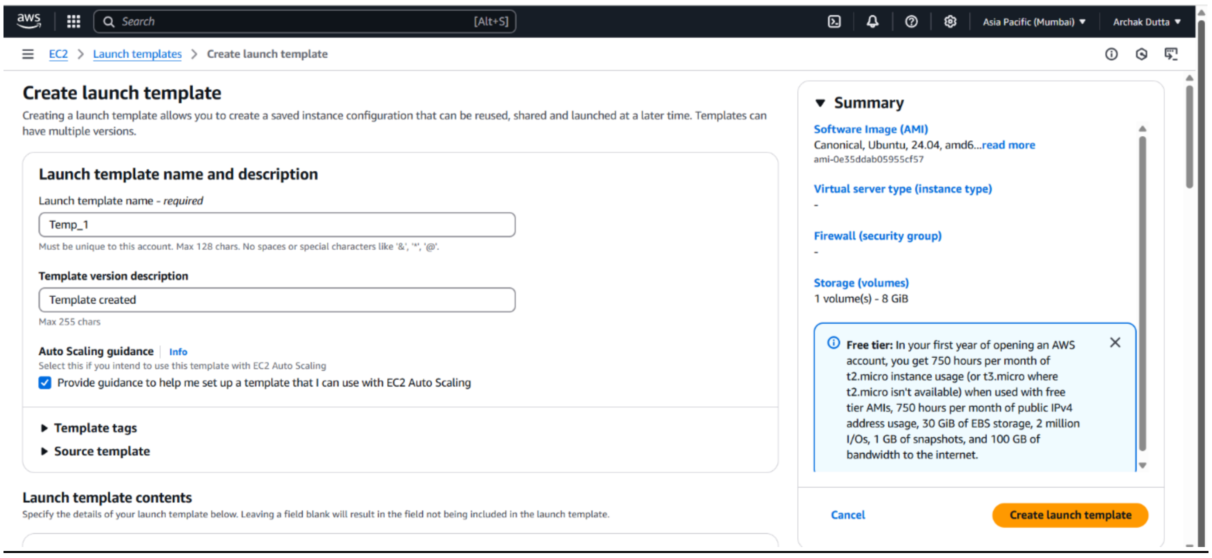
<u>Assignment No</u>: 11

<u>Problem Statement</u>: Build scaling plans in AWS that balance the load on different EC2 instances.

<u>Solution</u>: To build different scaling plans in AWS, the steps are-

1.Sign into AWS console, go to EC2 and select **launch template**.Provide the **template name** along with a **description** and the auto scaling option is to be checked.Then select **ubuntu** as OS in quick start.



EC2 > Launch templates > Create launch template

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*

Temp_1

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

Template version description

Template created

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

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.

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e35ddab05955cf57

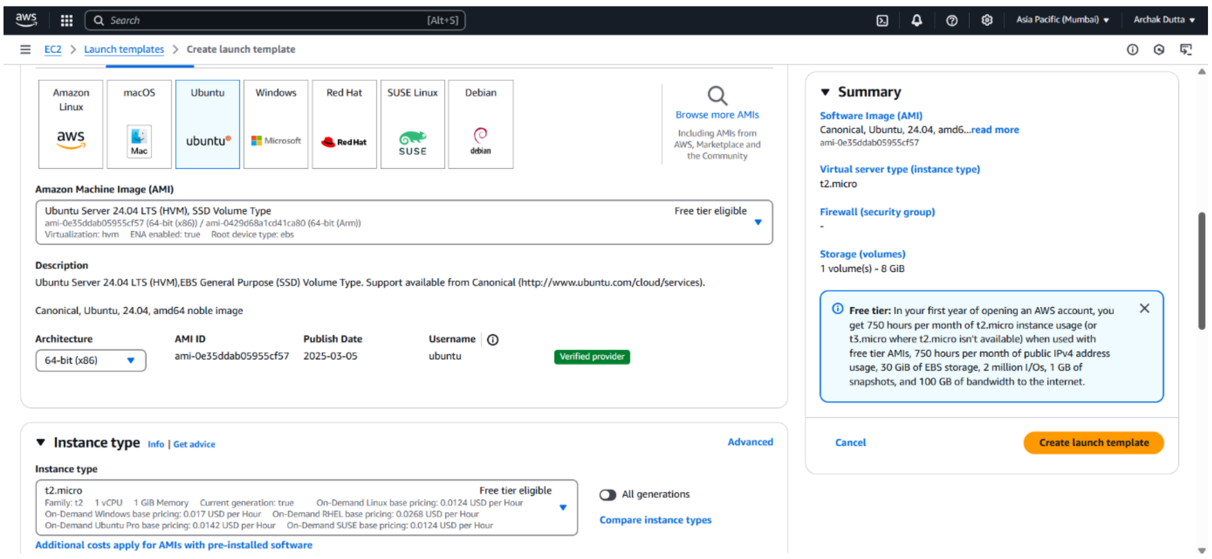
Virtual server type (instance type)
-

Firewall (security group)
-

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.

2.Select instance type as **t2.micro**.



EC2 > Launch templates > Create launch template

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0e35ddab05955cf57 (64-bit (x86)) / ami-0429d68a1cd41ca80 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username
64-bit (x86)	ami-0e35ddab05955cf57	2025-03-05	ubuntu

Verified provider

Instance type Info | Get advice

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

Additional costs apply for AMIs with pre-installed software

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e35ddab05955cf57

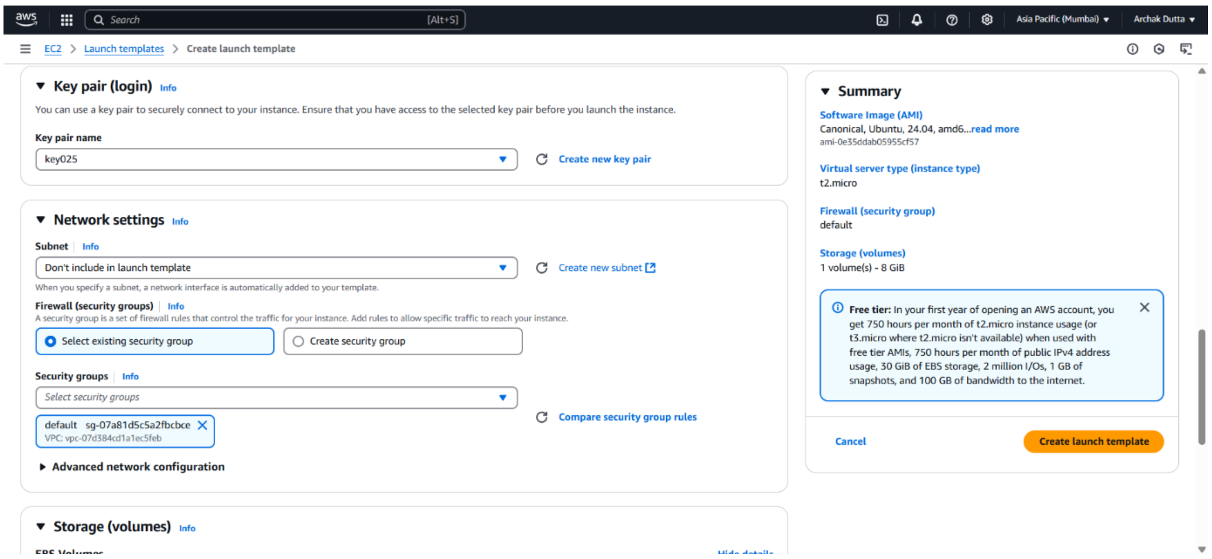
Virtual server type (instance type)
t2.micro

Firewall (security group)
-

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.

3.Select a key pair along with the existing security group from network section.Provide the user data under additional settings.Select create launch template and the template is created.


EC2 > Launch templates > Create launch template

Key pair (login)

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: key025

Network settings

Subnet: Don't include in launch template

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

Firewall (security groups)

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: default sg-07a81d5c5a2fbcbce
VPC: vpc-07d384cd1a1ec5feb

Advanced network configuration

Storage (volumes)

Summary

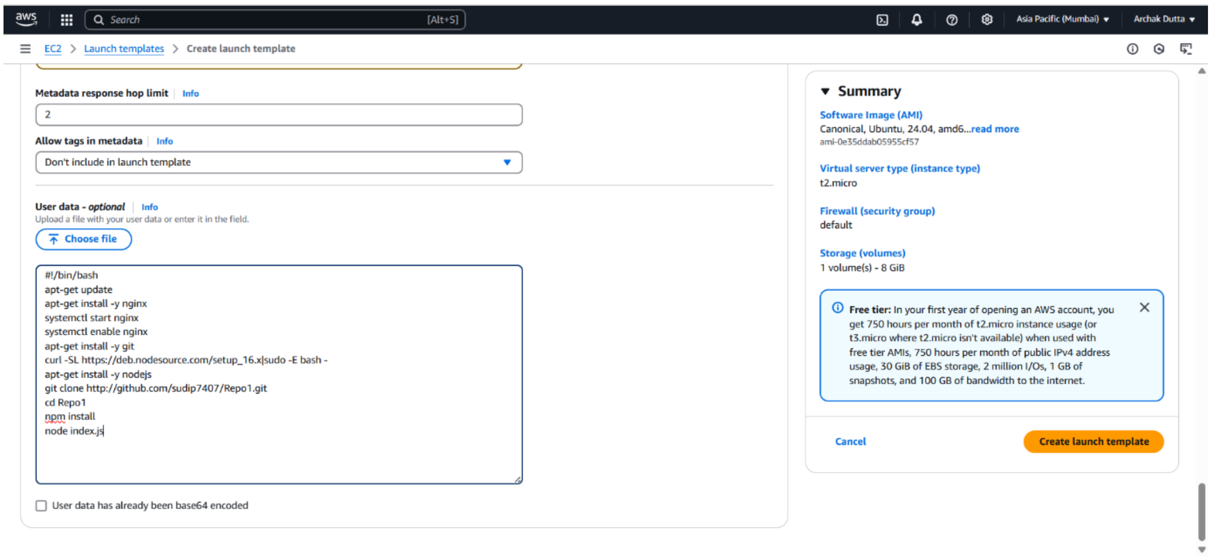
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0e35ddab05955cf57

Virtual server type (instance type): t2.micro

Firewall (security group): default

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.


EC2 > Launch templates > Create launch template

Metadata response hop limit: 2

Allow tags in metadata: Don't include in launch template

User data - *optional*

Upload a file with your user data or enter it 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_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

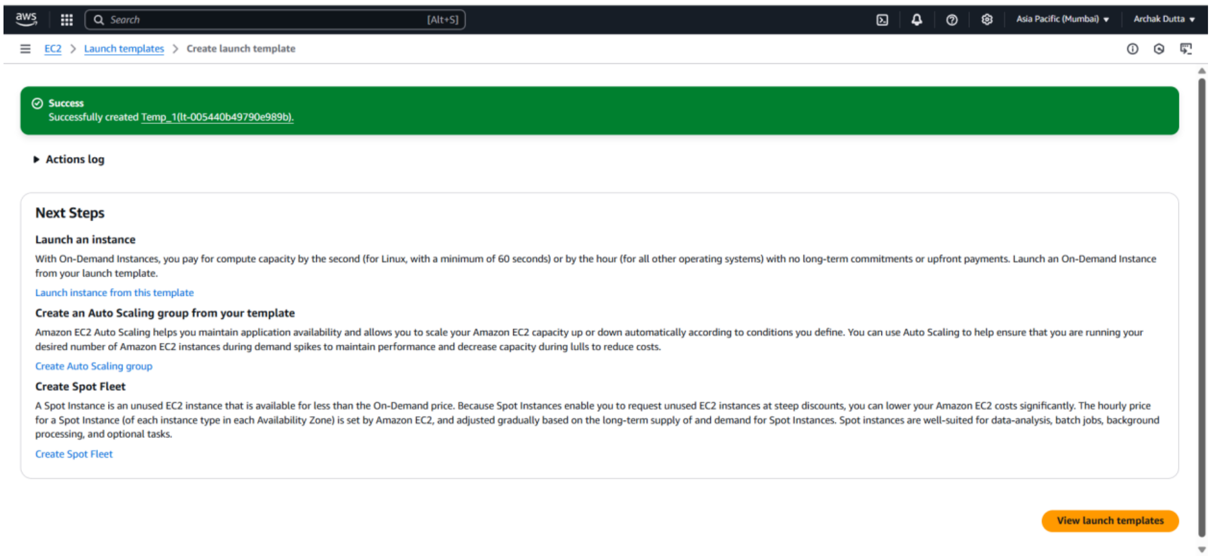
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0e35ddab05955cf57

Virtual server type (instance type): t2.micro

Firewall (security group): default

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.


EC2 > Launch templates > Create launch template

Success
Successfully created Temp_1(lt-005440b49790e989b).

Actions log

Next Steps

Launch an instance

With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

Launch instance from this template

Create an Auto Scaling group from your template

Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

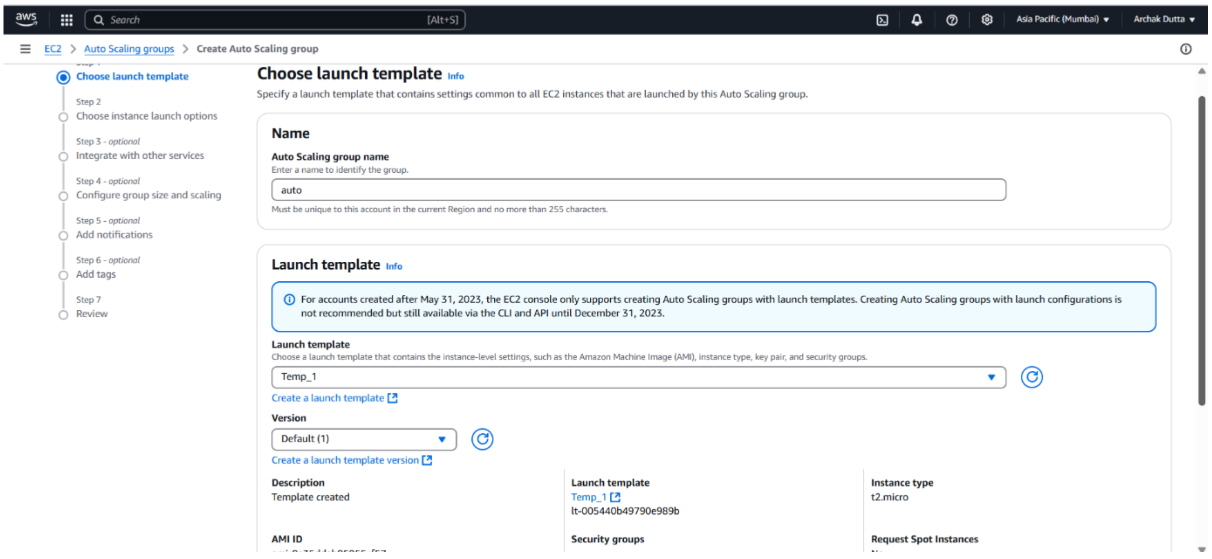
Create Auto Scaling group

Create Spot Fleet

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

Create Spot Fleet

4.Select auto scaling group.Provide a **name,select a templ**ate,**select the zones** under network section and choose **the balanced best effort** option.



EC2 > Auto Scaling groups > Create Auto Scaling group

- Choose launch template
- Step 2 Choose instance launch options
- 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

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.
auto
Must be unique to this account in the current Region and no more than 255 characters.

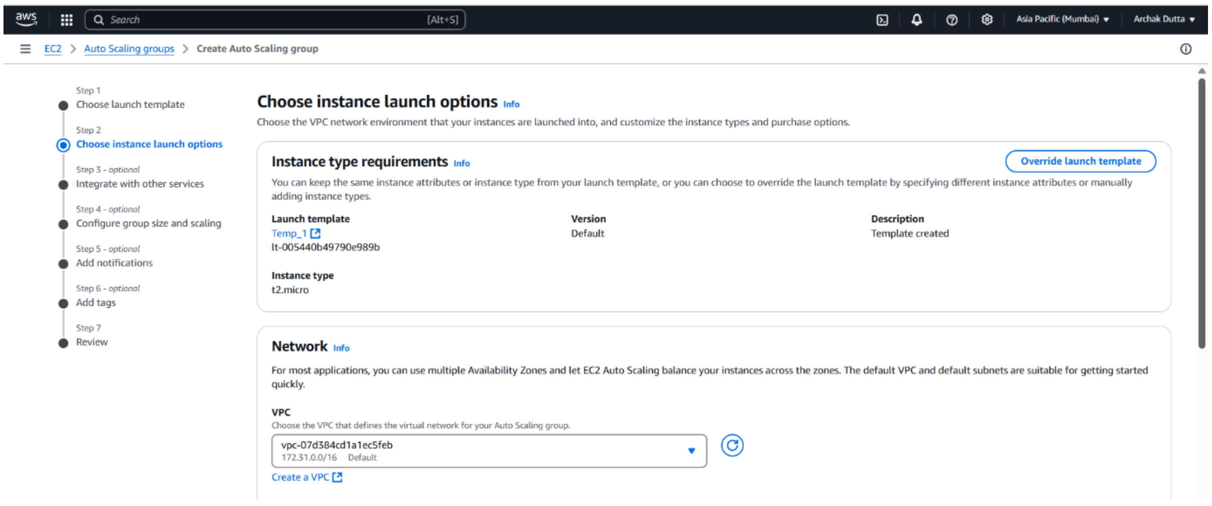
Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.
Temp_1
Create a launch template

Version
Default (1)
Create a launch template version

Description	**Launch template**	**Instance type**
Template created	Temp_1 lt-005440b49790e989b	t2.micro
AMI ID	**Security groups**	**Request Spot Instances**



EC2 > Auto Scaling groups > Create Auto Scaling group

- Step 1 Choose launch template
- Step 2 Choose instance launch options
- 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

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info — Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

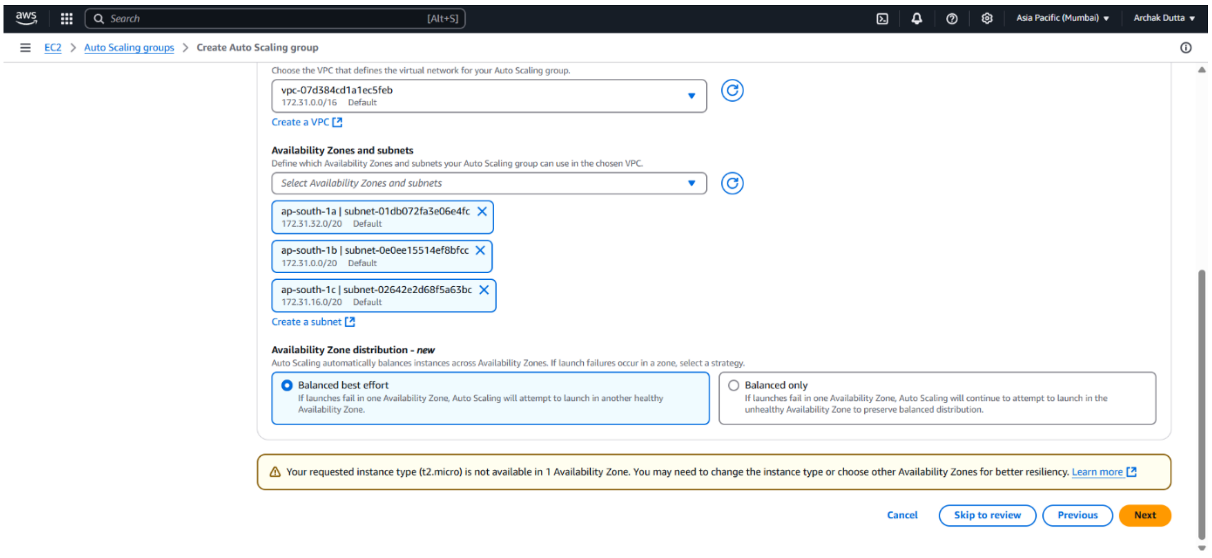
Launch template	**Version**	**Description**
Temp_1 lt-005440b49790e989b	Default	Template created

Instance type
t2.micro

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across 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-07d384cd1a1ec5feb
172.31.0.0/16 Default
Create a VPC



EC2 > Auto Scaling groups > Create Auto Scaling group

Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-07d384cd1a1ec5feb
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-01db072fa3e06e4fc — 172.31.32.0/20 Default
- ap-south-1b | subnet-0e0ee15514ef8bfcc — 172.31.0.0/20 Default
- ap-south-1c | subnet-02642e2d68f5a63bc — 172.31.16.0/20 Default

Create a subnet

Availability Zone distribution - *new*
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

[✓ Balanced best effort] If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

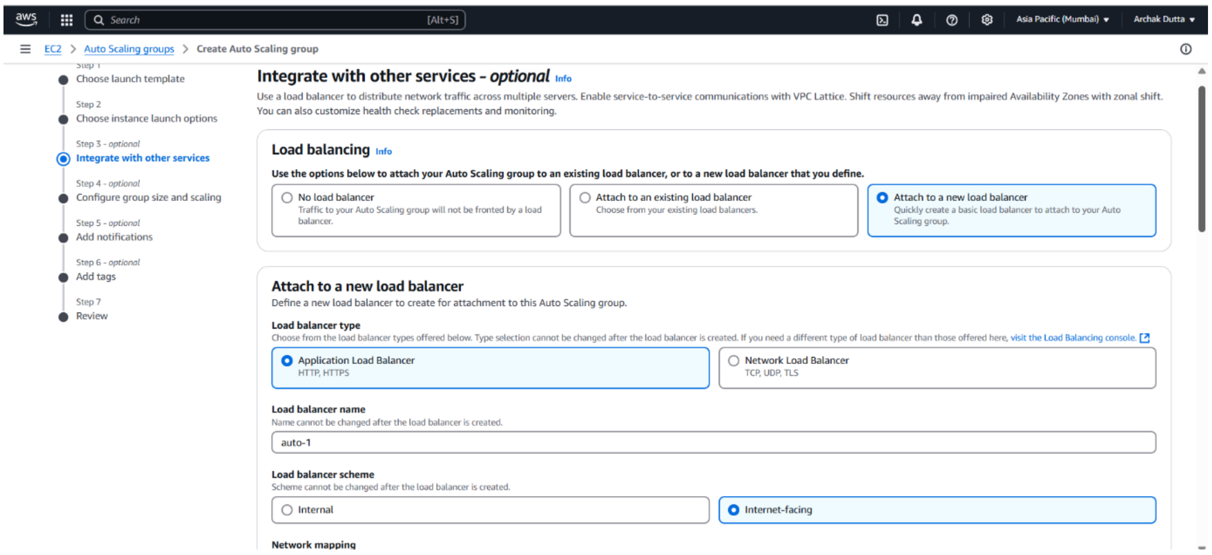
Balanced only — If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Your requested instance type (t2.micro) is not available in 1 Availability Zone. You may need to change the instance type or choose other Availability Zones for better resiliency. Learn more

Cancel | Skip to review | Previous | Next

5.Click on the next and select **attach to a new load balancer** along with **application load balancer** as **load balancer type** and internet-facing as **the load balancer scheme**.Provide port as **4000** and select create a target group under **listeners and routing**.Finally select **no**

VPC lattice service option and enable the additional health check types.Provide **health check grace period as 200 seconds.**



Step 1
Choose launch template

Step 2
Choose instance launch options

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

Integrate with other services - *optional* Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

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.

Attach to a new load balancer

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

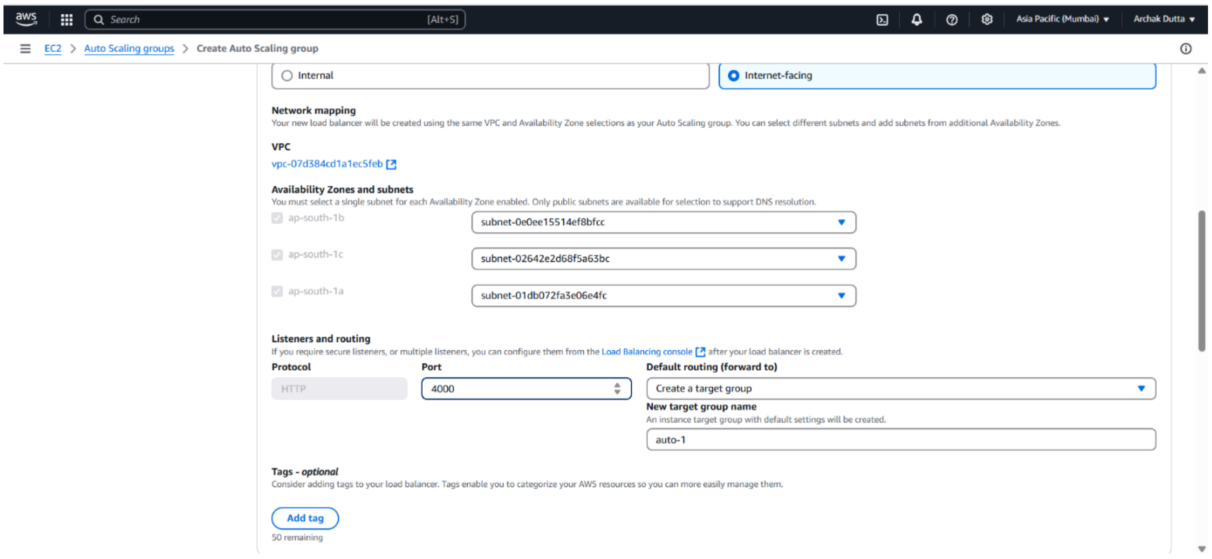
Load balancer type
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.

- [✓ Application Load Balancer] — HTTP, HTTPS
- Network Load Balancer — TCP, UDP, TLS

Load balancer name
Name cannot be changed after the load balancer is created.
auto-1

Load balancer scheme
Scheme cannot be changed after the load balancer is created.
Internal [✓ Internet-facing]

Network mapping



Internal [✓ Internet-facing]

Network mapping
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
vpc-07d384cd1a1ec5feb

Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

✓ ap-south-1b	subnet-0e0ee15514ef8bfcc
✓ ap-south-1c	subnet-02642e2d68f5a63bc
✓ ap-south-1a	subnet-01db072fa3e06e4fc

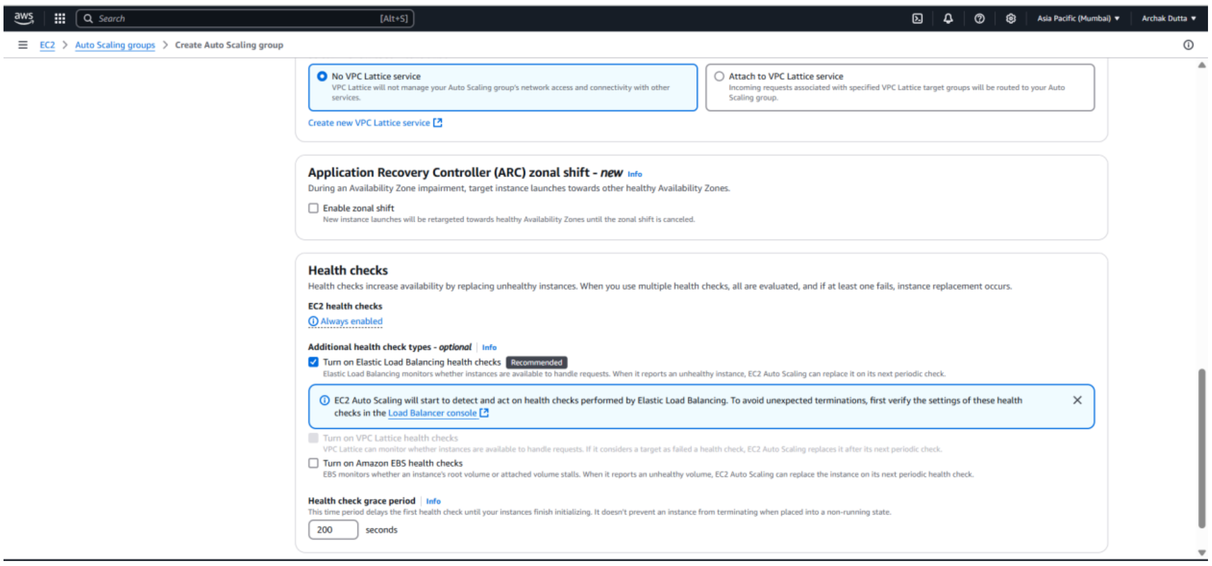
Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console after your load balancer is created.

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.
auto-1

Tags - *optional*
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag
50 remaining



- [✓ No VPC Lattice service] — VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.
- Attach to VPC Lattice service — Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service

Application Recovery Controller (ARC) zonal shift - *new* Info

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

- ☐ Enable zonal shift — New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.

Health checks

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

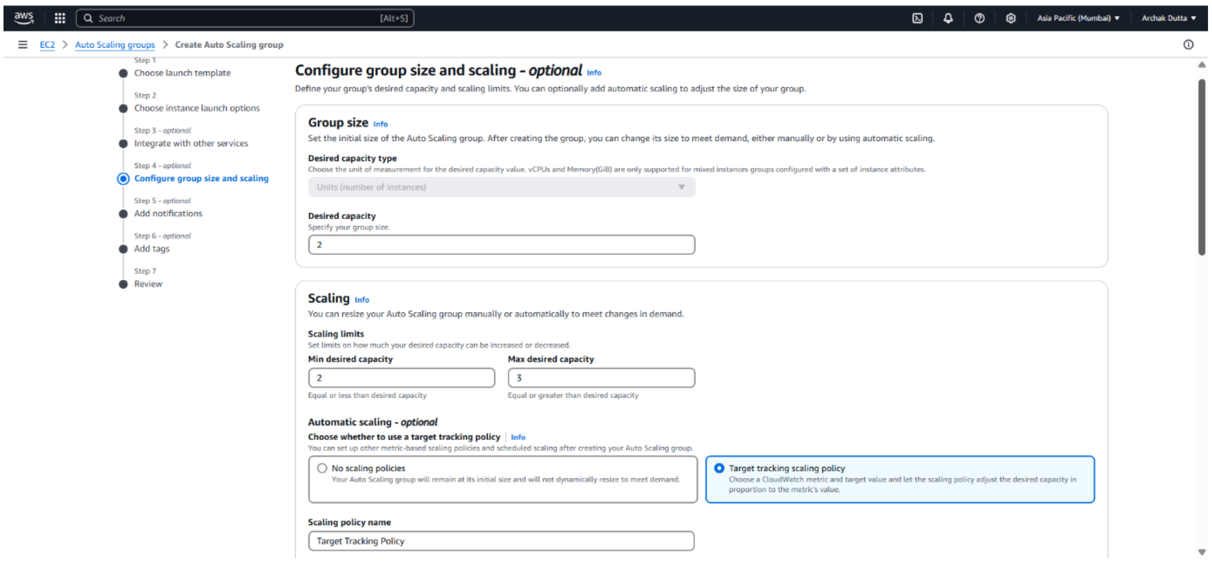
- ✓ Turn on Elastic Load Balancing health checks Recommended — Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console

- ☐ 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

6.In configure group and scaling window, select **desired capacity** as 2.Now under scaling select **min and max desired capacity** as 2 and 3 respectively.Select **target tracking scaling policy** under **automatic scaling**.Now provide **instance warmup as 200 seconds**.



Configure group size and scaling - *optional* Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info

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

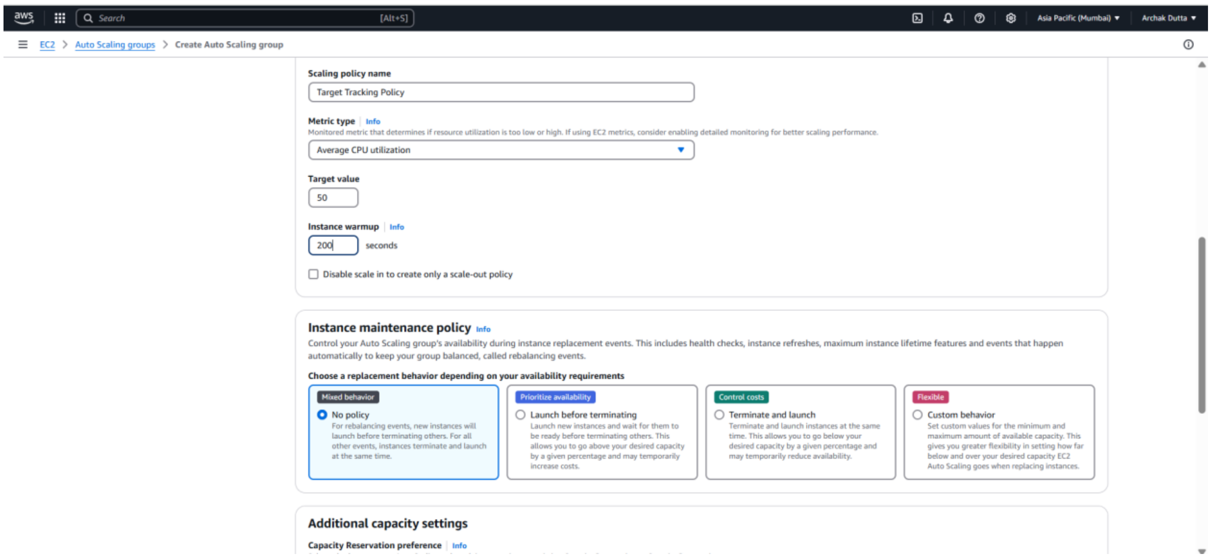
Max desired capacity
3
Equal or greater than desired capacity

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



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

Disable scale in to create only a scale-out policy

Instance maintenance policy Info

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements

Mixed behavior
[✓ No policy]
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

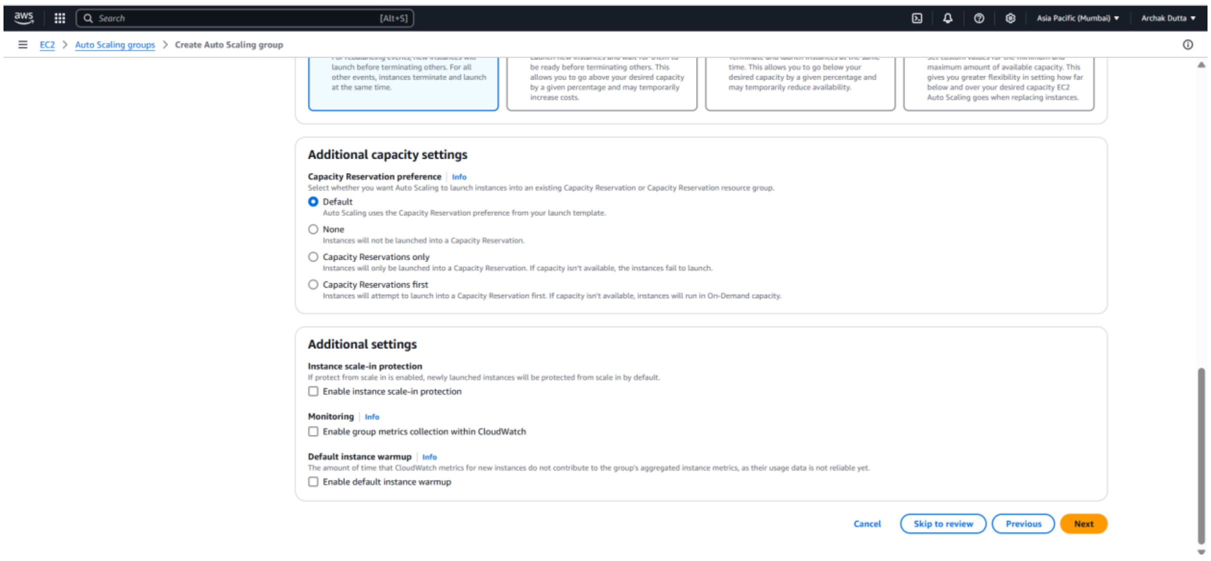
Prioritize availability
Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs
Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

Flexible
Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Additional capacity settings

Capacity Reservation preference | Info



launch before terminating others. For all other events, instances terminate and launch at the same time.

be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Additional capacity settings

Capacity Reservation preference | Info
Select whether you want Auto Scaling to launch instances into an existing Capacity Reservation or Capacity Reservation resource group.

[✓ Default]
Auto Scaling uses the Capacity Reservation preference from your launch template.

None
Instances will not be launched into a Capacity Reservation.

Capacity Reservations only
Instances will only be launched into a Capacity Reservation. If capacity isn't available, the instances fail to launch.

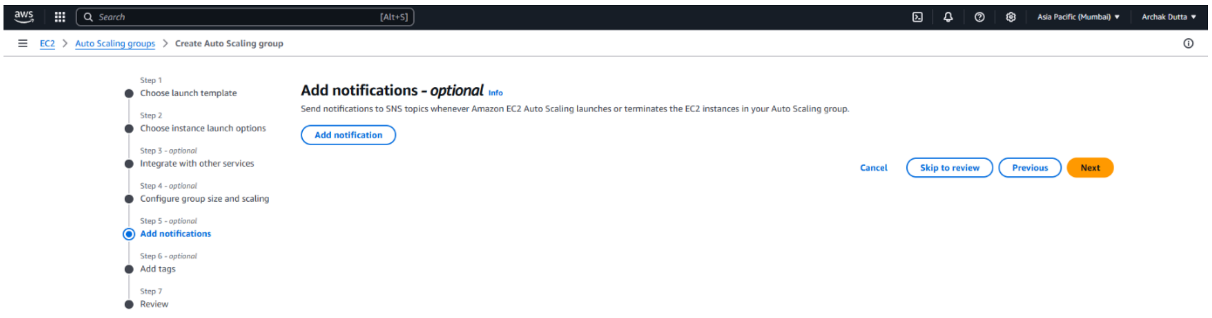
Capacity Reservations first
Instances will attempt to launch into a Capacity Reservation first. If capacity isn't available, instances will run in On-Demand capacity.

Additional settings

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

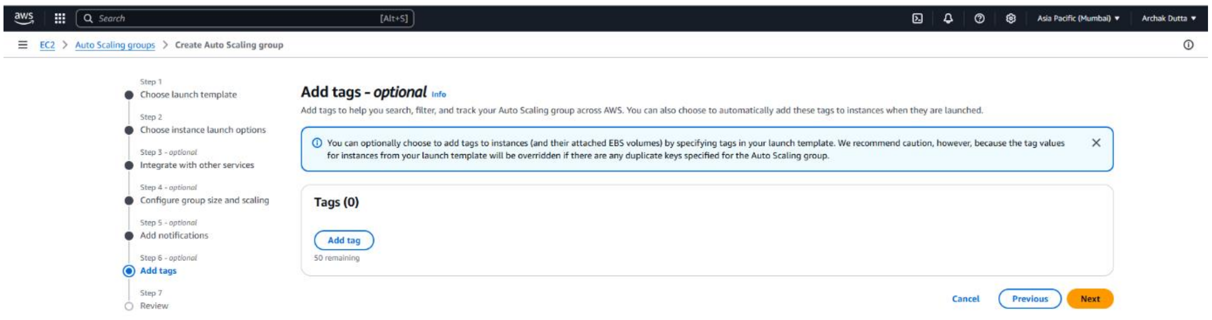
Monitoring | Info
Enable group metrics collection within CloudWatch

Default instance warmup | Info
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.
Enable default instance warmup


Search [Alt+S]
Asia Pacific (Mumbai)
Archak Dutta
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1 Choose launch template
Step 2 Choose instance launch options
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
Add notification
Cancel
Skip to review
Previous
Next

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.


Search [Alt+S]
Asia Pacific (Mumbai)
Archak Dutta
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1 Choose launch template
Step 2 Choose instance launch options
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
Add tag
Cancel
Previous
Next

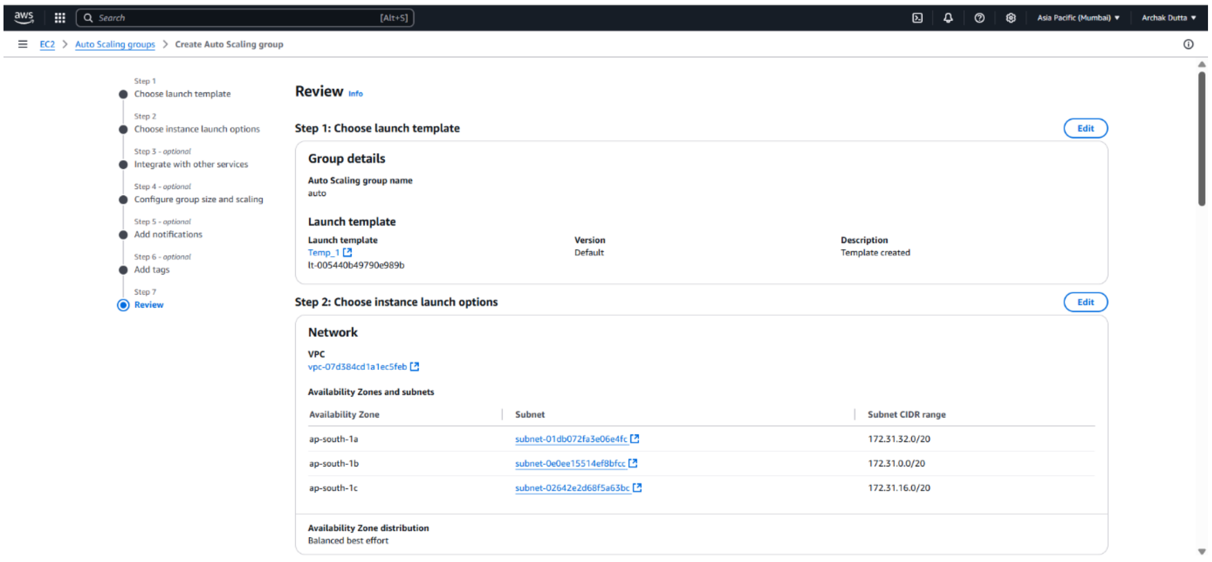
Add tags - *optional* Info

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

Tags (0)

50 remaining


Search [Alt+S]
Asia Pacific (Mumbai)
Archak Dutta
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1 Choose launch template
Step 2 Choose instance launch options
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
Edit
Edit

Review Info

Step 1: Choose launch template

Group details

Auto Scaling group name
auto

Launch template

Launch template	Version	Description
Temp_1 lt-005440b49790e989b	Default	Template created

Step 2: Choose instance launch options

Network

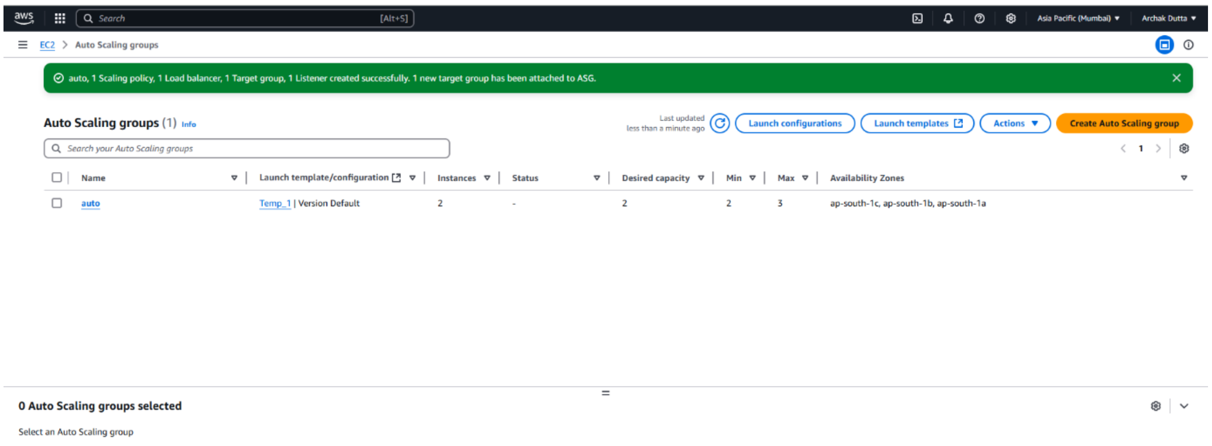
VPC
vpc-07d384cd1a1ec5feb

Availability Zones and subnets

Availability Zone	Subnet	Subnet CIDR range
ap-south-1a	subnet-01db072fa3e06e4fc	172.31.32.0/20
ap-south-1b	subnet-0e0ee15514ef8bfcc	172.31.0.0/20
ap-south-1c	subnet-02642e2d68f5a63bc	172.31.16.0/20

Availability Zone distribution
Balanced best effort

7.Click on next and finally the auto scaling group has been created.



auto, 1 Scaling policy, 1 Load balancer, 1 Target group, 1 Listener created successfully. 1 new target group has been attached to ASG.

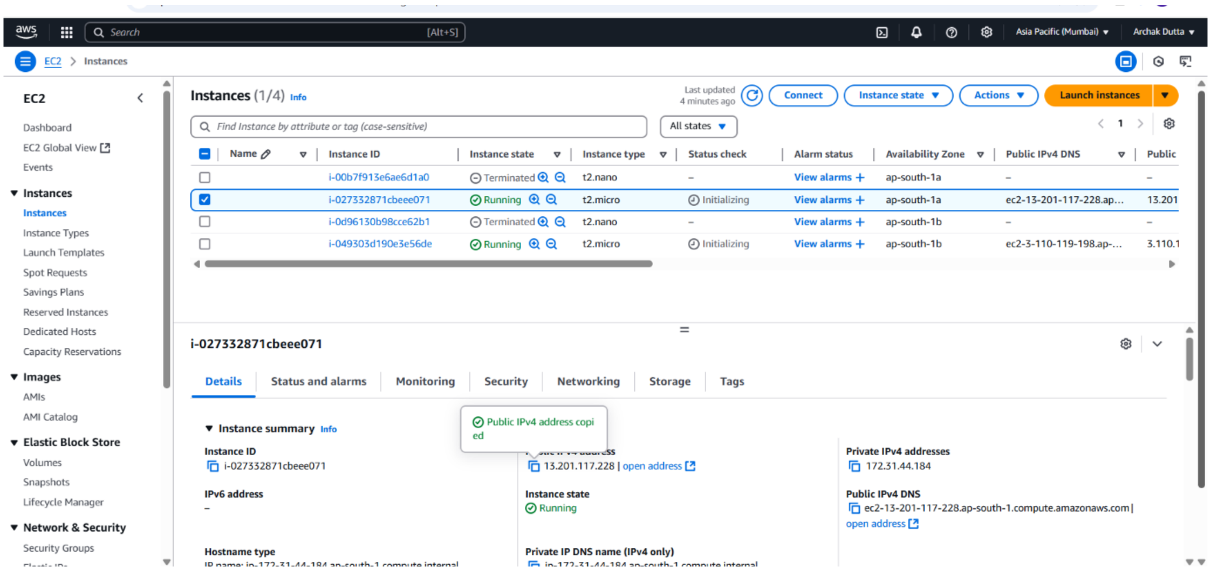
Auto Scaling groups (1) Info

Last updated less than a minute ago

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
auto	Temp_1 | Version Default	2	-	2	2	3	ap-south-1c, ap-south-1b, ap-south-1a

0 Auto Scaling groups selected

Select an Auto Scaling group


Instances (1/4)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
☐		i-00b7f913e6ae6d1a0	Terminated	t2.nano	–	View alarms +	ap-south-1a	–	–
☑		i-027332871cbeee071	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-201-117-228.ap...	13.201
☐		i-0d96130b98cce62b1	Terminated	t2.nano	–	View alarms +	ap-south-1b	–	–
☐		i-049303d190e3e56de	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-3-110-119-198.ap-...	3.110.1

i-027332871cbeee071

Instance summary

Public IPv4 address copied

Instance ID: i-027332871cbeee071

Public IPv4 address: 13.201.117.228 | open address

Private IPv4 addresses: 172.31.44.184

IPv6 address: –

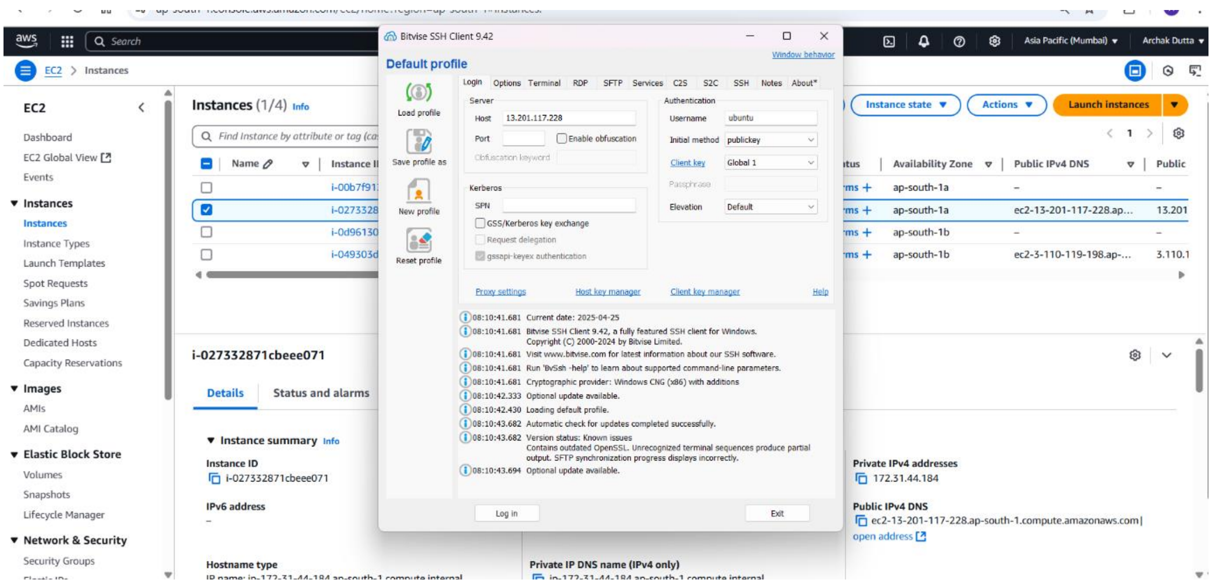
Instance state: Running

Public IPv4 DNS: ec2-13-201-117-228.ap-south-1.compute.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)

8.Open the bitvise ssh client and log in using the given key pair and execute the following commands in the terminal.


Bitvise SSH Client 9.42

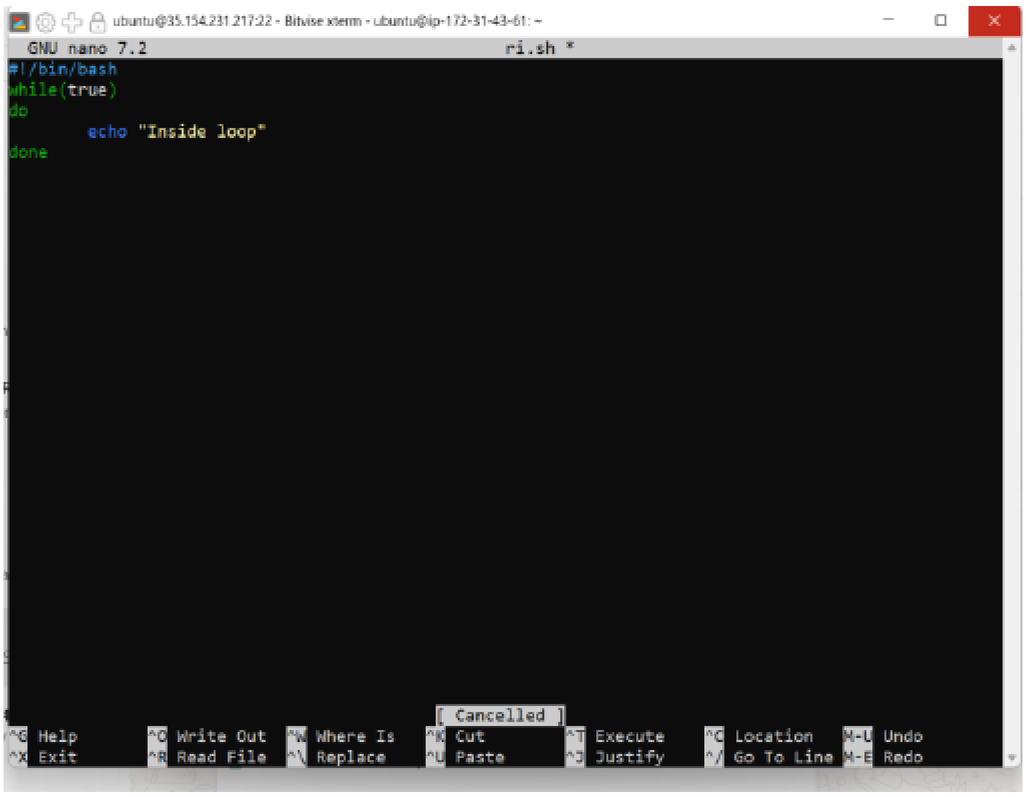
Default profile

Server
Host: 13.201.117.228
Port:
☐ Enable obfuscation

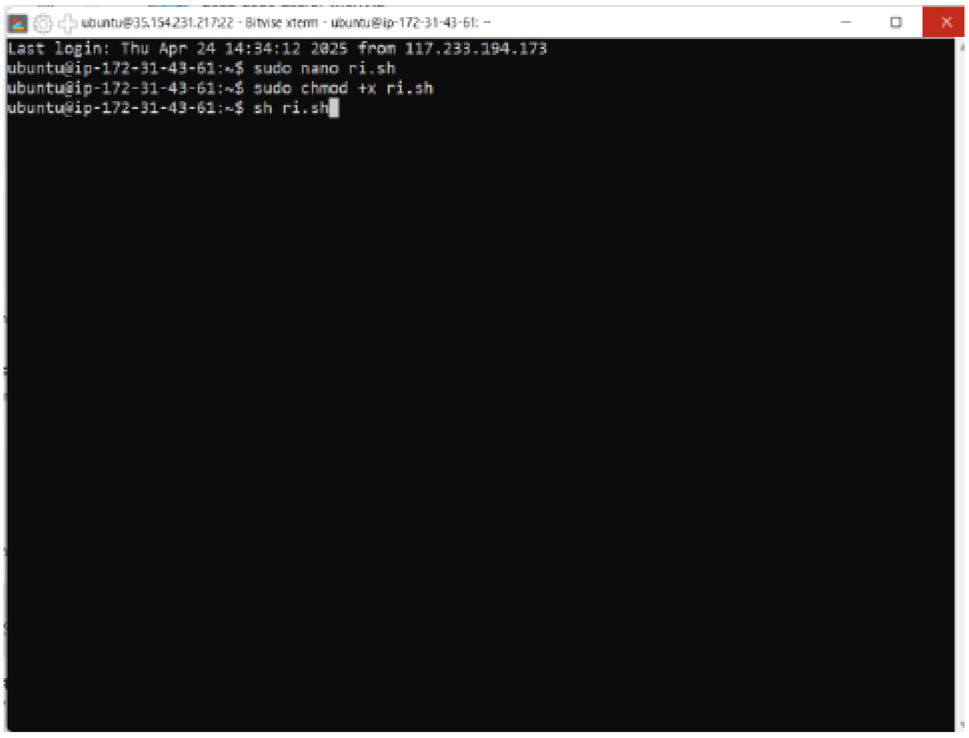
Authentication
Username: ubuntu
Initial method: publickey
Client key: Global 1
Elevation: Default

Kerberos
☐ GSS/Kerberos key exchange
☐ Request delegation
☑ gssapi-keyex authentication

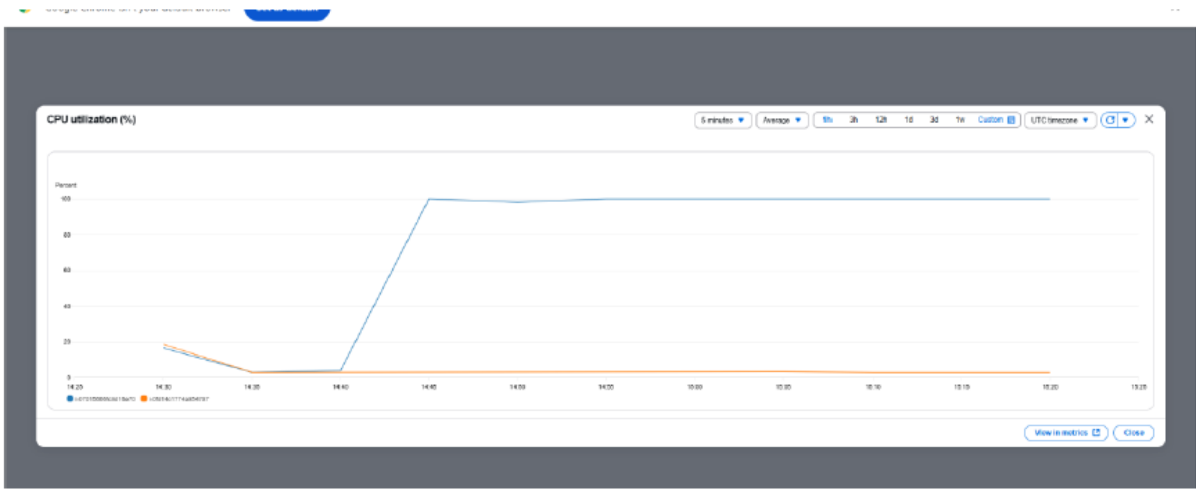
```
08:10:41.681 Current date: 2025-04-25
08:10:41.681 Bitvise SSH Client 9.42, a fully featured SSH client for Windows.
             Copyright (C) 2000-2024 by Bitvise Limited.
08:10:41.681 Visit www.bitvise.com for latest information about our SSH software.
08:10:41.681 Run 'BvSsh -help' to learn about supported command-line parameters.
08:10:41.681 Cryptographic provider: Windows CNG (x86) with additions
08:10:42.333 Optional update available.
08:10:42.430 Loading default profile.
08:10:43.682 Automatic check for updates completed successfully.
08:10:43.682 Version status: Known issues
             Contains outdated OpenSSL. Unrecognized terminal sequences produce partial
             output. SFTP synchronization progress displays incorrectly.
08:10:43.694 Optional update available.
```


ubuntu@35.154.231.217:22 - Bitvise xterm - ubuntu@ip-172-31-43-61: ~

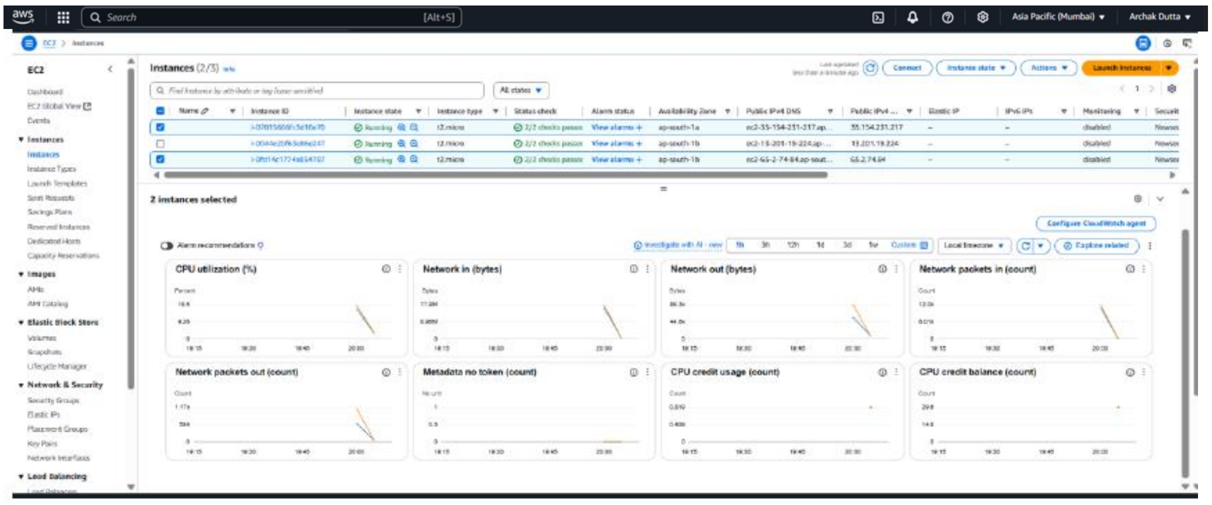
```
  GNU nano 7.2                          ri.sh *
#!/bin/bash
while(true)
do
          echo "Inside loop"
done




                                     [ Cancelled ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```



```
Last login: Thu Apr 24 14:34:12 2025 from 117.233.194.173
ubuntu@ip-172-31-43-61:~$ sudo nano ri.sh
ubuntu@ip-172-31-43-61:~$ sudo chmod +x ri.sh
ubuntu@ip-172-31-43-61:~$ sh ri.sh
```



9.Now check the CPU utilization on the monitoring section after executing the bash file and the third instance is created as a result of it.



Instances (2/3)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring
☑		[illegible]	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-35-154-231-217.ap...	35.154.231.217	–	–	disabled
☐		[illegible]	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-201-19-224.ap-...	13.201.19.224	–	–	disabled
☑		[illegible]	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-65-2-74-84.ap-sout...	65.2.74.84	–	–	disabled

2 instances selected

The commands used in this assignment are-

```
#!/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
```

```
//Sudo nano ri.sh
```

Within this file we will write the following code-

```
#!/bin/bash

while(true)

do

    echo "Inside loop"

done

//sudo chmod +x ri.sh

//sh ri.sh
```