

Department of Computer Engineering
Academic Term: JAN-MAY 2022

Class: *BE COMPUTERS*

Subject Name: *CLOUD COMPUTING LABORATORY*

Subject Code: CSL803

Practical No:	09
Title:	Implement AWS Load Balancer
Date of Performance:	23/02/2022
Date of Submission:	11/04/2022
Roll No:	8626
Name of the Student:	Divita Phadakale

Evaluation:

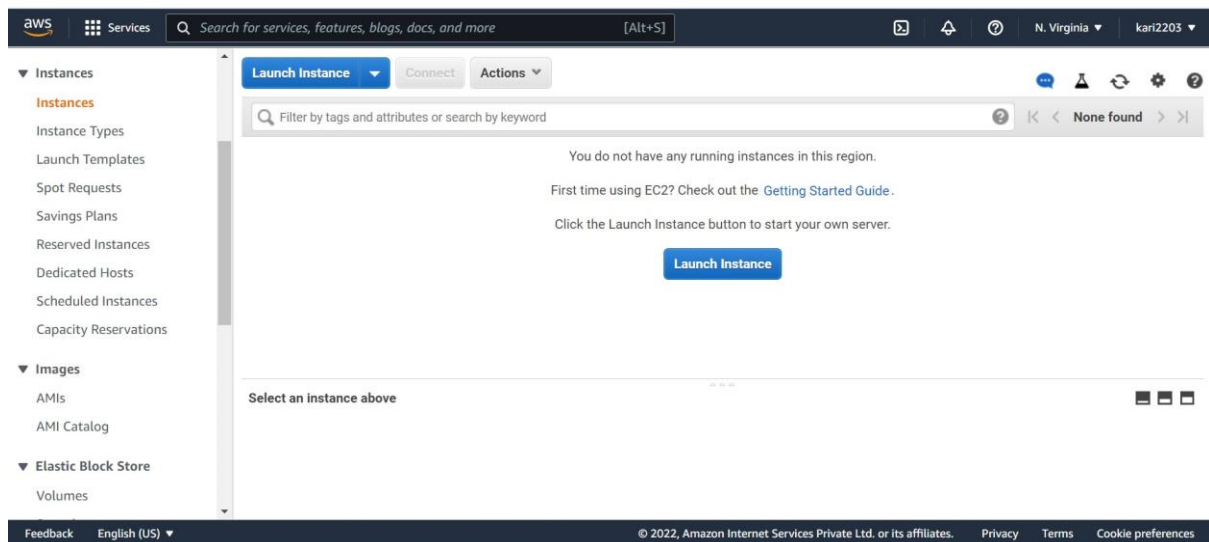
Sr. No	Rubric	Grade
1	On time submission(2)	
2	Preparedness(2)	
3	Output(2)	
4	Post Lab Questions (4)	
	TOTAL	

Signature of the Teacher:

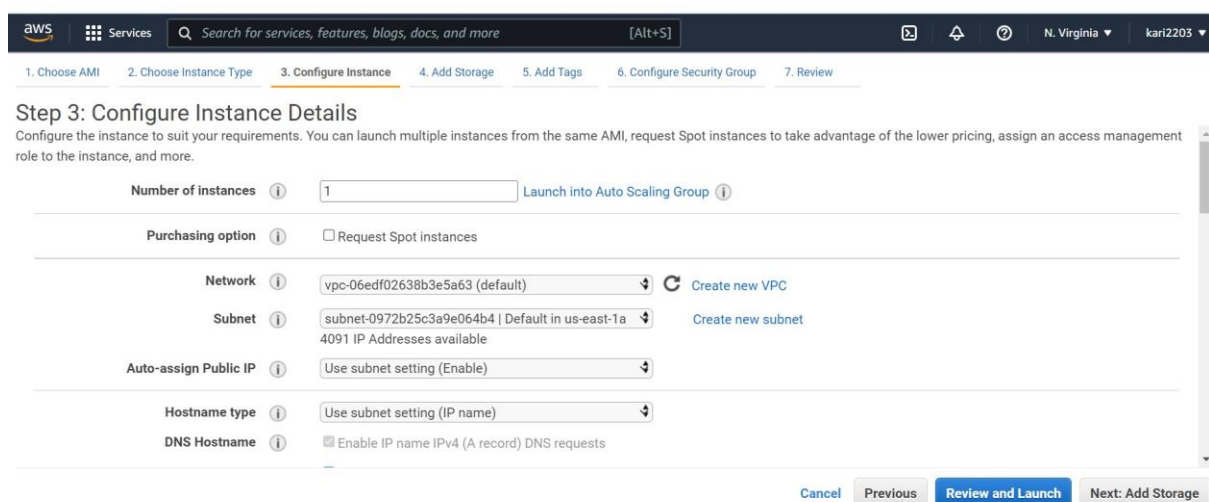
Setting up Application Load Balancer: -

1. Creating EC2 Instances:

- Click on Launch Instances for creating EC2 instances.



- Then select Amazon Linux 2 AMI and then select t2 free tier.
- In configuring Instance details, select subnet as us-east-1a



- Then add tag: Name Web1

- In configuring security group, set rules for SSH, HTTP and ICMP and then click on review and launch.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.

[Learn more about Amazon EC2 security groups.](#)

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
All ICMP - IPv4	ICMP	0 - 65535	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Cancel](#) [Previous](#) [Review and Launch](#)

Similarly add one more instance with subnet as us-east-1b and tag: Name Web2

Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Web2	i-0296ed57841b42ce3	t2.micro	us-east-1b	running	Initializing	None	ec2-54-198-133-227...
Web1	i-0796163752b81db...	t2.micro	us-east-1a	running	Initializing	None	ec2-54-227-131-106...

Instance: **i-0796163752b81dbb3 (Web1)** Public DNS: **ec2-54-227-131-106.compute-1.amazonaws.com**

Description | Status Checks | Monitoring | Tags

Instance ID: i-0796163752b81dbb3 | Public DNS (IPv4): ec2-54-227-131-106.compute-1.amazonaws.com

2. Install HTTPD servers on both the instances:

- After downloading EC2 Key-pair file (ie .pem file), open command prompt and navigate to the folder where it is downloaded.
- After that, click on the first instance and go to its SSH Client and paste the command for ssh.

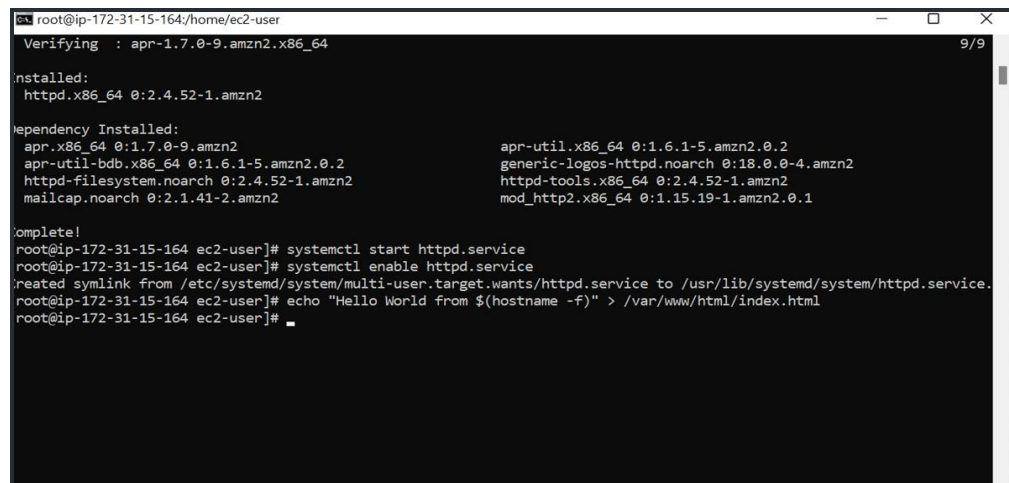
```
C:\Users\HP>cd Downloads

C:\Users\HP\Downloads>ssh -i "crce.pem" ec2-user@ec2-44-202-224-38.compute-1.amazonaws.com
The authenticity of host 'ec2-44-202-224-38.compute-1.amazonaws.com (44.202.224.38)' can't be established.
ECDSA key fingerprint is SHA256:bseB+WI456IayYzq8aA6yXWJyCH/VcgzsNyl7JNUV6k.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-202-224-38.compute-1.amazonaws.com,44.202.224.38' (ECDSA) to the list of known hosts.

 _ | _ | _ )
 _ | ( _ /   Amazon Linux 2 AMI
 _ \| _ | _ |
```

- Then type the following commands:

```
sudo su
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
echo "Hello World from $(hostname -f)" > /var/www/html/index.html
```



```
root@ip-172-31-15-164/home/ec2-user
Verifying : apr-1.7.0-9.amzn2.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.52-1.amzn2

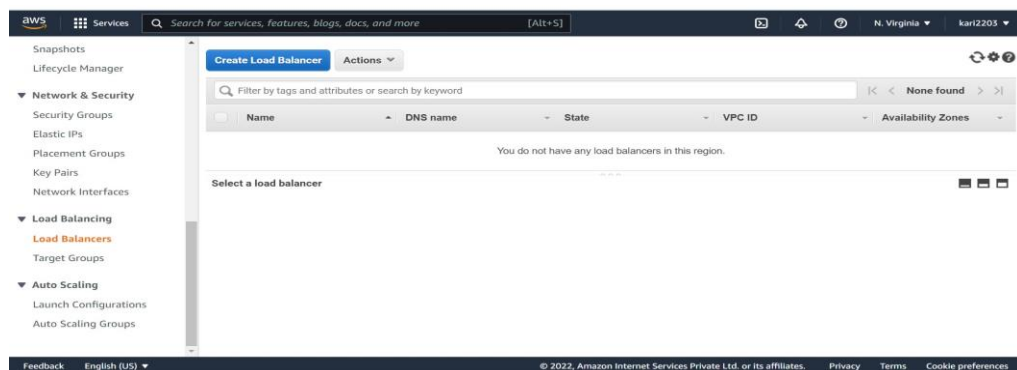
Dependency Installed:
apr.x86_64 0:1.7.0-9.amzn2
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
httpdfilesystem.noarch 0:2.4.52-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2
apr-util.x86_64 0:1.6.1-5.amzn2.0.2
generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-tools.x86_64 0:2.4.52-1.amzn2
mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
root@ip-172-31-15-164 ec2-user]# systemctl start httpd.service
root@ip-172-31-15-164 ec2-user]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
root@ip-172-31-15-164 ec2-user]# echo "Hello World from $(hostname -f)" > /var/www/html/index.html
root@ip-172-31-15-164 ec2-user]#
```

- Perform the same for the second instance as well.

3. Creating Load Balancer:

- Click on Load balancer option in the left scroll bar. - Then click on create load balancer.



- Then select Application Load Balancer.
- Give name to your load balancer (ex: crce-loadbalancer)

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia kari2203

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name

Scheme ☒ Internet-facing ☐ Internal

IP address type

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port

- Then select the availability zones.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia kari2203

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC

Availability Zones

- ☒ us-east-1a IPv4 address
- ☒ us-east-1b IPv4 address
- ☐ us-east-1c
- ☐ us-east-1d

Cancel Next: Configure Security Settings

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

- Then add new security group (ex: crce-sg)

EC2 Management Console x EC2 Management Console x crce-loadbalancer-1465167572 x +

console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-014f5bf00a73f447b

Apps Inbox - computersc... Inbox (1,241) kari... Classes ONGC Mail Accenture dashboard [PDF] XAI Previous... myIMS Animal Farm Reading list

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia kari2203

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Tags Limits

Instances

- Instances **New**
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances **New**
- Dedicated Hosts
- Scheduled Instances

Security group (sg-014f5bf00a73f447b | crce-sg) was created successfully

Details

EC2 > Security Groups > sg-014f5bf00a73f447b - crce-sg

sg-014f5bf00a73f447b - crce-sg

Actions

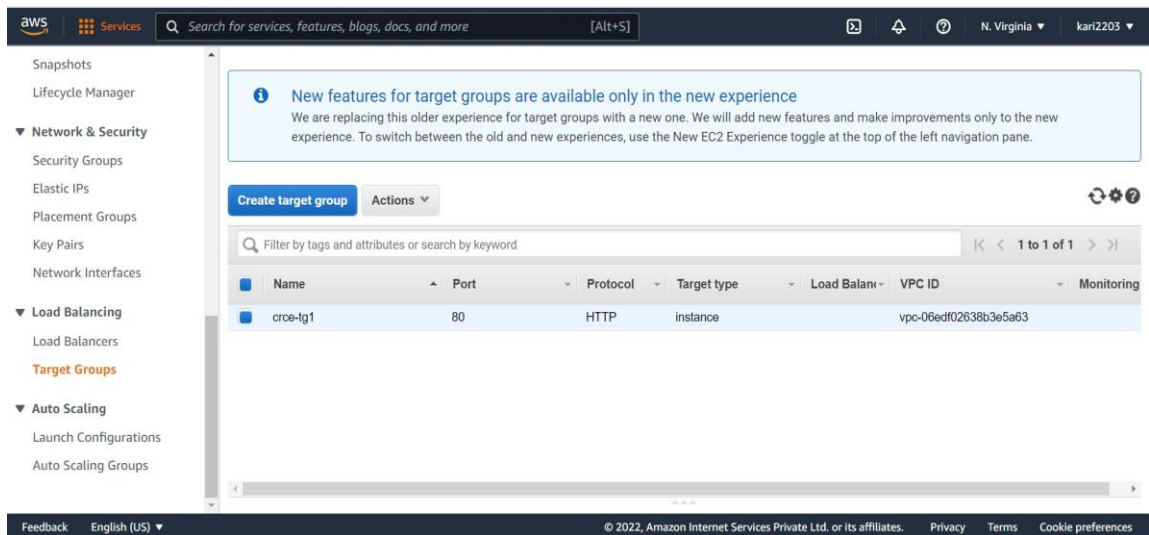
Details			
Security group name crce-sg	Security group ID sg-014f5bf00a73f447b	Description Load balancer sg	VPC ID vpc-06edf02638b3e5a63
Owner 333092234674	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

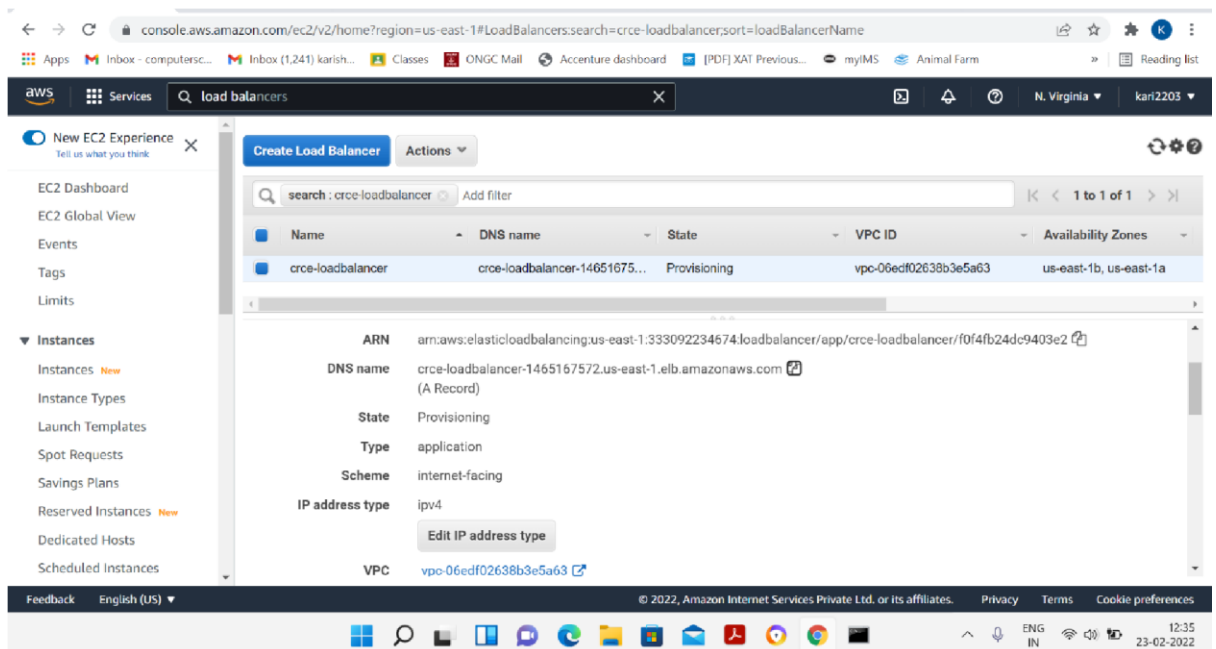
Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

12:56 23-02-2022

- Next step is to create target group with target type as Instance (ex: crce-tg1)

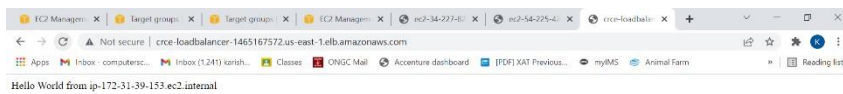
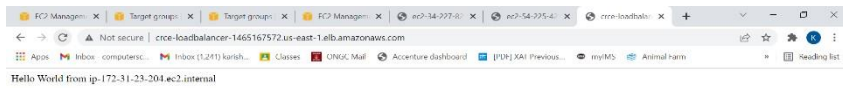


- Next, we have to register the targets. So, select the two EC2 instances and click the **Add to Registered** button and then click on **Create**.

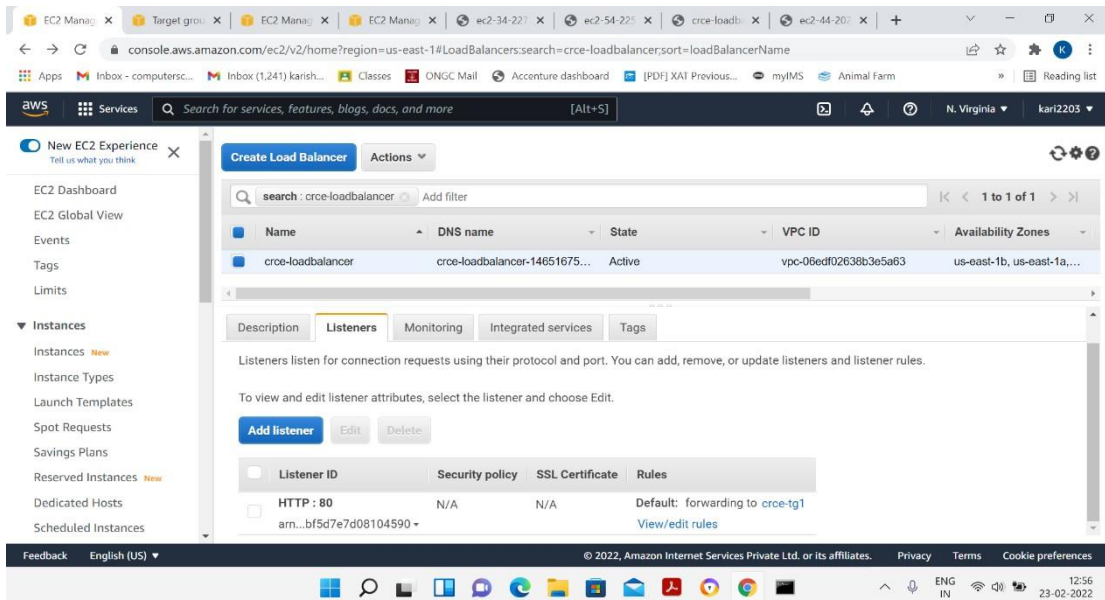


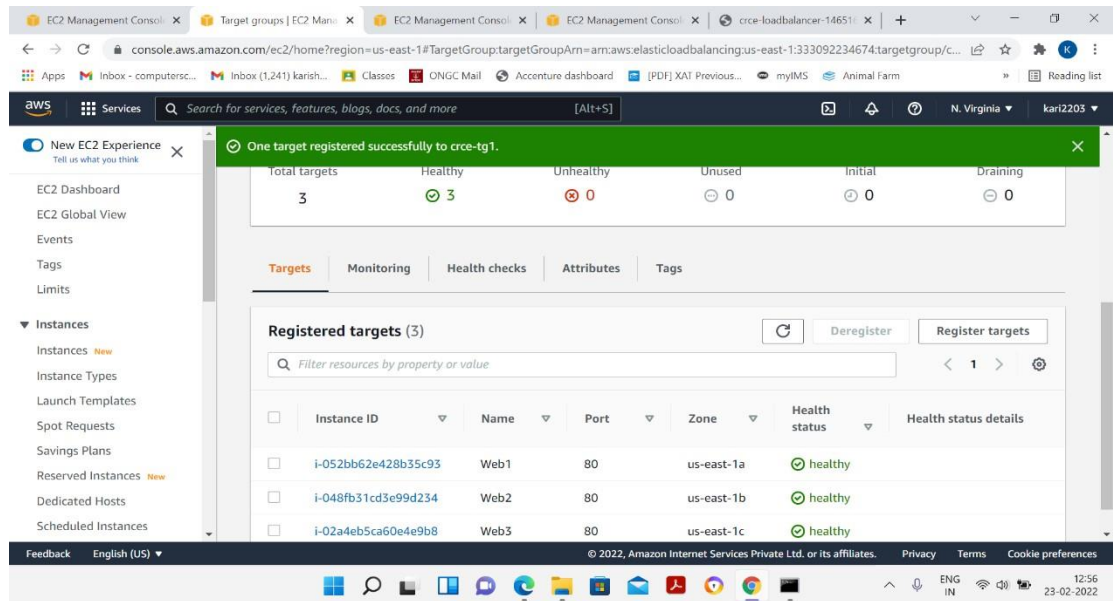
Now to check the Load balancer, click on the **DNS name** available in **Description** of the load balancer.

Output:

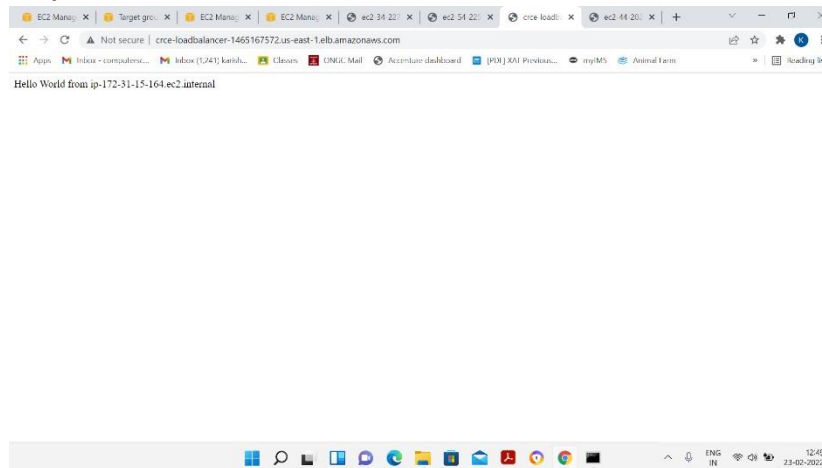


Similarly, we can create one more instance Web3 in the zone us-east-1c and add it on our load balancer by clicking on **Add listener**.





Output:



Postlab Question:

- Suppose you have an application where you have to render images and also do some general computing. From the following services which service will best fit your needs? (Explain your choice)
 - Classic Load Balancer
 - Application Load Balancer
 - Both of Them
 - None of These

Ans - b. Application Load Balancer

Application Load Balancer, since it supports path-based routing, which means it can take decisions based on the URL, therefore if our task needs image rendering it will route it to a different instance, and for general computing it will route it to a different instance.

2. When should we use a Classic Load Balancer and when should we use an Application load balancer?

Ans - A load balancer accepts requests for your application and distributes them to different servers which run your Rails application. The Classic Load Balancer is a connection-based balancer where requests are forwarded by the load balancer without “looking into” any of these requests. They just get forwarded to the backend section. Application Load Balancer enables content-based routing and allows requests to be routed to different applications behind a single load balance. While the Classic Load Balancer doesn't do that, a single ELB can host single application. ALB isn't an improved Classic Load balancer. In addition to cost savings, Application Load Balancer offers more features and flexibility compared to the Classic Load Balancer. If you'll be using TCP/SSL or EC2-Classic, then you should use the Classic Load Balancer.

3. Difference between Route53 and ELB in AWS

Ans - Both Route53 and ELB are used to distribute the network traffic. Their differences are -

1. ELBs are intended to load balance across EC2 instances in a 'single' region. Whereas DNS load-balancing (Route 53) is intended to help balance traffic 'across' regions. Route53 policies like geolocation may help direct traffic to preferred regions, then ELBs route between instances within one region.
2. Functionally, another difference is that DNS-based routing (e.g. Route 53) only changes the address that your clients' requests resolve to. On the other hand, an ELB actually reroutes traffic.
3. Another difference is with ELB you can use autoscaling to automatically register new instances added to the group with the ELB, you do not have to do anything yourself. With Route53 you have to either manually replace the old failed instance with the new one in the route or add some script to your launch configuration to automatically register the new instance with Route53 and remove the failed one.

4. What algorithm does Amazon ELB use to balance load?

Application Load Balancers applies listener rules and assigns the (HTTP/HTTPS) request to a target group. It selects a target from that target group using the round robin routing algorithm.

Network Load Balancers node that receives the connection, selects a target from its target group using a flow hash routing algorithm. Classic Load Balancers uses round robin routing algorithm for TCP listeners and least outstanding requests routing algorithm for HTTP and HTTPS listeners. The modern ELBs use round robin routing algorithm for HTTP/HTTPS requests and flow hash routing algorithm for TCP requests