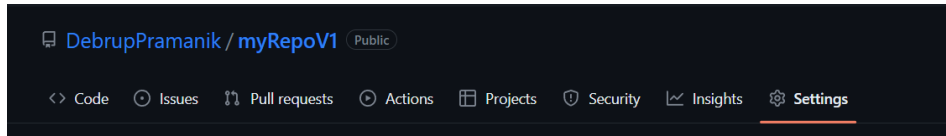


ASSIGNMENT – 11

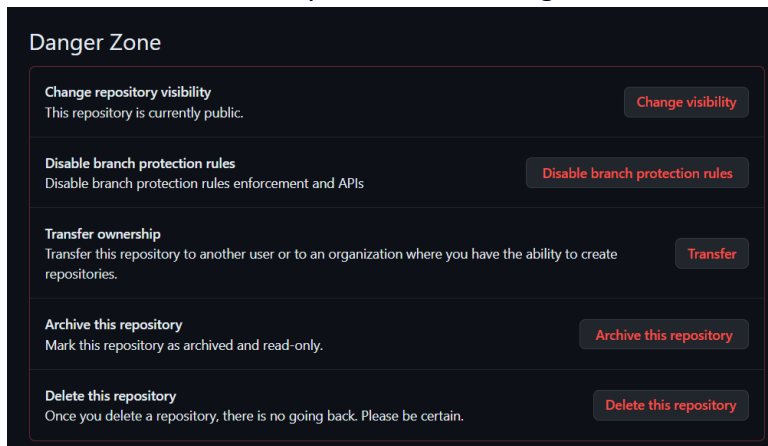
Problem Statement: Build Scaling plans in AWS that balance load on different EC2 instances.

Procedure:

1. Sign-in to GitHub. Make sure the Repository which will be cloned is made public or not.
 - a. For This, select the “settings” tab of your repository.



- b. Next Scroll Down until you reach the danger-zone section.

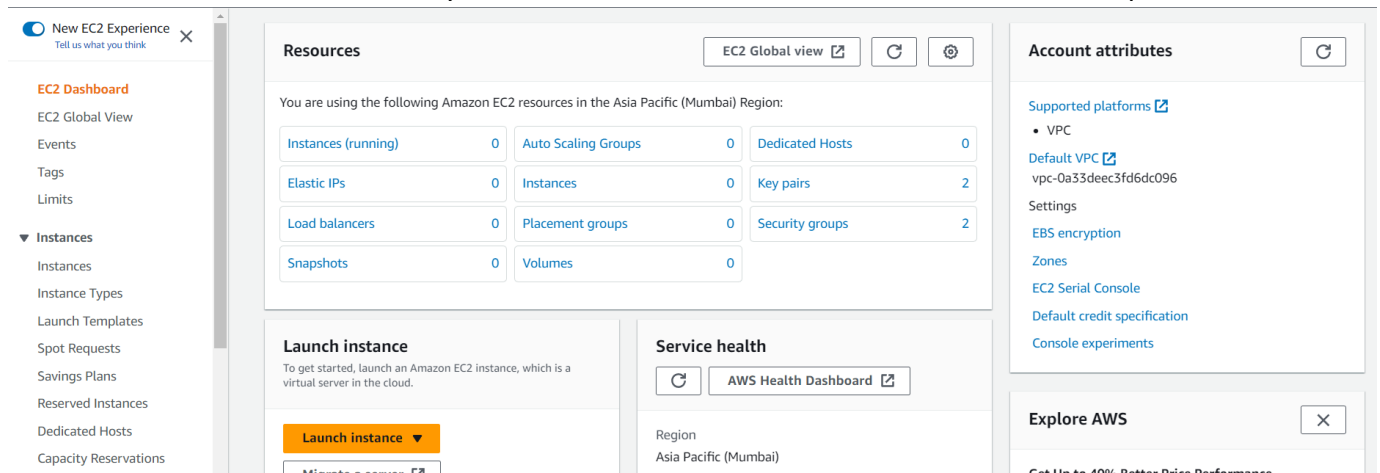


- c. In here, check The Change Repository visibility option. Here, we can see the repository is currently Public. If it shows Private then click on the Change Visibility option and follow the on-screen Instructions to make the repository Public.

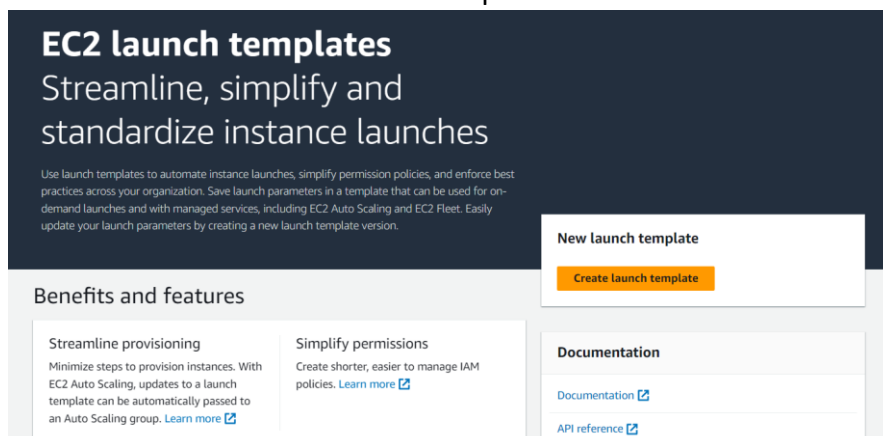
2. Now in another tab open AWS and Sign-in to your console.

3. Now go to your EC2 dashboard.

4. From the Left Side Nav-bar Go to your Instances Section. Under it Click the launch templates button.



5. Now click on the Create Launch Templates button.



6. Now, give a name and description for your EC2 template you are about to create. Here we gave the same for both the fields.

Next, Check the “Provide Guidance” box.

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required

debtemp1

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '"', '@'.

Template version description

debtemp1

Max 255 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags

▶ Source template

7. Next, under the quick start menu select Ubuntu as the OS.

RecentsQuick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-02eb7a4783e7e9317 (64-bit (x86)) / ami-0a5dcff6b7af3fc9 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture

AMI ID

64-bit (x86)

ami-02eb7a4783e7e9317

Verified provider

8. Under Instance type select t2.micro type of configuration.

▼ Instance type [Info](#) [Advanced](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux pricing: 0.0124 USD per Hour

On-Demand Windows pricing: 0.017 USD per Hour

On-Demand RHEL pricing: 0.0724 USD per Hour

On-Demand SUSE pricing: 0.0124 USD per Hour

All generations

Compare instance types

9. Select Existing Key-Pair and Security Group and if not applicable then Generate or Create a Key-Pair or Security Group wherever required.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

debkey2

Create new key pair

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group

Create security group

Security groups [Info](#)

Select security groups

mysec1 sg-0493398d43b761e55

VPC: vpc-0a33deec3fd6dc096

Compare security group rules

10. Now, Click on the Advanced Group Section at the bottom.

Scroll Down to User Data Section and paste the following commands in the provided box.

User data - optional [Info](#)
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone YourRepositoryURLhere
cd YourRepositoryNamehere/
npm install
node index.js
```

After adding the commands, it will look like this.....

User data - optional [Info](#)
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1/
npm install
node index.js
```

11. Now click on the Create Launch Template button.

Don't include in launch template

Metadata response hop limit [Info](#)
Don't include in launch template

Allow tags in metadata [Info](#)
Don't include in launch template

User data - optional [Info](#)
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1/
npm install
node index.js
```

☐ User data has already been base64 encoded

▼ Summary

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-02eb7a4783e7e9517

Virtual server type (instance type)
t2.micro

Firewall (security group)
mysec1

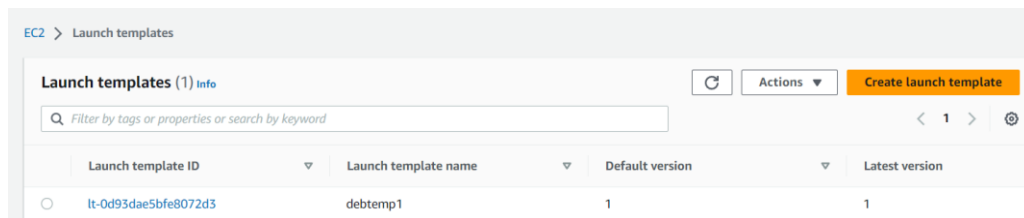
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

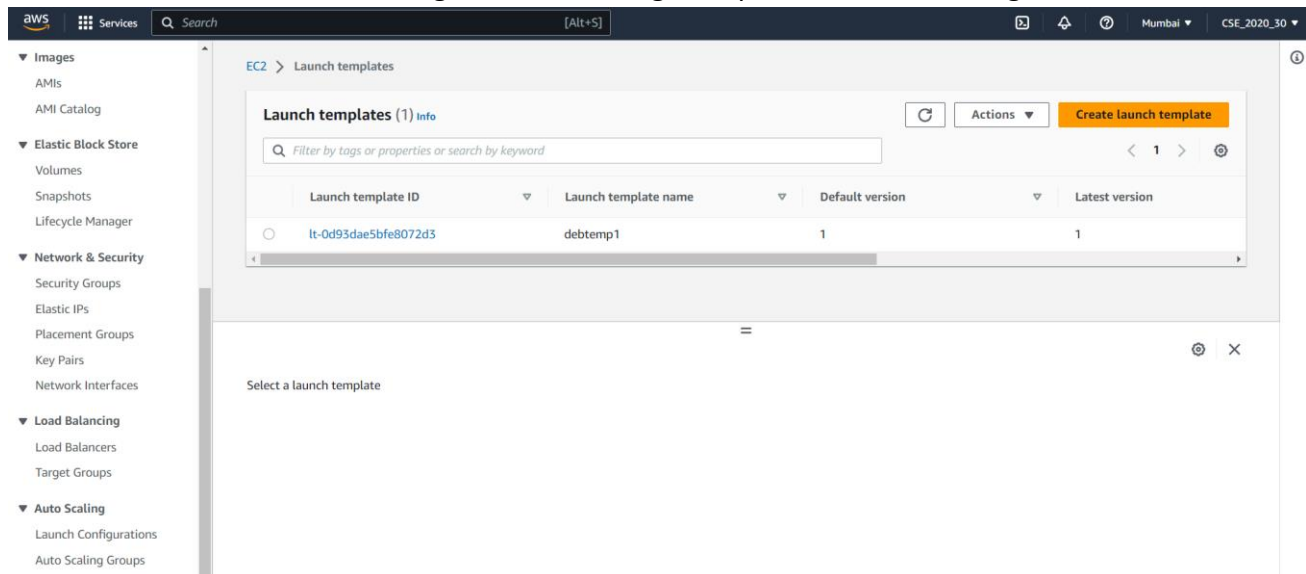
×

Cancel

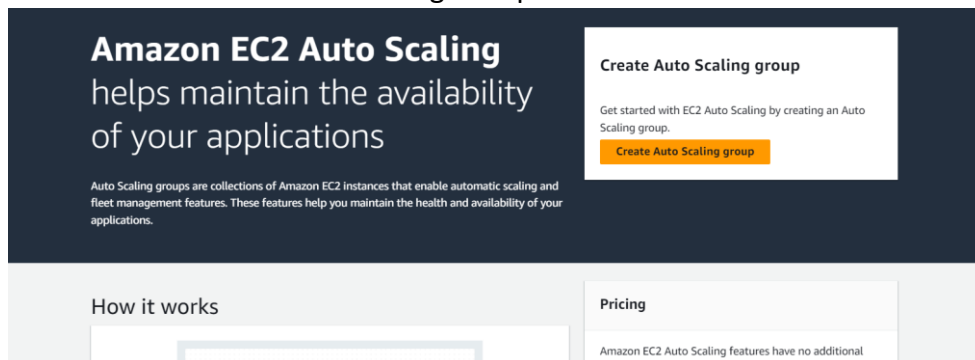
Create launch template



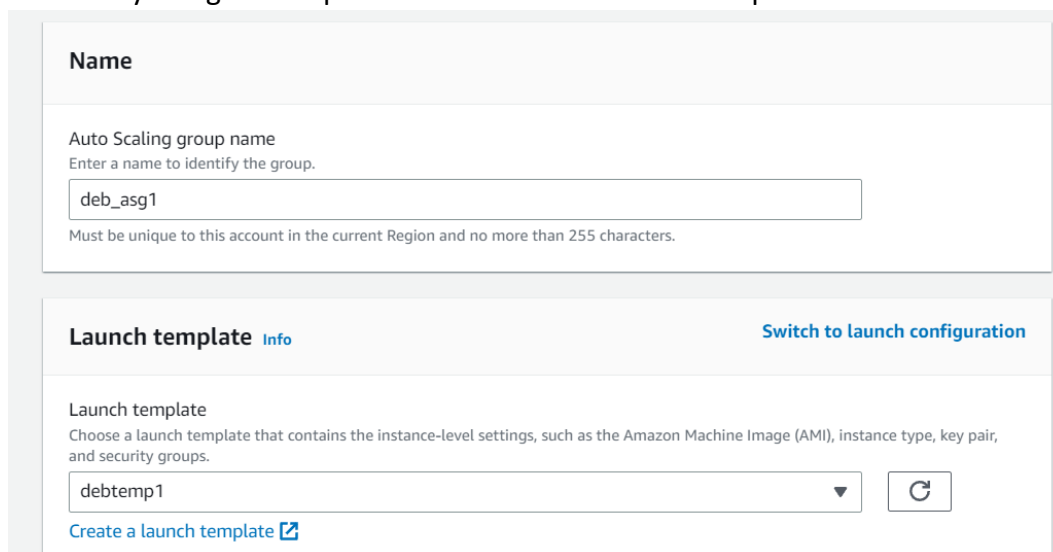
12. Now from the Left side Nav Bar go to Auto Scaling Groups under Auto-Scaling section.



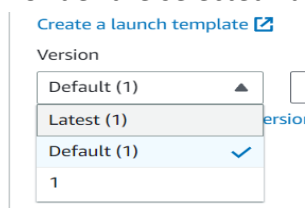
13. Now click on Create Auto Scaling Group.



14. Now, Give a unique name to your new Auto Scaling Group. Also select the Launch Template that we recently created by using the drop-down menu under Launch Templates section.



15. Under the selected Launch Template click on the version option and select Latest.



16. Now click on the Next button.

17. After that, Under Availability Zones and Subnets select all the zones that appear.

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0a33deec3fd6dc096
172.31.0.0/16 Default

Create a VPC

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

ap-south-1a | subnet-0bbe74a9835a07e38
172.31.32.0/20 Default

ap-south-1b | subnet-0916d7caf57f5d661
172.31.0.0/20 Default

ap-south-1c | subnet-09ddf920c63dcde50
172.31.16.0/20 Default

Create a subnet

18. Again, click on the Next button.

19. Now Under Load Balancers select the Attach to a New Load balancer option.

Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☐ Attach to an existing load balancer
Choose from your existing load balancers.

☒ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

20. Now select Internet-Facing under Load balancer scheme.

Load balancer scheme
Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

21. Under Listeners and Routing enter the port no. of the project and select Create target group followed by giving the target group a name.

Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol: HTTP

Port: 4000

Default routing (forward to): Create a target group

New target group name
An instance target group with default settings will be created.

debasg1

22. Now click on the next button.

23. After clicking on the Next button, a new page will open. Under Group Size mention:

- a) Desired Capacity = 2
- b) Minimum Capacity = 2
- c) Maximum Capacity = 3

Group size - optional

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity: 2

Minimum capacity: 2

Maximum capacity: 3

24. Now under Scaling policies Choose the Target Tracking Scaling policy option.

Select the metric type as Average CPU utilization.

Set Target Value to 50.

Set Warm-Up time to 300 seconds under Instances Need.

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

☒ Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

☐ None

Scaling policy name

Metric type

Target value

Instances need
 seconds warm up before including in metric

☐ Disable scale in to create only a scale-out policy

25. Then click on next.

26. Nothing to do in Notifications page. So again, click on the Next button.

27. No tags needed. Again, click on the Next button.

28. Now Review your Auto-Scaling Group you are going to create. Now click on the Create Auto-Scaling Group button.

Cancel

Previous

Create Auto Scaling group

29. Now we can go to the Auto Scaling Groups section and find our newly created Auto Scaling Group.

Auto Scaling groups (1) Info						
<input type="text" value="Search your Auto Scaling groups"/>						
<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	debasg1	debttemp1 Version Latest	2	-	2	2

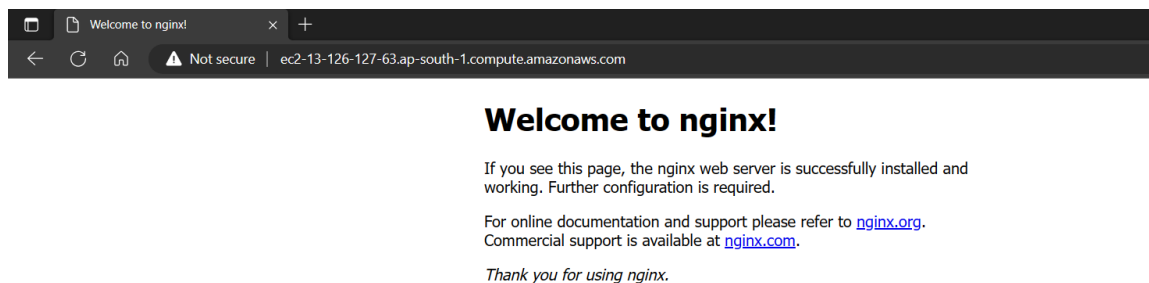
30. Return to the Instances Page using the Left side Nav bar.

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	-	i-0a7cf75ace548840f	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	-	i-0c51d43594a967f2b	Running	t2.micro	2/2 checks passed

31. Click on the First instance. Copy its public IPV4 DNS.

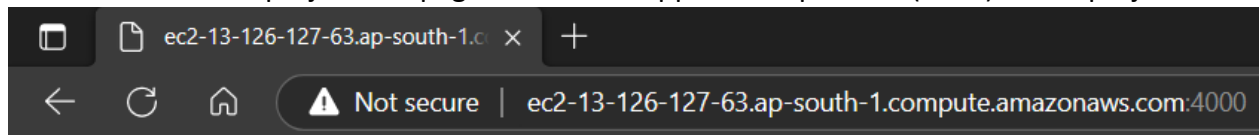
Instance summary for i-0a7cf75ace548840f Info		
Updated less than a minute ago		
Instance ID i-0a7cf75ace548840f	Public IPv4 address 13.126.127.63 open address	Private IPv4 addresses 172.31.14.78
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-13-126-127-63.ap-south-1.compute.amazonaws.com open address

32. Paste it in another browser.



We can successfully access the webpage.

33. Now to access our project webpage we need to append the port no. (4000) of our project with a “.”



Hello. My Name is Spider-Man!!! Nice to meet You!!!

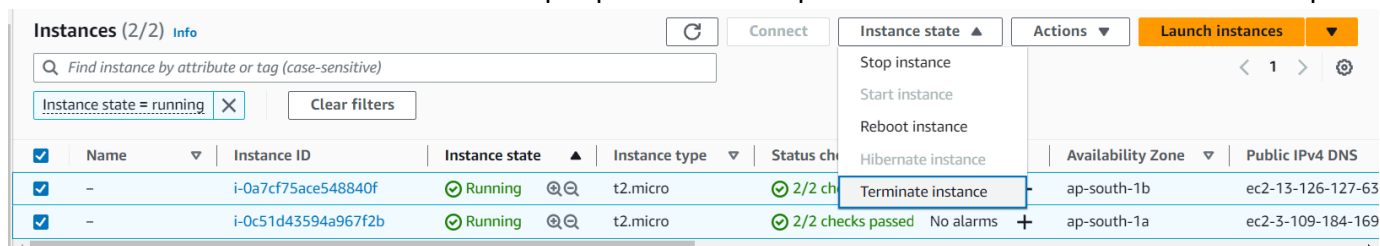
34. To test our Auto-Scaling Group actually works we need to crash or overload the existing instance servers. Then only our Auto-Scaling Group will provide fresh instance servers automatically in case of crash; or it can provide extra servers to handle overloads.

35. We will now **CRASH THE SERVER INSTANCES** by terminating them.

36. Go to the instances page. Select the server instances.

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/>	-	i-0a7cf75ace548840f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-126-127-63
<input checked="" type="checkbox"/>	-	i-0c51d43594a967f2b	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3-109-184-169

37. Now click on the Instance State button up top. From the drop-down select the Terminate instance option.



38. Refresh the instances the page from time to time. After few seconds the instances will get terminated.

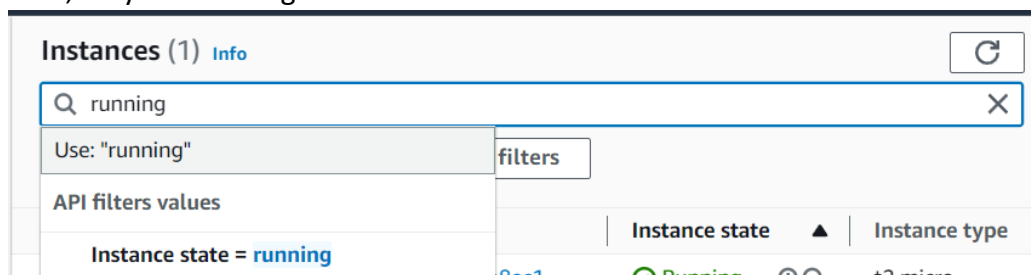
39. Wait for some time. Keep refreshing using the refresh button on the left side of Instance state button.

40. After some you will notice a new instance server will appear automatically! To help finding it more easily we need to activate the instance running filter. Click on the search box below the Instances Section Heading. Start typing running.

Select the option “Instance state = running option in the suggestion dropdown. The filter will be activated.

You can alternatively type the command directly in the search-box directly.

Now, only the running instances will be shown in the list.



41. After some few seconds of refreshing we will be able to see two new servers are running.

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	-	i-05f306bab83bb8ec1	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-43-204-214-47
<input type="checkbox"/>	-	i-04f330d210730d2a8	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3-6-86-146.ap

Now again copy paste the new public IPv4 DNS of the first instance and port no. in the other browser to see if the instances are working. It will be working.

So, our Auto-Scaling Group can handle instance crashing by providing new fresh instances.

42. We will now **OVERLOAD THE SERVER INSTANCES** by running scripts and increasing CPU utilization value above the threshold that we specified during Configuration of the Auto scaling group.

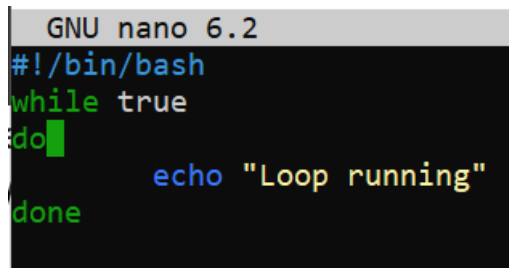
43. For it we will use:

- a) Use Bitwise SSH client for instance 1.
- b) Use direct connect terminal in AWS for instance 2.

44. For Instance-1:

- a) Copy the public IPv4 address
- b) Open Bitwise SSH client.
- c) Paste the IP and select/specify the necessary options. (**Refer Ass7**)
- d) Now Log-In to your server.
- e) Open the new Terminal.
- f) Now enter the command:
→ **nano infil.sh**
- g) After the command a new nano Editor window will open. Type the following in it.

```
#!/bin/bash
while true
do
    echo "Loop running"
done
```



GNU nano 6.2

```
#!/bin/bash
while true
do
    echo "Loop running"
done
```

- h) Now, to save and close the shell script we need to press the following shortcuts and keys sequentially:

Ctrl+X

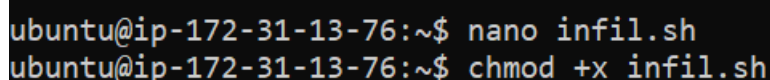
Y

Enter

- i) Now you will be returned back to the terminal.
- j) Now type the following commands:

→ **chmod +x infil.sh**

(Used to give the execute permission for the infil.sh file)



```
ubuntu@ip-172-31-13-76:~$ nano infil.sh
ubuntu@ip-172-31-13-76:~$ chmod +x infil.sh
```

→ **./infil.sh**

(Used to execute the infil.sh script)


```

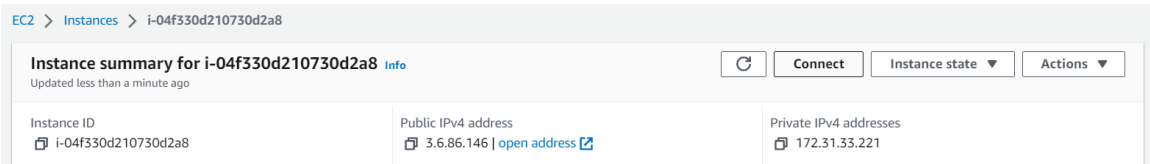
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running
Loop running

```

- k) Now the script will start running infinitely!
- l) DO NOT CLOSE THE TERMINAL. Keep it minimized.

45. For Instance-2:

- a) Click on the instance 2.
- b) Now click on the connect button



- c) Click on connect again
- d) After the terminal gets opened we again follow the same steps as we did for instance-2
- e) Enter the command:

→ **nano infil.sh**

- f) After the command a new nano Editor window will open. Type the following in it.

```

#!/bin/bash
while true
do
    echo "Loop running"
done

```

- g) Now, to save and close the shell script we need to press the following shortcuts and keys:

Ctrl+X

Y

Enter

- h) Now you will be returned back to the terminal.
- i) Now type the following commands:

→ **chmod +x infil.sh**

(Used to give the execute permission for the infil.sh file)

→ **./infil.sh**

(Used to execute the infil.sh script)

- j) Now the script will start running infinitely!
- k) DO NOT CLOSE THE TERMINAL TAB. Go back to the previous tab to keep working in AWS.

46. Now go to the instances page.

47. Select both the instances.

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/>	-	i-05f306bab83bb8ec1	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-43-204-214-47
<input checked="" type="checkbox"/>	-	i-04f330d210730d2a8	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3-6-86-146-ap-

48. Click on the instances white bar at the bottom of the page.

Instances (2/2) Info

↺

Connect

Instance state ▼

🔍

Find instance by attribute or tag (case-sensitive)

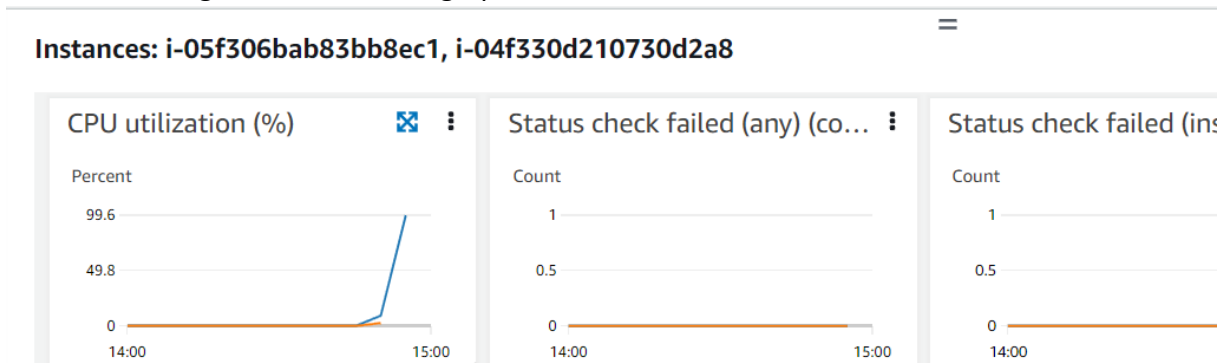
☑	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status
☑	-	i-05f306bab83bb8ec1	✔ Running 🔍🔍	t2.micro	✔ 2/2 checks passed	No alarms
☑	-	i-04f330d210730d2a8	✔ Running 🔍🔍	t2.micro	✔ 2/2 checks passed	No alarms

Instances: i-05f306bab83bb8ec1, i-04f330d210730d2a8

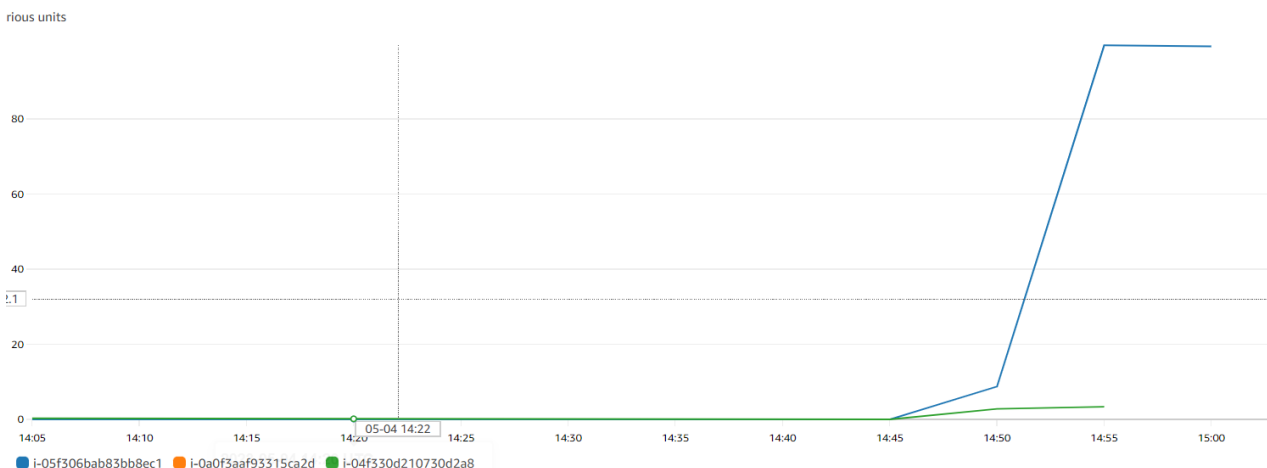
49. Now drag the two bars to expand the view.

Instances: i-05f306bab83bb8ec1, i-04f330d210730d2a8

50. We are interested only in the CPU utilization graph. Click on the maximize icon by hovering over the graph as shown in the fig to maximize this graph.



51. Our 1st instance has already crossed over 50% utilization. That's why we can see already a new third instance has been initiated by our auto-scaling group to compensate for the overload.



52. There can only be 3 servers running at a time for us as specified in our Auto-Scaling group when we were creating it. Hence, we have reached our maximum limit of instances running concurrently.

✓	Name	Instance ID	Instance state
✓	-	i-05f306bab83bb8ec1	Running
✓	-	i-04f330d210730d2a8	Running
✓	-	i-0a0f3aaf93315ca2d	Running

Security group name	Key name	Launch time
mysec1	debkey2	2023/05/04 16:36 GMT+5:30
mysec1	debkey2	2023/05/04 16:38 GMT+5:30
mysec1	debkey2	2023/05/04 20:33 GMT+5:30

Hence, our Auto-Scaling Group can handle instance overloading by providing new instances to handle the overloading.

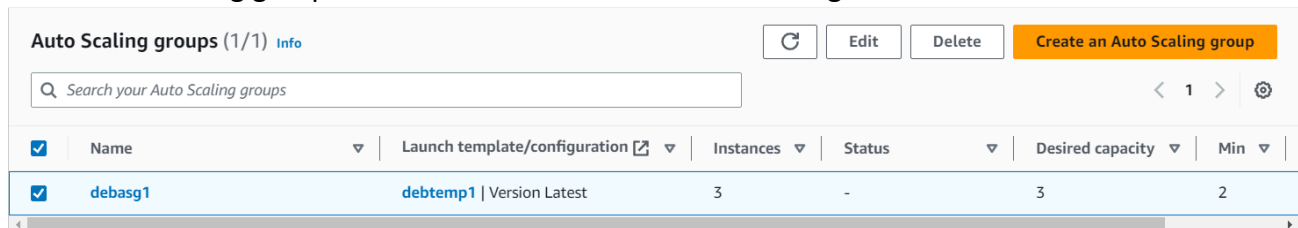
Note that our project webpage was not at all disconnected because of this.

We have successfully created, configured and tested our Auto-Scaling Group.

Now observe that whenever we close or terminate any instance then a new instance gets created. Hence, we cannot delete them if we want to delete them finally.

Follow these steps to cleanly remove Auto-Scaling Groups and everything related to it.

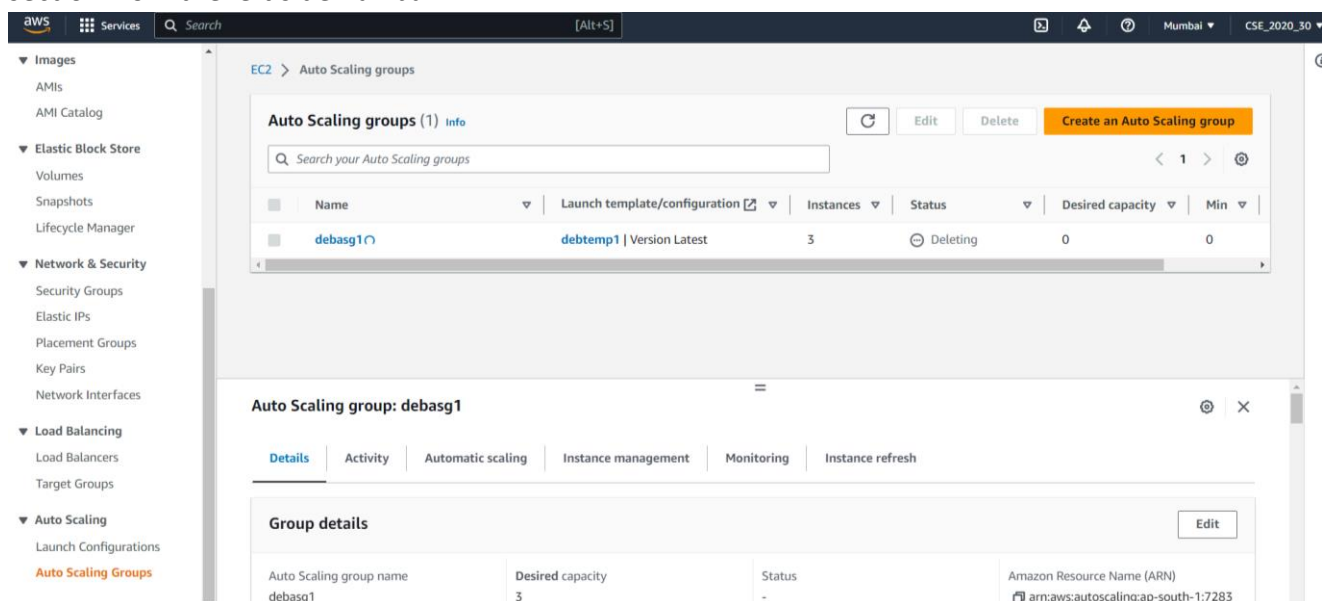
1) Go to Auto-Scaling groups and select the one which we are using.



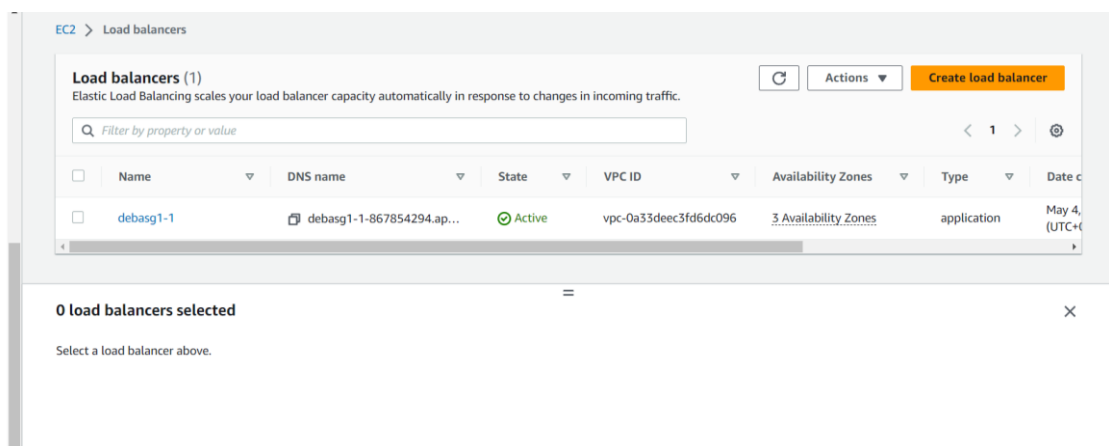
2) Now click on the delete button.

3) Type delete and finally delete it.

4) Notice the status changes to Deleting. It will take some minutes to fully delete. Now go to the Load balancer section from the left side nav bar.



5) Now select the load balancer and click on the action button on the top.



EC2 > Load balancers

Load balancers (1/1) [Info](#)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter by property or value

<input checked="" type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones
<input checked="" type="checkbox"/>	debasg1-1	debasg1-1-867854294.ap...	Active	vpc-0a33deec3fd6dc096	3 Available

Load balancer: debasg1-1

Actions:

- Edit IP address type
- Edit subnets
- Edit instances
- Edit health check settings
- Edit listener
- Edit security groups
- Edit load balancer attributes
- Manage tags
- Delete load balancer

Create load balancer

Now select the Delete Load balancer option to delete it.

6) Now go to the Target Groups section.

7) Select the target group and click on the action button on the top. Select the delete option.

EC2 > Target groups

Target groups (1/1) [Info](#)

Search or filter target groups

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target Type
<input checked="" type="checkbox"/>	debasg1-1	arn:aws:elasticloadbalanci...	4000	HTTP	Instance

Target group: debasg1-1

Details | Targets | Monitoring | Health checks | Attributes | Tags

Details

Actions:

- Delete
- Register targets
- Edit health check settings
- Edit target group attributes
- Manage tags
- Associate with a new load balancer
- Associate with an existing load balancer

Create target group

8) Now go to the instances page.

9) You will find that all the instances created by the Auto-Scaling group will automatically be terminated.

(If not wait for some time. Check if the Auto Scaling Group has been deleted by now.)

Instances (3) [Info](#)

Find instance by attribute or tag (case-sensitive)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	-	i-05f306bab83bb8ec1	Terminated	t2.micro	-	No alarms	ap-south-1b	-
<input type="checkbox"/>	-	i-0a0f3aaf93315ca2d	Terminated	t2.micro	-	No alarms	ap-south-1b	-
<input type="checkbox"/>	-	i-04f330d210730d2a8	Terminated	t2.micro	-	No alarms	ap-south-1a	-

10) Finally check the Auto-Scaling Group section to see if it is completely removed/deleted.

EC2 > Auto Scaling groups

Auto Scaling groups (0) [Info](#)

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
No Auto Scaling groups found in the current region							

Create an Auto Scaling group

0 Auto Scaling groups selected

Select an Auto Scaling group

Everything was deleted successfully.

Hence, we successfully deleted our Auto-Scaling Group and all the load balancers, target groups, instances that was created.