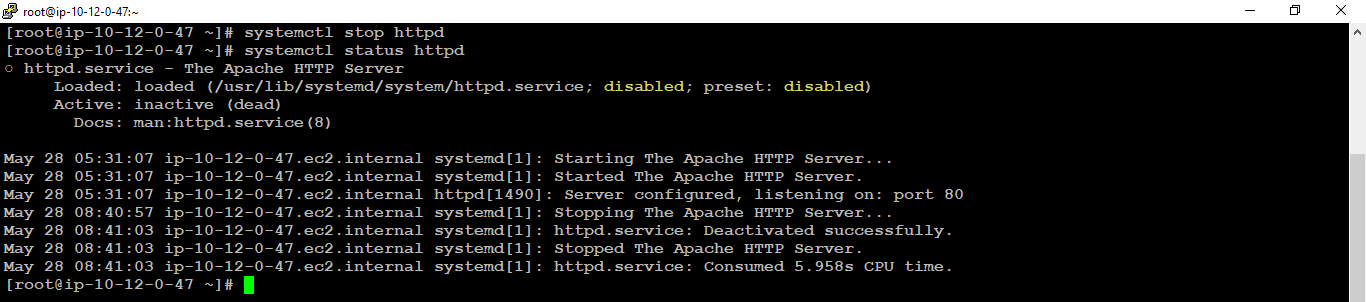
**Node1&2:**





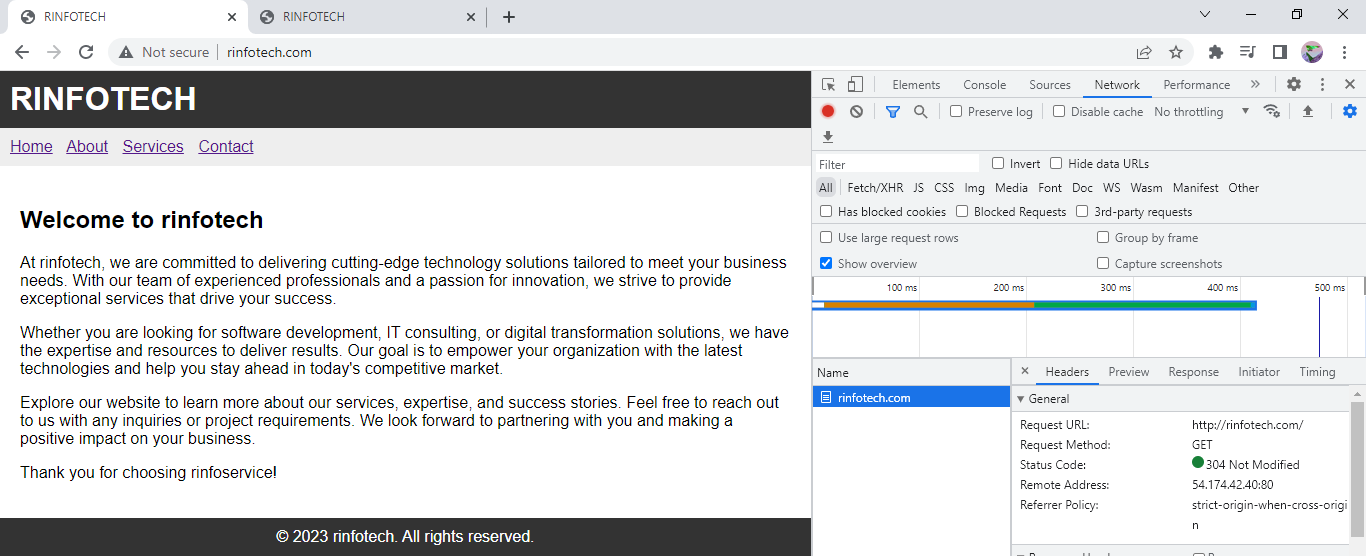
## -Stop the **Node1** and check the both websites traffic:

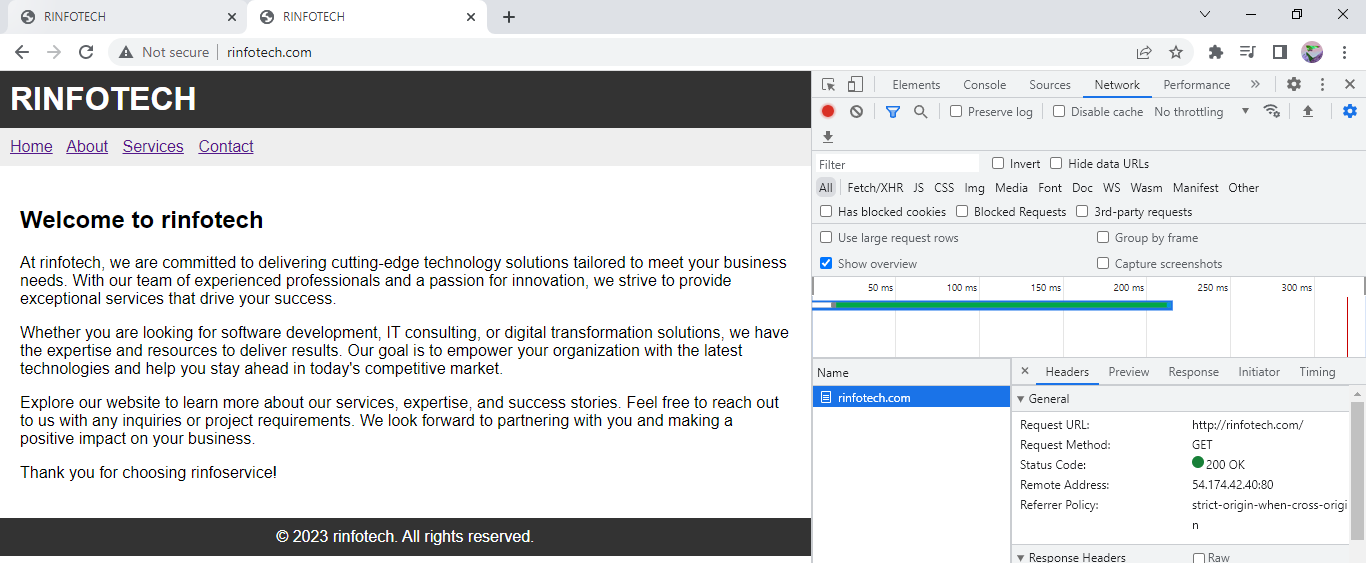
#systemctl stop httpd



## -Verify the traffic on web browser:

**node1&2:**

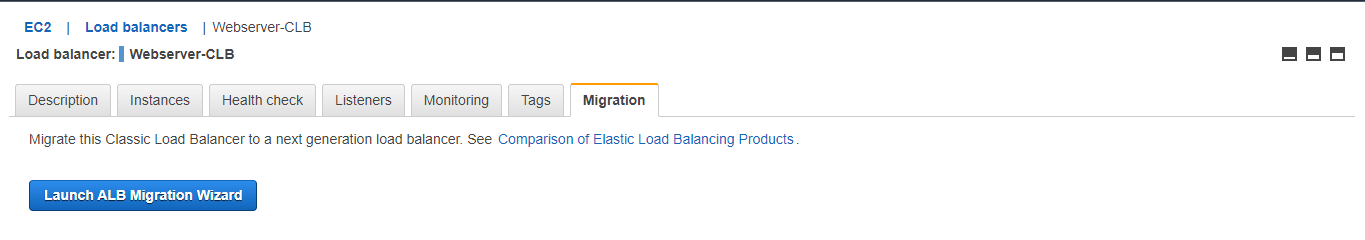


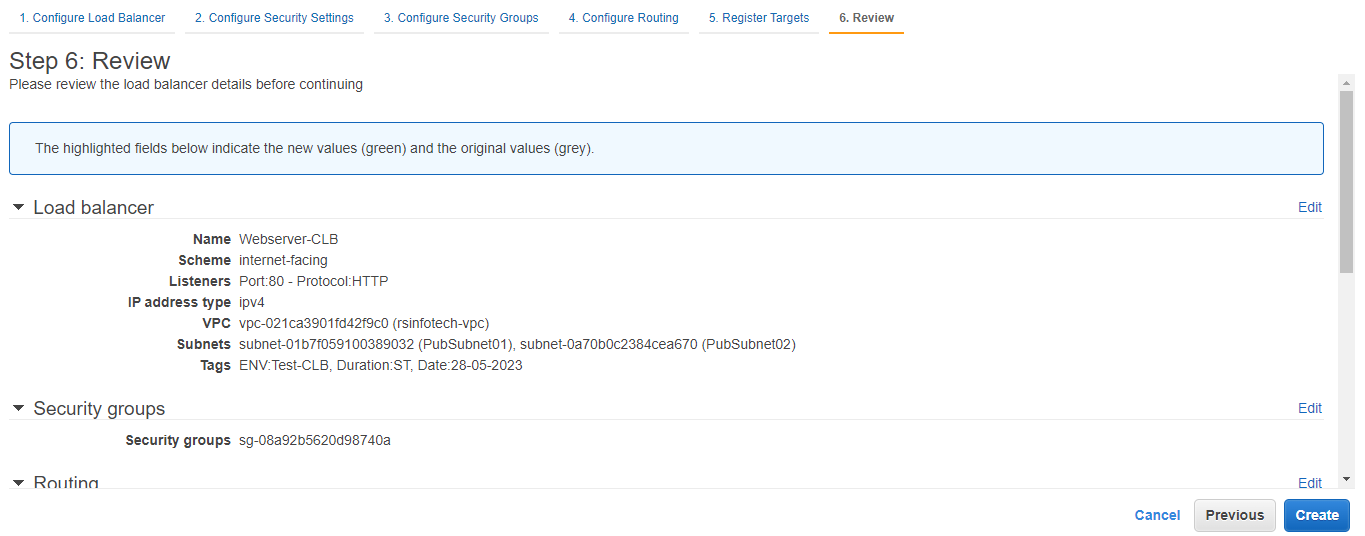


# **To Migrate the Classic load balancer to Application load balancer:**

## **-Migrate CLB to ALB:**

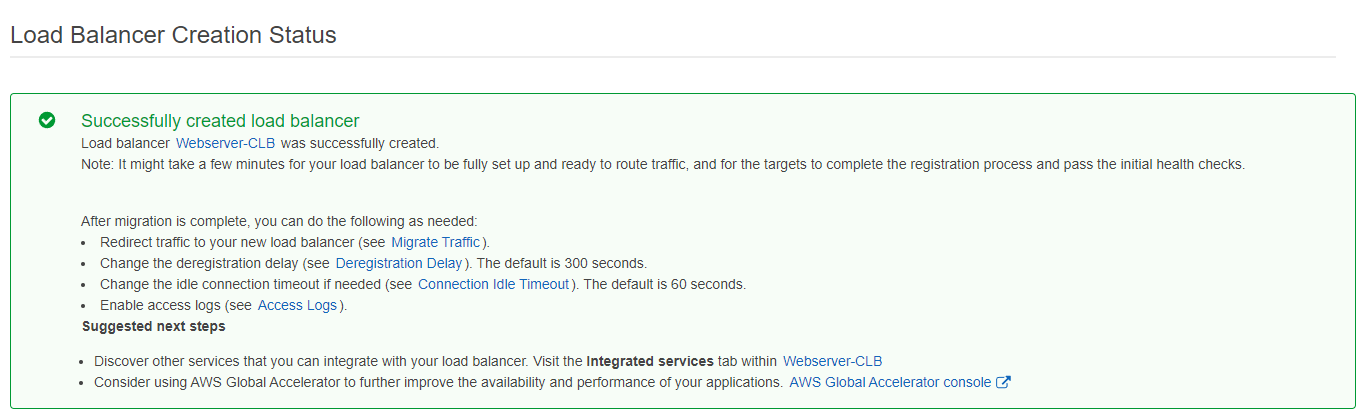
->Loadbalancer->CLB->Migrate->Launch->Review and Create

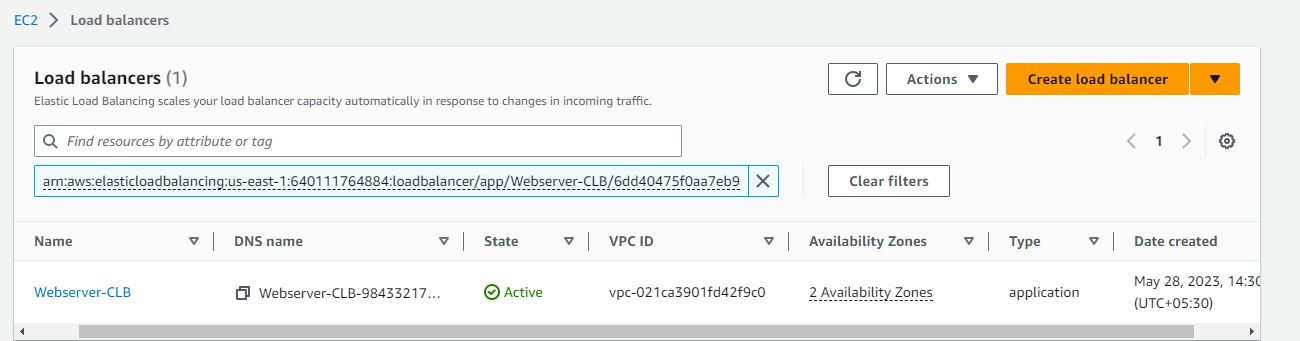




## **-Successfully created load balancer:**

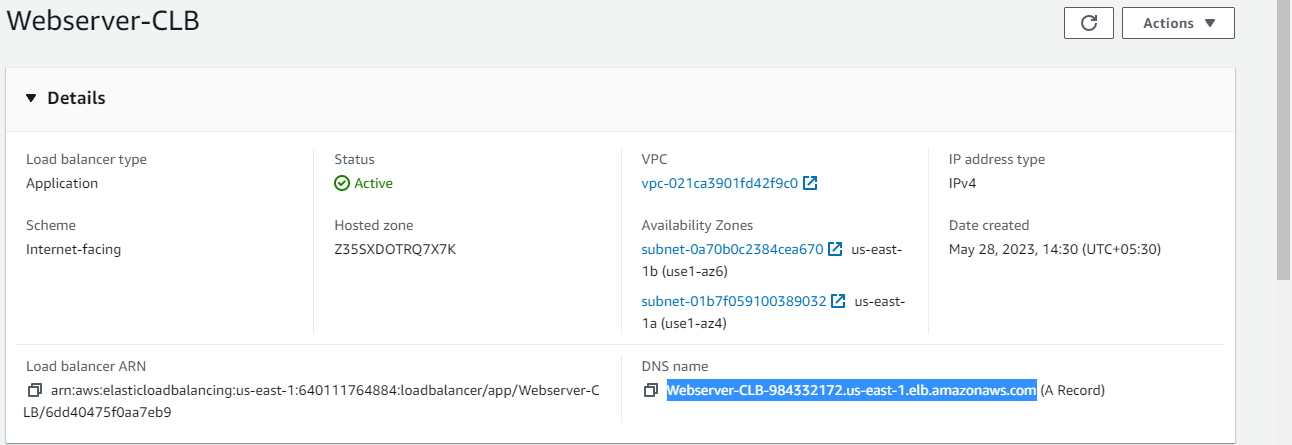
Wait for the load balancer to be provisioned. Once it is ready, you will see its status as "Active." and “Application load balancer”





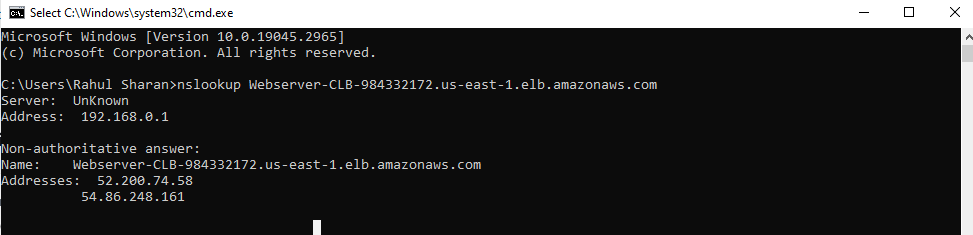
## -Obtain the DNS name of the load balancer from the EC2 dashboard.

-LB Dashboard-> Description->DNS Name

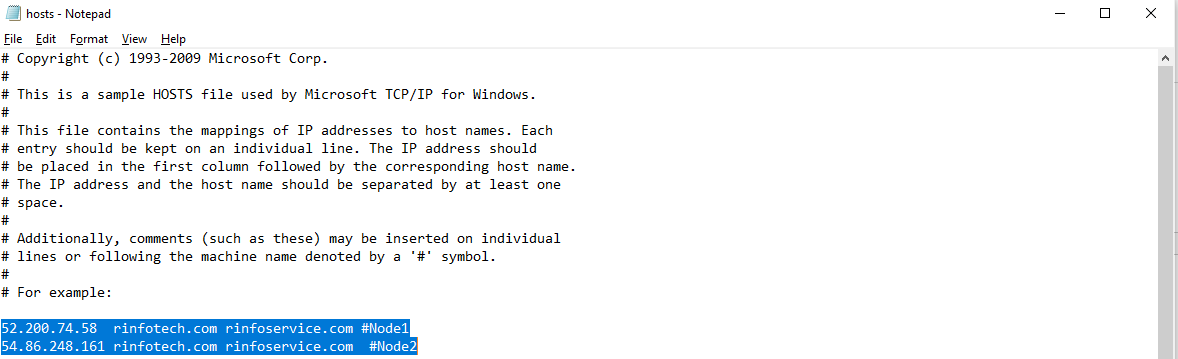


## -Check the LB Ips using DNS name:

Copy the DNS name and go local system win\_10

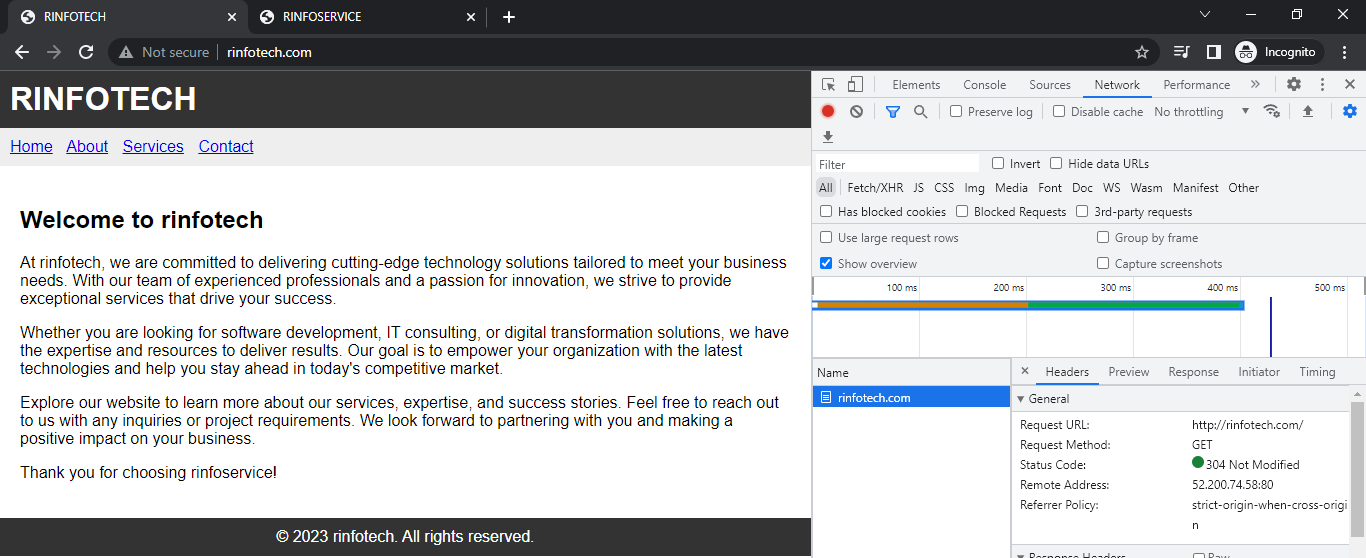


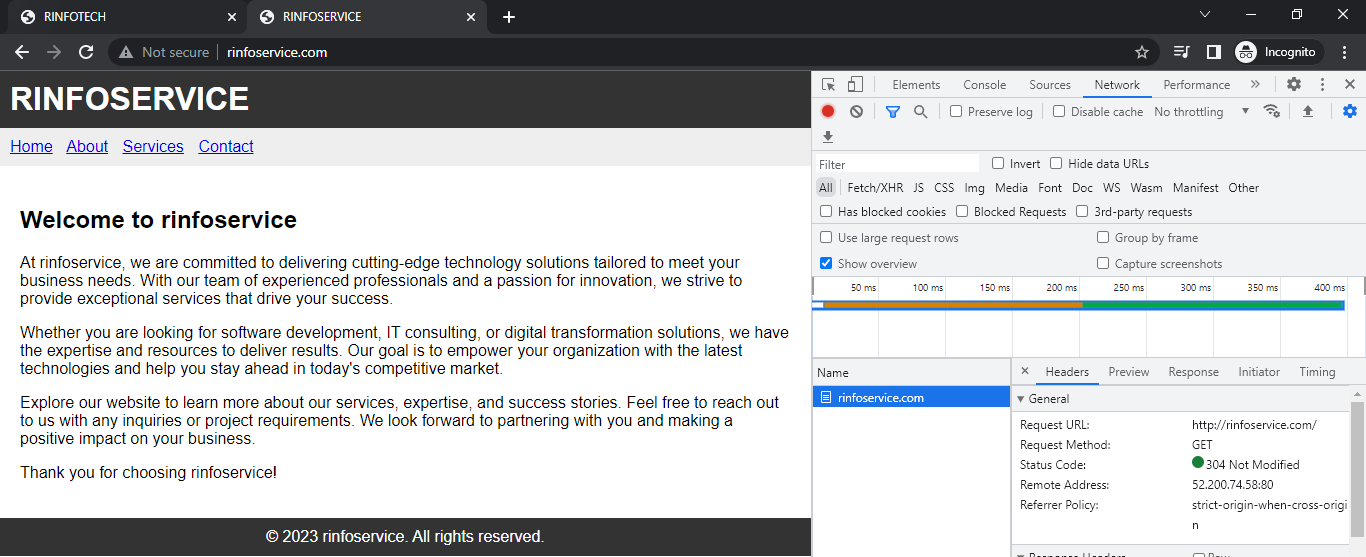
## -Update the host file:



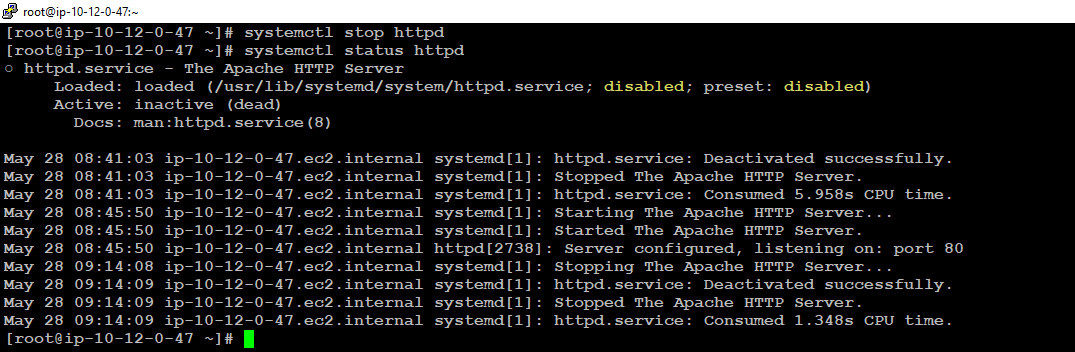
# **To verify the traffic on web browser on node1&2 using local DNS name:**

**Node1&2:**





## -Stop the **Node1** and check the both websites traffic:

#systemctl stop httpd 

## -Verify the traffic on web browser:

**node1&2:**

