

LABORATORY REPORT

Semester: Fall Semester 2025-26 (Fast Track)

Cloud Application Development
MCA2007

Slot : A11+A12+A14+F14

Submitted by

NAME: Aditya Pratap Singh

REGNO: 24MCA10024

Master of Computer Application

in



Submitted to

Dr. Munsifa Firdaus Khan

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING VIT BHOPAL
UNIVERSITY**

APRIL, 2025

TABLE OF CONTENTS

SL NO.	NAME OF EXPERIMENTS	PAGE NO.
1.	How to create an EC2 instance in Unix.	3-8
2.	How to launch an EC2 instance in Window.	9-18
3.	How to launch an EC2 instance with template.	19-21
4.	Creation of Bucket in S3.	22-29
5.	How to setup an EC2 instance in VPC.	30-34
6.	How to implement Load Balancer, Target Group C Auto-scaling group with EC2 instance.	35-53
7.	How to create EBS and attach to EC2 instance, modify size and create a snapshot.	54-60

Date: 23/05/2025	Title
Exp. No: 01	Creating an EC2 Instance on a Unix System.

Creating an EC2 Instance on a Unix System

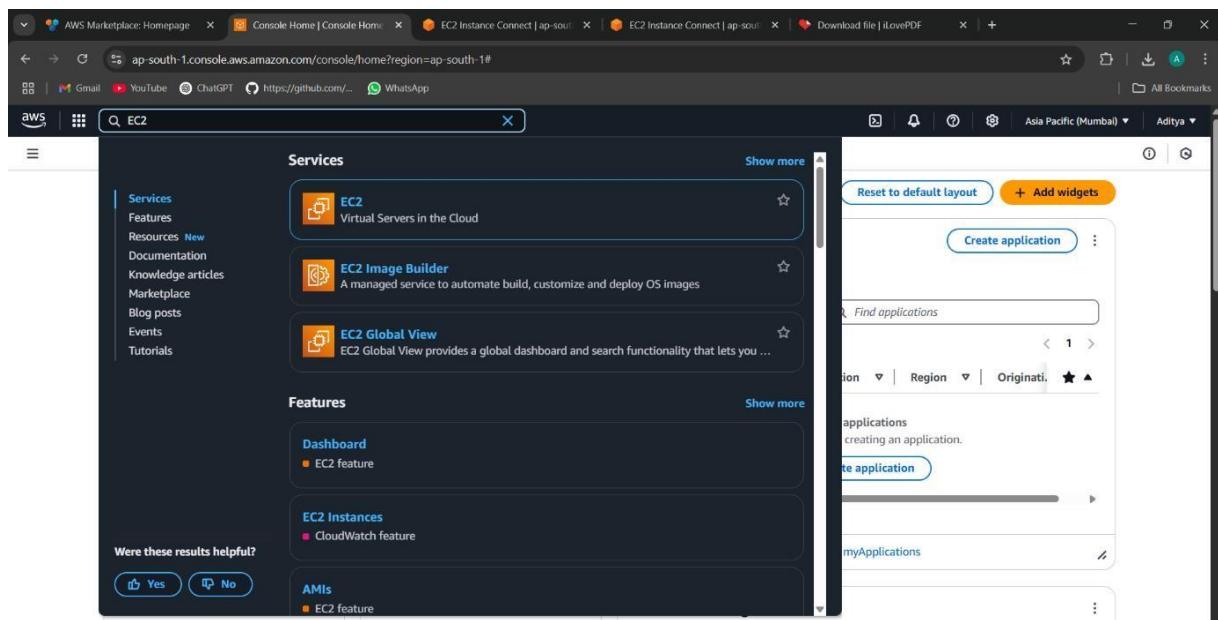
Creating an EC2 instance from a Unix-based system involves using the AWS Command Line Interface (CLI) to interact with AWS services programmatically. This method is ideal for developers and system administrators who prefer automation or terminal-based workflows.

The process begins by configuring the AWS CLI with user credentials and a default region. Once configured, you can launch an EC2 instance by specifying key parameters such as the Amazon Machine Image (AMI) ID, instance type, key pair name, security group, and subnet or VPC settings. These parameters define the operating system, hardware capacity, and network rules for the instance.

After running the appropriate command, AWS provisions a virtual machine in the cloud. The instance can then be accessed via SSH using the key pair specified during launch. This approach provides full control over the provisioning process and is particularly useful for scripting, automation, and infrastructure-as-code scenarios.

Using the CLI on Unix systems streamlines EC2 instance creation, making it efficient to deploy and manage instances in a consistent, repeatable manner.

STEP1: OPEN THE AWS ROOT ACCOUNT AND SEARCH FOR EC2



STEP2: CLICK ON EC2, EC2 DASHBOARD WILL APPEAR.

The screenshot shows the AWS EC2 Dashboard for the Asia Pacific (Mumbai) Region. The left sidebar navigation includes EC2, Dashboard, EC2 Global View, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). A prominent blue banner at the top states, "You can change your default landing page for EC2." Below this, the "Resources" section displays a table of current resources:

Instances (running)	Auto Scaling Groups	Capacity Reservations
0	0	0

Dedicated Hosts	Elastic IPs	Instances
0	0	0

Key Pairs	Load balancers	Placement groups
2	0	0

Security groups	Snapshots	Volumes
3	0	2

The "Account attributes" section on the right lists the Default VPC (vpc-0f4d0926b65ff0381), Settings (Data protection and security, Allowed AMIs, Zones, EC2 Serial Console, Default credit specification, EC2 console preferences), and Explore AWS (Get Up to 40% Better Price Performance, T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2).

STEP3: SELECT THE REGION>ASIA PACIFIC>MUMBAI

The screenshot shows the AWS EC2 console with the region dropdown menu open. The 'Asia Pacific (Mumbai)' region is selected. The menu lists various regions with their respective abbreviations:

- United States: N. Virginia (us-east-1), Ohio (us-east-2), N. California (us-west-1), Oregon (us-west-2)
- Asia Pacific: Mumbai (ap-south-1), Osaka (ap-northeast-3), Seoul (ap-northeast-2), Singapore (ap-southeast-1), Sydney (ap-southeast-2), Tokyo (ap-northeast-1)
- Canada: Central (ca-central-1)
- Europe: Frankfurt (eu-central-1), Ireland (eu-west-1), London (eu-west-2), Paris (eu-west-3)

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	2	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	2
Key pairs	2	Load balancers	0	Placement groups	0
Security groups	3	Snapshots	0	Volumes	2

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

AWS Health Dashboard

Region
Asia Pacific (Mumbai)

Status
This service is operating normally.

[Manage Regions](#) [Manage Local Zones](#)

STEP4: SELECT LAUNCH INSTANCE

The screenshot shows the AWS EC2 Compute landing page. The main heading is "Amazon Elastic Compute Cloud (EC2)". Below it, a large call-to-action button says "Launch a virtual server". To the right, there's a "Get started" section with a "Take our walkthroughs to help" link.

Compute

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.

Benefits and features

Launch a virtual server

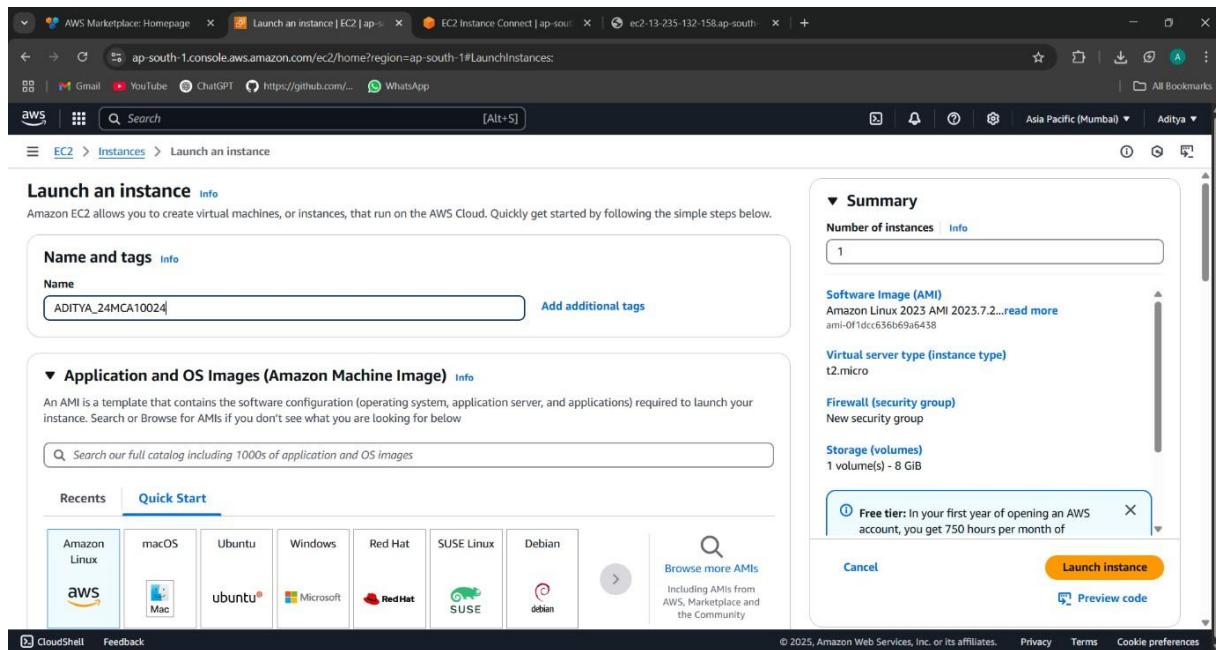
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance **View dashboard**

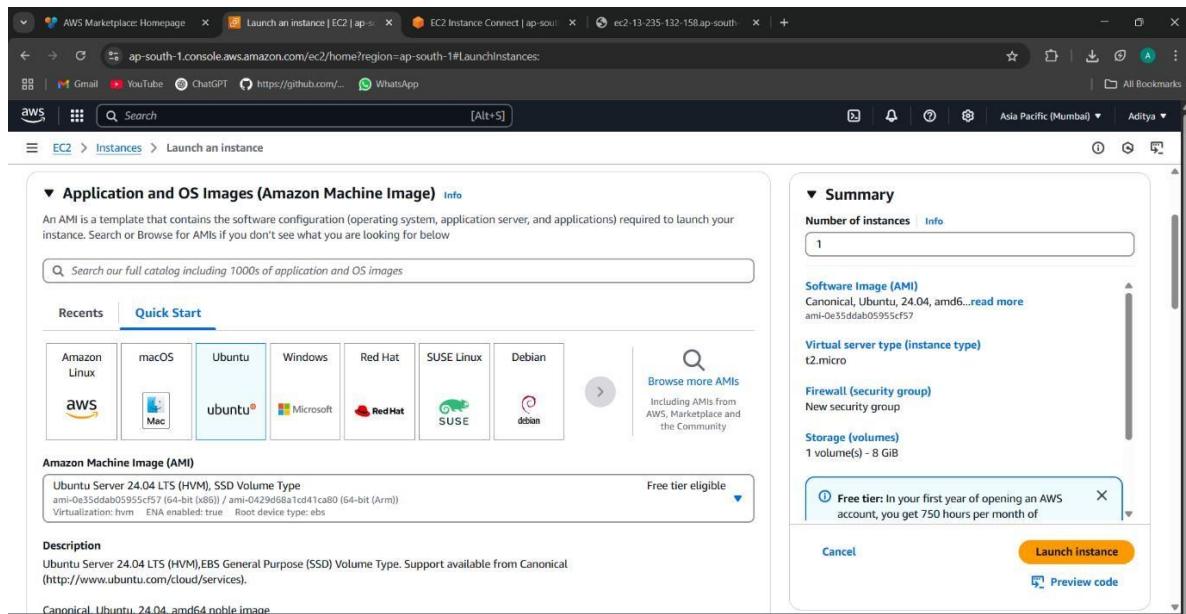
Get started

Take our walkthroughs to help

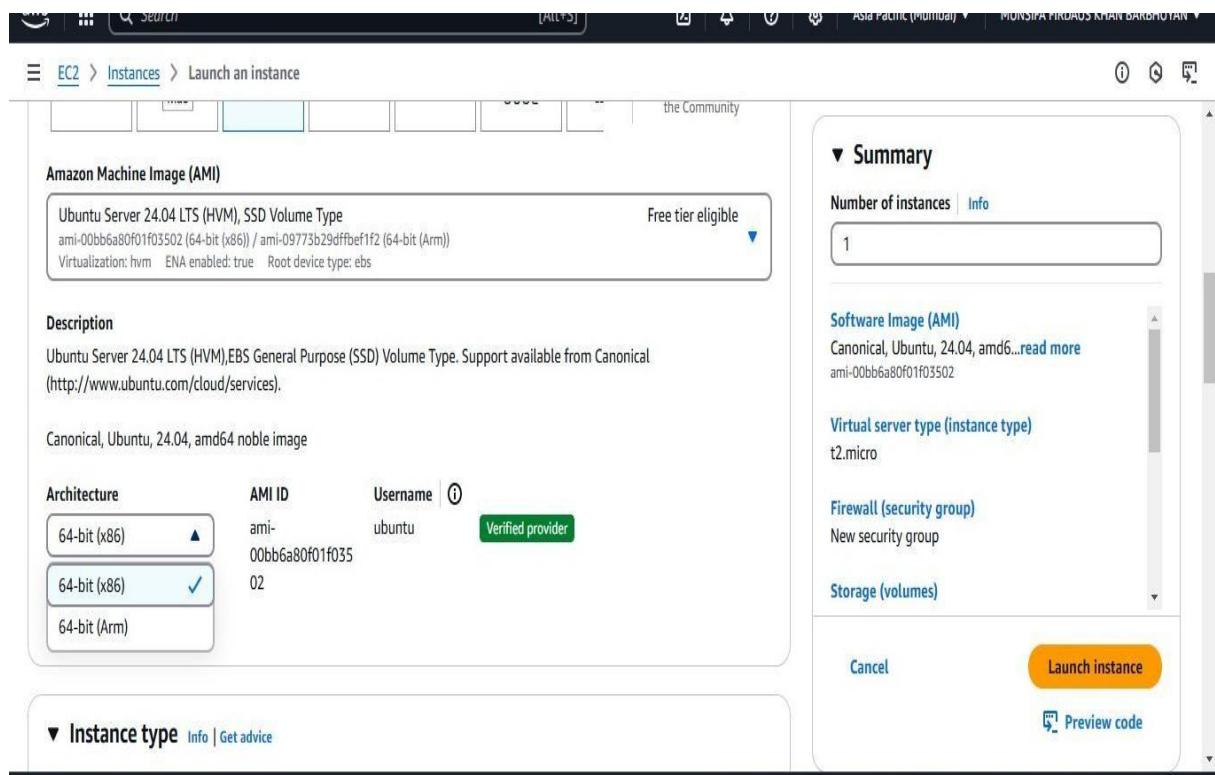
STEP5: TYPE THE NAME OF THE INSTANCE



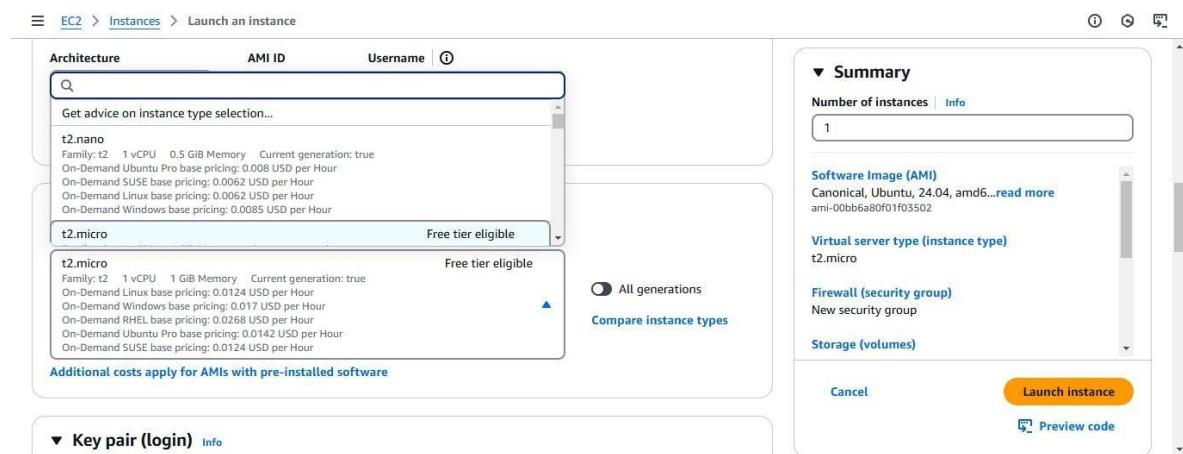
STEP6: CHOOSE THE OPERATING SYSTEM>UBUNTU



STEP7: CHOOSE THE ARCHITECTURE>64-bit(x86)

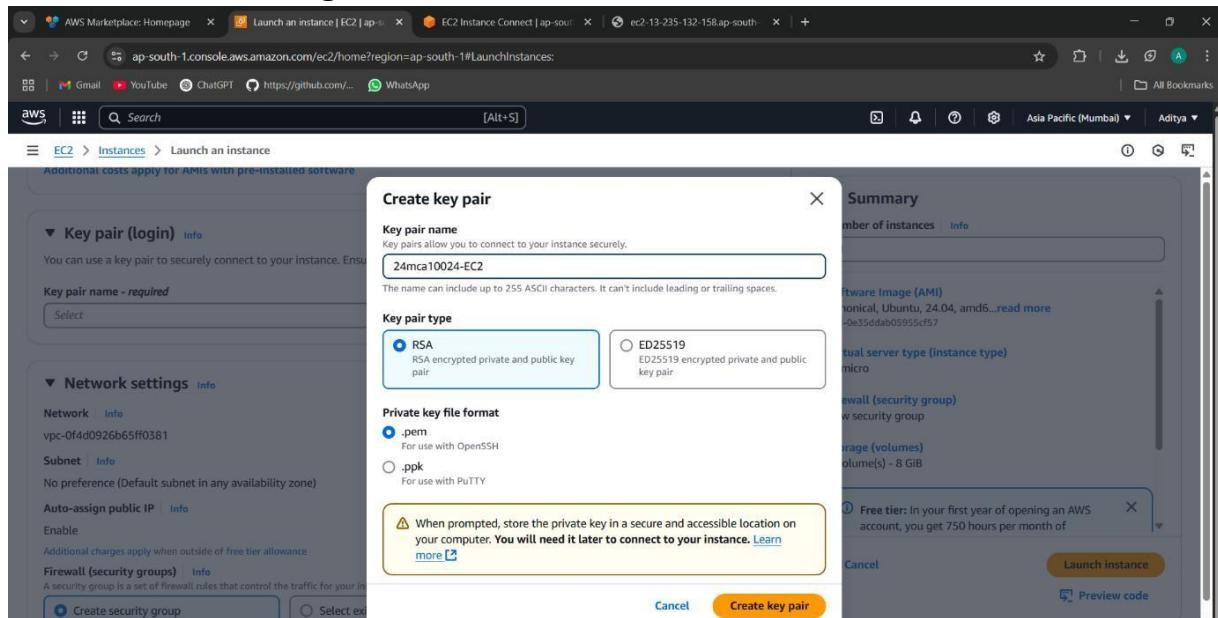


STEP8: CHOOSE INSTANCE TYPE: t2 micro (Free tier eligible)

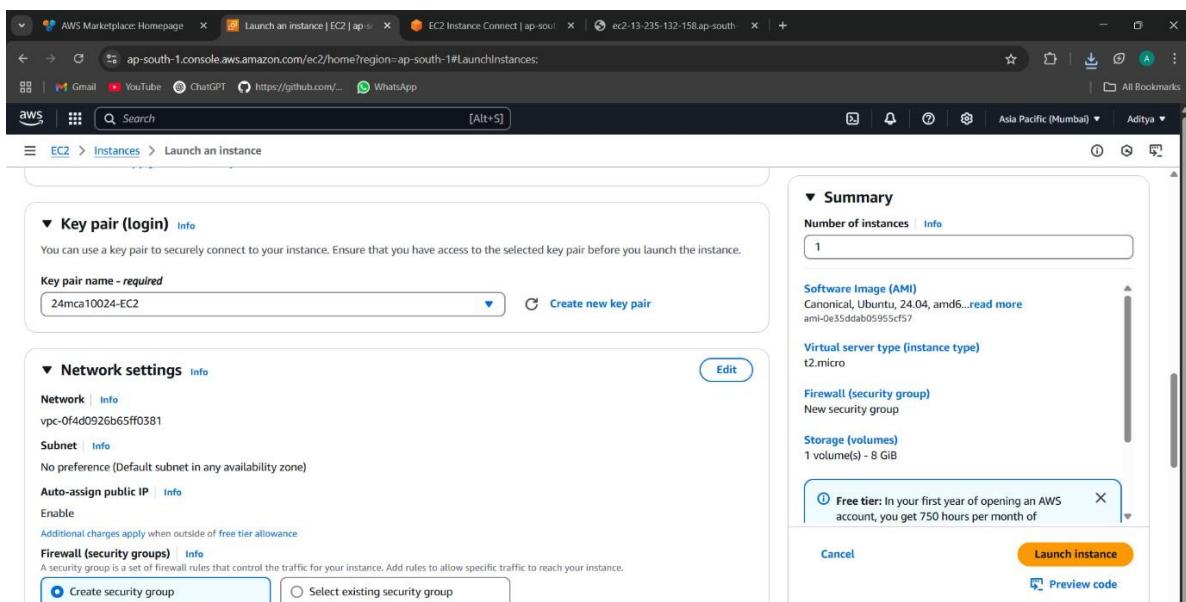


STEP9: CREATE A NEW KEY PAIR:

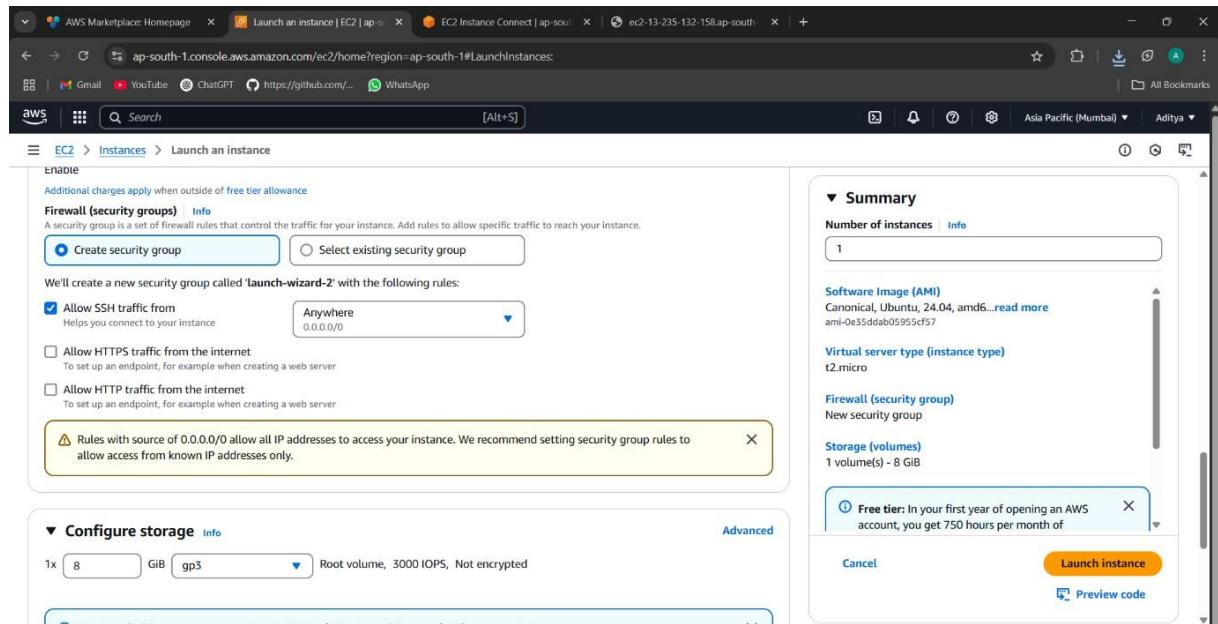
There are two keys involved in this step. One is private key for user and another is public key for the EC2 instance. Once we create a pair, both the keys will be generated. Public key need to saved separately for future use. Both keys will have same name as given.



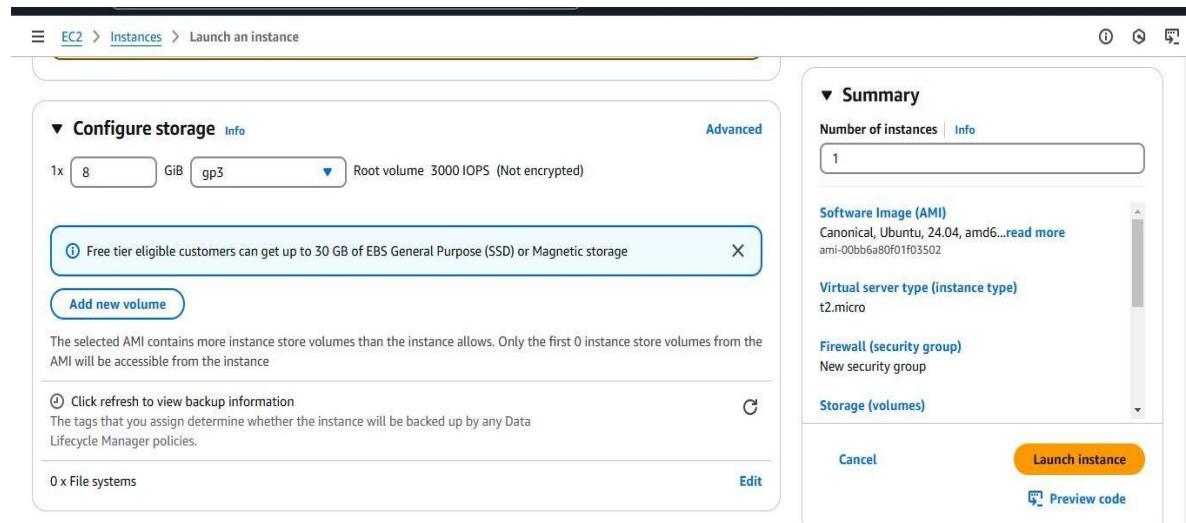
STEP10: PUBLIC KEY IS AUTOMATICALLY ENABLED IN EC2 INSTANCE

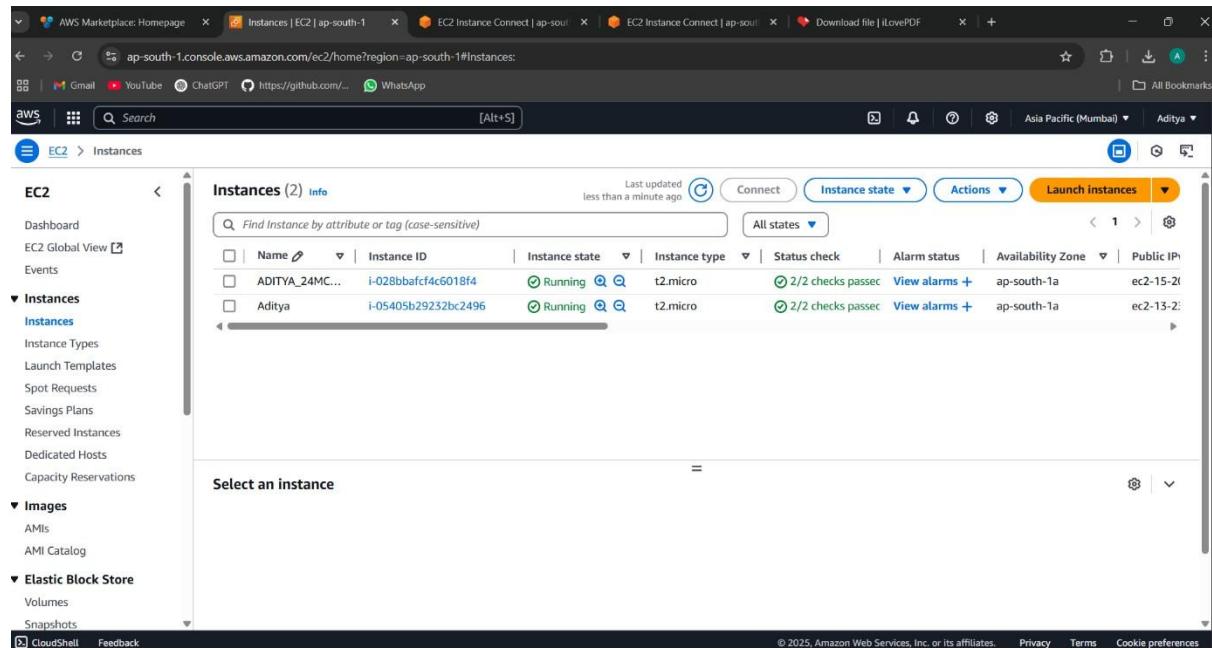


STEP11: CHOOSE THE DEFAULT NETWORK SETTINGS(NOTE: Auto-assign public IP should be enabled and source:0.0.0.0/0 allows all the IP addresses to access the EC2 instance)



STEP12: CHOOSE THE DEFAULT CONFIGURE STORAGE AND CLICK ON LAUNCH INSTANCE



STEP13: EC2 INSTANCE IS CREATED, CLICK ON THE INSTANCE.**STEP14: NOW CLICK ON THE EC2 INSTANCE**

The screenshot shows the AWS EC2 Instances page with two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
ADITYA_24MC...	i-028bbafcfc6018f4	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-15-2...
Aditya	i-05405b29232bc2496	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-2...

The left sidebar shows navigation options like Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store. The Instances section is currently selected. The bottom of the page includes links for CloudShell, Feedback, and copyright information.

STEP15: DETAILS OF THE EC2 INSTANCE IS SHOWN, CLICK ON CONNECT

The screenshot shows the AWS EC2 Instances details page for an instance named i-028bbafcf4c6018f4. The 'Connect' button is highlighted in blue at the top right of the main content area.

Instance summary for i-028bbafcf4c6018f4 (ADITYA_24MCA10024)

- Instance ID:** i-028bbafcf4c6018f4
- IPv6 address:** -
- Hostname type:** IP name: ip-172-31-33-232.ap-south-1.compute.internal
- Answer private resource DNS name:** IPv4 (A)
- Auto-assigned IP address:** 15.206.91.222 [Public IP]
- IAM Role:** -
- IMDSv2:** -
- Public IPv4 address:** 15.206.91.222 [open address]
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-33-232.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0f4d0926b65ff0581 [open]
- Subnet ID:** subnet-0a0130ca69d76c491 [open]
- Instance ARN:** -
- Private IPv4 addresses:** 172.31.33.232
- Public IPv4 DNS:** ec2-15-206-91-222.ap-south-1.compute.amazonaws.com [open address]
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendation s. [Learn more]
- Auto Scaling Group name:** -
- Managed:** -

STEP16: CLICK ON EC2 INSTANCE CONNECT>CONNECT

The screenshot shows the 'Connect to instance' dialog for the same EC2 instance. The 'Public IPv4 address' connection type is selected. The 'Connect' button is highlighted in yellow at the bottom right.

Connect to instance

Connect to your instance i-028bbafcf4c6018f4 (ADITYA_24MCA10024) using any of these options

EC2 Instance Connect **Session Manager** **SSH client** **EC2 serial console**

Instance ID: i-028bbafcf4c6018f4 (ADITYA_24MCA10024)

Connection Type:

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
- Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address: 15.206.91.222

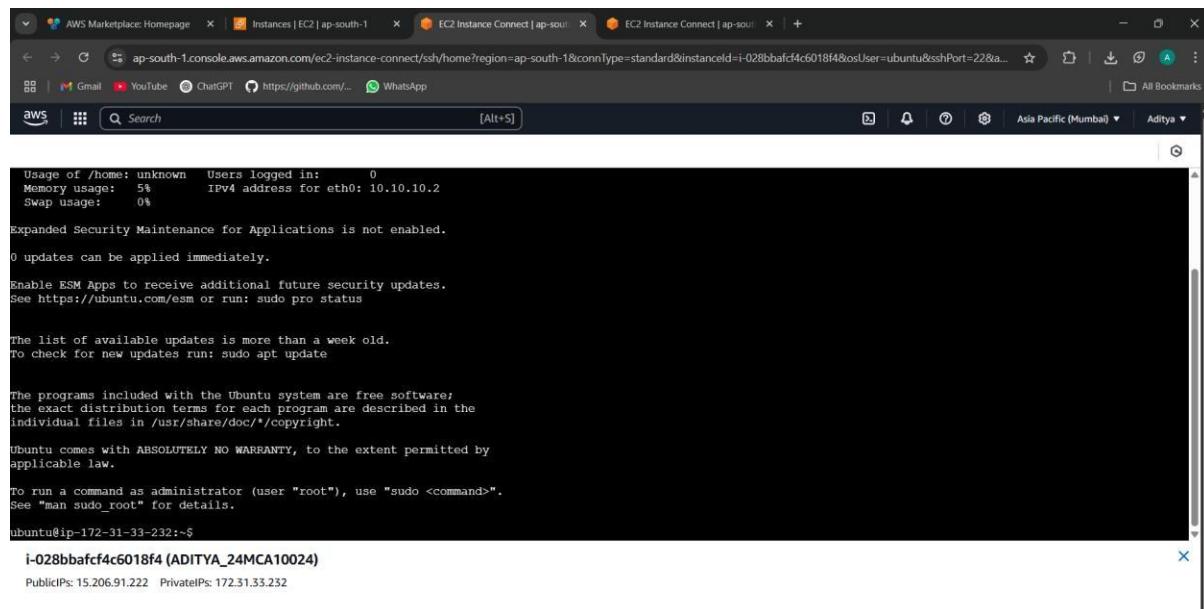
IPv6 address: -

Username: Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel **Connect**

STEP17: IT WILL LAND YOU TO THE TERMINAL PAGE



The screenshot shows a web browser window with multiple tabs open. The active tab is titled "EC2 Instance Connect | ap-south-1" and displays a terminal session on an Ubuntu 22.04 LTS system. The terminal output includes system usage statistics, security maintenance information, and a copyright notice. At the bottom, it shows the user's name, IP address, and public/private IP details.

```
Usage of /home: unknown  Users logged in:      0
Memory usage:  5%      IPv4 address for eth0: 10.10.10.2
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
see https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
to check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-33-232:~$
```

i-028bbafcf4c6018f4 (ADITYA_24MCA10024)
PublicIPs: 15.206.91.222 PrivateIPs: 172.31.33.232

Date: 23/05/2025	Title
Exp. No: 02	How to launch EC2 Instance in Windows.

HOW TO LAUNCH AN EC2 INSTANCE IN WINDOWS

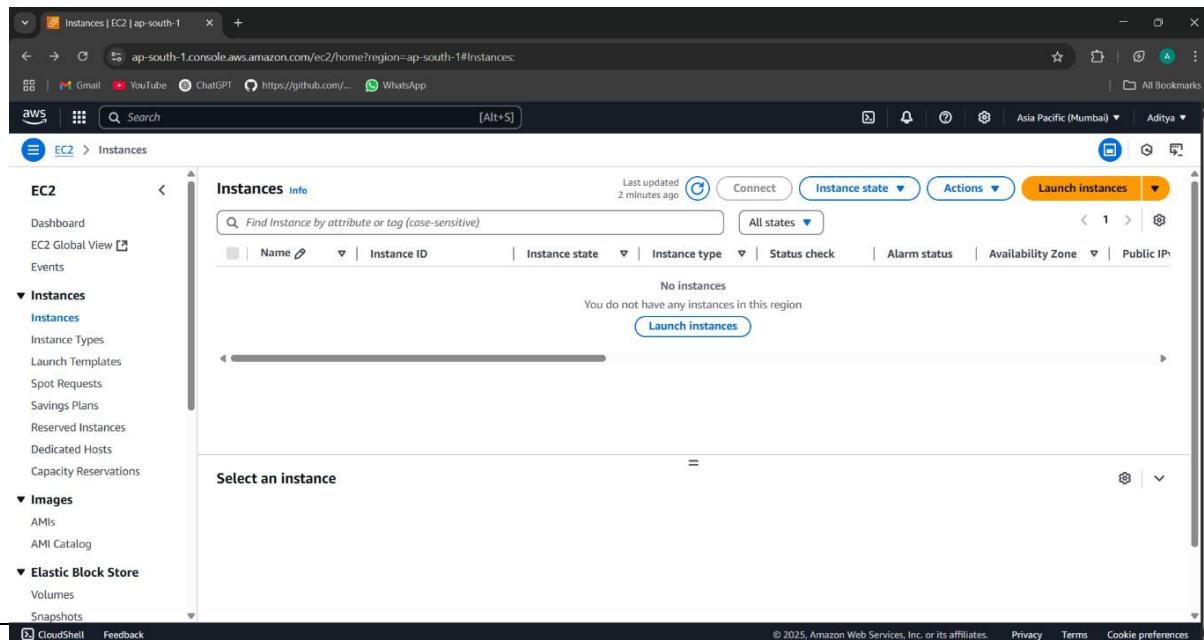
Launching an EC2 instance from a Windows system is typically done through the **Amazon Management Console**, a web-based interface that provides an intuitive and user-friendly way to provision cloud resources.

The process begins by logging into the AWS Console and navigating to the EC2 dashboard. From there, users can initiate the instance launch process by choosing an **Amazon Machine Image (AMI)**—which determines the operating system (e.g., Amazon Linux, Ubuntu, or Windows Server)—and selecting an **instance type** that defines the hardware specifications like CPU and memory.

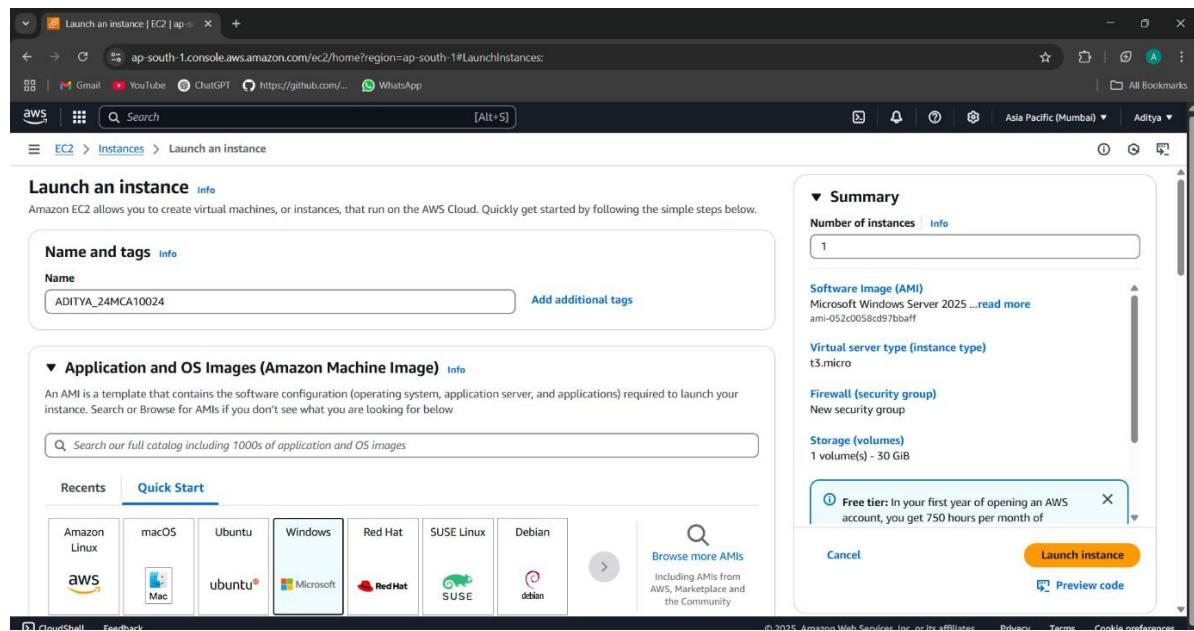
Next, users configure instance settings, including networking (VPC and subnet), storage, security groups (firewall rules), and an SSH key pair (or RDP credentials for Windows AMIs) for remote access. Once launched, the instance appears in the EC2 dashboard and can be accessed using tools like **PuTTY** (for Linux-based instances) or **Remote Desktop Connection (RDP)** (for Windows-based instances).

This GUI-based method is ideal for beginners or those who prefer visual configuration over command-line tools. It allows detailed customization and easy monitoring, making it a straightforward way to deploy virtual servers in the AWS Cloud using a Windows environment.

STEP1: LOG IN WITH ROOT USER>EC2 DASHBOARD>CLICK ON INSTANCES> LAUNCH INSTANCES



STEP2: WRITE AN INSTANCE NAME



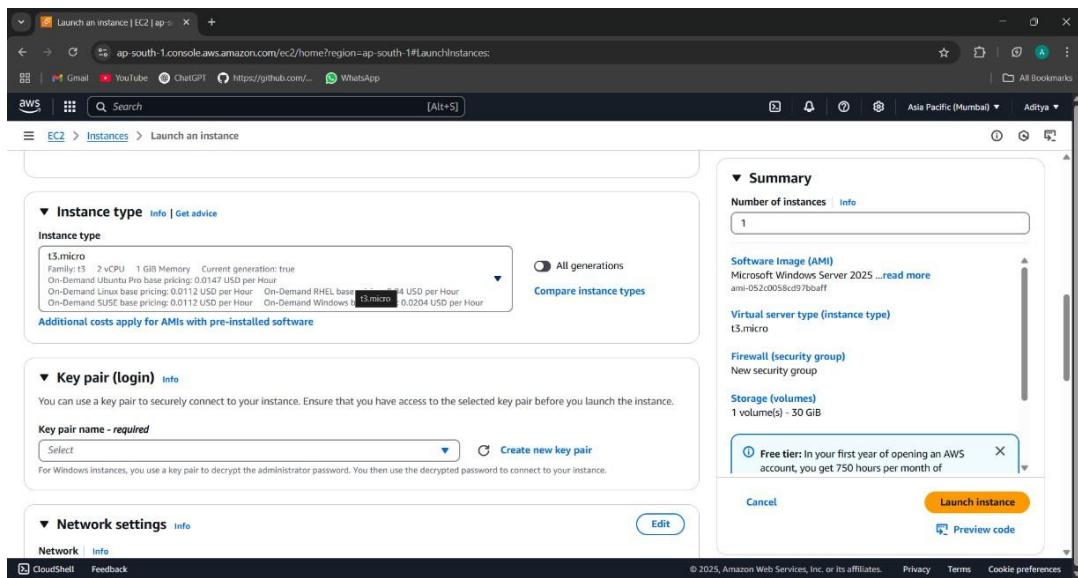
STEP3: CHOOSE THE OS-WINDOWS

The screenshot shows the AWS EC2 'Launch an instance' page. In the 'Amazon Machine Image (AMI)' section, the 'Microsoft Windows Server 2025 Base' AMI is selected. It is described as 'Free tier eligible'. The 'Description' section notes it's the Microsoft Windows 2025 Datacenter edition in English. On the right, a 'Summary' sidebar provides details: 'Number of instances' (1), 'Software image (AMI)' (Microsoft Windows Server 2025), 'Virtual server type' (t3.micro), 'Firewall (security group)' (New security group), and 'Storage (volumes)' (1 volume(s) - 30 GB). A 'Launch instance' button is at the bottom.

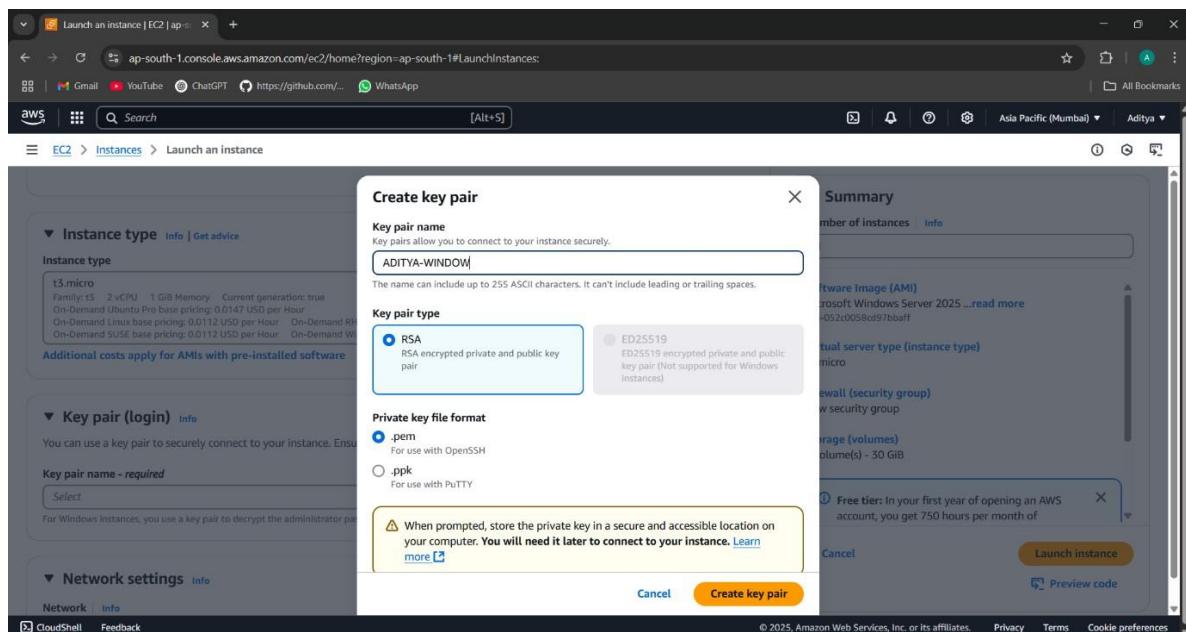
STEP4: CHOOSE THE LOCAL REGION

The screenshot shows the AWS EC2 'Launch an instance' page with the 'Region' dropdown open. The 'ap-south-1' region is selected. The dropdown menu lists regions grouped by continent: United States (N. Virginia, us-east-1; Ohio, us-east-2; N. California, us-west-1; Oregon, us-west-2), Asia Pacific (Mumbai, ap-south-1; Osaka, ap-northeast-3; Seoul, ap-northeast-2; Singapore, ap-southeast-1; Sydney, ap-southeast-2; Tokyo, ap-northeast-1), Canada (Central, ca-central-1), Europe (Frankfurt, eu-central-1; Ireland, eu-west-1; London, eu-west-2; Paris, eu-west-3), and others. Other visible regions include 'us-east-1', 'us-east-2', 'us-west-1', 'us-west-2', 'ap-northeast-3', 'ap-northeast-2', 'ap-southeast-1', 'ap-southeast-2', 'ap-northeast-1', 'ca-central-1', 'eu-central-1', 'eu-west-1', 'eu-west-2', and 'eu-west-3'. The 'Manage Regions' and 'Manage Local Zones' buttons are at the bottom of the dropdown.

STEP5: CHOOSE THE INSTANCE TYPE AS t3 micro



STEP6: CREATE THE KEY PAIR



STEP7: CHOOSE THE PUBLIC KEY THAT YOU HAVE CREATED FROM THE DROP DOWN

STEP8: CHOOSE FROM NETWORK SETTINGS> CREATE SECURITY GROUP> ALLOW RDP TRAFFIC FROM> ENABLE AUTO-ASSIGN PUBLIC IP

STEP9: CHOOSE 30GB FOR CONFIGURE STORAGE

The screenshot shows the 'Configure storage' section of the AWS EC2 instance launch wizard. A 30 GiB gp3 volume is selected as the root volume. A tooltip indicates that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. The summary panel on the right shows one instance being launched with a Microsoft Windows Server 2025 AMI, t3.micro instance type, and a new security group. A note about free tier usage is also present.

STEP10: CLICK ON LAUNCH INSTANCE

The screenshot shows the confirmation page after launching the instance. It displays a green success message: 'Successfully initiated launch of instance (i-09dee0c706e5952)'. Below this, there's a 'Launch log' section and a 'Next Steps' section with links to 'Create billing and free tier usage alerts', 'Connect to your instance', 'Connect an RDS database', and 'Create EBS snapshot policy'.

STEP11: CLICK ON THE INSTANCE THAT IS CREATED

The screenshot shows the AWS EC2 'Launch an instance' page. At the top, there's a green success message: 'Successfully initiated launch of instance i-09dee0c0c706e5952'. Below this, there's a 'Launch log' section with a link to 'Launch log'. Under 'Next Steps', there are four cards: 'Create billing and free tier usage alerts', 'Connect to your instance', 'Connect an RDS database', and 'Create EBS snapshot policy'. Each card has a corresponding button: 'Create billing alerts', 'Connect to instance', 'Connect an RDS database', and 'Create EBS snapshot policy'. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and a footer with copyright information.

STEP12: CLICK ON CONNECT

The screenshot shows the AWS EC2 'Instance details' page for the instance i-09dee0c0c706e5952. On the left, there's a sidebar with navigation links for EC2, Instances, Images, and Elastic Block Store. The main content area displays the instance summary for 'ADITYA_24MCA10024'. It includes sections for Instance ID (i-09dee0c0c706e5952), IPv6 address, Hostname type (IP name: ip-172-31-10-132.ap-south-1.compute.internal), Answer private resource DNS name (IPv4 A), Auto-assigned IP address (13.232.90.103 [Public IP]), IAM Role, IMDSv2, Public IPv4 address (13.232.90.103), Instance state (Running), Private IP DNS name (IPv4 only) (ip-172-31-10-132.ap-south-1.compute.internal), Instance type (t3.micro), VPC ID (vpc-0fd0d0926b65ff0381), Subnet ID (subnet-0c05fc0c70e6893ce), and Instance ARN. On the right, there are buttons for 'Connect', 'Instance state', and 'Actions'. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and a footer with copyright information.

STEP13: CLICK ON RDP CLIENT>CONNECT USING RDP CLIENT> GET PASSWORD

The screenshot shows the AWS EC2 Connect to instance page for an instance with ID i-09dee0c0c706e5952. The 'RDP client' tab is selected. It displays options for connecting using RDP client or Fleet Manager. A link to download the remote desktop file is provided. Below, there are fields for Public DNS (ec2-13-232-90-103.ap-south-1.compute.amazonaws.com) and Username (Administrator). A note about directory credentials is present. The bottom of the page includes CloudShell, Feedback, and copyright information.

STEP14: HERE COPY & PASTE THE PRIVATE KEY THAT WAS DOWNLOADED EARLIER

The screenshot shows the AWS EC2 Get Windows password page for the same instance. It prompts for a private key to decrypt the administrator password. A 'Private key' section allows uploading a file, with a note that contents are optional. A text area for 'Private key contents' is also shown. At the bottom, there are 'Cancel' and 'Decrypt password' buttons.

STEP15: CLICK ON DECRYPT PASSWORD

The screenshot shows a browser window with the URL ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#GetWindowsPasswordInstanceId=i-09dee0c706e5952;previousPlace=ConnectToInstance;lang=English. The page title is "Get Windows password". It contains a section titled "Get Windows password" with a "info" link. Below it, a note says "Use your private key to retrieve and decrypt the initial Windows administrator password for this instance." It shows the "Instance ID" as "i-09dee0c706e5952 (ADITYA_24MCA10024)" and the "Key pair associated with this instance" as "ADITYA-WINDOW". A "Private key" section allows uploading a private key file or pasting its contents. A large text area contains a long RSA private key. At the bottom right are "Cancel" and "Decrypt password" buttons.

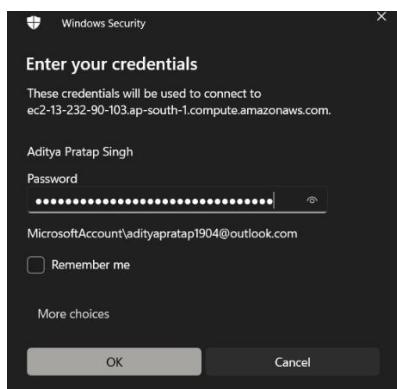
STEP16: COPY THE USERNAME, PASSWORD AND PUBLIC DNS IN A SEPARATE FILE

The screenshot shows the "Connect to instance" dialog for an EC2 instance with ID "i-082c5ceedfaf65e6f" named "mansifa-windows". It has two main sections: "Connection Type" and "Public DNS". Under "Connection Type", there are two options: "Connect using RDP client" (selected) and "Connect using Fleet Manager". The "RDP client" section includes a download link for "Download remote desktop file". The "Public DNS" section shows the address "ec2-35-154-209-141.ap-south-1.compute.amazonaws.com" and a dropdown for "Username Info" set to "Administrator". A note at the bottom says "If you've joined your instance to a directory, you can use your directory credentials to connect to your instance." At the bottom right is a "Cancel" button.

STEP17: ENABLE REMOTE DESKTOP IN YOUR MACHINE AND TYPE/PASTE PUBLIC DNS IN COMPUTER AND CLICK ON CONNECT



STEP18: TYPE USER NAME AND PASSWORD THAT IS SAVED EARLIER AND CLICK OK. YOUR REMOTE DESKTOP MACHINE FOR WINDOWS WILL OPEN



Date: 23/05/2025	Title
Exp. No: 03	How to launch an EC2 Instance with Template.

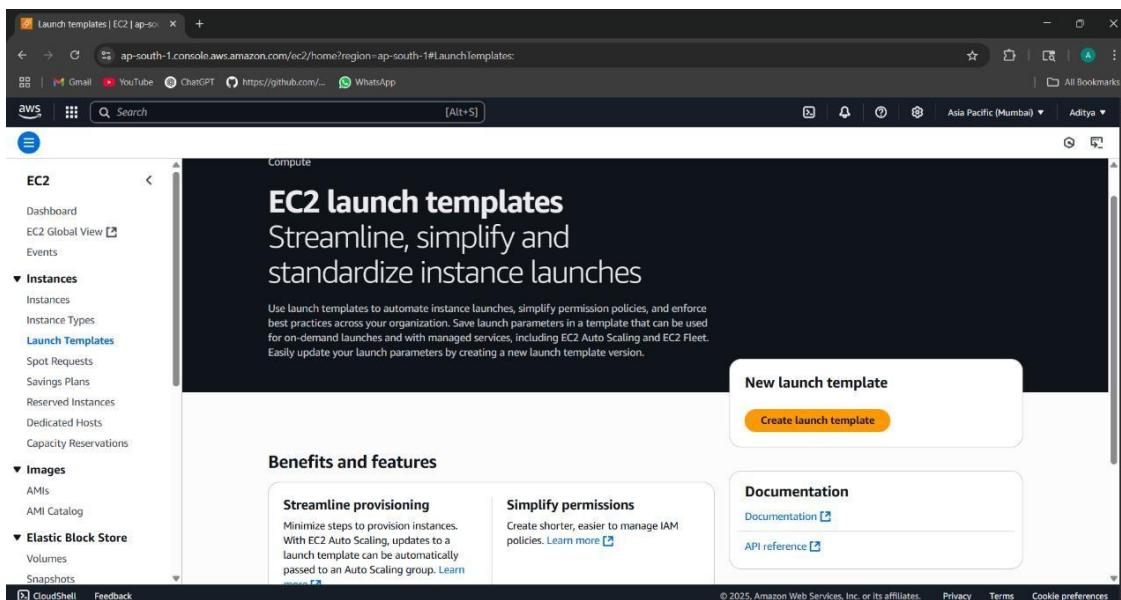
HOW TO LAUNCH AN EC2 INSTANCE WITH TEMPLATE

A **Launch Template** in AWS EC2 is a reusable configuration that defines the parameters needed to launch an instance, such as the Amazon Machine Image (AMI), instance type, key pair, security groups, network settings, and storage volumes. Using a launch template simplifies and standardizes the instance creation process, ensuring consistency across multiple deployments.

To launch an EC2 instance using a launch template, users first create the template by specifying the desired configuration. This includes details like the operating system, instance size, user data scripts, and associated IAM roles. Once the template is created, it can be used to launch new instances directly from the EC2 dashboard, AWS CLI, or as part of an Auto Scaling Group.

This method is particularly useful in environments that require repeatable and automated instance deployments. It reduces the risk of human error, saves time, and supports versioning, allowing updates and rollbacks to instance configurations when needed. Launch templates are commonly used in production environments, CI/CD pipelines, and when working with scaling solutions in AWS.

STEP1: OPEN THE EC2 DASHBOARD AND CLICK ON LAUNCH TEMPLATES>CREATE LAUNCH TEMPLATE



STEP2: GIVE A SUITABLE NAME AND DESCRIPTION TO THE TEMPLATE

Create launch template | EC2 | + ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate: Gmail YouTube ChatGPT https://github.com/... WhatsApp

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

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
aditya_ec2_template

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

Template version description

This template is for demo purpose
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

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)

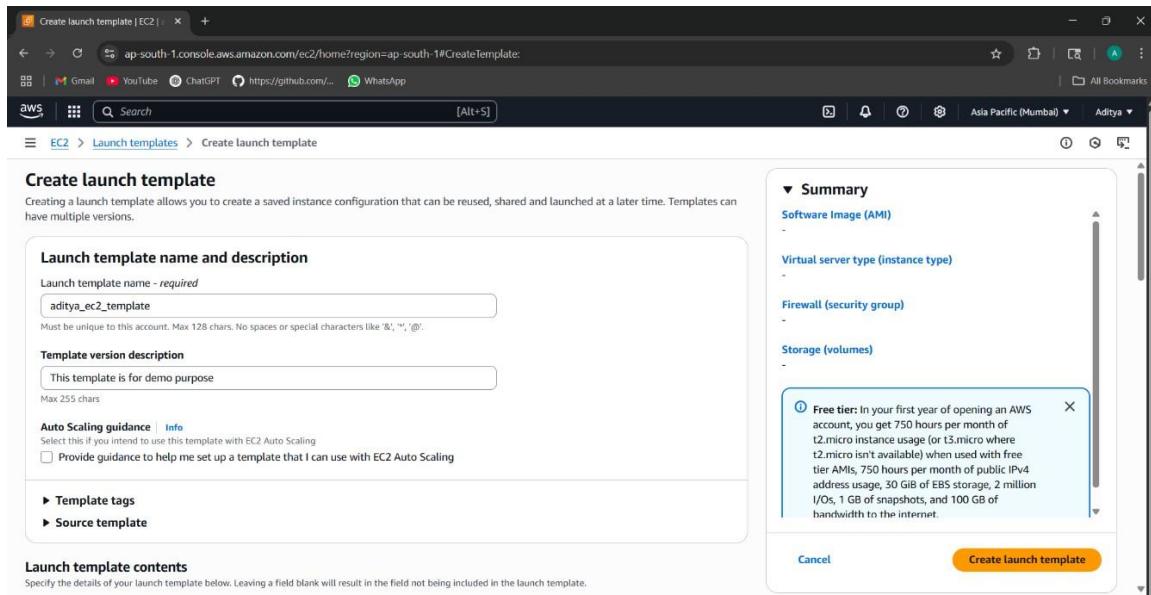
Virtual server type (instance type)

Firewall (security group)

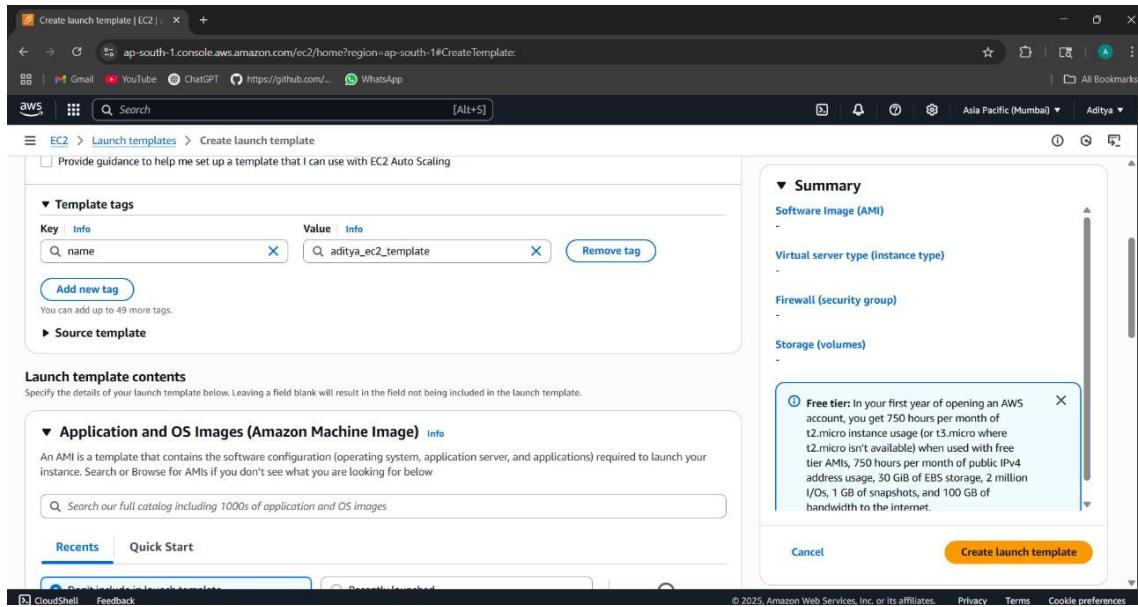
Storage (volumes)

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 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

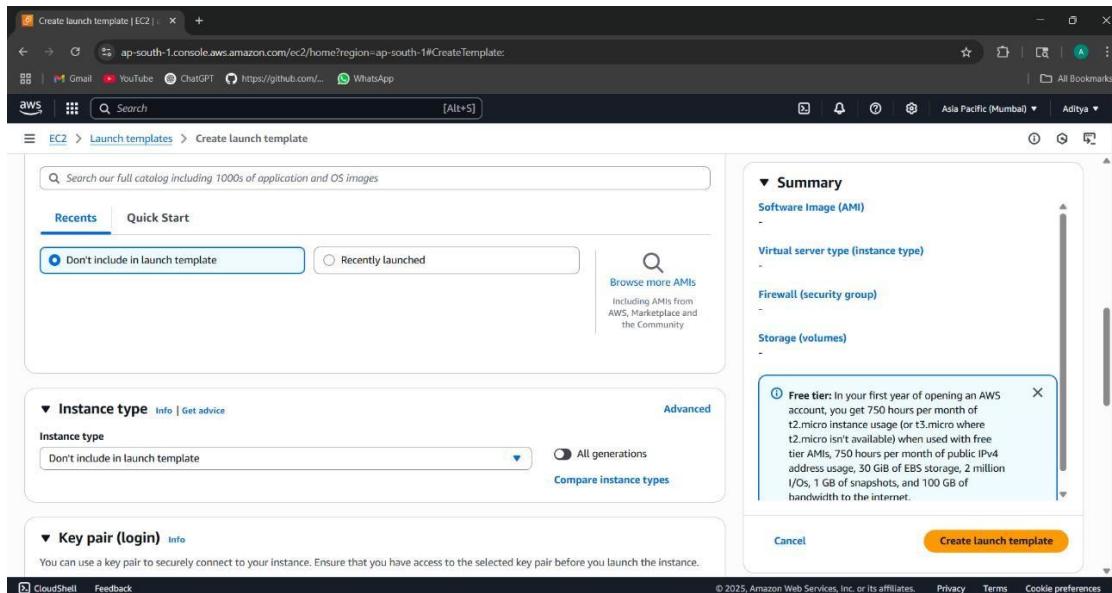
Cancel Create launch template



STEP3: CLICK ON TEMPLATE TAGS>ADD TAGS>INFO:NAME>VALUE:ANY NAME



STEP4: CLICK ON BROWSE MORE AMI



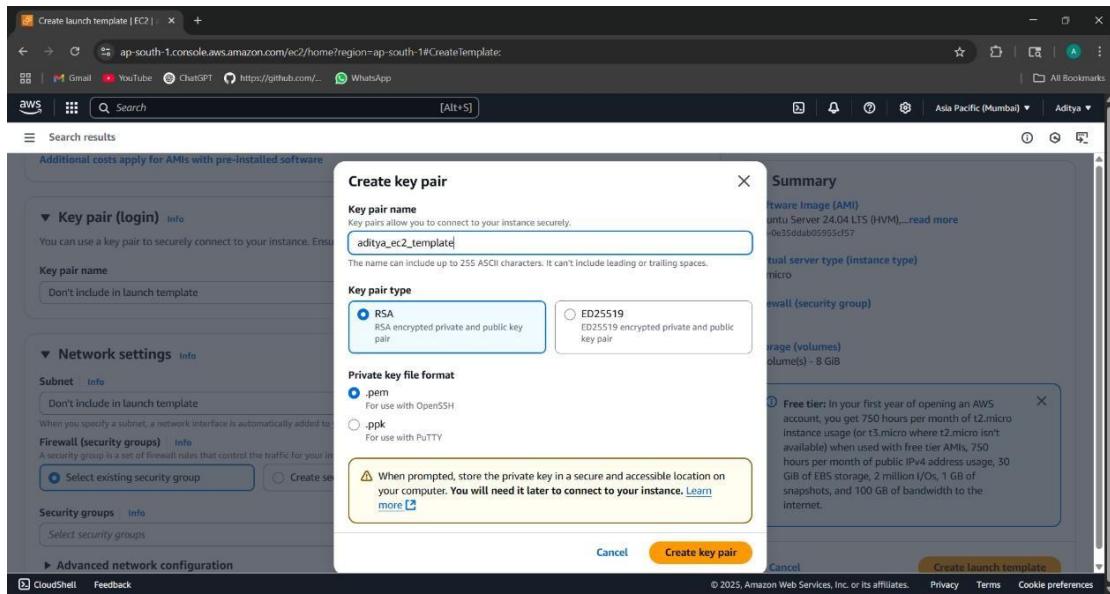
STEP5: SELECT UBUNTU FROM THE LIST

The screenshot shows the AWS EC2 'Create launch template' interface. In the top navigation bar, the URL is 'ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate'. The search bar contains 'Search results' and a placeholder 'Search for an AMI by entering a search term e.g. "Windows"'. Below the search bar, there are three tabs: 'Quick Start AMIs (45)', 'My AMIs (0)', and 'AWS Marketplace AMIs (5948)'. The 'AWS Marketplace AMIs' tab is selected, showing 'AWS & trusted third-party AMIs' and 'Published by anyone'. The main content area displays a list of AMIs under the heading 'Verified provider'. The first item is 'ubuntu' (Ubuntu Server 24.04 LTS (HVM), SSD Volume Type), which is highlighted with a green box. It includes details like AMI ID, Platform, Root device type, Virtualization, and ENA enabled status. A 'Select' button and a radio button for '64-bit (x86)' are visible. Other items listed include Microsoft Windows Server 2025 Base and Microsoft Windows Server 2025 Core Base.

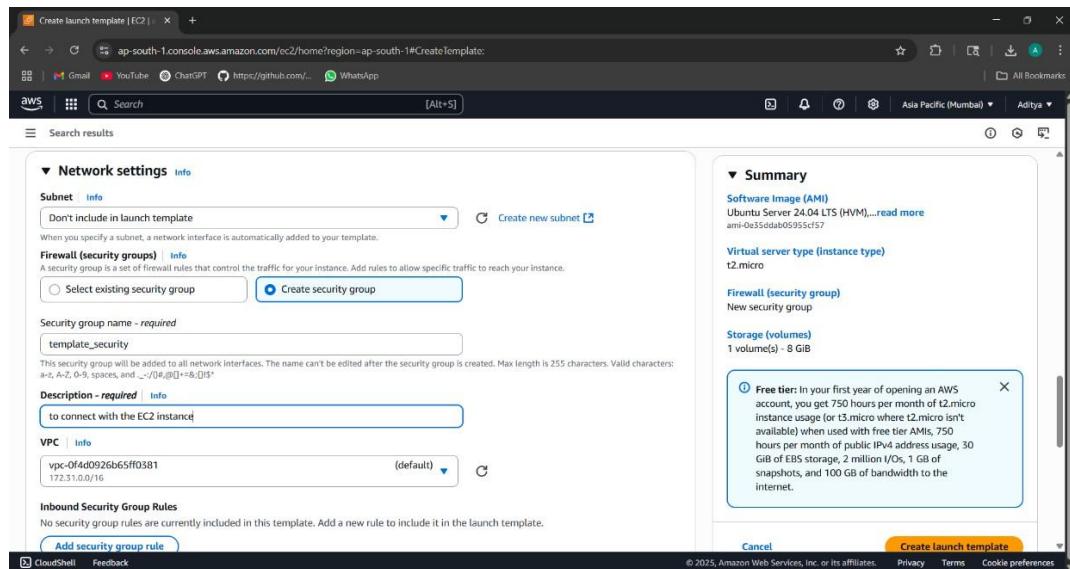
STEP6: SELECT THE INSTANCE TYPE:t2 micro (FREE TIER)

The screenshot shows the 'Create launch template' interface on the 'Instance type' configuration page. The URL is 'ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTemplate'. The 'Instance type' section shows the 't2.micro' option selected, with a green box around it. It provides details such as Family, CPU, Memory, Current generation, and various base pricing options. A note states 'Additional costs apply for AMIs with pre-installed software'. The 'Advanced' button is visible. To the right, the 'Summary' section includes details like Software Image (AMI), Virtual server type (instance type), Firewall (security group), and Storage (volumes). A callout box highlights the 'Free tier' information: '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 snapshots, and 100 GB of bandwidth to the internet.' Buttons for 'Cancel' and 'Create launch template' are at the bottom.

STEP7: CREATE KEY PAIR



STEP8: SELECT CREATE SECURITY GROUP> WRITE THE SECURITY GROUP NAME>DESCRIPTION> CLICK ON ADD SECURITY GROUP RULE



STEP9: CHOOSE TYPE:SSH> SOURCE TYPE: ANYWHERE

VPC: Info
vpc-0f44d925b65ff0381 (default)

Inbound Security Group Rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range	Info
ssh	TCP	22	

Source type: Anywhere

Description - optional: e.g. SSH for admin desktop

Summary:

- Software Image (AMI):** Ubuntu Server 24.04 LTS (HVM)... [read more](#)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New 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.

STEP 10: CLICK ON CREATE LAUNCH>LAUNCH TEMPLATE IS CREATED>CLICK ON VIEW TEMPLATE

Success
Successfully created aditya_ec2_template[lt-02b56e7e7682f7cf9].

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.

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

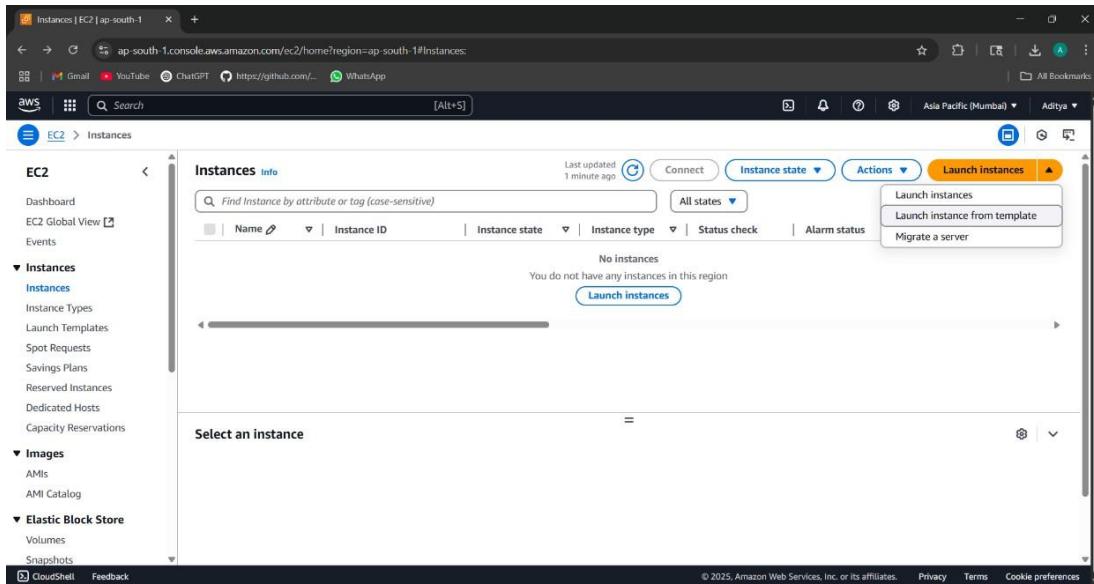
Create Spot Fleet
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance 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

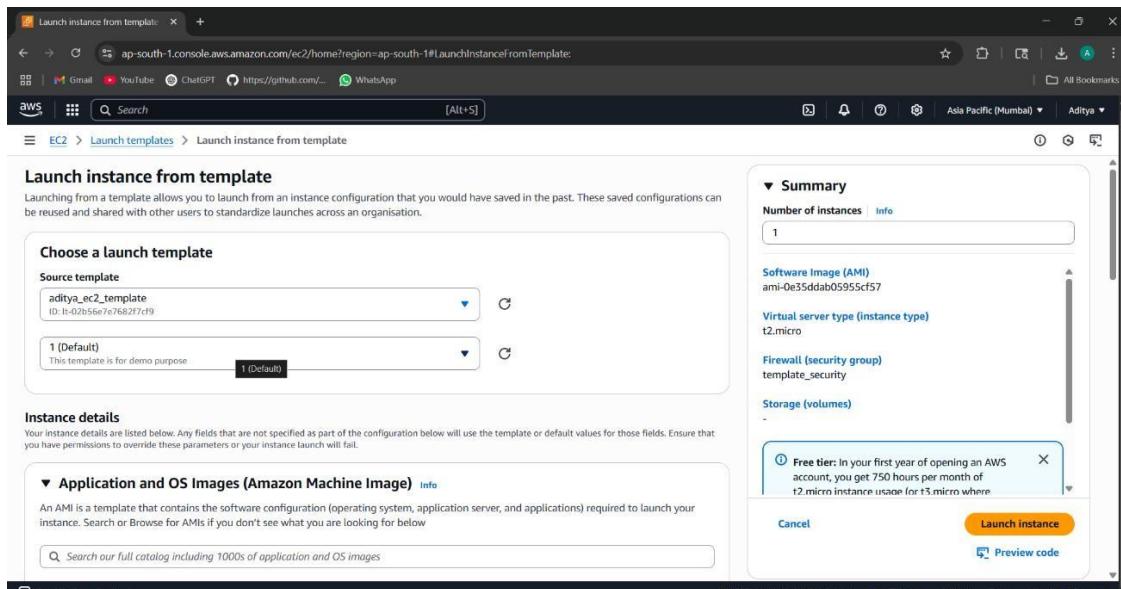
Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-02b56e7e7682f7cf9	aditya_ec2_template	1	1	2025-05-14T04:03:49.000Z	arnawsiam-2705

Select a launch template

STEP 10: CLICK ON INSTANCES >CLICK ON LAUNCH INSTANCES>LAUNCH INSTANCE FROM TEMPLATE



STEP 12: SELECT THE TEMPLATE THAT YOU HAVE CREATED>CLICK ON LAUNCH INSTANCE



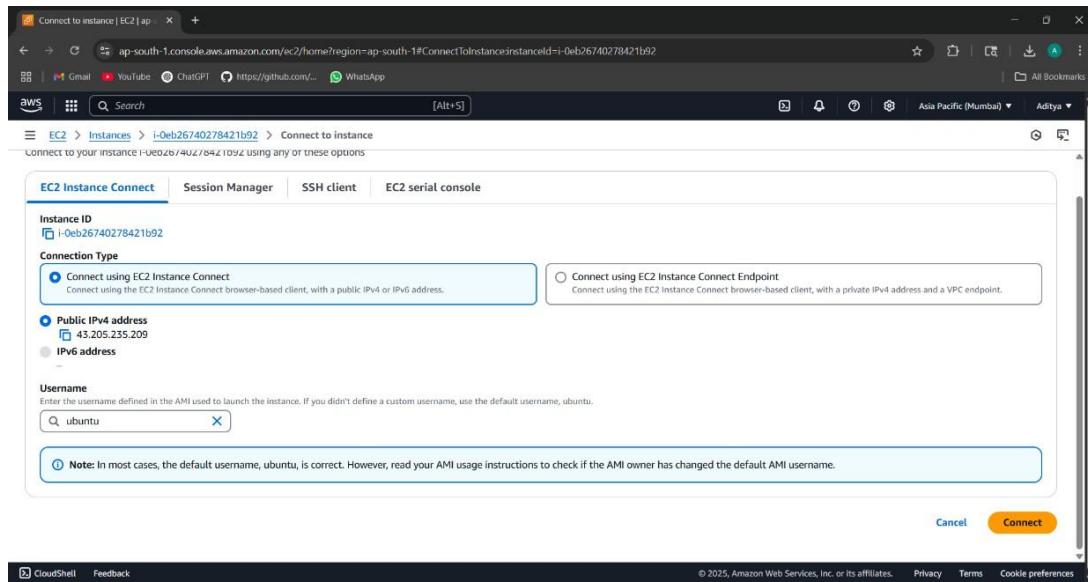
STEP 13: INSTANCE IS CREATED>CLICK ON THE INSTANCE THAT YOU HAVE CREATED

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (selected), AMIs, AMI Catalog, and Elastic Block Store. The main content area has a header 'Instances (1) info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. A table lists one instance: i-0eb26740278421b92, which is Running, t2.micro, Initializing, and located in ap-south-1a with a public IP of ec2-43-21-172-31-44-156.ap-south-1.compute.internal. Below the table is a modal titled 'Select an instance' with a single option: 'i-0eb26740278421b92'. At the bottom right of the main area, there are links for CloudShell, Feedback, and a copyright notice from 2025.

STEP 14: CLICK ON CONNECT

The screenshot shows the AWS EC2 Instance details page for the instance i-0eb26740278421b92. The left sidebar is identical to the previous screenshot. The main content area is titled 'Instance summary for i-0eb26740278421b92' and contains detailed information about the instance. It includes fields for Instance ID (i-0eb26740278421b92), IPv6 address (empty), Hostname type (IP name: ip-172-31-44-156.ap-south-1.compute.internal), Answer private resource DNS name (empty), Auto-assigned IP address (43.205.235.209 [Public IP]), IAM Role (empty), and IMDSv2 (empty). On the right side, there are sections for network details: Public IPv4 address (43.205.235.209 [open address]), Instance state (Running), Private IP4 addresses (172.31.44.156), Public IPv4 DNS (ec2-43-205-235-209.ap-south-1.compute.amazonaws.com [open address]), Instance type (t2.micro), VPC ID (vpc-0f4d0926b65ff0381 []), Subnet ID (subnet-0a0130ca69d76c491 []), and Instance ARN (Managed). At the top right of the main content area, there are buttons for Connect, Instance state, and Actions. The bottom of the page shows a URL (https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectForInsta...).

STEP 15: CLICK ON CONNECT TO INSTANCE



STEP 16: EC2 INSTANCE IS CONNECTED

```
Usage of /: 25.0% of 6.71GB  Users logged in: 0
Memory usage: 20%          IPv4 address for enx0: 172.31.44.156
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable KSM Apps to receive additional future security updates.
See https://ubuntu.com/csm or run sudo pro status

the list of available updates is more than a week old.
to check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
to run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
i-0eb26740278421b92:~$
```

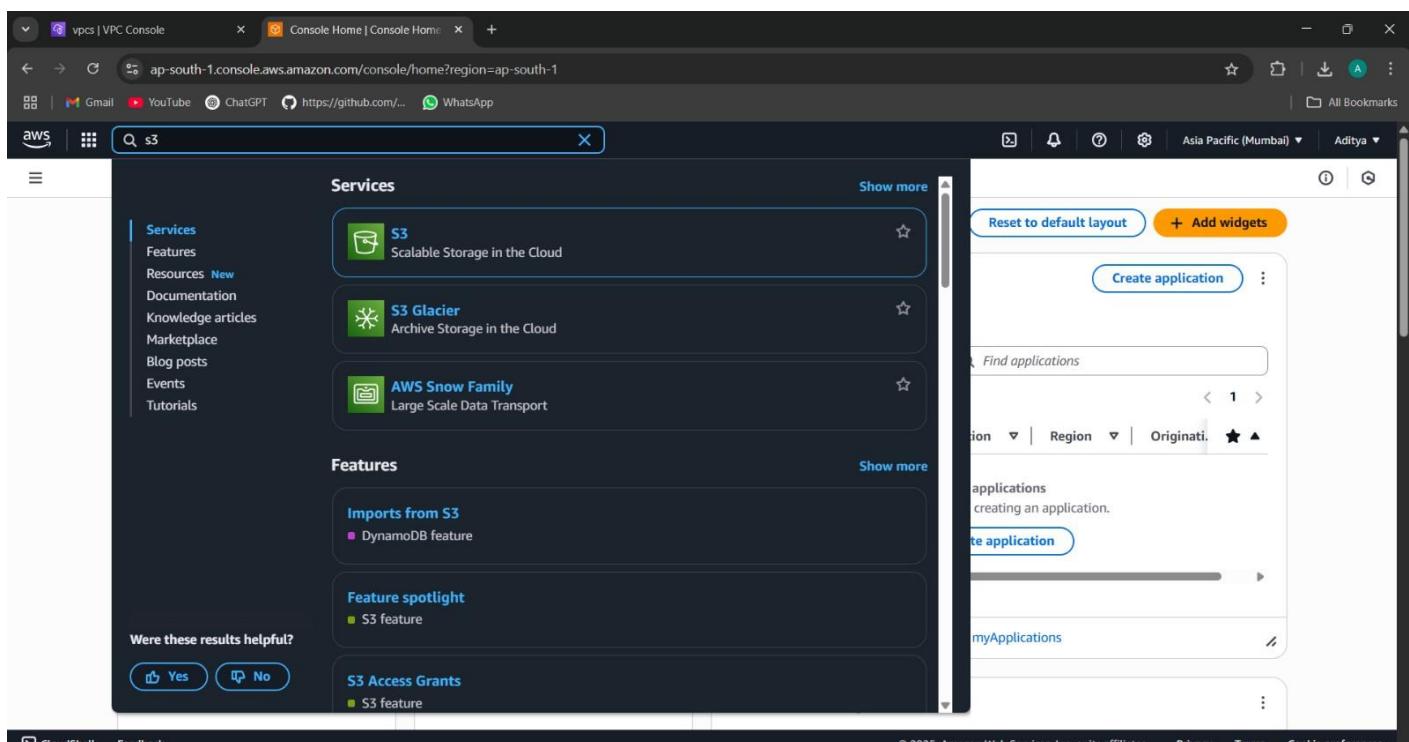
PublicIP: 43.205.235.209 PrivateIP: 172.31.44.156

Date: 23/05/2025	Title
Exp. No: 04	Amazon S3 (Bucket Creation, Creating URL and S3 Life Cycle Management.)

Amazon S3 (Bucket creation, creating URL and S3 Life cycle Management)

Amazon S3 (Simple Storage Service) is a cloud-based object storage service used to store and retrieve files. Users begin by creating a bucket, which serves as a container for their data. After uploading files, S3 generates unique URLs that allow access to the objects, either publicly or through secure, time-limited links. Lifecycle management lets users automate actions like moving files to cheaper storage classes or deleting them after a set time, helping to optimize storage costs and manage data efficiently.

Step 1: -Get in storage and then in storage dashboard.



Step 2: - Since no bucket is created so click on create bucket.

Homepage | S3 | ap-south-1

ap-south-1.console.aws.amazon.com/s3/get-started?region=ap-south-1

Gmail YouTube ChatGPT https://github.com/... WhatsApp

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

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

[Create bucket](#)

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the [AWS Simple Monthly Calculator](#)

[View pricing details](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 3: -Since the AWS region is already selected.

The screenshot shows the 'Create bucket' configuration page for AWS S3. In the 'Bucket type' section, 'General purpose' is selected. The 'Bucket name' field contains 'aditya-24mca10024'. The 'Object Ownership' section is visible at the bottom.

Step 4: - Now creating a bucket with unique name which does not contain caps.

The screenshot shows the 'Create bucket' configuration page for AWS S3. In the 'Bucket type' section, 'General purpose' is selected. The 'Bucket name' field contains 'aditya-24mca10024'. The 'Object Ownership' section is visible at the bottom.

Step 5: - Now for giving the security we provide encryption to Bucket.

The screenshot shows the 'Create bucket' page in the AWS S3 console. Under the 'Default encryption' section, 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' is selected. Below it, there's a 'Bucket Key' section where 'Enable' is selected. A note at the bottom says 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.' At the bottom right are 'Cancel' and 'Create bucket' buttons.

Step 6: -Now in this step bucket is created with name Aditya-24mca10024

The screenshot shows the 'Buckets' page in the AWS S3 console. A green notification bar at the top says 'Successfully created bucket "aditya-24mca10024"'. Below it, there's an 'Account snapshot' section and a table for 'General purpose buckets'. The table shows one bucket named 'aditya-24mca10024' with details: Name, AWS Region (Asia Pacific (Mumbai) ap-south-1), IAM Access Analyzer (View analyzer for ap-south-1), and Creation date (May 22, 2025, 09:36:14 (UTC+05:30)). At the bottom right are 'Copy ARN', 'Empty', 'Delete', and 'Create bucket' buttons.

Step 7: -Now we upload file by clicking upload.

The screenshot shows the AWS S3 console interface. The top navigation bar includes links for Gmail, YouTube, ChatGPT, GitHub, and WhatsApp. The main content area is titled 'aditya-24mca10024' with an 'Info' link. Below this, there are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected, showing a message stating 'Objects (0)'. It provides options to 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar allows filtering by prefix. Below the search bar, there are columns for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message indicates 'No objects' and 'You don't have any objects in this bucket.' A large blue 'Upload' button is centered at the bottom of the list.

Step 8:-Now under upload section we can upload the file as well as folder.

The screenshot shows the AWS S3 console interface, specifically the 'Upload' section. The top navigation bar includes links for Gmail, YouTube, ChatGPT, GitHub, and WhatsApp. The main content area is titled 'Upload objects - S3 bucket aditya-24mca10024' with an 'Info' link. Below this, there are tabs for 'Upload' and 'Info'. The 'Upload' tab is selected, with a message instructing users to 'Add files or Add folder' by dragging and dropping files or choosing them from a dropdown menu. A 'Files and folders' table lists one item: 'ADITYA.JPG' (1 total, 169.9 KB). The table includes columns for 'Name', 'Folder', 'Type', and 'Size'. Buttons for 'Remove', 'Add files', and 'Add folder' are located at the top right of the table. Below the table, a 'Destination' section shows the path 's3://aditya-24mca10024'. A 'Destination details' section provides information about bucket settings. The bottom of the screen includes standard AWS navigation links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Step 9:- In storage class there are different types so choose any one as per use.

The screenshot shows the 'Storage class' selection interface in the AWS S3 console. It lists various storage classes with their characteristics and fees:

Storage class	Designed for	Bucket type	Availability Zones	Min storage duration	Min billable object size	Monitoring and auto-tiering fees	Retrieval fees
Standard	Frequently accessed data (more than once a month) with milliseconds access	General purpose	≥ 3	-	-	-	-
Intelligent-Tiering	Data with changing or unknown access patterns	General purpose	≥ 3	-	-	Per-object fees apply for objects >= 128 KB	-
Standard-IA	Infrequently accessed data (once a month) with milliseconds access	General purpose	≥ 3	30 days	128 KB	-	Per-GB fees apply
One Zone-IA	Recreatable, infrequently accessed data (once a month) with milliseconds access	General purpose or directory	1	30 days	128 KB	-	Per-GB fees apply
Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	General purpose	≥ 3	90 days	128 KB	-	Per-GB fees apply
Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	General purpose	≥ 3	90 days	-	-	Per-GB fees apply

Step 10: -Now click on upload it show success.

The screenshot shows the 'Upload succeeded' confirmation page in the AWS S3 console. It provides details about the upload status and a summary of the files uploaded.

Upload succeeded
For more information, see the [Files and folders](#) table.

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary

Destination	Succeeded	Failed
s3://aditya-24mca10024	1 file, 169.9 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 total, 169.9 KB)

Name	Folder	Type	Size	Status	Error
ADITYA.JPG	-	image/jpeg	169.9 KB	Succeeded	-

Step 11: -Now open the file and we can see everything as shown in fig.

The screenshot shows the AWS S3 console interface. The URL in the address bar is `ap-south-1.console.aws.amazon.com/s3/object/aditya-24mca10024?region=ap-south-1&bucketType=general&prefix=ADITYA.JPG`. The object name is `ADITYA.JPG`. The object overview section displays the following details:

- Owner:** 5678da62d1c79439ef3763fbe9afdc334a0b8031e10872b1e7be621f059c3718
- AWS Region:** Asia Pacific (Mumbai) ap-south-1
- Last modified:** May 22, 2025, 09:37:46 (UTC+05:30)
- Size:** 169.9 KB
- Type:** JPG
- Key:** ADITYA.JPG

On the right side, there are several links:

- S3 URI: s3://aditya-24mca10024/ADITYA.JPG
- Amazon Resource Name (ARN): arn:aws:s3:::aditya-24mca10024/ADITYA.JPG
- Entity tag (Etag): [3cd9be6b87d862effdf0b47e8d95cf53](#)
- Object URL: <https://aditya-24mca10024.s3.ap-south-1.amazonaws.com/ADITYA.JPG>

Step 12: -Since public access is not given so the url will not open to access this we move to permission in bucket.

The screenshot shows the AWS S3 console interface. The URL in the address bar is `ap-south-1.console.aws.amazon.com/s3/object/aditya-24mca10024?region=ap-south-1&bucketType=general&tab=permissions`. The object name is `ADITYA.JPG`. The permissions tab is selected. The access control list (ACL) section displays the following information:

This bucket has the **bucket owner enforced** setting applied for Object Ownership. When bucket owner enforced is applied, use bucket policies to control access. [Learn more](#)

Grantee	Object	Object ACL
Object owner (your AWS account) Canonical ID: 5678da62d1c79439ef3763fbe9afdc334a0b8031e10872b1e7be621f059c3718	Read	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	-	-
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-

Step 13: -Under block public access we will edit the access crieteria.

The screenshot shows the AWS S3 console with the URL <https://ap-south-1.console.aws.amazon.com/s3/bucket/aditya-24mca10024/property/bpa/edit?region=ap-south-1&bucketType=general>. The page title is "Edit Block public access (bucket settings)". The "Block all public access" checkbox is checked. A modal dialog box is open, containing the text: "Updating the Block Public Access settings for this bucket will affect this bucket and all objects within. This may result in some objects becoming public." Below this, it says "To confirm the settings, enter **confirm** in the field." A text input field contains the word "confirm". At the bottom right of the modal are "Cancel" and "Confirm" buttons. The status bar at the bottom of the browser window shows "CloudShell Feedback" and "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Step 14: -now click on confirm. Then also its not open because public policy is not mentioned.

The screenshot shows the AWS S3 console with the same URL and page title as the previous screenshot. The "Block all public access" checkbox is checked. A modal dialog box is open, containing the same confirmation message and "confirm" input field. The "Confirm" button is highlighted with a yellow background. The status bar at the bottom of the browser window shows "CloudShell Feedback" and "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Step 15: -Now writing the code for policy for accessing the file of bucket. And after the file is viewed by everyone.

The screenshot shows the AWS S3 Bucket Policy editor. The URL in the browser is <https://ap-south-1.console.aws.amazon.com/s3/bucket/aditya-24mca10024/property/policy/edit?region=ap-south-1&bucketType=general>. The page title is "Edit bucket policy - S3 bucket".

Bucket ARN: arn:aws:s3:::aditya-24mca10024

Policy:

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "Statement1",  
6             "Principal": "*",  
7             "Effect": "Allow",  
8             "Action": "S3:GetObject",  
9             "Resource": "arn:aws:s3:::aditya-24mca10024/*"  
10        }  
11    ]  
12 }
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 16: -Now in this step we will understand the use of version id in this if upload same file format with same name then it is differentiated using version id as shown

The screenshot shows the AWS S3 console interface. The top navigation bar includes links for Gmail, YouTube, ChatGPT, GitHub, and WhatsApp. The main page displays the bucket 'aditya-24mca10024' with two objects listed:

Name	Type	Last modified	Size	Storage class
ADITYA.JPG	JPG	May 22, 2025, 09:37:46 (UTC+05:30)	169.9 KB	Standard
IMG_0265.JPG	JPG	May 22, 2025, 09:44:32 (UTC+05:30)	840.9 KB	Standard

Below the table, there is a search bar labeled 'Find objects by prefix' and a set of actions buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

Step 17: -Here we will see about life cycle rule of bucket by giving time period .

The screenshot shows the 'Create lifecycle rule' page for the 'aditya-24mca10024' bucket. The 'Actions' section contains several checkboxes:

- Transition current versions of objects between storage classes
This action will move current versions.
- Transition noncurrent versions of objects between storage classes
This action will move noncurrent versions.
- Expire current versions of objects
- Permanently delete noncurrent versions of objects
- Delete expired object delete markers or incomplete multipart uploads
These actions are not supported when filtering by object tags or object size.

The 'Permanently delete noncurrent versions of objects' section includes a dropdown for 'Days after objects become noncurrent' (set to 30) and a field for 'Number of newer versions to retain - Optional'.

The 'Review transition and expiration actions' section shows the following details:

Current version actions	Noncurrent versions actions
Day 0 No actions defined.	Day 0 • Objects become noncurrent

Step 18: -Now in this we will see how to delete the object in bucket. Clicking on the object which we have to delete.

The screenshot shows the AWS S3 console interface. The top navigation bar includes links for Gmail, YouTube, ChatGPT, GitHub, and WhatsApp. The main navigation bar shows 'Amazon S3 > Buckets > aditya-24mca10024'. Below this, the 'aditya-24mca10024' bucket page is displayed with the 'Info' tab selected. The 'Objects' tab is active, showing two items: 'ADITYA.JPG' and 'IMG_0265.JPG'. Both files are of type 'JPG' and were last modified on May 22, 2025. The storage class for both is 'Standard'. Action buttons include Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

Step 19: -Now click on delete option and type delete to delete the object and empty the bucket.

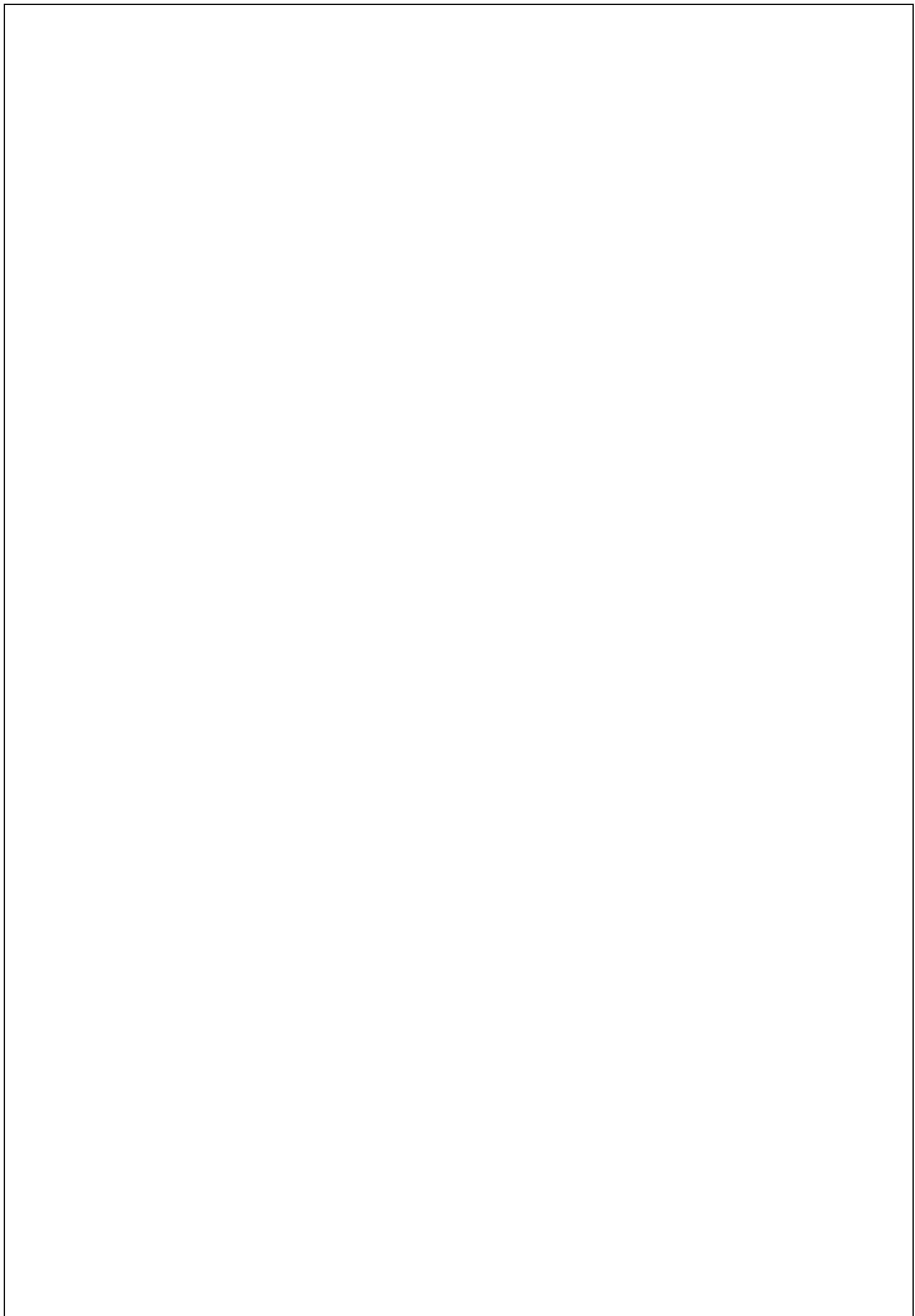
The screenshot shows the AWS S3 console after attempting to delete objects. The top navigation bar and main navigation bar are identical to the previous screenshot. A green success message box at the top left states 'Successfully deleted objects' and 'View details below.' Below this, a note says 'After you navigate away from this page, the following information is no longer available.' The 'Summary' section shows 'Source s3://aditya-24mca10024' and 'Successfully deleted 2 objects, 1010.8 KB'. The 'Failed to delete' section shows '0 objects'. The 'Failed to delete' tab is selected, showing a table with a single row: 'Find objects by name' (empty), 'Name' (empty), 'Type' (empty), 'Last modified' (empty), 'Size' (empty), and 'Error' (empty). A note at the bottom of this section says 'No objects failed to delete.'

Step 20: -Now in this we will delete the bucket which we have created .For this the most important part is the buget must be empty .

The screenshot shows the 'Delete bucket' confirmation page for an AWS S3 bucket named 'aditya-24mca10024'. The browser address bar shows the URL: ap-south-1.console.aws.amazon.com/s3/bucket/aditya-24mca10024/delete?region=ap-south-1&bucketType=general. The page header includes the AWS logo and navigation links for 'Amazon S3', 'Buckets', and 'aditya-24mca10024'. A warning box at the top lists several points about deleting buckets. Below it, a section titled 'Delete bucket "aditya-24mca10024"' asks for confirmation by entering the bucket name into a text input field, which contains 'aditya-24mca10024'. At the bottom right are 'Cancel' and 'Delete bucket' buttons.

Step 21: - After selecting delete option click on delete option and type bucket name to delete then click on delete the bucket will successfully be deleted.

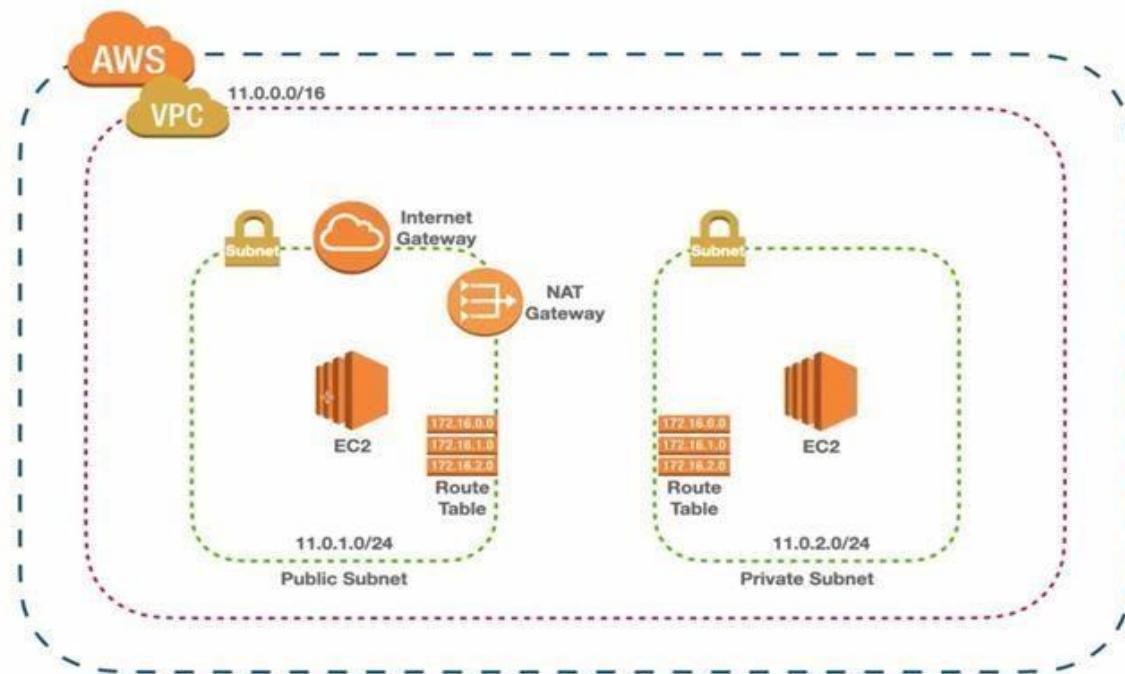
This screenshot is identical to the one above, showing the 'Delete bucket' confirmation page for the same bucket. The URL, header, warning box, and overall layout are the same, indicating a successful deletion process.



Date: 23/05/2025	Title
Exp. No: 05	HOW TO SETUP EC2 INSTANCE IN VPC

HOW TO SETUP EC2 INSTANCE IN VPC

AWS VPC (Amazon Virtual Private Cloud): It is a service that allows you to create a private network in the cloud. It gives you full access over your networking environment allowing you to define your IP address range also to create subnets and manage routing tables. With VPC you can securely connect your resources to the internet or keep them isolated from external traffic depending on your needs. It's like building your own private data center within AWS.



Internet Gateway: An Internet Gateway works by establishing a connection between a VPC and the internet. The VPC must have a public subnet, and the instances within that subnet must have a public IP address to communicate with the internet. An Internet Gateway acts as a bridge between the VPC and the internet. An Internet Gateway is commonly used when you want resources within a VPC to be accessible from the internet.

NAT Gateway: A NAT Gateway enables instances in a private subnet to connect to the internet or other AWS services but prevents the internet or other AWS services from initiating a connection with those instances. A NAT Gateway is commonly used when you have resources within a private subnet that require outbound internet access but should not be directly accessible from the internet.

Subnet: A subnet is like a smaller group within a large network. It is a way to split a large network into smaller networks so that devices present in one network can transmit data more easily.

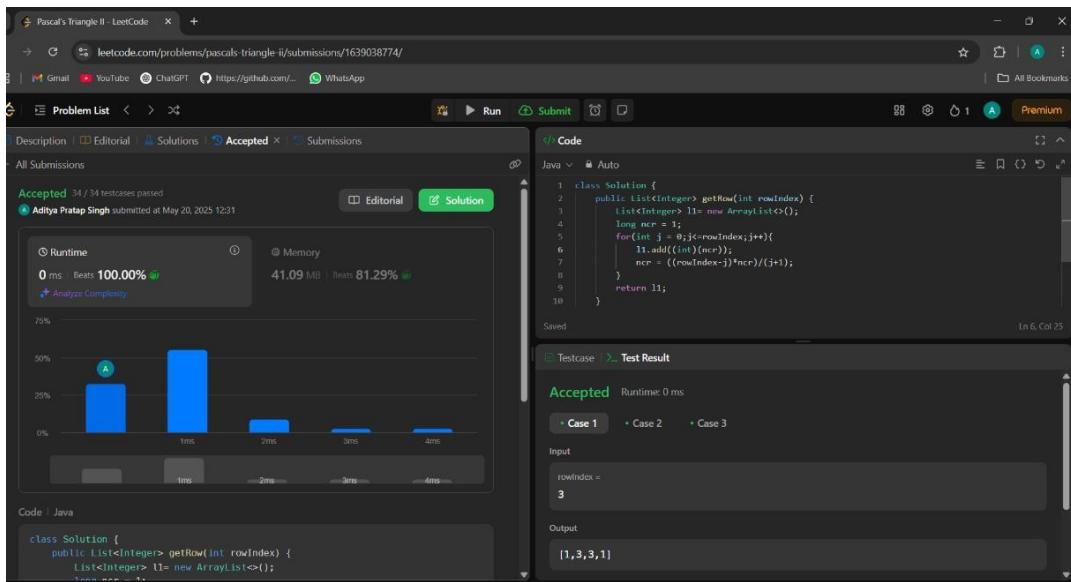
- The **public subnet** has instances with public IP addresses allowing them to

communicate directly with the internet. These instances can be web servers load balancers or other publicly accessible services. An Internet Gateway connects the public subnet to the internet providing inbound and outbound communication.

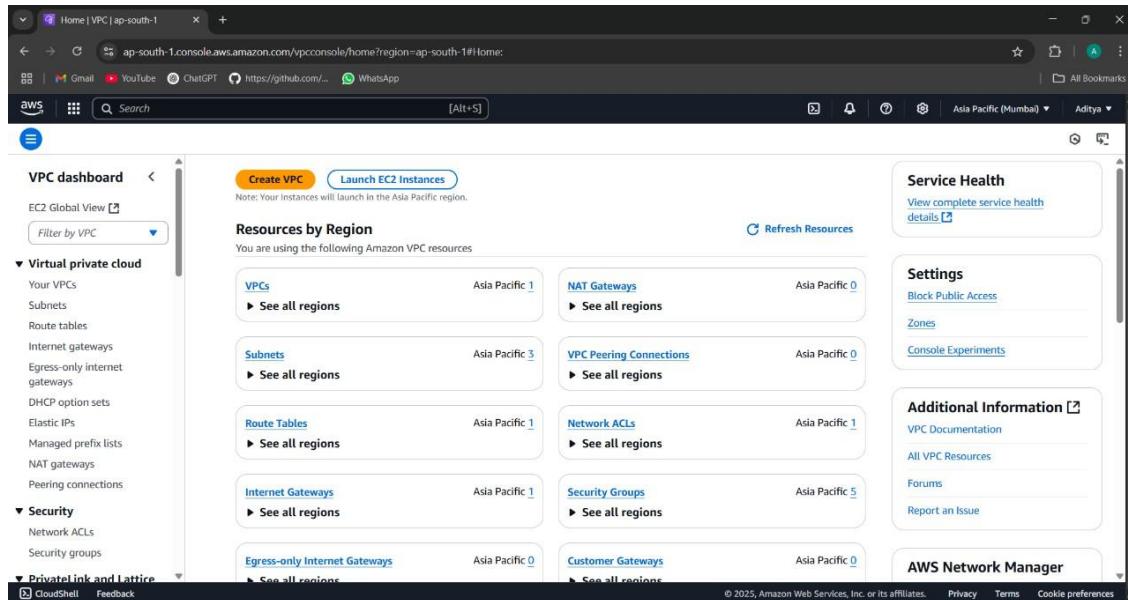
- The **private subnet** has instances with private IP addresses. These instances cannot directly access the internet but can connect to resources within the same VPC or other on-premises networks via a VPN or Direct Connect. Typically the instances in the private subnet access the internet via a NAT gateway or NAT instance which is usually placed in the public subnet.
- Public subnets have a route to the internet via an Internet Gateway. Resources in public subnets can be accessed from the internet such as web servers whereas Private subnets do not have a direct route to the internet it uses NAT device to get connected with Internet. Resources within private subnets cannot be accessed from outside the VPC unless specific configurations are made.

Route table: A route table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed. It is used to direct network traffic on a particular destination IP address. It allows customization of networks enabling them to connect to other subnets/gateways both within and outside of the VPC.

STEP1: LOGIN AS A ROOT USER AND SEARCH FOR VPC IN THE CONSOLE



STEP2: CLICK ON CREATE VPC



STEP3: SELECT VPC ONLY> TYPE THE VPC NAME> IPv4CIDR-11.0.0.0/16

NOTE: In AWS a VPC spans a specific IP address range using CIDR (Classless Inter-Domain Routing) blocks. The CIDR block defines the range of IP addresses that can be assigned to resources within the VPC

The screenshot shows the 'Create VPC' page in the AWS VPC console. Under 'VPC settings', the 'Resources to create' dropdown is set to 'VPC only'. A 'Name tag - optional' field contains 'aditya-24mca10024'. The 'IPv4 CIDR block' dropdown is set to 'IPv4 CIDR manual input' and shows '11.0.0.0/16'. Below it, a note says 'CIDR block size must be between /16 and /28.' The 'IPv6 CIDR block' dropdown is set to 'No IPv6 CIDR block'. The 'Tenancy' dropdown is set to 'Default'.

For better understanding of the CIDR range, Search CIDR Range Calculator in google and enter the range. Here its showing 65536 IP range, i.e. these are the maximum number of IP addresses which is accessible by the VPC. If we increase from /16 to /32, it will decrease the number of IP address and if we decrease it to /8, it will increase the number of IP addresses. (/16 means first 16bits are locked)

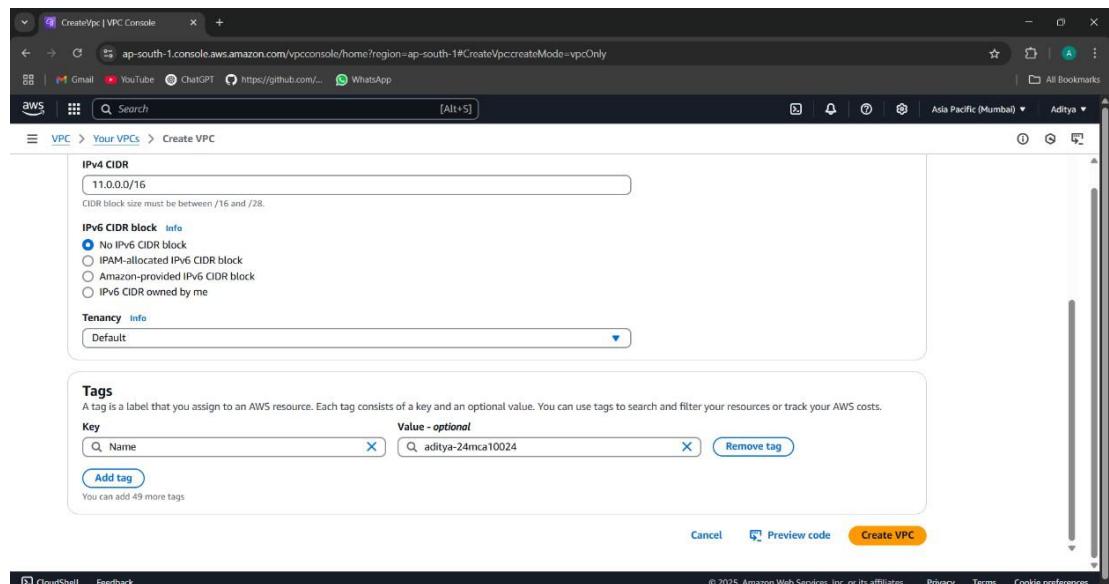
The screenshot shows the MXToolbox Subnet Calculator. The input CIDR is '11.0.0.0/16'. The results table shows:

Input	Input IP	Input Long	Input Hex
11.0.0.0/16	11.0.0.0	184549376	0B.00.00.00
CIDR	CIDR IP Range	CIDR Long Range	CIDR Hex Range
11.0.0.0/16	11.0.0.0 - 11.0.255.255	184549376 - 184614911	0B.00.00.00 - 0B.00.FF.FF

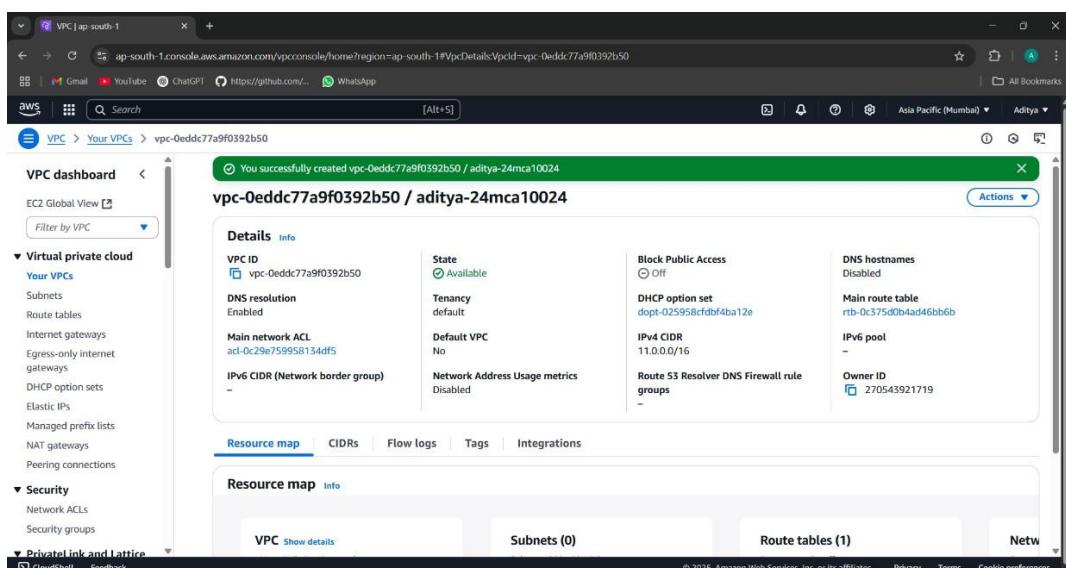
Below the table, it says 'IPs in Range 65,536', 'Mask Bits 16', 'Subnet Mask 255.255.0.0', and 'Hex Subnet Mask FF.FF.00.00'.

ABOUT SUBNET CALCULATOR

STEP4: CLICK ON CREATE VPC



STEP5: VPC CREATED



STEP6: CLICK ON SUBNET>CREATE SUBNET

The screenshot shows the AWS VPC Subnets console. On the left, there's a navigation sidebar with sections like VPC dashboard, Virtual private cloud, Security, and PrivateLink and Lattice. The main area is titled "Subnets (3) Info" and lists three subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-05fc0c70e6893ce	Available	vpc-0f4d0926b65ff0381	Off	172.31.0.0/16
-	subnet-037deac9c5d2d6cb	Available	vpc-0f4d0926b65ff0381	Off	172.31.16.0/16
-	subnet-0a0130ca69d76c491	Available	vpc-0f4d0926b65ff0381	Off	172.31.32.0/16

A prominent orange button at the top right says "Create subnet". Below the table, there's a section labeled "Select a subnet".

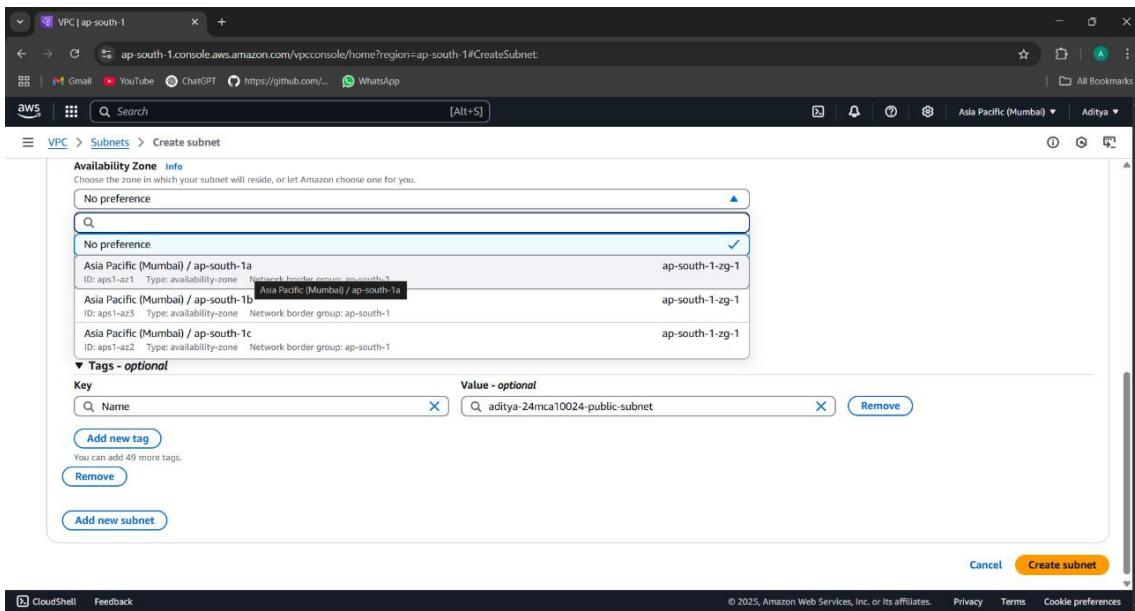
STEP7: SELECT THE VPC THAT YOU HAVE CREATED EARLIER

The screenshot shows the "Create subnet" dialog box. At the top, it says "VPC ID" and "Create subnets in this VPC." Below that is a dropdown menu labeled "Select a VPC". The dropdown shows two options:

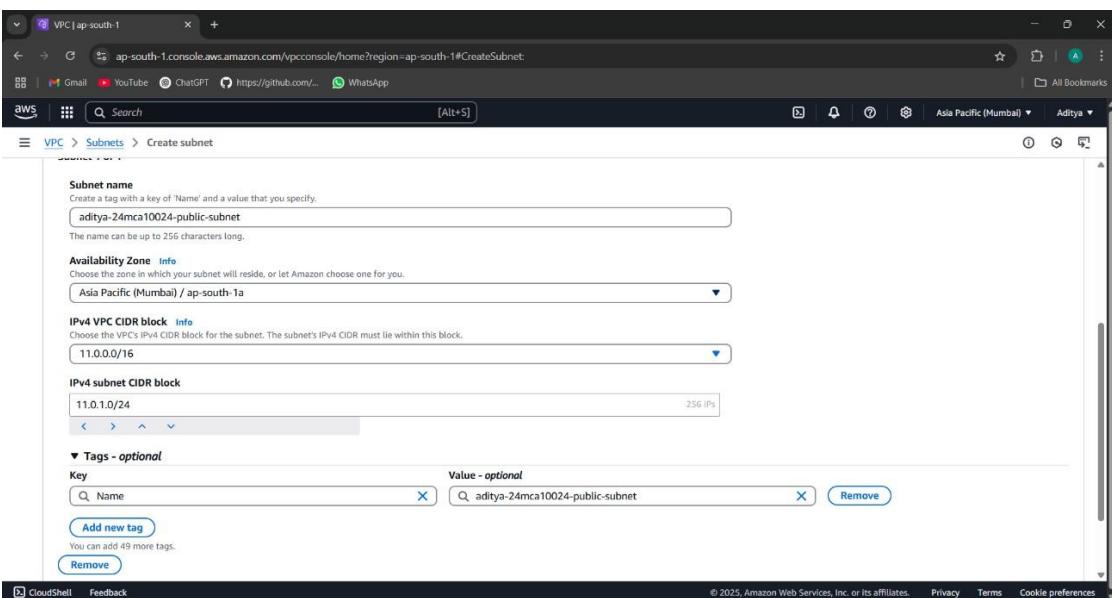
- vpc-0f4d0926b65ff0381 (default)
- vpc-0eddc7a9f0392b50 (aditya-24mca10024)

At the bottom of the dropdown, there's a link "Select a VPC first to create new" followed by "vpc-0eddc7a9f0392b50 (aditya-24mca10024)". There are "Add new subnet" and "Create subnet" buttons at the bottom right.

STEP8: SELECT ap-south-1a in the Availability Zone



STEP9: ENTER IPv4 SUBNET CIDR BLOCK-11.0.1.0/24> CLICK ON ADD NEW SUBNET



STEP9: ENTER THE NAME OF THE PRIVATE SUBNET> AVAILABILITY ZONE:ap-south-1b> IPv4 subnet CIDR Block-11.0.2.0/24> CLICK ON CREATE SUBNET

The screenshot shows the 'Create subnet' wizard in the AWS VPC console. The configuration is as follows:

- Name:** The name can be up to 255 characters long.
- Availability Zone:** Info - Choose the zone in which your subnet will reside, or let Amazon choose one for you. Set to Asia Pacific (Mumbai) / ap-south-1a.
- IPv4 VPC CIDR block:** Info - Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block. Set to 10.0.0.0/16.
- IPv4 subnet CIDR block:** Set to 11.0.1.0/24.
- Tags - optional:** No tags associated with the resource.
- Add new tag:** You can add 50 more tags.
- Remove:** Remove existing tags.
- Add new subnet:** A button to start the process.
- Create subnet:** The final button to submit the form.

STEP10: BOTH THE PUBLIC AND PRIVATE SUBNET IS CREATED

The screenshot shows the 'Subnets' dashboard in the AWS VPC console. The table displays the following information for the newly created subnet:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-097420788f8501a26	Available	vpc-0595ffe723c6ddf4d aditya...	Off	10.0.1.0/24

STEP11: NOW CREATE AN INTERNET GATEWAY FOR PUBLIC SUBNET. CLICK ON INTERNET GATEWAY FROM THE DASHBOARD> CLICK ON CREATE INTERNET GATEWAY

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A table lists one internet gateway:

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-039c0fc8c673765f	Attached	vpc-0f4d0926b65ff0381	270543921719

Below the table, a message says "Select an internet gateway above". The top right corner features a yellow button labeled "Create internet gateway".

STEP12: TYPE THE NAME OF THE GATEWAY

The screenshot shows the "Create internet gateway" wizard. The first step, "Internet gateway settings", is displayed. It includes a "Name tag" field containing "aditya-24mca10024-internet-gateway" and a "Tags - optional" section with a single tag "aditya-24mca10024-internet-gateway". At the bottom right are "Cancel" and "Create internet gateway" buttons.

STEP13: INTERNET GATEWAY IS CREATED

The screenshot shows the AWS VPC console in the 'ap-south-1' region. The left sidebar has 'Virtual private cloud' expanded, with 'Internet gateways' selected. The main content area displays the details of a newly created Internet Gateway, identified by the ID 'igw-0da9dca3a30ba4e7b'. The gateway is currently 'Detached'. A green banner at the top right says, 'The following internet gateway was created: igw-0da9dca3a30ba4e7b - aditya-24mca10024-internet-gateway. You can now attach to a VPC to enable the VPC to communicate with the internet.' Below the gateway details, there is a 'Tags' section with one tag named 'Name' with the value 'aditya-24mca10024-internet-gateway'. At the bottom right of the main content area, there is a link to 'Attach to a VPC'.

STEP14: CLICK ON ATTACH TO VPC> SELECT THE VPC CREATED EARLIER> CLICK ON ATTACH INTERNET GATEWAY

The screenshot shows the 'Attach to VPC' dialog box. The title bar says 'Attach to VPC (igw-0da9dca3a30ba4e7b)'. The main area is titled 'VPC' and contains the instruction: 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' Below this is a 'Available VPCs' section with the heading 'Attach the internet gateway to this VPC.' A search bar contains the text 'vpc-0eddc77a9f0392b5q'. At the bottom right of the dialog box is a large orange button labeled 'Attach internet gateway'.

STEP15: INTERNET GATEWAY IS ATTACHED TO THE VPC

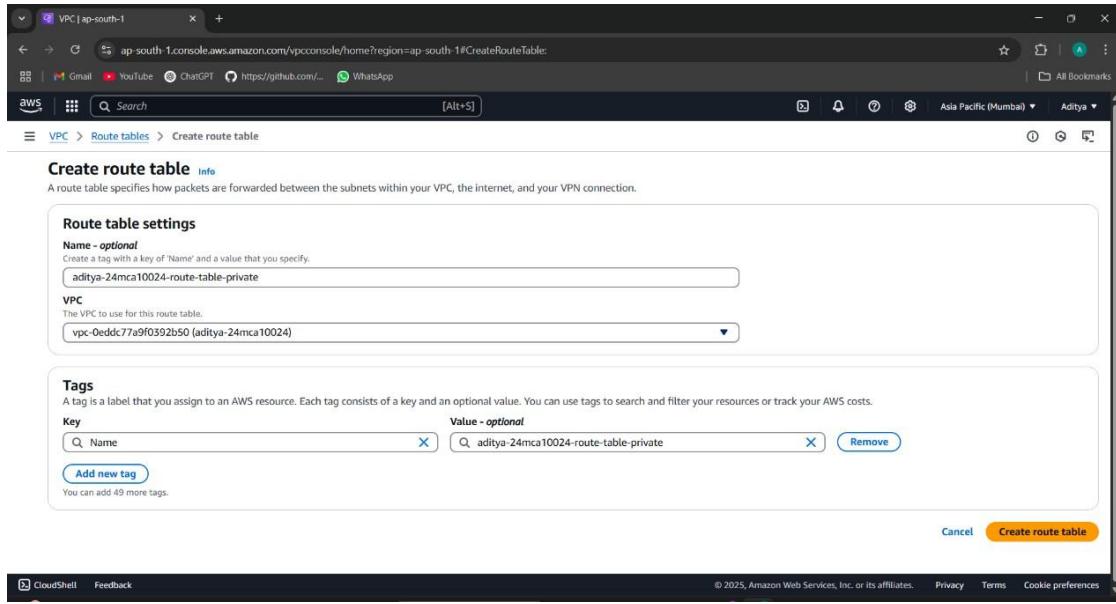
The screenshot shows the AWS VPC console with the URL ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#InternetGateway:id=igw-0da9dca3a30ba4e7b. A green notification bar at the top states "Internet gateway igw-0da9dca3a30ba4e7b successfully attached to vpc-0eddcc77a9f0392b50". The main view displays the details of the internet gateway "igw-0da9dca3a30ba4e7b / aditya-24mca10024-internet-gateway". The "Details" section shows the Internet gateway ID (igw-0da9dca3a30ba4e7b), State (Attached), VPC ID (vpc-0eddcc77a9f0392b50 | aditya-24mca10024), and Owner (270543921719). The "Tags" section lists a single tag: Name = aditya-24mca10024-internet-gateway.

STEP16: NEXT CREATE TWO ROUTE TABLES (PRIVATE AND PUBLIC) FOR THE SUBNETS.

STEP17: TO CREATE A PUBLIC ROUTE TABLE: CLICK ON ROUTE TABLES>CREATE A ROUTE TABLE> WRITE A SUITABLE NAME FOR THE ROUTE TABLE>CHOOSE THE VPC>CLICK ON CREATE ROUTE TABLE

The screenshot shows the AWS VPC console with the URL ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateRouteTable. The "Create route table" dialog is open. In the "Route table settings" section, the "Name" field is set to "aditya-24mca10024-route-table-public" and the "VPC" dropdown is set to "vpc-0eddcc77a9f0392b50 (aditya-24mca10024)". In the "Tags" section, there is one tag: "Name" = "aditya-24mca10024-route-table-public". At the bottom right, there are "Cancel" and "Create route table" buttons.

STEP18: TO CREATE A PRIVATE ROUTE TABLE: CLICK ON ROUTE TABLES>CREATE A ROUTE TABLE> WRITE A SUITABLE NAME FOR THE ROUTE TABLE>CHOOSE THE VPC>CLICK ON CREATE ROUTE TABLE



STEP19: BOTH THE PUBLIC AND PRIVATE ROUTE TABLES ARE CREATED AS SHOWN BELOW. NOW WE NEED TO CONNECT BOTH THE ROUTE TABLES WITH THEIR CORRESPONDING SUBNETS.

Route tables (4) <small>Info</small>						
<input type="checkbox"/>	Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
<input type="checkbox"/>	-	rtb-0f82bc4e43401306d	-	-	Yes	vpc-0f14d0926b65ff0381
<input type="checkbox"/>	-	rtb-0c375d0b4add46bb6b	-	-	Yes	vpc-0eddc77a9f0392b50 adity...
<input type="checkbox"/>	aditya-24mca10024-route-table-public	rtb-04a5620ca0ff086017	-	-	No	vpc-0eddc77a9f0392b50 adity...
<input type="checkbox"/>	aditya-24mca10024-route-table-private	rtb-0961193470068902a	-	-	No	vpc-0eddc77a9f0392b50 adity...

Select a route table

STEP20: CLICK ON THE PUBLIC ROUTE TABLE>CLICK ON SUBNET ASSOCIATIONS

The screenshot shows the AWS VPC console interface. The URL is ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#RouteTableDetails:RouteTableId=rtb-04a9629ca8fb86017. The page displays details for a route table with ID **rtb-04a9629ca8fb86017**, which is associated with the VPC **vpc-0eddcc77a9f0392b50** and owner **aditya-24mca10024**. The **Subnet associations** tab is selected. Under the **Explicit subnet associations** section, there are no entries. The **Subnets without explicit associations** section shows two subnets: **aditya-24mca10024-private-subnet** and **aditya-24mca10024-public-subnet**.

STEP 21: CLICK ON EDIT SUBNET ASSOCIATIONS > CHOOSE PUBLIC SUBNET FROM THE OPTIONS> CLICK ON SAVE ASSOCIATIONS

The screenshot shows the **Edit subnet associations** dialog box. It lists available subnets under the heading **Available subnets (1/2)**. There are two subnets: **aditya-24mca10024-private-subnet** and **aditya-24mca10024-public-subnet**. The **Selected subnets** section contains the entry **subnet-094ea7dc6fab2b58a / aditya-24mca10024-public-subnet**. At the bottom right, there are **Cancel** and **Save associations** buttons.

STEP 21: SUBNET ASSOCIATION IS DONE. REPEAT THE SAME STEP FOR PRIVATE SUBNET ASSOCIATION.

The screenshot shows the AWS VPC console with the URL ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-04a9629ca8fb86017. The page title is "Edit subnet associations". It displays a table of available subnets and a list of selected subnets. The selected subnet is "subnet-094ea7dc6fab2b58a / aditya-24mca10024-public-subnet". At the bottom right are "Cancel" and "Save associations" buttons.

Available subnets (1/2)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
aditya-24mca10024-private-subnet	subnet-072eb56414424c949	11.0.2.0/24	-	Main (rtb-0c375d0b4ad46bb6b)	
<input checked="" type="checkbox"/> aditya-24mca10024-public-subnet	subnet-094ea7dc6fab2b58a	11.0.1.0/24	-	Main (rtb-0c375d0b4ad46bb6b)	

Selected subnets
subnet-094ea7dc6fab2b58a / aditya-24mca10024-public-subnet

Cancel Save associations

STEP22: NEXT WE NEED TO CREATE THE ROUTE SO THAT INTERNET CAN BE ACCESSED WITH THE HELP OF INTERNET GATEWAY THROUGH THESE ROUTE TABLES.

STEP23: GO TO PUBLIC ROUTE TABLE FROM THE AWS CONSOLE> CLICK ON ROUTES>CLICK ON EDIT ROUTES

The screenshot shows the AWS VPC console with the URL ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#RouteTableDetails:RouteTableId=rtb-09b11934700b8902a. A green success message says "You have successfully updated subnet associations for rtb-09b11934700b8902a / aditya-24mca10024-route-table-private.". The page title is "rtb-09b11934700b8902a / aditya-24mca10024-route-table-private". It shows the "Details" tab with route table ID "rtb-09b11934700b8902a", VPC "vpc-0eddcc77a9f0392b50 | aditya-24mca10024", and explicit subnet associations for "subnet-072eb56414424c949 / aditya-24mca10024-private-subnet". The "Routes" tab shows one route: Destination "11.0.0.0/16" Target "local" Status "Active" Propagated "No".

You have successfully updated subnet associations for rtb-09b11934700b8902a / aditya-24mca10024-route-table-private.

rtb-09b11934700b8902a / aditya-24mca10024-route-table-private

Details

Route table ID: rtb-09b11934700b8902a
Main: No
Owner ID: 270543921719

Explicit subnet associations: subnet-072eb56414424c949 / aditya-24mca10024-private-subnet

Edge associations: -

Routes

Destination	Target	Status	Propagated
11.0.0.0/16	local	Active	No

Both Edit routes

**STEP24: CLICK ON ADD ROUTE>SELECT THE IP-0.0.0.0/0>CHOOSE INTERNET GATEWAY>
SELECT THE INTERNET GATEWAY THAT YOU HAVE CREATED> CLICK ON SAVE CHANGES
(NOTE:0.0.0.0/0 means it allows all the IP addresses to access the resources present in the
public subnet)**

Destination	Target	Status	Propagated
11.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-0da9dca3a30ba4e7b		

Add route Cancel Preview Save changes

Updated routes for rtb-04a9629ca8fb86017 / aditya-24mca10024-route-table-public successfully

Details	Info	Main	Explicit subnet associations	Edge associations
Route table ID	rtb-04a9629ca8fb86017	No	subnet-094ea7dc6fab2b58a / aditya-24mca10024-public-subnet	-
VPC	vpc-0eddcc77a9f0392b50 aditya-24mca10024	Owner ID	270543921719	

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

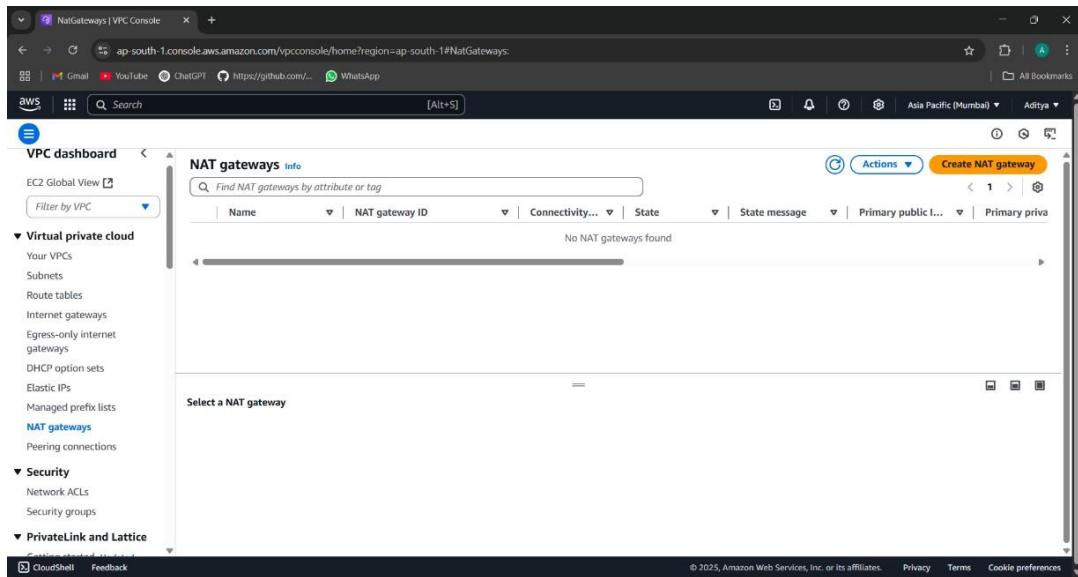
Destination	Target	Status	Propagated
0.0.0.0/0	igw-0da9dca3a30ba4e7b	Active	No
11.0.0.0/16	local	Active	No

**STEP25: WE NEED TO CREATE THE NAT GATEWAY FOR THE PRIVATE SUBNET SO THAT
RESOURCES PRESENT INSIDE THE PRIVATE SUBNET CAN ACCESS THE INTERNET WITH THE
HELP OF INTERNET GATEWAY.**

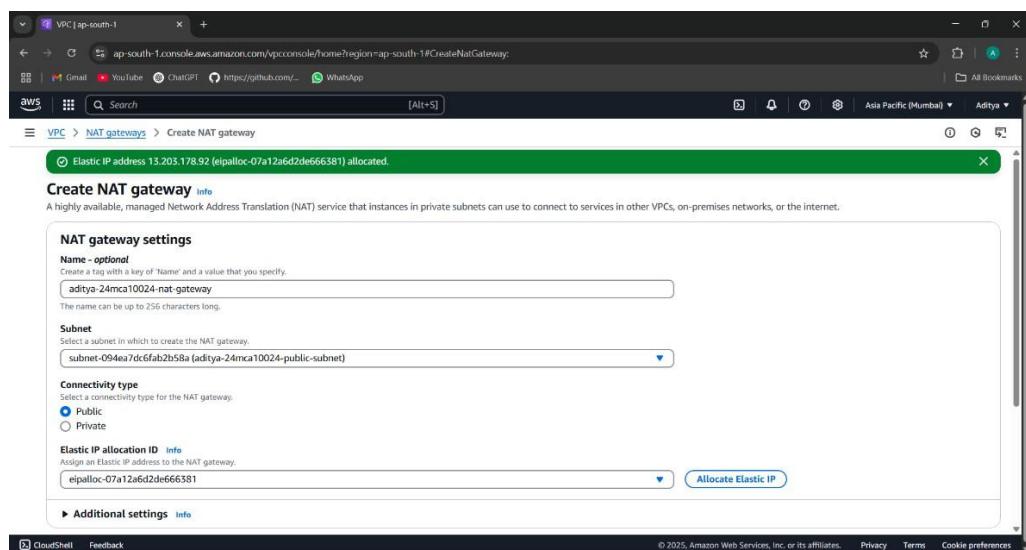
NOTE:

1. Internet Gateway is two-way process i.e., outside world can access the resources inside the public subnet as well as resources inside it can request for accessing outside.
2. NAT GATEWAY is not free and its created only in public subnet.
3. Private subnet will be able to access the NAT gateway with the help of its route table to access resources inside the public subnet.

STEP26: CLICK ON NAT GATEWAYS FROM THE DASHBOARD> CLICK ON CREATE NAT GATEWAY.



STEP27: GIVE A NAME TO THE NAT GATEWAY>CHOOSE THE PUBLIC SUBNET>ALLOCATE ELASTIC IP> CLICK ON CREATE NAT GATEWAY.



The screenshot shows the AWS VPC dashboard with the 'NAT gateways' section selected. A success message at the top states: 'NAT gateway nat-02b290595db249beb | aditya-24mca10024-nat-gateway was created successfully.' Below this, the 'nat-02b290595db249beb / aditya-24mca10024-nat-gateway' details are displayed. The 'Details' section includes:

- NAT gateway ID: nat-02b290595db249beb
- Connectivity type: Public
- Primary public IPv4 address: -
- Primary private IPv4 address: -
- State: Pending
- Created: Wednesday, May 21, 2025 at 09:25:58 GMT+5:30
- Deleted: -
- State message: Info
- Primary network interface ID: -

The 'Secondary IPv4 addresses' tab is selected, showing a table with columns: Private IPv4 address, Network interface ID, Status, and Failure message. A note at the bottom says: 'Secondary IPv4 addresses are not available for this nat gateway.'

STEP28: NOW UPDATE THE PRIVATE ROUTE TABLE: CLICK ON THE ROUTE TABLES>CLICK ON THE PRIVATE ROUTE TABLE>CLICK ON ROUTES>CLICK ON EDIT ROUTES

This screenshot is identical to the one above, showing the AWS VPC dashboard with the 'NAT gateways' section selected. A success message at the top states: 'NAT gateway nat-02b290595db249beb | aditya-24mca10024-nat-gateway was created successfully.' Below this, the 'nat-02b290595db249beb / aditya-24mca10024-nat-gateway' details are displayed. The 'Details' section includes:

- NAT gateway ID: nat-02b290595db249beb
- Connectivity type: Public
- Primary public IPv4 address: -
- Primary private IPv4 address: -
- State: Pending
- Created: Wednesday, May 21, 2025 at 09:25:58 GMT+5:30
- Deleted: -
- State message: Info
- Primary network interface ID: -

The 'Secondary IPv4 addresses' tab is selected, showing a table with columns: Private IPv4 address, Network interface ID, Status, and Failure message. A note at the bottom says: 'Secondary IPv4 addresses are not available for this nat gateway.'

STEP29: CLICK ON ADD ROUTE>SELECT IP ADDRESS

:0.0.0.0/0>SELECT NAT GATEWAY>CHOOSE THE NAT GATEWAY THAT YOU HAVE CREATED>
CLICK ON SAVE CHANGES.

The screenshot shows the 'Edit routes' interface for a specific route table. A new route is being added with the following details:

Destination	Target	Status	Propagated
11.0.0.0/16	local	Active	No
Q. 0.0.0.0/0	NAT Gateway	-	No
	Q. nat-02b290595db249beb	-	

At the bottom right, there are 'Cancel', 'Preview', and 'Save changes' buttons. The 'Save changes' button is highlighted in orange.

STEP30: PRIVATE SUBNET IS UPDATED WITH NAT GATEWAY

The screenshot shows the 'Route tables' page for a specific route table. A green success message at the top states: 'Updated routes for rtb-09b11934700b8902a / aditya-24mca10024-route-table-private successfully'. Below the message, the route table details are shown:

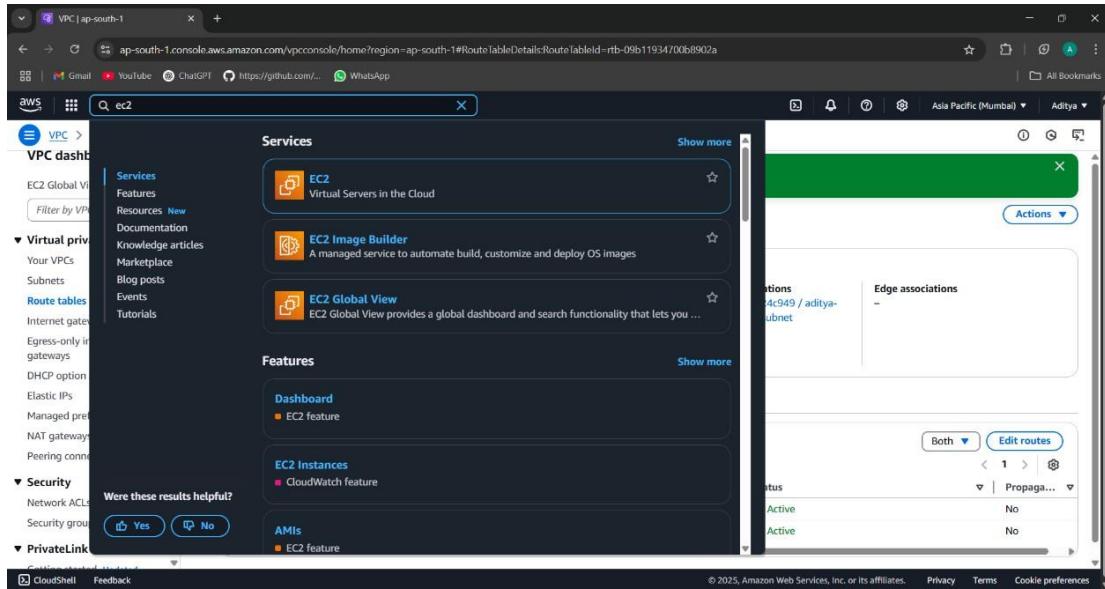
Route table ID	Main	Explicit subnet associations	Edge associations
rtb-09b11934700b8902a	No	subnet-072eb56414424c049 / aditya-24mca10024-private-subnet	-
VPC	Owner ID		
vpc-0eddcc77a9f0392b50 aditya-24mca10024	270543921719		

The 'Routes' tab is selected, displaying two routes:

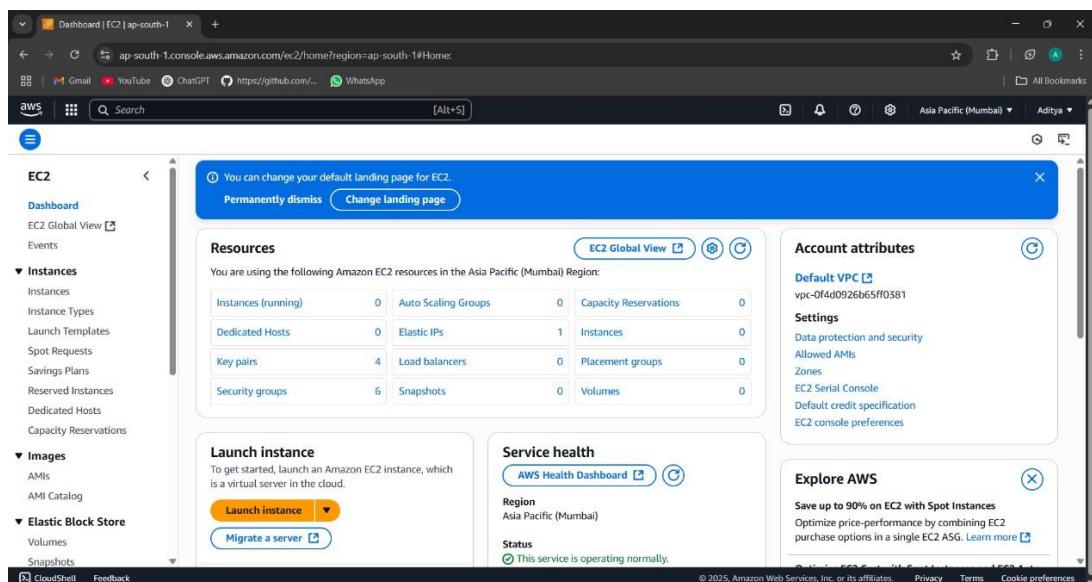
Destination	Target	Status	Propagation
0.0.0.0/0	nat-02b290595db249beb	Active	No
11.0.0.0/16	local	Active	No

At the bottom right, there are 'Both', 'Edit routes', and other navigation buttons.

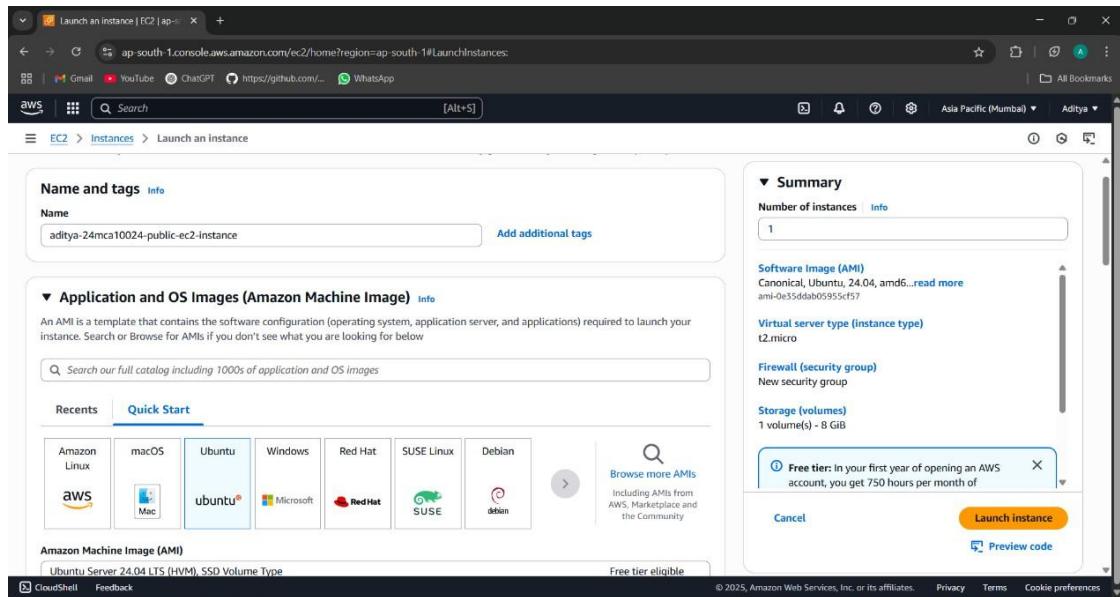
STEP31: WHOLE NETWORK SET UP IS READY, NOW WE HAVE TO CREATE EC2 INSTANCES FOR BOTH PUBLIC AND PRIVATE SUBNET. SEARCH FOR EC2>CLICK ON EC2



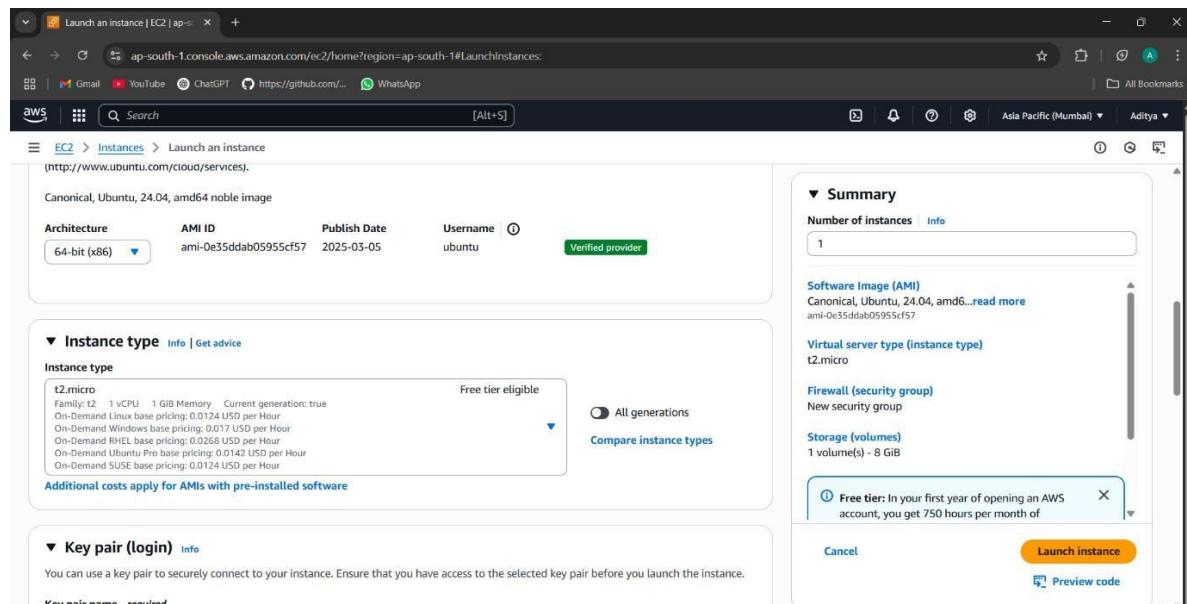
STEP 32: CLICK ON LAUNCH INSTANCE



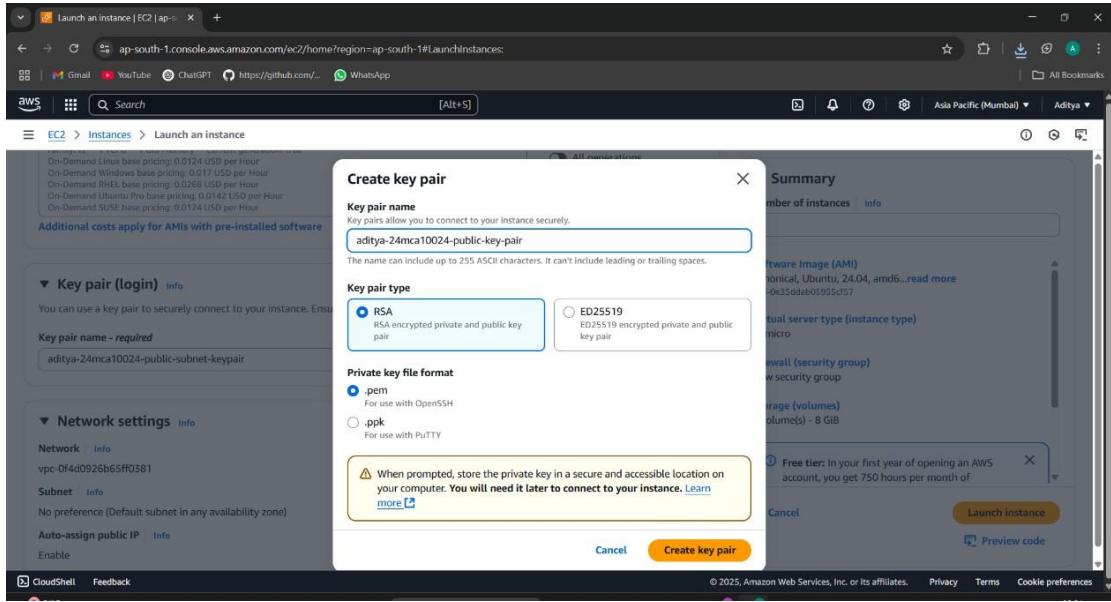
STEP 33: GIVE A SUITABLE INSTANCE NAME>CHOOSE OS:UBUNTU



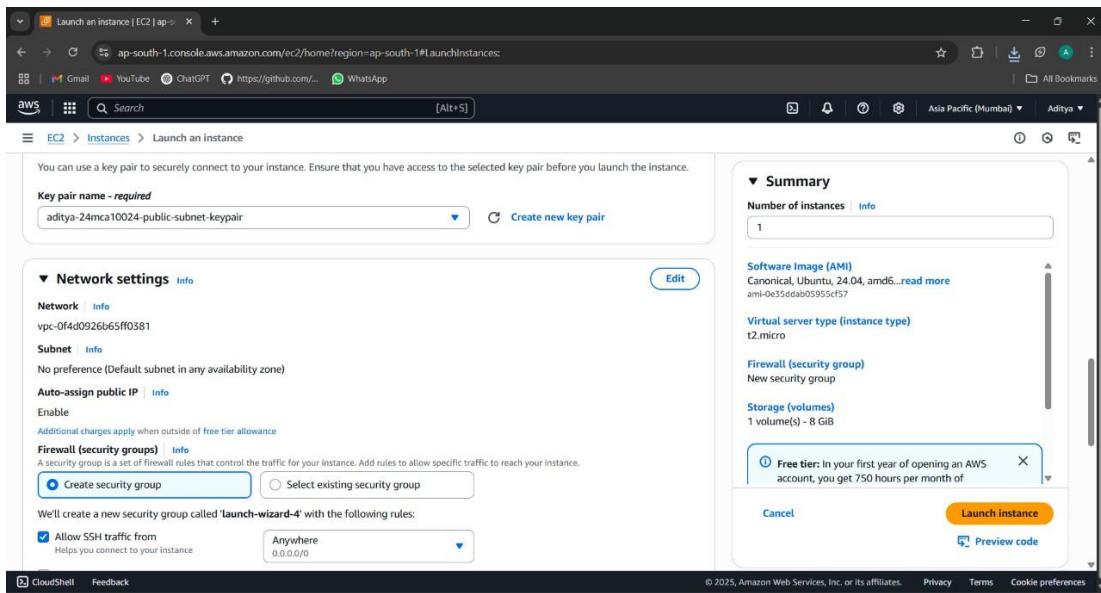
STEP 34: CHOOSE THE DEFAULT ARCHITECTURE>FREE TIER INSTANCE TYPE



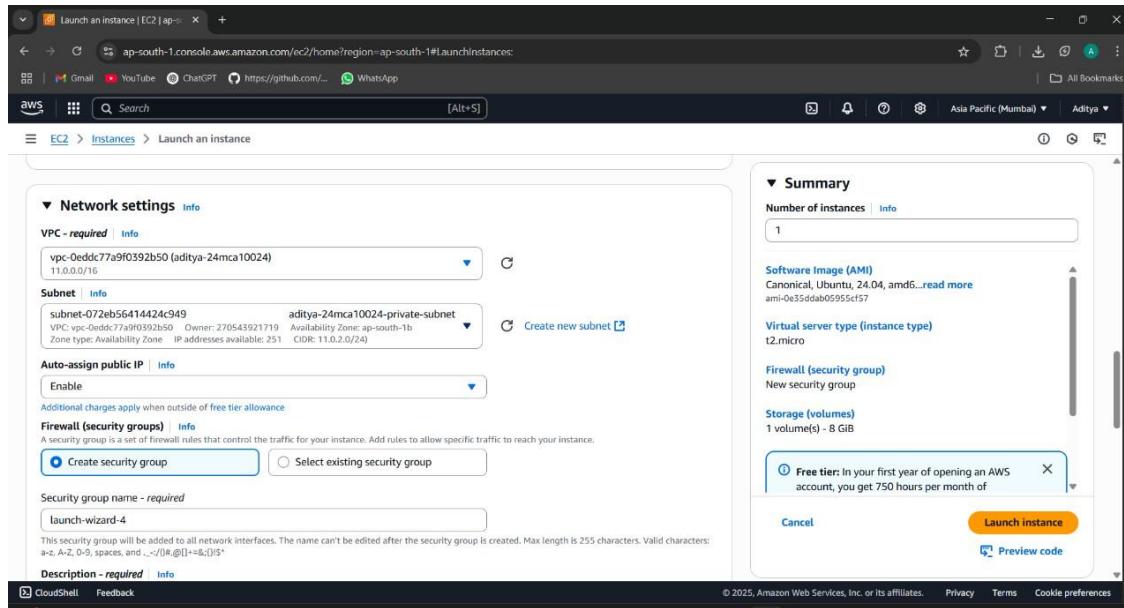
STEP 35: CREATE KEY PAIR



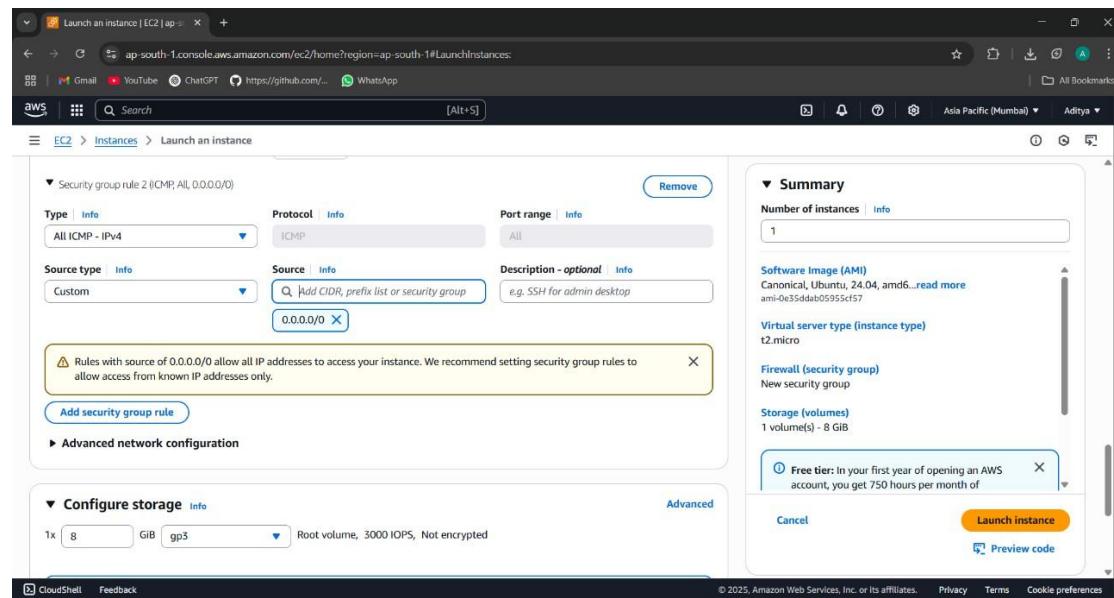
STEP 36: EDIT THE NETWORK SETTING>CHOOSE THE VPC CREATED EARLIER>SELECT THE PUBLIC SUBNET> ENABLE THE AUTO-ASSIGN PUBLIC IP>CHOOSE CREATE SECURITY GROUP



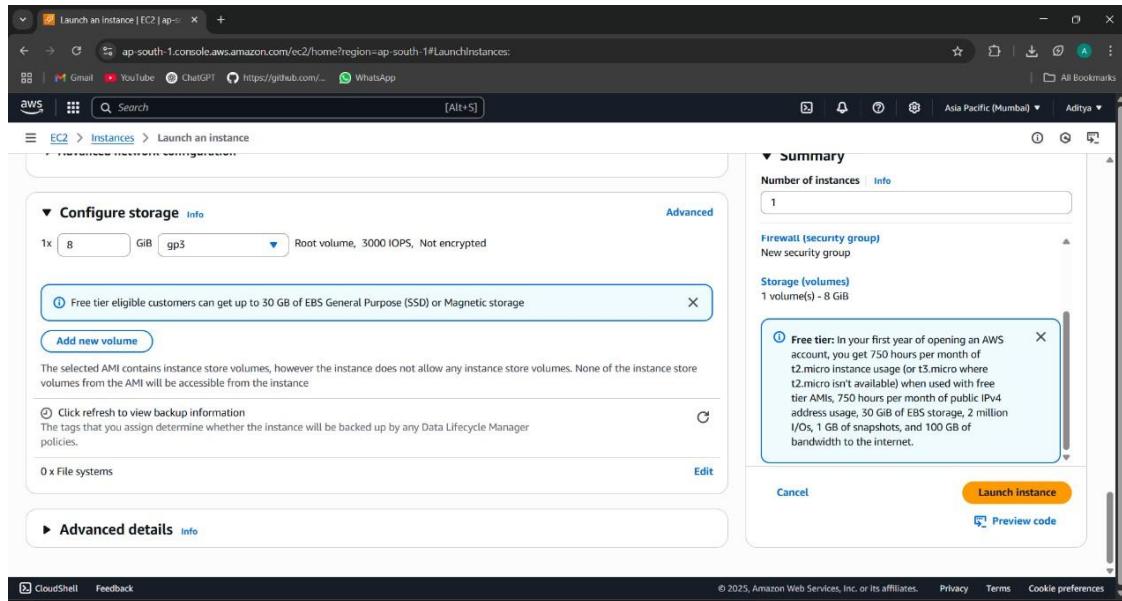
STEP 37: SECURITY GROUP RULE-1 IS DEFAULT> CLICK ON ADD SECURITY GROUP RULE



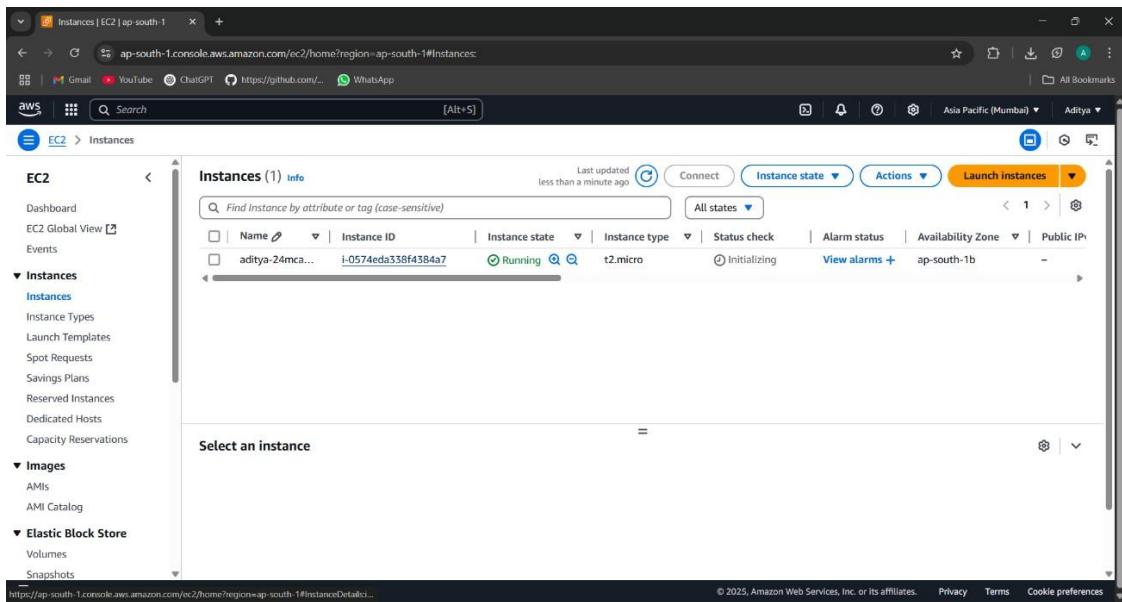
STEP 38: CHOOSE ALL ICMP-IPV4> SOURCE:0.0.0.0/0



STEP 39: CHOOSE DEFAULT CONFIGURE STORAGE: 8GB>CLICK ON LAUNCH INSTANCE

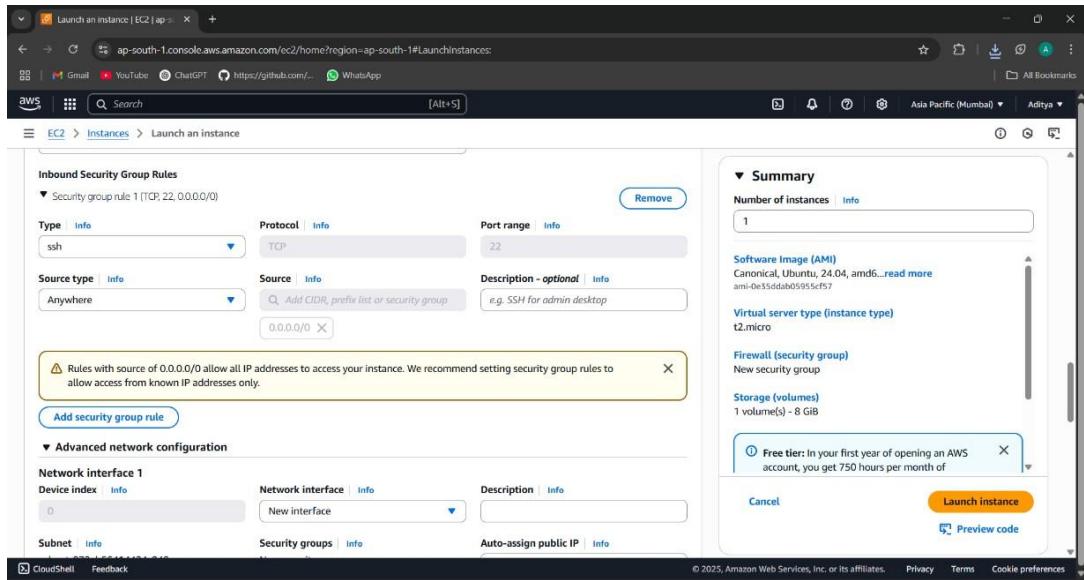


STEP 40: PUBLIC INSTANCE CREATED

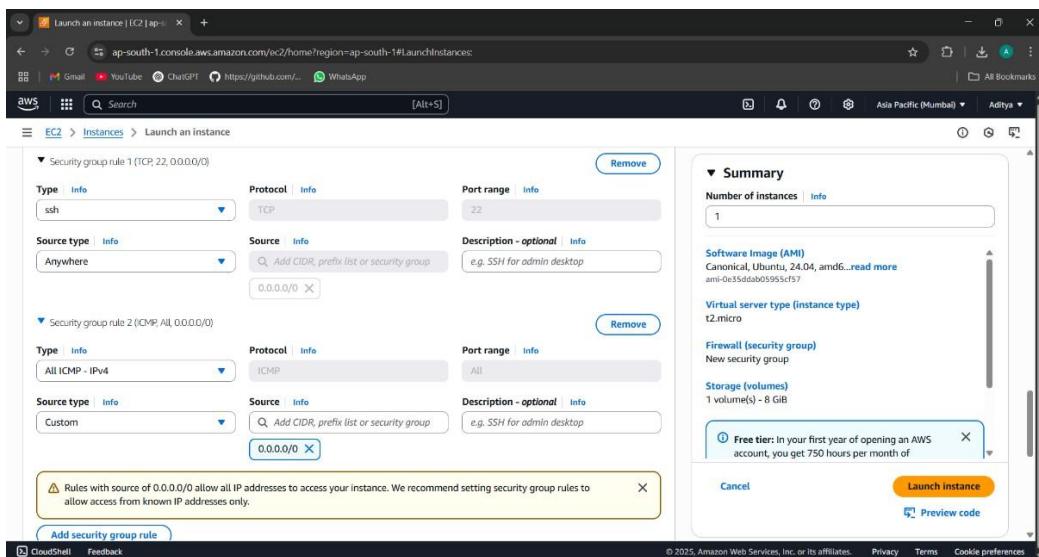


STEP 41: CREATE EC2 INSTANCE FOR PRIVATE SUBNET WITH THE SAME STEPS

DISABLE AUTO-ASSIGN-PUBLIC IP.



**STEP 42: ADD SECURITY GROUP RULE>CHOOSE ALL ICMP-IPv4>
SOURCE:11.0.1.0/24(PUBLIC SUBNET IP ADDRESS, COZ ONLY PUBLIC SUBNET CAN ACCESS
THE PRIVATE SUBNET)>KEEP DEFAULT STORAGE>CLICK ON LAUNCH INSTANCE**



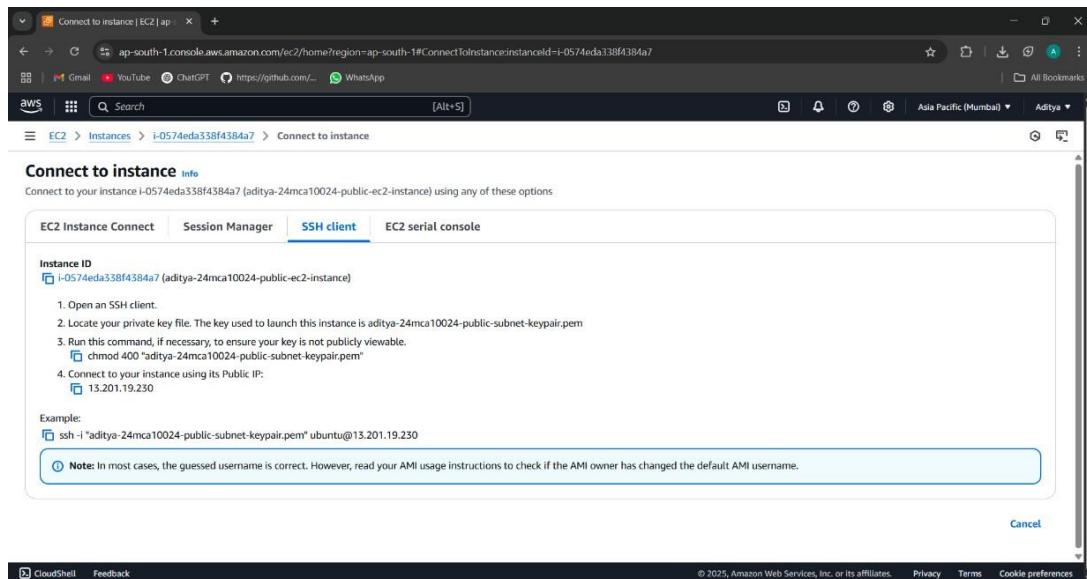
STEP 43: PRIVATE EC2 INSTANCE IS CREATED

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled 'Instances (2)'. The columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. There are two rows: one for 'aditya-24mca...' (Instance ID i-0574eda338f4384a7, State Running, Type t2.micro, Status 2/2 checks passed, Alarm status View alarms, AZ ap-south-1b, Public IP -) and another for 'aditya-24mca...' (Instance ID i-0e9aa28e10699d76, State Running, Type t2.micro, Status Initializing, Alarm status View alarms, AZ ap-south-1b, Public IP -). A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and a dropdown says 'All states'. Action buttons include 'Connect', 'Instance state', 'Actions', and 'Launch instances'.

STEP 44: NOW CONNECT THE PUBLIC EC2 INSTANCE. CLICK ON THE PUBLIC INSTANCE > CLICK ON CONNECT

The screenshot shows the AWS EC2 Instance details page for instance i-0574eda338f4384a7. The left sidebar is collapsed. The main area is titled 'Instance summary for i-0574eda338f4384a7 (aditya-24mca10024-public-ec2-instance)'. It contains several sections: 'Public IPv4 address' (13.201.19.230), 'Private IPv4 addresses' (11.0.2.54), 'Public DNS' (ip-11-0-2-54.ap-south-1.compute.internal), 'Private IP DNS name (IPv4 only)' (ip-11-0-2-54.ap-south-1.compute.internal), 'Instance type' (t2.micro), 'VPC ID' (vpc-0eddd77a9f0392b50), 'Subnet ID' (subnet-072eb5641442c549), and 'Instance ARN'. Other sections include 'Auto-assigned IP address' (13.201.19.230 [Public IP]), 'IAM Role' (none), and 'IMDSv2'. Action buttons include 'Connect', 'Instance state', and 'Actions'.

STEP 45: CLICK ON SSH CLIENT> COPY THE EXAMPLE AND PASTE IN TERMINAL TO CONNECT



STEP 46: OPEN THE TERMINAL IN YOUR LAPTOP AND TYPE THE FOLLOWING COMMAND> NOW PASTE THE COMMAND HERE FROM THE SSH CLIENT

```
ubuntu@ip-11-0-1-30:~      +  ~
System information as of Thu May 22 03:51:53 UTC 2025
System load:  0.02      Processes:          108
Usage of /:  25.0% of 6.71GB  Users logged in:    0
Memory usage: 21%          IPv4 address for enX0: 11.0.1.30
Swap usage:   0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-11-0-1-30:~$ |
```

STEP 47: PUBLIC EC2 INSTANCE IS CONNECTED

```
ubuntu@ip-11-0-1-30:~$ cat /proc/loadavg
System load: 0.02      Processes:          108
Usage of /: 25.0% of 6.71GB  Users logged in: 0
Memory usage: 21%          IPv4 address for enX0: 11.0.1.30
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-11-0-1-30:~$ |
```

STEP 48: NOW PING THE PUBLIC EC2 INSTANCE WITH THE HELP OF YOUR LOCAL TERMINAL. GO TO INSTANCE AND COPY ITS IP ADDRESS AND TYPE THE FOLLOWING COMMAND. (IT MEANS FROM OUTSIDE AWS, WE ARE ABLE TO CONNECT TO THE PUBLIC EC2 INSTANCE)

The screenshot shows the AWS Management Console with the EC2 service selected. In the left sidebar, under 'Instances', the instance 'i-01cdf56fda7e64544' is listed. The main content area displays the 'Instance summary' for this specific instance. Key details shown include:

- Instance ID:** i-01cdf56fda7e64544
- Public IPv4 address:** 13.235.245.162 [open address]
- Private IP4 addresses:** 11.0.1.30
- Private IP DNS name (IPv4 only):** ip-11-0-1-30.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-01efb8a8b5974b91 (aditya-24rma10024-vpc)
- Subnet ID:** subnet-0214083eb18d2976 (aditya-24rma10024-public-subnet)
- Instance ARN:** arn:aws:ec2:ap-south-1:123456789012:instance/i-01cdf56fda7e64544

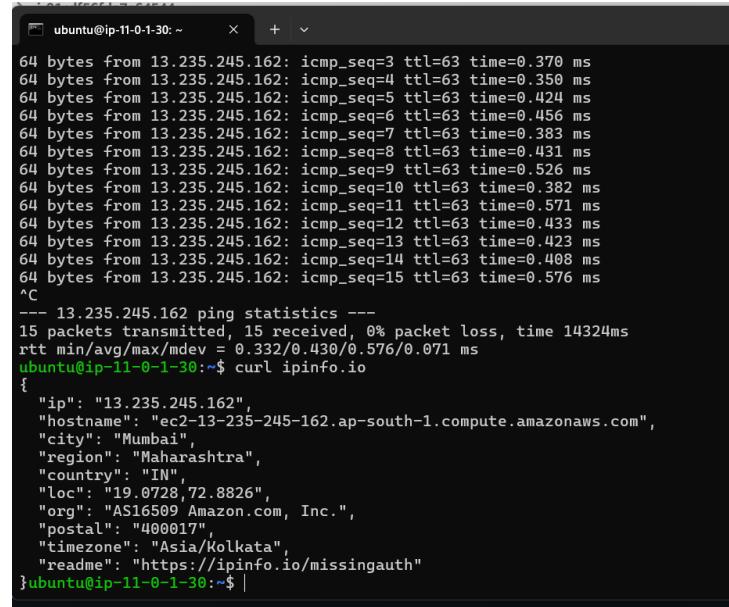
```
ubuntu@ip-11-0-1-30:~$ ping 13.235.245.162
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-11-0-1-30:~$ ping 13.235.245.162
PING 13.235.245.162 (13.235.245.162) 56(84) bytes of data.
64 bytes from 13.235.245.162: icmp_seq=1 ttl=63 time=0.332 ms
64 bytes from 13.235.245.162: icmp_seq=2 ttl=63 time=0.394 ms
64 bytes from 13.235.245.162: icmp_seq=3 ttl=63 time=0.378 ms
64 bytes from 13.235.245.162: icmp_seq=4 ttl=63 time=0.358 ms
64 bytes from 13.235.245.162: icmp_seq=5 ttl=63 time=0.424 ms
64 bytes from 13.235.245.162: icmp_seq=6 ttl=63 time=0.456 ms
64 bytes from 13.235.245.162: icmp_seq=7 ttl=63 time=0.383 ms
64 bytes from 13.235.245.162: icmp_seq=8 ttl=63 time=0.431 ms
64 bytes from 13.235.245.162: icmp_seq=9 ttl=63 time=0.526 ms
64 bytes from 13.235.245.162: icmp_seq=10 ttl=63 time=0.381 ms
64 bytes from 13.235.245.162: icmp_seq=11 ttl=63 time=0.411 ms
60 bytes from 13.235.245.162: icmp_seq=12 ttl=63 time=0.433 ms
60 bytes from 13.235.245.162: icmp_seq=13 ttl=63 time=0.423 ms
60 bytes from 13.235.245.162: icmp_seq=14 ttl=63 time=0.403 ms
64 bytes from 13.235.245.162: icmp_seq=15 ttl=63 time=0.576 ms
```
--- 13.235.245.162 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14324ms
rtt min/avg/max/mdev = 0.332/0.438/0.576/0.071 ms
ubuntu@ip-11-0-1-30:~$ |
```

## STEP 49: NOW NEXT FROM PUBLIC EC2 INSTANCE TERMINAL, TYPE THE FOLLOWING COMMAND TO ACCESS RESOURCE FROM OUTSIDE THE AWS

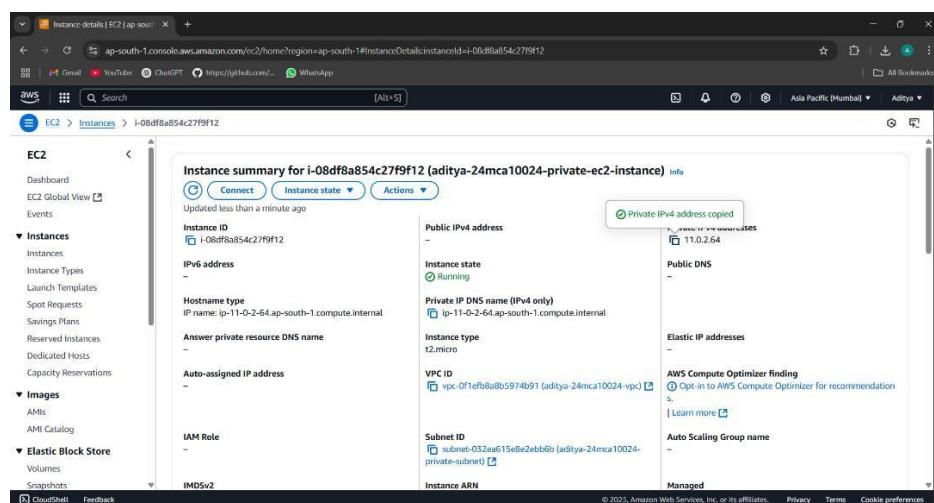
**NOTE:** curl ipinfo.io: It is a command used in the terminal to retrieve information about your current IP address using the "curl" tool and the "ipinfo.io" web service; essentially, it fetches details like your location, ISP, and public IP address by sending a request to the IPInfo.io API via the curl command.



```
ubuntu@ip-11-0-1-30:~$ ping 13.235.245.162
64 bytes from 13.235.245.162: icmp_seq=3 ttl=63 time=0.370 ms
64 bytes from 13.235.245.162: icmp_seq=4 ttl=63 time=0.350 ms
64 bytes from 13.235.245.162: icmp_seq=5 ttl=63 time=0.424 ms
64 bytes from 13.235.245.162: icmp_seq=6 ttl=63 time=0.456 ms
64 bytes from 13.235.245.162: icmp_seq=7 ttl=63 time=0.383 ms
64 bytes from 13.235.245.162: icmp_seq=8 ttl=63 time=0.431 ms
64 bytes from 13.235.245.162: icmp_seq=9 ttl=63 time=0.526 ms
64 bytes from 13.235.245.162: icmp_seq=10 ttl=63 time=0.382 ms
64 bytes from 13.235.245.162: icmp_seq=11 ttl=63 time=0.571 ms
64 bytes from 13.235.245.162: icmp_seq=12 ttl=63 time=0.433 ms
64 bytes from 13.235.245.162: icmp_seq=13 ttl=63 time=0.423 ms
64 bytes from 13.235.245.162: icmp_seq=14 ttl=63 time=0.408 ms
64 bytes from 13.235.245.162: icmp_seq=15 ttl=63 time=0.576 ms
^C
--- 13.235.245.162 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14324ms
rtt min/avg/max/mdev = 0.332/0.430/0.576/0.071 ms
ubuntu@ip-11-0-1-30:~$ curl ipinfo.io
{
 "ip": "13.235.245.162",
 "hostname": "ec2-13-235-245-162.ap-south-1.compute.amazonaws.com",
 "city": "Mumbai",
 "region": "Maharashtra",
 "country": "IN",
 "loc": "19.0728,72.8826",
 "org": "AS16509 Amazon.com, Inc.",
 "postal": "400017",
 "timezone": "Asia/Kolkata",
 "readme": "https://ipinfo.io/missingauth"
}ubuntu@ip-11-0-1-30:~$ |
```

It is seen that public EC2 instance is accessible from outside world as well as it can also access outside resources. But we know that, private instance that we have created under private subnet is not accessible from outside world but it can be accessed by the public EC2 instance only.

## STEP 50: PING PRIVATE EC2 INSTANCE FROM THE LOCAL TERMINAL. COPY THE PRIVATE IP ADDRESS OF THE INSTANCE AND TYPE (PING 11.0.2.160) IN THE LOCAL TERMINAL.



```
ubuntu@ip-11-0-1-30: ~ + v
"city": "Mumbai",
"region": "Maharashtra",
"country": "IN",
"loc": "19.0728,72.8826",
"org": "AS16509 Amazon.com, Inc.",
"postal": "400017",
"timezone": "Asia/Kolkata",
"readme": "https://ipinfo.io/missingauth"
]ubuntu@ip-11-0-1-30:~$ ping 11.0.2.64
PING 11.0.2.64 (11.0.2.64) 56(84) bytes of data.
64 bytes from 11.0.2.64: icmp_seq=1 ttl=64 time=1.16 ms
64 bytes from 11.0.2.64: icmp_seq=2 ttl=64 time=0.853 ms
64 bytes from 11.0.2.64: icmp_seq=3 ttl=64 time=0.802 ms
64 bytes from 11.0.2.64: icmp_seq=4 ttl=64 time=0.797 ms
64 bytes from 11.0.2.64: icmp_seq=5 ttl=64 time=0.793 ms
64 bytes from 11.0.2.64: icmp_seq=6 ttl=64 time=1.11 ms
64 bytes from 11.0.2.64: icmp_seq=7 ttl=64 time=0.786 ms
64 bytes from 11.0.2.64: icmp_seq=8 ttl=64 time=0.807 ms
64 bytes from 11.0.2.64: icmp_seq=9 ttl=64 time=0.823 ms
64 bytes from 11.0.2.64: icmp_seq=10 ttl=64 time=0.804 ms
64 bytes from 11.0.2.64: icmp_seq=11 ttl=64 time=0.821 ms
64 bytes from 11.0.2.64: icmp_seq=12 ttl=64 time=0.872 ms
64 bytes from 11.0.2.64: icmp_seq=13 ttl=64 time=0.888 ms
64 bytes from 11.0.2.64: icmp_seq=14 ttl=64 time=0.790 ms
64 bytes from 11.0.2.64: icmp_seq=15 ttl=64 time=0.822 ms
^C
--- 11.0.2.64 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14354ms
rtt min/avg/max/mdev = 0.786/0.861/1.159/0.110 ms
ubuntu@ip-11-0-1-30:~$ |
```

**ITS SHOWING REQUEST TIMED OUT. i.e., IT IS UNABLE TO CONNECT TO THE PRIVATE EC2 INSTANCE FROM THE LOCAL TERMINAL. NOW LET'S TRY TO CONNECT IT FROM THE PUBLIC EC2 INSTANCE TERMINAL.**

**STEP 50: TYPE THE FOLLOWING COMMAND TO CONNECT TO THE PRIVATE EC2 INSTANCE (PING 11.0.2.160). CONNECTION ESTABLISHED AS SHOWN BELOW.**

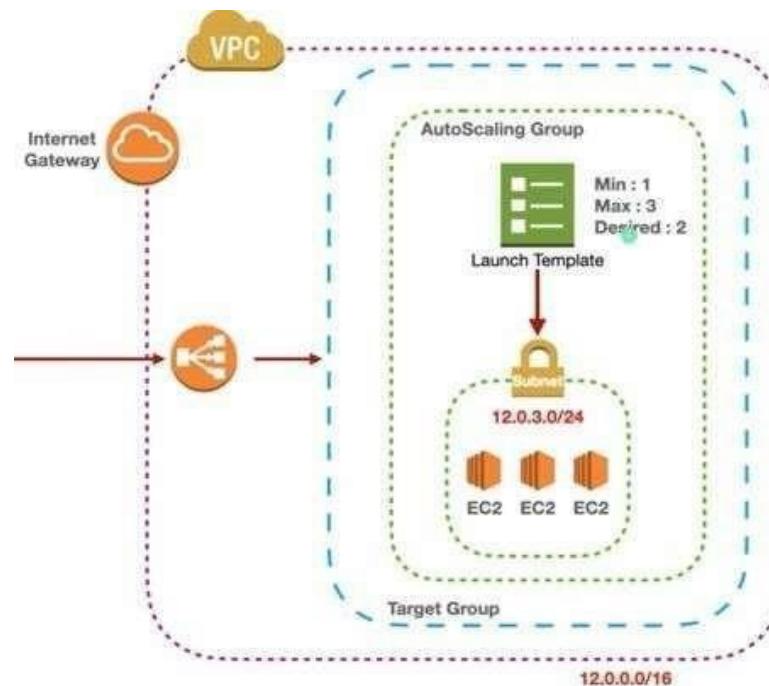
```
ubuntu@ip-11-0-1-30: ~ + v
"city": "Mumbai",
"region": "Maharashtra",
"country": "IN",
"loc": "19.0728,72.8826",
"org": "AS16509 Amazon.com, Inc.",
"postal": "400017",
"timezone": "Asia/Kolkata",
"readme": "https://ipinfo.io/missingauth"
]ubuntu@ip-11-0-1-30:~$ ping 11.0.2.64
PING 11.0.2.64 (11.0.2.64) 56(84) bytes of data.
64 bytes from 11.0.2.64: icmp_seq=1 ttl=64 time=1.16 ms
64 bytes from 11.0.2.64: icmp_seq=2 ttl=64 time=0.853 ms
64 bytes from 11.0.2.64: icmp_seq=3 ttl=64 time=0.802 ms
64 bytes from 11.0.2.64: icmp_seq=4 ttl=64 time=0.797 ms
64 bytes from 11.0.2.64: icmp_seq=5 ttl=64 time=0.793 ms
64 bytes from 11.0.2.64: icmp_seq=6 ttl=64 time=1.11 ms
64 bytes from 11.0.2.64: icmp_seq=7 ttl=64 time=0.786 ms
64 bytes from 11.0.2.64: icmp_seq=8 ttl=64 time=0.807 ms
64 bytes from 11.0.2.64: icmp_seq=9 ttl=64 time=0.823 ms
64 bytes from 11.0.2.64: icmp_seq=10 ttl=64 time=0.804 ms
64 bytes from 11.0.2.64: icmp_seq=11 ttl=64 time=0.821 ms
64 bytes from 11.0.2.64: icmp_seq=12 ttl=64 time=0.872 ms
64 bytes from 11.0.2.64: icmp_seq=13 ttl=64 time=0.888 ms
64 bytes from 11.0.2.64: icmp_seq=14 ttl=64 time=0.790 ms
64 bytes from 11.0.2.64: icmp_seq=15 ttl=64 time=0.822 ms
^C
--- 11.0.2.64 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14354ms
rtt min/avg/max/mdev = 0.786/0.861/1.159/0.110 ms
ubuntu@ip-11-0-1-30:~$ |
```

**RELEASE ALL THE RESOURCES THAT ARE USED IN THE EXECUTION OF THE LAB.**

1. TERMINATE THE EC2 INSTANCES.
2. DELETE ROUTE TABLE:
  - a. ROUTE TABLE>EDIT ROUTE>REMOVE NAT GATEWAY (PRIVATE)
  - b. ROUTE TABLE>EDIT ROUTE>REMOVE INTERNET GATEWAY (PUBLIC)
  - c. ROUTE TABLE>SUBNET ASSOCIATIONS>EDIT SUBNET ASSOCIATIONS>DESELECT>SAVE CHANGES
  - d. SELECT BOTH THE ROUTE TABLES>ACTIONS>DELETE ROUTE TABLES.
3. DELETE SUBNETS:
  - a. SUBNETS> SELECT BOTH THE SUBNET ACTIONS>ACTIONS> DELETE SUBNET
4. DELETE NAT GATEWAY:
  - a. NAT GATEWAY>SELECT THE NAT GATEWAY>ACTIONS> DELETE NAT GATEWAY
5. DELETE VPC
  - a. VPC>SELECT THE VPC CREATED>ACTIONS>DELETE
6. RELEASE ELASTIC IP
  - a. ELASTIC IP>SELECT THE ELASTIC IP>ACTIONS>RELEASE THE ELASTIC IP

|                  |                                                                                       |
|------------------|---------------------------------------------------------------------------------------|
| Date: 23/05/2025 | Title                                                                                 |
| Exp. No: 06      | Show to implement load balancer, target group & auto-scaling group with EC2 Instance. |

## SHOW TO IMPLEMENT LOAD BALANCER, TARGET GROUP & AUTO-SCALING GROUP WITH EC2 INSTANCE



In AWS, combining a Load Balancer, Target Group, and Auto Scaling Group (ASG) provides a powerful and automated solution for managing application availability, scalability, and fault tolerance.

An Application Load Balancer (ALB) distributes incoming traffic across multiple EC2 instances based on routing rules. It improves performance and availability by ensuring that traffic is routed only to healthy instances. The Load Balancer listens on specified ports (like HTTP on port 80 or HTTPS on port 443) and forwards requests to registered targets.

A Target Group is a logical group of EC2 instances (or other resources) that receive traffic from the Load Balancer. Each target group is associated with a specific protocol and port, and includes health check settings to monitor the availability of its targets. EC2 instances are registered to a target group so that they can be used by the load balancer.

An Auto Scaling Group (ASG) automatically manages the number of EC2 instances based on scaling policies. It ensures that a minimum number of instances are always running and can scale out (add instances) or scale in (remove instances) based on metrics such as CPU usage, memory utilization, or request rate. The ASG launches EC2 instances using a Launch Template or Launch Configuration, and registers them with the appropriate Target Group so that they can receive traffic from the Load Balancer.

This architecture ensures high availability and fault tolerance by distributing load evenly and replacing unhealthy instances automatically. It also optimizes cost and performance by scaling the infrastructure dynamically according to demand. This combination is ideal for modern web applications, APIs, and microservices deployed in a cloud environment.

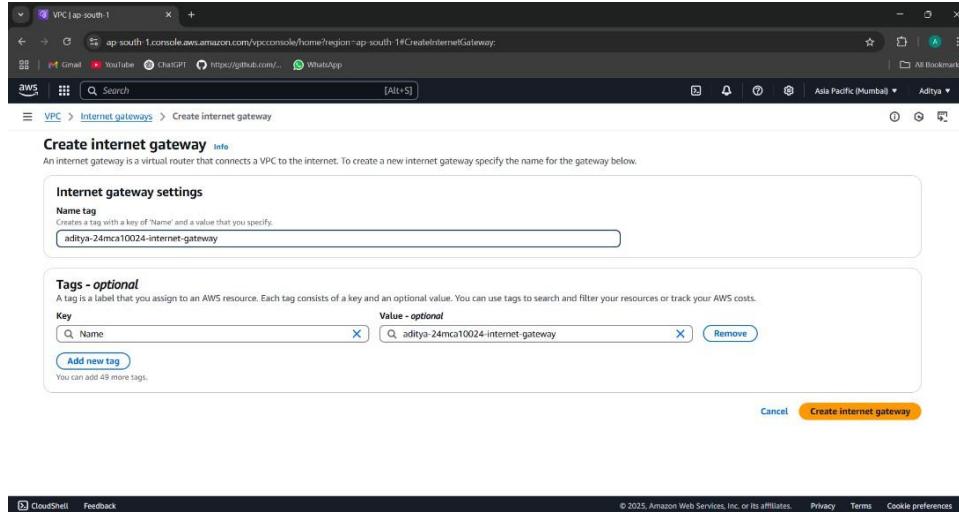
## STEP1: CREATE A VPC>WRITE VPC NAME>IPv4 CIDR:12.0.0.0/16>CREATE VPC

The screenshot shows the 'Create VPC' wizard in the AWS VPC service. The 'VPC settings' step is active. Key configuration details are visible:

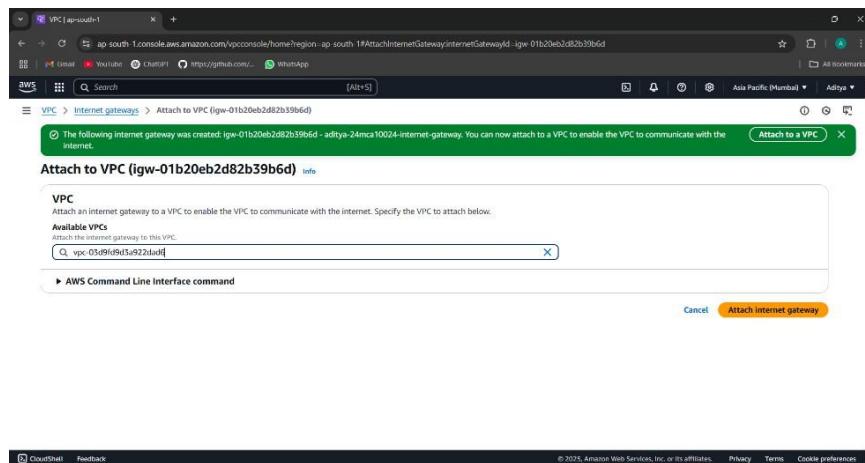
- Resources to create:** VPC only (selected)
- Name tag - optional:** aditya-24mca10024-vpc
- IPv4 CIDR block:** 12.0.0.16 (selected)
- IPv6 CIDR block:** No IPv6 CIDR block (selected)

At the bottom, there are 'Next Step' and 'Cancel' buttons, along with links for CloudShell and Feedback.

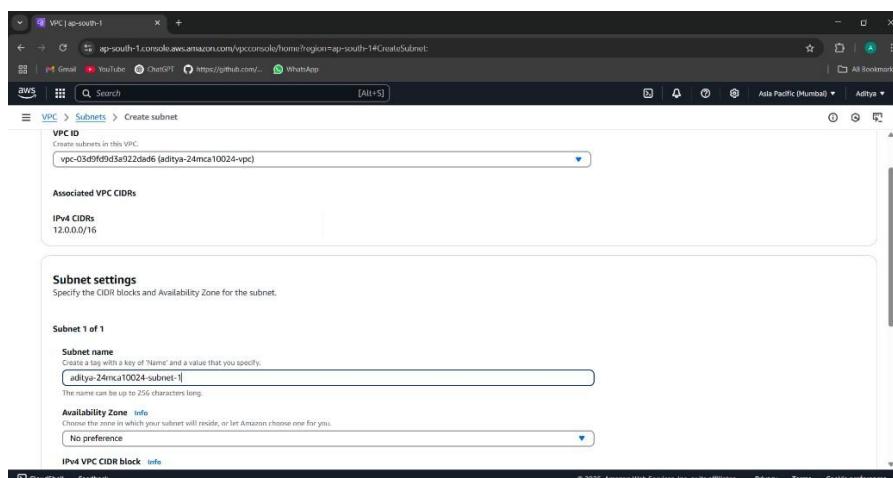
## STEP2: NEXT CLICK ON INTERNET GATEWAY> CREATE INTERNET GATEWAY>TYPE NAME> CLICK ON CREATE



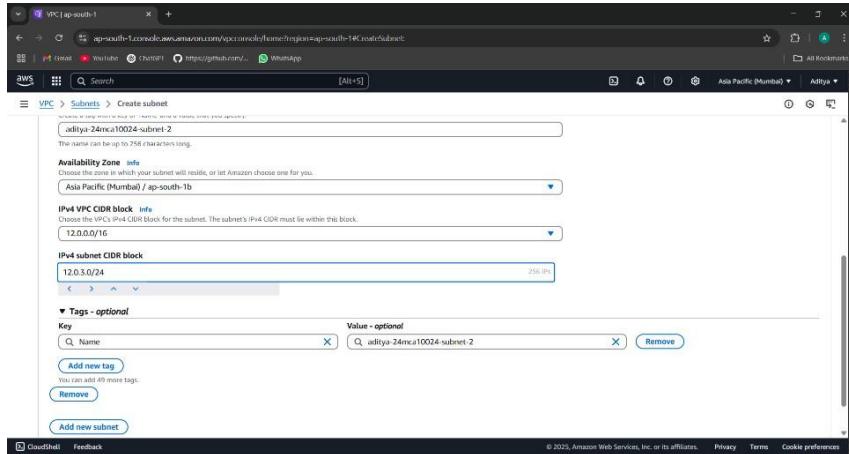
## STEP3: NEXT CLICK ON ATTACH TO VPC> CHOOSE THE INTERNET GATEWAY>CLICK ON ATTACH



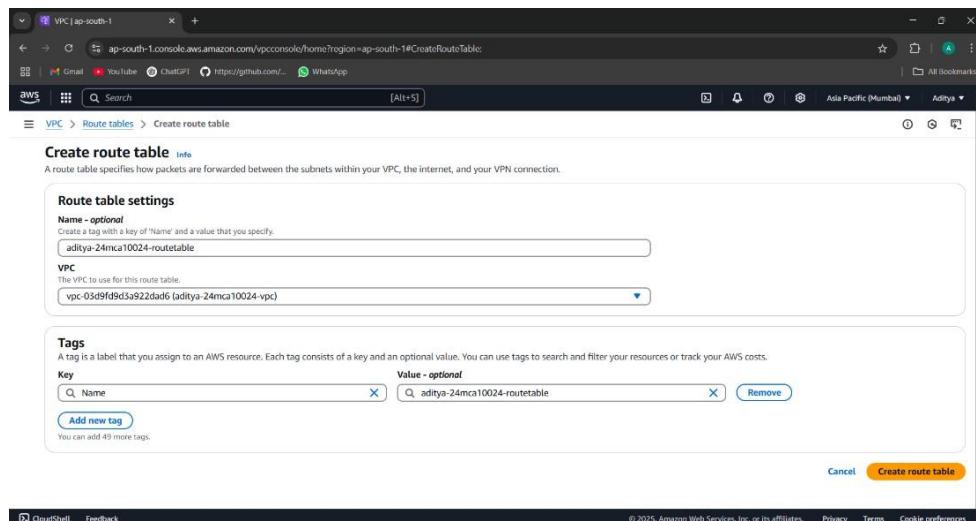
## STEP4: CREATE SUBNET> SELECT THE VPC>TYPE THE NAME OF SUBNET\_1>AVAILABILITY ZONE:ap-south 1a>IPv4 CIDR:12.0.0.1/24



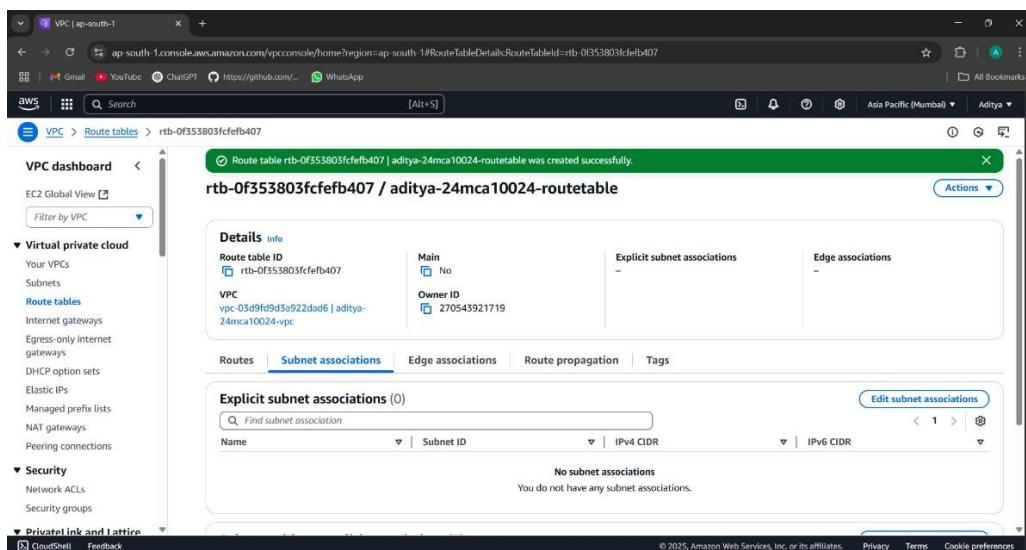
**STEP5: CLICK ON ADD SUBNET>TYPE THE NAME OF SUBNET\_2>AVAILABILITY ZONE:ap-south  
1b>IPv4 CIDR:12.0.3.0/24**



**STEP6: NOW CREATE ROUTE TABLE: CLICK ON ROUTE TABLES>CREATE ROUTE TABLE>ROUTE TABLE NAME> CHOOSE VPC>CREATE ROUTE TABLE**



**STEP7: NOW ASSOCIATE ROUTE TABLE WITH THE SUBNETS: CLICK ON SUBNET ASSOCIATIONS>EDIT SUBNET ASSOCIATION**



## STEP8: SELECT BOTH THE SUBNETS>SAVE ASSOCIATIONS

The screenshot shows the 'Edit subnet associations' dialog in the AWS VPC console. It lists two subnets under 'Available subnets': 'aditya-24mca10024-subnet-1' and 'aditya-24mca10024-subnet-2'. Both subnets are currently associated with route table 'rtb-0f555b03cfefb0d7'. In the 'Selected subnets' section, both subnets are selected. At the bottom right, there is a 'Save associations' button.

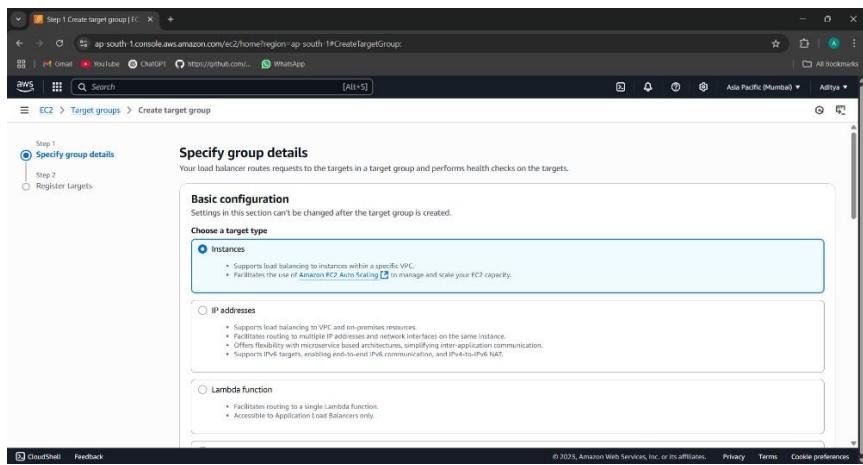
## STEP9: CLICK ON ROUTE>EDIT ROUTE>ADD ROUTE> SELECT THE DATA AS SHOWN>SAVE CHANGES

The screenshot shows the 'Edit routes' dialog in the AWS VPC console. It displays a table of routes. There is one existing route: Destination '12.0.0.0/16', Target 'local', Status 'Active', and Propagated 'No'. A new route is being added: Destination '0.0.0.0/0', Target 'aws-optimized', Status 'Active', and Propagated 'No'. At the bottom left, there is a 'Add route' button.

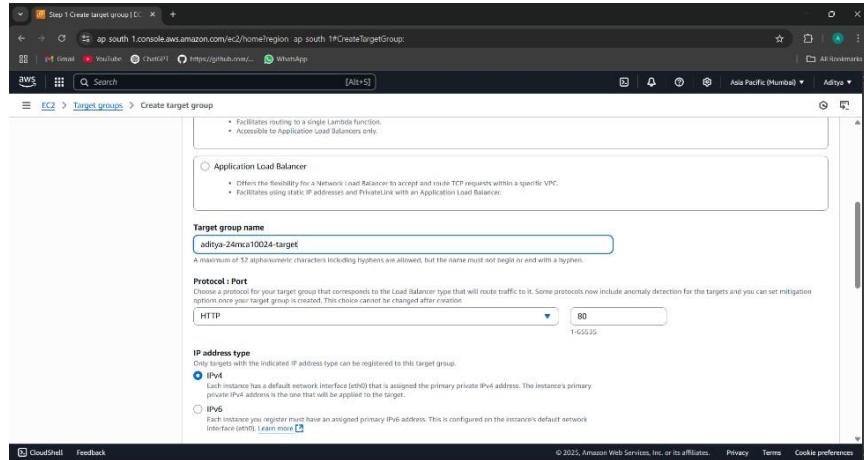
## STEP10: GO TO EC2 DASHBOARD>CLICK ON TARGET GROUPS

The screenshot shows the 'Target groups' section in the AWS EC2 dashboard. It displays a table with the message 'No target groups'. A 'Create target group' button is located at the top right.

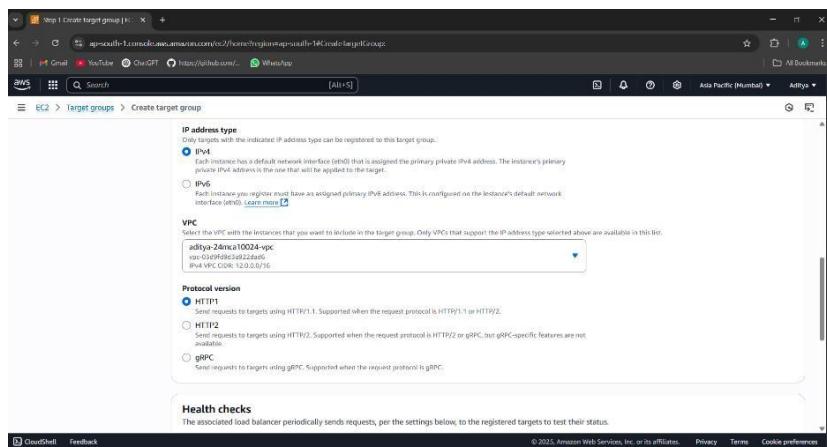
## STEP11: CHOOSE INSTANCES



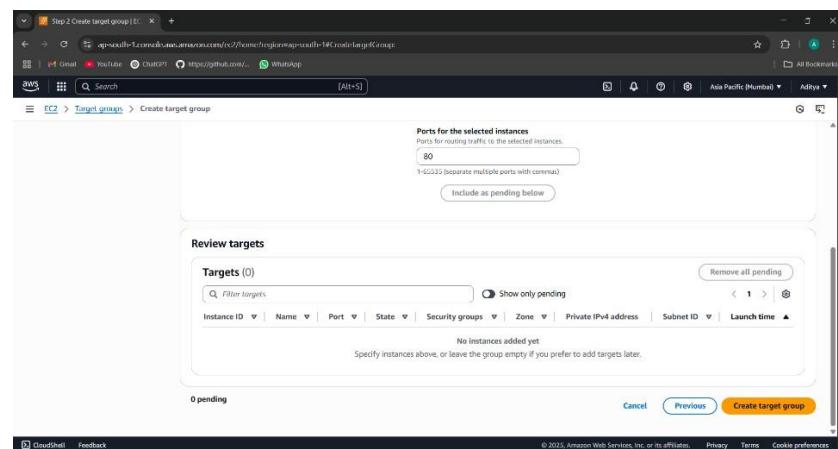
## STEP12: WRITE A NAME TO THE TARGET GROUP



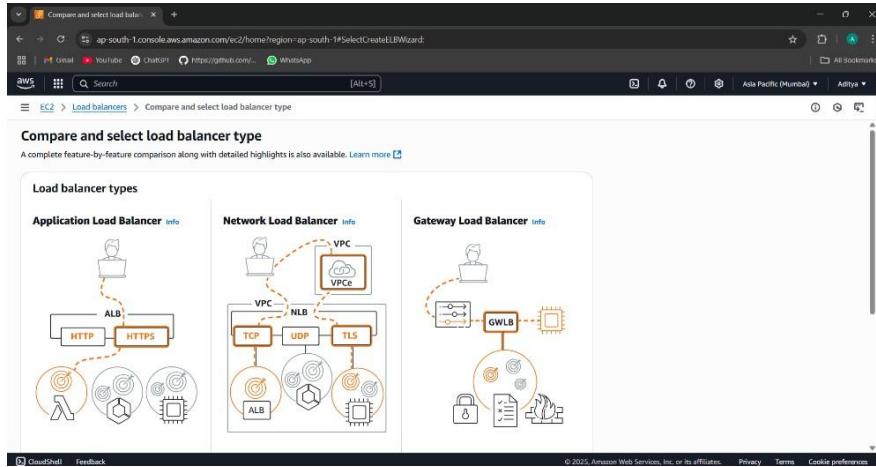
## STEP13: SELECT THE VPC> CLICK ON NEXT



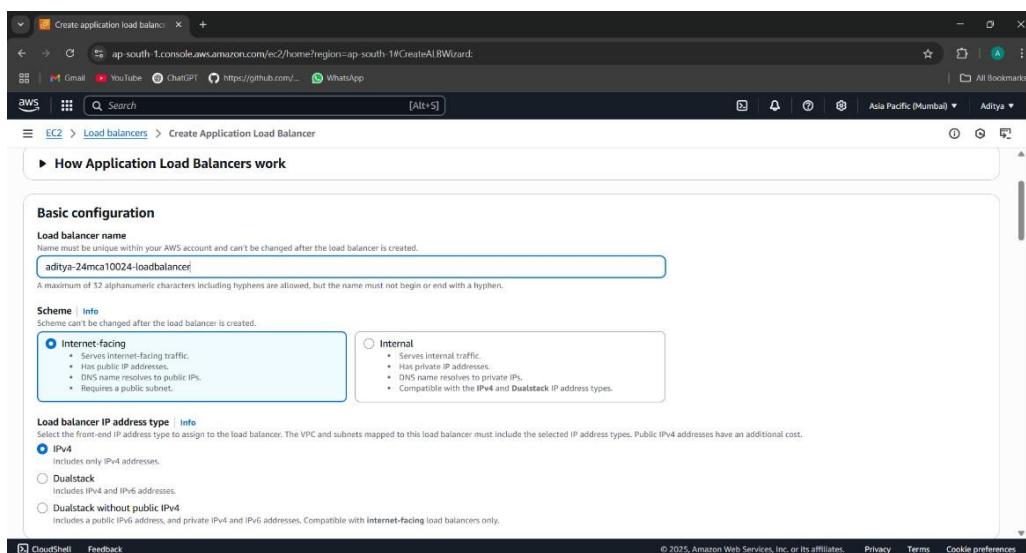
## STEP14: CLICK ON CREATE TARGET GROUP



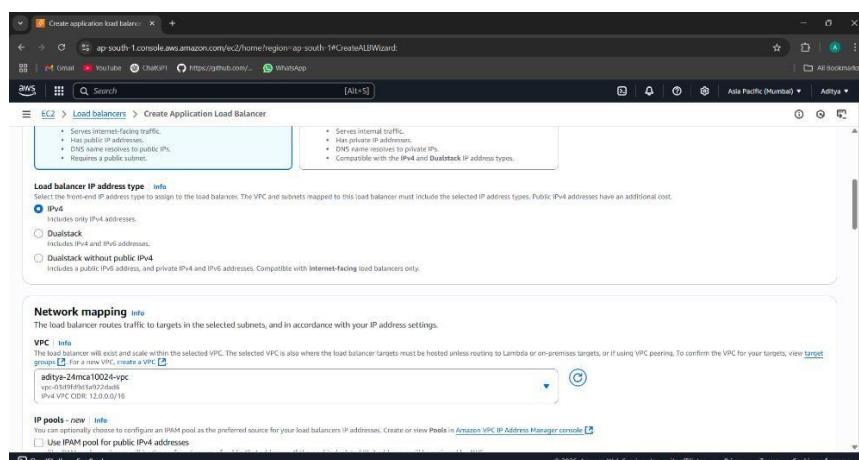
## STEP15: CLICK ON APPLICATION LOAD BALANCER



## STEP16: WRITE THE NAME OF THE LOAD BALANCER



## STEP17: SELECT THE VPC CREATED



## STEP18: SELECT BOTH THE ZONE(SUBNETS)

The screenshot shows the 'IP pools - new' step of the wizard. It lists two subnets under 'Availability Zones and subnets': 'ap-south-1a (ap-s1-a1)' and 'ap-south-1b (ap-s1-a2)'. Each subnet has its CIDR range and IP address count displayed.

## STEP19: CLICK ON CREATE A NEW SECURITY GROUP> TYPE THE NAME AND DESCRIPTON OF THE SECURITY GROUP>SELECT THE VPC

The screenshot shows the 'Security groups' step of the wizard. A new security group named 'default' is selected. Below it, a 'Listeners and routing' section is shown, featuring a single rule for 'Listener HTTP:80' with 'Protocol' set to 'HTTP' and 'Port' set to '80'. The 'Default action' is set to 'Forward to' and 'Select a target group'.

## STEP20: SET THE IN BOUND RULE: CLICK ON ADD RULES> SET HTTP>SOURCE:0.0.0.0/0> CLICK ON CREATE SECURITY GROUP.

The screenshot shows the 'Inbound rules' and 'Outbound rules' sections of the security group configuration. In the 'Inbound rules' section, a rule is defined for 'Type: HTTP', 'Protocol: TCP', and 'Port range: 80'. The source is set to 'Anywhere' with the value '0.0.0.0/0'. In the 'Outbound rules' section, a rule is defined for 'All traffic' with 'Protocol: All', 'Port range: All', and 'Destination: Custom' (set to '0.0.0.0/0').

## STEP21: NOW SELECT THE SECURITY GROUP THAT YOU HAVE CREATED

The screenshot shows the 'Security groups' section of the AWS Create Application Load Balancer wizard. It displays two security groups: 'aditya-24mca10024-security-group' and 'default'. A note at the top states: "Only CIDR blocks corresponding to the load balancer IP address type are used. At least 4 available IP addresses are required for your load balancer to scale efficiently." Below the security groups, there is a 'Listeners and routing' section.

## STEP22: LISTENERS & ROUTING SECTION: SELECT TARGET GROUP THAT YOU HAVE CREATED

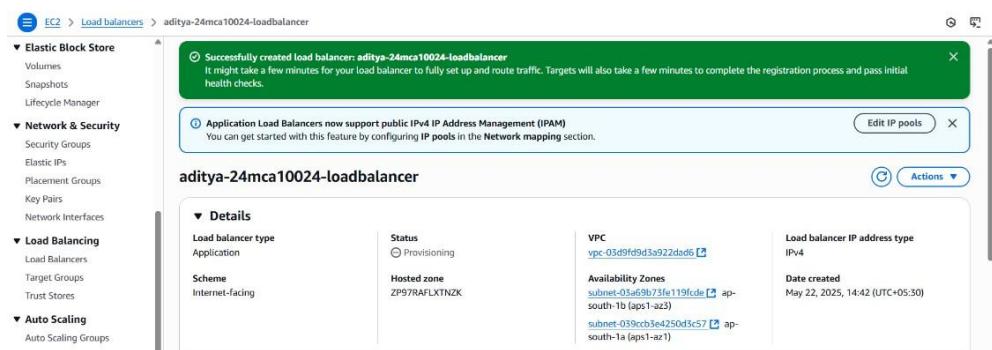
The screenshot shows the 'Listeners and routing' section of the AWS Listener and Routing configuration page. It lists a single listener for port 80, configured to forward requests to the 'aditya-24mca10024-target' target group. There is also a 'Create target group' link for creating new target groups.

## STEP23: VERIFY THE DETAILS IN THE SUMMARY>CLICK ON CREATE LOAD BALANCER

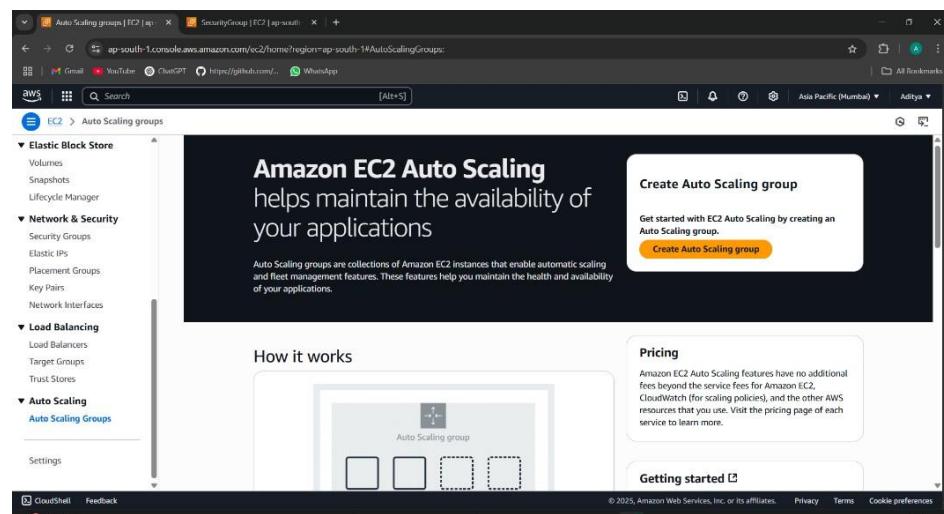
The screenshot shows the 'Summary' page of the AWS Load Balancer configuration. It provides a review of the configurations:

- Basic configuration:** Name: aditya-24mca10024-loadbalancer, Scheme: Internet-facing, IP address type: IPv4.
- Network mapping:** VPC: vpc-03d9fd9d3a922dad6, Public IPv4 IPAM pool: -.
- Availability Zones and subnets:** ap-south-1a (subnet-039ccb3e4250d3c57), ap-south-1b (subnet-03a69b73fe119fcde).
- Security groups:** aditya-24mca10024-security-group (sg-04af17c5d325bc2c9), default (sg-0ce010de57bae4a51).
- Listeners and routing:** HTTP:80 | Target group: aditya-24mca10024-target.
- Service integrations:** Amazon CloudFront + AWS Web Application Firewall (WAF): -, AWS WAF: -, AWS Global Accelerator: -.
- Tags:** -

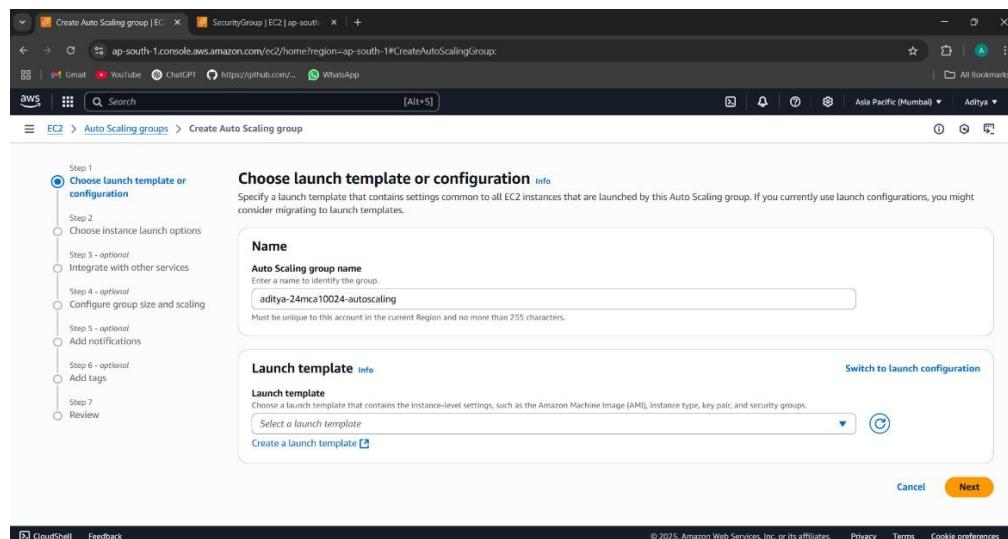
## STEP24: LOAD BALANCER IS CREATED



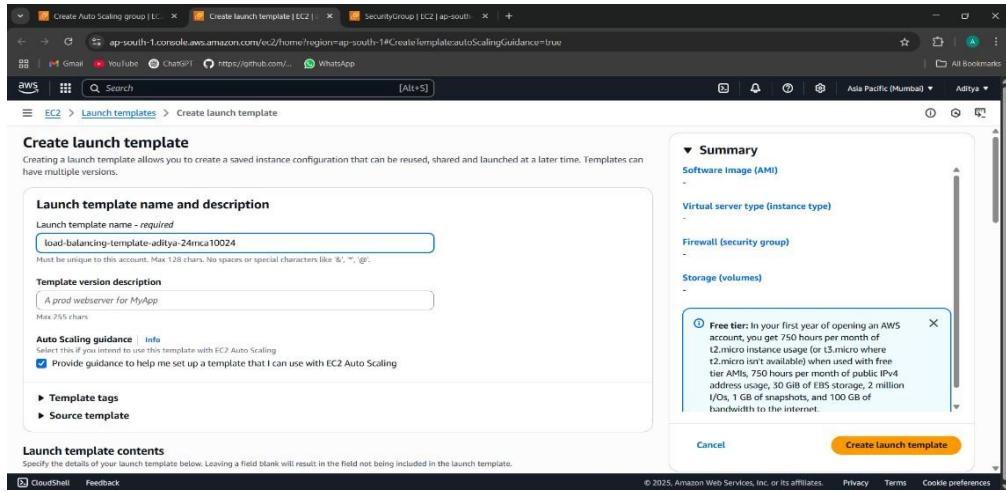
## STEP25: CLICK ON AUTO-SCALING GROUP> CREATE AUTO-SCALING GROUP



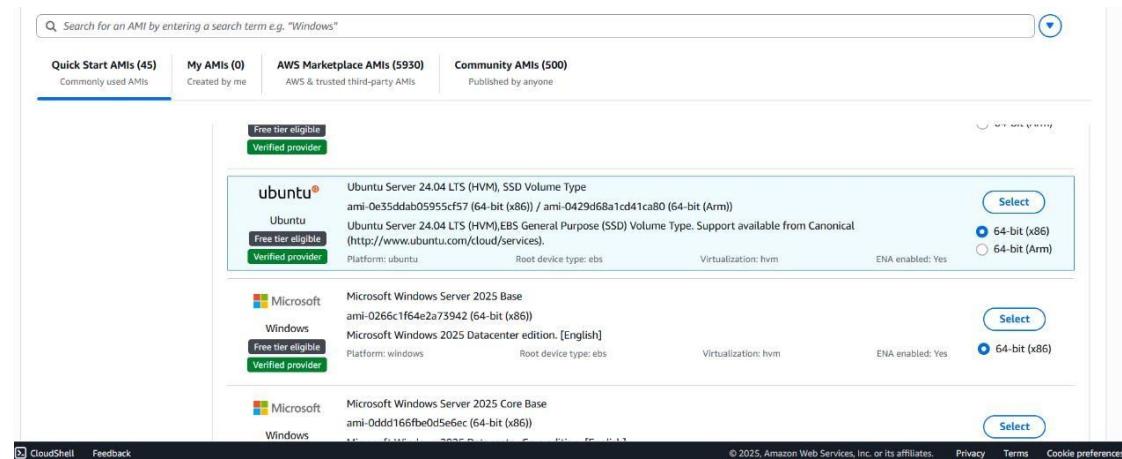
## STEP26: GIVE A NAME TO AUTO-SCALING >CLICK ON CREATE A LAUNCH TEMPLATE



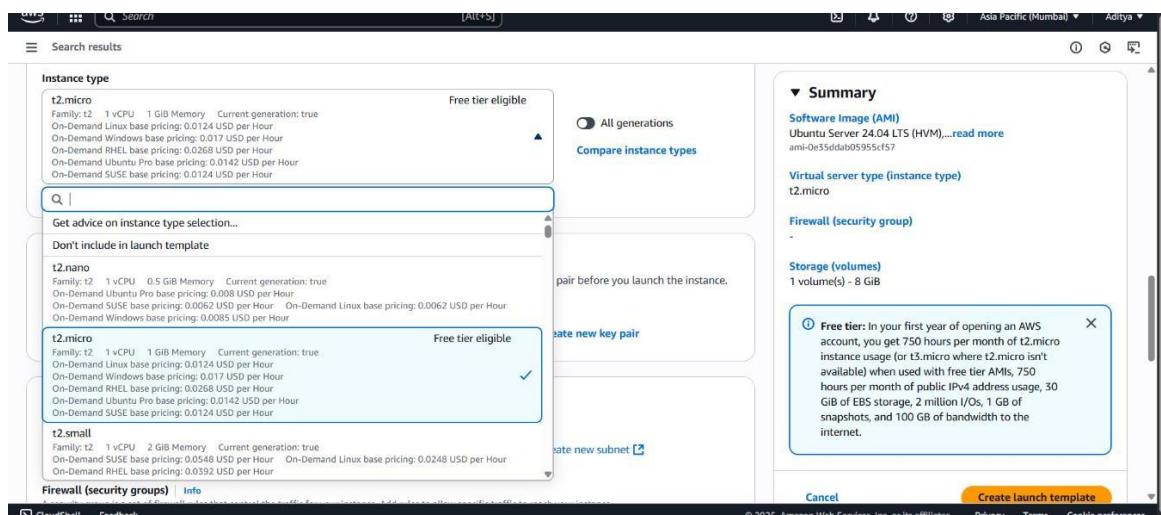
## STEP27: GIVE A TEMPLATE NAME



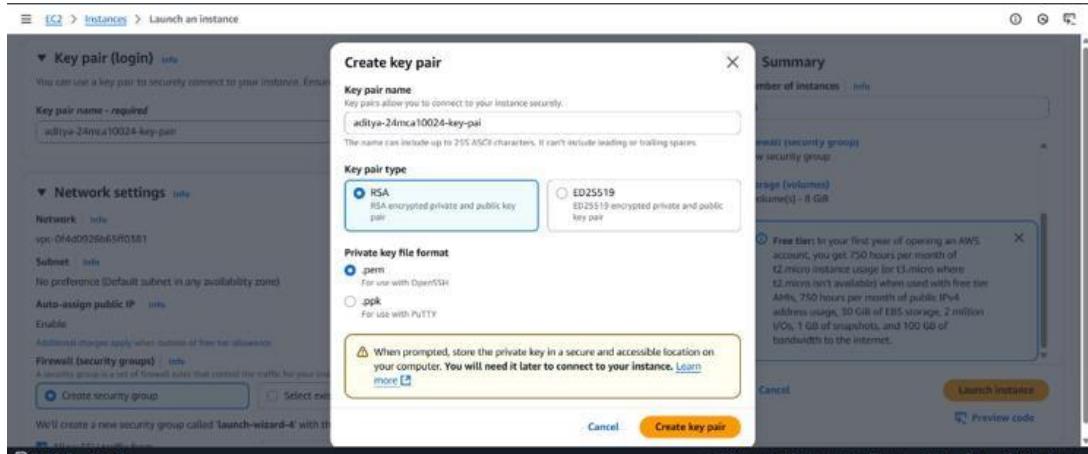
## STEP28: SELECT UBUNTU



## STEP29: SELECT INSTANCE TYPE



## STEP30: CREATE A KEY PAIR



## STEP31: NEXT WE HAVE TO CREATE A SECURITY GROUP TO BE ADDED IN THE LAUNCH TEMPLATE

- a. SEARCH VPC>OPEN VPC IN A NEW TAB>CLICK ON SECURITY GROUP>CREATE SECURITY GROUP
- b. WRITE A NAME & DESCRIPTION TO THE SECURITY GROUP> CHOOSE THE VPC THAT YOU HAVE CREATED

The screenshot shows the 'Basic details' step of the 'Create security group' wizard. The 'Security group name' field contains 'aditya-24mca10024-launch-temp-security-group'. The 'Description' field contains 'Launch template security group'. The 'VPC info' dropdown shows 'vpc-03d9fd9d3a922dad6 (aditya-24mca10024-vpc)'. The 'Create security group' button is highlighted in orange.

### STEP32: SET IN BOUND RULE:

- CLICK ON ADD RULE> SELECT HTTP & SOURCE: ANYWHERE (0.0.0.0/0).
- CLICK ON ADD RULE> SELECT SSH & SOURCE: ANYWHERE (0.0.0.0/0).

THEN CLICK ON ADD SECURITY GROUP

Inbound rules [Info](#)

| Type | Protocol | Port range | Source                       | Description - optional           |
|------|----------|------------|------------------------------|----------------------------------|
| HTTP | TCP      | 80         | Anyw... <a href="#">Info</a> | 0.0.0.0/0 <a href="#">Delete</a> |
| SSH  | TCP      | 22         | Anyw... <a href="#">Info</a> | 0.0.0.0/0 <a href="#">Delete</a> |

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. [X](#)

### STEP33: NOW SELECT THE SECURITY GROUP THAT YOU HAVE CREATED IN THE LAUNCH TEMPLATE> AUTO ASSIGN PUBLIC IP: ENABLE> KEEP DEFAULT EBS VOLUME>CLICK ON CREATE LAUNCH TEMPLATE

Search results

Select existing security group  Create security group

Common security groups [Info](#)

Select security groups [VPC: vpc-03d9ff951a9272ad06](#)

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Advanced network configuration

Network interface 1

Device index [Info](#) 0 Network interface [Info](#) New interface Description [Info](#) Remove

Existing network interface are not recommended when creating a template for auto-scaling.

Subnet [Info](#) Don't include in launch template Not applicable for EC2 Auto Scaling

Security groups [Info](#) Select security groups Show all selected (1)

Primary IP [Info](#) Secondary IP [Info](#) IPv6 IPs [Info](#)

Don't include in launch template Not applicable for EC2 Auto Scaling

Don't include in launch template Not applicable for EC2 Auto Scaling

Don't include in launch template Not applicable for EC2 Auto Scaling

Summary

Software Image (AMI)  
Ubuntu Server 24.04 LTS (HVM)... [read more](#)  
ami-0x15d6ab05595cf57

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
aditya-24mca10024-launch-temp-security-group

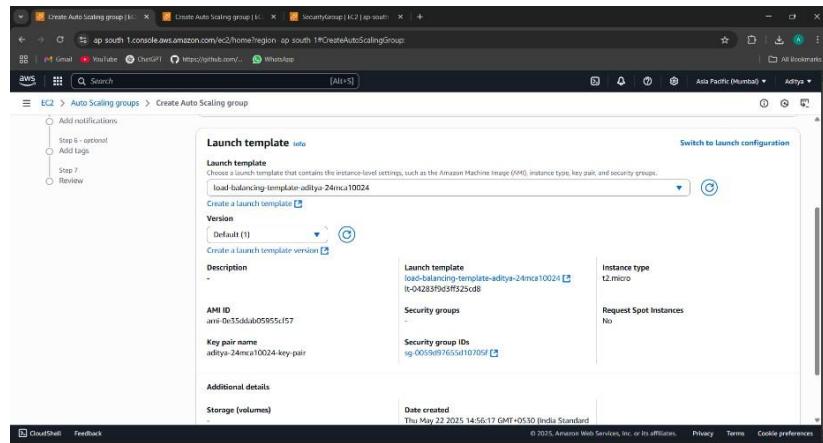
Storage (volumes)  
1 volume(s) - 8 GiB

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.

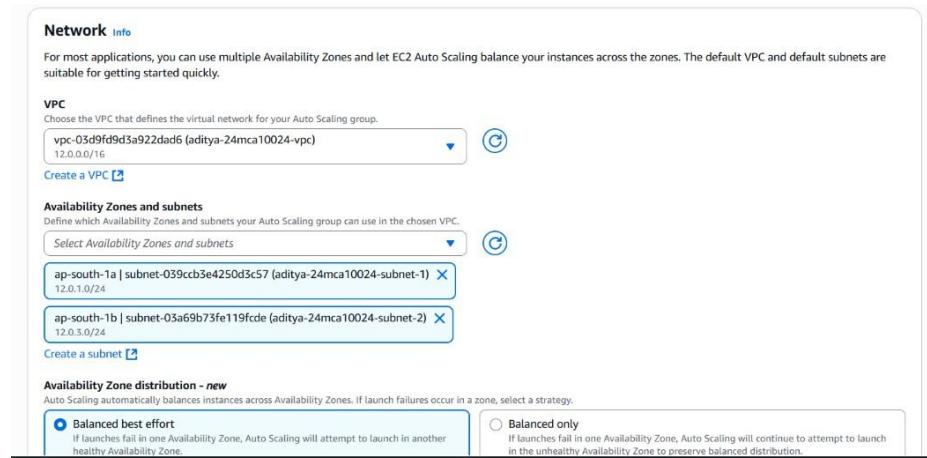
Create launch template [Cancel](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

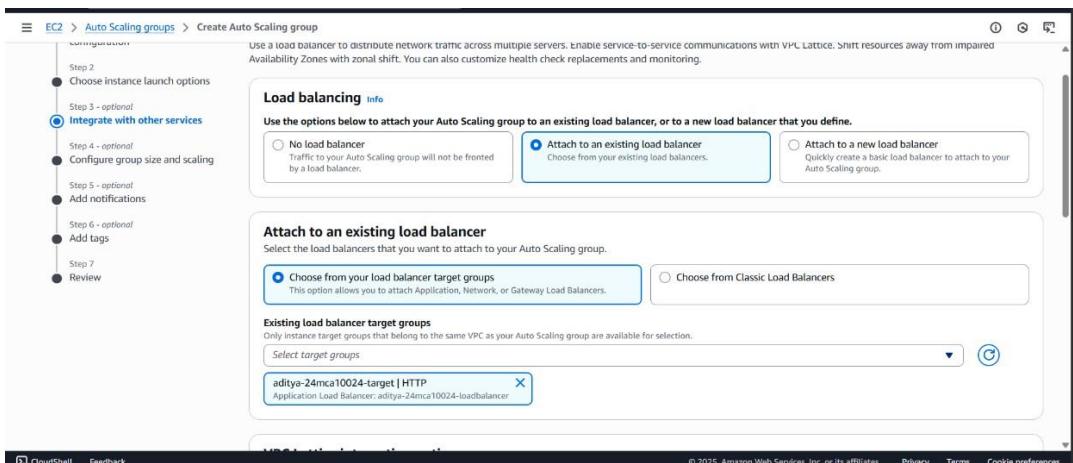
## STEP34: NOW SELECT THE LAUNCH TEMPLATE FOR AUTO-SCALING GROUP> CLICK ON NEXT



## STEP35: SELECT VPC THAT YOU HAVE CREATED>SELECT BOTH THE AVAILABILITY ZONES>CLICK ON NEXT.



## STEP36: SELECT ATTACH TO AN EXISTING LOAD BALANCER>CHOOSE FROM YOUR LOAD BALANCER TARGET GROUPS>CHOOSE THE TARGET GROUP THAT YOU HAVE CREATED



## STEP37: SELECT NO VPC LATTICE SERVICE

**VPC Lattice integration options** Info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

**Select VPC Lattice service to attach**

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.

(i) 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.

20 seconds

## STEP38: SELECT TURN ON ELASTIC LOAD BALANCING>SET 20 SECONDS>CLICK ON NEXT

**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.

(i) 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.

20 seconds

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

## STEP39: SET DESIRED CAPACITY:2, MIN CAPACITY:1 & MAXIMUM CAPACITY:3

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

**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   
Equal or less than desired capacity

Max desired capacity   
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.

## STEP40: 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.

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

Flexible  
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

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## STEP41: CLICK ON AUTO-SCALING GROUP

EC2 > Auto Scaling groups > Create Auto Scaling group

Capacity Reservation preference

Preference Default

Capacity Reservation IDs -

Resource Groups -

Step 5: Add notifications

Notifications  
No notifications

Step 6: Add tags

Tags (0)

Key Value Tag new instances

No tags

Preview code Cancel Previous Create Auto Scaling group

## STEP42: NOW CLICK ON EC2>INSTANCES TO CHECK THE NUMBER OF INSTANCES CREATED.

Instances (2) | Info

Last updated less than a minute ago | Connect | Instance state | Actions | Launch instances

|                          | Name                | Instance ID         | Instance state       | Instance type | Status check              | Alarm status               | Availability Zone | Public IP |
|--------------------------|---------------------|---------------------|----------------------|---------------|---------------------------|----------------------------|-------------------|-----------|
| <input type="checkbox"/> | i-0f8ca5d3329fb007  | i-0f8ca5d3329fb007  | <span>Running</span> | t2.micro      | <span>Initializing</span> | <span>View alarms +</span> | ap-south-1b       | -         |
| <input type="checkbox"/> | i-027d6445bd3188730 | i-027d6445bd3188730 | <span>Running</span> | t2.micro      | <span>Initializing</span> | <span>View alarms +</span> | ap-south-1a       | -         |

## STEP43: NOW TERMINATE ONE OF THE INSTANCES FROM EC2 INSTANCES

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Images, AMIs, AM Catalog, and Elastic Block Store. The main content area displays 'Instances (3) Info' with a table. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. The first instance is 'terminated' (t2.micro), while the other two are 'running' (t2.micro). Below the table is a section titled 'Select an instance'.

## STEP44: NOW GO TO AUTO-SCALING GROUPS>INSTANCE MANAGEMENT>HEALTH STATES: UNHEALTHY MEANS DELETED INSTANCE (IT WILL AUTOMATICALLY CREATE ANOTHER INSTANCE AFTER DELETION)

The screenshot shows the AWS Auto Scaling Groups page. The left sidebar includes sections for Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling (selected). The main content shows 'Auto Scaling groups (1/1) Info' for 'aditya-24mca10024-autoscaling'. It lists four instances: one terminating and unhealthy, and three others in service or pending. Below this is a detailed view of the 'aditya-24mca10024-autoscaling' group, showing the same four instances. At the bottom is a 'Lifecycle hooks (0)' section.

### NOTE:

1. A **load balancer** serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. You add one or more listeners to your load balancer.
2. A **listener** checks for connection requests from clients, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. Each rule consists of a priority, one or more actions, and one or more conditions. When the conditions for a rule are met, then its actions are performed. You must define a default rule for each listener, and you can optionally define additional rules. Before you start using your Application Load Balancer, you must add at least one listener. If your load balancer has no listeners, it can't receive traffic from clients. The rules that you define for your listeners determine how the load balancer routes requests to the targets that you register, such as EC2 instances.
3. Each **target group** routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify. You can register a target with multiple target groups. You can configure health checks on a per target group basis. Health checks are

performed on all targets registered to a target group that is specified in a listener rule for your load balancer.

4. An ***Auto Scaling group*** contains a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also lets you use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service. The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling

|                         |                                                                                  |
|-------------------------|----------------------------------------------------------------------------------|
| <b>Date:</b> 23/05/2025 | <b>Title</b>                                                                     |
| <b>Exp. No:</b> 07      | How to create EBS and attach to EC2 instance, modify size and create a snapshot. |

## **HOW TO CREATE EBS AND ATTACH TO EC2 INSTANCE, MODIFY SIZE AND CREATE A SNAPSHOT**

### **Working with EBS on EC2:**

#### **Overview –**

Amazon EBS and EC2: Creating, Attaching, Modifying, and Snapshots

Amazon Elastic Block Store (EBS) is a scalable, high-performance block storage service designed for use with Amazon EC2 instances. It enables users to create persistent volumes that remain available independently of the lifecycle of the EC2 instances they are attached to. EBS volumes behave like raw, unformatted block devices and can be formatted and mounted just like any physical disk.

Creating an EBS Volume involves provisioning storage of a specified size and type (such as gp3, gp2, or io1) in a chosen availability zone. The volume must reside in the same availability zone as the EC2 instance it will be attached to, ensuring low-latency connectivity. Once a volume is created, it can be attached to an EC2 instance, allowing the instance to use it for data storage. The attached volume can be formatted with a file system (e.g., XFS, ext4) and mounted to a directory path, making it accessible for storing application data, logs, backups, or any other data.

EBS also supports modifying volumes on the fly, which is essential for growing storage capacity as application needs evolve. You can increase the size of a volume or change its performance characteristics without detaching it or shutting down the EC2 instance. After resizing, the operating system on the instance may need to expand the file system to utilize the new space.

For data protection and disaster recovery, EBS snapshots offer a point-in-time backup mechanism. Snapshots are incremental and stored in Amazon S3, making them cost-efficient and highly durable. These snapshots can be used to restore volumes or create new ones across different availability zones or even regions, supporting scalability and resilience.

Overall, Amazon EBS provides a robust, flexible storage solution that integrates tightly with EC2, enabling high availability, easy scalability, and reliable data management for a wide range of cloud-based applications.

**STEP1: GO TO EC2 DASHBOARD>CLICK ON INSTANCES>CLICK ON LAUNCH INSTANCE>TYPE THE NAME OF THE INSTANCE>CHOOSE THE UBUNTU**

**Name and tags**

Name: aditya-24mca10024-ec2-instance

**Application and OS Images (Amazon Machine Image)**

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

Recent AMIs:

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

Browse more AMIs

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.

**Summary**

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6... [read more](#)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

[Launch instance](#) [Preview code](#)

## STEP2: CREATE A KEY PAIR

**Key pair (login)**

Key pair name - required: aditya-24mca10024-key-pair

**Network settings**

Network: vpc-0f4d0926b65ff0581

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enabled

Firewall (security groups): Create security group

**Create key pair**

Key pair name: aditya-24mca10024-key-pair

Key pair type: RSA

Private key file format: .pem

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.

**Summary**

Number of instances: 1

Firewall (security group): New 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.

[Launch instance](#) [Preview code](#)

**STEP3: EDIT THE NETWORK SETTING> SELECT THE VPC CREATED>CHOOSE THE PUBLIC SUBNET>ENABLE AUTO-ASSIGN PUBLIC IP>CREATE SECURITY GROUP  
STEP4: SET TYPE: SSH IN THE SECURITY GROUP>CLICK ON LAUNCH INSTANCE**

**Security group name - required**

Launch-wizard-4

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_/-/!@#&{<}&\_!\$^\*

**Description - required**

launch-wizard-4 created 2025-05-22T09:47:47.044Z

**Inbound Security Group Rules**

Security group rule 1 (TCP, 22, 0.0.0.0/0)

| Type     | Protocol | Port range | Action                  |
|----------|----------|------------|-------------------------|
| ssh      | TCP      | 22         | Remove                  |
| Anywhere |          |            | Add security group rule |

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

**Advanced network configuration**

[CloudShell](#) [Feedback](#)

**Summary**

Number of instances: 1

Firewall (security group): New 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.

[Launch instance](#) [Preview code](#)

## STEP5: CLICK ON THE EC2 INSTANCE CREATED>CLICK ON CONNECT

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store (Volumes, Snapshots). The main content area displays the instance summary for 'i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)'. Key details shown include:

- Instance ID:** i-06f102b3c4b6e4d95
- IPv6 address:** -
- Hostname type:** IP name: ip-12-0-1-55.ap-south-1.compute.internal
- Answer private resource DNS name:** -
- Auto-assigned IP address:** 13.201.94.9 [Public IP]
- IAM Role:** -
- IMDSv2:** Required
- Public IPv4 address:** 13.201.94.9 [open address]
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-12-0-1-55.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-058c4d3777ffd617c (aditya-24mca10024-vpc)
- Subnet ID:** subnet-05e65f90dc108f428 (Public\_Subnet)
- Instance ARN:** arn:aws:ec2:ap-south-1:270543921719:instance/i-06f102b3c4b6e4d95
- Private IPv4 addresses:** 12.0.1.55
- Public DNS:** -
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendation
- Auto Scaling Group name:** -
- Managed:** false

At the bottom, there are links for CloudShell and Feedback, and a footer with copyright information.

## STEP6: CLICK ON EC2 INSTANCE CONNECT

The screenshot shows the 'Connect to instance' dialog box. At the top, it says 'EC2 > Instances > i-06f102b3c4b6e4d95 > Connect to instance'. Below that, it says 'Connect to instance' with a link to 'Info'. It states: 'Connect to your instance i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance) using any of these options'.

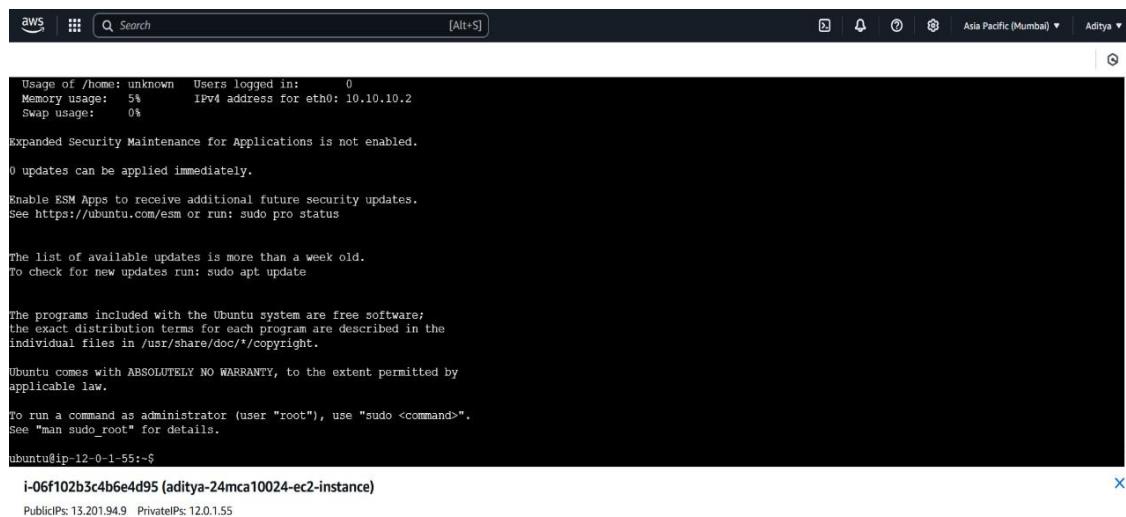
The dialog has four tabs at the top: 'EC2 Instance Connect' (selected), 'Session Manager', 'SSH client', and 'EC2 serial console'.

The 'EC2 Instance Connect' tab contains the following fields:

- Instance ID:** i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)
- Connect using EC2 Instance Connect:** (radio button selected) Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
- Public IPv4 address:** 13.201.94.9
- IPv6 address:** -
- Username:** ubuntu (input field)
- Note:** 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.'

On the right side of the dialog, there are 'Cancel' and 'Connect' buttons.

## STEP7: EC2 INSTANCE IS CONNECTED



A screenshot of a terminal window titled "aws" with a search bar and a "Search [Alt+S]" button. The window shows system status: "Usage of /home: unknown Users logged in: 0", "Memory usage: 5% IPV4 address for eth0: 10.10.10.2", and "Swap usage: 0%". It also displays security maintenance information, including ESM Apps status and update availability. The terminal prompt is "ubuntu@ip-12-0-1-55:~\$". The session is identified as "i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)" with public and private IP addresses listed.

**STEP 8: TYPE “lsblk” COMMAND IN THE TERMINAL. IT WILL SHOW YOU HOW MANY VOLUMES ARE ATTACHED TO THE EC2 INSTANCE. IT SHOWS THE DEFAULT 8G VOLUME WHICH WE SET WHILE CREATING EC2 INSTANCE.**



A screenshot of a terminal window showing the output of the "lsblk" command. The output lists various block devices, including loop0, loop1, loop2, xvda, and xvda1 through xvda16. The xvda device is mounted at "/". The xvda16 device is mounted at "/boot". The terminal prompt is "ubuntu@ip-12-0-1-55:~\$". The session is identified as "i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)".

## CREATION OF EBS VOLUME

### STEP 9: NOW CREATE EBS VOLUME FROM EC2 DASHBOARD: ELASTIC BLOCK STORE>VOLUMES>CREATE VOLUME

The screenshot shows the AWS EC2 Volumes page. On the left, there's a navigation sidebar with options like Instances, Images, Elastic Block Store (selected), and Network & Security. The main area displays a table titled 'Volumes (1) Info' with one row. The row details a volume named 'vol-068f9715fc48b89d' of type 'gp3' with a size of 8 GiB, IOPS of 3000, Throughput of 125, and a Snapshot ID of 'snap-00a5570...'. A 'Create volume' button is visible at the top right of the table. Below the table, a section titled 'Fault tolerance for all volumes in this Region' shows '0 / 1' recently backed up volumes.

### STEP 9: YOU CAN CHOOSE ANY OF THE VOLUME TYPES FROM THE DROP-DOWN >SELECT gp3

The screenshot shows the 'Create volume' wizard step. The 'Volume settings' section is open, showing a dropdown for 'Volume type' with 'General Purpose SSD (gp3)' selected. Other options listed include 'General Purpose SSD (gp3)', 'General Purpose SSD (gp2)', 'Provisioned IOPS SSD (io1)', 'Provisioned IOPS SSD (io2)', 'Cold HDD (sc1)', 'Throughput Optimized HDD (st1)', and 'Magnetic (standard)'. Below this, a 'Throughput (MiB/s)' input field is set to 125. An 'Availability Zone' dropdown shows 'ap-south-1a' selected. At the bottom, there are 'CloudShell' and 'Feedback' links, along with standard copyright and legal links.

## STEP 10: KEEP THE DATA AS SHOWN

Volume settings

Volume type: General Purpose SSD (gp3)

Size (GiB): 100

IOPS: 3000

Throughput (MiB/s): 125

Availability Zone: ap-south-1a

Snapshot ID - optional: Don't create volume from a snapshot

## STEP 11: CLICK ON CREATE VOLUME.

Encryption: Encrypt this volume

Tags - optional: Add tag

Snapshot summary: Click refresh to view backup information

Create volume

## STEP 12: EDIT THE VOLUME NAME BY CLICKING ON IT>CLICK ON THE VOLUME THAT YOU HAVE CREATED

Successfully created volume vol-0ca109e15e2f290d1.

Volumes (1/2)

| Name                  | Volume ID | Type    | Size | IOPS | Throughput      | Snapshot ID               | Created |
|-----------------------|-----------|---------|------|------|-----------------|---------------------------|---------|
| vol-0ca109e15e2f290d1 | gp3       | 100 GiB | 3000 | 125  | -               | 2025/05/22 15:24 GMT+5:30 |         |
| vol-068f97157fc48b09d | gp3       | 8 GiB   | 3000 | 125  | snap-00a5570... | 2025/05/22 15:21 GMT+5:30 |         |

Volume ID: vol-0ca109e15e2f290d1

Details: Volume ID vol-0ca109e15e2f290d1, Size 100 GiB, Type gp3, Status check Okay, Throughput 125, Created May 22, 2025 15:24:47 GMT+0530 (India Standard Time).

## STEP 13: CLICK ON ACTIONS>ATTACH VOLUME

The screenshot shows the AWS EC2 Volumes page. A volume named 'vol-0ca109e15e2f290d1' is selected. The 'Actions' dropdown menu is open, and 'Attach volume' is highlighted. Other options in the menu include 'Create snapshot', 'Detach volume', 'Force detach volume', and 'Manage auto-enabled I/O'.

## STEP 14: SELECT INSTANCE THAT YOU HAVE CREATED>CHOOSE THE DEVICE:/dev/sdk>CLICK ON ATTACH VOLUME

The screenshot shows the 'Attach volume' dialog box. It displays the 'Basic details' section with the Volume ID 'vol-0ca109e15e2f290d1' and the Availability Zone 'ap-south-1a'. Below this, an 'Instance' dropdown is set to 'i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance) (running)'. A note states: 'Only instances in the same Availability Zone as the selected volume are displayed.' Under 'Device name', the value '/dev/sdk' is selected. A note below says: 'Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdk through /dev/sdp.' At the bottom are 'Cancel' and 'Attach volume' buttons.

## STEP 15: NOW GO TO EC2 TERMINAL AND TYPE:lsblk>IT WILL LIST ALL THE VOLUME(100G IS SHOWN BELOW)

```
ubuntu@ip-12-0-1-55:~$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
loop0 7:0 0 26.3M 1 loop /snap/amazon-ssm-agent/9081
loop1 7:1 0 73.9M 1 loop /snap/core22/1748
loop2 7:2 0 44.4M 1 loop /snap/snappyd/23545
xvda 202:0 0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
xvda14 202:14 0 4M 0 part
xvda15 202:15 0 106M 0 part /boot/efi
└─xvda16 202:16 0 913M 0 part /boot
ubuntu@ip-12-0-1-55:~$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
loop0 7:0 0 26.3M 1 loop /snap/amazon-ssm-agent/9081
loop1 7:1 0 73.9M 1 loop /snap/core22/1748
loop2 7:2 0 44.4M 1 loop /snap/snappyd/23545
xvda 202:0 0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
└─xvda14 202:14 0 4M 0 part
└─xvda15 202:15 0 106M 0 part /boot/efi
└─xvda16 202:16 0 913M 0 part /boot
xvdk 202:160 0 100G 0 disk
```

## STEP 16: TYPE sudo fdisk -l (IT WILL LIST OUT ALL THE DISK PARTITION)

```

applicable law.

To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details.

ubuntu@ip-12-0-1-55:~$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
loop0 7:0 0 26.3M 1 loop /snap/amazon-ssm-agent/9881
loop1 7:1 0 73.9M 1 loop /snap/core22/1740
loop2 7:2 0 44.4M 1 loop /snap/snapd/3545
rwd0 202:0 0 8G 0 disk
└─xvdal 202:1 0 7G 0 part /
└─xvdal1 202:1:0 0 4M 0 part
└─xvdal5 202:1:5 0 106M 0 part /boot/efi
└─xvdal6 202:1:6 0 913M 0 part /boot
rwdk 202:160 0 100G 0 disk
ubuntu@ip-12-0-1-55:~$

```

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)  
PublicIPs: 13.201.94.9 PrivateIPs: 12.0.1.55

## STEP 17: TYPE sudo file -s /dev/xvdk (IT WILL SHOW “DATA” WHICH MEANS THERE IS NO FILE SYSTEM CREATED FOR THIS EBS VOLUME)

```

mount@ip-12-0-1-55:~$
ubuntu@ip-12-0-1-55:~$ sudo file -s /dev/xvdk
/dev/xvdk: data
mount@ip-12-0-1-55:~$

```

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)  
PublicIPs: 13.201.94.9 PrivateIPs: 12.0.1.55

## STEP 18: WE NEED TO CREATE FILE SYSTEM FOR EBS VOLUME: TYPE sudo mkfs -t xfs /dev/xvdk (FILE SYSTEM CREATED)

```

/dev/xvdk: data
ubuntu@ip-12-0-1-55:~$ sudo mkfs -t xfs /dev/xvdk
meta-data=/dev/xvdk isize=512 agcount=4, agsize=6553600 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=1 finobt=1, sparse=1, rmapbt=1
 = reflink=1 bigtime=1 inobtcount=1 nrext64=0
data = bsize=4096 blocks=26214400, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0, ftype=1
log =internal log bsize=4096 blocks=16384, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096
ubuntu@ip-12-0-1-55:~$

```

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)  
PublicIPs: 13.201.94.9 PrivateIPs: 12.0.1.55

## STEP 19: NOW AGAIN TYPE sudo file -s /dev/xvdk TO CHECK THE FILE SYSTEM (ITS SHIWS EBS VOLUME HAS xfs FILESYSTEM)

```

ubuntu@ip-12-0-1-55:~$ sudo file -s /dev/xvdk
/dev/xvdk: SGT XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
ubuntu@ip-12-0-1-55:~$

```

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)  
PublicIPs: 13.201.94.9 PrivateIPs: 12.0.1.55

## STEP 20: CREATE A DIRECTORY:

- TYPE sudo mkdir /munsifaebsvol
- NEXT TYPE sudo mount /dev/xvdk / munsifaebsvol
- TYPE ls -lart /munsifaebsvol

```
ubuntu@ip-12-0-1-55:~$ sudo mkdir /aditya24mca10024
ubuntu@ip-12-0-1-55:~$ sudo mount /dev/xvdk / aditya24mca10024
mount: bad usage.
Try 'mount --help' for more information.
ubuntu@ip-12-0-1-55:~$ ls -lart /aditya24mca10024
total 0
drwxr-xr-x 23 root root 4096 May 22 10:04 ..
drwxr-xr-x 2 root root 4096 May 22 10:04 .
ubuntu@ip-12-0-1-55:~$
```

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)

PublicIPs: 13.201.94.9 PrivateIPs: 12.0.1.55

## STEP 21: NOW TYPE df -h (IT WILL SHOW THAT EBS VOLUME HAS BEEN MOUNTED TO DIRECTORY "munsifaebsvol")

```
ubuntu@ip-12-0-1-55:~$ df -h
Filesystem Size Used Avail Use% Mounted on
/dev/root 6.8G 1.7G 5.1G 26% /
tmpfs 479M 0 479M 0% /dev/shm
tmpfs 1.92M 872K 1.91M 1% /run
tmpfs 5.0M 0 5.0M 0% /run/lock
/dev/xvda16 881M 79M 741M 10% /boot
/dev/xvda15 1.05M 6.1M 99M 6% /boot/efi
tmpfs 96M 12K 96M 1% /run/user/1000
ubuntu@ip-12-0-1-55:~$
```

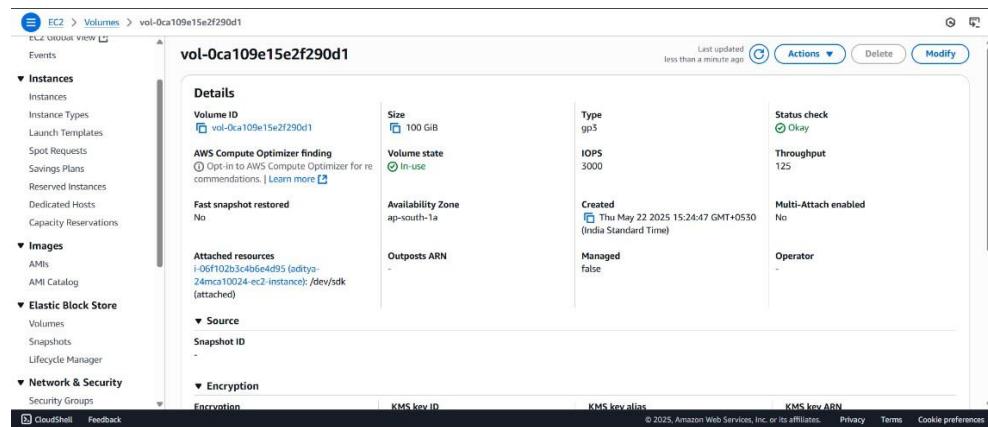
i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)

PublicIPs: 13.201.94.9 PrivateIPs: 12.0.1.55

## MODIFY VOLUME SIZE

## STEP 22: NOW NEXT GO TO EBS VOLUME TO INCREASE THE VOLUME(NOTE THAT EBS VOLUME CAN BE INCREASED BUT YOU CANNOT REDUCE THE VOLUME SIZE ONCE AN EBS VOLUME IS CREATED BECAUSE THERE IS A PROBABILITY OF DATA LOSS)

- CLICK ON EBS VOLUME>SELECT THE VOLUME THAT YOU HAVE CREATED>CLICK ON MODIFY
- INCREASE THE VOLUME SIZE FROM 100 TO 105>CLICK MODIFY



☰ EC2 > Volumes > vol-0f813db80abfc9414 > Modify volume

### Modify volume Info

Modify the type, size, and performance of an EBS volume.

---

**Volume details**

**Volume ID**  
vol-0f813db80abfc9414 (EBS volume 100G)

**Volume type** | [Info](#)  
General Purpose SSD (gp3)

**Size (GiB)** | [Info](#)  
105

Min: 1 GiB, Max: 16384 GiB.

**IOPS** | [Info](#)  
3000

Min: 3000 IOPS, Max: 16000 IOPS.

**Throughput (MiB/s)** | [Info](#)  
125

Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

**Modify vol-0f813db80abfc9414?**

If you are increasing the size of the volume, you must extend the file system to the new size of the volume. You can only do this when the volume enters the optimizing state. For more information see [Extend the file system after resizing an EBS volume](#).

The modification might take a few minutes to complete.

You are charged for the new volume configuration after volume modification starts. For pricing information, see [Amazon EBS Pricing](#).

Are you sure that you want to modify vol-0f813db80abfc9414?

[Cancel](#) [Modify](#)

## STEP 23: NOW GO TO THE EC2 TERMINAL AND VERIFY WHETHER THE DISK SIZE HAS CHANGED OR NOT. (TYPE sudo fdisk -l)

```
disk /dev/loop2: 44.44 MiB, 46596096 bytes, 91008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

disk /dev/vvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E3470801-32E3-4FC2-0879-1BCCDE9c2D7

Device Start End Sectors Size Type
/vvda/vvda1 2093200 16777182 14677983 7.5G Linux filesystem
/vvda/vvda14 2048 10239 8192 4M BIOS boot
/vvda/vvda15 10240 227327 217088 10.0M EFI System
/vvda/vvda16 227328 2097152 1869825 91.3M Linux extended boot

Partition table entries are not in disk order.

disk /dev/vvdk: 105 GiB, 112742891520 bytes, 220200960 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)
```

## STEP 24: NOW IN THE EC2 TERMINAL GO TO EBS VOLUME DIRECTORY AND CREATE FILES.

- a. TYPE cd /mnt/sda1 (GO TO EBS VOLUME DIRECTORY)
- b. TYPE sudo touch 1.txt (CREATE FILE1)
- c. TYPE sudo touch 2.txt (CREATE FILE2)
- d. TYPE ls -lart (TO VERIFY WHETHER FILES ARE THERE OR NOT)

```
ubuntu@ip-12-0-1-55:~$ cd /mnt/sda1
ubuntu@ip-12-0-1-55:/mnt/sda1$ aditya-24mca10024$ sudo touch 1.txt
ubuntu@ip-12-0-1-55:/mnt/sda1$ aditya-24mca10024$ sudo touch 2.txt
ubuntu@ip-12-0-1-55:/mnt/sda1$ aditya-24mca10024$ ls -lart
total 8
drwxr-xr-x 23 root root 4096 May 22 10:04 ..
-rw-r--r-- 1 root root 0 May 22 10:10 1.txt
-rw-r--r-- 1 root root 0 May 22 10:10 2.txt
drwxr-xr-x 2 root root 4096 May 22 10:10 .

i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance)
PublicIP: 13.201.94.9 PrivateIP: 12.0.1.55
```

## EC2 SNAPSHOT CREATION

**STEP 24: CREATE SECOND EC2 INSTANCE (CREATE IN A SIMILAR MANNER LIKE YOU HAVE CREATED FOR EC1 )>CONNECT EC2 INSTANCE. THEN CHECK THE DISK SIZE OF EC2 INSTANCE (TYPE sudo fdisk -l) IT SHOWS THE DEFAULT 8G**

```
disk /dev/loop1: 73.89 MiB, 77479936 bytes, 151328 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 44.44 MiB, 4656096 bytes, 91008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

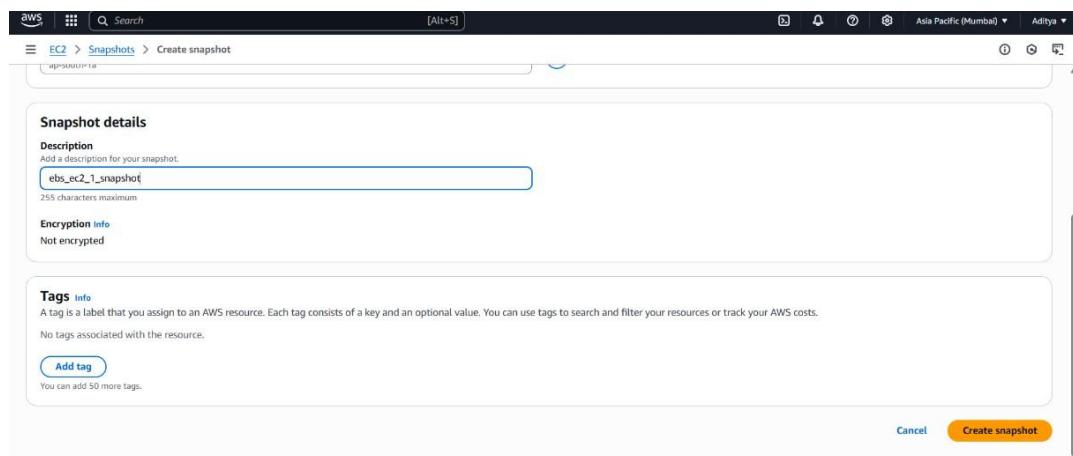
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E3478E01-32E3-4FC2-8E79-1RCCE89C2D7

Device Start End Sectors Size Type
/dev/xvda1 209200 16777182 14677983 7G Linux filesystem
/dev/xvda4 2048 10239 8192 4M BIOS boot
/dev/xvda5 10240 227327 217088 106M EFI System
/dev/xvda6 227328 2097152 1869825 913M Linux extended boot

Partition table entries are not in disk order.
ubuntu@ip-12-0-1-234:~
```

**STEP 25: NOW TO CREATE A SNAPSHOT GO TO EC2 DASHBOARD:**

- a. **EBS VOLUME>CLICK ON THE VOLUME THAT YOU HAVE CREATED>CLICK ON ACTIONS>CLICK ON CREATE SNAPSHOT**
- b. **TYPE THE DESCRIPTION>CLICK ON CREATE A SNAPSHOT**



**STEP 26: NOW GO TO SNAPSHOT>CLICK ON THE SNAPSHOT THAT YOU HAVE CREATED>CLICK ON ACTIONS>CREATE VOLUME FROM SNAPSHOT**

**Details**

- Snapshot ID:** snap-00d6b3c15d3b48027
- Owner:** 270543921719
- Description:** ebs\_ec2\_1\_snapshot
- Source volume:** Volume ID: vol-0ca109e15e2f290d1, Volume size: 105 GiB
- Encryption:** Encryption: Not encrypted, KMS key ID: -, KMS key alias: -, KMS key ARN: -

**Actions**

- Create volume from snapshot
- Create image from snapshot
- Copy snapshot
- Snapshot settings
- Fast snap
- Archiving

## STEP 27: KEEP THE DATA SAME>CLICK ON TAG>GIVE A NAME>CLICK ON CREATE SNAPSHOT.

**Tags - optional**

| Key                   | Value - optional  |
|-----------------------|-------------------|
| aditya-24mca10024-tag | ebs-from-snapshot |

**Snapshot summary**

Click refresh to view backup information  
The volume type that you select and the tags that you assign determine whether the volume will be backed up by any Data Lifecycle Manager policies.

**Create volume**

## STEP 28: NOW GO TO VOLUME>MODIFY THE VOLUME NAME

**Volumes (1/4) Info**

| Name     | Volume ID                            | Type | Size    | IOPS | Throughput | Snapshot ID     | Created                    |
|----------|--------------------------------------|------|---------|------|------------|-----------------|----------------------------|
| snapshot | vol-039d8439cd0ee14e8 (EBS Snapshot) | gp3  | 105 GiB | 3000 | 125        | snap-00a5570... | 2025/05/22 15:43 GMT+5:... |
|          | vol-0ca109e15e2f290d1                | gp3  | 105 GiB | 3000 | 125        | -               | 2025/05/22 15:24 GMT+5:... |
|          | vol-068f97157fc48b89d                | gp3  | 8 GiB   | 3000 | 125        | snap-00a5570... | 2025/05/22 15:21 GMT+5:... |

**Volume ID: vol-039d8439cd0ee14e8 (EBS Snapshot)**

| Details                                                                                          | Status checks                  | Monitoring                                 | Tags                     |
|--------------------------------------------------------------------------------------------------|--------------------------------|--------------------------------------------|--------------------------|
| Volume ID: vol-039d8439cd0ee14e8 (EBS Snapshot)                                                  | Size: 105 GiB                  | Type: gp3                                  | Status check: Okay       |
| AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.   Learn more | Volume state: Available        | IOPS: 3000                                 | Throughput: 125          |
| Fast snapshot restored: No                                                                       | Availability Zone: ap-south-1a | Created: Thu May 22 2025 15:49:53 GMT+5:30 | Multi-Attach enabled: No |

## STEP 29: CLICK ON THE VOLUME>CLICK ON ACTION>CLICK ON ATTACH VOLUME

The screenshot shows the AWS EC2 Volumes page. On the left, there's a sidebar with various navigation options like Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, CloudWatch Metrics, CloudWatch Metrics Insights, CloudShell, and Feedback. The main panel displays the details of an EBS Snapshot named 'vol-039d8439cd0ee14e8'. The 'Actions' button in the top right has a dropdown menu with options: Create snapshot, Attach volume, Detach volume, Force detach volume, Manage auto-enabled I/O, Throughput, Multi-Attach enabled, and Operator. The 'Attach volume' option is currently selected.

## STEP 30: SELECT EC2 INSTANCE>SELECT sdk IN DEVICE>CLICK ON ATTACH VOLUME

This screenshot shows the 'Attach volume' dialog box. In the 'Basic details' section, the 'Instance' dropdown is set to 'i-00d1e074123eb8a21 (aditya-24mca10024-ec2-instance2) (running)'. The 'Device name' dropdown is set to '/dev/sdk'. A note at the bottom states: 'Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.' At the bottom right are 'Cancel' and 'Attach volume' buttons.

## STEP 31: GO TO EC2 TERMINAL>TYPE sudo fdisk -l TO CHECK WHETHER THE NEW VOLUME IS ADDED OR NOT. 105G IS ADDED AS SHOWN BELOW

```
Disk /dev/loop2: 44.44 MB, 46596096 bytes, 91008 sectors
Unit: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E3478B01-3E23-4FC2-8E79-1BC0DDE89C2D7

Device Start End Sectors Size Type
/dev/xvdd1 20952000 16777216 1467512 8G filesystem
/dev/xvdd14 10240 10239 1 512 BIOS boot
/dev/xvdd15 10240 227327 217088 104M EFI System
/dev/xvdd16 227328 2097152 1869825 913M Linux extended boot

Partition table entries are not in disk order.

Disk /dev/xvd: 105 GB, 112742091520 bytes, 220200960 sectors
Unit: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
ubuntu@ip-12-0-1-234:~$ i-00d1e074123eb8a21 (aditya-24mca10024-ec2-instance2)
PublicIP: 13.234.117.58 PrivateIP: 12.0.1.234
```

**STEP 32: NOW CHECK WHETHER FILE SYSTEM IS ATTACHED OR NOT. TYPE sudo file -s /dev/xvdk**

```
Disk /dev/xvdk: 105 GiB, 12742891520 bytes, 220206960 sectors
 512-byte logical sector size; 512 bytes / sector
I/O size (minimum/optimal): 512 bytes / 512 bytes
dbsn@ip-12-0-1-234:~$ sudo file -s /dev/xvdk
/dev/xvdk: data (file offset 0x0, inodes 512, v2 dirs)
 512-byte logical sector size; 512 bytes / sector
i-00d1e074123eb8a21 (aditya-24mca10024-ec2-instance2)
```

**STEP 33: NOW CREATE A DIRECTORY:**

- TYPE sudo mkdir /munsifaebsvol2**
- MOUNT THE FILE, TYPE sudo mount /dev/xvdk /munsifaebsvol2**
- TYPE ls -lart /munsifaebsvol2**

```
ubuntu@ip-12-0-1-234:~$ sudo mkdir /aditya24mca10024
mkdir: cannot create directory '/aditya24mca10024': File exists
ubuntu@ip-12-0-1-234:~$ sudo mount /dev/xvdk /aditya24mca10024
ubuntu@ip-12-0-1-234:~$ ls -lart /aditya24mca10024
total 4
drwxr-xr-x 2 root root 6 May 22 10:01 .
drwxr-xr-x 23 root root 4096 May 22 10:23 ..
ubuntu@ip-12-0-1-234:~$ i-00d1e074123eb8a21 (aditya-24mca10024-ec2-instance2)
```

**STEP 34: NOW YOU CAN CHECK IN BOTH EC2 INSTANCES, FILE1 AND FILE2 IS PRESENT**

- TYPE ls (EC2 1)**
- TYPE cd \munsifaebsvol2 > ls (EC2 2)**

```
ubuntu@ip-12-0-1-55:~$ cd /aditya24mca10024
ubuntu@ip-12-0-1-55:~$ aditya24mca10024$ sudo touch 1.txt
ubuntu@ip-12-0-1-55:~$ aditya24mca10024$ sudo touch 2.txt
ubuntu@ip-12-0-1-55:~$ aditya24mca10024$ ls -lart
total 8
drwxr-xr-x 23 root root 4096 May 22 10:04 ..
-rw-r--r-- 1 root root 0 May 22 10:10 1.txt
-rw-r--r-- 1 root root 0 May 22 10:10 2.txt
drwxr-xr-x 2 root root 4096 May 22 10:10 .
ubuntu@ip-12-0-1-55:~$ aditya24mca10024$ i-06f102b3c4b6e4d95 (aditya-24mca10024-ec2-instance2)
```

**STEP 35: IF YOU CREATE ANOTHER FILE IN FIRST EC2 INSTANCE, IT WILL NOT REFLECT IN THE SECOND EC2 INSTANCE.**

- TYPE sudo touch 3.txt in FIRST EC2 INSTANCE**
- CHECK WITH ls (FIRST EC2 INSTANCE-IT WILL DISPLAY ALL THE THREE TEXT FILES.)**
- TYPE ls in SECOND EC2 INSTANCE (IT WILL ONLY DISPLAY TWO TEXT FILES)**