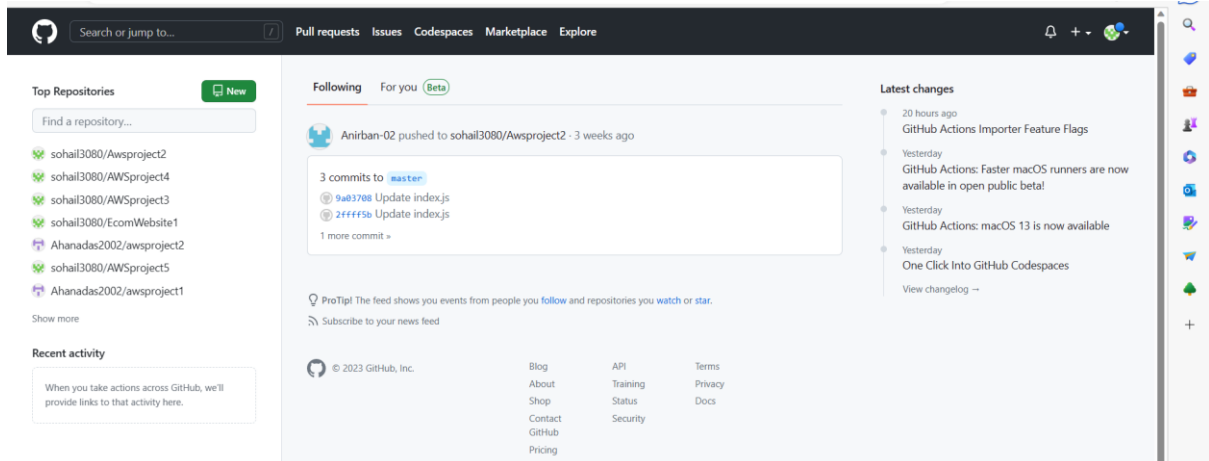


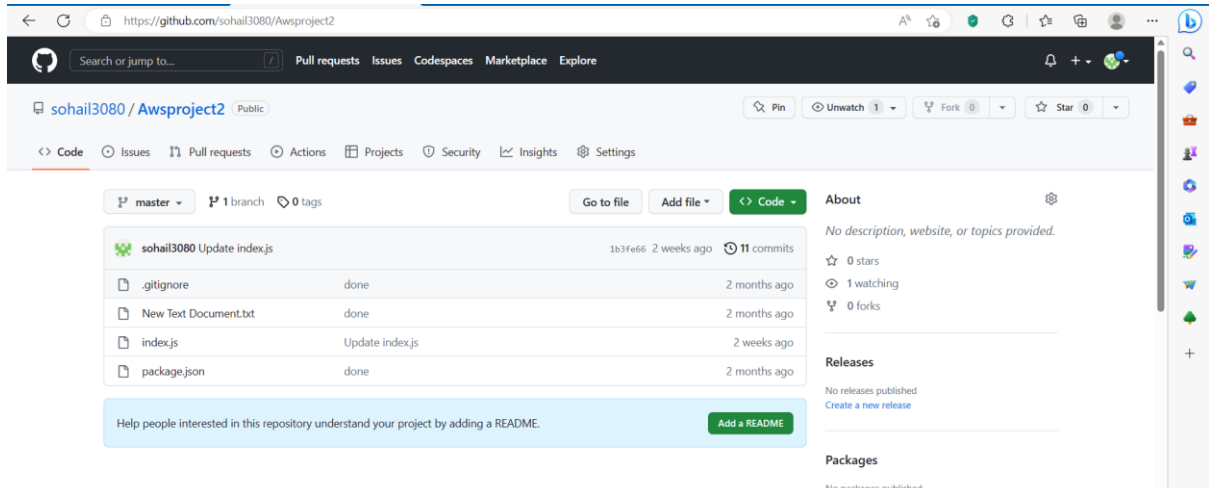
ASSIGNMENT 11

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

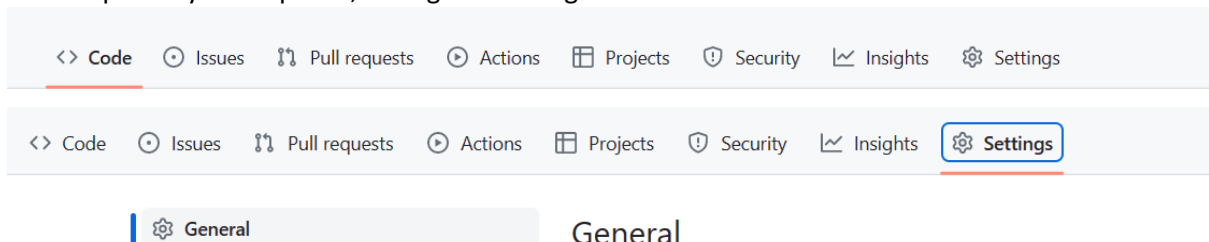
1. Sign in to your GitHub account.



2. Open the Repository which you want to use and make sure it is public.

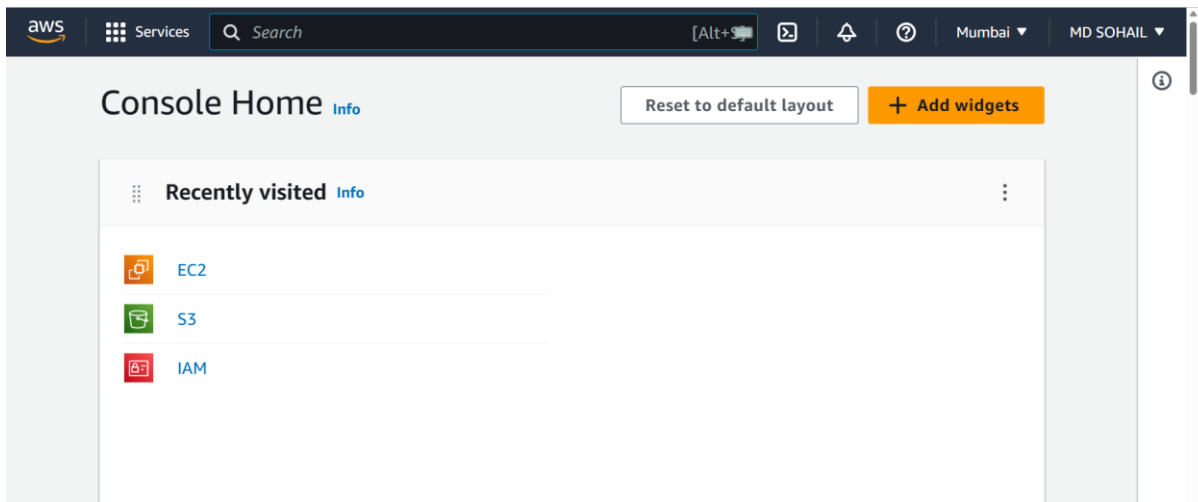


3. If the repository is not public, then go to Settings.

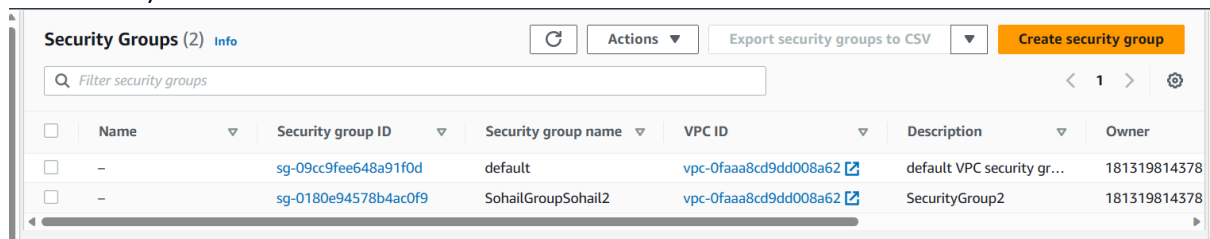


Next, scroll to the bottom in the Danger Zone. Click on Change visibility → Change to Public → I want to make this repository Public → I have read and understand these effects → Make this repository public. At last, give the Password. [My repository is already public so I have not followed these steps]

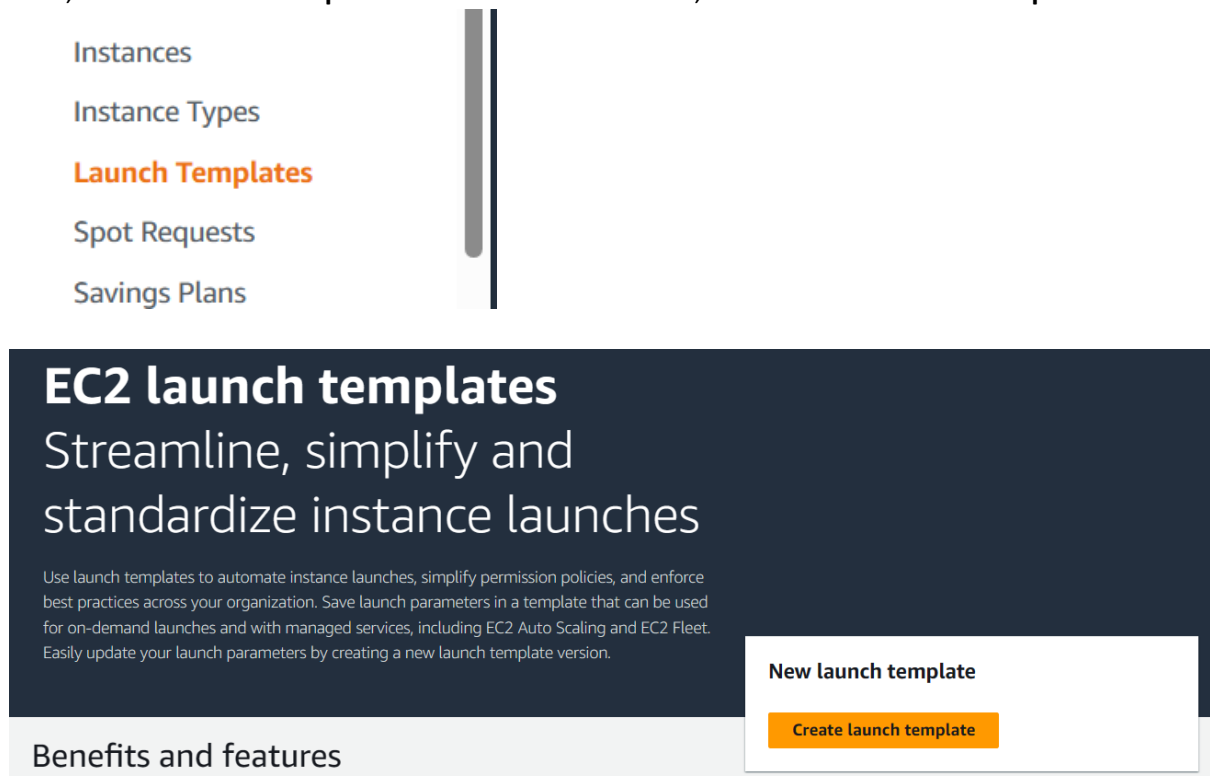
4. Sign in to your AWS account.



- Go to EC2. Make sure you have already created a Security Group (i.e security group other than the default one).



- Next, Click on **Launch Templates** on the left sidebar. Next, Click on **Create launch template**.



- Now, Enter the **Launch template name** and **Template version description**. Also, you may select Provide guidance box.

Launch template name and description

Launch template name - *required*

SohailTemplate1

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

Template version description

ver1

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

8. Next, within the Launch Template Contents, Click on Quick Start tab. Within the Quick start tab, select Ubuntu.

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

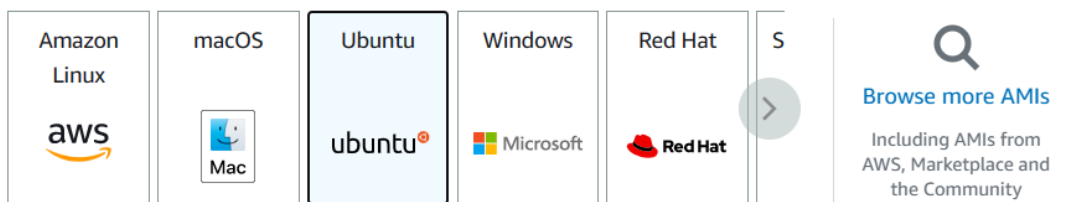
▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 Search our full catalog including 1000s of application and OS images

Recents

Quick Start



Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-02eb7a4783e7e9317 (64-bit (x86)) / ami-0a5dcff6fb7af3fc9 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

9. Select the Instance type as t2.micro.

▼ Instance type
Info
Advanced

Instance type

t2.micro

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

Free tier eligible

☒ All generations
[Compare instance types](#)

10. Select a key pair which you have already created. If not, create a new one.

▼ 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

keypair6

[Create new key pair](#)

11. Within the Network Settings, Click on **Select existing security group**.

▼ 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

SohailGroupSohail2 sg-0180e94578b4ac0f9 X

VPC: vpc-0faaa8cd9dd008a62

[Compare security group rules](#)

▶ Advanced network configuration

12. Next, Click on Advance details and scroll down to the bottom of it until you see the User data field and write the following commands in there.

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/sohail3080/Awsproject2.git
cd Awsproject2
npm install
node index.js
```

13. Click on Create Launch Template.

Cancel

Create launch template

Template was successfully created.



Success

Successfully created [SohailTemplate1 \(lt-008d1382aa90dd7f7\)](#)

► Actions log

14. Next, Go to EC2 and in the left sidebar, Click on Auto Scaling Groups.

▼ Auto Scaling

Launch Configurations

Auto Scaling Groups

15. Now, Click on **Create Auto Scaling group**.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

Create Auto Scaling group

16. Give a Auto Scaling group name.

Name

Auto Scaling group name

Enter a name to identify the group.

AutoScaling1

Must be unique to this account in the current Region and no more than 255 characters.

17. Within the Launch Template field, select the template you have created. Also select the Version to Latest(1).

Launch template [Info](#)

[Switch to launch configuration](#)

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

SohailTemplate1 ▼



[Create a launch template](#)

Version

Latest (1) ▼



[Create a launch template version](#)



Then Click on Next.

Cancel

Next

18. Now, within the Network field. Choose all Availability Zones and subnets.

FOR GETTING STARTED QUICKLY

<input checked="" type="checkbox"/>	ap-south-1a subnet-002b8fd86a875a861 172.31.32.0/20 Default	for your Aut
<input checked="" type="checkbox"/>	ap-south-1b subnet-0aeb810e2cd6851bc 172.31.0.0/20 Default	
<input checked="" type="checkbox"/>	ap-south-1c subnet-0f56dd852c5386b80 172.31.16.0/20 Default	Auto Scaling
Select Availability Zones and subnets ▲		

ap-south-1a | subnet-002b8fd86a875a861
172.31.32.0/20 Default

ap-south-1b | subnet-0aeb810e2cd6851bc
172.31.0.0/20 Default

ap-south-1c | subnet-0f56dd852c5386b80
172.31.16.0/20 Default

[Create a subnet](#)

Then, Click on Next.

Cancel

Skip to review

Previous

Next

19. Next, within the Load balancing field, select **Attach to a new load balancer**.

Load balancing [Info](#)

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. Select the Load balancer type as Application Load Balancer(default). Enter the Load balancer name (default AutoScaling1-1). Select the Load balancer scheme as Internet-facing.

Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, [visit the Load Balancing console](#). [↗](#)

☒ Application Load Balancer
HTTP, HTTPS

☐ Network Load Balancer
TCP, UDP, TLS

Load balancer name

Name cannot be changed after the load balancer is created.

AutoScaling1-1

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

21. Enter the Port number of Listeners and routing as 4000. [According to our project we have taken 4000]

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	Port	Default routing (forward to)
HTTP	4000	Select new or existing target group

22. Now, Click on Select a new or existing target group → Create a target group. New target group name should be AutoScaling1-1 according to our project.

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	Port	Default routing (forward to)
HTTP	4000	Create a target group

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

AutoScaling1-1

23. Set Health check grace period to 300 seconds.

Health check grace period [Info](#)

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

Next, Click on Next.

24. In the Group size field.

Set Desired capacity to 2, Minimum capacity to 2, Maximum capacity to 3.

Group size - optional [Info](#)

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

25. Now, Click on **Target tracking scaling policy**. Set the Target value to **50** and instance need to **300** seconds warm up before including in metric.

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

Average CPU utilization ▼

Target value

Instances need

seconds warm up before including in metric

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

Click on Next.

Instance scale-in protection - optional

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

Cancel

Skip to review

Previous

Next

26. Click on Next.

Step 5 of 7

Add notifications - *optional* [Info](#)

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

[Add notification](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

27. Click on Next.

Tags (0)

[Add tag](#)

50 remaining

[Cancel](#) [Previous](#) [Next](#)

28. Scroll down on the Review page.

Step 7 of 7

Review [Info](#)

At the bottom, Select Create Auto Scaling group.

[Cancel](#) [Previous](#) [Create Auto Scaling group](#)

29. You can see the Auto Scaling Group was created successfully.

Auto Scaling groups (1) [Info](#)

[Refresh](#) [Edit](#) [Delete](#) [Create an Auto Scaling group](#)

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	AutoScaling1	SohailTemplate1 Version Latest	0	Updating capacity...	2	2

30. Go to EC2 Instances. You can see two instances running. [Wait, It may take some time]

Instances (2) Info							
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/> Instance state = running Clear filters							
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z
<input type="checkbox"/>	-	i-05db78a325ae114ca	Running	t2.micro	Initializing	No alarms	ap-south-1b
<input type="checkbox"/>	-	i-00417602ee47d373a	Running	t2.micro	Initializing	No alarms	ap-south-1a

31. Go to the first instance, copy its Ipv4 address and paste it in the URL bar of the browser.

Public IPv4 address copied

3.109.56.144 | [open address](#)

Instance state
Running

Not secure | 3.109.56.144

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Add the Port Number 4000 at the end of the URL as follows.

Not secure | 3.109.56.144:4000

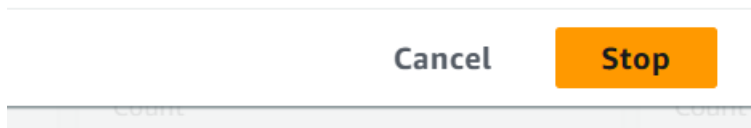
Hello Sohail

32. Now, Stop both the instances for sometime.

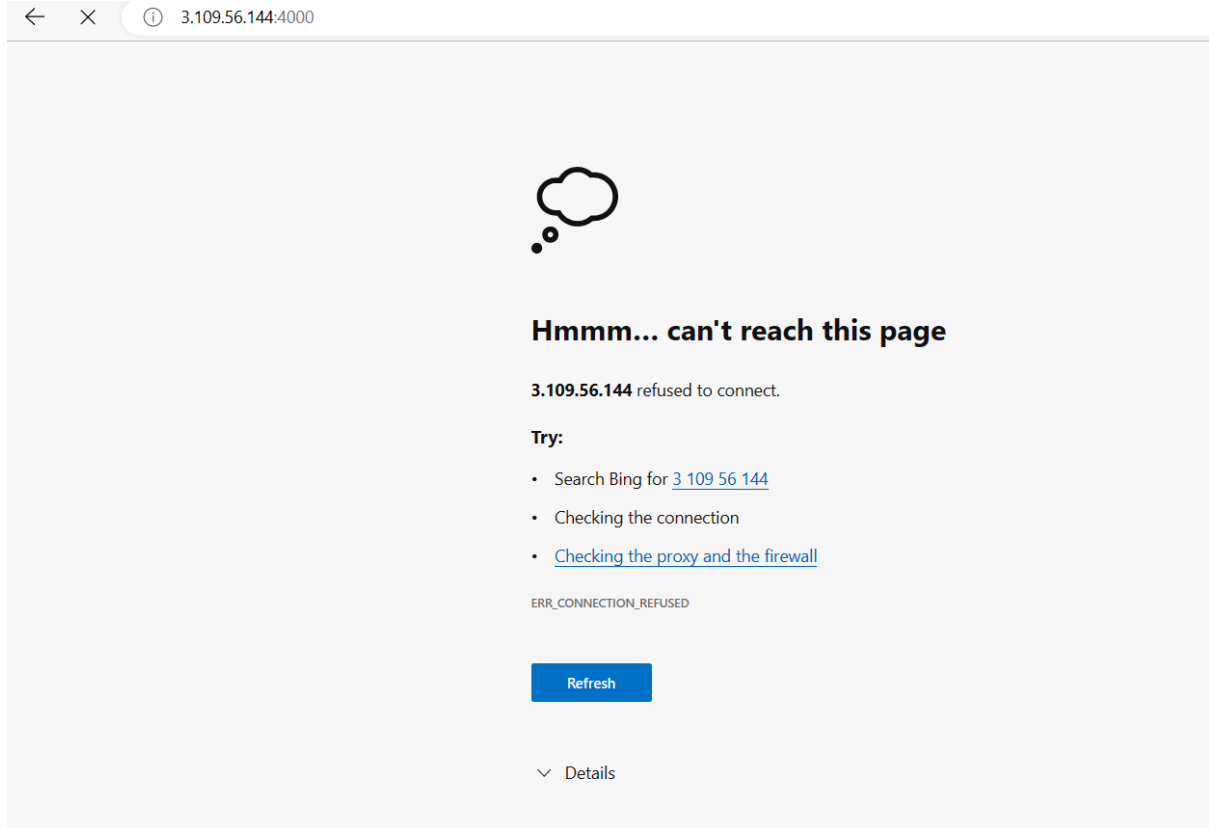
Instances (2/2) Info							
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/> Instance state = running Clear filters							
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z
<input checked="" type="checkbox"/>	-	i-05db78a325ae114ca	Running	t2.micro	2 checks passed	No alarms	ap-south-1b
<input checked="" type="checkbox"/>	-	i-00417602ee47d373a	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a

Click on Stop.

instances, choose the *Stop* button below.



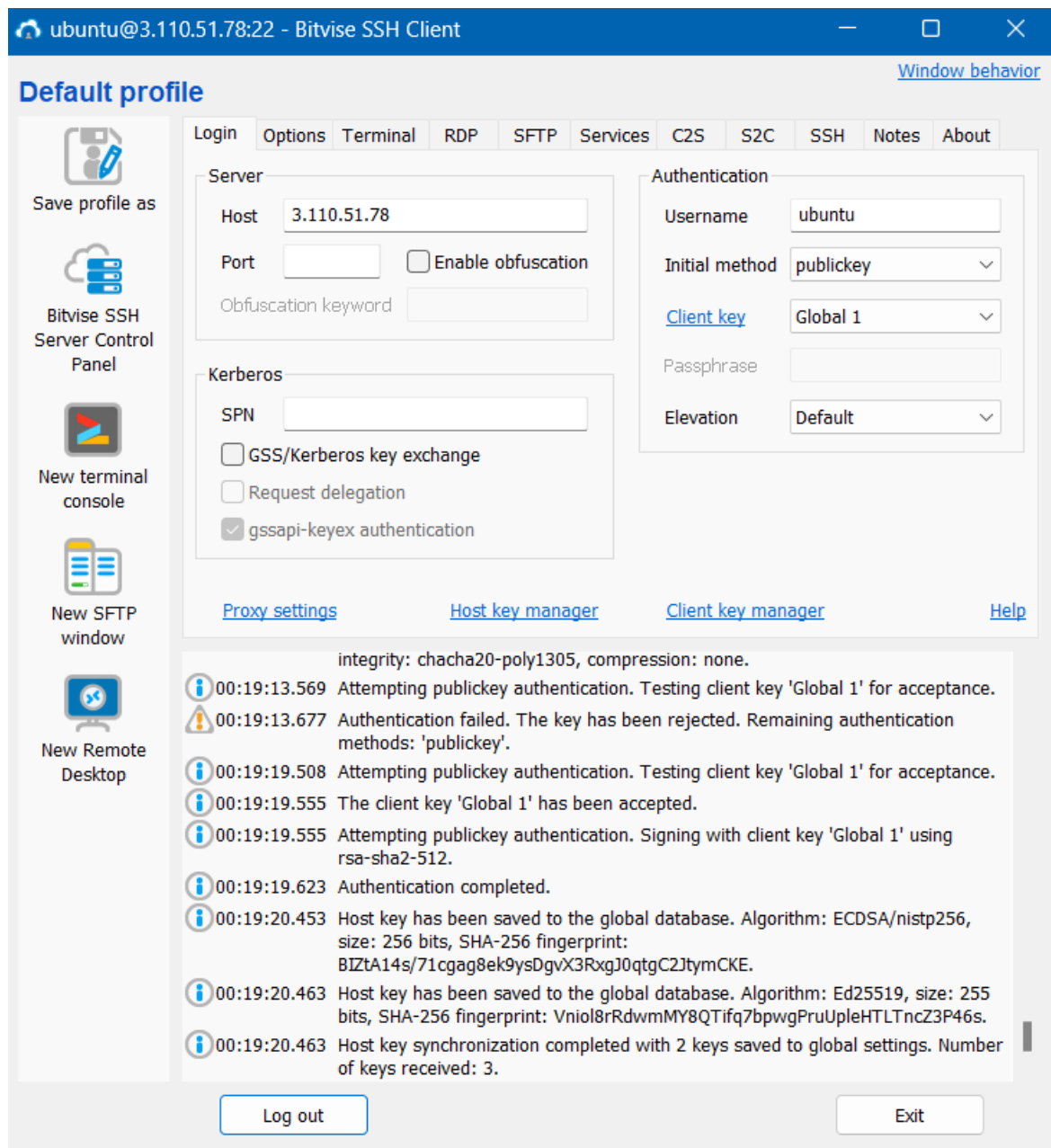
33. As the servers have stopped, our page will not show anymore.



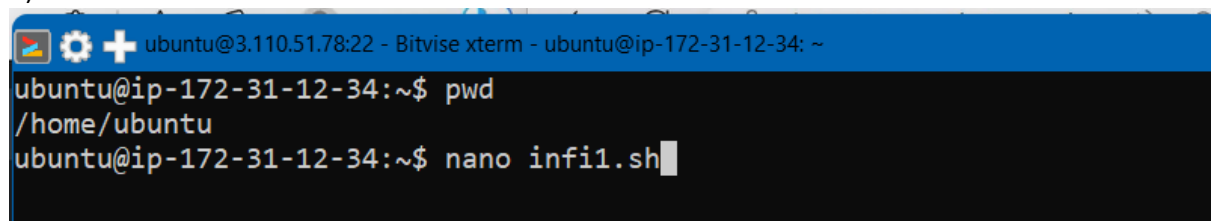
34. After some time, new instances got automatically created.

Instances (2) Info								
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>								
Instance state = running Clear filters								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	-	i-010d8157e9f81785b	Running	t2.micro	Initializing	No alarms +	ap-south-1b	ec2-3-11
<input type="checkbox"/>	-	i-0d4c341c6ea5b7669	Running	t2.micro	Initializing	No alarms +	ap-south-1a	ec2-65-

35. Click on the first instance and connect It to Bitwise SSH Client.



36. Open New terminal console.
Type the following commands respectively.
- `pwd`
 - `nano inf1.sh`



- `#!/bin/bash`
`while true`
`do`
 `echo "Looping forever"`
 `#Add other commands to run in the loop here`
`done`

done

```
ubuntu@3.110.51.78:22 - Bitvise xterm - ubuntu@ip-172-31-12-34: ~
GNU nano 6.2 infi1.sh *
#!/bin/bash
while true
do
    echo "Looping forever"
    #Add other commands to run in the loop here
done
```

Now Press Ctrl+X, Ctrl+Y and Enter respectively.

d)chmod +x infi1.sh

```
ubuntu@ip-172-31-12-34:~$ chmod +x infi1.sh
```

e) ./infi1.sh

```
ubuntu@ip-172-31-12-34:~$ ./infi1.sh
```

Runs Infinite Loop.

```
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
Looping forever
```

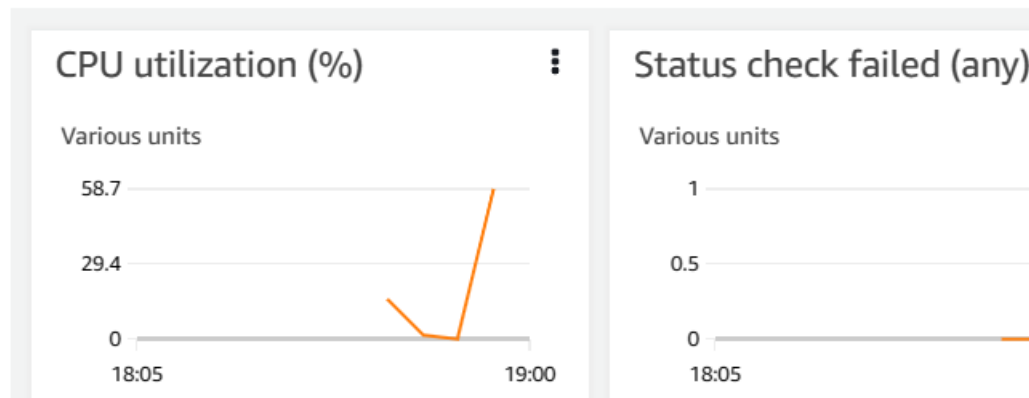
37. Now, go to your Instances in AWS and select both of them.

Instances (2/2) Info									
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>									
<input type="text" value="Instance state = running"/> <input type="button" value="Clear filters"/>									
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public	
<input checked="" type="checkbox"/>	-	i-0ab42305f501a4d4	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3	
<input checked="" type="checkbox"/>	-	i-00fce3a53b2ea0a7d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3	

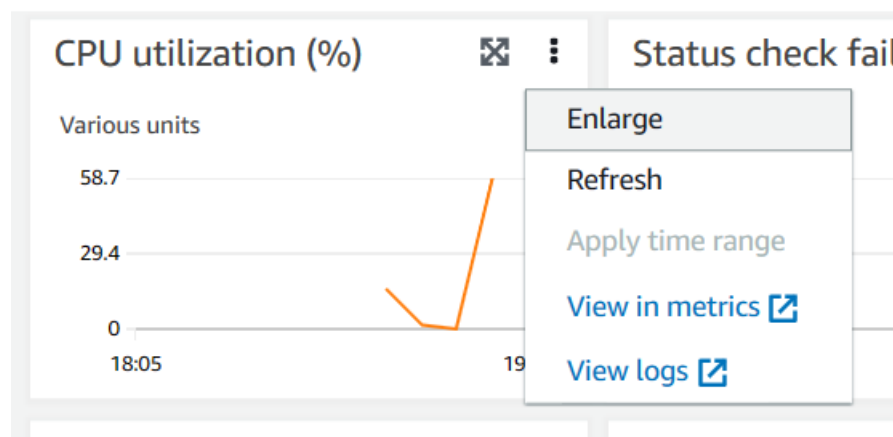
38. Click on Monitoring.

Instances: i-04ba6c4ba88f135b9, i-0ab423055f501a4d4

Monitoring



And in CPU utilization (%), click on Enlarge.



39. Click on Custom→Select UTC on the right upper side.

5 minutes ▼ Average ▼ 1h 3h 12h 1d 3d 1w Custom

Absolute **Relative** UTC ▼

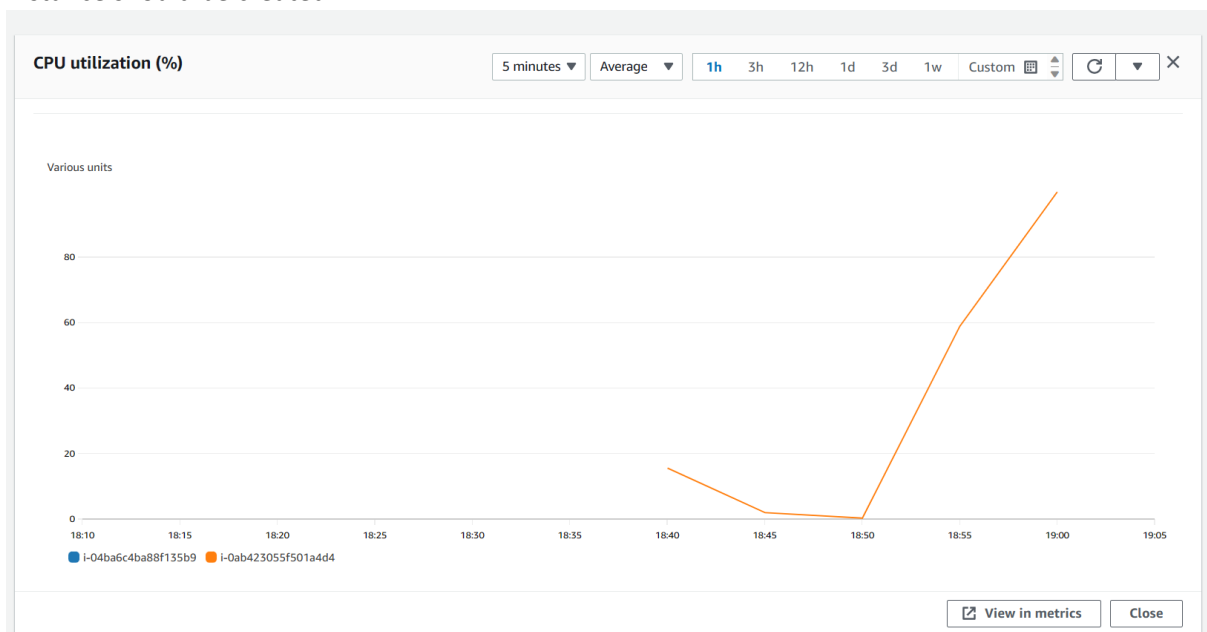
Minutes	1	3	5	15	30	45
Hours	1	2	3	6	8	12
Days	1	2	3	4	5	6
Weeks	1	2	4	6		
Months	3	6	12	15		

60 Minutes ▼

Clear Cancel **Apply**

Click on **Apply**.

40. As you can see, according to what we have set, the CPU utilization (%) is more than 50, so a new instance should be created.

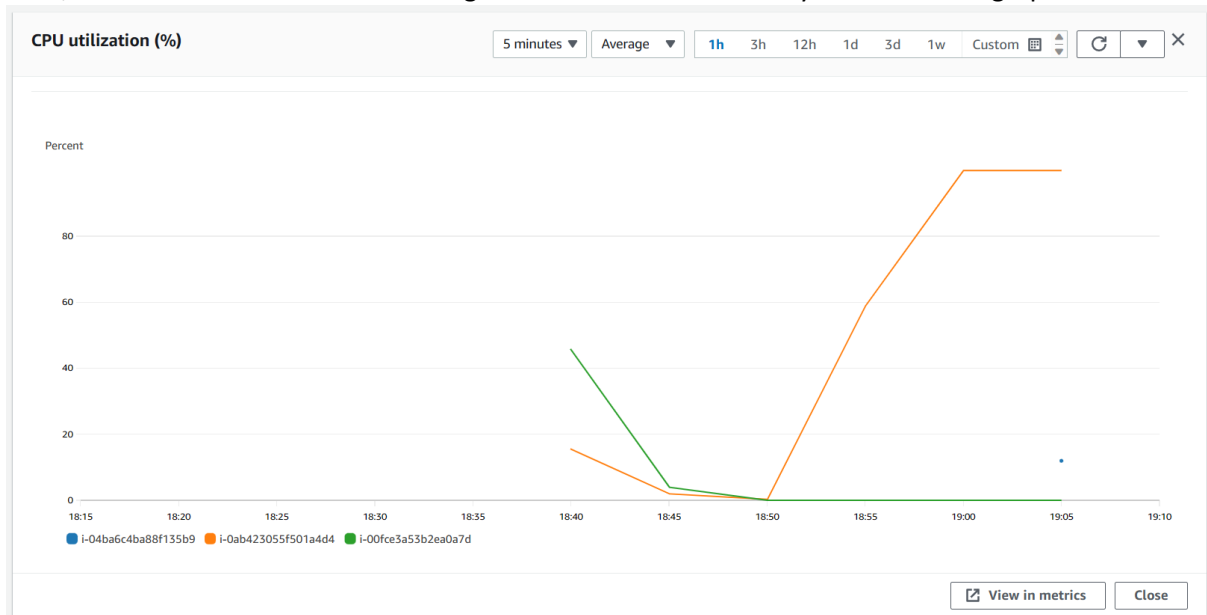


Let us check whether a new Instance is created or not.

41. As you can see, we have a new instance.

Instances (3) Info								
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/> Instance state = running Clear filters								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	-	i-04ba6c4ba88f135b9	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-10...
<input type="checkbox"/>	-	i-0ab423055f501a4d4	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-10...
<input type="checkbox"/>	-	i-00fce3a53b2ea0a7d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3-10...

42. Now, select all the instances and enlarge the CPU utilization. Here you can see the graph of them.



Hence, When the load was above the set value, it was balanced out by creating a new Instance.

43. To close

a) first, we have to remove Auto Scaling group.

Auto Scaling groups (1/1) Info					
<input type="text" value="Search your Auto Scaling groups"/>					
<input checked="" type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity
<input checked="" type="checkbox"/>	AutoScaling1	SohailTemplate4 Version Latest	3	-	3

Delete Auto Scaling group

Auto Scaling group contains running instances

Deleting these Auto Scaling groups will terminate all instances in each group. This action cannot be undone.

Are you sure you want to delete this Auto Scaling group?

- AutoScaling1

Deleting the Auto Scaling group will terminate all instances in the group. This action cannot be undone.

To confirm deletion, type *delete* in the field.

delete

Cancel
Delete

b) Then Load Balancer

Load balancers (1/1)
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
<input checked="" type="checkbox"/>	AutoScaling1-1	AutoScaling1-1-12729366...	Active	vpc-0faaa8cd9dd008a62

Actions
Create load balancer

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

Load balancer: AutoScaling1-1

c) Then Target Group.

EC2 > Target groups

Target groups (1/1) Info

Search or filter target groups

<input checked="" type="checkbox"/>	Name	ARN	Port
<input checked="" type="checkbox"/>	AutoScaling1-1	arn:aws:elasticloadbalanci...	4000

Actions
Create target group

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

d) And at last EC2.

Instances (3/3) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running X Clear filters

Refresh

Connect

Instance state ▲

Actions ▼

Launch instances ▼

Stop instance

Start instance

Reboot instance

Hibernate instance

Terminate instance

1

⌵

⌶

<input checked="" type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Health status	Alarm status	Availability Zone ▼	Public IP
<input checked="" type="checkbox"/>	-	i-04ba6c4ba88f135b9	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-10...
<input checked="" type="checkbox"/>	-	i-0ab423055f501a4d4	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-10...
<input checked="" type="checkbox"/>	-	i-00fce3a53b2ea0a7d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3-10...

Terminate instances?

X

⚠ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

🗑 i-04ba6c4ba88f135b9

🗑 i-0ab423055f501a4d4

🗑 i-00fce3a53b2ea0a7d

To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.

Cancel

Terminate