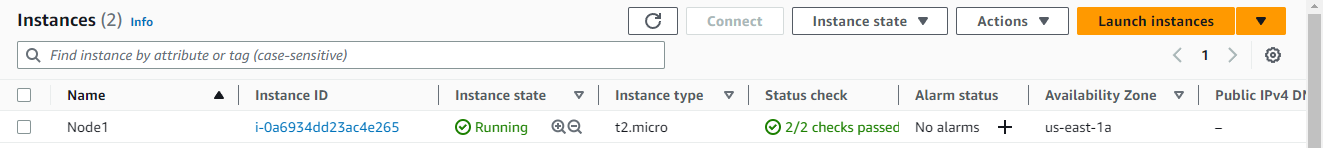
**Project C: 4**

# Secure web application from Hacker

## To create EC2 two instance:

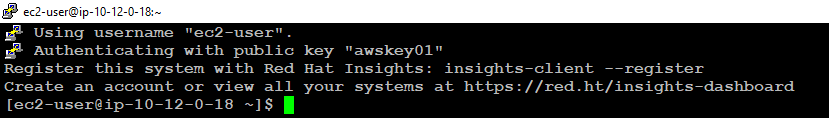
### - Create EC2 instance (Redhat server):



## Connect to EC2 instance using putty software:

### -Connect EC2: Using the IP address, username and keypair:

Node1:



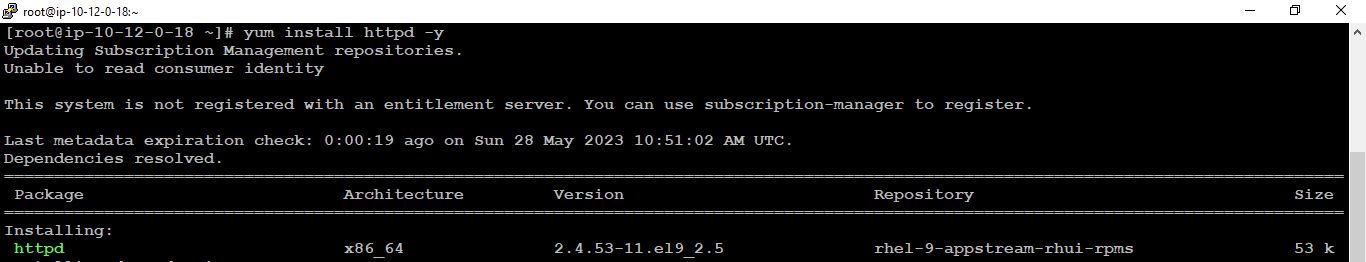
## **To configure the Apache webserver using the EC2 instance:**

### -Install the apache package and updates using the following commands:

$sudo su –

#yum install httpd –y

Node1:

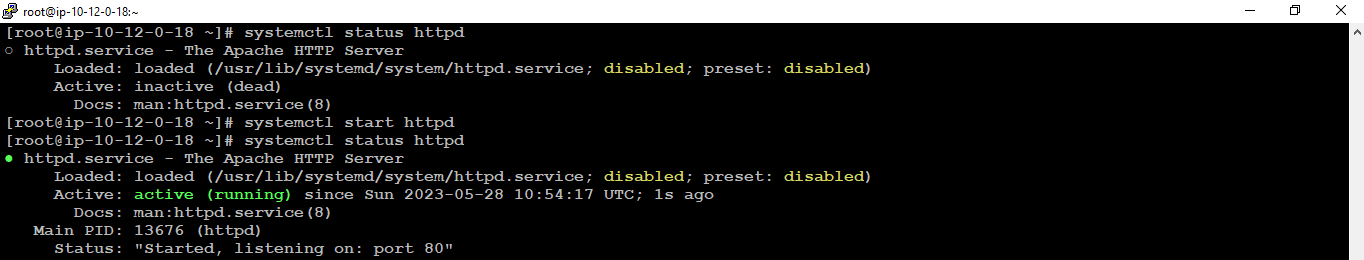


### To check and start the apache service using the following commands:

#systemctl status httpd

#systemctl start httpd

Node1:



### -Configure the document root of apache using the following path:

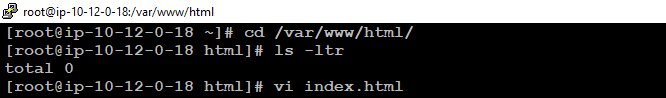
#cd /var/www/html/

#ls –ltr

#vim index.html

Insert: Paste the .html file and save using esc: wq!

Node1:



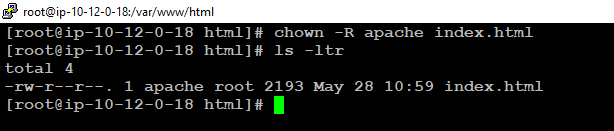
### -Change the index.html file owner as an apache using the following command:

#ls –ltr

#chown -R apache index.html

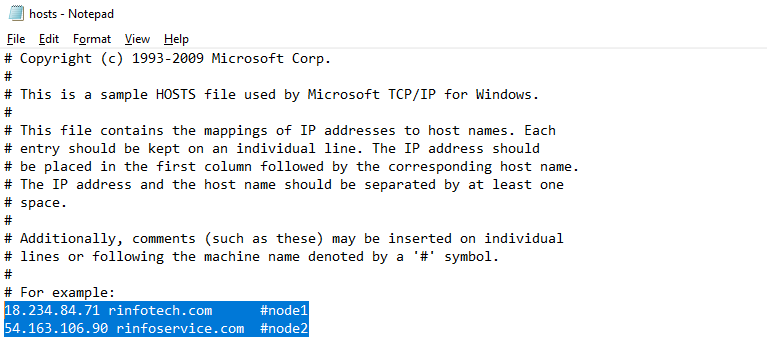
#ls –ltr

Node1:



### -Local DNS configuration on physical machine windows\_10 using the following path:

C:\Windows\System32\Drivers\etc\hosts

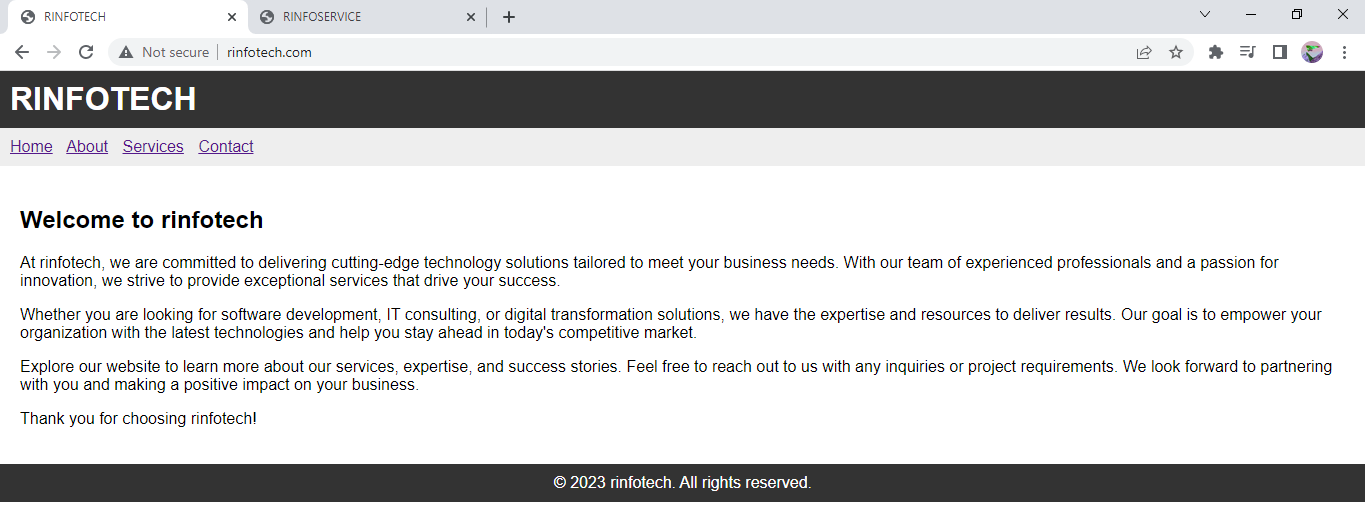


## **To validate the website using Google chrome web browser:**

### -Open the web-browser and enter the: Hostname

Node1:

<http://rinfotech.com/>

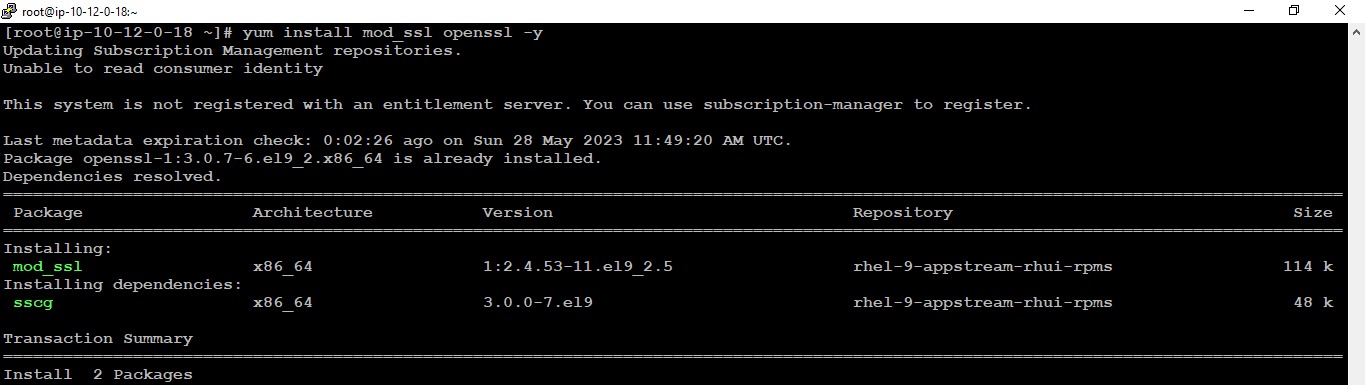


## **To create a self-signed certificate SSL Certificate for securing a multi-node web application:**

Node1:

### - Install mod and openssl:

# yum install mod\_ssl openssl -y



### -Create a private key for the SSL certificate

#cd /etc/httpd/

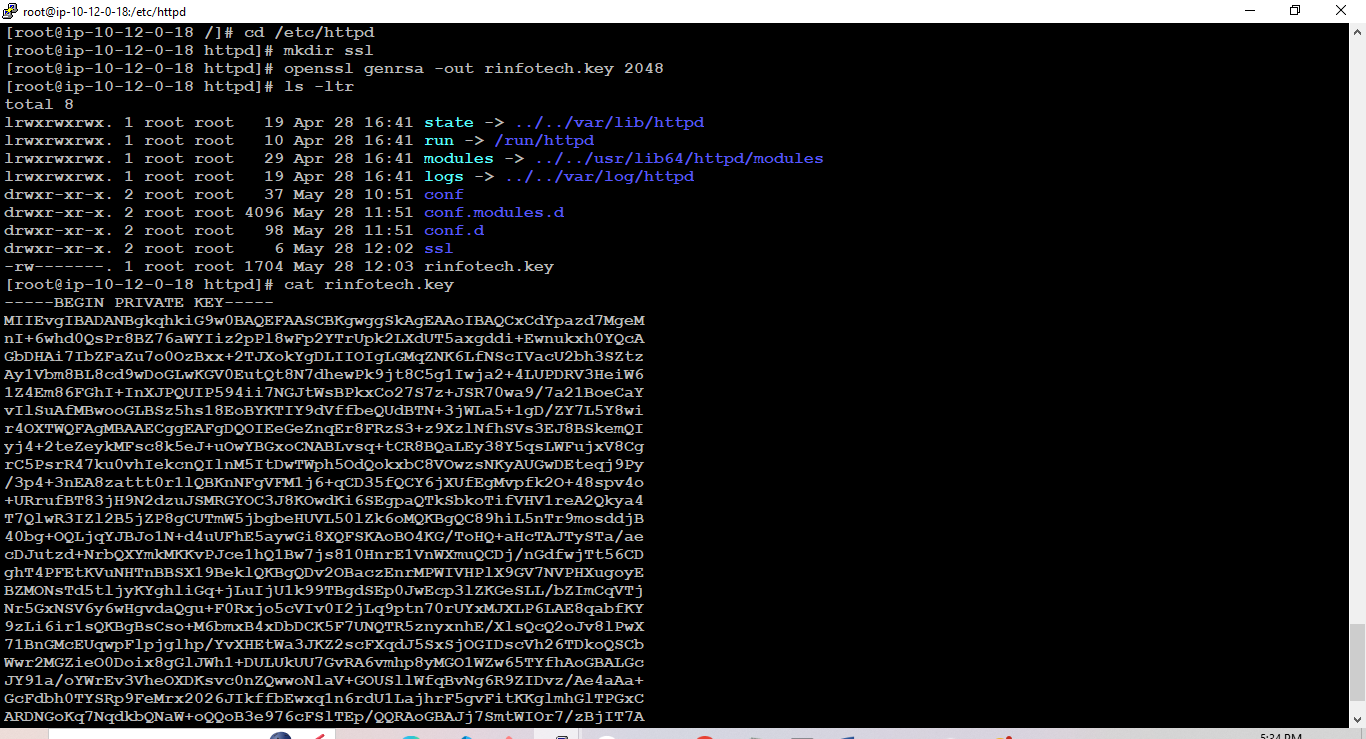
#mkdir ssl

#cd ssl

# openssl genrsa -out rinfotech.key 2048

#ls –ltr

#cat rinfotech

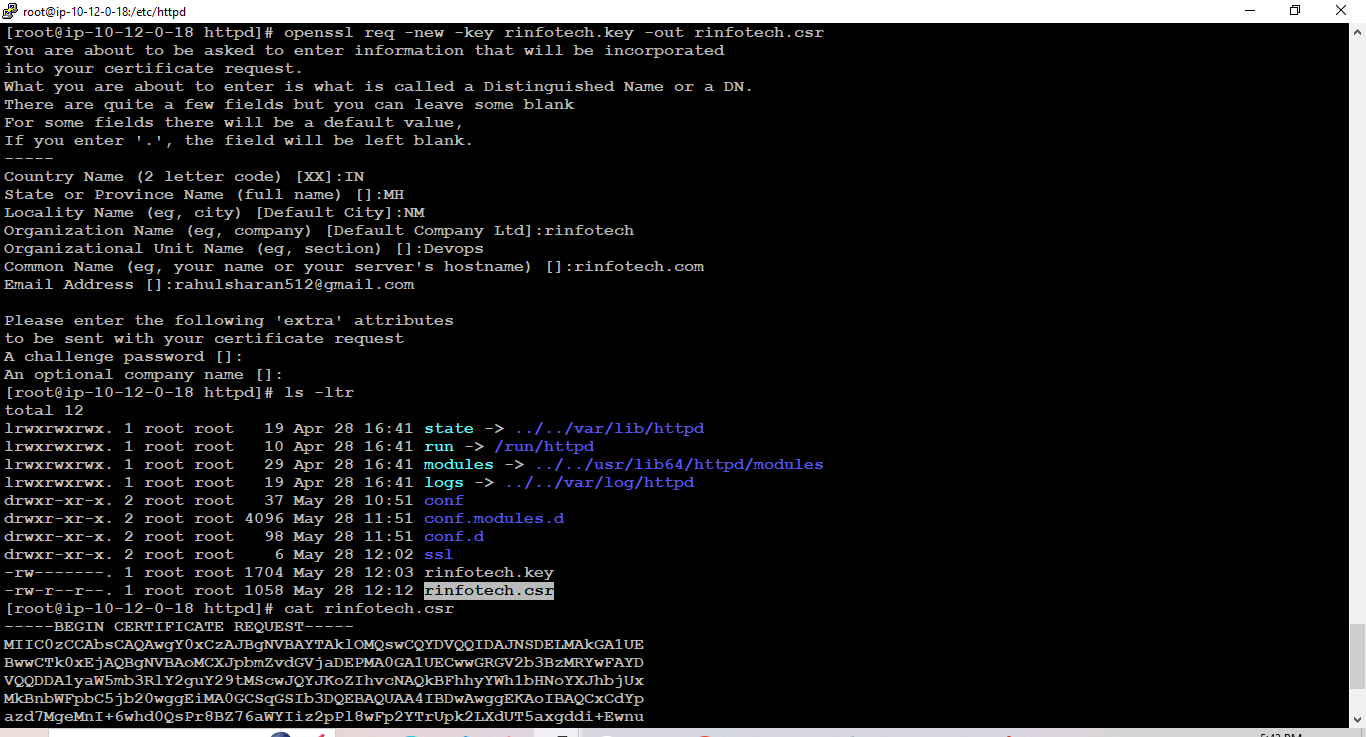


### -Generate a Certificate Signing Request (CSR):

# openssl req -new -key rinfotech.key -out rinfotech.csr

#ls –ltr

#cat rinfotech.csr



### -Generate the Self-Signed SSL Certificate using the CSR and Private Key:

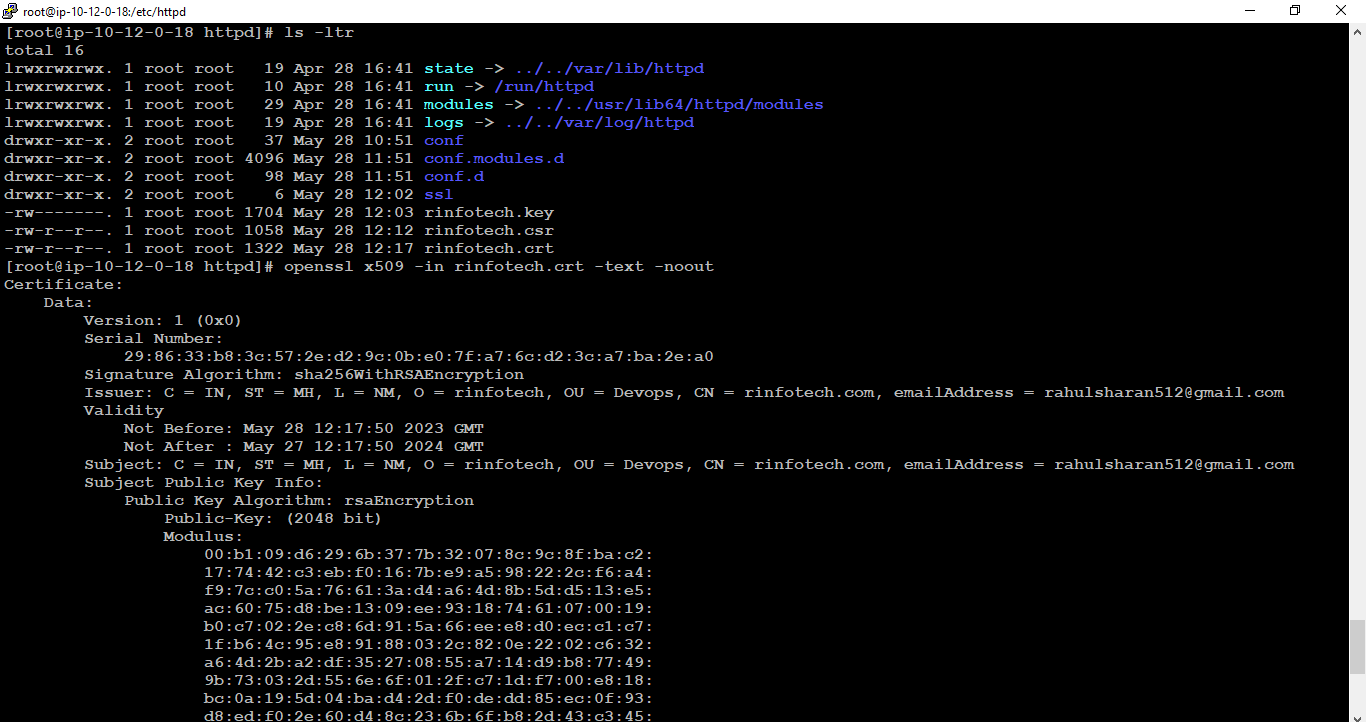
# openssl x509 -req -days 365 -in rinfotech.csr -signkey rinfotech.key -out rinfotech.crt

#ls –ltr



### -Validate the certificate:

# openssl x509 -in rinfotech.crt -text -noout



### -Edit the ssl.conf:

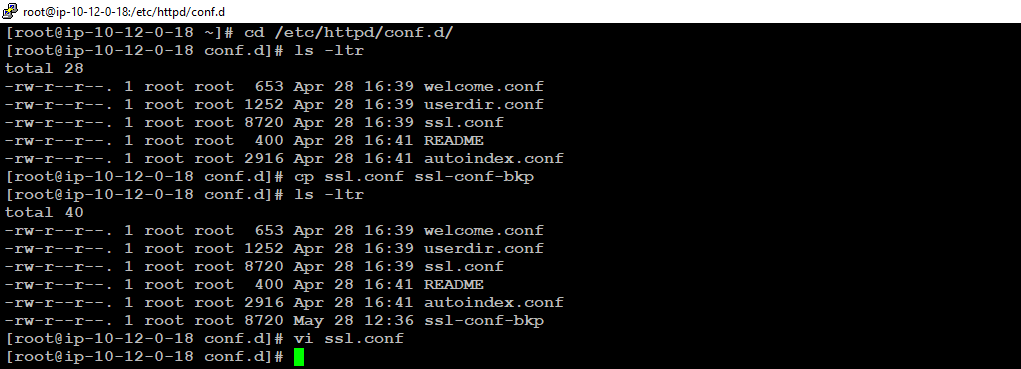
#cd /etc/httpd/conf.d

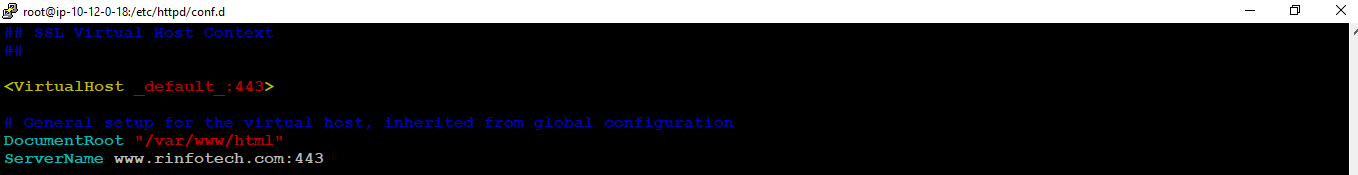
#ls –ltr

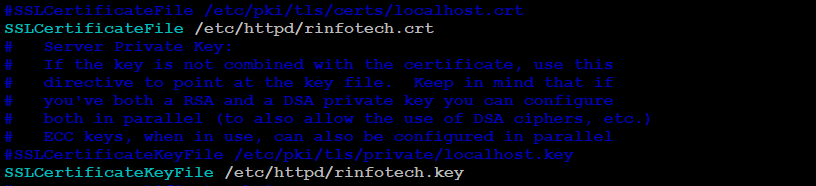
#cp ssl.conf ssl-conf-bkp

#ls -ltr

#vim ssl.conf





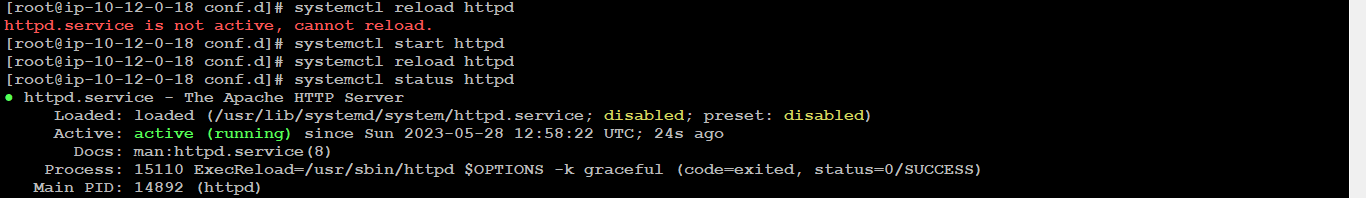


### -Reload the apache service:

#Systemctl reload httpd

# Systemctl start httpd

# Systemctl status httpd



### -Upload key and crt in **AWS ACM**:

->ACM-> Import certificate:

Certificate body: rinfotech.crt

Certificate private key:rinfotech.key

Certificate chain : rinfotech.crt

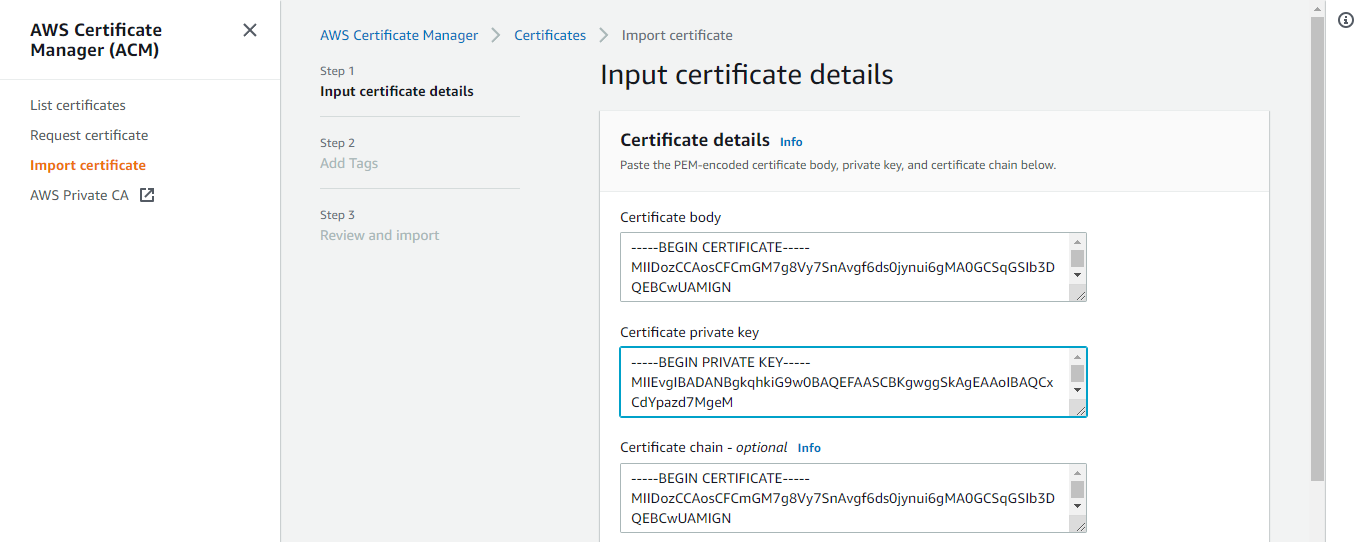
Node1:

#cd /etc/httpd/

#ls –ltr

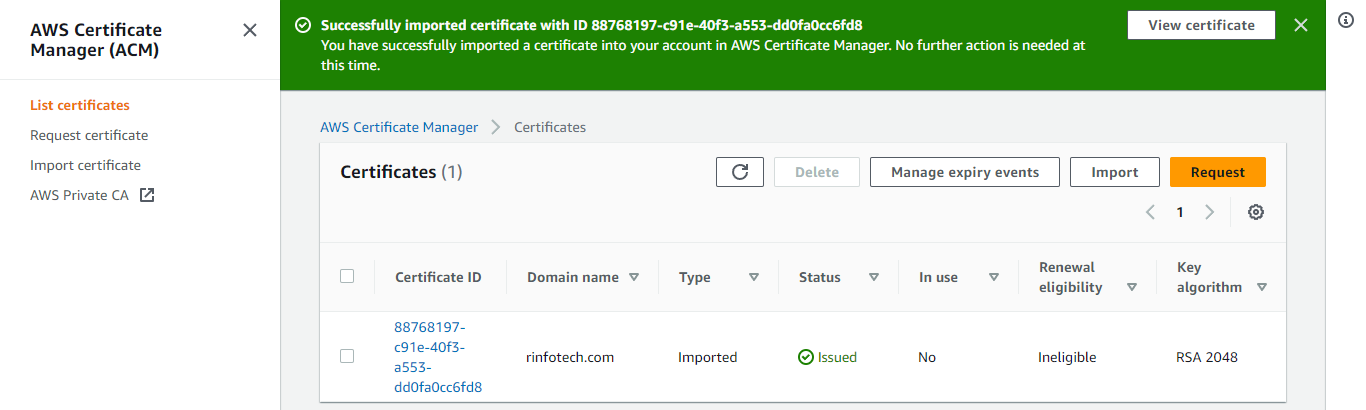
#cat rinfotech.crt

#cat rinfotech.key



### -Successfully imported certificate:

->NeXT: Tag: Name rinfotech->Next->Import

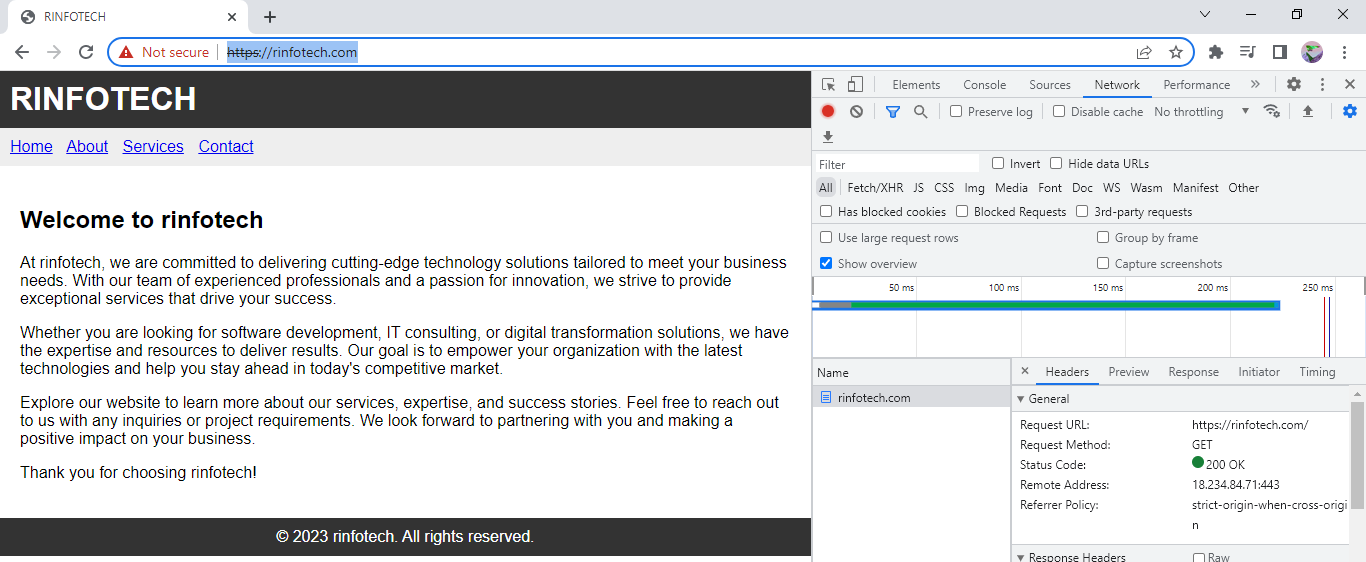


## **To validate the securing access of a website:**

Node1:

### -Open the web-browser and enter the:

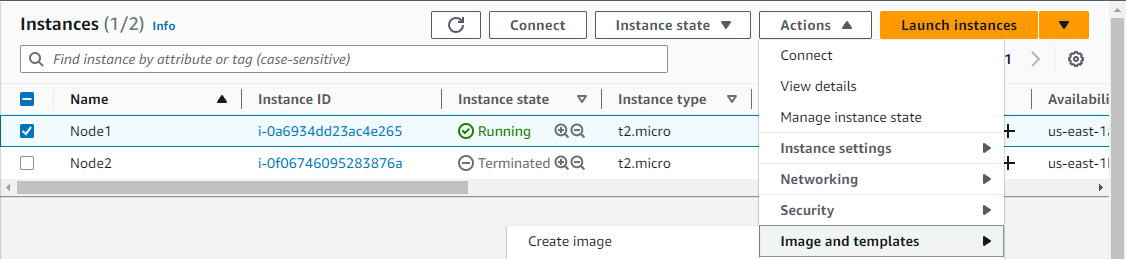
<https://rinfotech.com/>

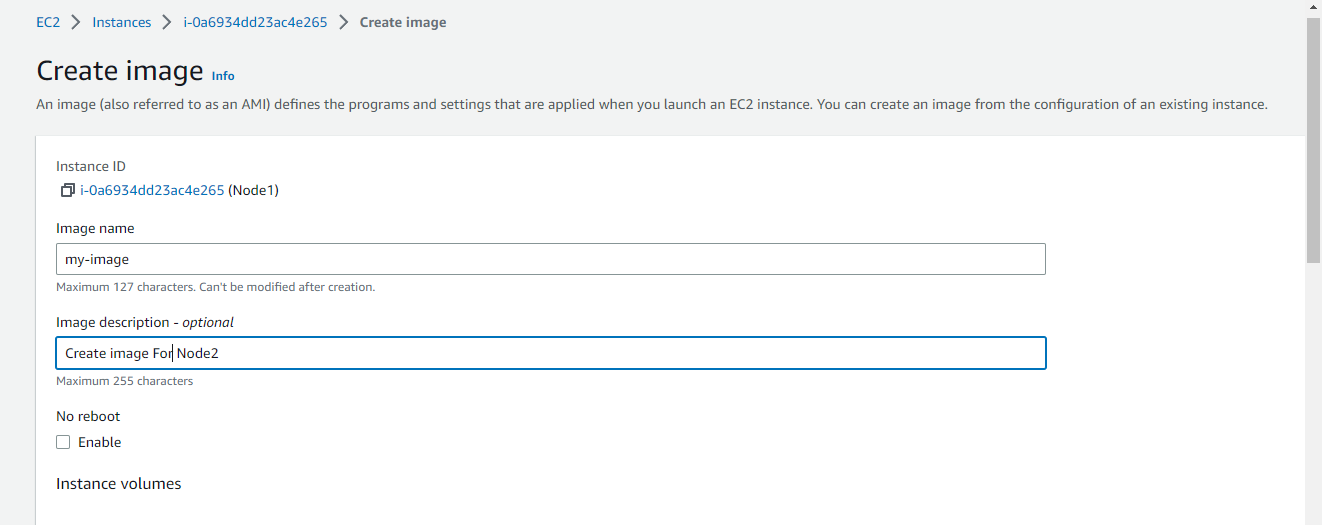


## **To create Node1 image on aws portal:**

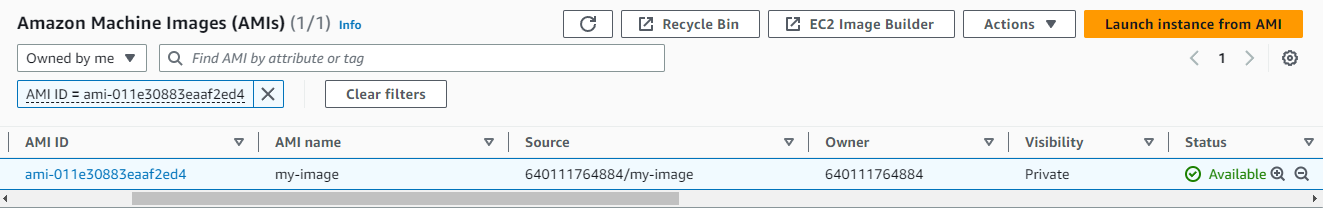
### -Create image:

->Node1->Action->Image and Templates: Create image





### -Successfully created AMI:



## **To create instance: Node2 using the AMI:**

### -Launch instance from AMI:

->EC2->launch instance->Name:Node2->My AMI:Select Created AMI:myimage

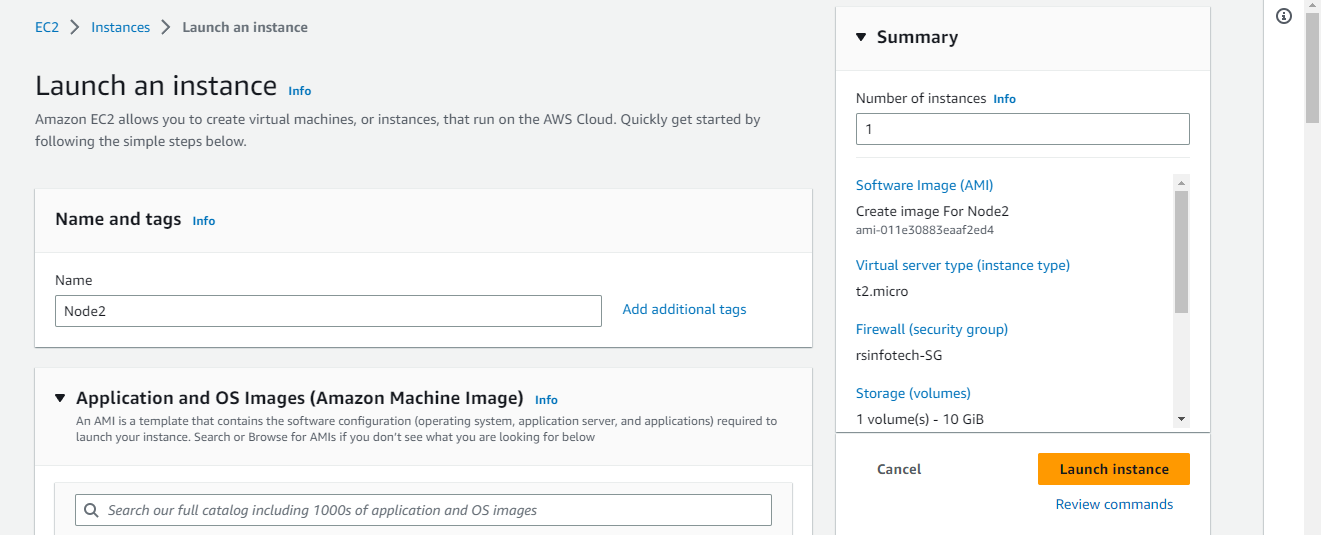
->Instance type:t2:micro->Keypair:awskey01

->Network setting:Edit

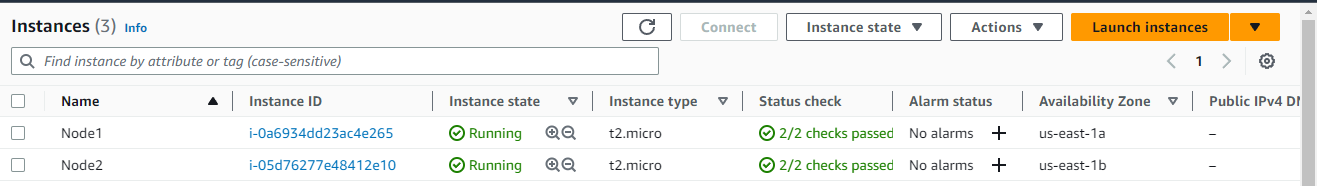
-Select Vpc,subnet select us-east-1b,Auto-assign public ip:Enable

-SG:select existing

->Create



### -Successfully created Node2:

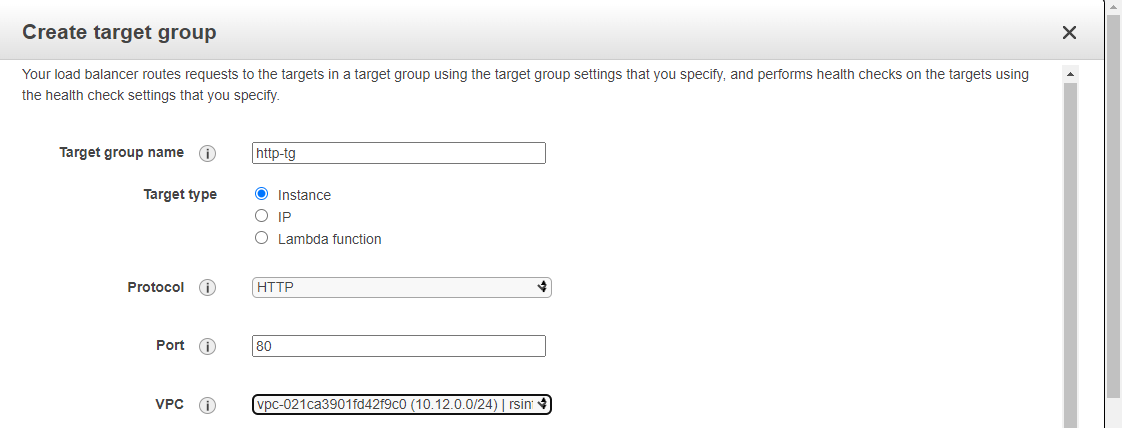


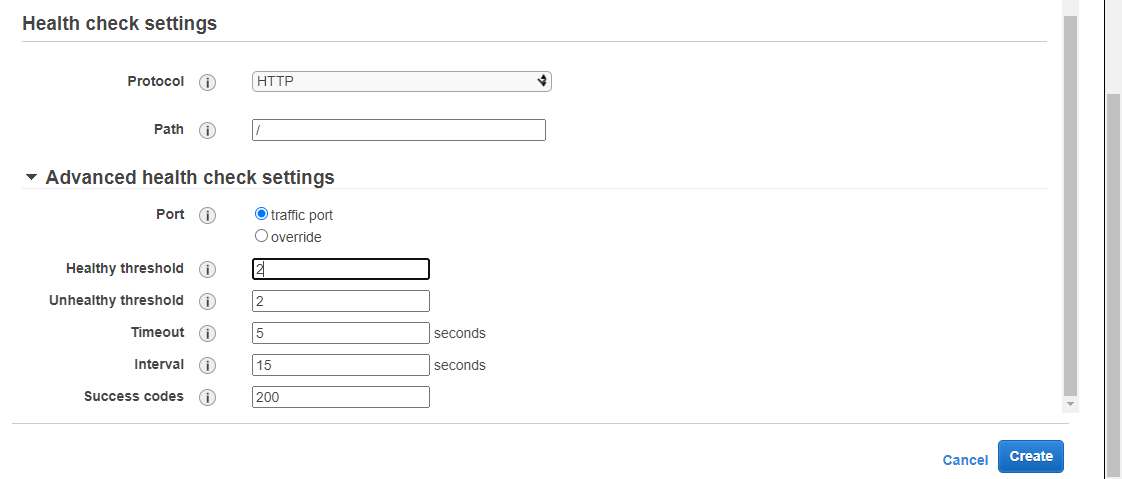
## **To create an Application Load Balancer (ALB) for a multi-node setup:**

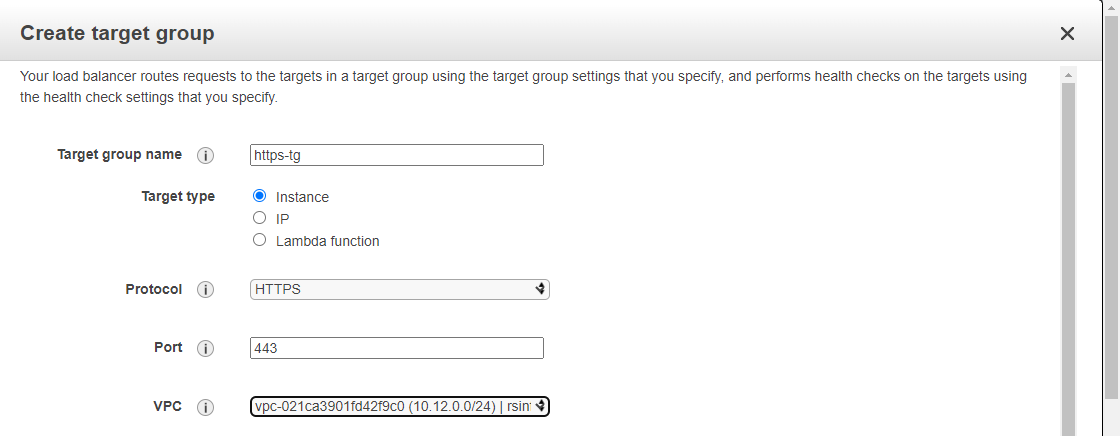
### -Create Target group:

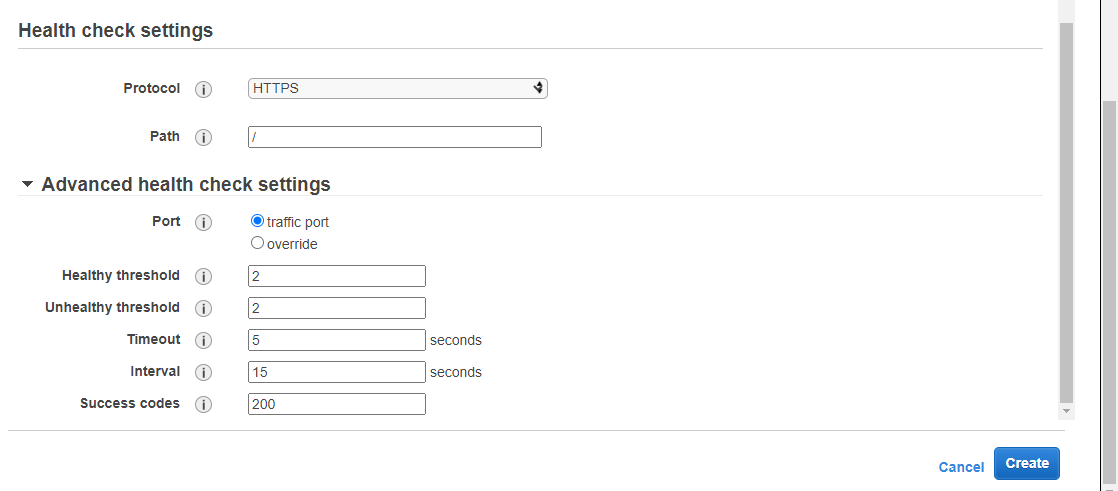
Go to the EC2 dashboard and navigate to the "Load Balancers" section

Create Http and https target group:

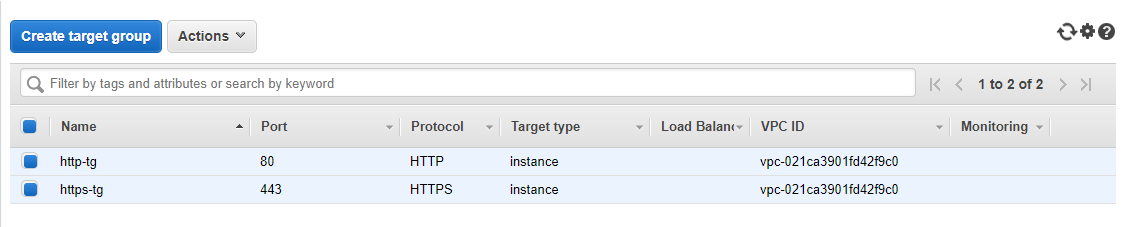




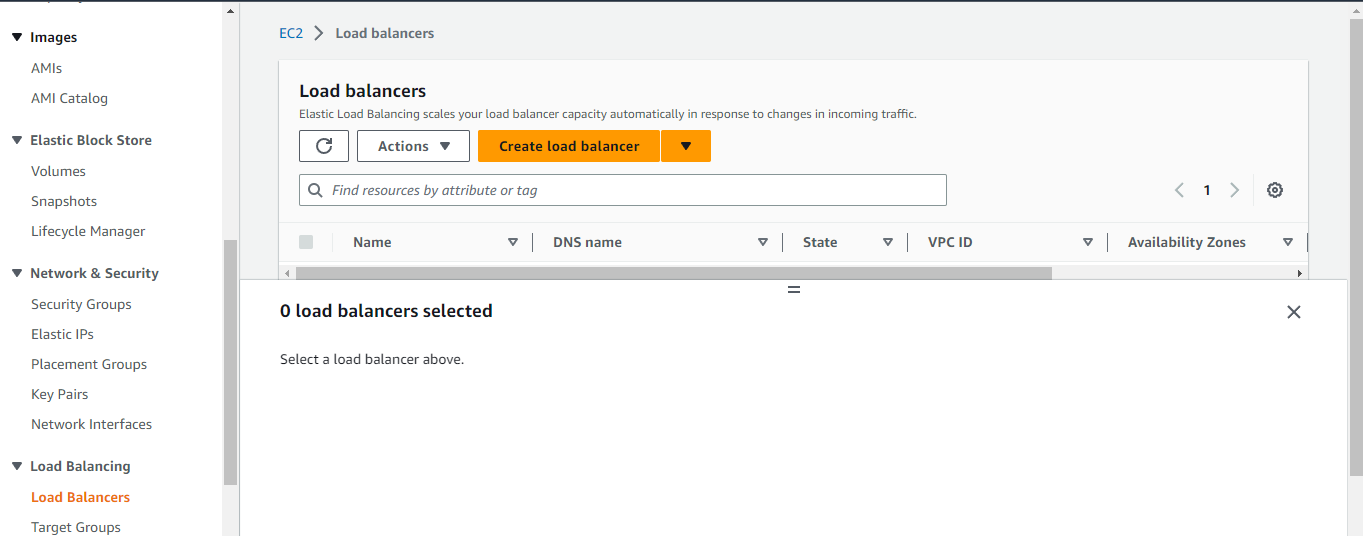




### -Successfully created http and https target group:

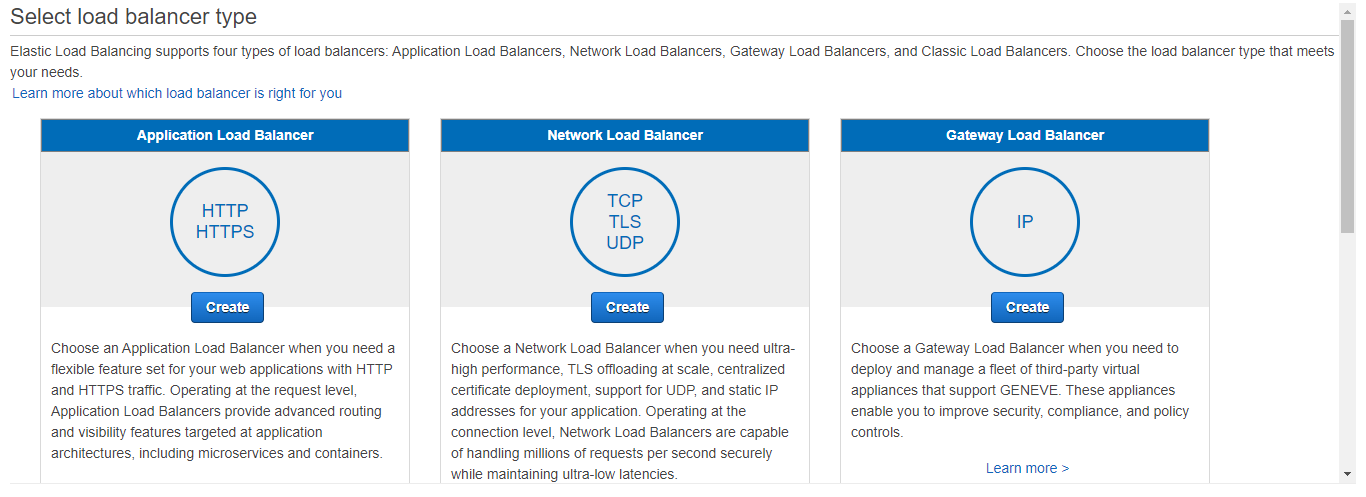


### - Click on the "Create Load Balancer" button.



### -Create the Load balancer:

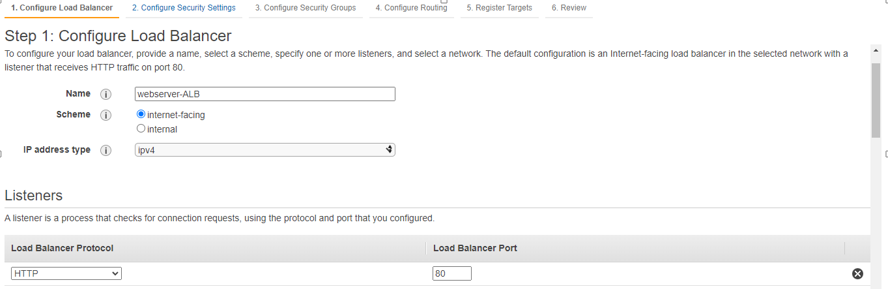
->Choose "Application Load Balancer" as the load balancer type.

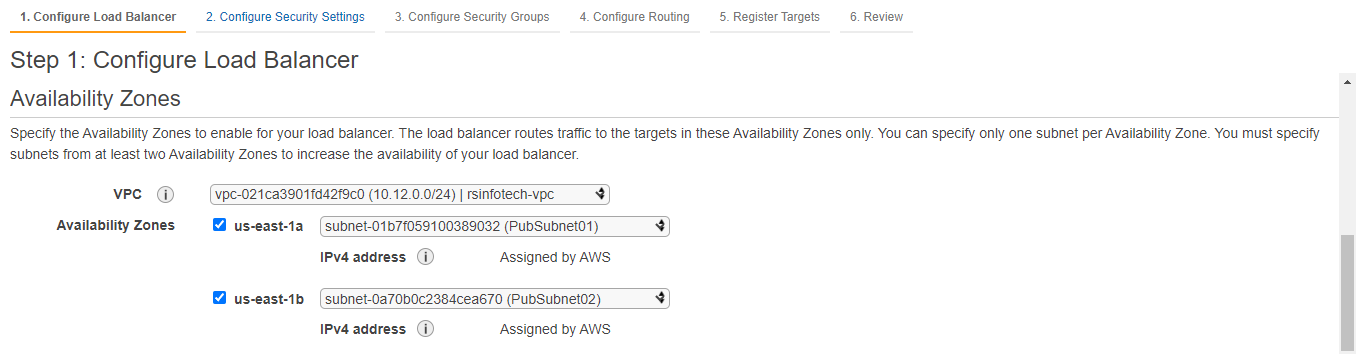


### -Configure the load balancer settings:

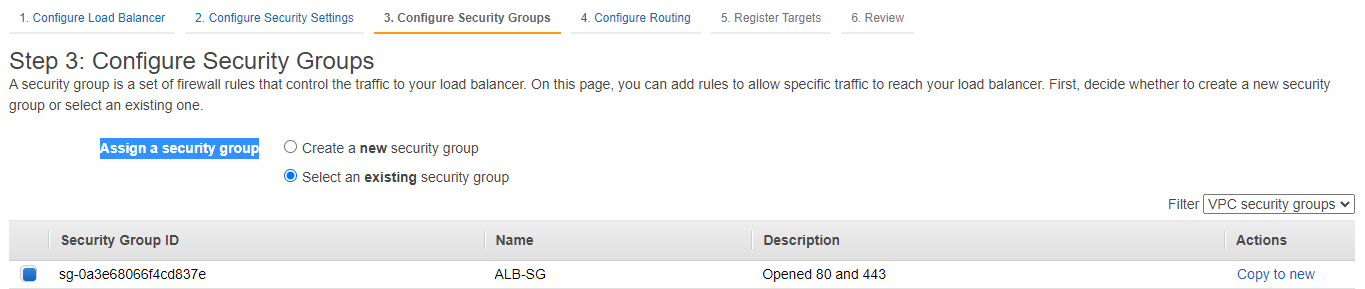
Including the name, listeners, and availability zones

Specify the port and protocol on which the load balancer will listen for incoming requests.



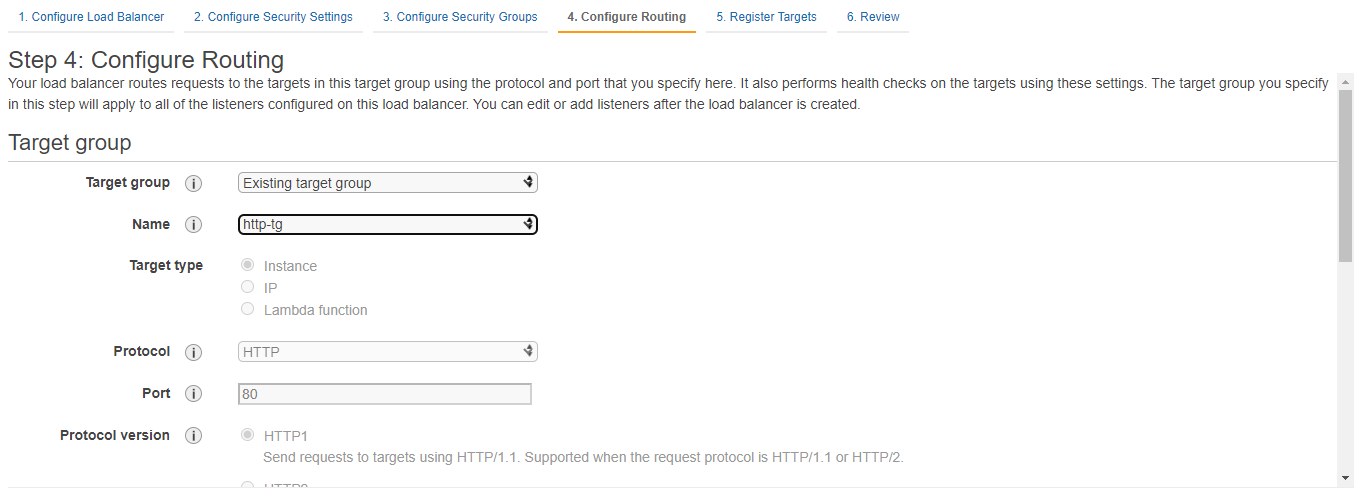
Availability zones:

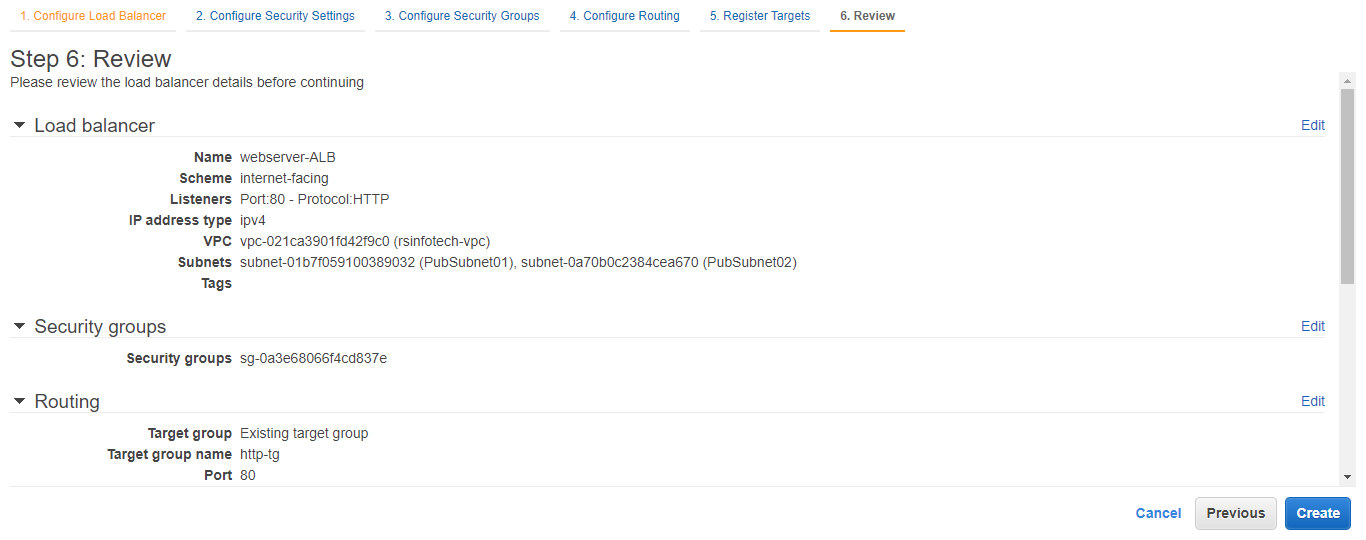
Assign a security group:



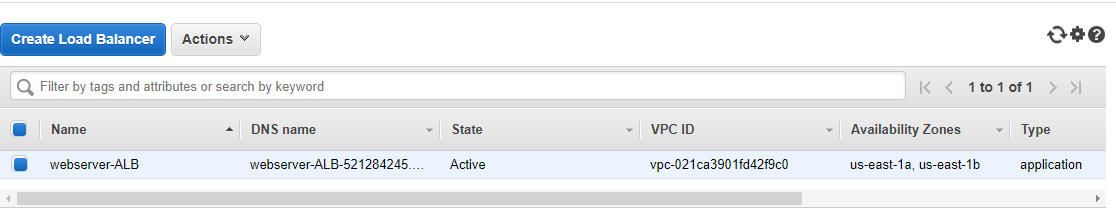
Configure Routing:

-Target groups:



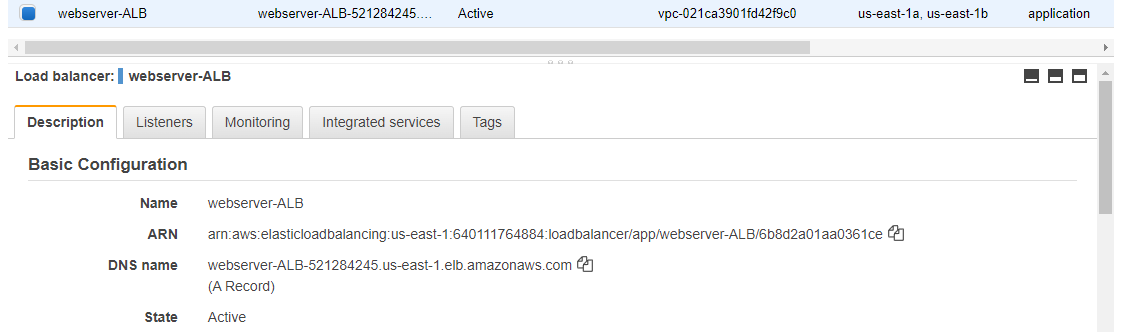


### -Successfully created ALB:



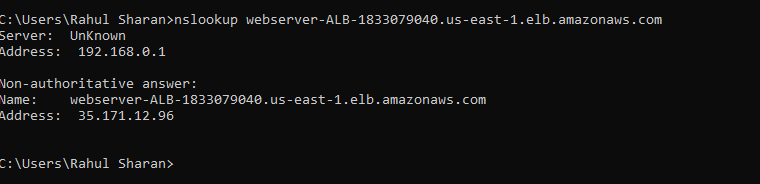
## **-Obtain the DNS name of the load balancer from the EC2 dashboard.**

### -LB Dashboard-> Description->DNS Name

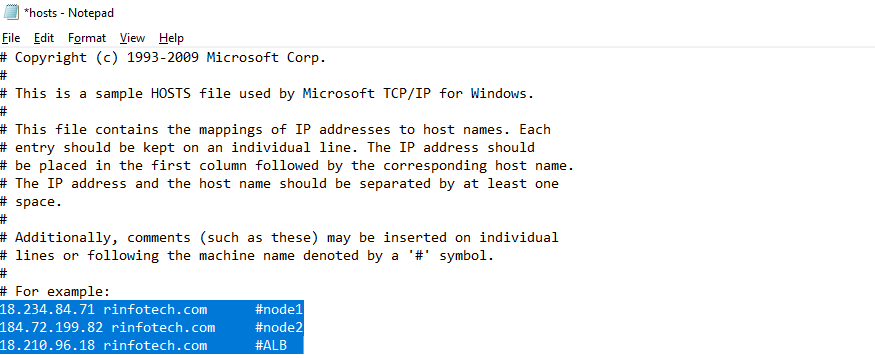


### -Check the LB Ips using DNS name:

Copy the DNS name and go local system win\_10

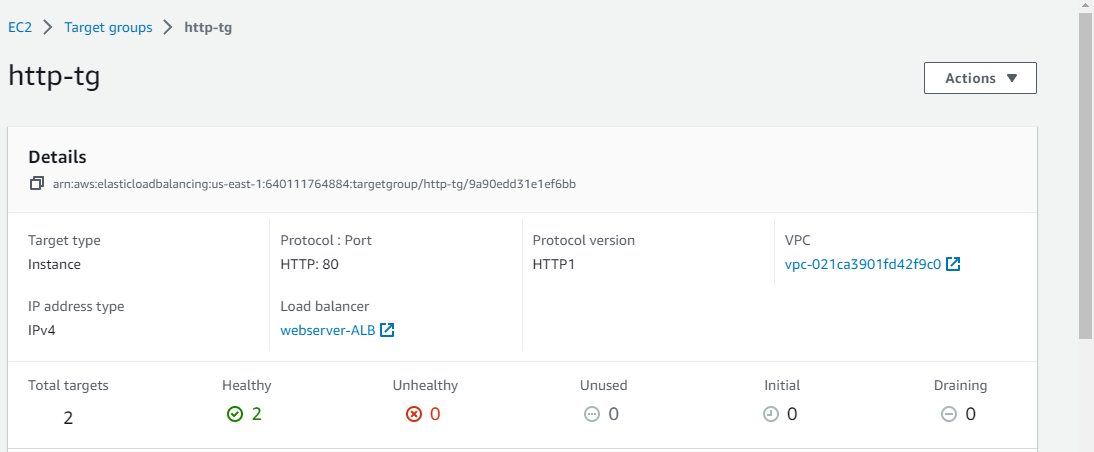


### -Update the host file:



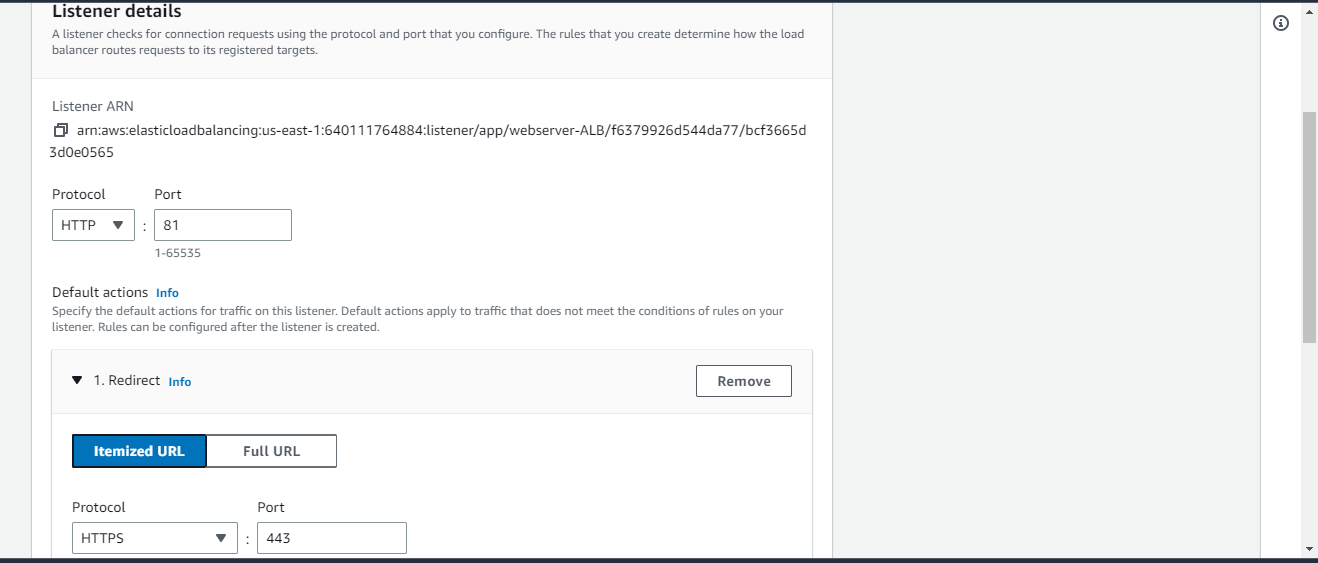
### -Registered targets:

HTTP:

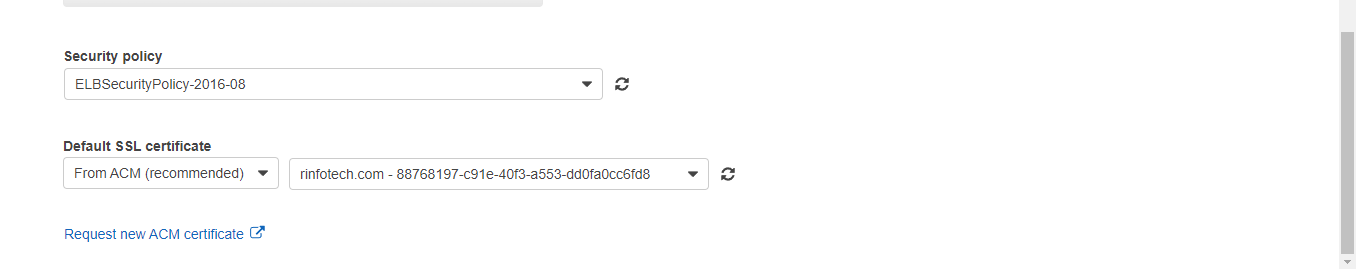


### -Add listener: HTTPS:443

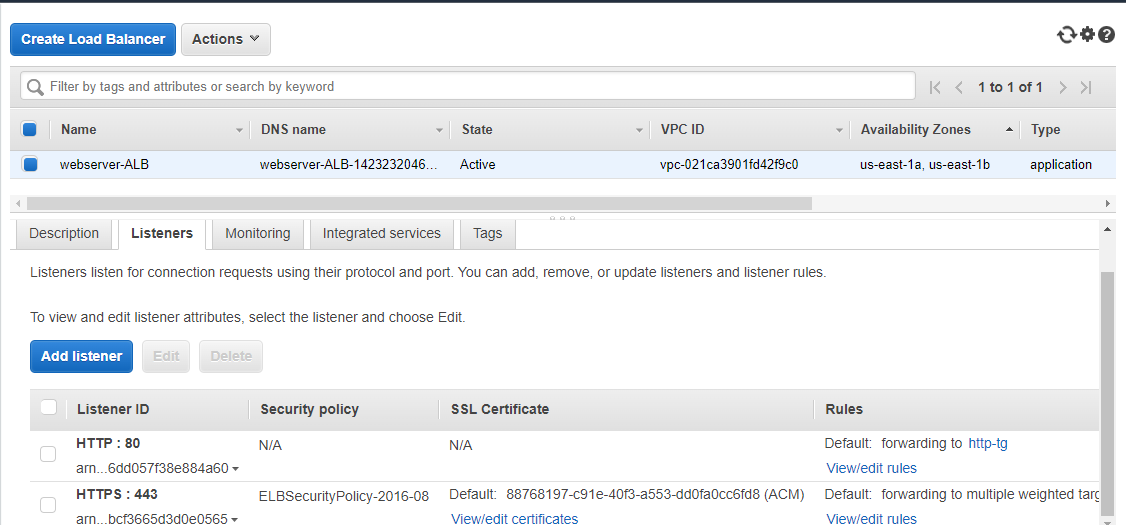
->EC2->Load balancer->Webserver-ALB: Add listener



Select Default SSL certificate:



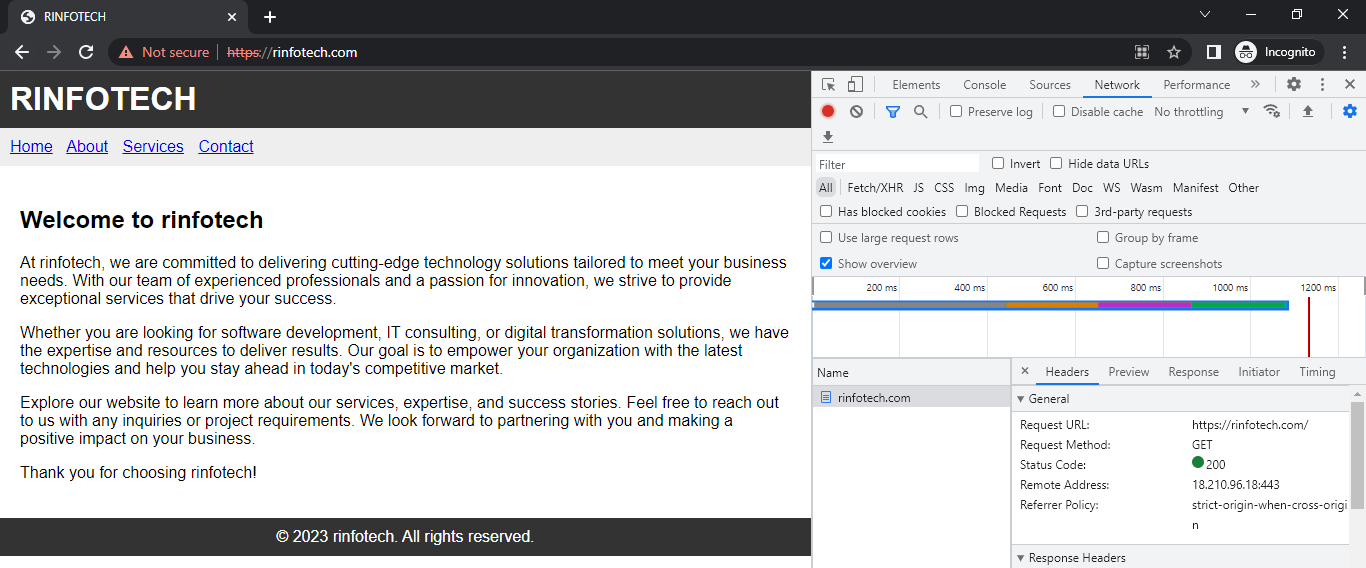
### -Successfully added Listener port 443:



## **To validate the securing access of a website:**

### -Open the web-browser and enter the:

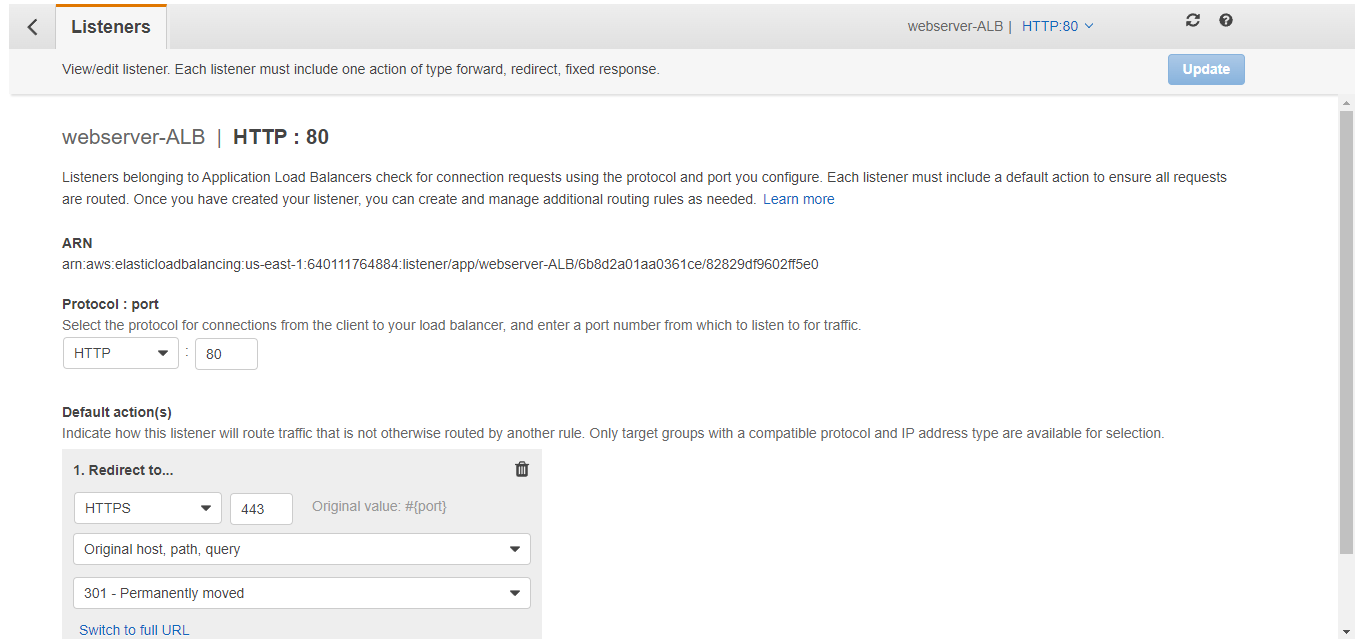
<https://rinfotech.com/>



## **Redirecting the HTTP to HTTPS:**

### -Redirecting 80 to 443:

Go to load balancer :-> webserver-ALB->Listeners: HTTP ->Edit



## **To validate the http to https access of a website:**

### -Open the web-browser and enter the:

<http://rinfotech.com/>

