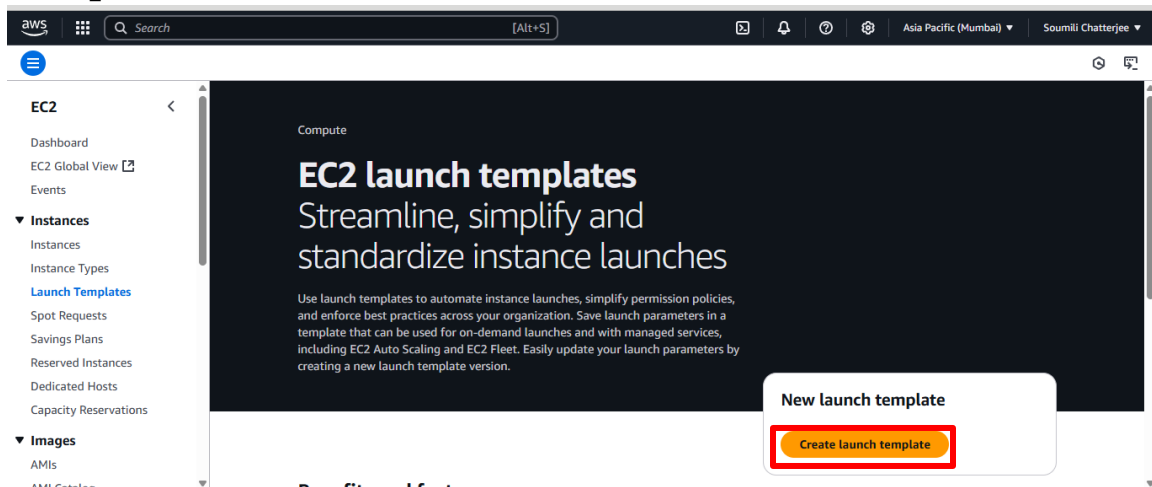


## Assignment No. – 11

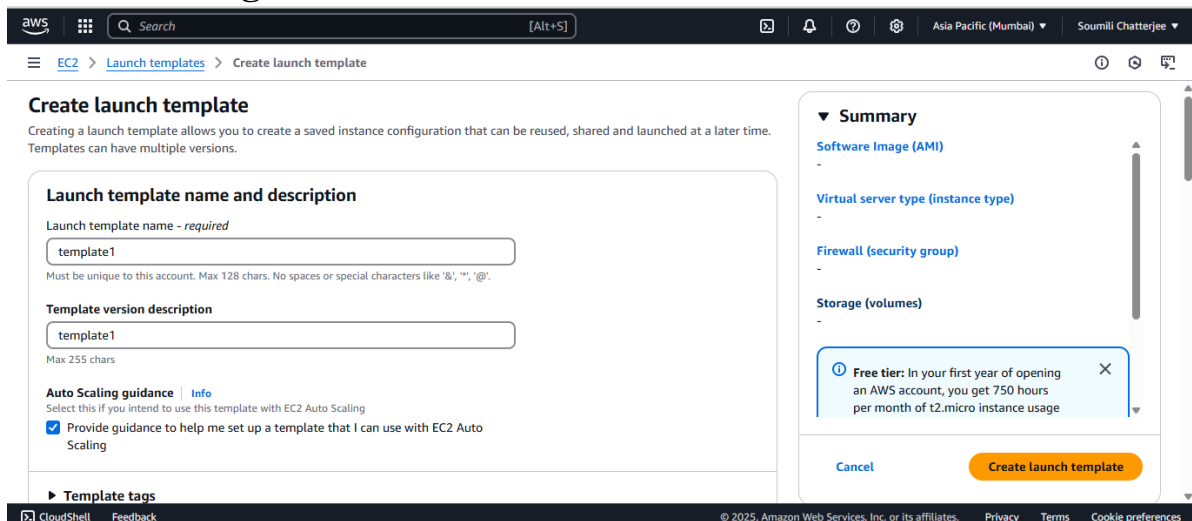
**Problem Statement:** Build scaling plans in AWS that balance the load on different EC2 instances.

### **Procedure:**

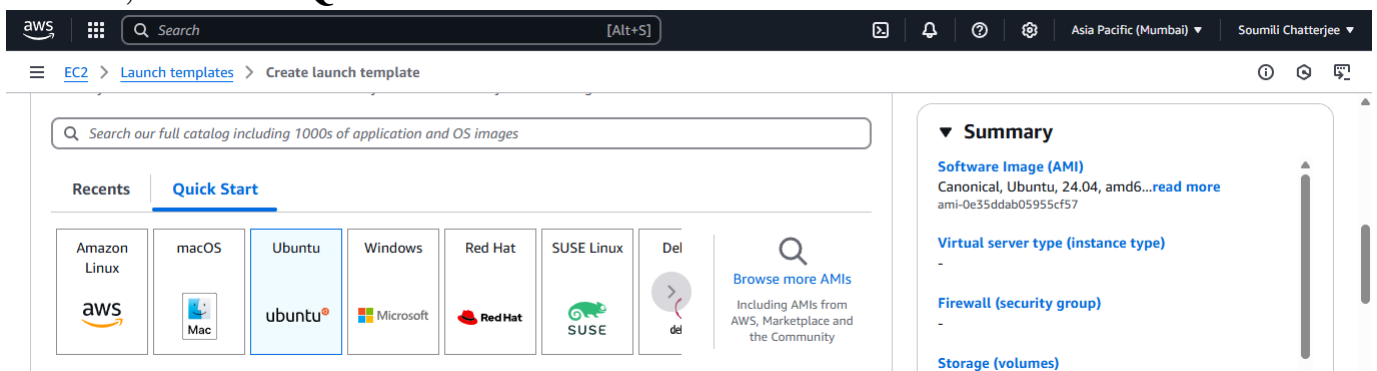
1. **Sign In** to the **AWS** account.
2. Go to **EC2**.
3. Under **Instances** section click on **Launch Templates**. Next, click on **Create launch template**.



4. Now, give a name to the template and write the same in the description below and check the “**Provide guidance**” box.



5. Next, under the **Quick Start** menu select **Ubuntu** as the OS.



6. Next, under **Instance type** select **t2.micro** type of configuration.

▼ **Instance type** Info | Get advice Advanced

**Instance type**

**t2.micro** Free tier eligible

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

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

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

aws Search [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Launch templates > Create launch template

launch the instance:

Key pair name: key1 [Create new key pair](#)

▼ **Network settings** Info

Subnet: 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: mysecurity sg-00e8288ed8abe1cd3 [Compare security group rules](#)

mysecurity vpc: vpc-06dd141e6310b8408

▼ **Summary**

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd64...read more  
ami-0e35ddab05955cf57

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
mysecurity

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

Free tier: In your first year of opening an AWS account, you get 750 hours

[Cancel](#) [Create launch template](#)

8. Now, click on the **Advanced Details** at the bottom. Scroll down to **User Data** Section and paste the commands in the box provided and click on the **Create Launch Template** button.

aws Search [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Launch templates > Create launch template

User data - optional Info

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/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_16.x | sudo -E bash -
apt-get install -y nodejs
git clone http://github.com/sudip7407/Repo1.git
cd Repo1
npm install
node index.js
```

☐ User data has already been base64 encoded

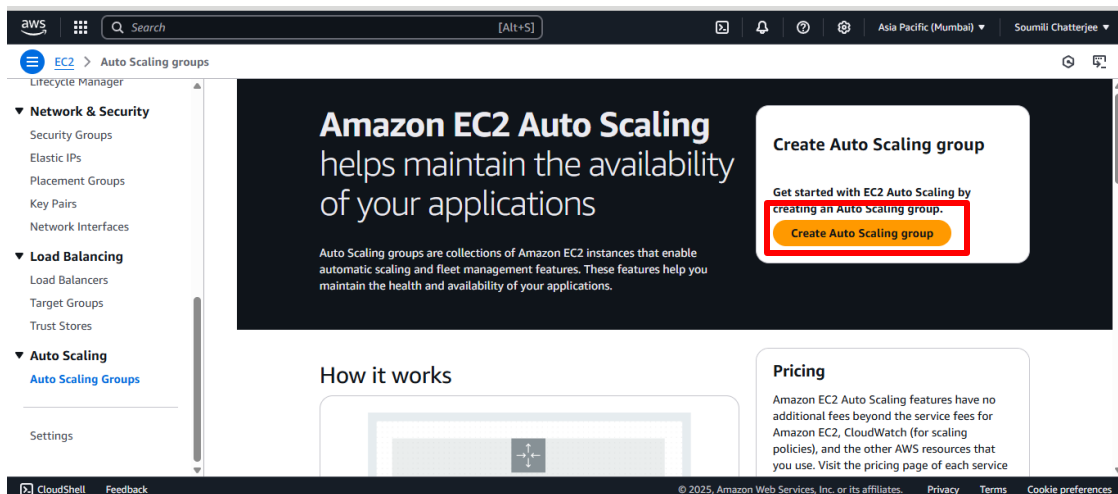
▼ **Summary**

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

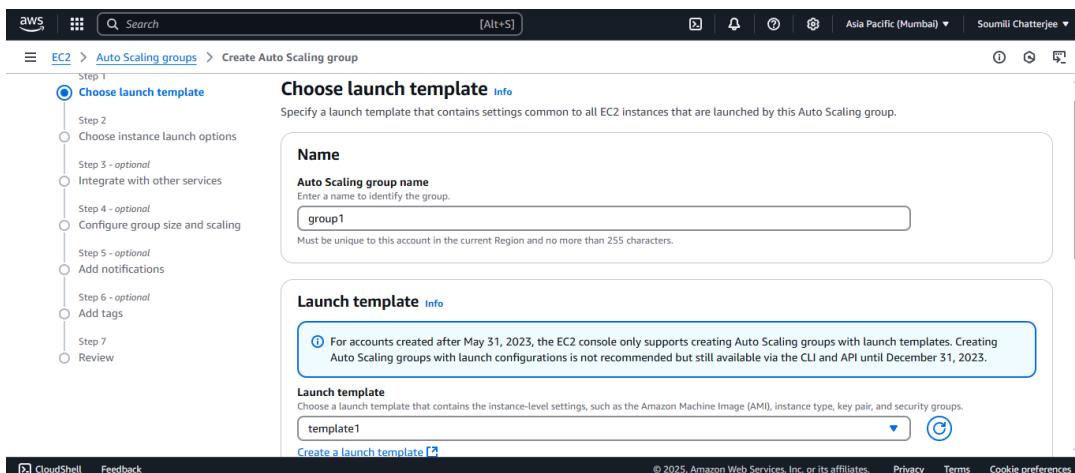
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Create launch template](#)

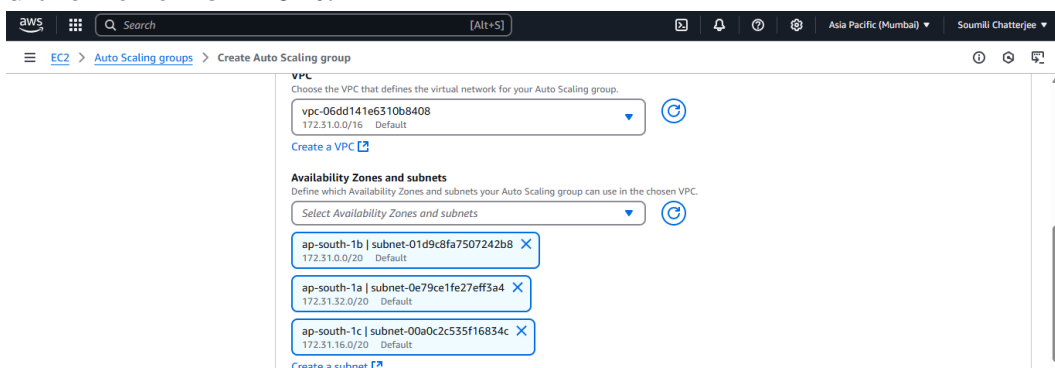
9. Go back to EC2 and from the Left side navigation bar go to **Auto Scaling Groups** under **Auto-Scaling** section and click on **Create Auto Scaling Group**.



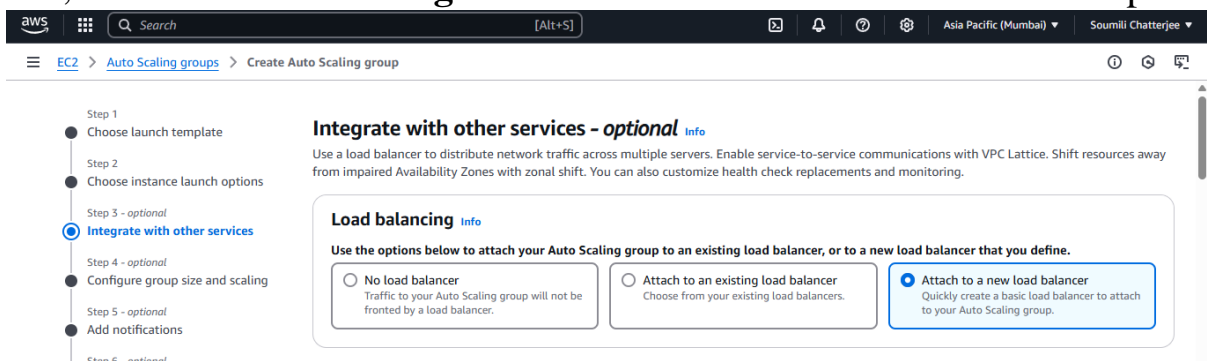
10. Give a **name** to your new Auto Scaling Group. Also, select the Launch Template that we recently created by using the drop-down menu under **Launch Templates** section and then click on **next**.



11. After that, under **Availability Zones and Subnets** select all the zones that appear and then click on **next**.



12. Now, under **Load Balancing** select **Attach to a New Load balancer** option.



13. Now, select **Internet-Facing** under Load balancer scheme.

**Load balancer name**

Name cannot be changed after the load balancer is created.

group1-1

**Load balancer scheme**

Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

14. Under **Listeners and Routing** enter the port number (4000) and select **Create target group**.

**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.

group1-1

15. Next, under **Health checks** section, check the **turn of elastic load balancing health checks** box and change the health check grace period to **200** seconds and then click on **next**.

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Auto Scaling groups > Create Auto Scaling group

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

**EC2 health checks**  
[Always enabled](#)

**Additional health check types - optional** [Info](#)

☒ **Turn on Elastic Load Balancing health checks** **Recommended**  
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

☐ Turn on VPC Lattice health checks  
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

☐ Turn on Amazon EBS health checks  
EBS monitors whether an instance's root volume or attached volume stalls. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance on its next periodic health check.

**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.

200 seconds

16. Under Group Size mention:

Desired Capacity = 2

Minimum Capacity = 2

Maximum Capacity = 3

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 3 - optional: Integrate with other services  
Step 4 - optional: **Configure group size and scaling**  
Step 5 - optional: Add notifications  
Step 6 - optional: Add tags  
Step 7: Review

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

**Desired capacity type**  
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

**Desired capacity**  
Specify your group size.

2

**Scaling** [Info](#)  
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

**Scaling limits**  
Set limits on how much your desired capacity can be increased or decreased.

**Min desired capacity**  
2  
Equal or less than desired capacity

**Max desired capacity**  
3  
Equal or greater than desired capacity

17. Now, under **Scaling policies** choose the **Target Tracking Scaling policy** option and set the **instance warmup** to **200** seconds and then click on **next**.

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Auto Scaling groups > Create Auto Scaling group

**Automatic scaling - optional**

**Choose whether to use a target tracking policy** | Info  
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☐ No scaling policies  
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☒ **Target tracking scaling policy**  
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

**Scaling policy name**  
Target Tracking Policy

**Metric type** | Info  
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

**Target value**  
50

**Instance warmup** | Info  
200 seconds

18. Nothing to change in **Notifications and Tags** page. So, click on the **Next** button. Under **Review** page, click on the **Create Auto-Scaling Group**.

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Auto Scaling groups

**Auto Scaling groups (1)** | Info  
Last updated 3 minutes ago

Launch configurations Launch templates Actions Create Auto Scaling group

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	
<input type="checkbox"/>	group1	template1   Version Default	2	-	2	2	3	a...

19. Again go back to EC2 and from the Left side navigation bar go to **Load Balancers** under **Load Balancing** section.

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Load balancers

**Load balancers (1)** | Info  
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

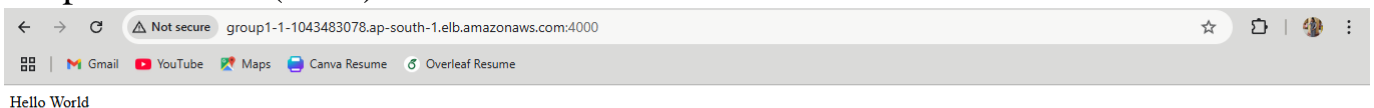
Filter load balancers

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	group1-1	group1-1-1043483078.ap-south-1.elb.amazonaws.com	Active	vpc-06dd141e6310b8408	3 Availability Zones	application

0 load balancers selected

Select a load balancer above.

20. Copy the **DNS name** and paste it in a new window followed by a colon and then the port number (4000).



21. We will now **overload the server instances** and for that we need **Bitwise SSH client** for **instance 1**.

### For instance 1:

- Copy the public IPv4 address.
- Open Bitvise SSH client.
- Paste the IP and specify the necessary options.
- Now Log-In to the server.
- Open the new Terminal.
- Now enter the command:

**sudo nano infy.sh**

- After the command a new nano Editor window will open. Type the following:

**#!/bin/bash**

**while(true)**

**do**

**echo "Inside loop"**

**done**

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

**Ctrl+X**

**Y**


**Enter**

- Now in the terminal type the following commands:

**sudo chmod +x infy.sh**

**sh infy.sh**

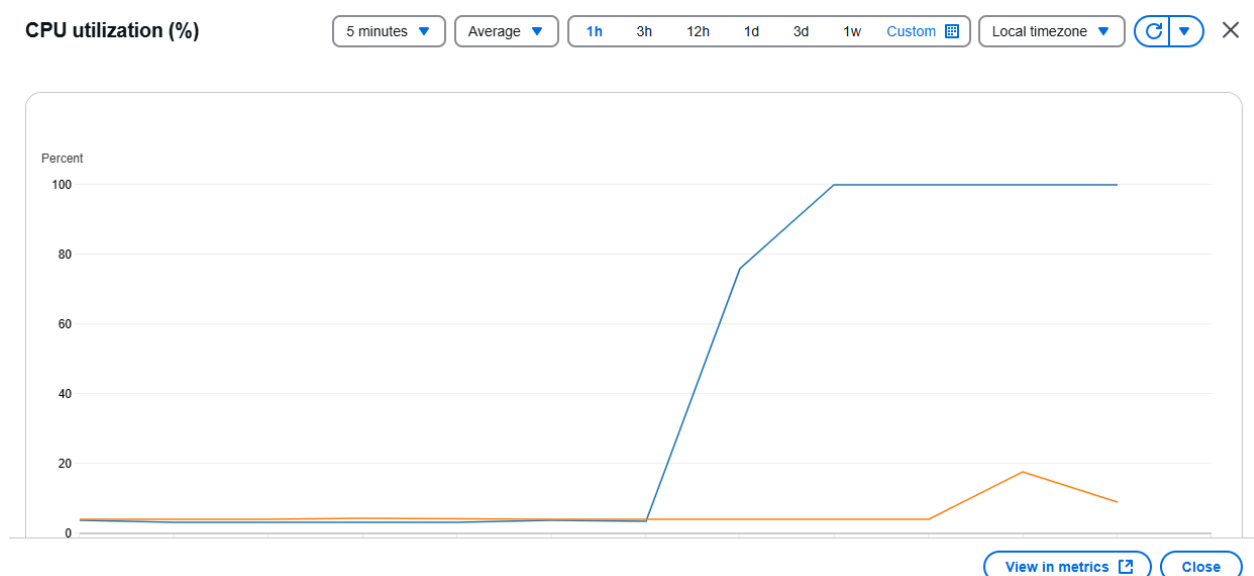




```
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
Inside loop
```

Now the script will start running infinitely. Do not close the terminal. Keep it Minimized and go to Ec2.

22. Now, select both the running instances and under **monitoring** go to **CPU Utilisation** and **enlarge** the window by clicking on the three dots. Choose the timezone as **Local Timezone**.



23. After some time, we can see there are 3 servers running. 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.

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Instances

EC2

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Instances (3) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>		i-05decc83e0c8e5c9c	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a
<input type="checkbox"/>		i-076bdba696b9aa36d	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b
<input type="checkbox"/>		i-0755dbe10bcc13330	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b

aws [Search] [Alt+S] Asia Pacific (Mumbai) Soumili Chatterjee

EC2 > Instances

EC2

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Instances (3) Info

Last updated 3 minutes ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Running

Security group name	Key name	Launch time	Platform...	Managed	Operator
mysecurity	key1	2025/04/25 01:22 GMT+5:30	Linux/UNIX	false	-
mysecurity	key1	2025/04/25 01:21 GMT+5:30	Linux/UNIX	false	-
mysecurity	key1	2025/04/25 02:29 GMT+5:30	Linux/UNIX	false	-

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