



K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

11/28/2025

Cloud Security Fundamentals

Practical File ENCS-353
B. Tech CSE (Spec. Cyber Security)

Index

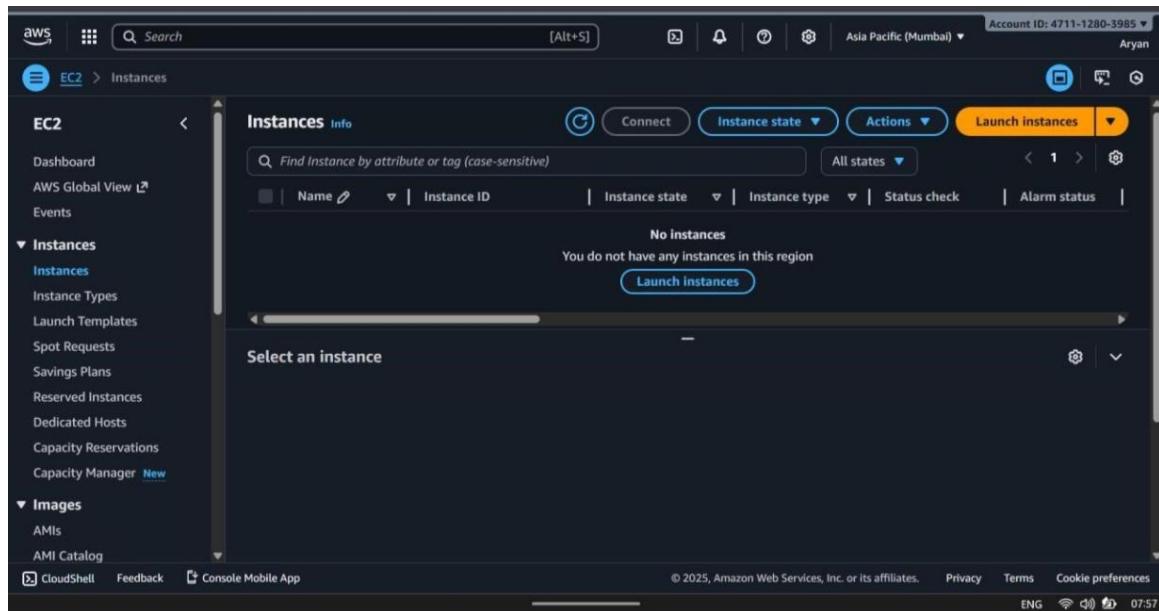
S. No.	Name of Practical	Sign
1	To launch an EC2 instance in AWS and connect to it using SSH.	
2	To create an Elastic Block Store (EBS) volume in AWS and attach it to an EC2 instance.	
3	To create and configure an Amazon S3 bucket in AWS Management Console.	
4	To create an AWS Lambda function and test its execution in AWS Management Console.	
5	To Create an IAM User in AWS	
6	To secure an AWS IAM user account by enabling Multi-Factor Authentication (MFA) .	
7	To create a user group in AWS IAM to manage permissions collectively for multiple users who share similar roles or responsibilities.	
8	To create a security role in AWS IAM for providing secure and temporary access to specific AWS resources.	
9	To Create a Virtual Private Cloud.	

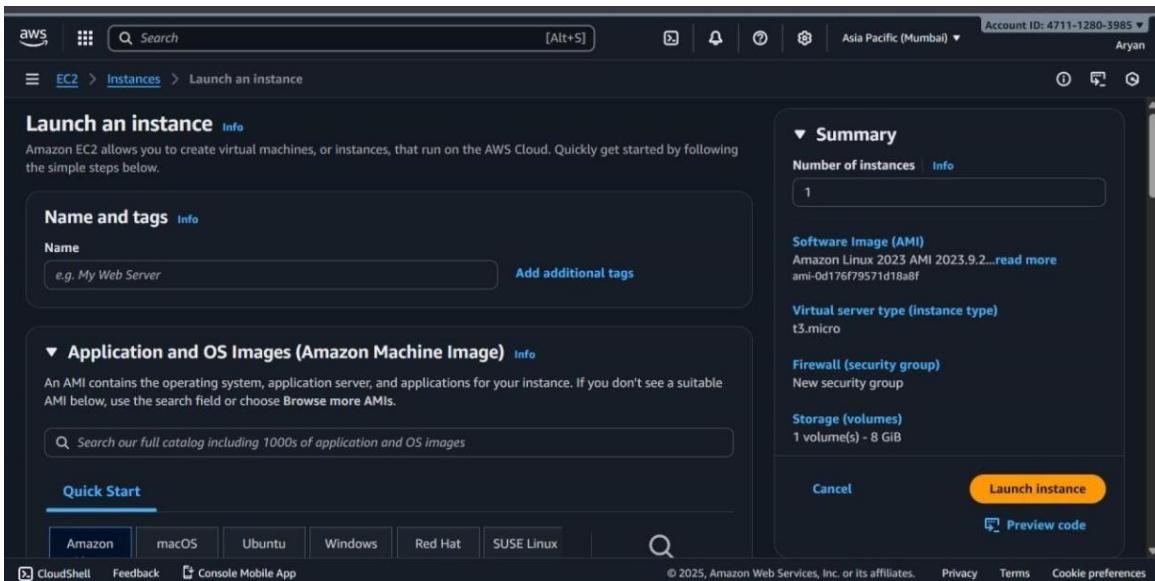
PRACTICAL-1

OBJECTIVE - TO CREATE EC2 INSTANCE

Steps to Create an EC2 Instance

1. **Login to AWS** ○ Go to AWS Management Console.
 - Sign in with your credentials.
2. **Navigate to EC2** ○ In the search bar, type **EC2** and select it.
3. **Launch Instance** ○ Click **Instances** → **Launch Instance**.
 - Give your instance a **name**.





4. Choose Amazon Machine Image (AMI)

- Select an OS image (e.g., Amazon Linux 2, Ubuntu, Windows).
- For beginners, choose **Amazon Linux 2 (Free tier eligible)**.

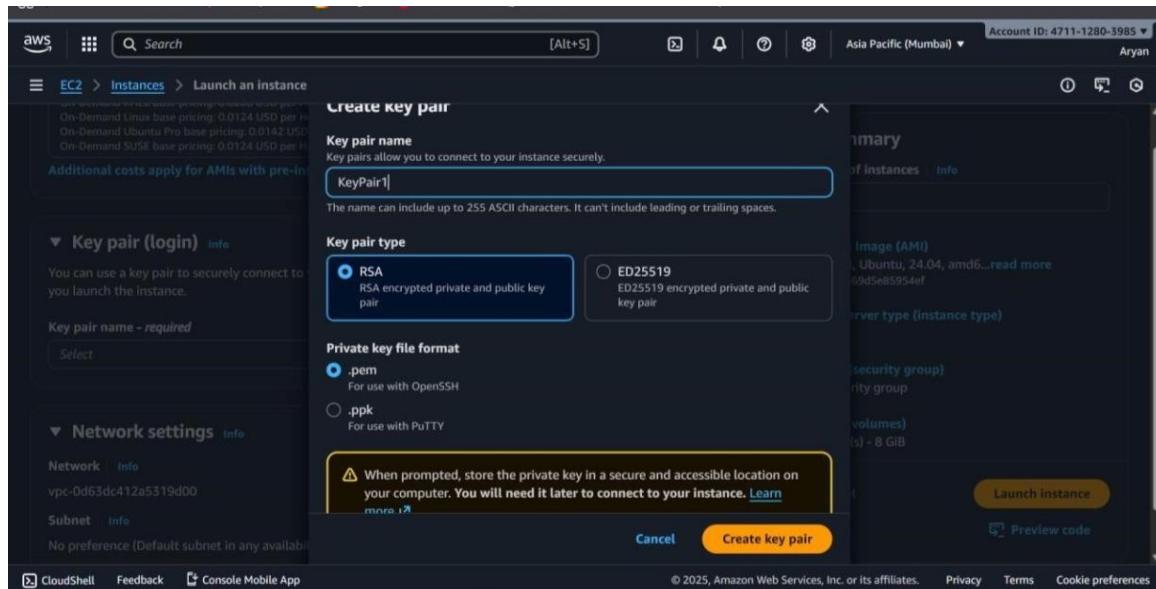
I chose **Ubuntu**.

5. Choose Instance Type

- Select **t2.micro** (free tier eligible).

6. Configure Key Pair

- Create a new key pair (RSA) and download the **.pem** file.



This will be used later to connect to your instance.

- 7. Configure Network & Security** ○ Leave default **VPC and subnet** (unless you want custom networking).
- Under **Security group**, allow:
 - **SSH (port 22)** for Linux
 - **RDP (port 3389)** for Windows
 - **HTTP/HTTPS** if you're hosting a website.
- 8. Configure Storage** ○ Default 8 GB is fine (increase if needed).
- 9. Launch Instance** ○ Click **Launch Instance**.
- AWS will start your instance.

The screenshot shows two consecutive pages of the AWS EC2 'Launch an instance' wizard.

Page 1: Configure storage

- Storage Configuration:** 1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted.
- Advanced Options:** Click refresh to view backup information. Tags determine backup by Data Lifecycle Manager policies.
- Summary:** Number of instances: 1. Software image (AMI): Canonical, Ubuntu, 24.04, amd64...ami-02b8269d5e85954ef. Virtual server type (instance type): t2.micro. Firewall (security group): New security group. Storage (volumes): 1 volume(s) - 8 GiB.
- Buttons:** Cancel, Launch instance, Preview code.

Page 2: Success

Success message: Successfully initiated launch of instance (i-062d4052b9682d105).

Next Steps:

- Create billing usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. Create billing alerts.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Connect to instance, Learn more.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Connect an RDS database, Create a new RDS database.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Create EBS snapshot policy.

Screenshot of the AWS EC2 Instances page and the Connect to instance dialog.

EC2 Instances Page:

- Account ID: 4711-1280-3985
- Region: Asia Pacific (Mumbai)
- User: Aryan
- Instances (1) Info: MyEC2Instance, i-062d4052b9682d105, Running, t2.micro, Initializing, View alarms +
- Actions: Connect, Instance state, Actions, Launch instances
- Search bar: Find Instance by attribute or tag (case-sensitive)
- Filter: All states
- Table Headers: Name, Instance ID, Instance state, Instance type, Status check, Alarm status
- Table Data: MyEC2Instance, i-062d4052b9682d105, Running, t2.micro, Initializing
- Select an instance: MyEC2Instance

Connect to instance Dialog:

- EC2 Instance Connect tab is selected.
- Session Manager, SSH client, EC2 serial console tabs are available.
- Instance ID: i-062d4052b9682d105 (MyEC2Instance)
- Connection type:
 - Connect using a Public IP: Connect using a public IPv4 or IPv6 address.
 - Connect using a Private IP: Connect using a private IP address and a VPC endpoint.
- Public IPv4 address: 13.233.153.165
- IPv6 address: (disabled)
- Username: 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.
- Buttons: Cancel, Connect

AWS Search [Alt+S] Asia Pacific (Mumbai) Account ID: 4711-1280-3985 Aryan

* Support: <https://ubuntu.com/pro>

System information as of Fri Nov 28 02:41:10 UTC 2025

```
System load: 0.31 Processes: 110
Usage of /: 25.8% of 6.71GB Users logged in: 0
Memory usage: 21% IPv4 address for enX0: 172.31.6.85
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-172-31-6-85:~$ sudo su
root@ip-172-31-6-85:/home/ubuntu# ~
```

i-062d4052b9682d105 (MyEC2Instance)

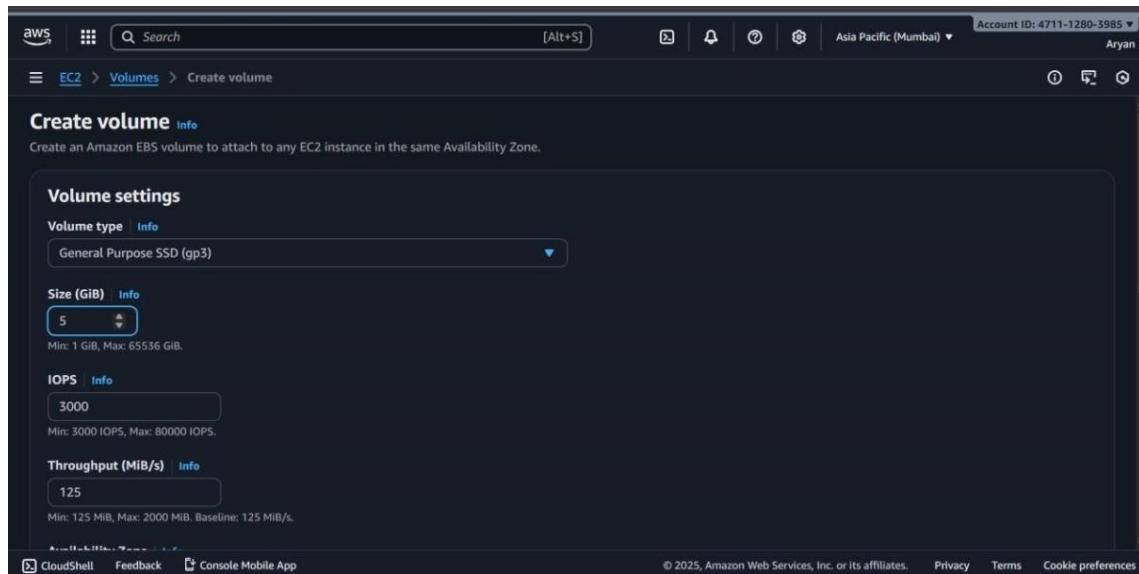
Public IPs: 13.233.153.163 PrivateIP: 172.31.6.85

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

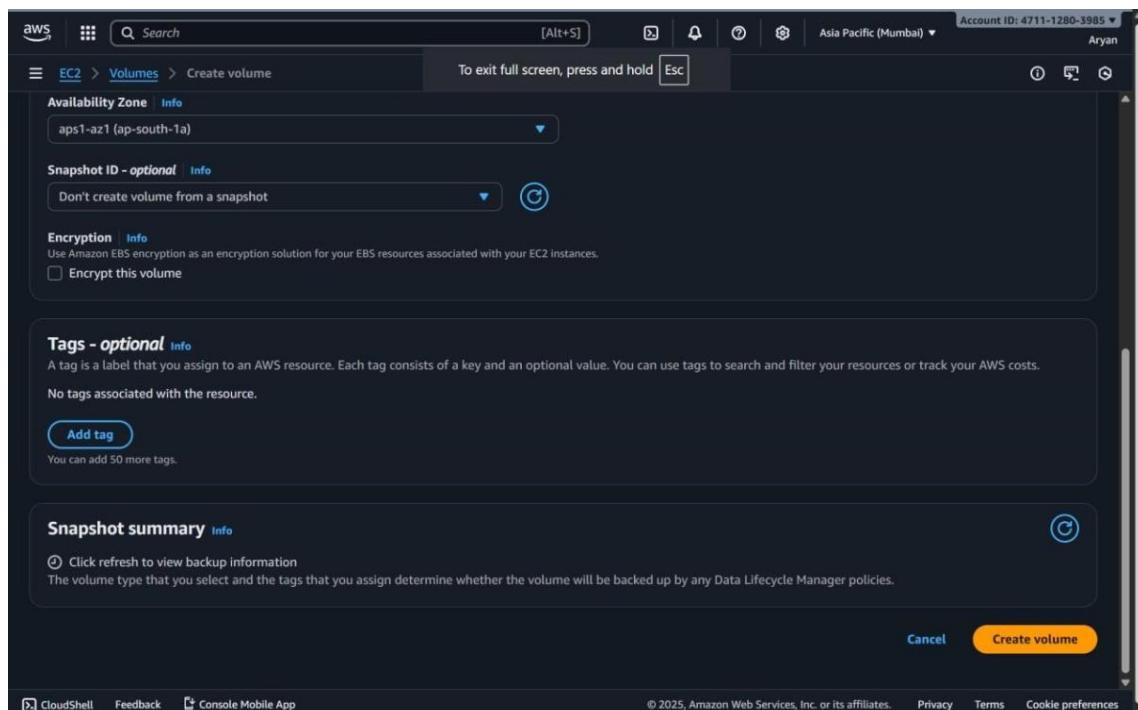
Practical:- 2

Objective:- Create Amazon Elastic Block Store (EBS) volumes in AWS

Step1:- Search EC2 and scroll to volumes.



Step 2:- Click on create volume and choose the settings as per the requirements. (here we have chosen gp3 -> size=5 and rest are default.



Step 3:- Keep the availability zone as the required zone in other activities too.

The screenshot shows the AWS Management Console with the following details:

- EC2 > Volumes > vol-05cd0b9b6dabcb4d1 > Attach volume**
- Attach volume** (Info)
- Attach a volume to an instance to use it as you would a regular physical hard disk drive.
- Basic details** section:
 - Volume ID: vol-05cd0b9b6dabcb4d1
 - Availability Zone: ap-south-1a
 - Instance: i-00122ecfde1397844 (mywebserver) (running)
 - Device name: /dev/xvda
- A note at the bottom left 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/sdf".
- Volumes (2) Info** (Last updated 1 minute ago)

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID
vol-0e434cccb9e8cbf92	gp3	8 GiB	3000	125	-	snap-0a07784...
vol-0c08d2f9a05266840	gp3	5 GiB	3000	125	-	-
- Fault tolerance for all volumes in this Region**
- Snapshot summary** (Last updated on Fri, Nov 28, 2025, 08:17:19 AM (GMT+05:30))

Recently backed up volumes / Total # volumes	0 / 0
--	-------
- Data Lifecycle Manager default policy for EBS Snapshots status**: No default policy set up | Create policy

Newly created volume is marked by the red colour.

PRACTICAL -3

OBJECTIVE- CREATE AN S3 BUCKET IN AWS

1. Log in to AWS Management Console

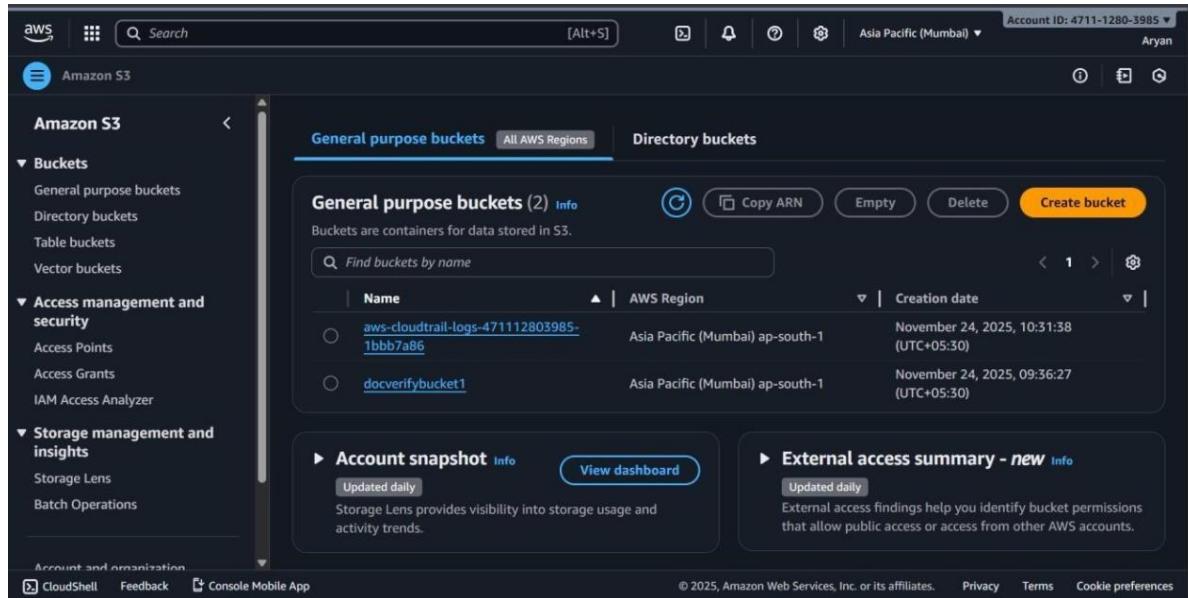
- Go to AWS Console.
- Sign in with your AWS account credentials.

2. Navigate to S3

- In the search bar at the top, type **S3**.
- Click on **S3 (Scalable Storage in the Cloud)**.

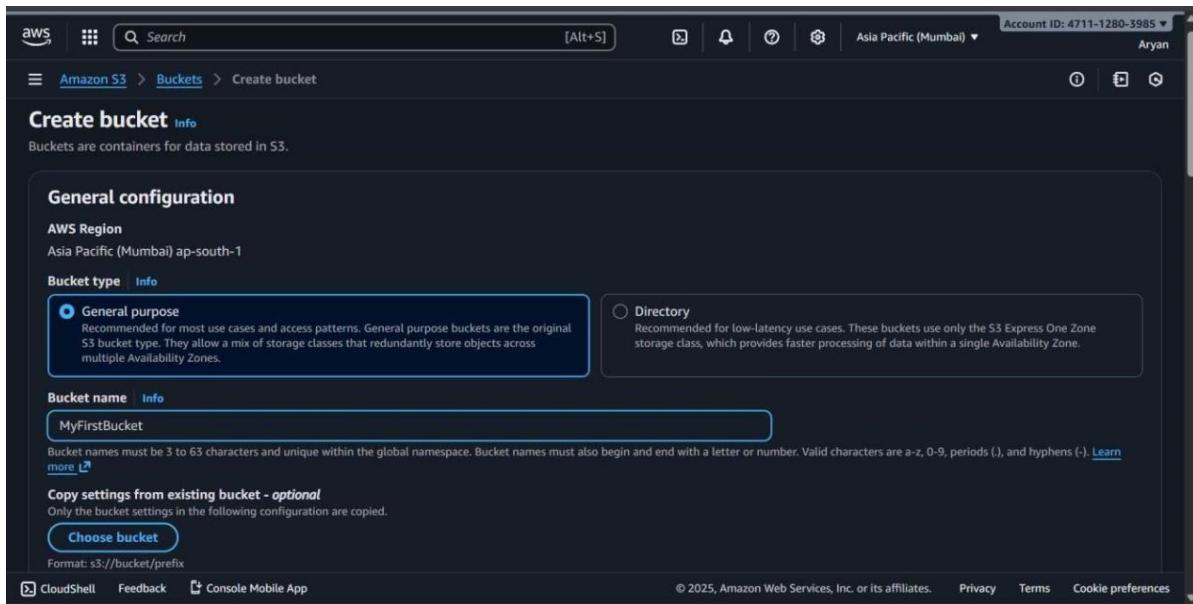
3. Create a New Bucket

- Click on “Create bucket” button.



4. Configure Bucket Settings

- Bucket name:** Enter a unique name (e.g., my-first-s3-bucket- 2025).
Bucket names must be globally unique across AWS.
- AWS Region:** Select a region (choose one closest to your users for better performance).



5. Set Bucket Options

- **Object Ownership:**
 - Choose **ACLs disabled (recommended)** for most use cases.
- **Block Public Access:**
 - By default, all public access is blocked (recommended for private buckets).
 - If you need a public bucket (for hosting static websites), uncheck the option and acknowledge the warning.

6. Configure Bucket Settings (Optional)

- **Versioning:** Enable if you want to keep multiple versions of an object.
- **Encryption:** Enable default encryption if required.
- **Tags:** Add key-value tags if you want to manage cost tracking or organization.

7. Review and Create

- Review your settings. • Click **Create bucket**.

The screenshot shows the AWS S3 Buckets page. At the top, there is a green success message: "Successfully created bucket "my.first.bucket.124". To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there are two tabs: "General purpose buckets" (selected) and "Directory buckets". Under "General purpose buckets", there is a table with three rows:

Name	AWS Region	Creation date
aws-cloudtrail-logs-47112803985-1bbb7a86	Asia Pacific (Mumbai) ap-south-1	November 24, 2025, 10:31:38 (UTC+05:30)
docverifybucket1	Asia Pacific (Mumbai) ap-south-1	November 24, 2025, 09:36:27 (UTC+05:30)
my.first.bucket.124	Asia Pacific (Mumbai) ap-south-1	November 28, 2025, 08:35:01 (UTC+05:30)

On the right side of the page, there are two callout boxes: "Account snapshot" (updated daily) and "External access summary - new" (updated daily). The "External access summary" box notes that external access findings help identify bucket permissions for public access or access from other AWS accounts.

8. Upload Objects (Optional)

- After creation, open the bucket.
- Click **Upload → Add files**.
- Select your file(s), then click **Upload**.

The screenshot shows the AWS S3 Bucket Objects page for the bucket "my.first.bucket.124". The top navigation bar includes "Amazon S3 > Buckets > my.first.bucket.124". The main content area has a header "my.first.bucket.124" with a "Info" link. Below the header are tabs: "Objects" (selected), "Metadata", "Properties", "Permissions", "Metrics", "Management", and "Access Points". A toolbar at the top provides actions: "Copy S3 URI", "Copy URL", "Download", "Open", "Delete", "Actions", "Create folder", and "Upload". A note states: "Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)". A search bar and a table header with columns "Name", "Type", "Last modified", "Size", and "Storage class" are present. A message "No objects" indicates the bucket is currently empty. At the bottom, there is a large "Upload" button.

PRACTICAL-4

OBJECTIVE – CREATE AN AWS LAMBDA FUNCTION

1. Log in to AWS Management Console

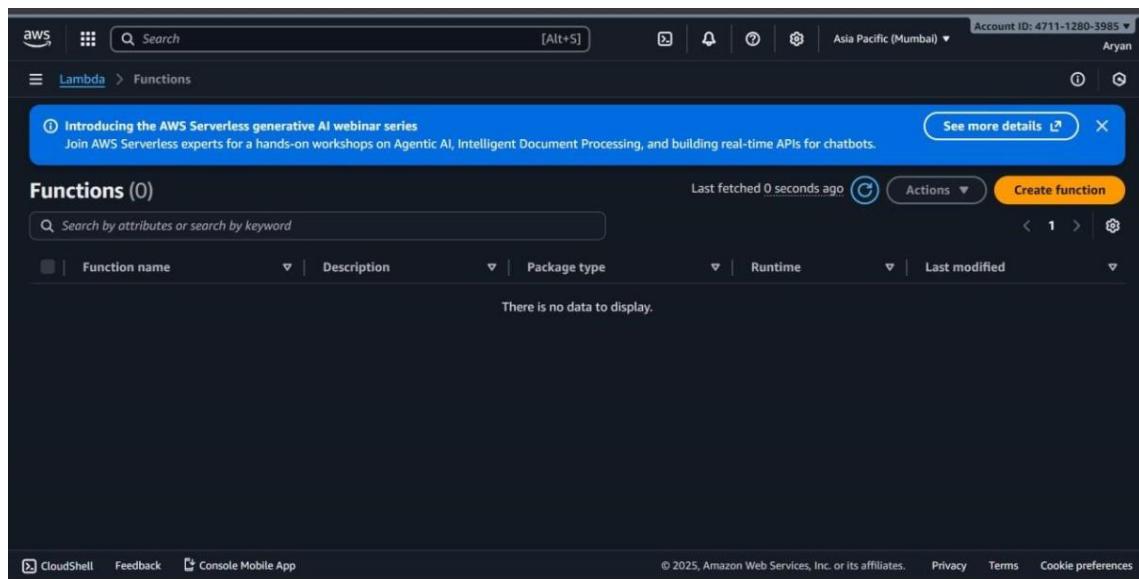
- Go to AWS Console.
- Sign in with your credentials.

2. Navigate to AWS Lambda

- In the search bar, type **Lambda**.
- Click on **Lambda** service.

3. Create a New Lambda Function

- Click **Create function**.



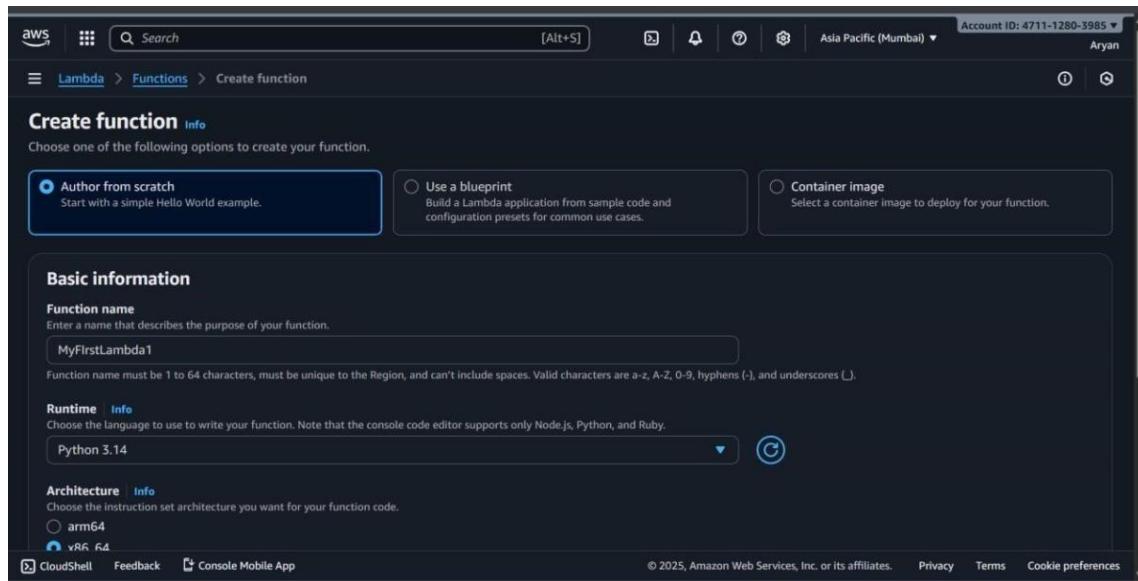
4. Choose a Creation Method You'll see 3 options:

1. **Author from scratch** → (most common, start fresh).
2. **Use a blueprint** → predefined templates.

3. **Container image** → deploy code as Docker container. Select **Author from scratch**.

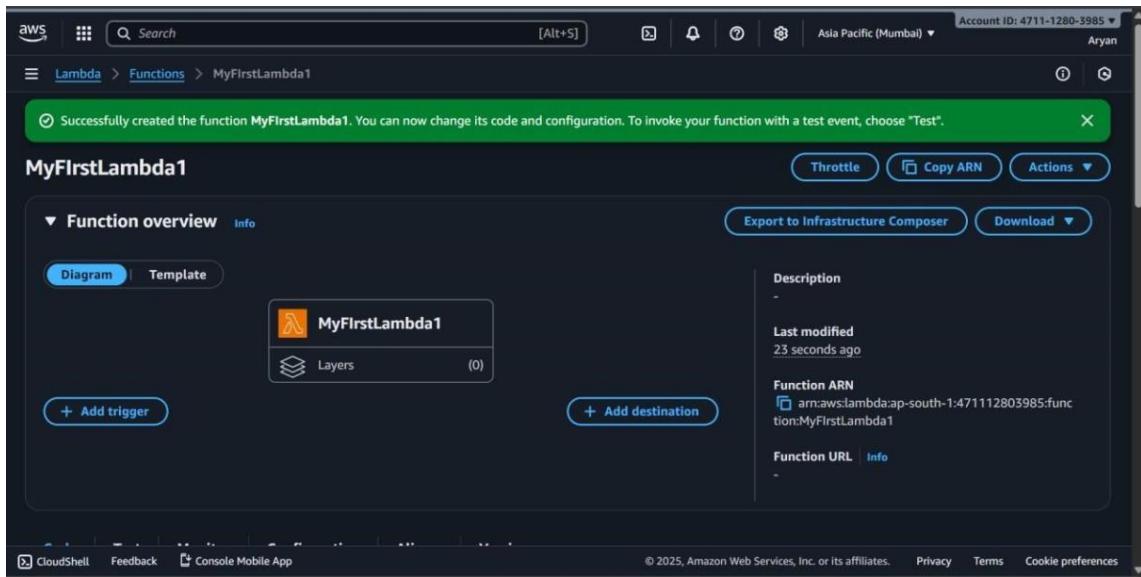
5. Configure Basic Settings

- **Function name:** Enter a unique name (e.g., MyFirstLambda).
- **Runtime:** Choose a runtime (Node.js, Python, Java, Go, etc. → example: **Python 3.9**).
- **Permissions (Execution Role):**
 - **Create a new role with basic Lambda permissions** (recommended if you're new).
 - Or choose an existing IAM role if you already have one.



6. Click Create Function

- Wait a few seconds while AWS provisions the function.

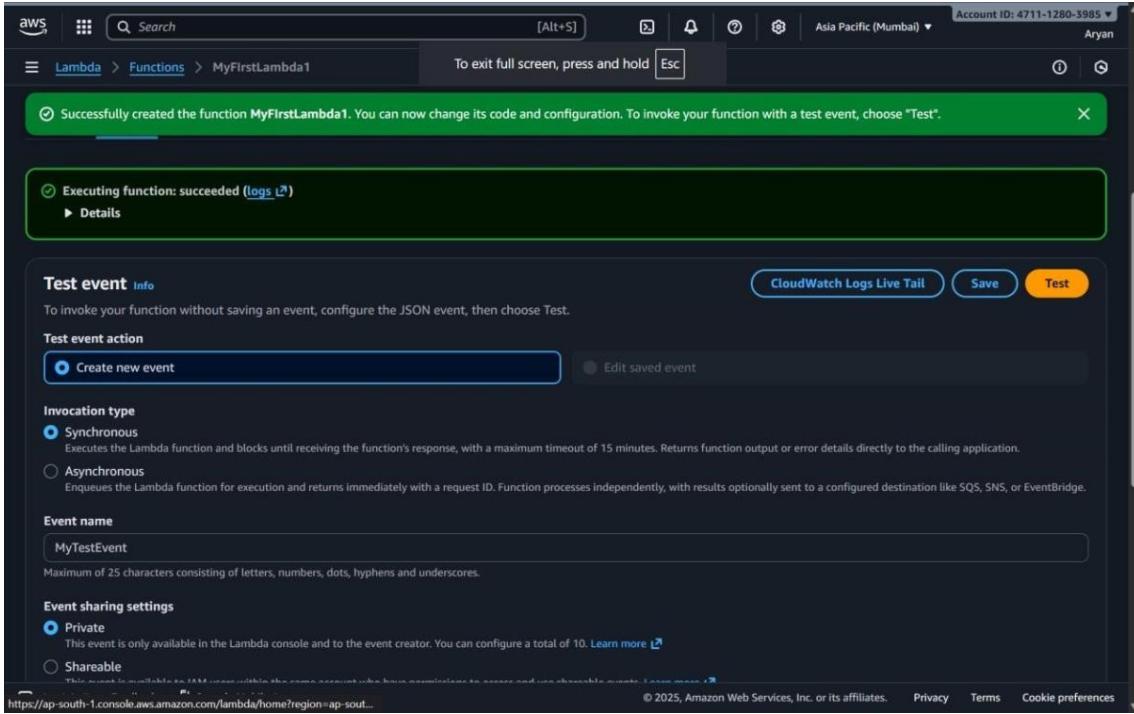


7. Add Your Code

- In the **Function code** section, you can:
 - Write inline code in the editor.
 - Or upload a .zip file.
 - Or use **Amazon S3** (if your code is stored there).

Example default code in Python:

```
def lambda_handler(event, context):  
  
    return {  
  
        'statusCode': 200,  
  
        'body': 'Hello from Lambda!'  
  
    }
```



8. Configure Test Event

- Click **Test** → **Configure test event**.
- Give it a name (e.g., TestEvent).
- Keep the default event JSON or modify as needed.
- Save.

9. Run the Function

- Click **Test** again.
- Check the execution results (output, logs, and status).

10. (Optional) Add a Trigger

- You can connect Lambda to services like:
 - API Gateway (for REST APIs)
 - S3 (trigger on file uploads)
 - DynamoDB (trigger on database changes)
 - EventBridge (scheduled events)

Screenshot of the AWS Lambda console showing the creation of a trigger for a function.

Trigger configuration (Info)

Bucket: s3/my.first.bucket.124

Event types: All object create events

Prefix - optional: e.g. images/

Function overview (Info)

Description: -

Last modified: 15 minutes ago

Function ARN: arn:aws:lambda:ap-south-1:471112803985:function:MyFirstLambda1

Function URL: -

Destinations:

- MyFirstLambda1 (Function)
- S3 (Trigger)

Actions: Throttle, Copy ARN, Actions

The screenshot shows the AWS S3 console interface. At the top, there is a green success message: "Upload succeeded" and "For more information, see the Files and folders table." Below this, the "Summary" section shows the destination as "s3://my.first.bucket.124". Under "Succeeded", it lists "1 file, 33.0 B (100.00%)". Under "Failed", it lists "0 files, 0 B (0%)". The "Files and folders" tab is selected, showing a table with one row: "Random Test File.txt" (text/plain, 33.0 B, Succeeded). The status bar at the bottom indicates "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

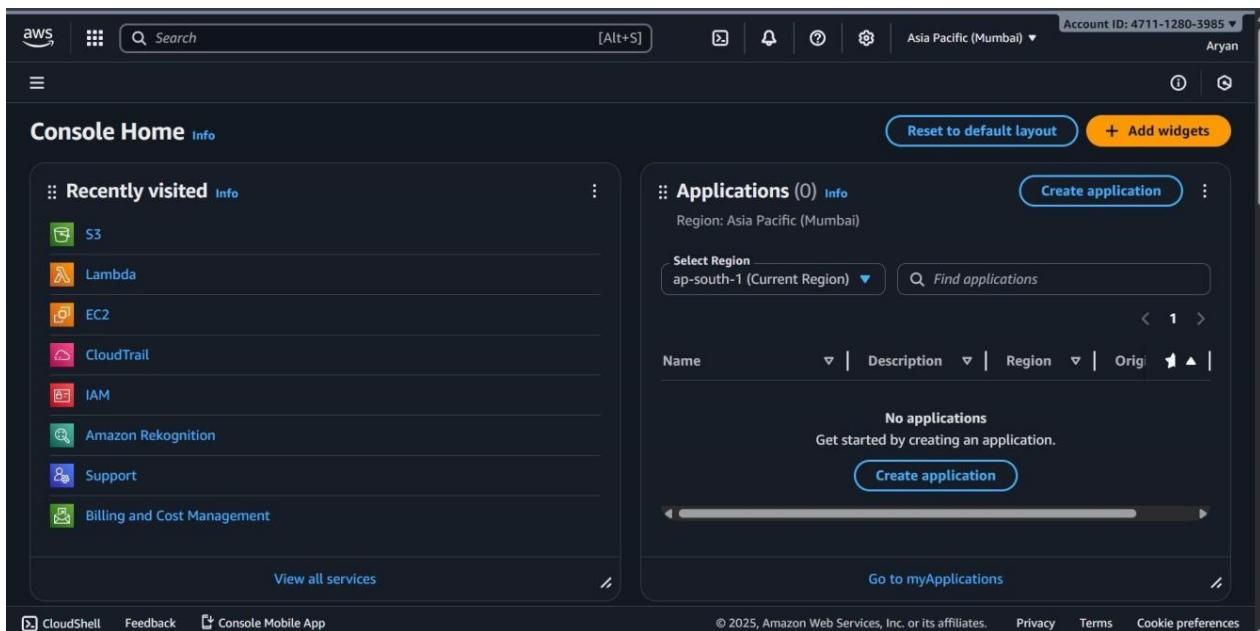
The screenshot shows the AWS Lambda console interface. At the top, there is a green success message: "Successfully updated the function MyFirstLambda1." Below this, the "Code source" tab is selected, showing the "lambda_function.py" code editor. The code defines a lambda function with a handler named "lambda_handler" that prints "Hello World". The status bar at the bottom indicates "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Practical:-5

Objective:- Creating an IAM (Identity and Access Management) user in AWS is to provide secure and controlled access to AWS resources for individuals or applications without using the root account.

Step 1: Sign in to AWS Console

Log in to your **AWS Management Console** using your **root account** or an **IAM user** who has **Administrator privileges**.



Step 2: Open IAM Service

In the search bar at the top, type **IAM**. Select **IAM (Identity and Access Management)**.

The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', 'Users' is selected. The main area displays 'Security recommendations' with two items: 'Root user has MFA' (green checkmark) and 'Root user has no active access keys' (green checkmark). Below this is the 'IAM resources' section, which lists 1 User group, 1 User, 5 Roles, 2 Policies, and 0 Identity providers. To the right is the 'AWS Account' section, showing the Account ID (471112803985), Account Alias (Create), and Sign-in URL (https://471112803985.signin.aws.amazon.com/console). A 'Quick Links' section includes 'My security credentials'.

Step 3: Go to Users Section

In the left sidebar, click on **Users**. You will see a list of all existing users. Click “**Create user**”

The screenshot shows the 'Specify user details' step of the 'Create user' wizard. On the left, a sidebar shows 'Step 1 Specify user details' (selected), 'Step 2 Set permissions', and 'Step 3 Review and create'. The main area is titled 'User details' and contains a 'User name' input field. Below it is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ _ - (hyphen)'. There is also an optional checkbox for 'Provide user access to the AWS Management Console - optional', with a note: 'In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.' A callout box provides information about generating programmatic access keys. At the bottom are 'Cancel' and 'Next >' buttons.

to add a new one.

Step 4: Enter User Details

Enter a **User name**. Choose the type of access: **Password access** → if the user needs to log in to the AWS Console. Click **Next**.

Specify user details

User details

User name
Aryan_IAM_User_1
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ _ - (hyphen)

Provide user access to the AWS Management Console - optional
In addition to console access, users with SigninLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

Console password

Autogenerated password
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

 Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password

Step 5: Set Permissions

Choose **Attach policies directly** → assign permissions manually (e.g.,

AmazonS3FullAccess, AdministratorAccess, etc.). Then click **Next**.

Set permissions

Attach policies directly
Attach a managed policy directly to a user.
As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1425)

Choose one or more policies to attach to your new user.

Filter by Type

Policy name	Type	Attached entities
AccessAnalyzerServiceRole...	AWS managed	0
AdministratorAccess	AWS managed - job function	0
AdministratorAccess-Amp...	AWS managed	0

The screenshot shows the 'Permissions policies' section of the 'Create user' wizard. It lists 17 matches for 'S3'. One policy, 'AmazonS3FullAccess', is selected and highlighted with a blue border. Other policies listed include 'AmazonDMSRedshiftS3Role', 'AmazonS3ObjectLambda...', 'AmazonS3OutpostsFullAc...', 'AmazonS3OutpostsRead...', and 'AmazonS3ReadOnlyAccess'. At the top right, there is a 'Create policy' button.

Step 6: Review and Create

Review all details carefully. Click **Create user**.

This screenshot shows the final review step before creating the user. It includes:

- User details:** User name is set to 'Aryan_IAM_User_1'. The 'Console password type' is 'Custom password' and 'Require password reset' is set to 'Yes'.
- Permissions summary:** Shows three policies assigned: 'AmazonS3FullAccess', 'IAMUserChangePassword', and 'IAMUserChangePassword' (repeated).
- Tags - optional:** A note stating that tags are key-value pairs used for identifying resources.

Step 7: Save Login Details

- Once the user is created, AWS will show:

- **User ARN (Amazon Resource Name)** ○ **Console login link**
- **Password or Access key/Secret key** (Download the .csvfile — it won't be shown again).

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

View user

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details	
Console sign-in URL	https://471112803985.signin.aws.amazon.com/console
User name	Aryan_IAM_User_1
Console password	***** Show

Email sign-in instructions ↗

Download .csv file

Return to users list

Practical :-6

Objective:- To secure an AWS IAM user account by enabling Multi-Factor Authentication (MFA), adding an extra verification step to prevent unauthorized access.

Step 1:- Sign in to AWS Management Console

- Log in using your **root account** or **IAM admin user**.
- Open the console.

Console

Recent

- IAM
- S3
- Lambda
- EC2
- CloudWatch
- Amazon
- Support
- Billing

Were these results helpful?

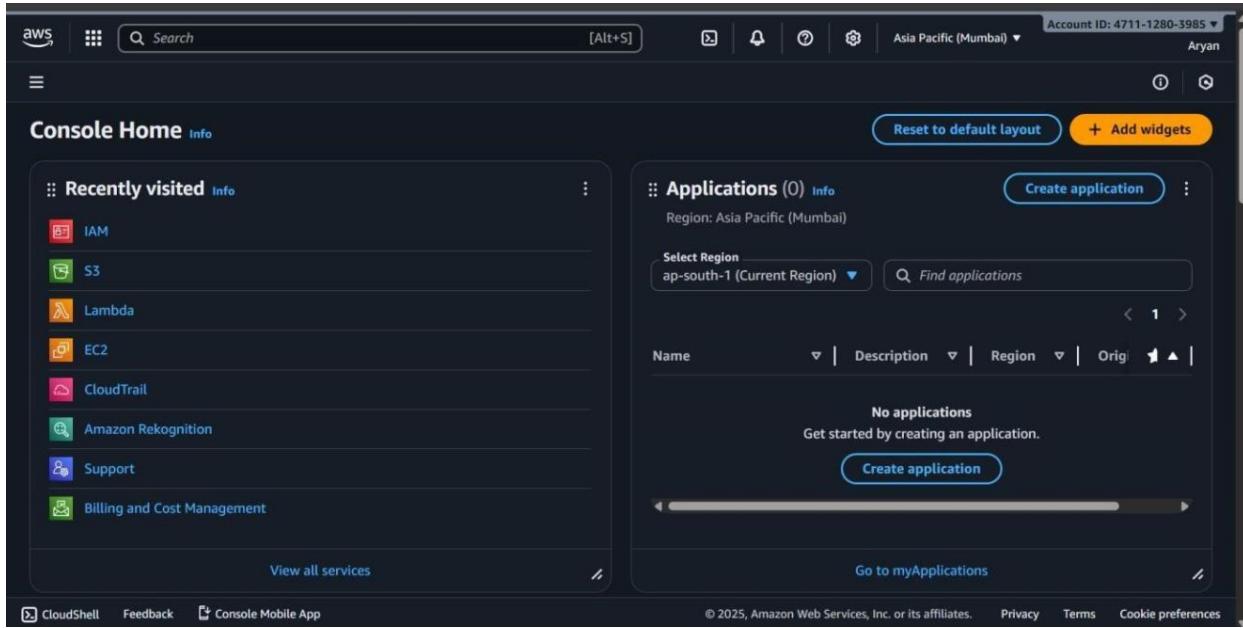
Yes **No**

Services

- IAM** Manage access to AWS resources
- IAM Identity Center** Manage workforce user access to multiple AWS accounts and cloud applications
- Resource Access Manager** Share AWS resources with other accounts or AWS Organizations

Features

- IAM Access analyzer for S3** S3 feature
- Groups** IAM feature



Step 2:- Go to IAM Service

- In the search bar, type **IAM**.
- Click **IAM (Identity and Access Management)**.

Step 3:-Open “Users”

The screenshot shows the AWS IAM service interface. In the left navigation panel, under the 'Access management' section, the 'Users' option is selected. The main content area displays a table titled 'Users (2)'. The table has columns for 'User name', 'Path', 'Group', 'Last activity', 'MFA', and 'Password age'. Two users are listed: 'Aryan_IAM_User_1' and 'doc-verifier-bot'. The 'Aryan_IAM_User_1' row shows '1' in the Group column and '1 hour' in the Password age column. The 'doc-verifier-bot' row shows '0' in the Group column and '-' in the Password age column. At the top right of the table, there are 'Delete' and 'Create user' buttons.

- In the left navigation panel, select **Users**.

Click the **username** for which you want to enable MFA.

Step 4:-Go to the “Security Credentials” Tab

- After opening the user's profile, click on **Security credentials**.
- Scroll down to the section **Multi-Factor Authentication (MFA)**.

The screenshot shows the detailed view of the 'Aryan_IAM_User_1' user profile. In the left navigation panel, under the 'Access management' section, the 'Users' option is selected. The main content area is divided into tabs: 'Permissions', 'Groups (1)', 'Tags', 'Security credentials', and 'Last Accessed'. The 'Security credentials' tab is currently active. It displays information such as ARN (arn:aws:iam::471112803985:user/Aryan_IAM_User_1), Console access (Enabled without MFA), and Access key 1 (Create access key). Below this, the 'Permissions policies' section lists permissions attached to the user. At the bottom right of the page, there are 'Remove' and 'Add permissions' buttons.

Step 5:-Click “Assign MFA Device”

- A popup will open with 3 options:
 1. **FIDO2 Security Key**
 2. **Authenticator App** (Google Authenticator / Authy / Microsoft Authenticator)
 3. **Hardware TOTP Device**

For most labs, choose **Authenticator App**.

The screenshot shows the AWS IAM User Details page for 'Aryan_IAM_User_1'. The left sidebar shows 'Access management' with 'Users' selected. The main content area displays the 'Console sign-in' section with a 'Console sign-in link' (https://471112803985.signin.aws.amazon.com/console) and a 'Console password' updated 1 hour ago. The 'Multi-factor authentication (MFA)' section indicates 0 devices assigned, with a 'Assign MFA device' button. The 'Access keys (0)' section has a 'Create access key' button. The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App.

Step 1
Select MFA device

Step 2
Set up device

Select MFA device Info

MFA device name

Device name

This name will be used within the identifying ARN for this device.

Redmi

Maximum 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

MFA device

Device options

In addition to username and password, you will use this device to authenticate into your account.

Passkey or security key

Authenticate using your fingerprint, face, or screen lock. Create a passkey on this device or use another device, like a FIDO2 security key.

Step 6:-Select “Authenticator App” and Continue

- Click **Continue**.
- AWS will show a **QR code** for MFA enrollment.

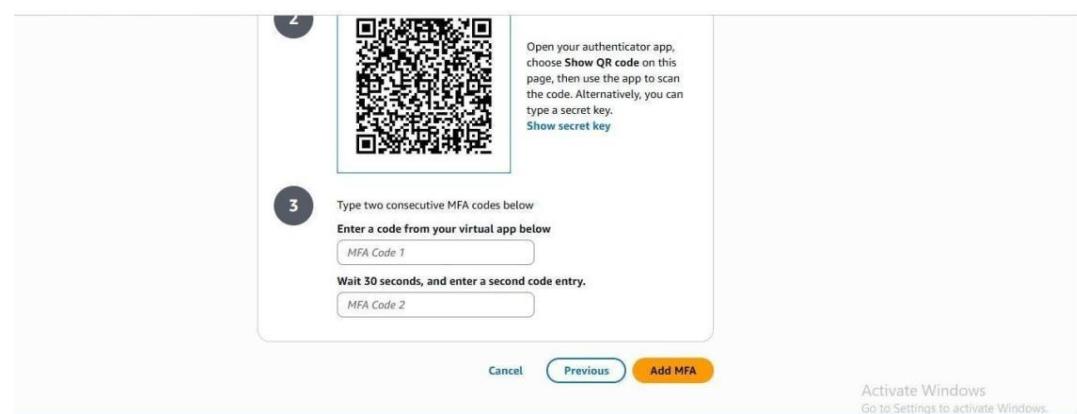
Step 7:-Open your Authenticator App

You can use any app:

- Google Authenticator
- Authy
- Microsoft Authenticator
- LastPass Authenticator

Click **Add Account → Scan QR Code**.

The app will generate a **6-digit one-time password (OTP)** that refreshes every 30 seconds.



Step 8:-MFA Successfully Enabled

“MFA device assigned successfully”

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The user is navigating through the 'Users' section, specifically viewing the details for 'Aryan_IAM_User_1'. A prominent green notification bar at the top states: 'MFA device assigned' with a link to 'View details'. Below this, the 'Summary' section provides key information about the user:

- ARN:** arn:aws:iam::471112803985:user/Aryan_IAM_User_1
- Console access:** Enabled with MFA
- Created:** November 28, 2025, 09:11 (UTC+05:30)
- Last console sign-in:** Never
- Access key 1:** Create access key

The 'Security credentials' tab is currently selected. At the bottom of this tab, there is a 'Console sign-in' section with a 'Manage console access' button.

Practical:-7

Objective: To create a user group in AWS IAM in order to manage permissions collectively for multiple users having similar roles or responsibilities.

Step1:- Open the IAM Service:

In the search bar at the top of the console, type **IAM**, then select **Identity and Access Management** from the results.

The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', 'User groups' is selected. The main area displays 'Security recommendations' with two items: 'Root user has MFA' and 'Root user has no active access keys'. Below this is the 'IAM resources' section, which shows 1 User group, 1 User, 5 Roles, 2 Policies, and 0 Identity providers. To the right is the 'AWS Account' section, displaying the Account ID (471112803985), Account Alias (Create), and Sign-in URL (https://471112803985.siginin.aws.amazon.com/console). A 'Quick Links' section at the bottom right provides links to 'My security credentials' and 'Manage your access keys, multi-factor authentication (MFA) and other credentials'.

Step 2:- Go to User Groups Section:

In the left-hand sidebar, click on **User groups**.

The screenshot shows the AWS IAM User Groups page. On the left, there's a navigation sidebar with options like Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management, Temporary delegation requests), and Access reports. The main area is titled "User groups (1) Info" and contains a table with one row. The table columns are Group name, Users, Permissions, and Creation time. The row shows "IAM_User_Group_1", "0", "Defined", and "31 days ago". There are "Delete" and "Create group" buttons at the top right of the table.

Step 3:- Click on “Create group”:

On the User Groups page, click the “Create group” button to start creating a new group

The screenshot shows the "Create user group" wizard. The first step, "Name the group", has a "User group name" input field containing "Administrators". Below it, there's a section for "Add users to the group - Optional (1/2)" with a search bar and a table showing two users: "Aryan_IAM_User_1" and "doc-verifier-bot". The "Aryan_IAM_User_1" row is selected, indicated by a checked checkbox. The table includes columns for User name, Group, Last activity, and Creation time.

Step 4:- Enter Group Name:

Type a **unique name** for your group (for example, *Developers*, *Admins*, or *ReadOnlyUsers*). For Example, here Administrators.

Step 5:- Attach Permissions Policies (Optional):

You can choose policies to attach to this group, such as:

- AmazonS3FullAccess
- AmazonEC2ReadOnlyAccess
- AdministratorAccess

If you want to add permissions later, you can **skip this step** and click **Next**.

The screenshot shows the AWS IAM console with the path **IAM > User groups > Create user group**. On the left, the navigation menu is visible under the **Access management** section, with **User groups** selected. The main area is titled **Attach permissions policies - Optional (3/1097)**. A search bar at the top of the list allows filtering by policy name, currently set to "administrator". The list displays several AWS managed policies:

Policy name	Type	Used as	Description
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job ...	None	Provides ful...
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	None	Grants acco...
<input type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants acco...
<input type="checkbox"/> AmazonAPIGatewayAdministrator	AWS managed	None	Provides ful...
<input type="checkbox"/> AmazonSecurityLakeAdministrator	AWS managed	None	Provides ful...
<input type="checkbox"/> AWSAppSyncAdministrator	AWS managed	None	Provides adm...
<input type="checkbox"/> AWSAuditManagerAdministratorAccess	AWS managed	None	Provides adm...
<input type="checkbox"/> AWSCloud9Administrator	AWS managed	None	Provides adm...

Step 6:- Add Users to the Group (Optional):

You can select existing IAM users to include in this group now, or you can add users later after creating the group.

Step 7:- Review and Create Group:

Review the group details and attached policies, then click **Create group**.

The screenshot shows the AWS IAM console with the path **IAM > User groups**. The left sidebar shows the **Access management** section with **User groups** selected. The main area displays a green success message: **Administrators user group created.** Below it, the heading **User groups (2)** is shown. A table lists the two user groups:

Group name	Users	Permissions	Creation time
Administrators	1	Defined	Now
IAM_User_Group_1	0	Defined	31 days ago

Practical:-8

Objective:-To create a security role in AWS IAM that allows AWS services or users to securely access specific AWS resources with defined permissions, ensuring controlled and temporary access without sharing long-term credentials.

Step 1:- Open the IAM Service:

In the search bar at the top of the console, type **IAM**, then select **Identity and Access Management** from the results.

The screenshot shows the AWS IAM Dashboard. On the left, the navigation pane includes sections like Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management, Temporary delegation requests), and Access reports. The main area displays security recommendations (Root user has MFA, Root user has no active access keys) and IAM resources (User groups: 1, Users: 1, Roles: 5, Policies: 2, Identity providers: 0). A green banner at the top indicates "Administrators user group created." To the right, there's an "AWS Account" summary with Account ID (471112803985), Account Alias (Create), and a sign-in URL (https://471112803985.siginin.aws.amazon.com/console). A "Quick Links" section provides links to security credentials and manages access keys, MFA, and other credentials.

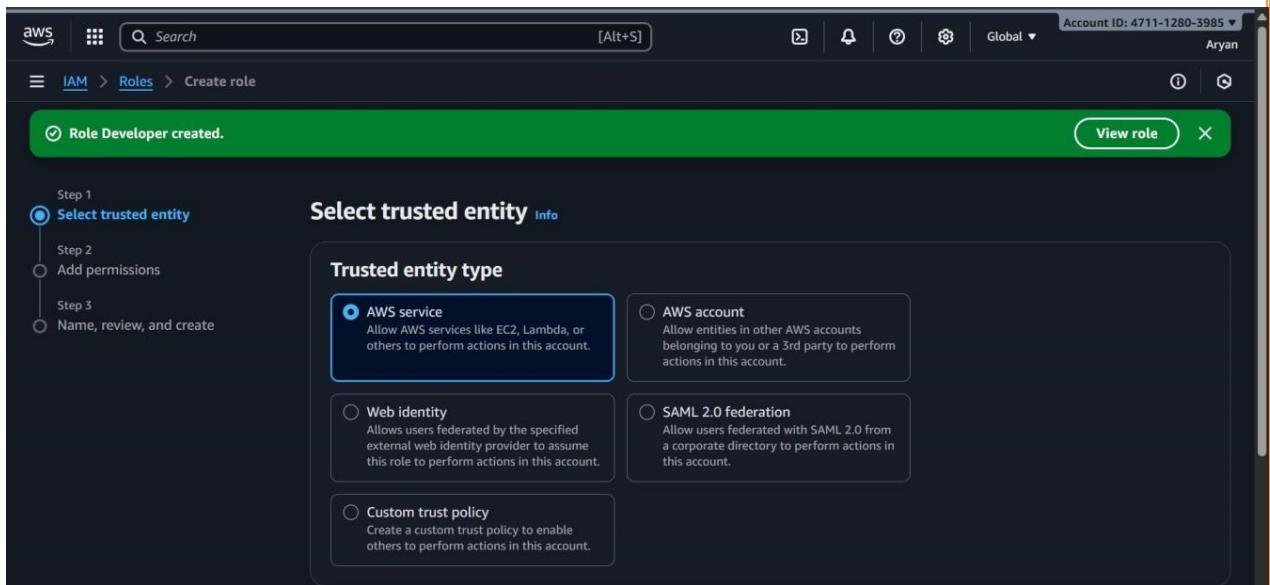
Step 2:- Go to Roles Section:

In the left-hand navigation pane, click on **Roles**.

The screenshot shows the AWS IAM Roles page. The left navigation pane highlights the **Roles** section under Access management. The main content area lists five roles: AWSServiceRoleForResourceExplorer, AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, MyFirstLambda1-role-ngu9r9ff, and mylambda-role-qyk5z9i5. Each role entry includes a "Delete" button and columns for Trusted entities and Last activity. Below the role list, there's a "Roles Anywhere" section with a "Manage" button for authenticating non-AWS workloads.

Step 3:- Click on “Create role”:

On the Roles page, click the “**Create role**” button to start the process.

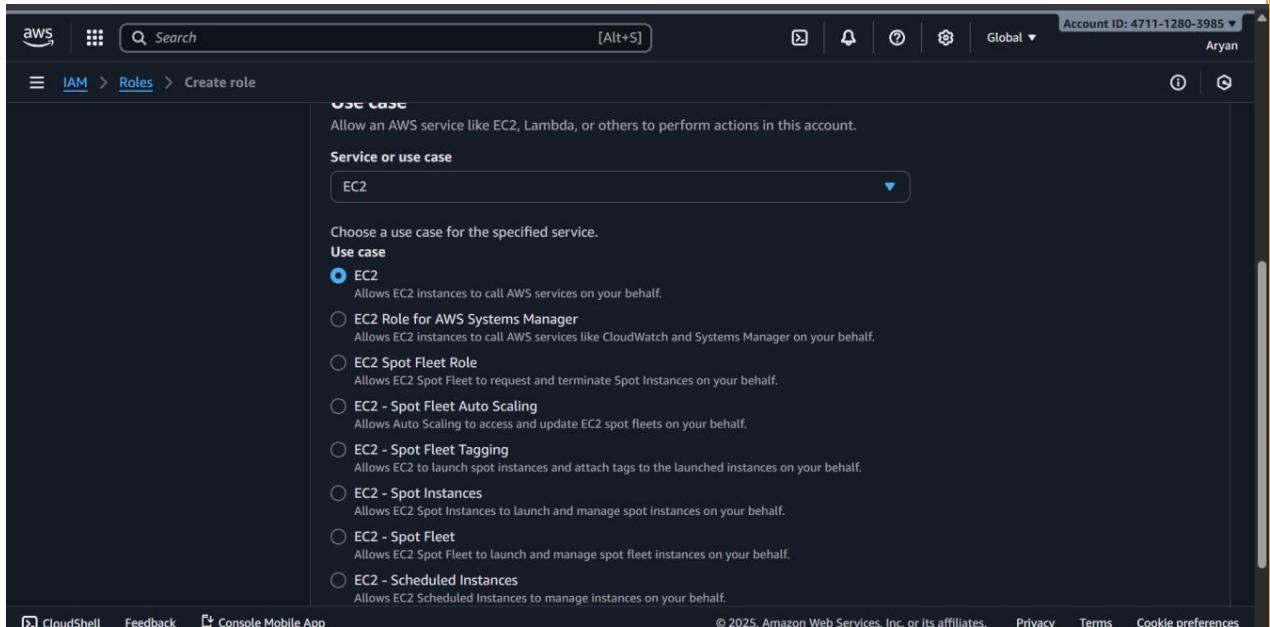


Step 4:- Select Trusted Entity Type:

Choose who will use the role, such as:

- **AWS Service** (e.g., EC2, Lambda)
- **Another AWS Account**
- **Web Identity or SAML 2.0 Federation**

Click **Next** after selecting the appropriate option.



Step 5:-Attach Permissions Policies:

Select the **permissions policies** that define what actions the role can perform (for example, AmazonS3FullAccess or AmazonEC2FullAccess).

The screenshot shows the 'Add permissions' step of the IAM role creation wizard. On the left, a sidebar lists three steps: 'Select trusted entity', 'Add permissions' (which is selected and highlighted in blue), and 'Name, review, and create'. The main area is titled 'Add permissions' with a 'Permissions policies (2/1097)' sub-section. A search bar contains 'Ec2'. A filter bar shows 'All types' and '2 matches'. Two policies are listed: 'AmazonEC2FullAccess' (selected with a checked checkbox) and 'EC2FastLaunchFullAccess'. Below this is a section titled 'Set permissions boundary - optional'. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

Step 6:- Name and Review the Role:

Enter a **role name** (for example, *EC2SecurityRole* or *LambdaAccessRole*) and review all selected settings.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. The sidebar shows 'Name, review, and create' is selected. The main area has a 'Role details' section. Under 'Role name', the value 'Developer' is entered. Under 'Description', the value 'Allows EC2 instances to call AWS services on your behalf.' is entered. At the bottom, a note states: 'Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=.,@-_'. Below this is a 'Step 1: Select trusted entities' section with an 'Edit' button.

The screenshot shows the 'Permissions policy summary' section of the 'Create role' wizard. It lists two managed policies: 'AmazonEC2FullAccess' and 'AmazonS3FullAccess', both categorized as 'Permissions policy'. Below this, the 'Step 3: Add tags' section is shown. It includes a note about optional tags for identifying resources, a button to 'Add new tag', and a note that up to 50 more tags can be added. At the bottom right are 'Cancel', 'Previous', and 'Create role' buttons.

Step7:- Create the Role:

Click **Create role** to finish.

The screenshot shows the 'Roles' page in the AWS IAM console. A green success message at the top states 'Role Developer created.' The main table lists six roles, including the newly created 'Developer' role. The table columns include 'Role name', 'Trusted entities', and 'Last activity'. The 'Developer' role is associated with the 'AWS Service: ec2' and was created 1 hour ago. Other listed roles include 'AWS ServiceRoleForResourceExplorer', 'AWS ServiceRoleForSupport', 'AWS ServiceRoleForTrustedAdvisor', 'MyFirstLambda1-role-ngu9r9ff', and 'mylambda-role-qyk5z9j5'. Navigation links for 'View role', 'Delete', and 'Create role' are visible at the top right of the table area.

Practical :- 9

Objective:- To understand how to design and create a Virtual Private Cloud (VPC) in AWS.

Step1. Open VPC Dashboard

- Sign in to AWS Console.
- Search **VPC** in the services search bar.
- Click **VPC Dashboard**.

The screenshot shows the AWS VPC Dashboard. At the top, there are buttons for 'Create VPC' and 'Launch EC2 Instances'. A note says 'Your Instances will launch in the Asia Pacific region.' Below this is a section titled 'Resources by Region' with a 'Refresh Resources' button. It lists resources categorized by region:

- VPCs:** Mumbai 1 (See all regions)
- NAT Gateways:** Mumbai 0 (See all regions)
- Subnets:** Mumbai 3 (See all regions)
- VPC Peering Connections:** Mumbai 0 (See all regions)
- Route Tables:** Mumbai 1 (See all regions)
- Network ACLs:** Mumbai 1 (See all regions)
- Internet Gateways:** Mumbai 1 (See all regions)
- Security Groups:** Mumbai 3 (See all regions)

On the left, a sidebar menu includes 'Virtual private cloud' sections for 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'NAT gateways', 'Peering connections', and 'Route servers'. On the right, there are 'Service Health' and 'Settings' sections, and a 'Additional Information' section with links to 'VPC Documentation', 'All VPC Resources', and 'Forums'.

Step2. Start Creating a VPC

- On the left menu, select **Your VPCs**.
- Click **Create VPC**.

The screenshot shows the 'Your VPCs' page. A blue banner at the top says 'Introducing VPC encryption control' with a description: 'Manage and enforce encryption settings across your Virtual Private Cloud (VPC) resources. This centralized control helps ensure compliance with security policies and data protection regulations.' Below this, there's a 'Create encryption control' button. The main area shows a table for 'Your VPCs' with one entry:

Your VPCs (1)		Last updated less than a minute ago	Actions	Create VPC
Select a VPC above				

The left sidebar is identical to the one in the previous screenshot, showing options for managing VPC components like Subnets, Route tables, and Internet gateways.

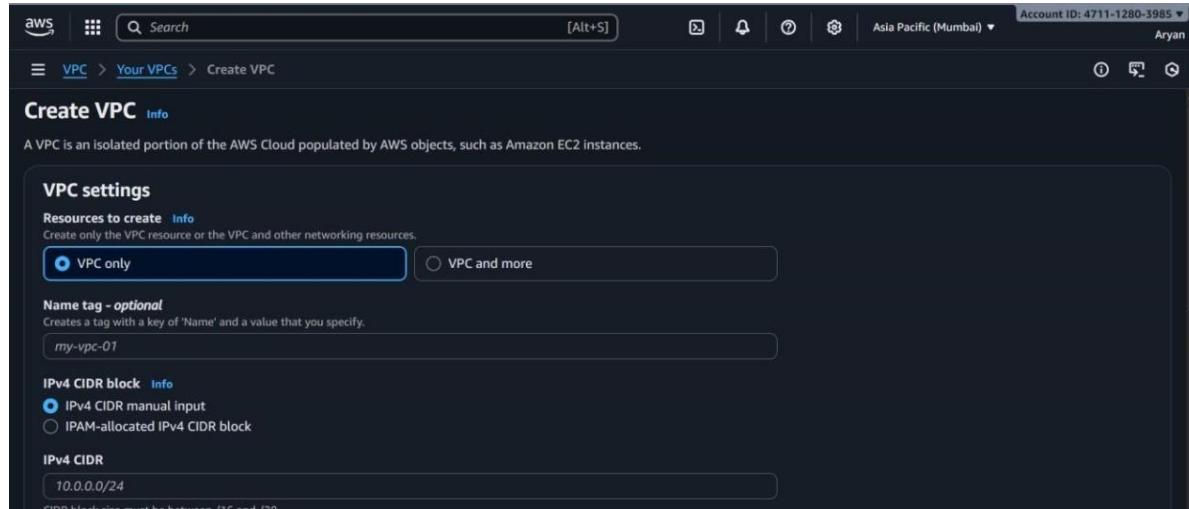
Step3. Select VPC Creation Method

You will get two options:

1. **VPC Only** → Create VPC manually

2. **VPC and more** → Automatically create VPC with subnets, IGW, route tables, etc.

Choose **VPC Only** for full control.



Step 4. Configure VPC Settings

Fill the form:

(a) Name tag • Example: MyVPC

(b) IPv4 CIDR Block

- Example: 10.0.0.0/16

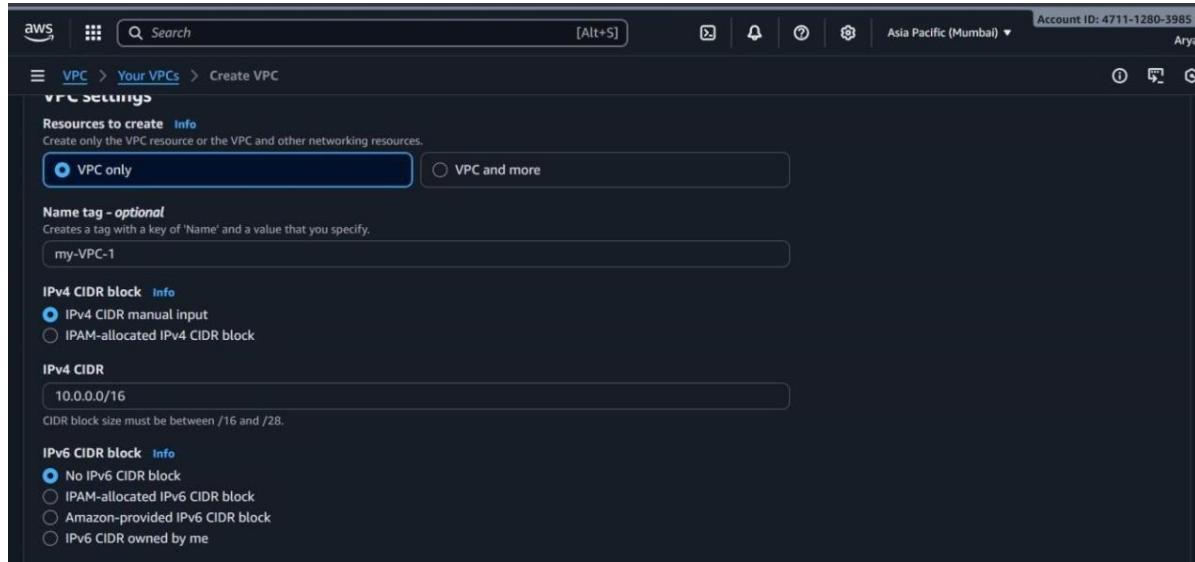
(c) IPv6 CIDR Block

- Choose **No IPv6 CIDR block** (optional)

(d) Tenancy

- Default (recommended)

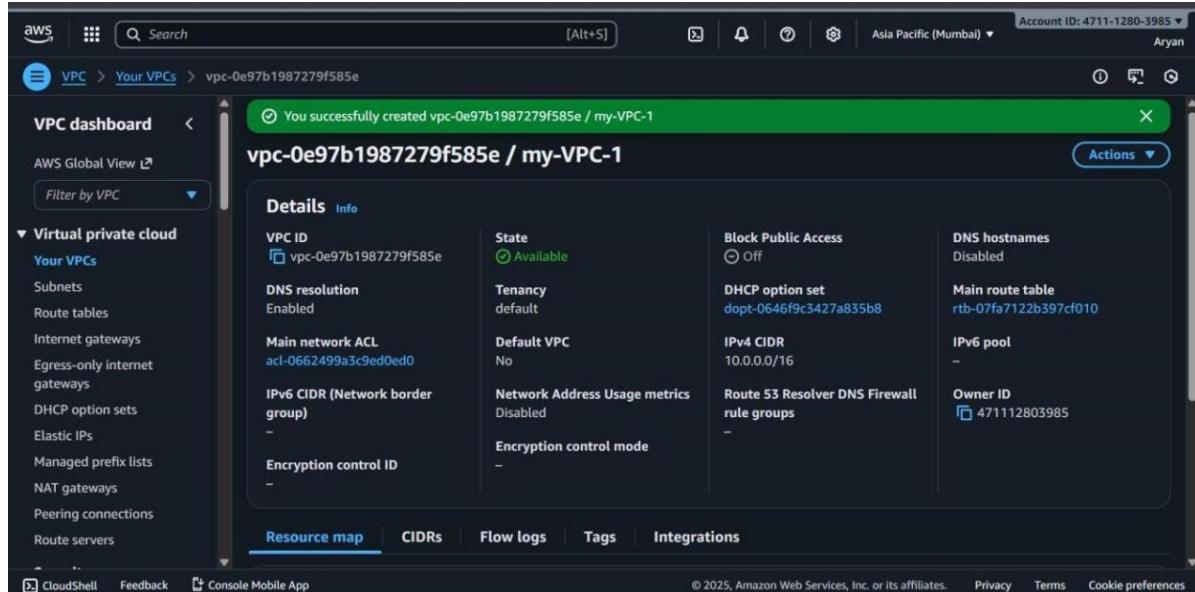
Click **Create VPC**.



Step 5. Create Subnets

- On the left side, click **Subnets** → **Create subnet**.
- Select your **VPC**.
- Add at least **two subnets**: o Public → 10.0.1.0/24 o Private → 10.0.2.0/24
- Choose different **Availability Zones**.

Click **Create Subnet**.



Name	Subnet ID	State	VPC
-	subnet-028b6465a5ab1a760	Available	vpc-0d63dc412a5319d00
-	subnet-05c71ff4dd5119034	Available	vpc-0d63dc412a5319d00
-	subnet-0dbc1a60df83f79864	Available	vpc-0d63dc412a5319d00

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.
vpc-0e97b1987279f585e (my-VPC-1)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The screenshot shows two consecutive screenshots of the AWS VPC console.

Screenshot 1: Create subnet

This page allows you to define a new subnet. It includes fields for:

- Availability Zone**: Info - Choose the zone in which your subnet will reside, or let Amazon choose one for you. (No preference)
- 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. (10.0.0.0/16)
- IPv4 subnet CIDR block**: 10.0.0.0/24 (256 IPs)
- Tags - optional**: No tags associated with the resource. (Add new tag, Remove)

Screenshot 2: Subnet creation confirmation and dashboard

A green success message at the top states: "You have successfully created 1 subnet: subnet-04aabde6b35ee9cab".

The dashboard shows the following information:

- Subnets (1) Info**: Last updated 1 minute ago.
- Actions**: Create subnet.
- Table Headers**: Subnet ID, Name, Subnet ID, State, VPC.
- Table Data**: A single row for the newly created subnet: Subnet ID: subnet-04aabde6b35ee9cab, Name: -, State: Available, VPC: [vpc-0e97b1987279f585e | my...](#).
- Select a subnet**: A dropdown menu for selecting the subnet.

Step 6. Create an Internet Gateway (IGW)

- Click **Internet Gateways** → **Create IGW**.
- Name it e.g., MyIGW.
- After creation, select it → click **Attach to VPC** → choose your **VPC**.

Screenshot of the AWS VPC dashboard showing the Internet gateways section. A single internet gateway is listed:

Name	Internet gateway ID	State	VPC ID
-	igw-0a8578345feb0b585	Attached	vpc-0d63dc412a5319d00

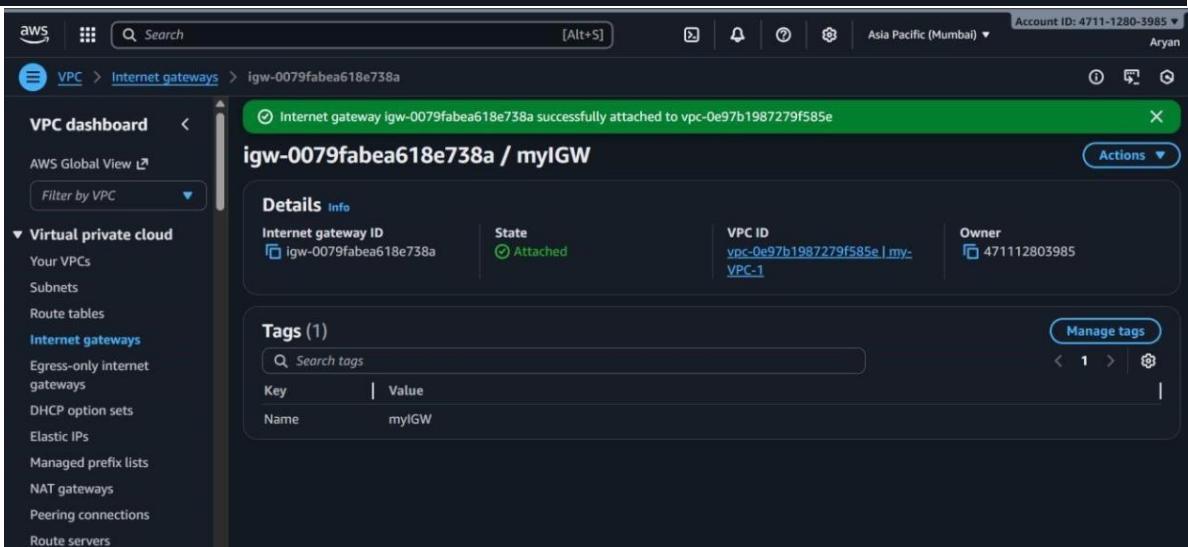
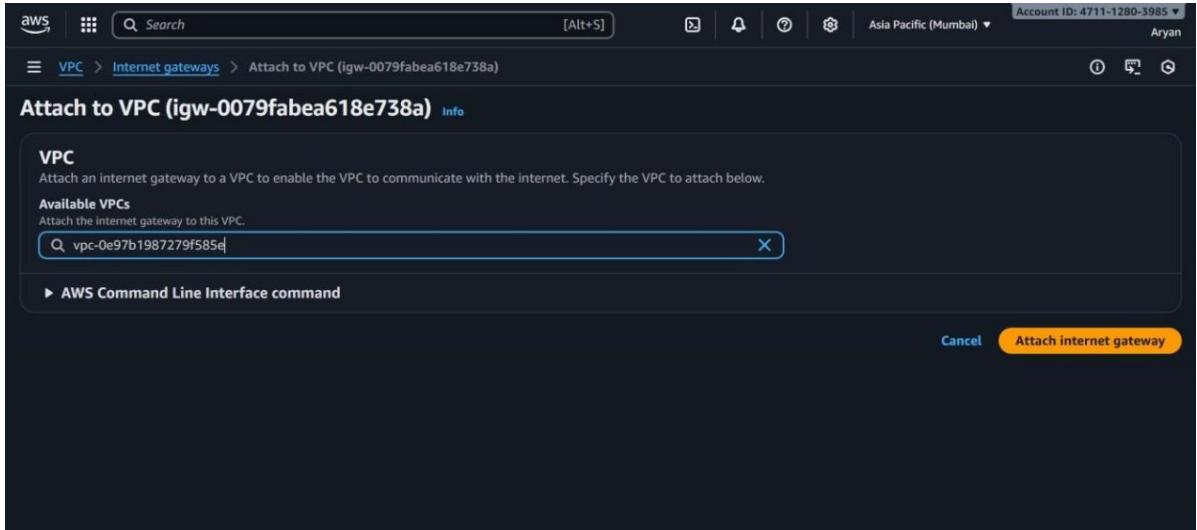
A modal window titled "Select an internet gateway above" is displayed, indicating that no gateway has been selected.

Screenshot of the "Create internet gateway" wizard. In the "Internet gateway settings" step, a name tag "myIGW" is specified. In the "Tags - optional" step, a tag "Name" with value "myIGW" is added.

Screenshot of the AWS VPC dashboard showing the details of the newly created internet gateway "igw-0079fabea618e738a / myIGW". The gateway is currently detached from any VPC.

Internet gateway ID	State	VPC ID	Owner
igw-0079fabea618e738a	Detached	-	471112803985

The "Tags (1)" section shows the tag "Name" with value "myIGW".



Step 7. Configure Route Tables

Public Route Table

- Go to **Route Tables** → Create route table.
- Name: PublicRT.
- Select your VPC and create.
- Go to **Routes** → Edit → Add route:
 - Destination: 0.0.0.0/0
 - Target: **Internet**
- Go to **Subnet Associations** → Associate with **public subnet**.

Private Route Table

- Create another route table named PrivateRT.
- Associate it with **private subnet** (no internet route).

Screenshot of the AWS VPC dashboard showing the Route tables section. A single route table is listed:

Name	Route table ID	Explicit subnet associations	Edge associations	Main
-	rtb-0023ef657ee309e5b	-	-	Yes

The sidebar shows the following navigation paths under Virtual private cloud:

- Your VPCs
- Subnets
- Route tables** (selected)
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers

Screenshot of the "Create route table" wizard.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Public-RT"/> (X) Remove

Add new tag
You can add 49 more tags.

Cancel Create route table

Screenshot of the "Edit routes" page for the route table rtb-0b1c88c61c45abea4.

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	<input type="text" value="local"/> (X) <input type="text" value="local"/> (X)	<input checked="" type="radio"/> Active	No	CreateRouteTable

Add route Cancel Preview Save changes

aws Search [Alt+S] Account ID: 4711-1280-3985 Asia Pacific (Mumbai) Aryan

VPC > Route tables > rtb-0b1c88c61c45abea4 > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute

Add route Remove

Cancel Preview Save changes

aws Search [Alt+S] Account ID: 4711-1280-3985 Asia Pacific (Mumbai) Aryan

VPC > Route tables > rtb-0b1c88c61c45abea4

Updated routes for rtb-0b1c88c61c45abea4 / Public-RT successfully

rtb-0b1c88c61c45abea4 / Public-RT

Details Info

Route table ID rtb-0b1c88c61c45abea4	Main <input checked="" type="checkbox"/> No	Explicit subnet associations -	Edge associations -
VPC vpc-0e97b1987279f585e my-VPC-1	Owner ID <input checked="" type="checkbox"/> 471112803985		

Actions ▾

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-0079fabea618e...	Active	No	Create Route

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Name **Value - optional** Private-RT **Add new tag**

You can add 49 more tags.

Create route table

Details

Route table ID rtb-0e296cd139a06a61b

Main No

Owner ID vpc-0e97b1987279f585e | my-VPC-1

Explicit subnet associations -

Edge associations -

Routes (1)

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

Step 8. Create a NAT Gateway (Optional for private subnet internet)

- Go to **NAT Gateways** → Create NAT Gateway.
- Select:
 - Subnet: **Public subnet**
 - Allocate Elastic IP
- Update **PrivateRT** → **Routes** to:
 - Destination: 0.0.0.0/0
 - Target: **NAT Gateway**

aws Search [Alt+S] Asia Pacific (Mumbai) Account ID: 4711-1280-3985 Aryan

VPC > NAT gateways

Subnets Route tables Internet gateways Egress-only internet gateways DHCP option sets Elastic IPs Managed prefix lists **NAT gateways** Peering connections Route servers

Security Network ACLs Security groups

PrivateLink and Lattice Getting started Endpoints Endpoint services

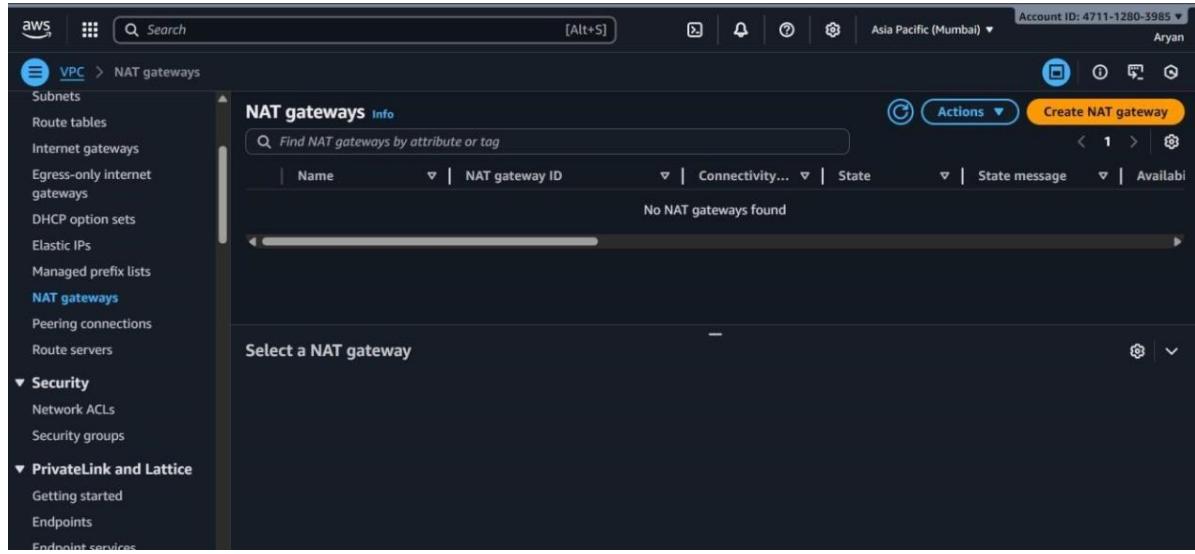
NAT gateways Info

Find NAT gateways by attribute or tag

Name	NAT gateway ID	Connectivity...	State	State message	Available
No NAT gateways found					

Select a NAT gateway

Actions Create NAT gateway



aws Search [Alt+S] Asia Pacific (Mumbai) Account ID: 4711-1280-3985 Aryan

VPC > NAT gateways > Create NAT gateway

Select a connectivity type for the NAT gateway.

Public

Private

Method of Elastic IP (EIP) allocation Info

Choose how IP addresses are associated with NAT gateways.

Automatic

AWS automatically manages EIPs and AZ coverage for NAT gateways. This ensures easy scaling—adding AZs automatically allocates EIPs, simplifying management.

Manual

Manually assigns specific IP addresses for compliance or whitelisting. Note: Requires manual scaling to new AZs as workloads expand.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

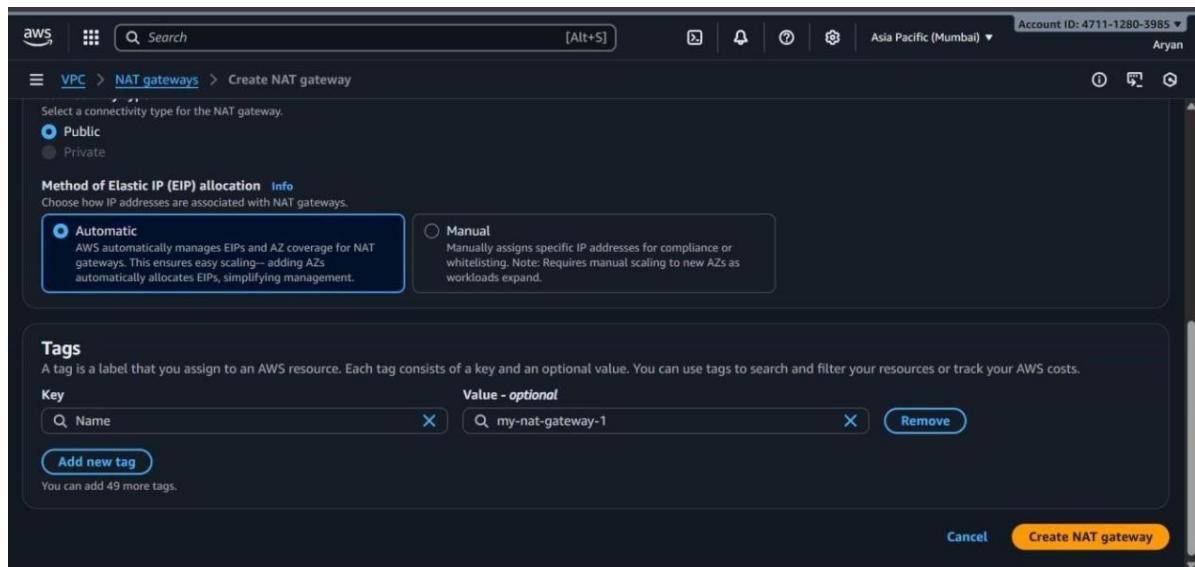
Key Value - optional

Name my-nat-gateway-1 Remove

Add new tag

You can add 49 more tags.

Cancel Create NAT gateway



aws Search [Alt+S] Asia Pacific (Mumbai) Account ID: 4711-1280-3985 Aryan

VPC > NAT gateways > nat-1017dbc3d228e408f

NAT gateway nat-1017dbc3d228e408f | my-nat-gateway-1 was created successfully.

nat-1017dbc3d228e408f / my-nat-gateway-1

Details

NAT gateway ID nat-1017dbc3d228e408f	Availability mode Regional	State Pending	State message Info -
NAT gateway ARN arn:aws:ec2:ap-south-1:471112803985:natgateway/nat-1017dbc3d228e408f	Connectivity type Public	Created Friday, November 28, 2025 at 11:02:29 GMT+5:30	Deleted -
VPC vpc-0e97b1987279f585e / my-VPC-1	Method of EIP allocation Automatic		

IP addresses Monitoring Flow logs Tags

Associated IP addresses

Search Edit IP address associations

