

Assignment : 11

Build scaling plans in AWS that balance the load on different EC2 instances.

Step 1: Go to the EC2 page > Security Groups (under the heading Network & Security).

Amazon Elastic Compute Cloud (EC2)
Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

Launch a virtual server
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
[Launch instance](#)
[View dashboard](#)

Benefits and features
EC2 offers ultimate scalability and control
Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

Get started
Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.
[Get started walkthroughs](#)
[Get started tutorial](#)

Security Groups (1)
Find resources by attribute or tag

Name	Security group ID	Security group name	VPC ID	Description
-	sg-034eb43712646092d	default	vpc-0e2181ce3777a313c	default VPC sec

Step 2: Click on “Create security group” to create a new security group. Name the group and add description.

Create security group
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details
Security group name
Security51
Name cannot be edited after creation.
Description
new security group
VPC
vpc-0e2181ce3777a313c

Inbound rules
This security group has no inbound rules.
[Add rule](#)

Step 3: Add the below 4 inbound rules (SSH, HTTP, HTTPS, Custom TCP) and select Anywhere IPv4 in the Source. Add port range 4000 in the Custom TCP type. Afterwards click on Create Security Group.

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

EC2 > Security Groups > Create security group

Inbound rules

Type	Protocol	Port range	Source	Description - optional	
SSH	TCP	22	Anyw...		Delete
			0.0.0.0		
HTTP	TCP	80	Anyw...		Delete
			0.0.0.0		
HTTPS	TCP	443	Anyw...		Delete
			0.0.0.0		
Custom TCP	TCP	4000	Anyw...		Delete
			0.0.0.0		

Add rule

Rules with source of 0.0.0.0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

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

EC2 > Security Groups > sg-0026f27c5ffcd4d49 - Security51

EC2

- Dashboard
- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
- Images
 - AMIs

Security group (sg-0026f27c5ffcd4d49 | Security51) was created successfully

Details

sg-0026f27c5ffcd4d49 - Security51

Actions

Details			
<div>Security group name</div> <div>Security51</div>	<div>Security group ID</div> <div>sg-0026f27c5ffcd4d49</div>	<div>Description</div> <div>new security group</div>	<div>VPC ID</div> <div>vpc-0e2181ce3777a313c</div>
<div>Owner</div> <div>586794457897</div>	<div>Inbound rules count</div> <div>4 Permission entries</div>	<div>Outbound rules count</div> <div>1 Permission entry</div>	

Inbound rules Outbound rules Sharing - new VPC associations - new Tags

Inbound rules (4)

Manage tags Edit inbound rules

Step 4: From the lefthand side menu bar, go to Launch Templates and click on "Launch Template".

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

EC2 > Security Groups > sg-0026f27c5ffcd4d49 - Security51

EC2

- Dashboard
- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests

Launch Templates

Search

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
You do not have any Launch Templates in this region					

Create launch template

Step 5: Name the Template and set the version as "v0", also check the Auto scaling guidance box.

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

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

Template51

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

Template version description

v0

Max 255 chars

Auto Scaling guidance

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

Summary

Software image (AMI)

Virtual server type (instance type)

Firewall (security group)

Storage (volumes)

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.

Step 6: Scroll down to Application and OS > Quick Start > Ubuntu.

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 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
ami-0e35ddab05955cf57 (64-bit (x86)) / ami-0429d68a1cd41ca80 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

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

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Create launch template

Step 7: In the instance type, select t2.micro as it is the Free tier eligible instance.

Instance type Info | Get advice Advanced

Instance type

t2.micro
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

Free tier eligible

All generations Compare instance types

Additional costs apply for AMIs with pre-installed software

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

Don't include in launch template Create new key pair

Network settings Info

Subnet Info

Don't include in launch template Create new subnet

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Create launch template

Step 8: Create a new key pair for login.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
key-51
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Create launch template

Step 9: Network Settings > Firewall (security groups) > Select Existing groups. Now select the above made security group.

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

EC2 > Launch templates > Create launch template

Network settings

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

☐ Specify a custom value...

<input type="checkbox"/> default	sg-034eb43712646092d
VPC: vpc-0e2181ce3777a313c	
<input type="checkbox"/> Security51	sg-0026f27c5ffcd4d49
VPC: vpc-0e2181ce3777a313c	

[Compare security group rules](#)

[Hide details](#)

Volume 1 (AMI Root) : 8 GiB, EBS, General purpose SSD (gp3)

AMI Volumes are not included in the template unless modified

Summary

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[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.

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

Step 10: In the Advanced details, write the below mentioned code,

```
#!/bin/bash
apt-get update
apt-get upgrade
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          .git
cd          aws
npm install
node index.js
```

And the click on “Create Launch Template”.

aws Search [Alt+S] Asia Pacific (Mumbai) 51_Prasun

EC2 > Launch templates > Create launch template

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.

[Choose file](#)

```
#!/bin/bash
apt-get update
apt-get upgrade
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          aws.git
cd          aws
npm install
node index.js
```

☐ User data has already been base64 encoded

Summary

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

Virtual server type (instance type)
t2.micro

Firewall (security group)
Security51

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](#)

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

EC2 > Launch templates > Create launch template

Success
Successfully created Template51(lt-0edc13f38e715efa6).

Actions log

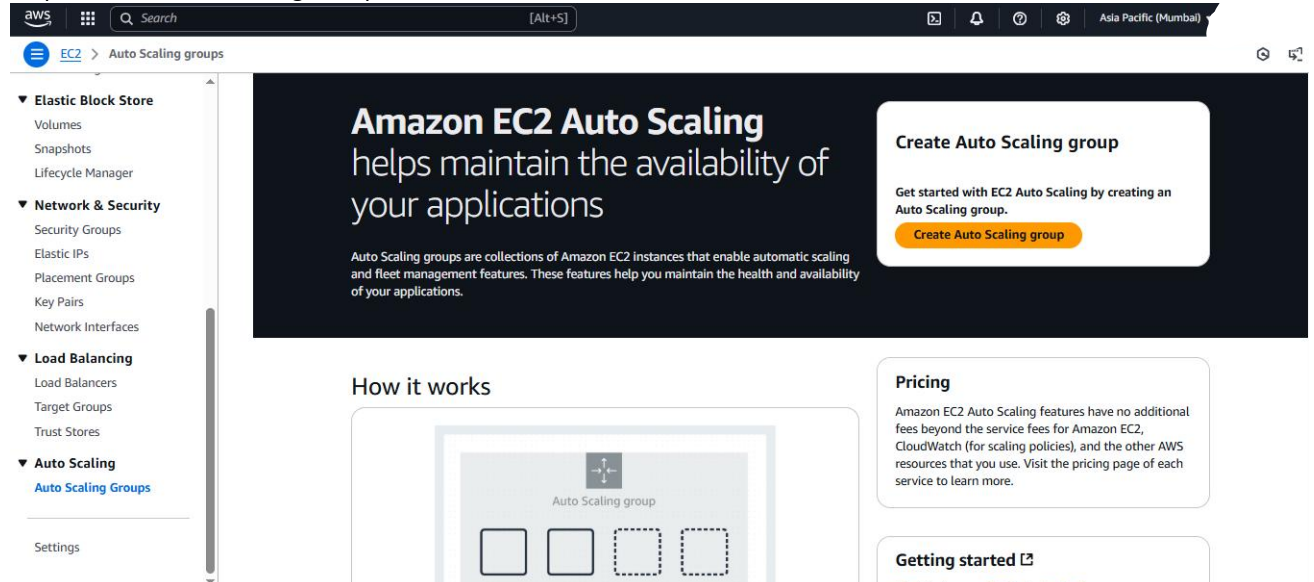
Next Steps

Launch an instance

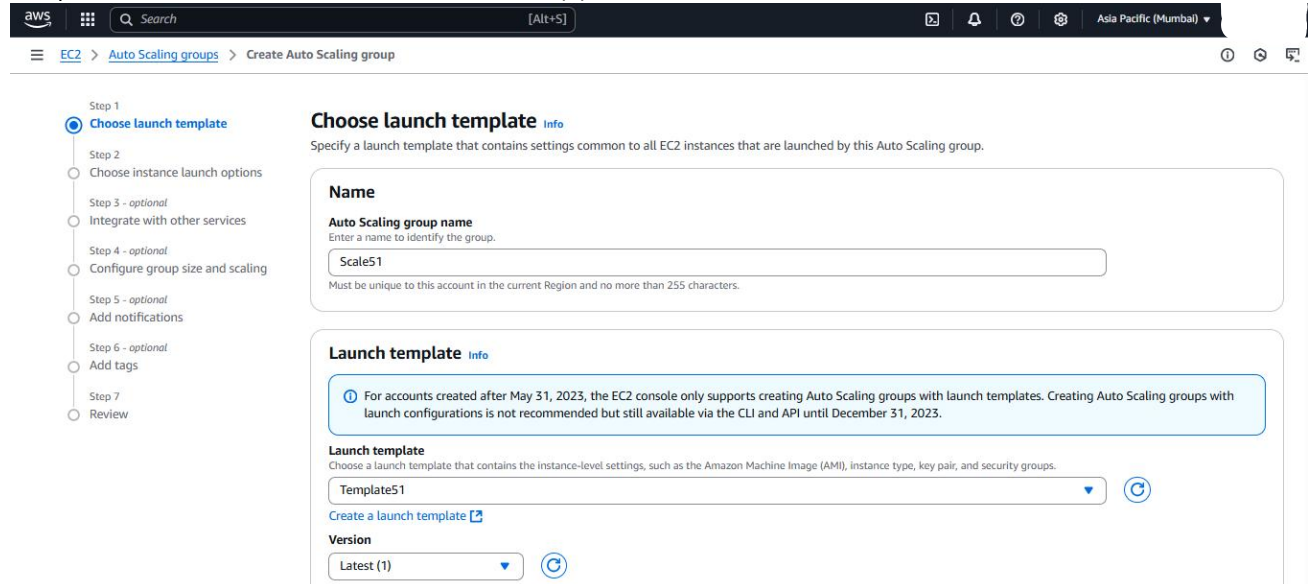
With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

[Launch instance from this template](#)

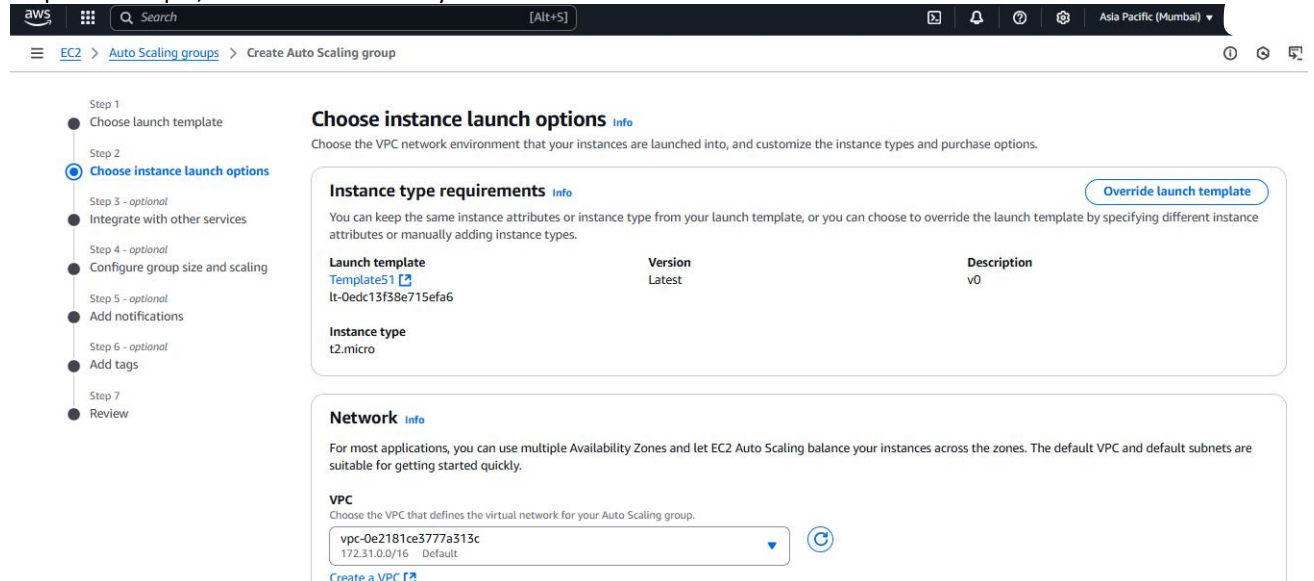
Step 11: Go to Auto Scaling Groups in the lefthand side menu.



Step 12: Click on “Create Auto Scaling Group”, name the group, select the above made template in the Launch Template section, and select the version as Latest(1).



Step 13: In step 2, select all 3 Availability Zones and Subnets and click on next.



aws

Search

[Alt+S]

Asia Pacific (Mumbai)

EC2

>

Auto Scaling groups

>

Create Auto Scaling group

Step 7

Review

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-0e2181ce3777a313c

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-02fbd93920842d759

172.31.32.0/20

Default

ap-south-1b | subnet-0e06b1c09b402835f

172.31.0.0/20

Default

ap-south-1c | subnet-0542c368c04490950

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

Balanced only

Step 14: In step 3, select the below shown options and Create a new target group in Default Routing and click on Next.

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

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

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.

Scale51-1

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

EC2

>

Auto Scaling groups

>

Create Auto Scaling group

Load balancer name

Name cannot be changed after the load balancer is created.

Scale51-1

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

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-0e2181ce3777a313c

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-0e06b1c09b402835f

ap-south-1c

subnet-0542c368c04490950

ap-south-1a

subnet-02fbd93920842d759

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)

VPC
vpc-0e2181ce3777a313c

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-0e06b1c09b402835f

☒ ap-south-1c subnet-0542c368c04490950

☒ ap-south-1a subnet-02fbd93920842d759

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 TCP **Port** 80 **Default routing (forward to)** Create a target group

New target group name
An instance target group with default settings will be created.
Scale51-1

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

Step 15: In step 4, Select Desired capacity as 2, Min desired capacity as 2 and Max desired capacity as 3.

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

Step 16: In Automatic Scaling, Choose Target tracking scaling policy, set the policy name, Target value as 32 and Instance Warmup as 300 seconds and click on Next.

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
32

Instance warmup [Info](#)
300 seconds

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

Instance maintenance policy [Info](#)

Step 17: In step 5 & 6, click on Next. In step 7, after reviewing the whole Auto Scaling Group, click on Create Auto Scaling Group.

[illegible]

Step 19: Click on the first instance and copy its IPv4 address and run it in a separate incognito window to check if it's functional or not.

The screenshot shows the AWS Management Console interface for an EC2 instance named `i-0da4332b356f62c28`. The instance is in the `Running` state. Key details include:

- Instance ID:** `i-0da4332b356f62c28`
- Public IPv4 address:** `13.233.216.39` (with a link to `open address`)
- Instance state:** `Running`
- Private IPv4 addresses:** `172.31.38.56`
- Public IPv4 DNS:** `ec2-13-233-216-39.ap-south-1.compute.amazonaws.com` (with a link to `open address`)
- Private IP DNS name (IPv4 only):** `ip-172-31-38-56.ap-south-1.compute.internal`
- Instance type:** `t2.micro`
- VPC ID:** `vpc-0e2181ce3777a313c`
- Subnet ID:** `subnet-02fbd93920842d759`
- Instance ARN:** `arn:aws:ec2:ap-south-1:586794457897:instance/i-0da4332b356f62c28`
- Managed:** `false`

Below the console, a web browser window (Incognito) shows the `Welcome to nginx!` page, confirming that the web server is successfully installed and working on the instance.

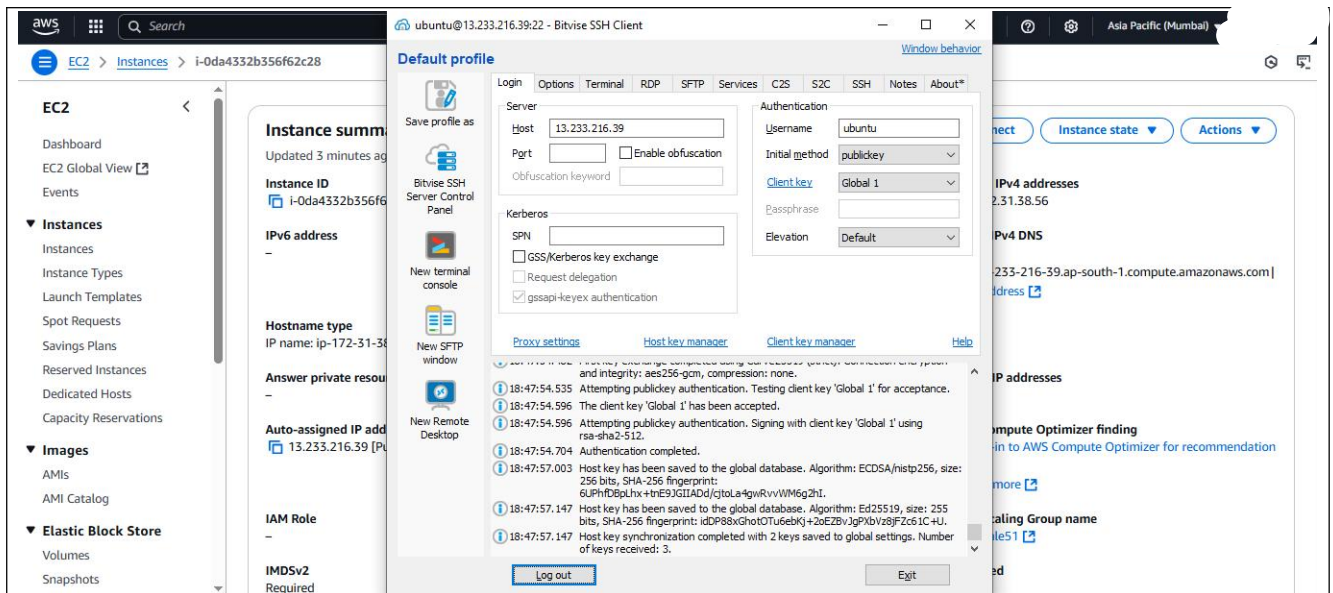
Step 20: Copy the same IPv4 address in Bitwise SSH Client, import the newly made key pair and Log in.

The screenshot shows the AWS Management Console for the same EC2 instance, with the `Instance summary` page visible. Overlaid on this is the Bitwise SSH Client 9.42 window, configured for a new profile. The configuration includes:

- Server:** `Host: 13.233.216.39`
- Authentication:** `Username: ubuntu`, `Initial method: publickey`, `Client key: Global 1`
- Kerberos:** `SPN:` (empty), `Request delegation:` (unchecked), `gssapi-keyex authentication:` (checked)

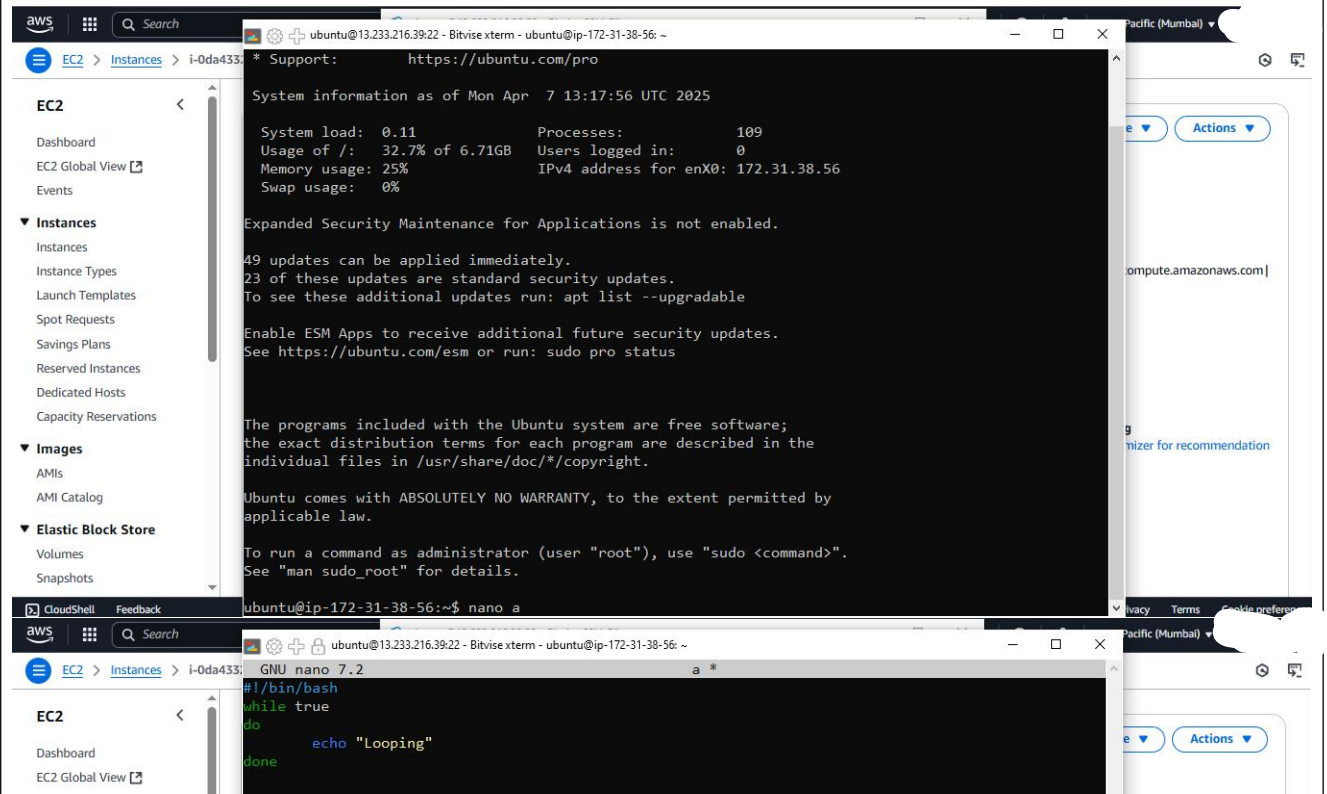
The Bitwise SSH Client window also displays a list of system messages, including updates and version information.

Step 21: Open a new Terminal.

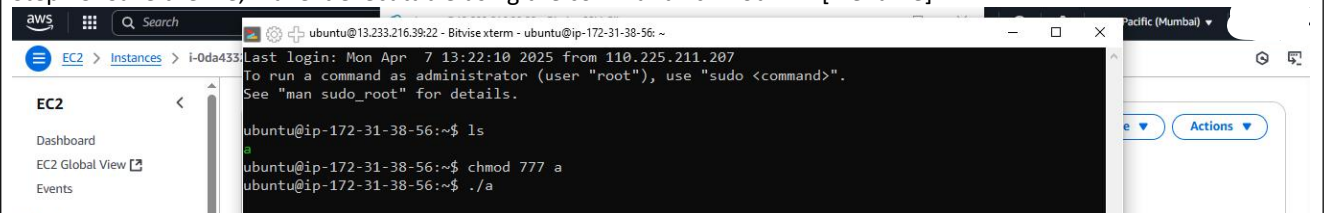


Step 22: Create a new file using “nano” command and write the following block of code,

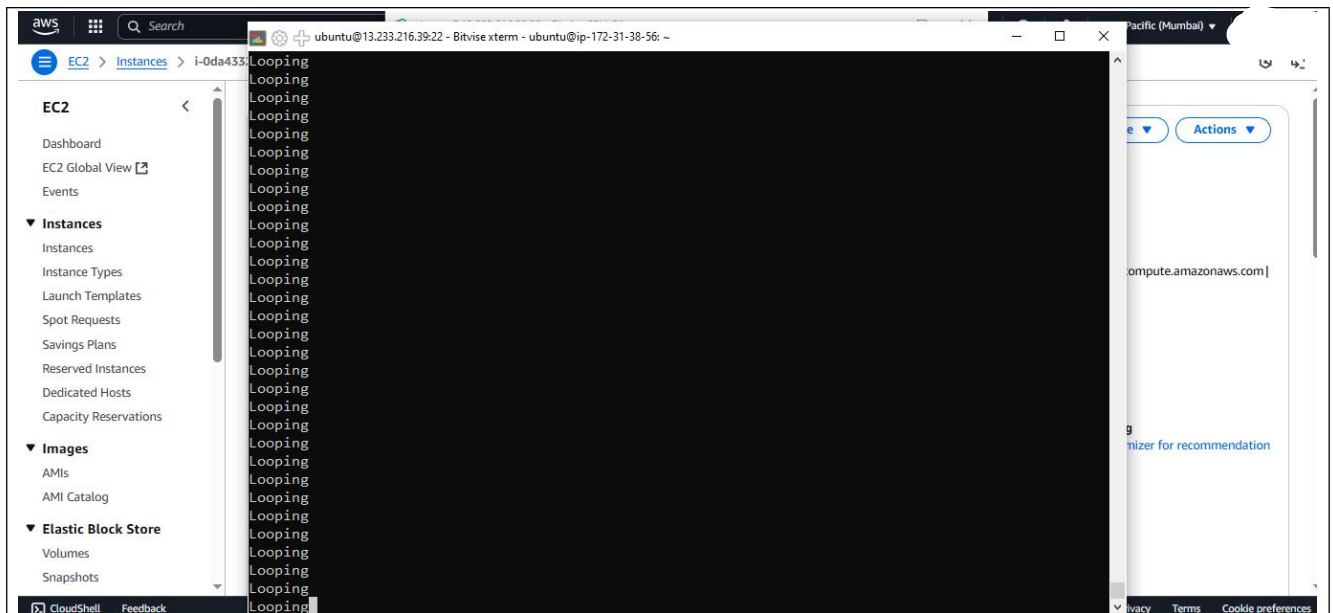
```
#!/bin/bash
while true
do
    echo "Looping"
done
```



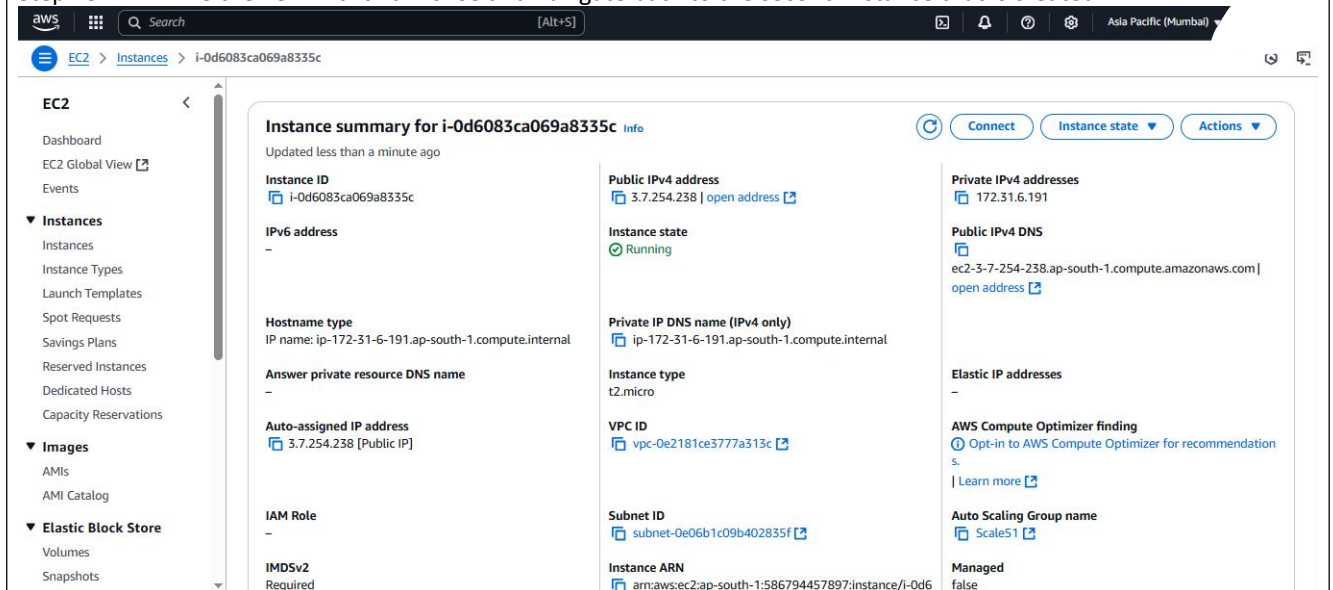
Step 23: Save the file, make it executable using the command “chmod 777 [filename]”.



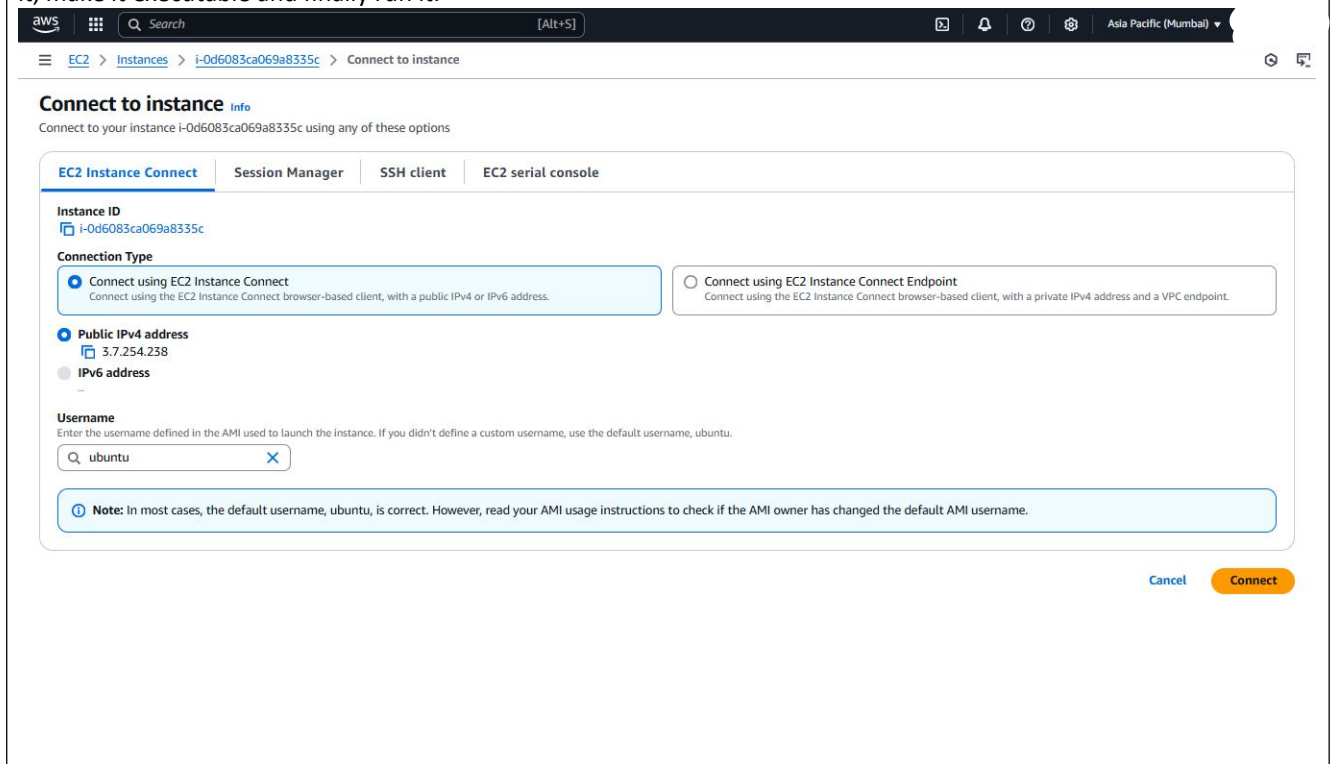
Step 24: Run the file using the command “./[filename]”



Step 25: Minimize the Terminal and Bitvise and navigate back to the second instance that is created.



Step 26: Click on Connect and follow the same steps as we've done in the Bitvise to create a new file, write the code in it, make it executable and finally run it.



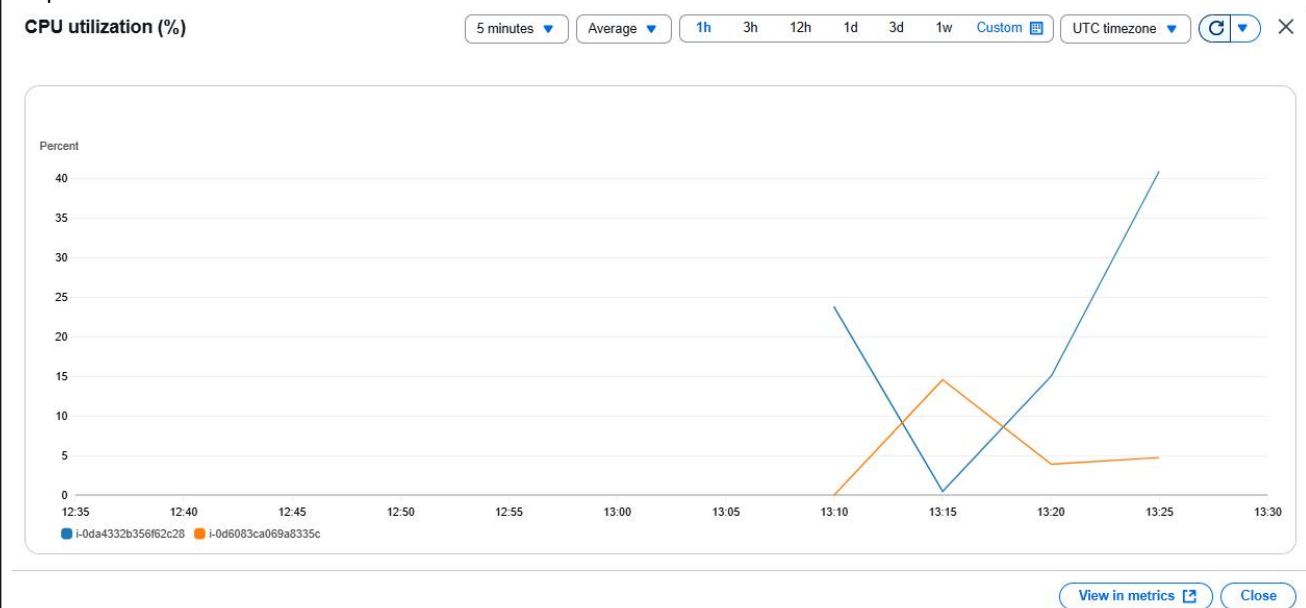
[illegible]

Step 27: Go to the previous tab and select both the instances to check the CPU Utilization.

The screenshot shows the AWS Management Console for EC2 Instances. Two instances are selected: i-0da4332b356f62c28 and i-0d6083ca069a8335c. The CPU utilization graph is visible, showing a sharp increase for instance i-0d6083ca069a8335c.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
i-0da4332b356f62c28	i-0da4332b356f62c28	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-2
i-0d6083ca069a8335c	i-0d6083ca069a8335c	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-7-

Step 29: Maximize the CPU Utilization window to have a better view.



So, finally we've Build scaling plans in AWS that balance the load on different EC2 instances.