

Experiments for the Course: AWS Cloud Practitioner

Index

S. No.	Name of experiment	Experiment date	Submission date	Page no.
1.	Create a new IAM role for EC2 with the policy AmazonS3ReadOnlyAccess.	03/08/25	15/08/25	4
2.	Create a custom policy that allows listing all S3 buckets but denies deleting them.	03/08/25	15/08/25	7
3.	View the IAM Credential Report.	03/08/25	15/08/25	11
4.	Create a Service Control Policy (SCP) using AWS Organizations that blocks deletion of EC2 instances.	03/08/25	15/08/25	14
5.	Create a new security group that allows SSH (port 22) access only from your public IP.	04/08/25	15/08/25	16
6.	Open PuTTYgen. Load your .pem key file and convert it to .ppk.	04/08/25	15/08/25	19
7.	Create a simple text file on your EC2 instance named test.txt using the nano or vi editor.	04/08/25	15/08/25	21
8.	Access the environment's Configuration page. List any two resources (e.g., EC2, Load Balancer) that were automatically provisioned.	05/08/25	15/08/25	22

9.	Create a new volume from your snapshot.	05/08/25	15/08/25	24
10.	Use the Amazon Data Lifecycle Manager (DLM) to create a scheduled backup policy.	05/08/25	15/08/25	26
11.	Upload another file and select S3 One Zone-IA or S3 Glacier as the storage class.	06/08/25	15/08/25	29
12.	Change the storage class of an existing object to S3 Intelligent-Tiering.	06/08/25	15/08/25	33
13.	Create a new IAM user with programmatic access and S3 read-only permissions.	07/08/25	15/08/25	35
14.	Generate Access Keys for this user. How would they use the AWS CLI to list the contents of your bucket?	07/08/25	15/08/25	41
	Create a bucket policy to allow public	08/08/25	15/08/25	47

15.	read access to all objects in the bucket.			
16.	Enable server access logging for your S3 bucket. Where are the logs stored?	10/08/25	15/08/25	49
17.	Write a simple handler function that returns: "Welcome to VIT BHOPAL!"	11/08/25	15/08/25	52
18.	Launch a Linux EC2 instance into the PublicSubnet.	12/08/25	15/08/25	56
19.	Modify the security group of the public EC2 instance to allow only your IP on port 22.	13/08/25	15/08/25	59
20.	Create and attach an Internet Gateway (IGW) to your VPC.	14/08/25	15/08/25	62

1.Create a new IAM role for EC2 with the policy AmazonS3ReadOnlyAccess.

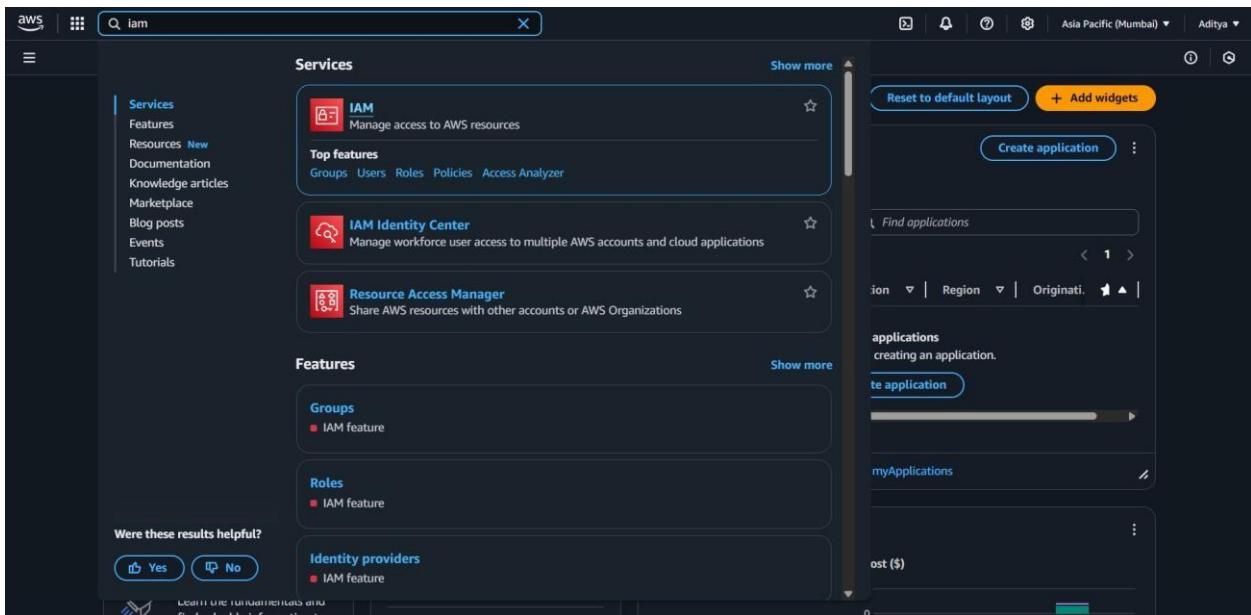
objective:

To create an IAM role that grants an EC2 instance read-only access to Amazon S3 by attaching the built-in AmazonS3ReadOnlyAccess policy.

prerequisite:

- An active AWS account.
- Basic understanding of AWS IAM and EC2 services.
- Access permissions to create IAM roles and attach policies.
- AWS Management Console or AWS CLI access.

Step1 : Search for IAM and click on **IAM (Identity and Access Management)**.



Step2 : In the IAM Dashboard, select **Roles** from the left-hand sidebar.

The screenshot shows the AWS IAM Roles page. On the left, there's a navigation sidebar with options like Dashboard, Access management, and Access reports. The main area displays a table titled 'Roles (7)'. The table has columns for Role name, Trusted entities, and Last activity. The roles listed are: aws-elasticbeanstalk-service-role_NEW, aws-elasticbeanstalk-service-role_Public, aws-elasticbeanstalk-service-role_python, AWSServiceRoleForAutoScaling, AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, and EC2_Role. Most roles have 'aws-elasticbeanstalk' as the trusted entity, with last activity dates ranging from 14 days ago to now.

Step 3 : Click the “Create role” button.

Step 4 :Under "Trusted entity type", choose AWS service

Step5 Under “Use case”, select EC2.

This screenshot shows the 'Create role' wizard at Step 3: Trusted entity type. It lists five options: AWS service (selected), AWS account, Web identity, SAML 2.0 federation, and Custom trust policy. Below this, the 'Use case' section is shown, which is currently set to EC2. A dropdown menu shows 'EC2' selected. The bottom of the screen shows standard AWS navigation links for CloudShell, Feedback, and Copyright information.

Step 6 : click Next ,

In the permissions page, search for the policy: **AmazonS3ReadOnlyAccess**

Select the checkbox next to it and Click **Next**.

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 currently selected), and 'Name, review, and create'. The main area is titled 'Add permissions' and shows a search bar for 'Permissions policies (1/1066)'. A table lists one policy: 'AmazonS3ReadOnlyAccess' (AWS managed, provides read-only access). Below the table is a section for setting a 'permissions boundary - optional'. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

Step 7: : Name and Create the Role.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. The sidebar shows the completed steps: 'Select trusted entity', 'Add permissions', and 'Name, review, and create'. The main area is titled 'Role details' and contains fields for 'Role name' (set to 'amazonS3readonlyaccess') and 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf'). Below this is the 'Step 1: Select trusted entities' section, which displays a JSON-based 'Trust policy' with the following code:

```
1 ~ [ { "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "sts:AssumeRole" ], "Principal": { "Service": [ "ec2.amazonaws.com" ] } } ] }
```

Step 8 : click on the create role.

The screenshot shows the AWS IAM Roles page. At the top, a green banner indicates that 'Role amazonS3readonlyaccess created.' Below this, the title 'Roles (1/8)' is displayed. A table lists the roles, with the newly created 'amazonS3readonlyaccess' role highlighted in blue. The table columns are 'Role name', 'Trusted entities', and 'Last activity'. The 'amazonS3readonlyaccess' role is associated with the 'ec2' service and was created 14 days ago. Other listed roles include 'aws-elasticbeanstalk-service-role_NEW', 'aws-elasticbeanstalk-service-role_Public', 'aws-elasticbeanstalk-service-role_python', 'AWSServiceRoleForAutoScaling', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', and 'EC2_Role', all of which were created 14 days ago.

2.Create a custom policy that allows listing all S3 buckets but denies deleting them.

objective :

To create a custom IAM policy that grants permissions to list all Amazon S3 buckets while explicitly denying any delete operations on them.

prerequisite :

- An active AWS account.
- Knowledge of AWS IAM policies and JSON policy structure.
- Permissions to create and manage IAM policies.
- AWS Management Console or AWS CLI access.

Step 1 : In the IAM Dashboard, click on **Policies** from the left sidebar.

The screenshot shows the AWS IAM Policies page. The left sidebar navigation includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups', 'Users', 'Roles', and 'Policies' selected), 'Identity providers', 'Account settings', and 'Root access management'. Below these are sections for 'Access reports' (with 'Access Analyzer', 'Resource analysis', 'Unused access', 'Analyzer settings', 'Credential report', 'Organization activity', and 'Service control policies'). The main content area displays a table titled 'Policies (1381)'. The table has columns for 'Policy name', 'Type', 'Used as', and 'Description'. Some policy names are truncated with ellipses. The 'Description' column contains brief descriptions such as 'Provides full access to AWS services an...', 'Grants account administrative permissi...', and 'Provides ReadOnly permissions requir...'. A search bar at the top right is set to 'All types'. The bottom of the page includes a URL, copyright information, and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step 2 : Click the “Create policy” button

Step 3 : Switch to JSON Tab.

The screenshot shows the 'Specify permissions' step of the 'Create policy' wizard. The left sidebar shows 'Step 1 Specify permissions' (selected) and 'Step 2 Review and create'. The main content area is titled 'Specify permissions' and contains the instruction 'Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.' Below this is a 'Policy editor' section with a JSON code editor and a 'Visual' tab. The JSON code is as follows:

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "Statement1",  
6       "Effect": "Allow",  
7       "Action": [],  
8       "Resource": []  
9     }  
10   ]  
11 }
```

To the right of the editor is a sidebar with tabs for 'Edit statement', 'Remove', 'Add actions', 'Choose a service' (with a dropdown menu for 'Filter services'), and a list of available services: 'Available', 'AI Operations', 'AMP', 'API Gateway', 'API Gateway V2', 'ARC Region switch', 'ARC Zonal Shift', 'ASC', 'Access Analyzer', and 'Account'.

Step 4: Replace the default content with the following custom policy:

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "s3>ListAllMyBuckets"  
8       ],  
9       "Resource": "*"  
10    },  
11    {  
12      "Effect": "Deny",  
13      "Action": [  
14        "s3>DeleteBucket",  
15        "s3>DeleteObject"  
16      ],  
17      "Resource": "*"  
18    }  
19  ]  
20}  
21
```

Visual JSON Actions

Edit statement

Select a statement

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

+ Add new statement

Step 5: click next.

Step 6: Enter the **Name** of the policy and Click **Create policy**.

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

listingS3bucketsAllowance

Description - optional

Add a short explanation for this policy.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Search

Explicit deny (1 of 447 services)

Service	Access level	Resource	Request condition
S3	Limited: Write	All resources	None

Step 7: Search for your policy name.

IAM > Policies

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies**
- Identity providers
- Account settings
- Root access management New

Access reports

- Access Analyzer
 - Resource analysis New
 - Unused access
 - Analyzer settings
- Credential report
- Organization activity
- Service control policies

CloudShell Feedback

Policy listingS3bucketsAllowance created.

View policy X

Policies (1382) Info

A policy is an object in AWS that defines permissions.

Filter by Type

listing All types 1 match

Policy name	Type	Used as	Description
listingS3bucketsAllowance	Customer managed	None	-

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

The screenshot shows the AWS IAM Policies page. The left sidebar has sections for Dashboard, Access management (User groups, Users, Roles, Policies), and Access reports (Access Analyzer, Credential report, Organization activity, Service control policies). The Policies section is expanded, showing sub-options like Identity providers, Account settings, and Root access management. The main content area shows a green banner at the top indicating a new policy was created: 'Policy listingS3bucketsAllowance created.' Below this is a search bar with 'listing' and a filter bar set to 'All types'. A table lists the single policy: 'listingS3bucketsAllowance' (Customer managed, None, -). At the bottom, there's a copyright notice and links for Privacy, Terms, and Cookie preferences.

3. View the IAM Credential Report.

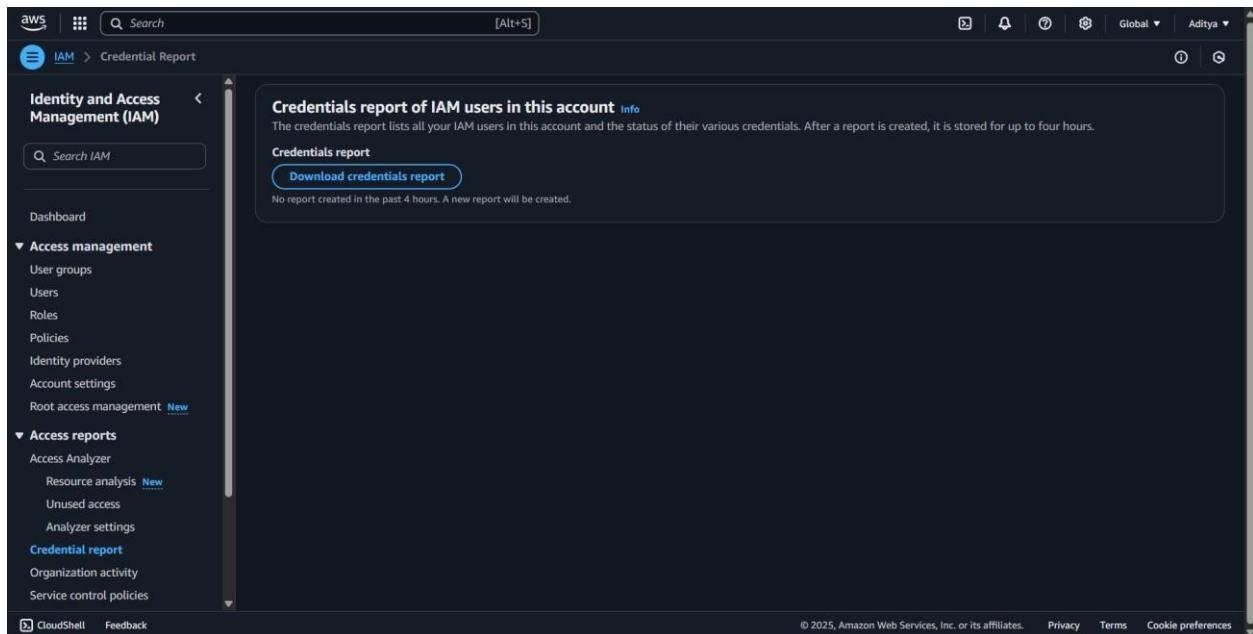
objective:

To generate and view the IAM Credential Report, which provides security-related information about all IAM users in the AWS account.

prerequisite:

- An active AWS account.
- Permissions to access and download the IAM Credential Report.
- AWS Management Console or AWS CLI access.

Step 1: In the IAM Dashboard, select **Credential Report** from the left sidebar.



Step 2 : Click on the “Download Report” button.

Step 3: Open the downloaded file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	user	arn	user_create_password	password	password	mfa_active	access_ke_access_ke_access_ke_access_ke_access_ke_access_ke_access_ke_access_ke_access_ke_access_ke_access_ke_cert_1_act												
2	<root_acc_arn:aws:ia	2025-07-0	TRUE	2025-08-0	2025-07-0	not_suppo	TRUE	TRUE	2025-07-2	2025-07-2	ap-south-1:glacier								
3	Addityo_Ram:arn:aws:ia	2025-07-0	FALSE	N/A	N/A	N/A	FALSE	FALSE	N/A	N/A	N/A								
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			

4. Create a Service Control Policy (SCP) using AWS Organizations that blocks deletion of EC2 instances.

objective :

To create a Service Control Policy (SCP) in AWS Organizations that prevents the deletion of EC2 instances across specified AWS accounts or organizational units.

prerequisite :

- An active AWS account with AWS Organizations enabled.
- Permissions to create and manage Service Control Policies.
- Basic knowledge of SCP JSON structure and EC2 actions.
- AWS Management Console or AWS CLI access.

Step 1 : In the AWS Console, go to **Organizations** from the **Services** menu.

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar with the query "organization activity", and various global settings like "Global" and "Aditya". The left sidebar has a "Services" section with links for IAM, Identity and Access Management, Access management, Access repository, Credential report, and Service control policies. The main content area displays the "Services" menu with three items: "AWS Organizations" (selected), "Resource Groups & Tag Editor", and "Trusted Advisor". Below this is the "Features" section with "Organize Accounts", "Organization activity" (which is highlighted in red), and "Organizational view". At the bottom of the sidebar, there's a feedback section asking "Were these results helpful?" with "Yes" and "No" buttons, and a footer with copyright information and links to Privacy, Terms, and Cookie preferences.

Step 2: In the left-hand menu, click on **Policies**.

The screenshot shows the AWS Organizations Policies page. A green success message at the top right says "You successfully created an AWS organization." On the left, a sidebar lists "AWS accounts", "Multi-party approval", "Services", "Policies" (which is selected), "Settings", and "Get started". Below the sidebar, the "Organization ID" is listed as o-24yk4pzae9. The main content area is titled "Policies" and contains a section about "Introducing resource control policies (RCPs)". It lists four supported policy types: "AI services opt-out policies", "Backup policies", "Chat applications policies", and "Declarative policies for EC2". Each policy type has a status indicator (disabled) and a "Learn more" link. At the bottom right of the main content area, there are links for "CloudShell", "Feedback", "© 2025, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Step 3: Ensure that **Service Control Policies (SCPs)** are enabled for your organization.

If not enabled, click on **Enable SCPs**.

The screenshot shows the AWS Organizations Service control policies page. A green success message at the top right says "Service control policies have been enabled." On the left, a sidebar lists "AWS accounts", "Multi-party approval", "Services", "Policies" (selected), "Settings", and "Get started". Below the sidebar, the "Organization ID" is listed as o-24yk4pzae9. The main content area is titled "Service control policies" and contains a section about "Available policies". It lists one policy: "FullAWSAccess" (AWS managed policy) which "Allows access to every operation". At the bottom right of the main content area, there are links for "Actions", "Create policy", "CloudShell", "Feedback", "© 2025, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Step 4: **Create a New SCP by Clicking on the “Create policy” button.**

Step 5 : Under the **JSON** tab, enter the following SCP:

The screenshot shows the AWS Organizations console with the path: AWS Organizations > Policies > Service control policies > Create new service control policy. On the left, there's a sidebar with AWS accounts, Multi-party approval, Services, Policies (selected), Settings, and Get started. Below that is the Organization ID: 0-24yk4pzae9. The main area has a text input for tags, an 'Add tag' button, and a note about adding up to 50 more tags. A code editor shows the JSON for a deny statement:

```
1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Effect": "Deny",
5         "Action": "ec2:TerminateInstances",
6         "Resource": "*"
7     }
8 ]
9
10 ]
```

To the right, there's an 'Edit statement' section with a 'Select a statement' dropdown and a note to select an existing statement or add a new one, plus a '+ Add new statement' button. At the bottom right, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Step 6: click next.

The screenshot shows the AWS Organizations console with the path: AWS Organizations > Policies > Service control policies. A green success message at the top says 'Successfully created the policy named 'organization''. Below it, the 'Service control policies' section shows a table of available policies:

Name	Kind	Description
FullAWSAccess	AWS managed policy	Allows access to every operation
organization	Customer managed policy	-

At the bottom right, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

5. Create a new security group that allows SSH (port 22) access only from your public IP.

objective:

To enhance EC2 instance security by configuring a security group that restricts SSH (port 22) access to only the trusted public IP of the user.

prerequisite:

- An active AWS account.
- Knowledge of the current public IP address.
- Permissions to create and manage security groups.
- AWS Management Console or AWS CLI access.

Step 1: Go to the EC2 Dashboard

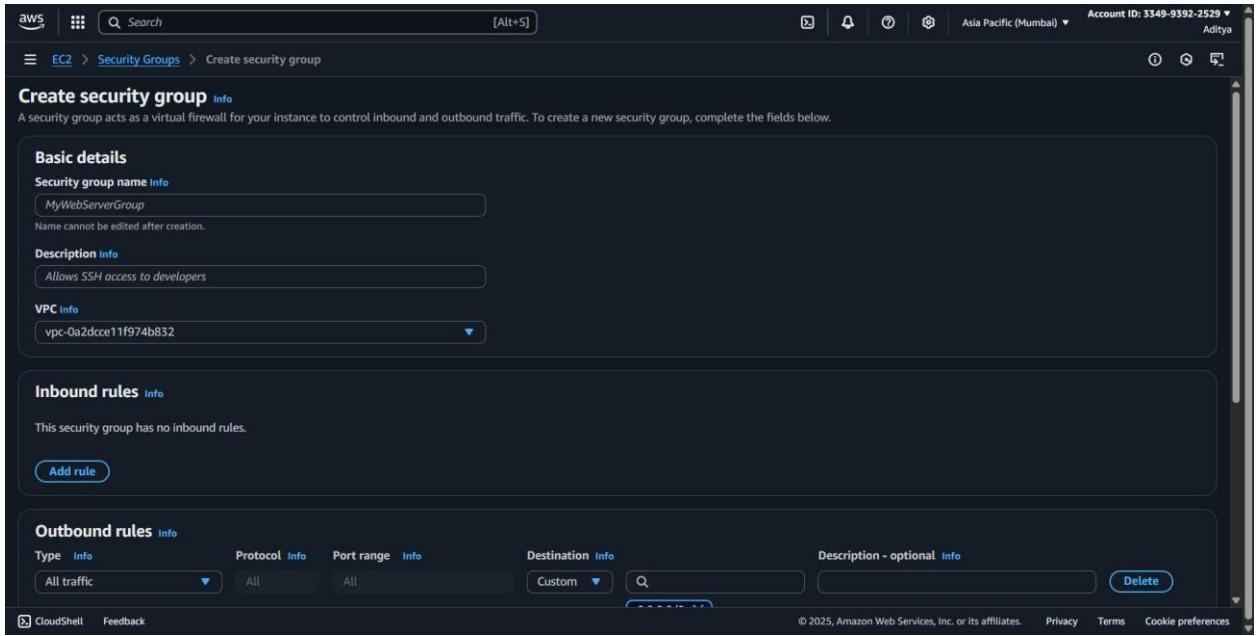
Step2: Open the Security Groups Page.

The screenshot shows the AWS EC2 Security Groups page. The left sidebar includes options like Launch Templates, 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 selected), Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and CloudWatch Metrics. The main content area displays a table titled "Security Groups (3) Info" with columns: Name, Security group ID, Security group name, VPC ID, and Description. The table lists three entries:

Name	Security group ID	Security group name	VPC ID	Description
-	sg-07a9ecf73bb358bd	default	vpc-0a2dcce11f974b832	default VPC secur
-	sg-0d0ba0168270fa257	default	vpc-002fa8defa3a88021	default VPC secur
-	sg-02d28a659073c2317	launch-wizard-1	vpc-0a2dcce11f974b832	launch-wizard-1 c

A "Select a security group" dropdown is visible at the bottom of the table. The top right corner shows the Account ID: 5349-9392-2529, Region: Asia Pacific (Mumbai), and a user name Aditya.

Step 3: Create a Security Group.



Step 4: Enter Basic Details

Security group name: SSH-Access-From-My-IP (you can choose your own name).

VPC: Select the VPC you want to use (usually the default VPC).

Step 5: Add an Inbound Rule for SSH

In Inbound rules, click Add rule.

Type: Select SSH.

Protocol: Will be auto-filled as TCP.

Port range: Will be auto-filled as 22.

EC2 > Security Groups > Create security group

Name mySecurityGroup
Name cannot be edited after creation.

Description Allows SSH access to developers

VPC vpc-0a2dcce11f974b832

Inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Custom	

Add rule

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

Add rule

⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

Step 6: Leave Outbound Rules as Default.

Step 7: Create the Security Group.

EC2 > Security Groups > sg-05e056db0a91ca09b - mySecurityGroup

Security group (sg-05e056db0a91ca09b | mySecurityGroup) was created successfully

sg-05e056db0a91ca09b - mySecurityGroup

Details

Security group name mySecurityGroup	Security group ID sg-05e056db0a91ca09b	Description Allows SSH access only from my public IP
Owner 334993922529	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

VPC ID vpc-0a2dcce11f974b832

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0dba513a182f6690	-	SSH	TCP	22

6. Open PuTTYgen. Load your .pem key file and convert it to .ppk.

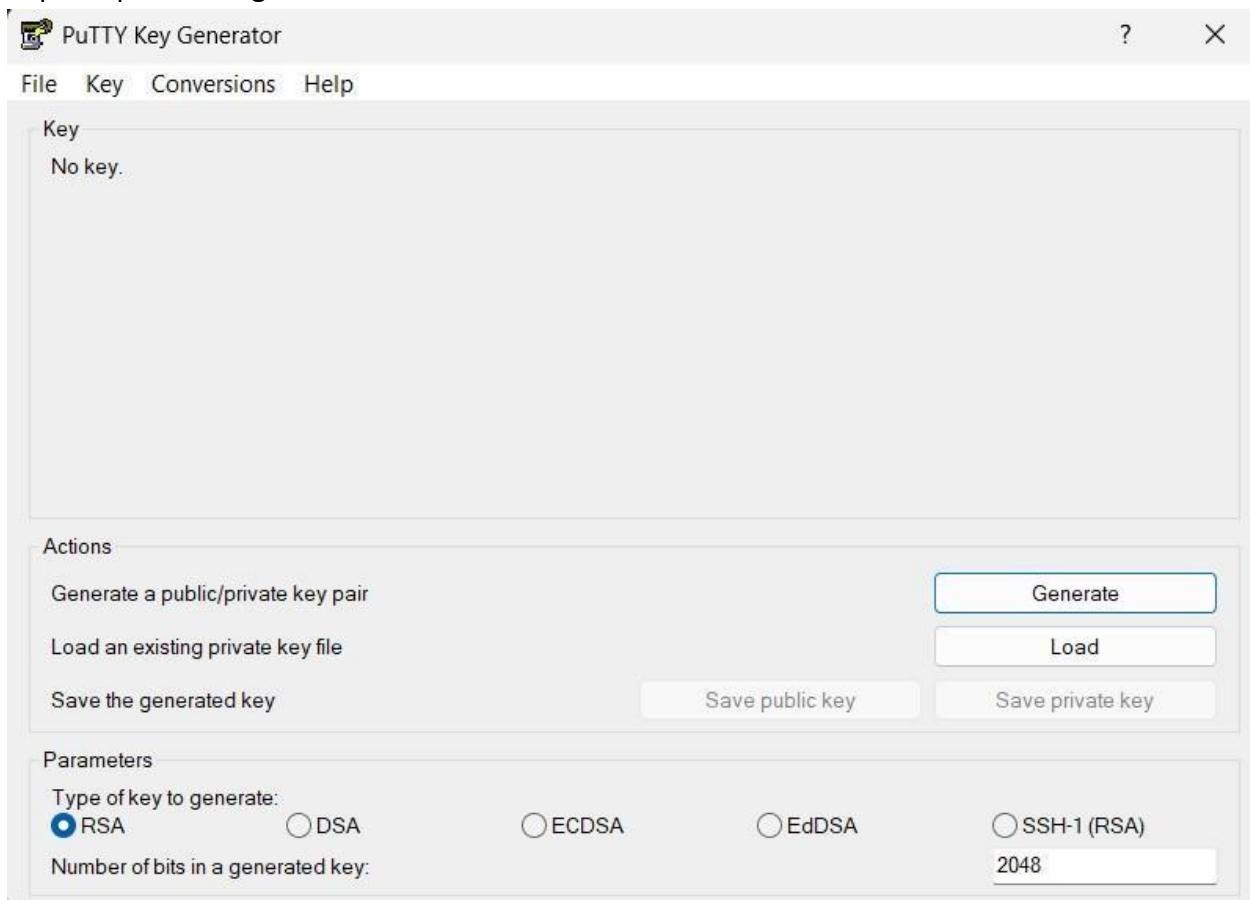
Objective:

To enable secure SSH access from PuTTY by converting an AWS-generated .pem key file into the .ppk format using PuTTYgen.

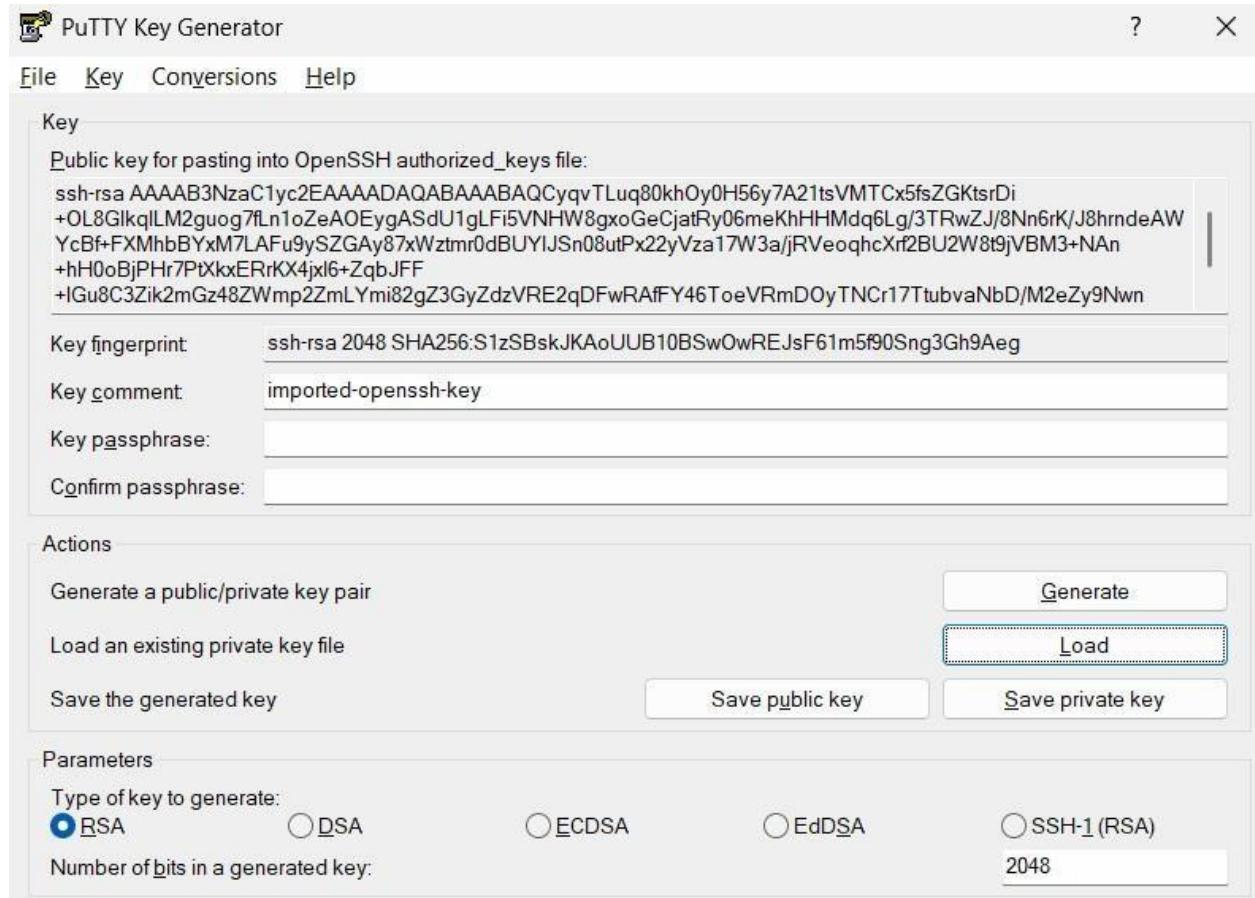
Prerequisite:

- PuTTYgen installed on the system.
- An existing .pem private key file downloaded from AWS.
- Basic understanding of SSH authentication.

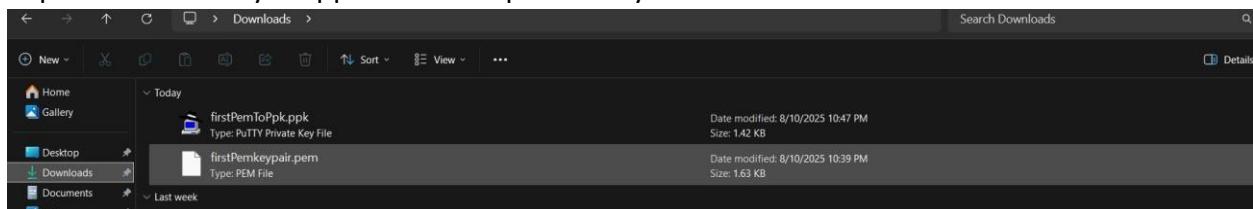
Step 1: Open PuTTYgen



Step 2: Load .pem file.



Step 3: Save the Key as .ppk Click Save private key and then check in downloads.



7.Create a simple text file on your EC2 instance named test.txt using the nano or vi editor.

Objective:

To practice basic file creation and editing on an EC2 instance by using command-line text editors like nano or vi to create a file named test.txt.

Prerequisite:

- An active AWS EC2 instance.
- SSH access to the EC2 instance.

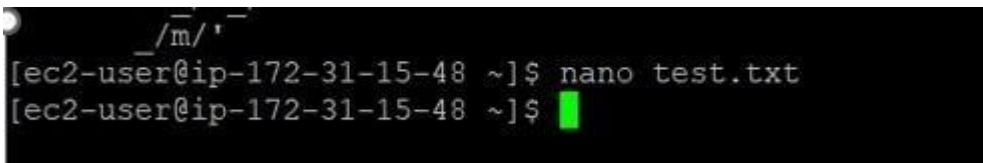
- Basic knowledge of Linux command-line operations.
- Familiarity with nano or vi text editors.

Step 1: open putty



```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-15-48 ~]$
```

Step 2: open nano editor



```
[ec2-user@ip-172-31-15-48 ~]$ nano test.txt
[ec2-user@ip-172-31-15-48 ~]$
```

Step 3: Type your content and then save and exit.



```
/m/
[ec2-user@ip-172-31-15-48 ~]$ nano test.txt
[ec2-user@ip-172-31-15-48 ~]$ nano test.txt
[ec2-user@ip-172-31-15-48 ~]$ Cat test.txt
-bash: Cat: command not found
[ec2-user@ip-172-31-15-48 ~]$ cat test.txt
This is a test file
[ec2-user@ip-172-31-15-48 ~]$
```

8. Access the environment's Configuration page. List any two resources (e.g., EC2, Load Balancer) that were automatically provisioned.

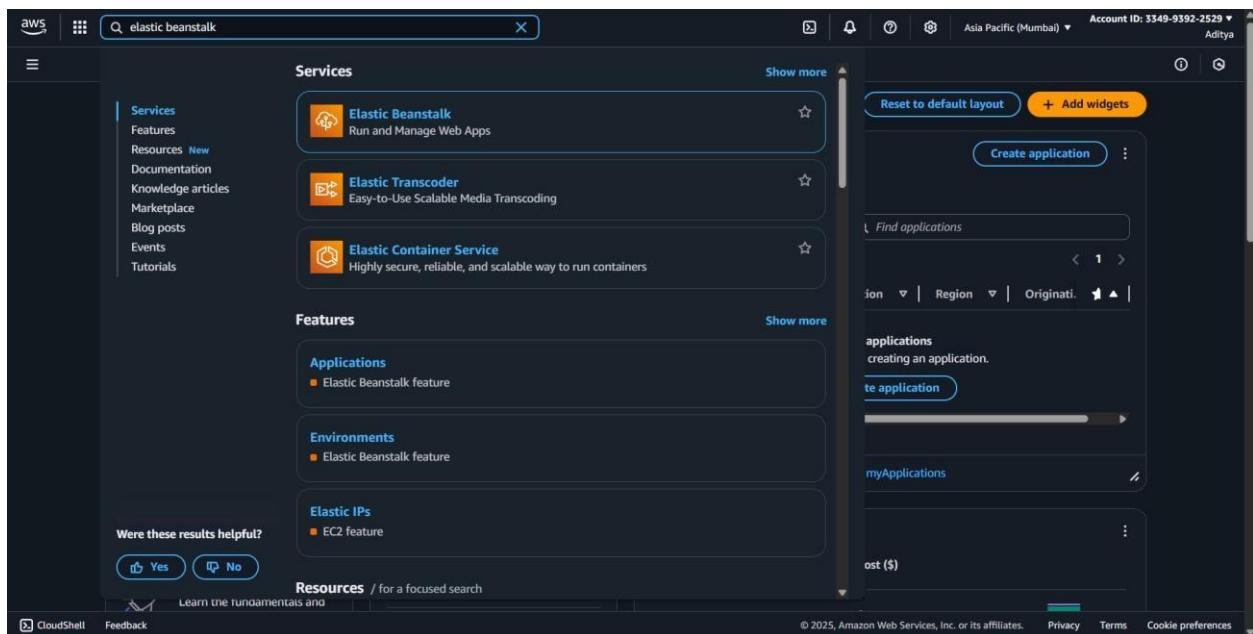
Objective:

To explore the environment's Configuration page and identify resources automatically provisioned by AWS, enhancing understanding of environment setup and resource management.

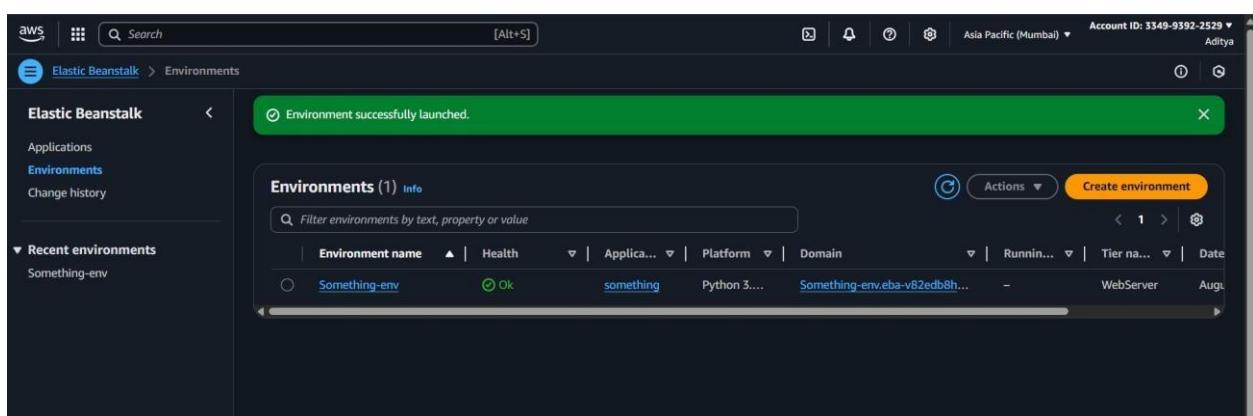
Prerequisite:

- An active AWS account.
- A deployed environment in AWS (e.g., Elastic Beanstalk).
- Permissions to view environment configuration details.
- AWS Management Console access.

Step 1: Go to AWS Management Console and type elastic beanstalk.



Step 2: click the environment you have created.



Step 3: click configuration.

The screenshot shows the AWS Elastic Beanstalk Configuration page for an environment named 'Something-env'. A green success message at the top states 'Environment successfully launched.' The main section is titled 'Configuration' with tabs for 'Service access', 'Networking and database', and 'Instance traffic and scaling'. Under 'Service access', it shows a service role (arn:aws:iam::33493922529:role/aws-elasticbeanstalk-service-role-2ndtime) and an EC2 instance profile (EC2_Role). Under 'Networking and database', it shows a VPC (vpc-0a2dcce11f974b832), a public IP address (true), and instance subnets (subnet-04032afa9d53504f0). Under 'Instance traffic and scaling', it shows instances. The left sidebar lists options like Applications, Environments, Change history, Application versions, Saved configurations, Environment: Something-env (with Go to environment, Configuration, Events, Health, Logs, Monitoring, Alarms, Managed updates, Tags), and Recent environments. The top right corner shows account information (Account ID: 3349-9392-2529, Aditya) and navigation icons.

Step 4: On the **Configuration** page, scroll and check the categories such as:

Instances → shows the EC2 instances that were provisioned.

Load balancer → shows the Elastic Load Balancer created.

Security → shows security groups created for the environment.

9. Create a new volume from your snapshot.

Objective:

To utilize an existing EBS snapshot to create a new storage volume, demonstrating data restoration and replication capabilities in AWS.

Prerequisite:

- An active AWS account.
- An existing EBS snapshot available.
- Permissions to create and manage EBS volumes.
- AWS Management Console or AWS CLI access.

Step 1: Go to the EC2 Dashboard

Step 2: open the snapshots page Click on the checkbox of your snapshot

The screenshot shows the AWS EC2 Snapshots page. On the left, there's a navigation sidebar with options like Dashboard, Instances, Images, and Elastic Block Store. Under 'Elastic Block Store', 'Snapshots' is selected. The main area displays a table titled 'Snapshots (1/1) Info'. A single row is listed, representing a completed snapshot. The snapshot details are as follows:

Snapshot ID	Full snapshot size	Progress	Snapshot status
snap-0adf77492744a02f2	0 B	100%	Completed

Below the table, a detailed view for the snapshot is shown with tabs for Details, Snapshot settings, Storage tier, and Tags. The 'Details' tab is active, displaying information such as Snapshot ID, Owner, Started time, Description, and Source volume.

Step 3: With the snapshot selected, click the Actions dropdown then choose Create volume.

This screenshot is similar to the previous one, showing the EC2 Snapshots page. However, the Actions dropdown menu is now open over the selected snapshot row. The visible options in the dropdown include:

- Create volume from snapshot
- Create image from snapshot
- Copy snapshot
- Launch copy duration calculator
- Delete snapshot
- Manage tags
- Snapshot settings
- Archiving

The 'Create volume from snapshot' option is highlighted, indicating it is the next step to be performed.

Step 4: check in the volume section

Volumes (1/5) Info

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Source volume ID	Create
vol-054657dfaa755c252	gp3	10 GiB	3000	125	snap-0adf774...	-	-	2025/
vol-076aa25342e697ae0	gp3	8 GiB	3000	125	snap-0bf8c38...	-	-	2025/
vol-05fa5a515859add77	gp3	8 GiB	3000	125	snap-0bf8c38...	-	-	2025/
vol-0157fb462c1e28f4d	gp3	10 GiB	3000	125	-	-	-	2025/
vol-09327595a195c1310	gp3	8 GiB	3000	125	snap-0bf8c38...	-	-	2025/

Volume ID: vol-054657dfaa755c252

Details **Status checks** **Monitoring** **Tags**

Volume ID vol-054657dfaa755c252	Size 10 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-1b	Created Mon Aug 11 2025 00:04:19 GMT+0530 (India Standard Time)	Multi-Attach enabled No
Attached resources	Outposts ARN	Managed	Operator

10. Use the Amazon Data Lifecycle Manager (DLM) to create a scheduled backup policy.

Objective:

To automate EBS volume backups by creating a scheduled snapshot policy using Amazon Data Lifecycle Manager (DLM), ensuring regular data protection without manual intervention.

Prerequisite:

- An active AWS account.
- Existing EBS volumes to back up.
- Permissions to create and manage DLM lifecycle policies.
- AWS Management Console access.

Step 1: Go to EC2 dashboard

Step 2: Open Lifecycle Manager

The screenshot shows the AWS EC2 Compute page. On the left, there's a navigation sidebar with sections like Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled "Amazon Data Lifecycle Manager" and describes its purpose: "Automate the creation, retention, copy and deletion of snapshots and AMIs". Below this, there's a section titled "Benefits and features" with two items: "Automated snapshot and AMI creation" and "Fast snapshot restore integration". A large callout box on the right is titled "Create new lifecycle policy" and contains options for "Create custom or default policy" (with "Custom policy" selected), "Policy type" (set to "EBS snapshot policy"), and a "Next step" button.

Step 3: click create lifecycle policy and choose EBS snapshot policy.

The screenshot shows the 'Specify settings' step of the 'Create lifecycle policy' wizard. The left sidebar lists three steps: Step 1 (Specify settings, selected), Step 2 (Configure schedule 1 - Schedule 1), and Step 3 (Review and create). The main area is titled 'Specify settings'.

Target resources (Info): Specify the resources that are to be targeted by this policy.

Target resource types: Select the type of resources that are to be targeted. The 'Volume' option is selected.

Target resource tags: All resources of the selected type that have at least one of these tags will be targeted by the policy. Two tags are entered: 'Backup' and 'yes'.

Description

Policy description: My Data Lifecycle Manager policy

IAM role (Info): This policy must be associated with an IAM role that has the appropriate permissions. If you choose to create a new role, you must grant relevant role permissions and set up trust relationships correctly. If you are unsure of what role to use, choose Default role.

Bottom navigation bar: CloudShell, Feedback, Search, [Alt+S], Account ID: 3349-9392-2529, Aditya, Asia Pacific (Mumbai), © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

This screenshot is identical to the one above, but the 'yes' tag has been added to the 'Backup' tag in the target resource tags section.

Bottom navigation bar: CloudShell, Feedback, Search, [Alt+S], Account ID: 3349-9392-2529, Aditya, Asia Pacific (Mumbai), © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

Step 4: Click next

Step 5: Configure Schedule.

The screenshot shows the AWS Lambda 'Create function' wizard, Step 2: Set runtime and permissions. It includes sections for 'Runtime', 'AWS Lambda execution role', 'AWS Lambda trigger role', and 'Lambda layers'. A note at the bottom states: 'Lambda layers are optional. You can add up to 10 layers to your function.'

Step 6: Review Policy Settings

Step 7: Click **Create policy**.

The screenshot shows the AWS Lambda 'Create function' wizard Step 7: Review configuration. It displays the 'Function configuration' section with fields for 'Function name', 'Memory size', 'Timeout', 'Runtime', 'Handler', and 'Environment variables'. A note at the bottom says: 'You can always change these settings later in the Lambda console.'

11. Upload another file and select S3 One Zone-IA or S3 Glacier as the storage class.

Objective:

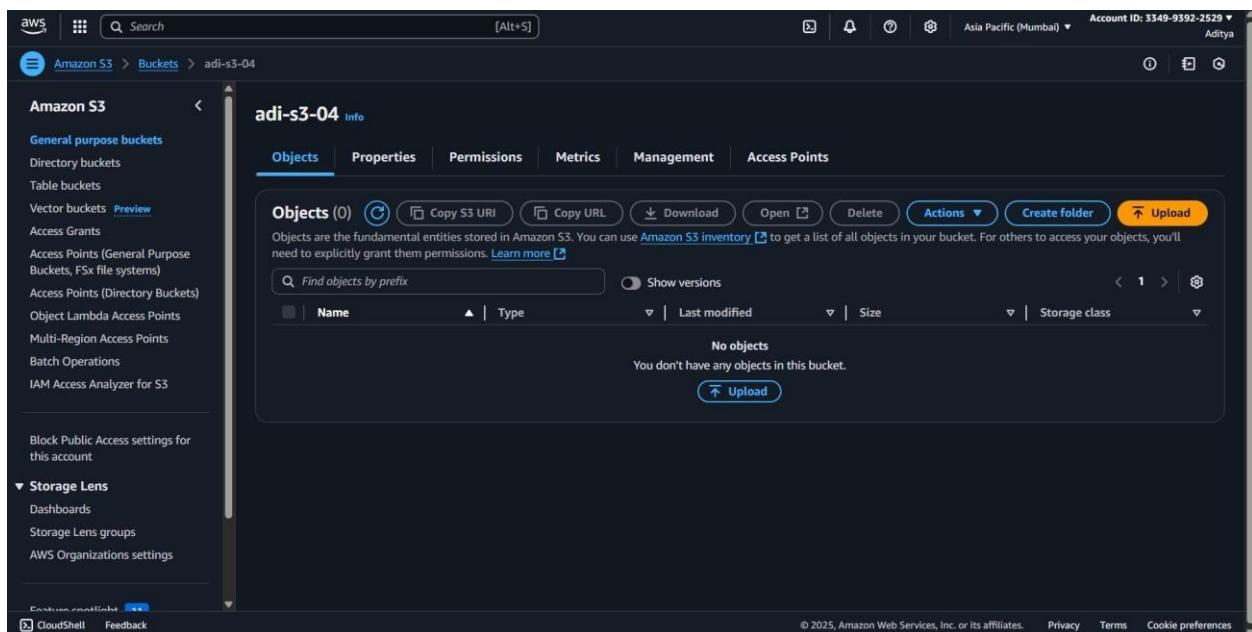
To optimize storage costs by uploading a file to Amazon S3 and selecting a cost-effective storage class such as S3 One Zone-IA or S3 Glacier based on data access needs.

Prerequisite:

- An active AWS account.
- An existing S3 bucket.
- Permissions to upload objects and change storage classes.
- AWS Management Console or AWS CLI access.

Step 1: Go to the S3 Dashboard.

Step 2: Click the bucket name where you want to upload the file.



Step 3: On the bucket page, click the **Upload** button.

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a search bar and navigation links for 'Amazon S3 > Buckets > adi-s3-04 > Upload'. On the right, account information 'Account ID: 3349-9392-2529' and 'Aditya' are displayed. Below the navigation, a section titled 'Upload Info' contains instructions: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more [?]' and a large dashed box for dragging files. A 'Files and folders (0)' table is shown with columns for Name, Folder, Type, and Size, and a note: 'No files or folders. You have not chosen any files or folders to upload.' Buttons for 'Remove', 'Add files', and 'Add folder' are available. A 'Destination' section follows, showing 'Destination s3://adi-s3-04 [?]' and 'Destination details' (Bucket settings). The bottom includes links for 'Permissions', 'CloudShell', and 'Feedback', along with copyright and legal information.

Step 4: In the **Files and folders** section, click **Add files**.

This screenshot shows the same AWS S3 'Upload' interface after a file has been added. The 'Files and folders (1 total, 2.6 MB)' table now lists '2.png' with a size of '2.6 MB'. The rest of the interface remains the same, including the 'Destination' section and the bottom navigation links.

Step 5: Scroll down to **Properties**.

The screenshot shows the 'Properties' section of an AWS S3 bucket named 'adi-s3-04'. The 'Storage class' dropdown is open, displaying various options with their descriptions and characteristics. The 'Standard' option is selected, highlighted with a blue border. Other options shown include Intelligent-Tiering, Standard-IA, One Zone-IA, Glacier Instant Retrieval, and Glacier Flexible Retrieval (formerly Glacier). The table includes columns for Storage class, Designed for, Bucket type, Availability Zones, Min storage duration, Min billable object size, Monitoring and auto-tiering fees, and Retrieval fees.

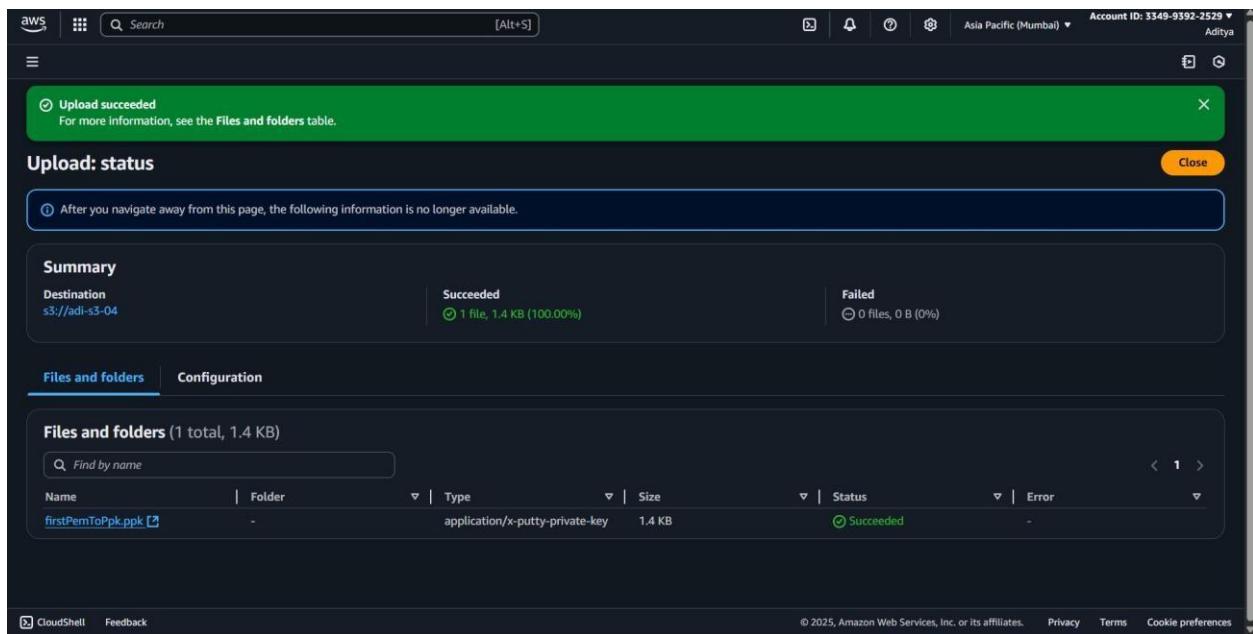
Storage class	Designed for	Bucket type	Availability Zones	Min storage duration	Min billable object size	Monitoring and auto-tiering fees	Retrieval fees
Standard	Frequently accessed data (more than once a month) with milliseconds access	General purpose	≥ 3	-	-	-	-
Intelligent-Tiering	Data with changing 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 6: select one zone

This screenshot is identical to the previous one, showing the 'Properties' section of the AWS S3 bucket 'adi-s3-04'. The 'Storage class' dropdown is open, and the 'One Zone-IA' option is now selected, highlighted with a blue border. The other options (Standard, Intelligent-Tiering, Standard-IA, Glacier Instant Retrieval, Glacier Flexible Retrieval, and Glacier Deep Archive) are visible but not selected. The table structure remains the same, detailing the characteristics of each storage class.

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
Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	General purpose	≥ 3	180 days	-	-	Per-GB fees apply
Reduced	Noncritical, frequently accessed data with	General	-	-	-	-	-

Step 7 : Scroll down and click Upload.



12. Change the storage class of an existing object to S3 Intelligent-Tiering.

Objective:

To improve storage efficiency by changing the storage class of an existing S3 object to S3 Intelligent-Tiering, enabling automatic cost optimization based on access patterns.

Prerequisite:

- An active AWS account.
- An existing S3 bucket with stored objects.
- Permissions to modify object storage classes.
- AWS Management Console or AWS CLI access.

Step 1: Go to the S3 Dashboard

Step 2: Open the Bucket

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various navigation options like 'General purpose buckets', 'Storage Lens', and 'CloudShell'. The main area is titled 'adi-s3-04' and shows a table of objects. There is one object listed: 'firstPemToPpk.ppk', which is a file type ('ppk') last modified on August 12, 2025, at 22:26:13 (UTC+05:30). The file size is 1.4 KB and is stored in the 'One Zone-IA' storage class.

Step 3: Select bucket

This screenshot is identical to the previous one, but it includes a checkbox next to the file name 'firstPemToPpk.ppk' in the object list, indicating that the file has been selected.

Step 4 : Click the **Actions** dropdown (top-right) and Select **Change storage class**.

Step 5: choose the s3 intelligent tiering

The screenshot shows the AWS S3 console with the path: Amazon S3 > Buckets > adi-s3-04 > Edit storage class. A table lists various storage classes with their characteristics and fees. The 'Intelligent-Tiering' row is highlighted with a blue background, indicating it is selected. The table columns include Storage class, Designed for, Availability Zones, Min storage duration, Min billable object size, Monitoring and auto-tiering fees, and Retrieval fees.

Storage class	Designed for	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	≥ 3	-	-	-	-
Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-	Per-object fees apply for objects ≥ 128 KB	-
Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	128 KB	-	Per-GB fees apply
One Zone-IA	Recreatable, infrequently accessed data (once a month) with milliseconds access	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	≥ 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	≥ 3	90 days	-	-	Per-GB fees apply
Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days	-	-	Per-GB fees apply
Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-	-	-	Per-GB fees apply

Specified objects
Find objects by name

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

Step 6: save changes

The screenshot shows the AWS S3 console with the path: Amazon S3 > Buckets > adi-s3-04 > Edit storage class: status. A green success message box at the top says "Successfully edited storage class" and "View details below." Below it, a summary table shows the status of edits across different categories: Source (s3://adi-s3-04), Successfully edited (1 object, 1.4 KB), and Failed to edit (0 objects). The "Failed to edit" tab is selected, showing a table with no failed edits.

Name	Type	Last modified	Size	Error
No failed edits.				

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

13. Create a new IAM user with programmatic access and S3 read-only permissions.

Objective:

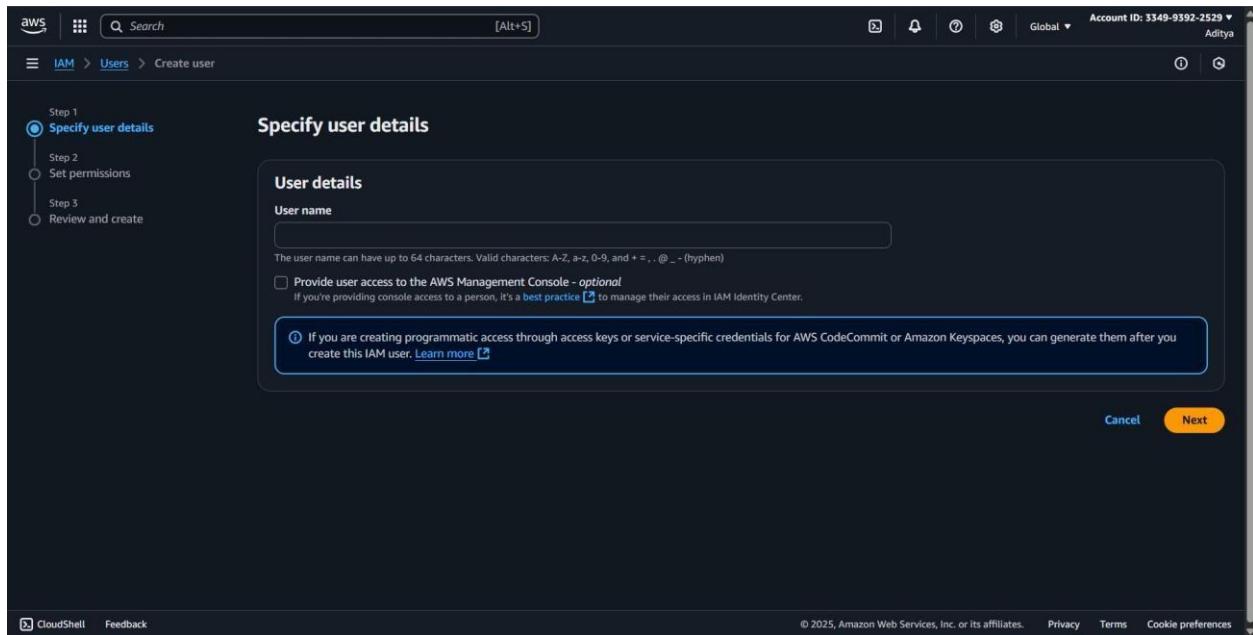
To set up a new IAM user with programmatic access and grant S3 read-only permissions, enabling secure API or CLI-based interactions without write privileges.

Prerequisite:

- An active AWS account.
- Permissions to create IAM users and attach policies.
- Basic understanding of programmatic access (Access Keys).
- AWS Management Console or AWS CLI access.

Step 1: In AWS Management Console, search for **IAM** and open it.

Step 2: In the left-hand menu, click **Users** and Click **Create user**.



Step 3: set user details

Screenshot of the AWS IAM 'Create user' wizard Step 1: Specify user details.

User details

User name: s3-agair

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
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#).

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Step 4: Choose Attach policies directly.

Screenshot of the AWS IAM 'Create user' wizard Step 2: Set permissions.

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

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 (1387)

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

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
Adi_policy	Customer managed	1
AdministratorAccess	AWS managed - job function	2
AdministratorAccess-Amplify	AWS managed	0
AdministratorAccess-AWSFleetLaunch	AWS managed	0

Create policy

Step 5: Check the box for **AmazonS3ReadOnlyAccess**.

Step 1
 Specify user details
 Set permissions
 Review and create

Set permissions
 Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- 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 (1/1387)
 Choose one or more policies to attach to your new user.

Filter by Type	
<input type="text" value="amazons3rea"/>	All types
<input checked="" type="checkbox"/> Policy name	Type
<input checked="" type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed
1	

Set permissions boundary - optional

[Create policy](#)

[Cancel](#) [Previous](#) [Next](#)

Step 6: click next and click create user

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

[Learn more](#) | [Watch how it works](#)

Users (2) Info
 An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
Aditya_Raj	/	1	-	-	-	-
s5-again	/	0	-	-	-	-

[View user](#) [Delete](#) [Create user](#)

Step 7: After creation, click the user name you just created Go to **Security credentials** tab.

Summary

ARN: arn:aws:iam::33499392259:user/s3-again

Created: August 12, 2025, 22:41 (UTC+05:30)

Console access: Disabled

Last console sign-in: -

Access key 1: Create access key

Permissions | Groups | Tags | **Security credentials** | Last Accessed

Console sign-in

Console sign-in link: https://33499392259.sigin.aws.amazon.com/console

Console password: Not enabled

Multi-factor authentication (MFA) (0)

No MFA devices. Assign an MFA device to improve the security of your AWS environment

Assign MFA device

Step 8: In the Access keys section, click Create access key.

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Use case

Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

Other
Your use case is not listed here.

Cancel Next

Step 9: select command line interface

The screenshot shows the AWS IAM 'Create access key' interface. At the top, there are three navigation steps: Step 2 - optional, Step 3, and Retrieve access keys. Below these, a 'Use case' section lists several options with descriptions:

- Command Line Interface (CLI)**: You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**: You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**: You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service**: You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS**: You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- Other**: Your use case is not listed here.

At the bottom of the 'Use case' section, there is a yellow warning box titled 'Alternatives recommended' containing two items:

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

At the very bottom of the page, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Step 10: Download the .csv file containing the **Access key ID** and **Secret access key**.

aws Search [Alt+S]

IAM > Users > s3-again > Create access key

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1 Access key best practices & alternatives

Step 2 - optional Set description tag

Step 3 Retrieve access keys

Retrieve access keys Info

Access key If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAU37ZRTHQX2OGTICC	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file Done

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

i-9392-2529 Aditya

Recent download history s3-again_accessKeys.csv 99 B • Done

Full download history

14. Generate Access Keys for this user. How would they use the AWS CLI to list the contents of your bucket?

Objective:

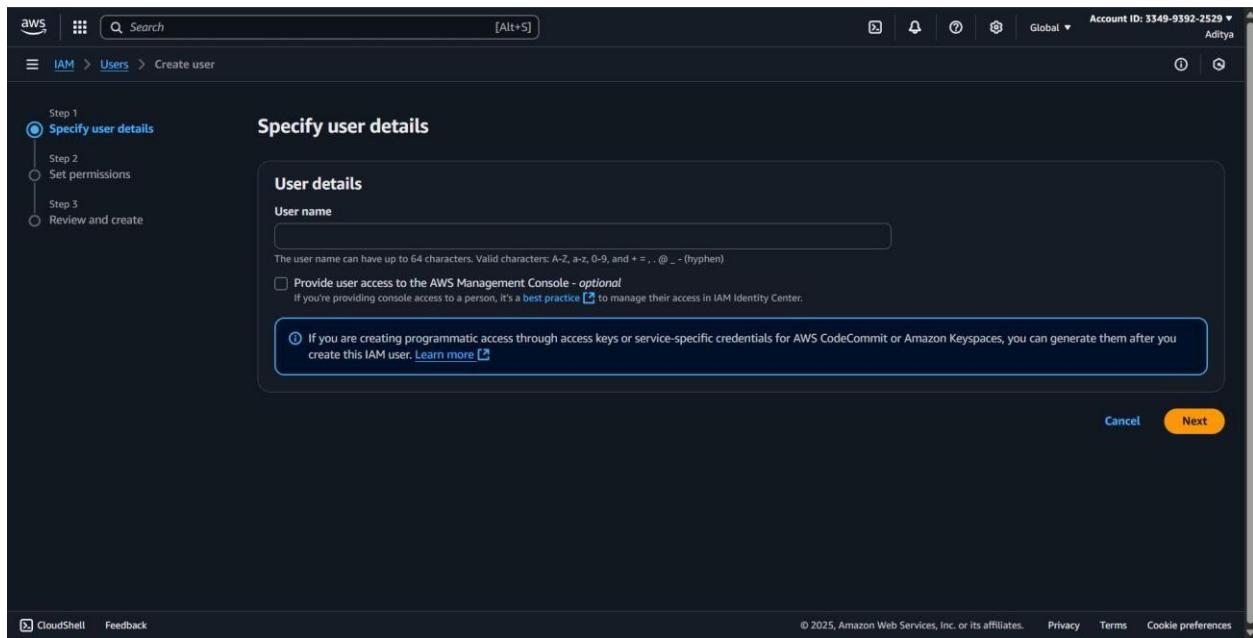
To generate access keys for an IAM user and demonstrate their usage in AWS CLI to securely list the contents of an S3 bucket.

Prerequisite:

- An active AWS account.
- An IAM user with S3 read permissions.
- AWS CLI installed and configured.
- Basic knowledge of AWS CLI commands.

Step 1: In AWS Management Console, search for **IAM** and open it.

Step 2: In the left-hand menu, click **Users** and Click **Create user**.



Step 3: set user details

Screenshot of the AWS IAM 'Create user' wizard Step 1: Specify user details.

User details

User name: s3-agair

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
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#).

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Step 4: Choose Attach policies directly.

Screenshot of the AWS IAM 'Create user' wizard Step 2: Set permissions.

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

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 (1387)

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

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
Adi_policy	Customer managed	1
AdministratorAccess	AWS managed - job function	2
AdministratorAccess-Amplify	AWS managed	0
AdministratorAccess-AWSFleetLaunch	AWS managed	0

Create policy

Step 5: Check the box for **AmazonS3ReadOnlyAccess**.

The screenshot shows the 'Set permissions' step of the 'Create user' wizard. It includes a sidebar with steps: Step 1 (Specify user details), Step 2 (Set permissions, which is selected), and Step 3 (Review and create). The main area has a heading 'Permissions options' with three choices: 'Add user to group' (radio button), 'Copy permissions' (radio button), and 'Attach policies directly' (radio button, selected). Below this is a section titled 'Permissions policies (1/1387)' showing a search bar with 'amazons3rea', a filter bar, and a table with one item: 'AmazonS3ReadOnlyAccess' (AWS managed, Type: Attached entities). A note says 'Choose one or more policies to attach to your new user.' At the bottom are buttons for 'Set permissions boundary - optional', 'Cancel', 'Previous', and 'Next'.

Step 6: click next and click create user

The screenshot shows the 'Users' page in the AWS IAM console. A green success message at the top says 'User created successfully'. Below it, there are four icons: 'One-time set up for workforce user access', 'Centrally manage access to multiple AWS accounts', 'Provide access centrally to the cloud applications your workforce uses', and 'All with one-click access through a simple web portal'. The main table lists two users: 'Aditya_Raj' and 's3-again'. The table columns include 'User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', and 'Console last sign-in'. At the bottom are buttons for 'View user', 'Delete', and 'Create user'.

Step 7: After creation, click the user name you just created Go to **Security credentials** tab.

The screenshot shows the AWS IAM User Summary page for a user named 's3-again'. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 3349-9392-2529, Aditya). The left sidebar has sections for Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies), and IAM Identity Center/AWS Organizations.

The main content area displays the 'Summary' tab for the user 's3-again'. It shows the ARN (arn:aws:iam::334993922529:user/s3-again), creation date (August 12, 2025, 22:41 (UTC+05:30)), and access status (Console access: Disabled, Last console sign-in: -). There is a link to 'Create access key' under the access key section.

The 'Security credentials' tab is selected, showing the 'Console sign-in' section with a 'Console sign-in link' (https://334993922529.signin.aws.amazon.com/console) and a note that the 'Console password' is 'Not enabled'. A 'Enable console access' button is available.

The 'Multi-factor authentication (MFA)' section shows 0 MFA devices assigned. It includes buttons for 'Remove', 'Resync', and 'Assign MFA device'.

At the bottom, there are tabs for 'Permissions', 'Groups', 'Tags', 'Security credentials', and 'Last Accessed'. A note states 'No MFA devices. Assign an MFA device to improve the security of your AWS environment' with a 'Assign MFA device' button.

Step 8: In the Access keys section, click Create access key.

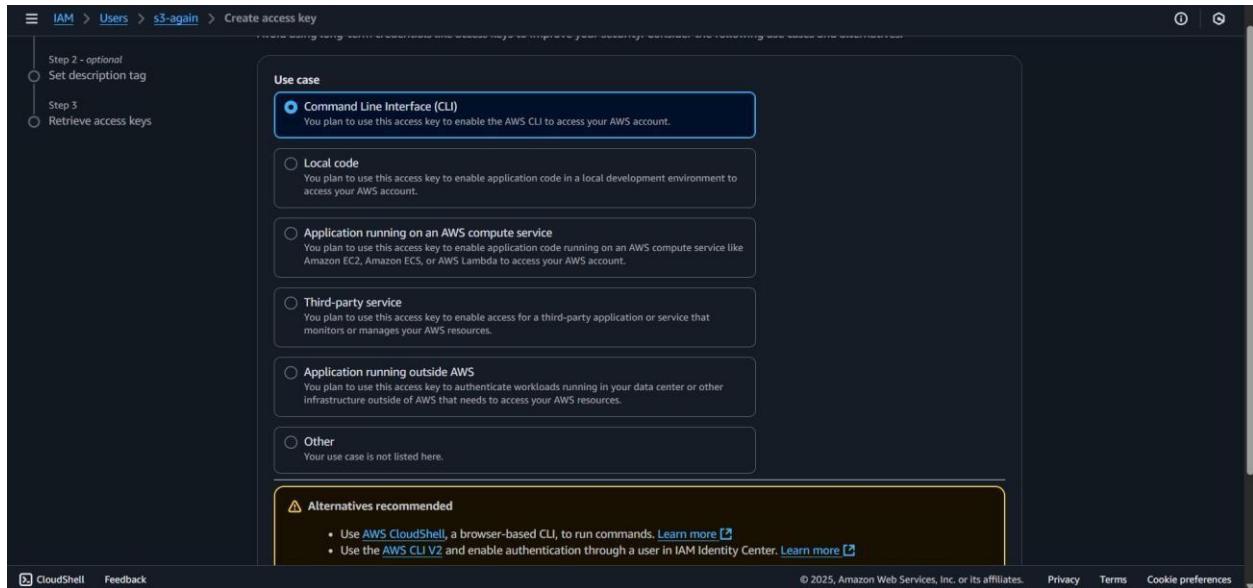
The screenshot shows the 'Create access key' wizard, Step 2 - optional. The top navigation bar is identical to the previous screenshot. The left sidebar shows 'Step 2 - optional' with options: 'Set description tag' (selected) and 'Retrieve access keys'.

The main content area is titled 'Use case' and lists six options:

- Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- Other
Your use case is not listed here.

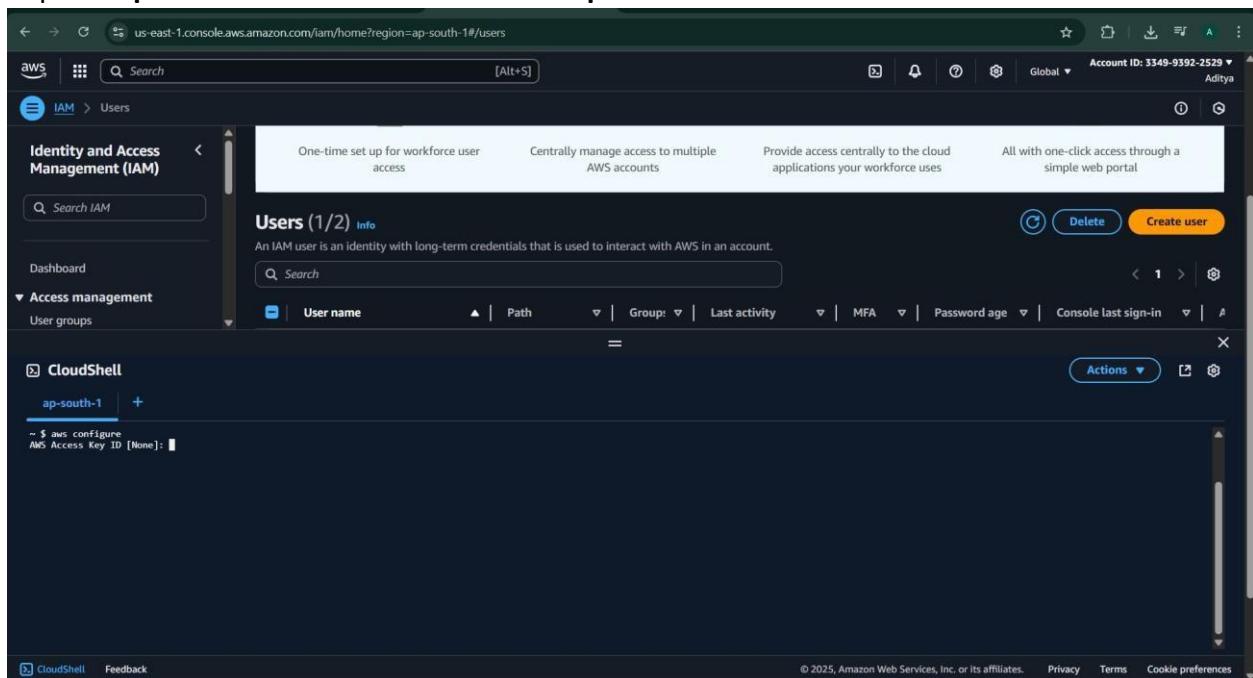
At the bottom right are 'Cancel' and 'Next' buttons.

Step 9: select command line interface



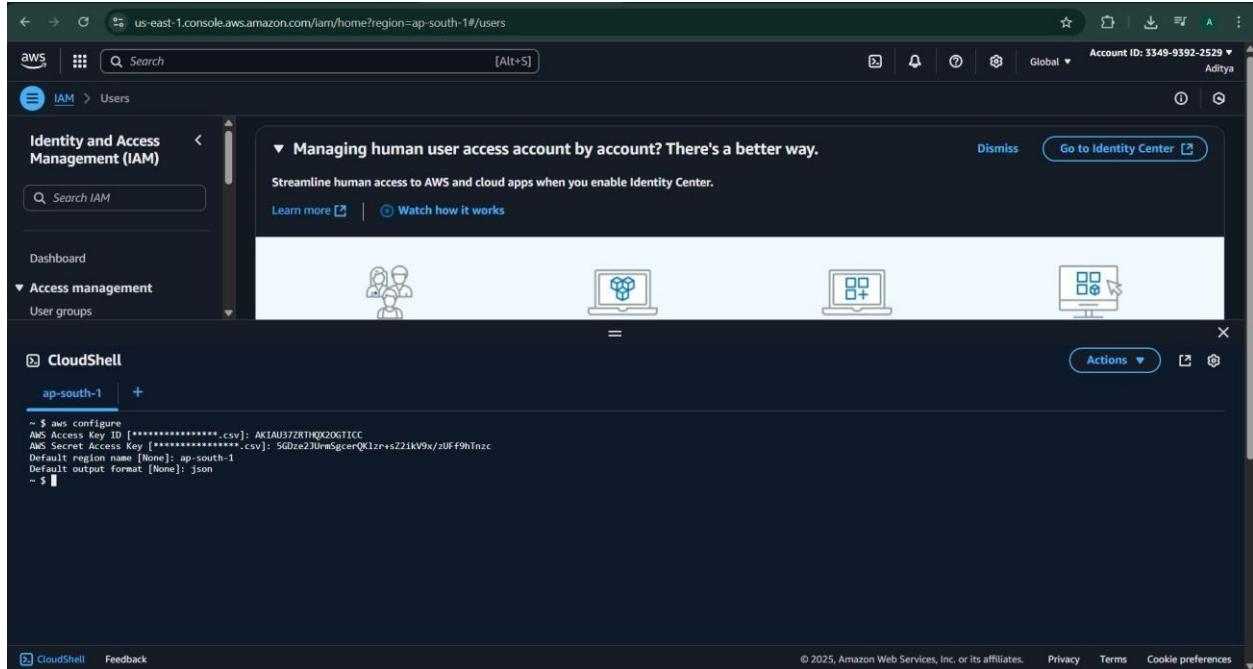
Step 10: Download the .csv file containing the **Access key ID** and **Secret access key**.

Step 11: Open a Terminal or Command Prompt.



Step 12: Run `aws configure`.

Step 13: fill essential details in the cloudshell



```
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#users

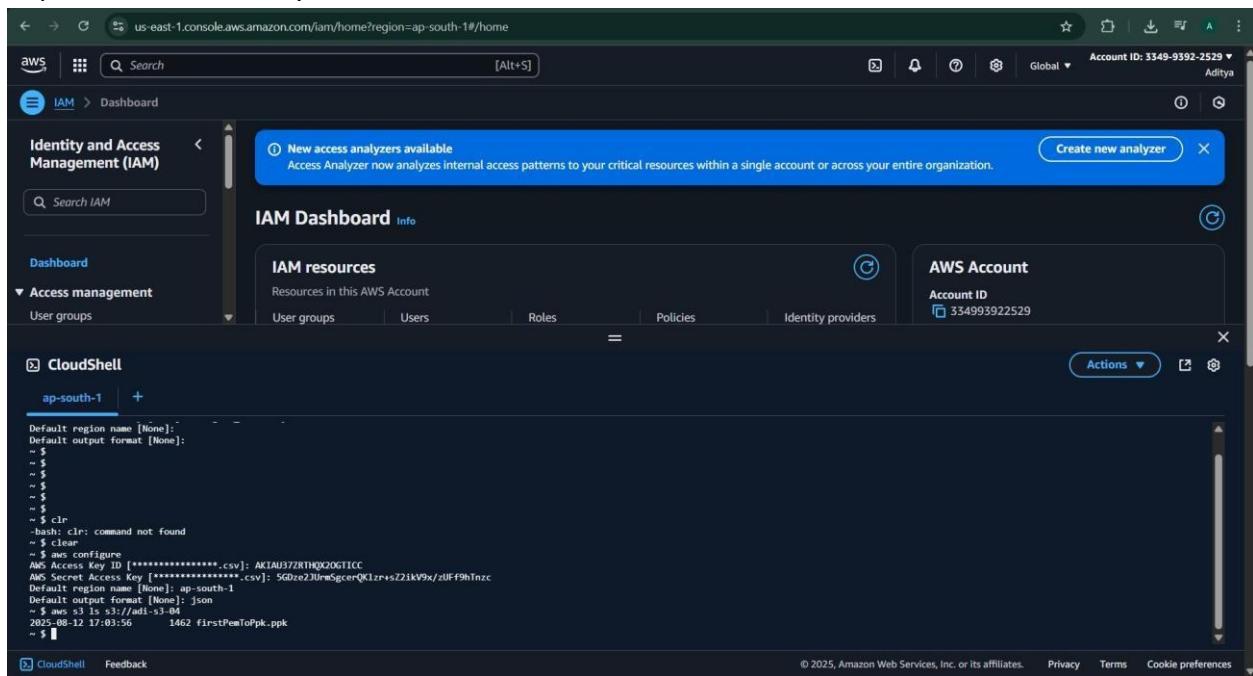
AWS Search [Alt+S] Account ID: 3349-9392-2529 Aditya
IAM Users Dismiss Go to Identity Center

Identity and Access Management (IAM)
Search IAM

Dashboard Access management User groups

CloudShell ap-south-1 +
aws configure
AWS Access Key ID [*****.csv]: AKIAU3ZRTHQX20GTC
AWS Secret Access Key [*****.csv]: SG0ze230wSgcerQK1zr+s221kV9x/zUff9hTnzc
Default region name [None]: ap-south-1
Default output format [None]: json
$ 
```

Step 14: aws s3 ls s3://your-bucket-name



```
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#home

AWS Search [Alt+S] Account ID: 3349-9392-2529 Aditya
IAM Dashboard Create new analyzer

Identity and Access Management (IAM)
Search IAM

Dashboard Access management User groups

CloudShell ap-south-1 +
aws configure
AWS Access Key ID [*****.csv]: AKIAU3ZRTHQX20GTC
AWS Secret Access Key [*****.csv]: SG0ze230wSgcerQK1zr+s221kV9x/zUff9hTnzc
Default region name [None]: ap-south-1
Default output format [None]: json
$ aws s3 ls s3://aditya-s3-bucket
2025-08-12 17:03:56 1462 FirstRemToPpk.ppk
$ 
```

15.Create a bucket policy to allow public read access to all objects in the bucket.

Objective:

To configure a bucket policy that grants public read access to all objects in an S3 bucket, enabling unrestricted object viewing over the internet.

Prerequisite:

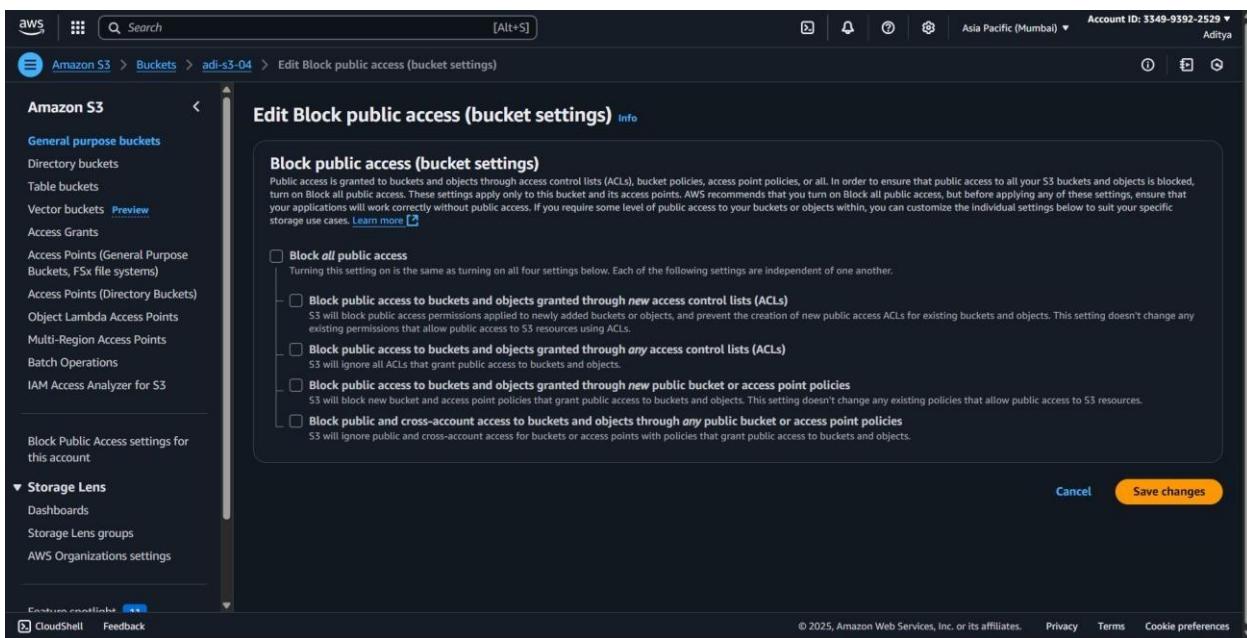
- An active AWS account.
- An existing S3 bucket.
- Permissions to edit bucket policies.
- Basic knowledge of JSON policy syntax.
- AWS Management Console or AWS CLI access.

Step 1: Go to the S3 Dashboard.

Step 2: Open Your Bucket

Step 3: Enable Public Access

- Go to **Permissions** tab.
- Under **Block public access (bucket settings)**, click **Edit**.
- Turn off **Block all public access** (uncheck it) → confirm by typing **confirm** → Save changes.



Step 4: Go to Bucket Policy

In **Permissions** tab, scroll to **Bucket policy** section → click **Edit**.

The screenshot shows the 'Edit bucket policy' page in the AWS S3 console. The left sidebar lists various S3 features like General purpose buckets, Storage Lens, and IAM Access Analyzer. The main area displays a JSON policy document:

```
1 v {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "PublicReadGetObject",  
6       "Effect": "Allow",  
7       "Principal": "*",  
8       "Action": "s3:GetObject",  
9       "Resource": "arn:aws:s3:::adi-s3-04/*"  
10    }  
11  ]  
12 }  
13
```

On the right, there's a sidebar for managing statements, showing one statement named 'PublicReadGetObject'. It includes options to 'Edit', 'Remove', and 'Add actions' (choosing 'S3'). Below that are sections for 'Included' and 'Available' services.

Step 5: save policy

The screenshot shows the 'Edit bucket policy' page after saving. A green success message at the top says 'Successfully edited bucket policy.' The policy document remains the same as in the previous screenshot.

16. Enable server access logging for your S3 bucket. Where are the logs stored?

Objective:

To enable server access logging for an S3 bucket in order to track detailed access requests, and identify the location where these logs are stored.

Prerequisite:

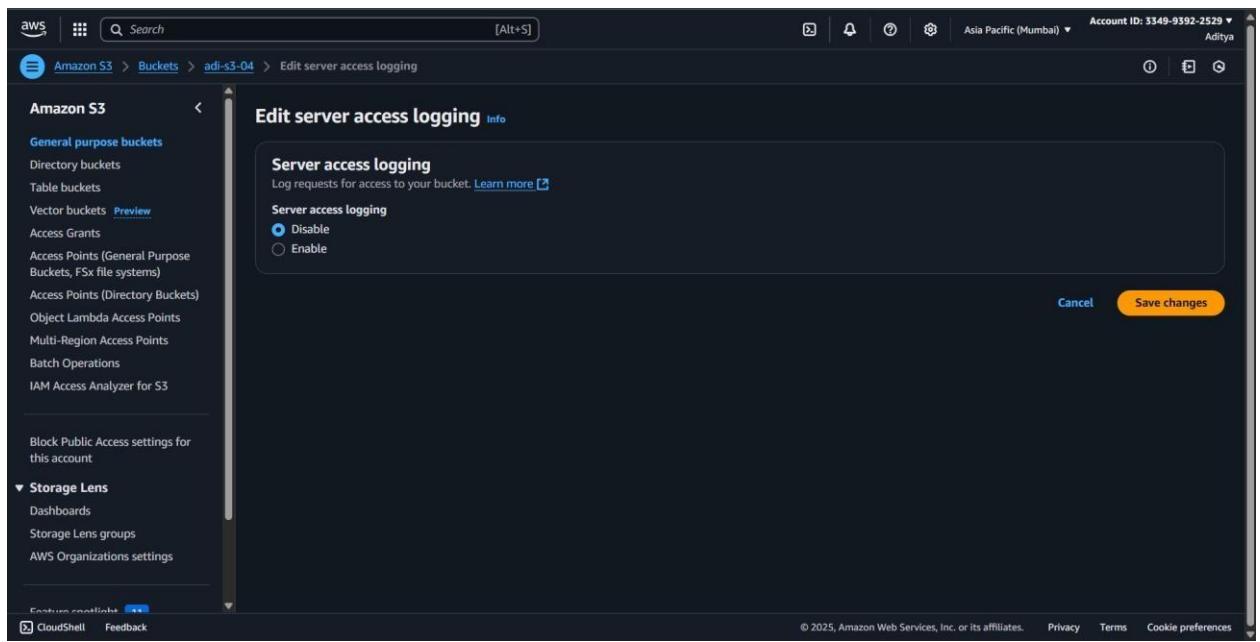
- An active AWS account.
- An existing S3 bucket.
- A target S3 bucket to store access logs.
- Permissions to modify bucket settings and enable logging.
- AWS Management Console or AWS CLI access.

Step 1: Go to s3 dashboard Step

2: Open Your Bucket.

Step 3: Go to Properties

Step 4: Scroll down to **Server access logging** → click **Edit**.



Step 5: Enable Logging.

Buckets (4)

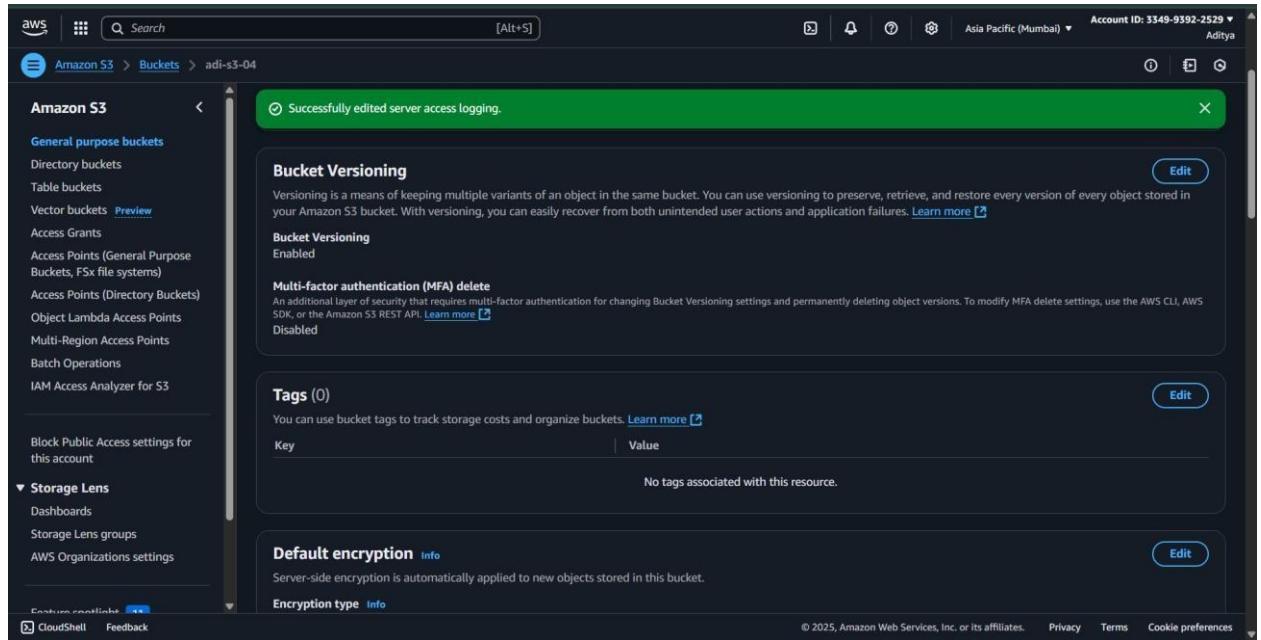
Name	AWS Region	Creation date
adi-s3-04	Asia Pacific (Mumbai) ap-south-1	July 24, 2025, 11:09:01 (UTC+05:30)
bucketforglacier2428	Asia Pacific (Mumbai) ap-south-1	July 29, 2025, 10:46:39 (UTC+05:30)
elasticbeanstalk-ap-south-1-334993922529	Asia Pacific (Mumbai) ap-south-1	July 21, 2025, 23:22:31 (UTC+05:30)
secondawsbucket-04	Asia Pacific (Mumbai) ap-south-1	July 25, 2025, 15:07:13 (UTC+05:30)

Step 6: choose another S3 bucket where logs will be stored.

Buckets (1/4)

Name	AWS Region	Creation date
adi-s3-04	Asia Pacific (Mumbai) ap-south-1	July 24, 2025, 11:09:01 (UTC+05:30)
bucketforglacier2428	Asia Pacific (Mumbai) ap-south-1	July 29, 2025, 10:46:39 (UTC+05:30)
elasticbeanstalk-ap-south-1-334993922529	Asia Pacific (Mumbai) ap-south-1	July 21, 2025, 23:22:31 (UTC+05:30)
secondawsbucket-04	Asia Pacific (Mumbai) ap-south-1	July 25, 2025, 15:07:13 (UTC+05:30)

Step7: save changes:



17. Write a simple handler function that returns: "Welcome to VIT BHOPAL!"

Objective:

To create a basic handler function that outputs the message "*Welcome to VIT BHOPAL!*", demonstrating the fundamentals of function creation and return values.

Prerequisite:

- A programming environment set up (e.g., AWS Lambda, Python, or Node.js runtime).
- Basic understanding of function syntax in the chosen language.
- Access to the development or deployment platform.

Step 1: Go to **AWS Management Console** → search **Lambda**.

Step 2: Click **Create function** → **Author from scratch**.

The screenshot shows the AWS Lambda 'Create function' wizard. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 3349-9392-2529, Aditya). The left sidebar shows the navigation path: Lambda > Functions > Create function. The main content area is titled 'Create function' with an 'Info' link. It provides three options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section contains fields for 'Function name' (myFunctionName), 'Runtime' (Node.js 22.x), and 'Architecture' (x86_64). A 'Permissions' section indicates that Lambda will create an execution role with CloudWatch Logs permissions. On the right side, there's a 'Tutorials' tab (selected) with a 'Create a simple web app' section, a 'Learn more' link, and a 'Start tutorial' button.

Step 3: create function

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Author from scratch' option is selected, with a note to start with a simple Hello World example. Other options include 'Use a blueprint' (Build a Lambda application from sample code and configuration presets for common use cases) and 'Container image' (Select a container image to deploy for your function). The right sidebar features a 'Tutorials' section titled 'Create a simple web app' with a link to learn how to build a simple web app using Lambda.

The screenshot shows the details page for the Lambda function 'welcomeFunction-24'. A success message indicates the function was created successfully. The 'Function overview' tab is selected, showing the function name 'welcomeFunction-24', a placeholder icon for a diagram, and a note that there are no layers. Below this are buttons for '+ Add trigger' and '+ Add destination'. On the right, there are sections for 'Description' (empty), 'Last modified' (12 seconds ago), 'Function ARN' (arn:aws:lambda:ap-south-1:334993922529:function:welcomeFunction-24), and 'Function URL' (empty). The 'Code' tab is selected at the bottom, showing the code source information. The right sidebar continues the 'Create a simple web app' tutorial.

Step 4: change the code

The screenshot shows the AWS Lambda Function Editor interface. At the top, there's a green success message: "Successfully created the function welcomeFunction-24. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the "Code" tab is selected. The code editor displays a file named `lambda_function.py` with the following content:

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('WELCOME TO VIT BHOPAL')
    }
```

On the right side of the editor, there's a sidebar titled "Create a simple web app" which provides a tutorial on how to build a Lambda function that outputs a webpage. The sidebar includes a "Start tutorial" button.

Step 5: click deploy

The screenshot shows the AWS Lambda Function Editor after the code has been updated. A green success message at the top states: "Successfully updated the function welcomeFunction-24." Below the code editor, there's a "DEPLOY" section with two buttons: "Deploy (Ctrl+Shift+U)" and "Test (Ctrl+Shift+I)". The "Deploy" button is highlighted in blue. The sidebar on the right still displays the "Create a simple web app" tutorial.

Step 6: Click Test

aws | Search [Alt+S]

Lambda > Functions > welcomeFunction-24

Successfully updated the function welcomeFunction-24.

lambda_function.py

```
3 def lambda_handler(event, context):
4     return {
5         'statusCode': 200,
6         'body': json.dumps('WELCOME TO')
7     }
8
```

Create new test event

Event Name: function

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings:

- Private: This event is only available in the Lambda Console and to the event creator. You can configure a total of ten. [Learn more](#)
- Shareable: This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional: Hello World

Event JSON:

```
{
    "body": "WELCOME TO VIT BHOPAL"
}
```

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Execution Results

Function Logs:

```
START RequestId: 367c08a0-aee7-4242-bc97-2cc3ae37ccf7 Version: $LATEST
END RequestId: 367c08a0-aee7-4242-bc97-2cc3ae37ccf7
REPORT RequestId: 367c08a0-aee7-4242-bc97-2cc3ae37ccf7 Duration: 2.07 ms Billed Duration: 3 ms
Memory Size: 128 MB Max Memory Used: 35 MB Init Duration: 84.44 ms
```

Lambda Layout: US

ENVIRONMENT VARIABLES

CloudShell Feedback

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

Info Tutorials >

Test your function

You must deploy code changes before you can test them.

An event is an input to your Lambda function. You can create up to 10 test events per function. Saved events are stored in Lambda and are preserved if you switch browsers or machines. Unsaved events are available when switching to [Code](#), but will be lost if the session ends. If your function doesn't require input, the event can be an empty document ().

When you run a test event in the console, Lambda synchronously invokes your function with the test event. The function runtime converts the JSON document into an object and passes it to your code's handler method for processing.

Was this content helpful?

Yes No

18.Launch a Linux EC2 instance into the Public Subnet.

Objective:

To deploy a Linux-based EC2 instance within a public subnet, enabling direct internet access for management and application hosting.

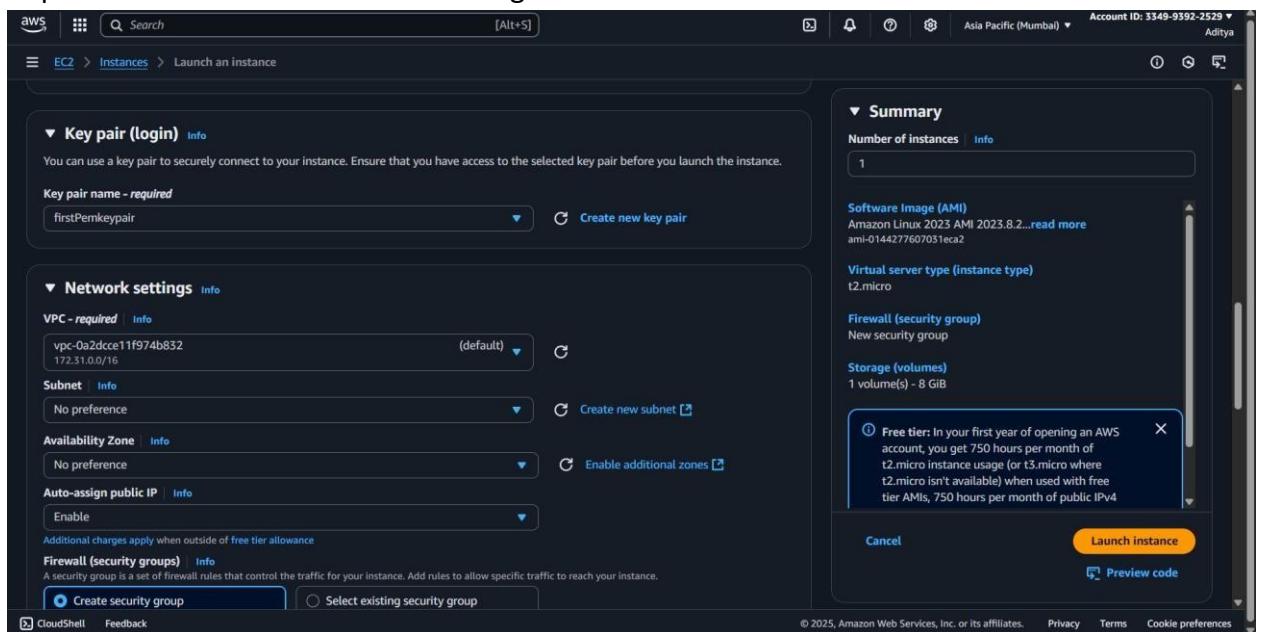
Prerequisite:

- An active AWS account.
- A VPC with a configured public subnet.
- Permissions to launch EC2 instances.
- An existing key pair for SSH access.
- AWS Management Console or AWS CLI access.

Step 1: create an ec2 instance.

Step 2: create with the help of a keypair

Step 3: Click **Edit** in the “Network settings” section.



Step 4: Choose your VPC (the one that contains the Public Subnet).

The screenshot shows the AWS EC2 'Launch an instance' wizard. On the left, under 'Inbound Security Group Rules', a new rule is being configured:

- Type: ssh
- Protocol: TCP
- Port range: 22
- Source type: Anywhere
- Description: SSH for admin desktop

A note at the bottom of the rule table says: "⚠️ 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."

On the right, the 'Summary' section shows:

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2... (ami-0144277607031eca2)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

At the bottom right are 'Cancel', 'Launch Instance', and 'Preview code' buttons.

Step 5: launch the instance

The screenshot shows the 'Success' page after launching an instance (i-0b281e53f9889bc3c). It includes a 'Launch log' link and a 'Next Steps' section with the following options:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy
- Manage detailed monitoring
- Create Load Balancer
- Create AWS budget
- Manage CloudWatch alarms

At the bottom right are 'Cancel', 'Launch another instance', and 'Preview code' buttons.

Step 6: connect to instance and proceed to ssh client option

Screenshot of the AWS EC2 Connect interface showing the SSH client tab selected.

Connect Info

Connect to an instance using the browser-based client.

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

Instance ID
i-0b281e53f9889bc3c (LinuxPublicInstance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is firstPemkeypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
4. Connect to your instance using its Public DNS:
ec2-13-232-151-7.ap-south-1.compute.amazonaws.com

Example:
ssh -i "firstPemkeypair.pem" ec2-user@ec2-13-232-151-7.ap-south-1.compute.amazonaws.com

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

Cancel

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

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

19. Modify the security group of the public EC2 instance to allow only your IP on port 22.

Objective:

To secure SSH access for a public EC2 instance by configuring its security group to allow port 22 connections exclusively from the user's own public IP address.

Prerequisite:

- An active AWS account.
- A running EC2 instance in a public subnet.
- Knowledge of the current public IP address.
- Permissions to edit security group inbound rules.
- AWS Management Console or AWS CLI access.

Step 1: Find the Security Group Select

your EC2 instance.

In the **Description** tab, note the **Security group ID** under **Security**.

The screenshot shows the AWS EC2 Instances details page for a specific instance. The instance ID is i-0b281e53f9889bc3c. The public IP is 13.232.151.7 [Public IP]. The security group is sg-01e7c481e4a055e95 (launch-wizard-2). The instance is running in a VPC with Subnet ID subnet-04032afa9d53504f0 and Instance ARN arn:aws:ec2:ap-south-1:334993922529:instance/i-0b281e53f9889bc3c. The Security tab is selected, showing the Security details section which includes the IAM Role (None), Owner ID (334993922529), and Launch time (Wed Aug 13 2025 18:25:34 GMT+0530 (India Standard Time)). The Networking tab is also visible at the bottom.

Step 2: Edit Inbound Rules

Click on the Security Group ID → Go to the **Inbound rules** tab. Click **Edit inbound rules**.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e598fea658e93222	SSH	TCP	22	Custom	0.0.0.0/0

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.

Cancel Preview changes Save rules

Step 3: Update SSH Rule

- Remove any existing **SSH** rules allowing access from **0.0.0.0/0** (which means open to everyone).
- Add a new rule:
 - Type:** SSH
 - Protocol:** TCP
 - Port Range:** 22
 - Source:** My IP (AWS auto-detects your current public IP).

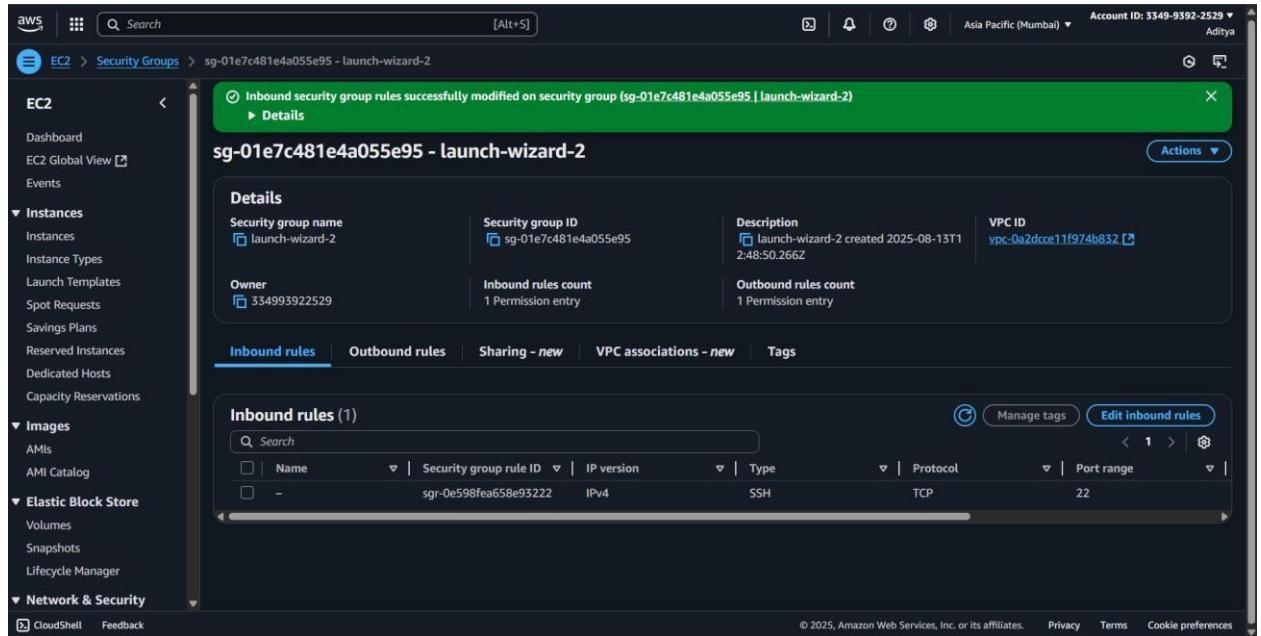
Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e598fea658e93222	SSH	TCP	22	My IP	152.58.56.208/32

Add rule

Cancel Preview changes Save rules

Step 4: save changes



20.Create and attach an Internet Gateway (IGW) to your VPC.

Objective:

To enable internet connectivity for resources within a VPC by creating an Internet Gateway (IGW) and attaching it to the VPC.

Prerequisite:

- An active AWS account.
- An existing VPC.
- Permissions to create and attach Internet Gateways.
- AWS Management Console or AWS CLI access.

Step 1: Go to vpc dashboard.

Step 2: Create an Internet Gateway

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. The left sidebar includes options like EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), Security (Network ACLs, Security groups), PrivateLink and Lattice (Getting started, Endpoints), and CloudShell/Feedback.

The main area displays a table titled 'Internet gateways (2) Info' with the following data:

Name	Internet gateway ID	State	VPC ID	Owner
project-igw	igw-00fa639fa981cd4da	Attached	vpc-002fa8defa3a88021 project-vpc	334993922529
-	igw-0c30a3bb6749a9a84	Attached	vpc-0a2dcce11f974b832	334993922529

A message below the table says 'Select an internet gateway above'.

Step 3: click create internet gateway

The screenshot shows the 'Create internet gateway' settings page. It includes sections for 'Internet gateway settings' (Name tag: MY-IGW) and 'Tags - optional' (Key: Name, Value: MY-IGW). A note at the bottom says 'You can add 49 more tags.' The page has 'Cancel' and 'Create internet gateway' buttons.

Step 4: Attach the IGW to Your VPC □

Select the newly created IGW.

- Click **Actions** → **Attach to VPC**.
- Choose your **VPC** (e.g., My-VPC).
- Click **Attach internet gateway**

Internet gateways (1/4) Info

Name	Internet gateway ID	State	VPC ID	Owner
project-igw	igw-00fa639fa981cd4da	Attached	vpc-002fa8defa3a88021 project-vpc	334993922529
project468-igw	igw-03df7bf0ca2d212c8	Attached	vpc-08c3b178db695a87a project468-... vpc	334993922529
-	igw-0x30a3bb6749a9a84	Attached	vpc-0a2dcce11f974b832	334993922529
MY-IGW	igw-0ea200e7306ba41bc	Detached	-	334993922529

igw-03df7bf0ca2d212c8 / project468-igw

Details **Tags**

Details

Internet gateway ID igw-03df7bf0ca2d212c8	State Attached	VPC ID vpc-08c3b178db695a87a project468-vpc	Owner 334993922529
--	-----------------------------------	--	---------------------------------------

File 2

Index

S. No.	Name of experiment	Experiment date	Submission date	Page no.

1.	Deploy and configure Storage Gateway appliance. Set up File, Volume, and Tape Gateway modes. Integrate on-premises environment with AWS storage. Monitor and manage gateway operations.	15/09/25	24/09/25	4
2.	Design and implement custom VPC architecture. Configure subnets, route tables, and gateways. Implement network security with Security Groups and NACLs. Set up VPC peering and transit gateway.	15/09/25	24/09/25	7
3.	Implement comprehensive IAM strategy. Configure users, groups, roles, and policies. Set up MFA and access controls. Audit and monitor IAM activities.	15/09/25	24/09/25	11
4.	Deploy and configure RDS instances. Implement backup and recovery strategies. Set up read replicas and Multi-AZ deployments. Monitor and optimize database performance.	15/09/25	24/09/25	14
5.	Create and configure DynamoDB tables. Design efficient partition and sort keys. Implement Global Secondary Indexes. Set up DynamoDB Streams for realtime processing.	17/09/25	24/09/25	16

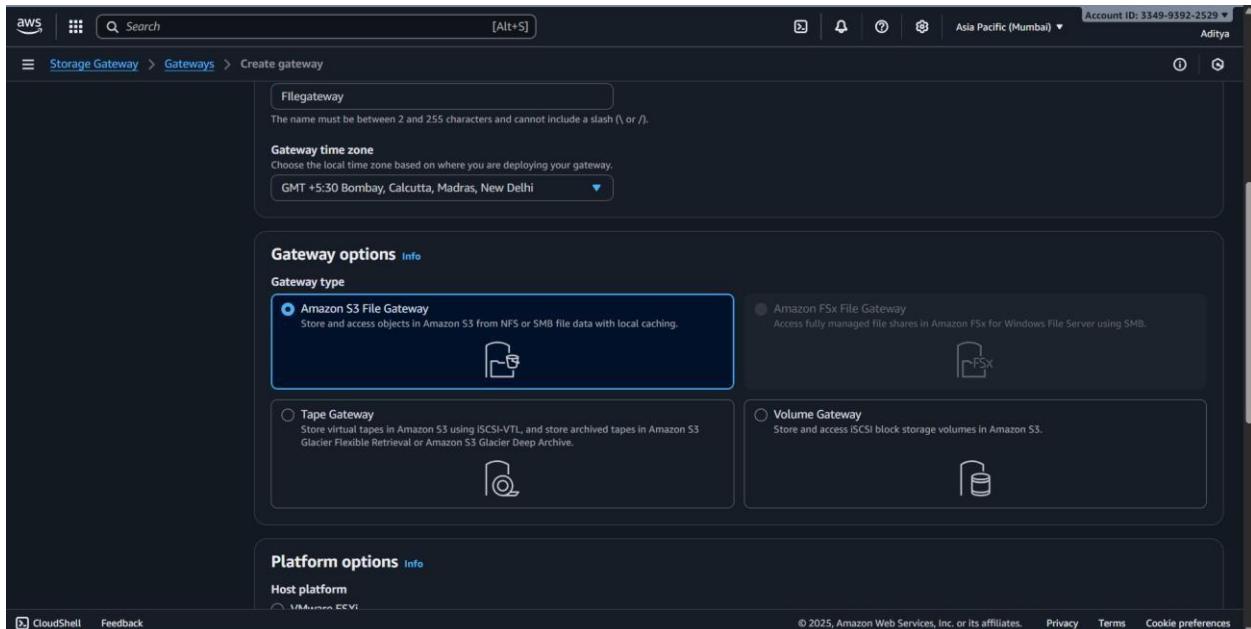
6.	Create CloudFormation templates for infrastructure. Implement nested stacks and cross-stack references. Manage stack updates and rollbacks. Integrate with CI/CD pipelines.	17/09/25	24/09/25	19
7.	Deploy applications using Elastic Beanstalk. Configure application environments. Implement blue-green deployments. Monitor application health and performance.	17/09/25	24/09/25	21
8.	Set up comprehensive logging and monitoring. Create custom metrics and alarms. Implement log analysis and alertingBuild operational dashboards.	19/09/25	24/09/25	22
9.	Design and implement complex workflows. Handle error conditions and retries. Integrate multiple AWS services. Monitor workflow execution.	20/09/25	24/09/25	24
10.	Set up comprehensive cost monitoring. Create budgets and alerts. Analyze spending patterns. Implement cost optimization strategies	20/09/25	24/09/25	26

1. Deploy and configure Storage Gateway appliance. Set up File, Volume, and Tape Gateway modes. Integrate on-premises environment with AWS storage. Monitor and manage gateway operations.

Steps:

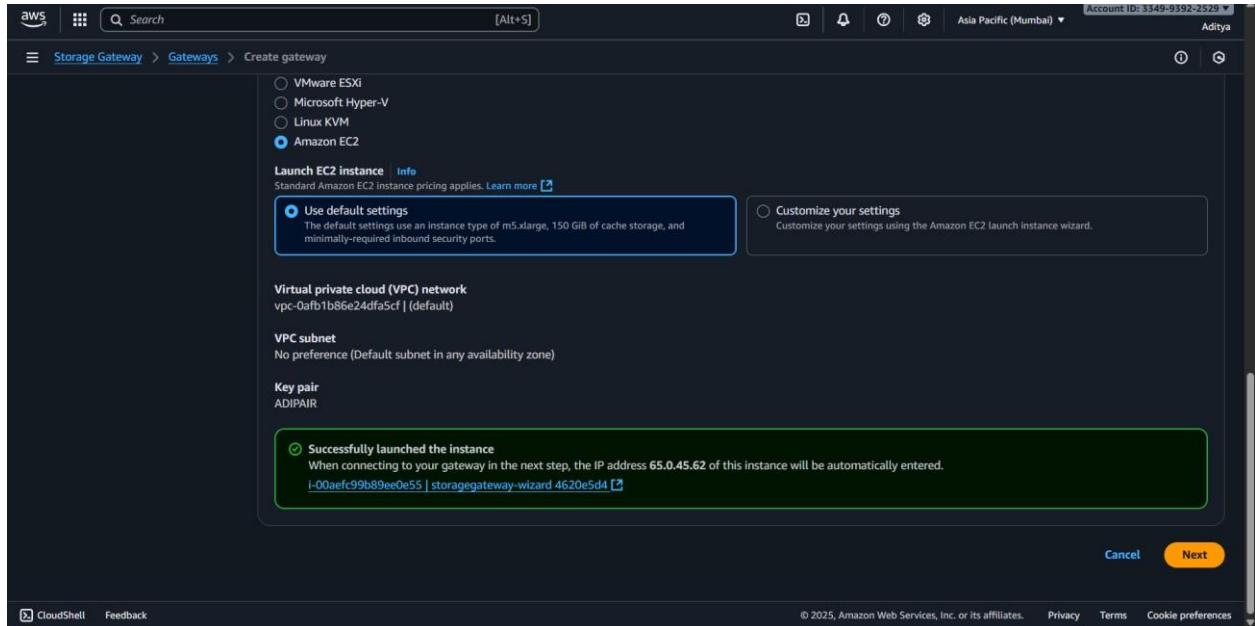
1. Deploy AWS Storage Gateway Appliance

1. Login to AWS Console → search **Storage Gateway**.
2. Click **Create gateway**.
3. Select **Gateway type** → choose one (File / Volume / Tape).
4. Choose **Host platform**:
 - **VMware ESXi / Hyper-V / KVM** → if you have virtualization.
 - **Amazon EC2** → simplest for lab environment.

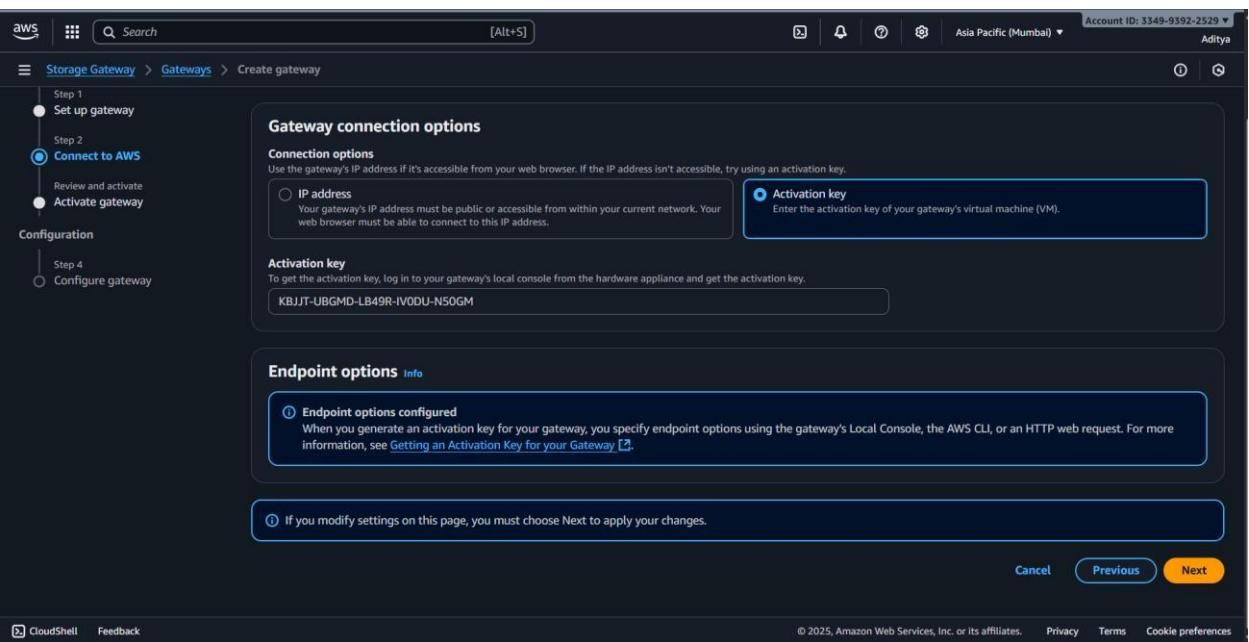


5. Deploy the gateway:

- If using **EC2** → select an **instance type (m5.xlarge recommended)**. ○ Launch with **default settings**.



Once deployed → copy paste the activation key .



2. Activate the Gateway

1. Go back to **AWS Console → Storage Gateway**.
2. Paste the **activation key**.
3. Give a **name** (example: *MyFileGateway*).
4. Choose **time zone and AWS region**.
5. Click **Activate gateway**.

Storage Gateway > Gateways > Create gateway

Step 2: Connection details

Gateway options

- Gateway name: Filegateway
- Gateway type: Amazon S3 File Gateway
- AWS region: Asia Pacific (Mumbai)
- Gateway time zone: GMT +5:30 Bombay, Calcutta, Madras, New Delhi
- Host platform type: Amazon EC2

Connection settings

- Gateway connection option: Activation key
- Activation key: KBUJT-UBGMD-LB49R-IV0DU-N50GM

Review your gateway and connection details

After your gateway is created, you can't modify the gateway settings or connection settings.

Buttons: Cancel, Previous, Activate gateway

Storage Gateway > Gateways > Create gateway

Successfully activated gateway Filegateway

You must configure cache storage before you can use the gateway. You can also configure other settings on this page.

Activation

- Set up gateway
- Connect to AWS
- Review and activate
- Activate gateway**

Configuration

- Configure gateway

Configure gateway

Configure cache storage

The following disks were detected on your gateway's host platform. Allocate one or more disks with a total of at least 150 GiB to the Cache.

Disk ID	Capacity	Allocated to
/dev/sdb	150 GiB	Cache

CloudWatch log group

You can monitor the health of your gateway using Amazon CloudWatch log groups.

Choose how to set up log group

You can activate or deactivate logging at any time.

- Create a new log group** A new CloudWatch log group will be created.
- Use an existing log group** Choose an existing CloudWatch log group.
- Deactivate logging** No CloudWatch log group will be created.

Links: CloudShell, Feedback, Privacy, Terms, Cookie preferences

3. Configure Gateway Modes

(A) File Gateway

- Provides **NFS (Linux/Unix)** or **SMB (Windows)** file shares.

- In the Storage Gateway console → choose **File Gateway**.
- Create a **File Share**:
 - Backend storage = **Amazon S3 bucket**.
 - Protocol = **NFS / SMB**.

3. Attach this file share to your **on-premises client**.
4. Now, files written to the share automatically sync to **S3**.

The screenshot shows the AWS Storage Gateway interface for creating a new file share. In the 'Basics' section, the 'Gateway' dropdown is set to 'Filegateway (sgw-25EB644C)'. Under 'File share protocol', 'SMB' is selected, indicated by a blue outline around the radio button. The 'S3 bucket' dropdown is set to 'Choose an option'. A 'Default configuration' info box provides details about the current settings, including the storage class, encryption, and VPC connection. At the bottom right, there are 'Cancel', 'Customize configuration', and 'Create file share' buttons.

(B) Volume Gateway

- Provides **iSCSI block storage** (two modes: Cached / Stored volumes).
1. Choose **Volume Gateway** during setup.
 2. Configure **iSCSI connection** to on-premises servers.
 3. Select **Cached Volume** → data stored in S3, frequently accessed in cache.
 4. Select **Stored Volume** → data stored locally, backup copies to S3.
 5. Attach volume to on-premises server → use like a local disk.

(C) Tape Gateway

- Used for backup/archival with **Virtual Tape Library (VTL)**.
1. Choose **Tape Gateway** in setup.
 2. Create **Virtual Tapes** (each tape stored in **Amazon S3**).
 3. Integrate with **backup software** (like Veeam, Commvault, NetBackup).
 4. Tapes can be archived to **Amazon S3 Glacier** for cost efficiency.

4. Integrate On-Premises with AWS

1. Install Storage Gateway appliance **on-premises (VMware/Hyper-V)**.
2. Connect the appliance to AWS Storage Gateway service via **Internet or Direct Connect**.
3. Mount File Shares (NFS/SMB) or connect Volumes (iSCSI) or Tapes (VTL) on local servers.
4. Verify data transfer → data moves securely to **AWS S3, EBS, or Glacier**.

5. Monitor and Manage Gateway

1. Go to **AWS Console** → **CloudWatch**.
2. Check metrics like: **Read/Write throughput** **Cache hit ratio** **Disk utilization**
3. Use **CloudWatch Alarms** to alert on performance/storage issues.
4. In **Storage Gateway Console**, you can:
 - Expand volumes or file shares. Add/remove tapes.
 - Stop/Start the gateway.

2. Design and implement custom VPC architecture. Configure subnets, route tables, and gateways. Implement network security with Security Groups and NACLs. Set up VPC peering and transit gateway.

Steps-

Steps to Design and Implement Custom VPC Architecture

2. Log in to the AWS Management Console.
3. Navigate to VPC service.

4. Click Create VPC.
5. Select VPC only.
6. Enter:
 - Name: CustomVPC
 - IPv4 CIDR block: 10.0.0.0/16 ○ Leave other defaults and click Create VPC.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
custom-vpc

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block
10.0.0.0/16

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy [Info](#)
Default

VPC dashboard [Actions](#)

vpc-0182803cb7e94349e / custom-vpc

Details [Info](#)

VPC ID vpc-0182803cb7e94349e	State Available	Block Public Access <input type="radio"/> Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0f501955f598ee4b0	Main route table rtb-0cc294ad4c58677b5
Main network ACL acl-06316b4b72c2ceec7	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 334993922529

Resource map [Info](#)

VPC Your AWS virtual network custom-vpc	Subnets (0) Subnets within this VPC	Route tables (1) Route network traffic to resources rtb-0cc294ad4c58677b5	Network Connections (0) Connections to other networks
---	--	---	--

Steps to Configure Subnets

1. In the VPC dashboard, go to Subnets → Create subnet.
2. Select CustomVPC.

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.
PublicSubnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Mumbai) / aps1-az1 (ap-south-1)

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

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

Subnets (2) Info

You have successfully created 2 subnets: subnet-00819643819e006d3, subnet-069c60146f4b970c1

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
PrivateSubnet	subnet-069c60146f4b970c1	Available	vpc-0182803cb7e94349e cust...	Off	10.0.2.0/24
PublicSubnet	subnet-00819643819e006d3	Available	vpc-0182803cb7e94349e cust...	Off	10.0.1.0/24

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

3. Create two subnets:

- PublicSubnet: 10.0.1.0/24 (Availability Zone 1).
- PrivateSubnet: 10.0.2.0/24 (Availability Zone 2).

4. Click Create subnet

Steps to Configure Route Tables

1. In the **VPC dashboard**, go to **Route tables** → **Create route table**.

- Name: PublicRouteTable → Associate with **CustomVPC**.

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

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

PublicRouteTable

VPC
The VPC to use for this route table.
vpc-0182803cb7e94349e (custom-vpc)

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 PublicRouteTable X Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

2. Add a route:

- Destination: 0.0.0.0/0 ○

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

igw-077125afafaaa5b01

Add route

Cancel Preview Save changes

Target: Internet Gateway

(IGW).

3. Associate this route table with **PublicSubnet**.

The screenshot shows the 'Edit subnet associations' page in the AWS VPC console. At the top, there's a search bar and account information. Below it, the breadcrumb navigation shows 'VPC > Route tables > rtb-020965c52b4ced1e6 > Edit subnet associations'. The main section is titled 'Edit subnet associations' with the sub-instruction 'Change which subnets are associated with this route table.' Below this, there are two sections: 'Available subnets (1/2)' and 'Selected subnets'.

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
PrivateSubnet	subnet-069c60146f4b970c1	10.0.2.0/24	-	Main (rtb-0cc294ad4c58677b5)
<input checked="" type="checkbox"/> PublicSubnet	subnet-00819643819e006d3	10.0.1.0/24	-	Main (rtb-0cc294ad4c58677b5)

Selected subnets

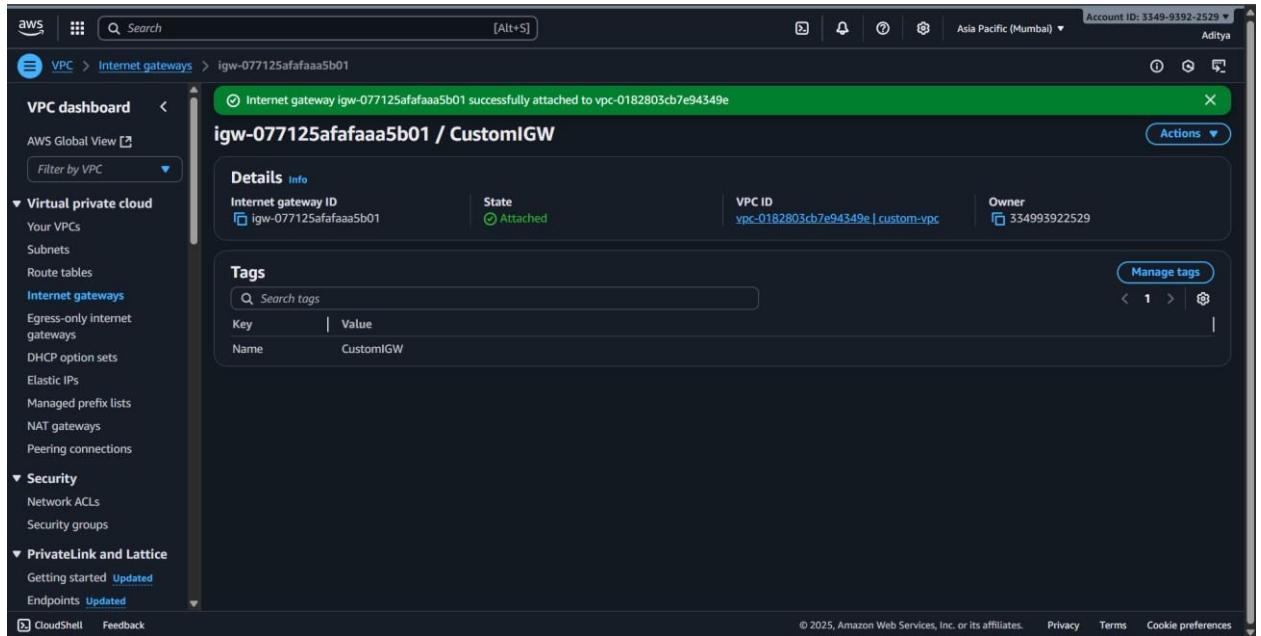
subnet-00819643819e006d3 / PublicSubnet

Buttons at the bottom right: 'Cancel' and 'Save associations'.

4. For **PrivateSubnet**, keep the **Main Route Table** (no direct Internet access).

5. Steps to Configure Gateways

6. Go to **Internet Gateways → Create internet gateway**.
7. Name: CustomIGW.
8. Attach the IGW to **CustomVPC**.



9. (Optional for private subnet) Create **NAT Gateway**:

10. Place it inside **PublicSubnet**.

11. Associate with **Elastic IP**.

12. Add route in **PrivateSubnet's Route Table** to point 0.0.0.0/0 → NAT Gateway.

Steps to Implement Network Security

1. Security Groups:

- Create a **WebSG** for public instances: Allow inbound HTTP (80), HTTPS (443), and SSH (22).
- Create a **DBSG** for database instances: Allow inbound MySQL (3306) only from **WebSG**.

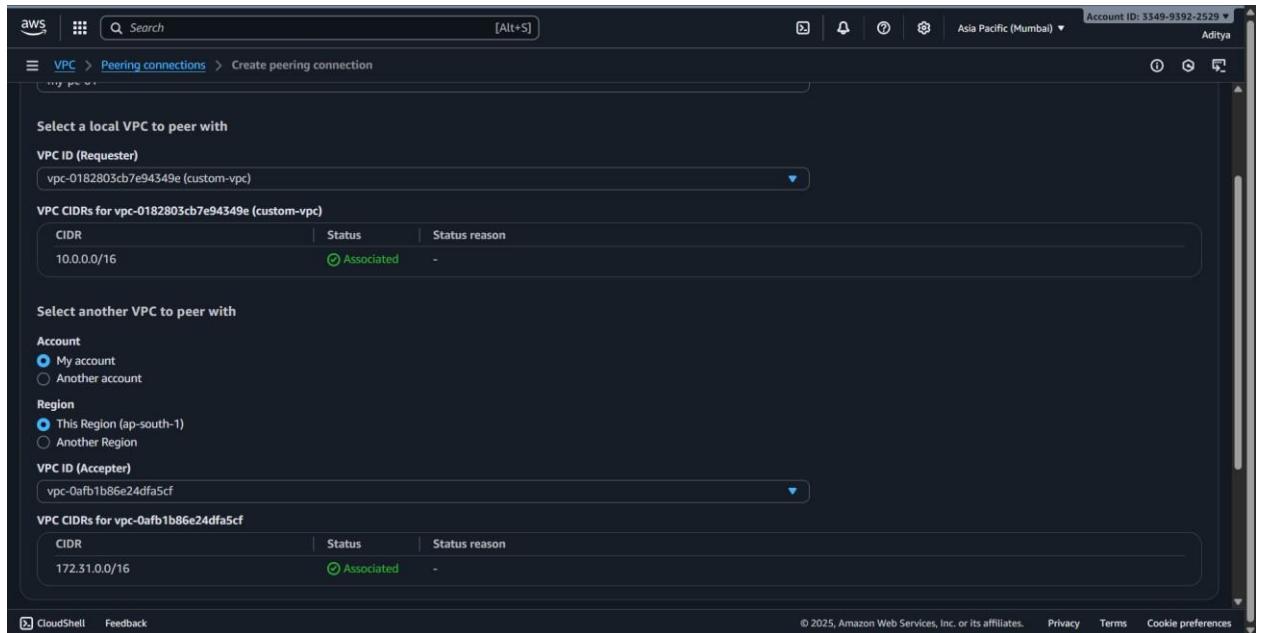
2. Network ACLs (NACLs):

- For **PublicSubnet** NACL: Allow inbound 80, 443, 22 and outbound all.
- For **PrivateSubnet** NACL: Allow inbound 3306 and outbound traffic restricted to application/web tier.

Steps to Set Up VPC Peering

1. In the **VPC dashboard**, select **Peering Connections** → **Create peering connection**.

- Select **CustomVPC** as requester, and another VPC (e.g., OtherVPC) as accepter.



- Click **Create peering connection**.
- Update **route tables** in both VPCs to enable communication.

Steps to Set Up Transit Gateway

- Navigate to **Transit Gateways** → **Create transit gateway**.
 - Name: CustomTransitGW.
 - Keep default ASN or assign custom.
- Attach **CustomVPC** to the Transit Gateway.
- Attach other VPCs or VPN connections as needed.
- Update route tables in each VPC to forward traffic via **Transit Gateway**.

3. Implement comprehensive IAM strategy. Configure users, groups, roles, and policies. Set up MFA and access controls. Audit and monitor IAM activities.

Create Users

3. Go to Users → Add users.
4. Enter username (e.g., StudentUser1).
5. Select Provide user access:
 - Check AWS Management Console access (with password).
 - Optionally check Programmatic access (for CLI/SDK).
6. Click Next.

Specify user details

User details

User name: StudentUser1

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

If you're providing console access to a person, it's a best practice [link] to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type:

- Specify a user in Identity Center - Recommended
- We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
- I want to create an IAM user
- We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more link]

User created successfully

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

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: <http://334993922529.siginin.aws.amazon.com/console>

User name: StudentUser1

Console password: ***** [Show](#)

[Email sign-in instructions](#)

[Cancel](#) [Download .csv file](#) [Return to users list](#)

Create Groups

7. Go to **User groups** → **Create group**.
8. Enter group name: DevelopersGroup.
9. Attach a managed policy like AmazonEC2ReadOnlyAccess.
10. Add the user (StudentUser1) to this group.

The screenshot shows the AWS IAM 'Create user group' interface. In the 'Name the group' section, 'DevelopersGroup' is entered. In the 'Add users to the group - Optional (1/3)' section, 'StudentUser1' is selected. In the 'Attach permissions policies - Optional (1/1077)' section, 'amazonec2re' is searched.

Create Roles

11. Go to **Roles** → **Create role**.
12. Select trusted entity:
 - For AWS service, choose **EC2** (so EC2 can assume this role).
13. Attach policy (e.g., **AmazonS3FullAccess**).
14. Name the role: **EC2S3AccessRole**.
15. Launch an EC2 instance and attach this role to allow S3 access.

EC2S3AccessOnlyAditya Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
September 20, 2025, 23:26 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::334993922529:role/EC2S3AccessOnlyAditya

Maximum session duration
1 hour

Permissions **Trust relationships** **Tags** **Last Accessed** **Revoke sessions**

Permissions policies Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
Loading policies		

Permissions boundary (not set)

Create Custom Policies

16. Go to **Policies** → **Create policy**.
17. Choose **JSON editor** and define a custom policy, e.g., to allow only s3>ListBucket on a specific bucket.
18. Save policy as **CustomS3ListPolicy**.
19. Attach this policy to the desired user or group.

Policies (1/1396) Info

A policy is an object in AWS that defines permissions.

Policy name	Type	Used as	Description
Adi_policy	Customer managed	Permissions policy (1)	-
AWSLambdaBasicExecutionR...	Customer managed	Permissions policy (1)	-
AWSLambdaBasicExecutionR...	Customer managed	Permissions policy (1)	-
AWSLambdaBasicExecutionR...	Customer managed	Permissions policy (1)	-
customS3ListPolicy	Customer managed	None	-
listing5bucketsAllowance	Customer managed	None	-
XRayAccessPolicy-a0f37630-f...	Customer managed	Permissions policy (1)	Allow AWS Step Functions to call X-Ra...

Steps to Set Up MFA and Access Controls

20. Go to **Users** → select **StudentUser1** → **Security credentials**.

The screenshot shows the AWS IAM Security credentials page for the user 'StudentUser1'. The top navigation bar includes the AWS logo, a search bar, and account information ('Account ID: 3349-9392-2529' and 'Aditya'). The left sidebar contains links for Identity and Access Management (IAM), Access management, Access reports, and CloudShell/Feedback. The main content area has tabs for Permissions, Groups (1), Tags, Security credentials (which is selected), and Last Accessed. Under 'Console sign-in', there is a link to the AWS console sign-in page. Under 'Multi-factor authentication (MFA) (0)', it says 'Use MFA to increase the security of your AWS environment.' Under 'Access keys (0)', it says 'No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials.' At the bottom right, there are links for 'Assign MFA device', 'Create access key', 'Privacy', 'Terms', and 'Cookie preferences'.

21. Under **Multi-factor authentication (MFA)**, choose **Assign MFA device**.

22. Select **Authenticator app** → scan QR code with Google Authenticator/Authy.

23. Enter the 2 codes generated → Activate MFA.

24. Now login requires username, password, and MFA code.

The screenshot shows the AWS IAM Security credentials page for the user 'StudentUser1' after an MFA device has been assigned. A green notification bar at the top states: 'Passkey MFA device assigned. As a security best practice, we encourage registering multiple devices in the case that your primary method is lost, disabled, or unavailable. Choose any of your MFA devices to use to sign in to your AWS account.' Below this, the 'Summary' section shows the ARN (arn:awsiam:33493922529:user/StudentUser1) and creation date (September 20, 2025, 23:19 (UTC+05:30)). The 'Security credentials' tab is selected, showing 'Console sign-in' and 'Multi-factor authentication (MFA) (0)'. The MFA section includes a note: 'Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned.' At the bottom right, there are links for 'Assign MFA device', 'Create access key', 'Privacy', 'Terms', and 'Cookie preferences'.

Steps to Audit and Monitor IAM Activities

25. Go to **IAM** → **Access analyzer** to review policies granting public or cross-account access.

26. Enable CloudTrail service:

- This records all IAM API activities (user logins, policy changes, role assumptions).

27. Enable AWS Config to track configuration changes in IAM (e.g., user/group creation).

28. Review logs in CloudWatch Logs for monitoring unusual login attempts or denied actions.

The screenshot shows the AWS IAM Access Analyzer console. The left sidebar navigation includes: Identity and Access Management (IAM) (Dashboard, Access management (User Groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), Access reports (Access Analyzer (Resource analysis New), Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies). The main content area is titled "IAM Access Analyzer" and "Simplify your journey to least privilege". It features a "How it works" section with three steps: 1. Create an analyzer (Icon: hexagon with nodes), 2. Review findings (Icon: hexagon with nodes and arrows), 3. Take action (Icon: hexagon with nodes and a minus sign). To the right, there's a "Get started" section with "External access findings" (Identify resources shared with an external entity), "Internal access findings" (Identify internal IAM users and roles that can access specific resources), and "Unused access findings" (Identify IAM users and roles with unused access). A "Create analyzer" button is present. Below that is a "Pricing (USD)" section showing rates for External access (Free), Internal access (\$9 / AWS resource / month), and Unused access (\$0.2 / IAM user and role / month). An "AWS Pricing Calculator" link is also provided. The bottom of the page includes standard AWS footer links: CloudShell, Feedback, © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

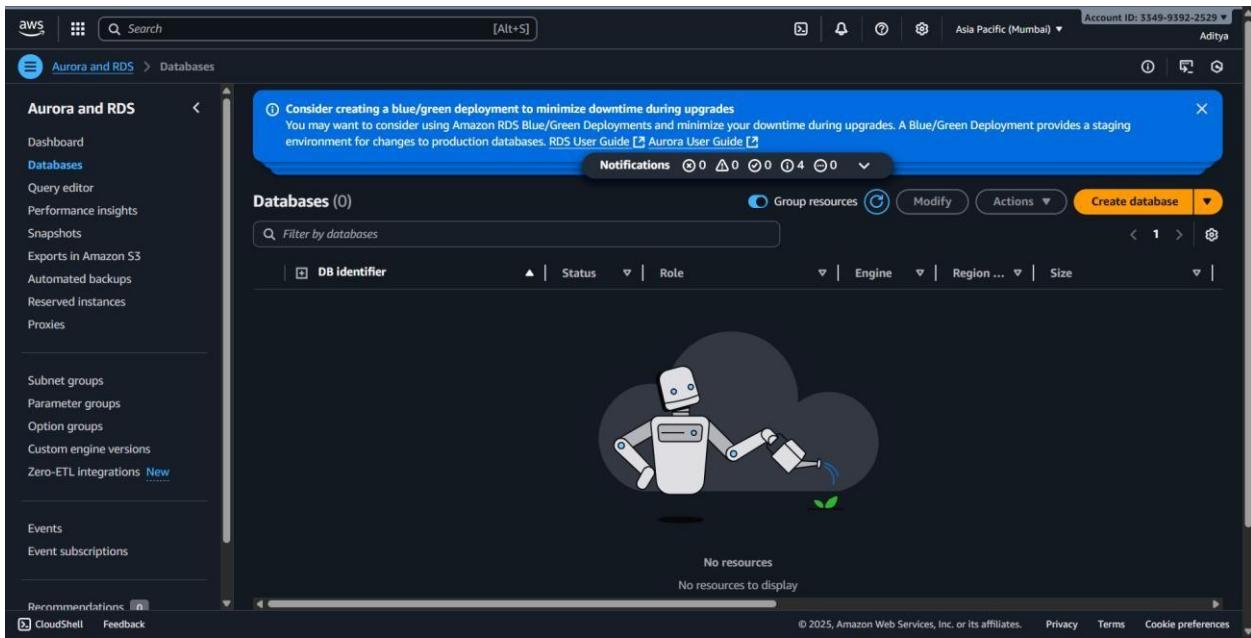
4. Deploy and configure RDS instances. Implement backup and recovery strategies.

Set up read replicas and Multi-AZ deployments. Monitor and optimize database

Performance

Steps:

1. Go to RDS service.
2. Click **Create database**.



3. Choose **Standard create**.
5. Select **Engine**: MySQL or PostgreSQL (free-tier eligible).

The screenshot shows the 'Engine options' section of the AWS RDS 'Create database' wizard. It lists several database engines with their respective icons:

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL** (selected)
- PostgreSQL
- MariaDB
- Oracle
- Microsoft SQL Server
- IBM Db2

Below the engines, there are sections for 'Edition' (MySQL Community selected) and 'Engine version' (Info). At the bottom, there are links for 'CloudShell' and 'Feedback'.

6. Choose **Templates**: Select **Free tier** (to reduce cost).

7. Set details:

- DB identifier: StudentDB.
- Username: admin.
- Password: set a secure one.

The screenshot shows the 'Master username' section of the AWS RDS 'Create database' wizard. It includes fields for the master user name ('admin') and password strength ('Very strong'). The 'Self managed' password option is selected.

Below this, there are sections for 'Master password' and 'Confirm master password'. The 'Instance configuration' section at the bottom includes options for 'DB instance class' (db.t3.micro selected), 'Show instance classes that support Amazon RDS Optimized Writes' (unchecked), and 'Hide filters'.

8. Instance size: choose **db.t3.micro** (free-tier).

9. Storage: Select **20 GB General Purpose SSD** (free-tier).

10. VPC: Use your default or custom VPC.

11. Connectivity: Enable public access (for lab only, not production).

12. Click **Create database**.

The screenshot shows the AWS Aurora and RDS Databases interface. On the left, there's a sidebar with various options like Dashboard, Databases, Query editor, etc. The main area is titled 'Creating database studentdb' with a message: 'Your database might take a few minutes to launch. You can use settings from studentdb to simplify configuration of suggested database add-ons while we finish creating your DB for you.' Below this, there's a table titled 'Databases (1)' with one entry: 'studentdb' (Status: Creating, Instance: MySQL Community, Engine: db.t3.micro). There are buttons for 'Group resources', 'Modify', 'Actions', and 'Create database'.

Steps to Implement Backup and Recovery Strategies

1. Go to your RDS instance → Maintenance & backups tab.

This screenshot shows the 'Maintenance & backups' tab for the 'studentdb' database. The 'Summary' section includes details like DB identifier (studentdb), Status (Creating), Role (Instance), Engine (MySQL Community), and Region & AZ (ap-south-1b). The 'Maintenance' section shows 'Auto minor version upgrade' is 'Enabled'. The 'Pending maintenance' section indicates 'none'. At the bottom, there are buttons for 'Apply now' and 'Apply at next maintenance window'.

2. Enable Automated backups.

Backup

- Automated backups Enabled (1 Day)
- Latest restore time September 22, 2025, 20:57 (UTC+05:30)
- Backup window 18:14-18:44 UTC (GMT)
- Replicate to Region -
- Replicated automated backup -

Schemas (1)

Name	Type	Size (MB)	Last modified
studentdb	MySQL 8.0	100	September 22, 2025, 20:57 (UTC+05:30)

Events

No recent events.

3. Take a Manual Snapshot:

- Actions → Take snapshot.

4. For Recovery:

- Go to Snapshots → Select snapshot → **Restore snapshot** → Launch new RDS instance from snapshot.

Aurora and RDS > Databases

Snapshot is being restored
Restoring rds:studentdb-2025-09-22-15-26 to RestoredDB. This might take several minutes.

Databases (2)

DB identifier	Status	Role	Engine	Region ...	Size
restoreddb	Creating	Instance	MySQL Co...	-	db.m7g.large
studentdb	Available	Instance	MySQL Co...	ap-south-1b	db.t3.micro

Notifications 0 0 0 4 1

Create database

CloudShell Feedback

Steps to Set Up Read Replicas

1. Go to your RDS instance.
2. Click Actions → Create read replica.

Aurora and RDS > Databases

Snapshot is being restored
Restoring rds:studentdb-2025-09-22-15-26 to RestoredDB. This might take several minutes.

Databases (2)

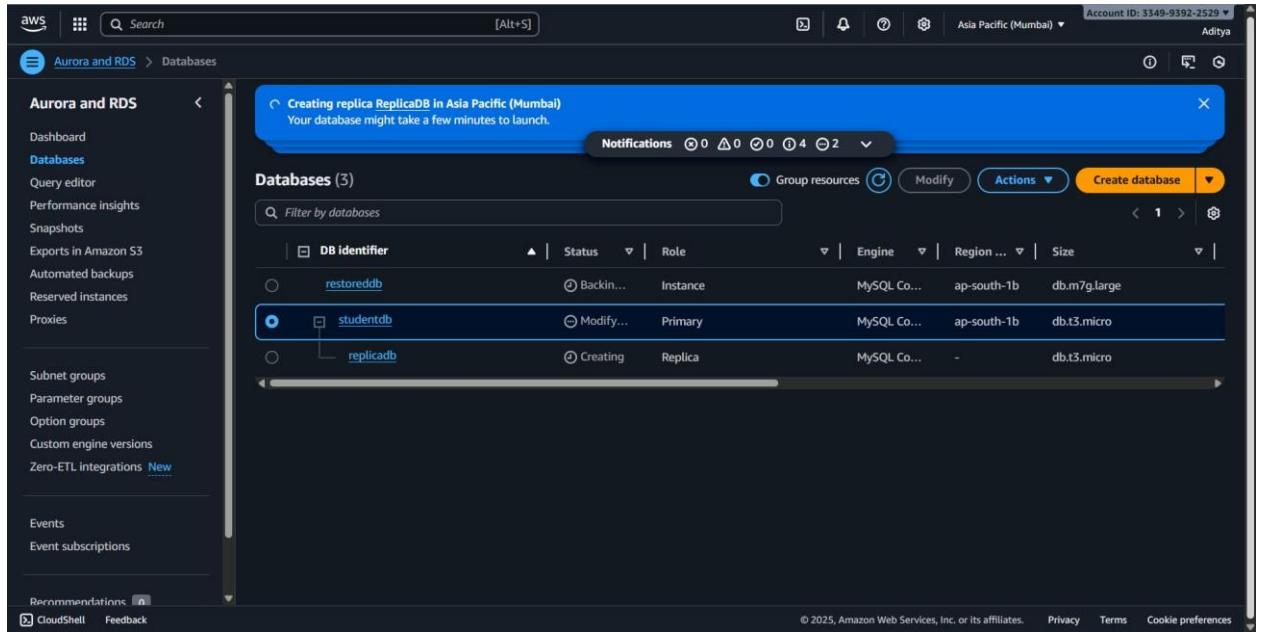
DB identifier	Status	Role
restoreddb	Creating	Instance
studentdb	Available	Instance

Actions ▾

- Stop temporarily
- Reboot
- Delete
- Set up EC2 connection
- Set up Lambda connection
- Migrate data from EC2 database - new
- Create read replica**
- Create Aurora read replica
- Create blue/green deployment
- Promote
- Convert to Multi-AZ deployment
- Take snapshot
- Restore to point in time
- Migrate snapshot
- Create zero-ETL integration
- Create RDS Proxy
- Create ElastiCache cluster

CloudShell Feedback

3. Choose instance size → db.t3.micro (to save cost).
4. Keep in same region (lower cost) unless cross-region is required.
5. Create the replica.

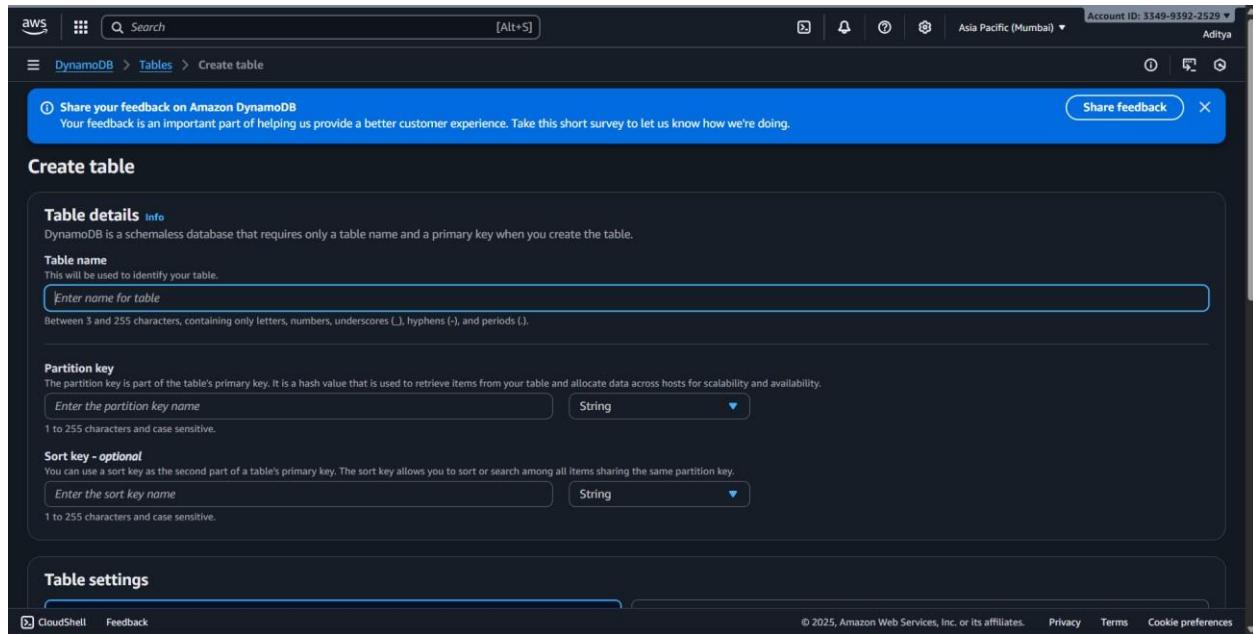


6. Use the **replica endpoint** for read queries (reporting, analytics).

5.Create and configure DynamoDB tables. Design efficient partition and sort keys. Implement Global Secondary Indexes. Set up DynamoDB Streams for real-time processing.

STEPS:

Navigate to **DynamoDB** service and Click on **Create table**.



Enter **Table name**: StudentData.

Define the **Partition key**:

- Attribute name: StudentID (String).

Define the **Sort key**:

- Attribute name: CourseID (String).
(This helps uniquely identify students enrolled in different courses.)

Leave default settings for **Table settings** or enable **Provisioned capacity** if you want to define read/write units.

Table details

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
 String
1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
 String
1 to 255 characters and case sensitive.

Table settings

Default settings
The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose "Customize settings".

Customize settings
Use these advanced features to make DynamoDB work better for your needs.

Default table settings
These are the default settings for your new table. You can change some of these settings after creating the table.

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

Click Create table.

DynamoDB

- Dashboard
- Tables**
- Explore items
- PartiQL editor
- Backups
- Exports to S3
- Imports from S3
- Integrations
- Reserved capacity
- Settings

DAX

- Clusters
- Subnet groups
- Parameter groups
- Events

Tables (1)

The StudentData table was created successfully.

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity
StudentData	Active	studentID (\$)	CourseID (\$)	0	0	Off	☆	On-demand

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

Steps to Design Efficient Partition and Sort Keys

1. Choose StudentID as **Partition Key** → ensures even distribution of student records.

2. Choose CourseID as **Sort Key** → allows multiple course enrollments per student.
3. Add attributes such as:
 - Name (String) ○
Department (String)
 - Year (Number) ○
GPA (Number).

The screenshot shows the AWS DynamoDB console interface. On the left, the navigation bar includes 'DynamoDB', 'Explore Items', and 'StudentData'. The main area is titled 'Scan or query items' with 'Scan' selected. It shows the 'Table - StudentData' and 'Select attribute projection' dropdown set to 'All attributes'. Below are 'Filters - optional' and 'Run' and 'Reset' buttons. A success message at the bottom says 'Completed - Items returned: 0 - Items scanned: 0 - Efficiency: 100% - RCU consumed: 2'. The table view below shows one item:

	studentID (String)	CourseID (String)	Departme... (String)	GPA (Number)	Name (String)	Year (Number)
23MIP10070	CSE3015	Data science	9.9	Aditya Raj	3	

Steps to Implement Global Secondary Indexes (GSI)

1. Open the created **StudentData** table.
2. Go to **Indexes** → **Create index**.

The screenshot shows the AWS DynamoDB console with the 'StudentData' table selected. The left sidebar shows navigation options like Dashboard, Tables, Explore items, PartQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Under 'Tables', there is a 'DAX' section with Clusters, Subnet groups, Parameter groups, and Events. The main area displays the 'StudentData' table details, including its primary key (not shown), and its global secondary indexes. One index, 'StudentData', is listed with 'No global secondary indexes'. A 'Create index' button is located at the bottom right of this section.

3. Define **Partition key** for the GSI: ○ Example: Department (String).

4. Define **Sort key** for the GSI:

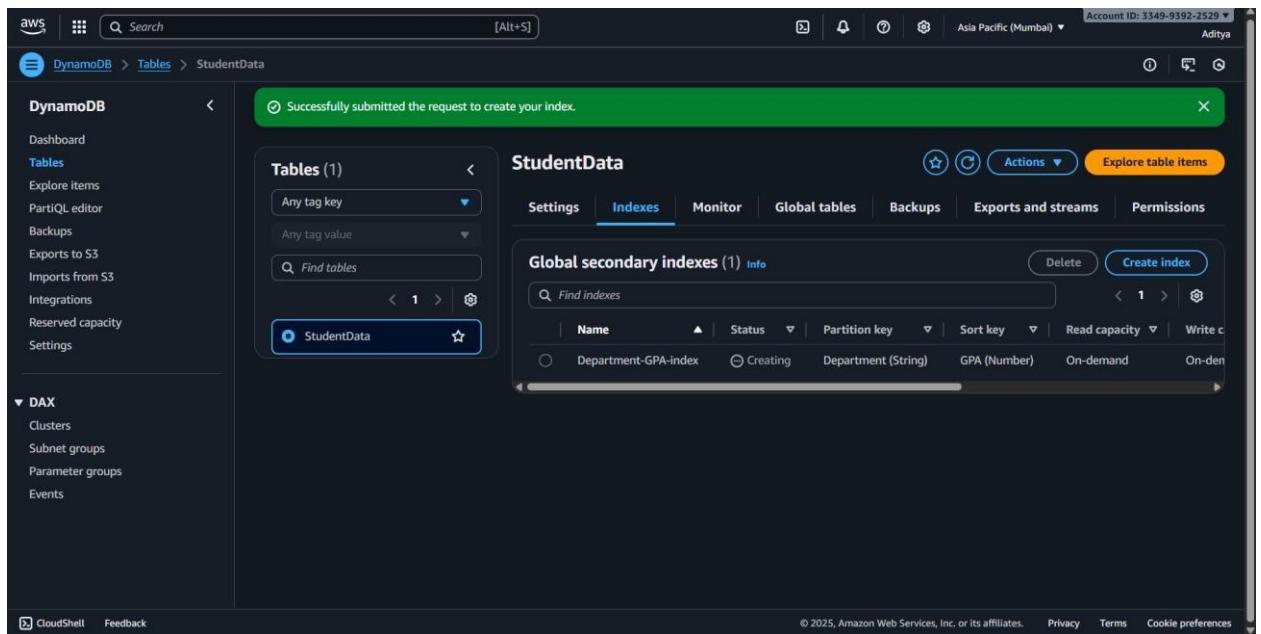
- Example: GPA (Number).

(This allows queries like “Find students by Department and sort by GPA.”)

Select **Projected attributes**: All or specific (Name, Year, GPA).

The screenshot shows the 'Create index' wizard for the 'StudentData' table. The first step, 'Index details', is completed with the following settings: Partition key is 'Department' (String type), Sort key is 'GPA' (Number type), and the Index name is 'Department-GPA-index'. The second step, 'Index capacity', shows 'Capacity mode' as 'On-demand'. A note at the bottom of this step states: 'Your global secondary indexes follow your base table's capacity mode. To switch to provisioned capacity, change your table's capacity mode to Provisioned in Settings.' The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

6. Click **Create index**.

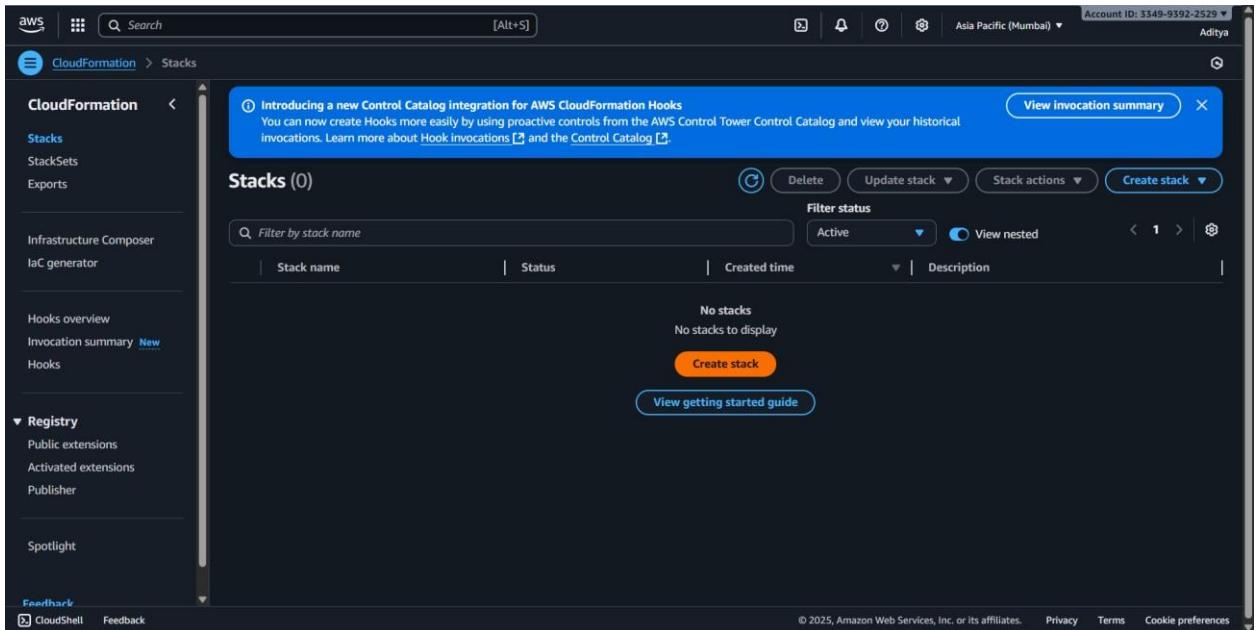


6.Create CloudFormation templates for infrastructure. Implement nested stacks and cross-stack references. Manage stack updates and rollbacks. Integrate with CI/CD pipelines.

Steps:

Open AWS Console → CloudFormation

- Click Create stack → With new resources (standard).



- Upload Root Template** ○ Upload `root-stack.yml` (this will call child stacks).
- Nested Stacks** ○ In `root-stack.yml`, reference child templates (e.g., `vpc.yml`, `ec2.yml`).
- Cross-Stack References** ○ In `vpc.yml`, define Outputs (e.g., VPC ID).
 - In `ec2.yml`, use `Fn::ImportValue` to fetch that VPC ID.
- Deploy Stack** ○ Click Next → Next → **Create stack**.
- Update & Rollback** ○ Modify a child template (e.g., change EC2 type). ○ Deploy again → CloudFormation shows update in progress.
 - If error, AWS automatically rolls back to last good state.
- CI/CD Integration** ○ In CodePipeline, add a stage to deploy CloudFormation template after build.

7. Deploy applications using Elastic Beanstalk. Configure application environments.

Implement blue-green deployments. Monitor application health and performance.

Step 1: Open Elastic Beanstalk

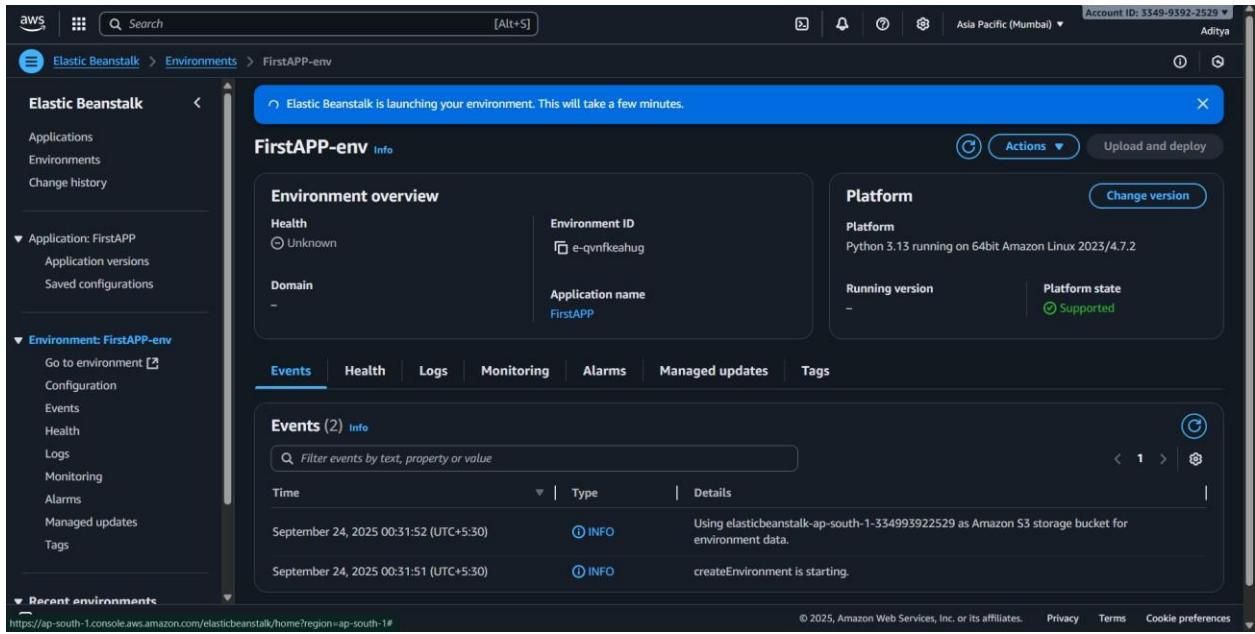
1. Sign in to **AWS Management Console**.
2. Search for **Elastic Beanstalk** in the search bar.
3. Click **Create Application**.

Step 2: Create an Application

1. Enter **Application Name** (e.g., my-eb-app).
2. Choose **Platform** (e.g., Python, Node.js, Java, or Tomcat).
3. For **Application code**, you can either:
 - o Upload your .zip application code, OR o
 - Choose **Sample application** (easier for lab testing).

The screenshot shows the 'Create environment' step in the AWS Elastic Beanstalk console. The 'Environment name' field is filled with 'FirstAPP-env'. The 'Domain' field contains '.ap-south-1.elasticbeanstalk.com' with a 'Check availability' button next to it. Under 'Platform', there are dropdown menus for 'Platform', 'Platform branch', and 'Platform version', all currently set to 'Choose a platform'. In the 'Application code' section, the 'Sample application' radio button is selected. At the bottom, there are 'Presets' and other navigation links like CloudShell and Feedback.

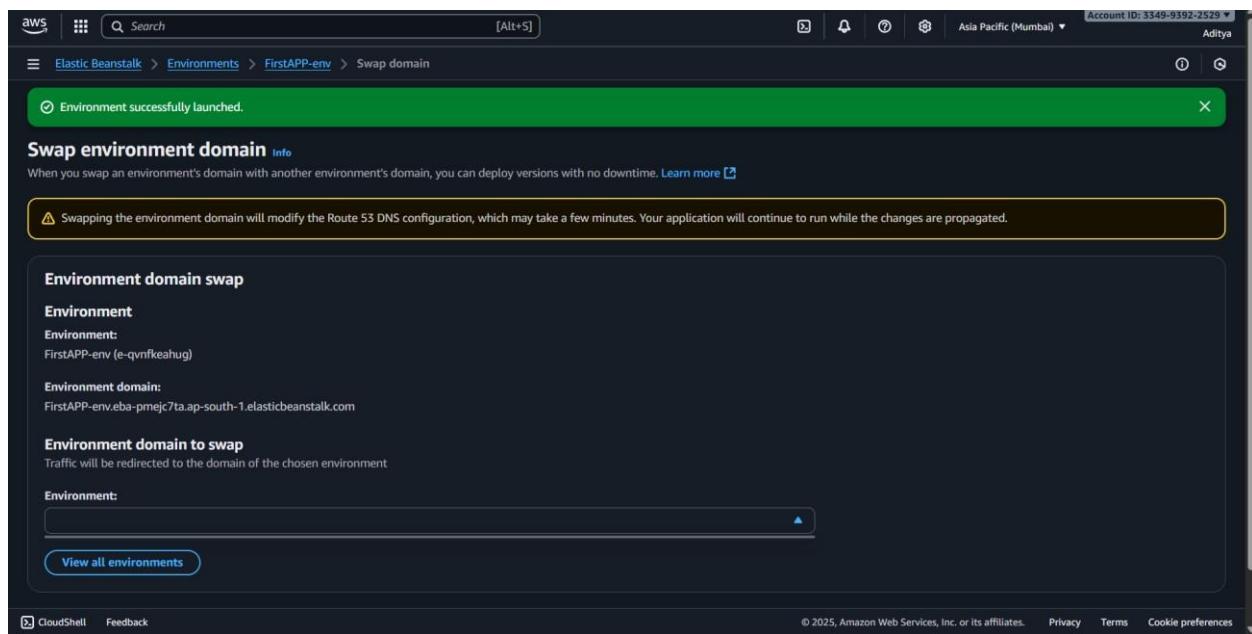
4. Click **Create application** → AWS will automatically create an **Environment** with EC2, Load Balancer, Auto Scaling, and S3 bucket.



Step 3: Configure the Environment

Step 4:

1. Click on your environment (e.g. Create a **new environment** → choose **Web server environment**.
 - Example: my-eb-app-green.
 - Upload the updated version of your app here.
2. Once it is deployed, test the **URL of green environment**.
3. To switch traffic:
 - Go to **Elastic Beanstalk** → **Environments**.
 - Select **Swap environment URLs** between **blue** and **green**.



8. Set up comprehensive logging and monitoring. Create custom metrics and alarms.

Implement log analysis and alerting, Build operational dashboards.

Steps:

1. Enable CloudTrail Logging:

- Open AWS CloudTrail → Click Create trail.

Screenshot of the AWS CloudTrail homepage.

Management & Governance

AWS CloudTrail

Continuously log your AWS account activity

Use CloudTrail to meet your governance, compliance, and auditing needs for your AWS accounts.

Create a trail with AWS CloudTrail

Get started with AWS CloudTrail by creating a trail to log your AWS account activity.

[Create a trail](#)

Pricing

[Pricing](#)

Getting started

- [What is AWS CloudTrail?](#)
- [How AWS CloudTrail works](#)
- [Services that integrate with AWS CloudTrail](#)

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

- Give a name for the trail.

Screenshot of the "Quick trail create" wizard.

Quick trail create

Trail details

Start logging management events by creating a trail with simplified settings. Logs are sent to an S3 bucket we create on your behalf. To choose a different bucket or additional events, go to the full [Create trail](#) workflow.

A trail created in the console is a multi-region trail. [Learn more](#)

Trail name

Enter a display name for your trail.

3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

Trail log bucket and folder

aws-cloudtrail-logs-334993922529-3cabf0a8

Logs will be stored in aws-cloudtrail-logs-334993922529-3cabf0a8/AWSLogs/334993922529

Info Though there is no cost to log these events, you incur charges for the S3 bucket that we create to store your logs.

[Cancel](#) [Create trail](#)

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

- Choose or create an S3 bucket to store logs.
- Enable CloudWatch Logs integration (optional for real-time monitoring).
- Click Create.

The screenshot shows two consecutive pages from the AWS CloudTrail 'Create trail' wizard.

Create trail (Step 2)

- General details:** A trail named 'FirstTrail' is being created. It is a multi-region trail. The 'Enable for all accounts in my organization' option is unchecked.
- Storage location:** The 'Create new S3 bucket' option is selected, and the bucket name is 'aws-cloudtrail-logs-334993922529-87e754fb'. The 'Use existing S3 bucket' option is also available.
- Trail log bucket and folder:** The bucket name and folder prefix are displayed as 'aws-cloudtrail-logs-334993922529-87e754fb/AWSLogs/334993922529'.
- Log file SSE-KMS encryption:** Enabled.
- Customer managed AWS KMS key:** New.
- AWS KMS alias:** An input field for entering a KMS alias is present.

Trail configuration (Step 3)

The trail 'FirstTrail' has been successfully created. Key details include:

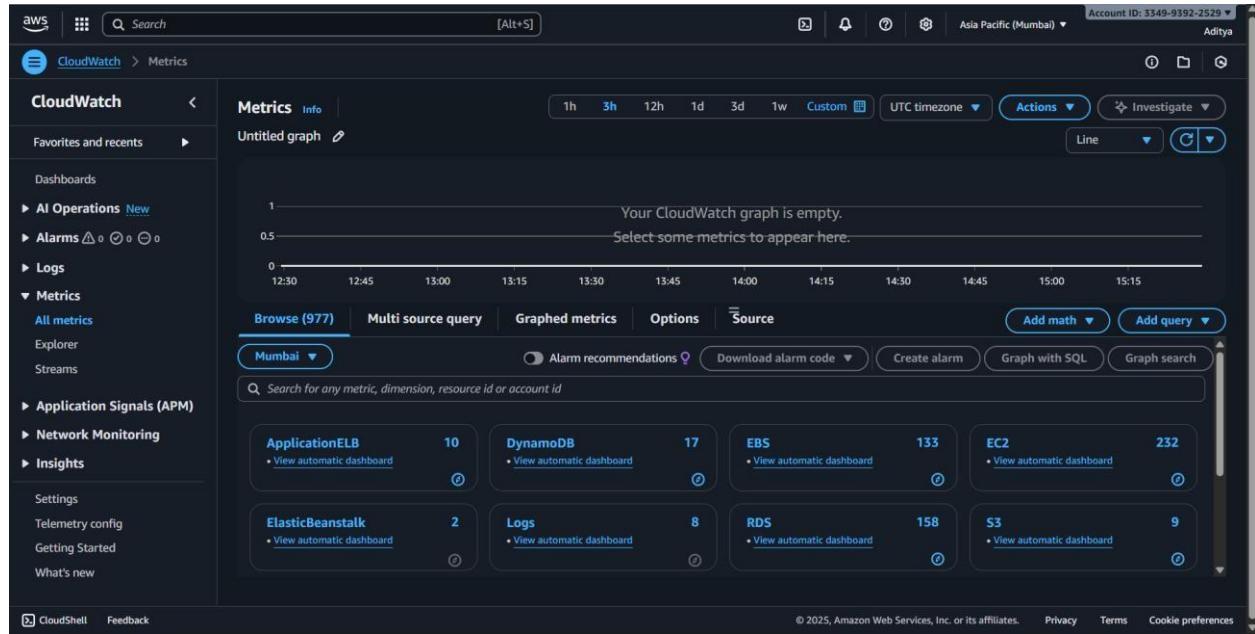
- Trail logging:** Logging is enabled.
- Trail name:** FirstTrail.
- Multi-region trail:** Yes.
- Apply trail to my organization:** Not enabled.
- Trail log location:** aws-cloudtrail-logs-334993922529-Scabf0a8/AWSLogs/334993922529.
- Log file validation:** Disabled.
- SNS notification delivery:** Disabled.
- Last log file delivered:** -
- Last file validation delivered:** -
- Last SNS notification:** -

CloudWatch Logs: No CloudWatch Logs log groups are configured for this trail.

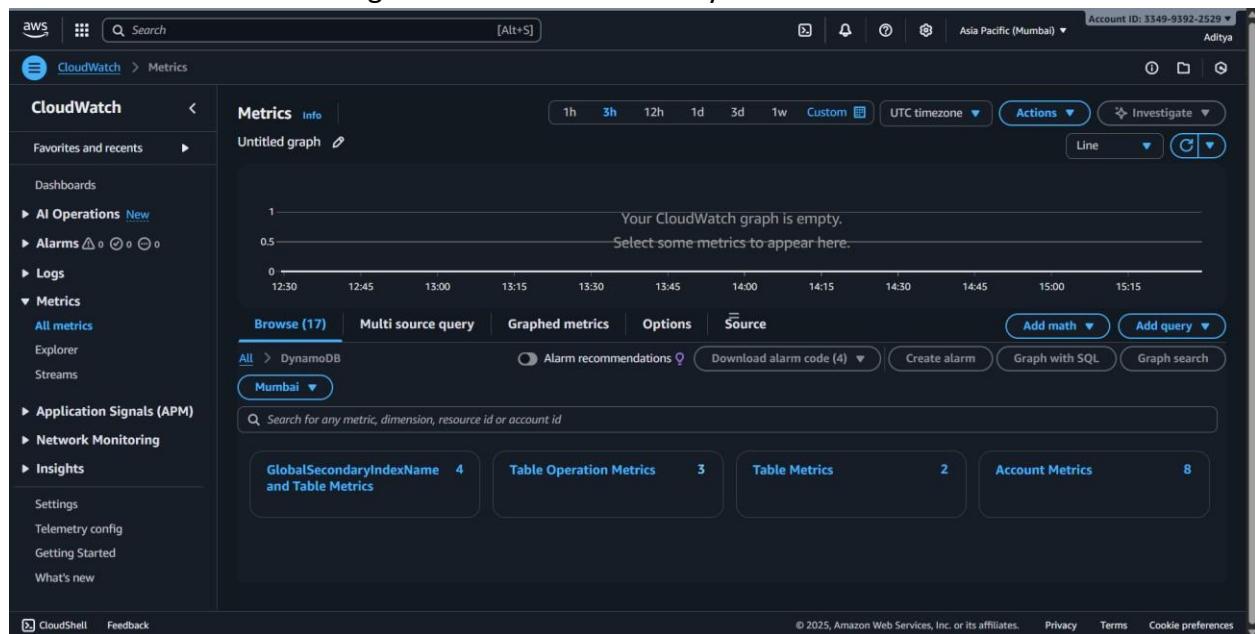
Tags: No tags are applied to this trail.

2. Enable CloudWatch Monitoring:

- Open AWS CloudWatch.
- Check default metrics for EC2, RDS, Lambda, etc.

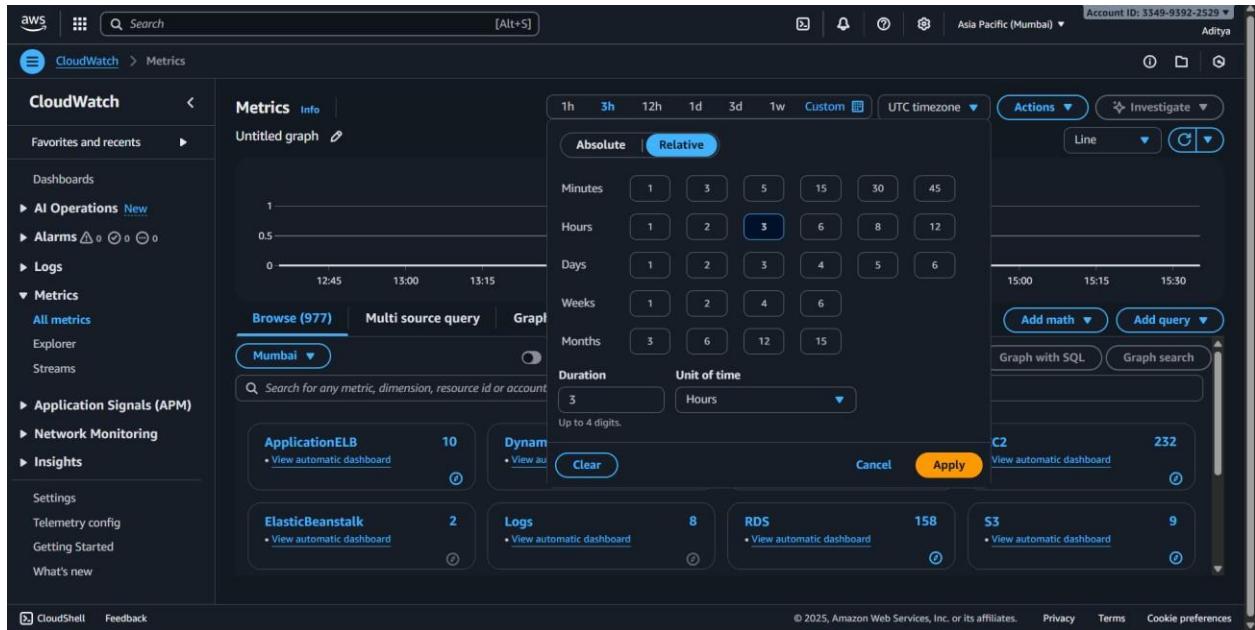


- Make sure monitoring is enabled for resources you want to track.



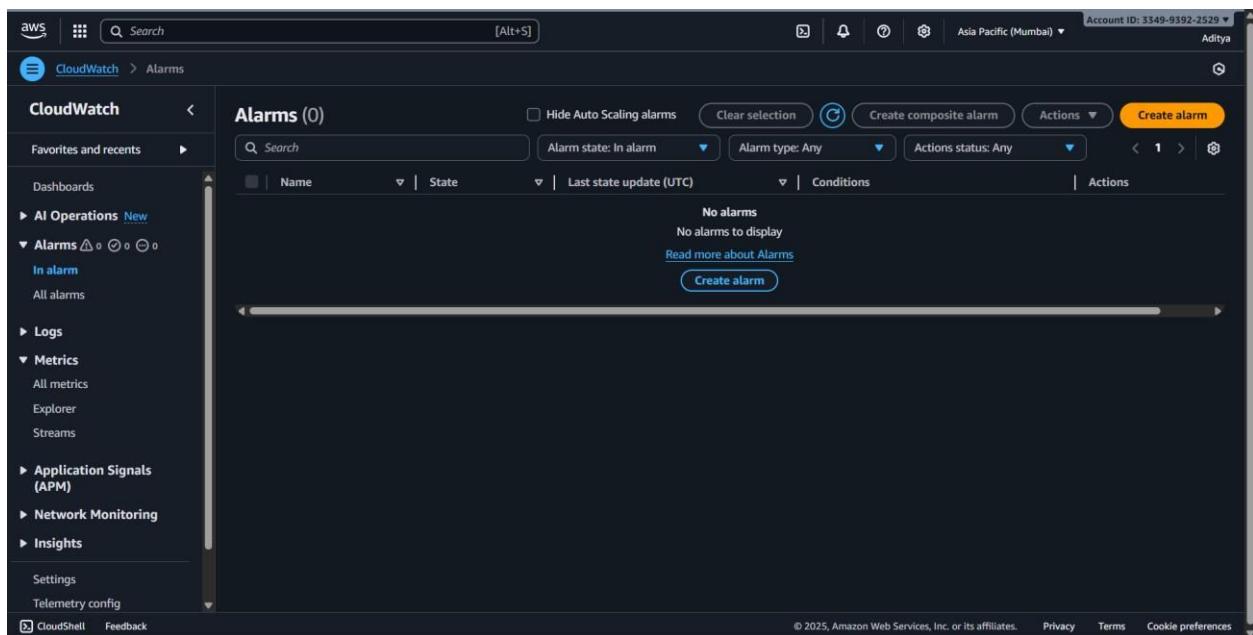
3. Create Custom Metric:

- In CloudWatch → Metrics → Click All Metrics → Create custom metric.
- Publish a metric using AWS CLI or console (e.g., number of requests or application-specific data).



4. Create Alarms:

- CloudWatch → Alarms → Create alarm.



- Select the metric (CPU Utilization, custom metric, etc.).

Step 1 Specify metric and conditions

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

No unit

1
0.906
0.5
0

13:00 13:05 13:10 13:14 13:18 13:20 15:00 15:30

ConsumedReadCapacityUnits

Conditions

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

- Set threshold (e.g., > 70%).
- Configure notification via SNS (email or SMS).
- Click Create alarm.

CloudWatch

Favorites and recents

Dashboards

▶ AI Operations New

▼ Alarms ▲ ○ ○ ○ ○
In alarm
All alarms

▶ Logs

▼ Metrics
All metrics
Explorer
Streams

▶ Application Signals (APM)

▶ Network Monitoring

▶ Insights

Settings
Telemetry config

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

BillsDynamo

Alarms (1)

Search
Alarm state: Any
Alarm type: Any
Actions status: Any
Hide Auto Scaling alarms

Graph

ConsumedReadCapacityUnits
ConsumedReadCapacityUnits > 70 for 1 datapoints within 5 minutes

1h 3h 12h 1d 3d 1w Custom UTC timezone

71
70
69

13:00 13:15 13:30 14:00 14:30 15:00 15:30

ConsumedReadCapacityUnits

In alarm OK Insufficient data Disabled actions

5. Log Analysis:

- CloudWatch → Logs → Create Log Group.

The screenshot shows the AWS CloudWatch Log groups interface. The left sidebar includes sections for Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics (All metrics, Explorer, Streams), Application Signals (APM), and Network Monitoring. The main area displays 'Log groups (3)' with a table:

Log group	Log class	Anomaly d...	Data pr...	Sensitiv...	Retention	Metric fi...
/aws/lambda/APIFunction	Standard	Configure	-	-	Never expire	-
/aws/lambda/welcomeFunction-24	Standard	Configure	-	-	Never expire	-
/aws/storagegateway/sgw-25EB644C	Standard	Configure	-	-	Never expire	-

At the bottom right of the table, there is a 'Create log group' button.

- Stream logs from EC2, Lambda, or CloudTrail.

The screenshot shows the AWS CloudWatch Log groups interface after creating a new log group. A green notification bar at the top states: 'Log group "FirstLogGroup" has been created.' The main area displays 'Log groups (4)' with the same table structure as the first screenshot, now including the new log group 'FirstLogGroup'.

- Use CloudWatch Logs Insights → Run queries like fields @timestamp, @message | filter @message like /ERROR/.

The screenshot shows the AWS CloudWatch Log groups interface. On the left, there's a navigation sidebar with sections like Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics, Application Signals (APM), and Network Monitoring. The main area is titled "FirstLogGroup" and displays "Log group details". It includes fields for Log class (Info, Standard), ARN (arn:aws:logs:ap-south-1:334993922529:log-group:FirstLogGroup:"), Creation time (Now), Retention (Never expire), and Stored bytes (-). To the right, there are sections for Metric filters (0), Subscription filters (0), Contributor Insights rules, KMS key ID, and Anomaly detection (Configure). Below this, tabs for Log streams, Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, Data protection, and Field are visible. At the bottom, a table for "Log streams (0)" shows columns for Actions, View in Logs Insights, Start tailing, Create log stream, and Search all log streams. A status bar at the bottom indicates the URL as https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1.

6. Alerting via Log Filters:

- CloudWatch Logs → Create metric filter.

The screenshot shows the AWS CloudWatch Log groups interface. The left sidebar is identical to the previous one. The main area shows a list of log groups: "/aws/lambda/APIFunction", "/aws/lambda/welcomeFunction-_24" (selected with a checked checkbox), and "/aws/storagegateway/sgw-25EB644C". A context menu is open over the selected log group, listing options: Delete log group(s), Edit retention setting(s), Create metric filter (which is highlighted in blue), Create contributor insights rules, Create data protection policy, Anomaly detection, Subscription filters, Export data to Amazon S3, and View all exports to Amazon S3. The status bar at the bottom indicates the URL as https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1.

- Define pattern (e.g., "ERROR" in logs).

The following log group(s) have been deleted:
FirstLogGroup

Step 1
Define pattern
Step 2
Assign metric
Step 3
Review and create

Create filter pattern
Specify the terms or pattern to match in your log events to create metrics.
 ERROR

Enable metric filter on transformed logs
When enabled, metric filter will be applied to transformed logs. When disabled, metric filter will be applied to original logs.

Field selection criteria - optional
Filtering options based on account and region provided for centralized log groups

Test pattern
⚠ The test does not include the field selection criteria

- Assign **alarm** to the filter → Choose SNS topic for notifications.

The following log group(s) have been deleted:
FirstLogGroup

Step 3
Review and create

Create filter pattern
Filter pattern
ERROR

Step 2: Metric

Assign metric

Filter name FirstFilter	Metric name triggerMetric
Metric namespace something that is triggered	Applied on transformed logs
Metric value 5	Default value
Unit -	

Cancel Previous Create metric filter

7. Build Operational Dashboard:

- CloudWatch → Dashboards → Create dashboard.

The screenshot shows the AWS CloudWatch Metrics Filter creation process. The left sidebar includes sections for CloudWatch (Logs, Metrics, Application Signals, Network Monitoring), Favorites and recents, Alarms (In alarm, All alarms), and Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights). The Metrics section is also visible. The main area shows a green notification bar stating "The following log group(s) have been deleted:" with "FirstLogGroup" listed. Below this is a "Step 3 Review and create" button. The "Create filter pattern" step shows a "Filter pattern" field containing "ERROR". The "Step 2: Metric" step shows the configuration of a metric filter:

Setting	Value
Filter name	FirstFilter
Metric namespace	something that is triggered
Metric value	5
Unit	-
Metric name	triggerMetric
Applied on transformed logs	-
Default value	-

Buttons at the bottom include "Cancel", "Previous", and "Create metric filter". The top right corner shows account information ("Account ID: 3349-9392-2529") and location ("Asia Pacific (Mumbai)").

- Add **widgets** for metrics, alarms, and log visualizations.

The screenshot shows the 'Add widget' configuration interface. On the left, under 'Data sources types', 'Cloudwatch' is selected. In the center, 'Widget Configuration' is set to 'Metrics'. The 'Widget type' section contains nine options: Line, Data table, Number, Gauge, Stacked area, Bar, Pie, and Explorer. Each option includes a description and a small icon. At the bottom right are 'Cancel' and 'Next' buttons.

- Save the dashboard for monitoring overall system health.

9. Design and implement complex workflows. Handle error conditions and retries.

Integrate multiple AWS services. Monitor workflow execution.

Steps

Experiment: Designing and Implementing Complex Workflows in AWS

Steps:

- 1. Create a Lambda Function** ○ Open AWS Console →
Lambda → Create function.

The screenshot shows the AWS Lambda 'Create function' wizard. In the 'Basic information' section, the function name is set to 'welcomeFunction-_24'. The runtime is chosen as 'Python 3.13'. The architecture is set to 'x86_64'. Under 'Permissions', there is a note about creating an execution role. The 'Additional configurations' section is collapsed. At the bottom right, there are 'Cancel' and 'Create function' buttons.

- Choose Author from scratch, give function name LabLambda, runtime Python

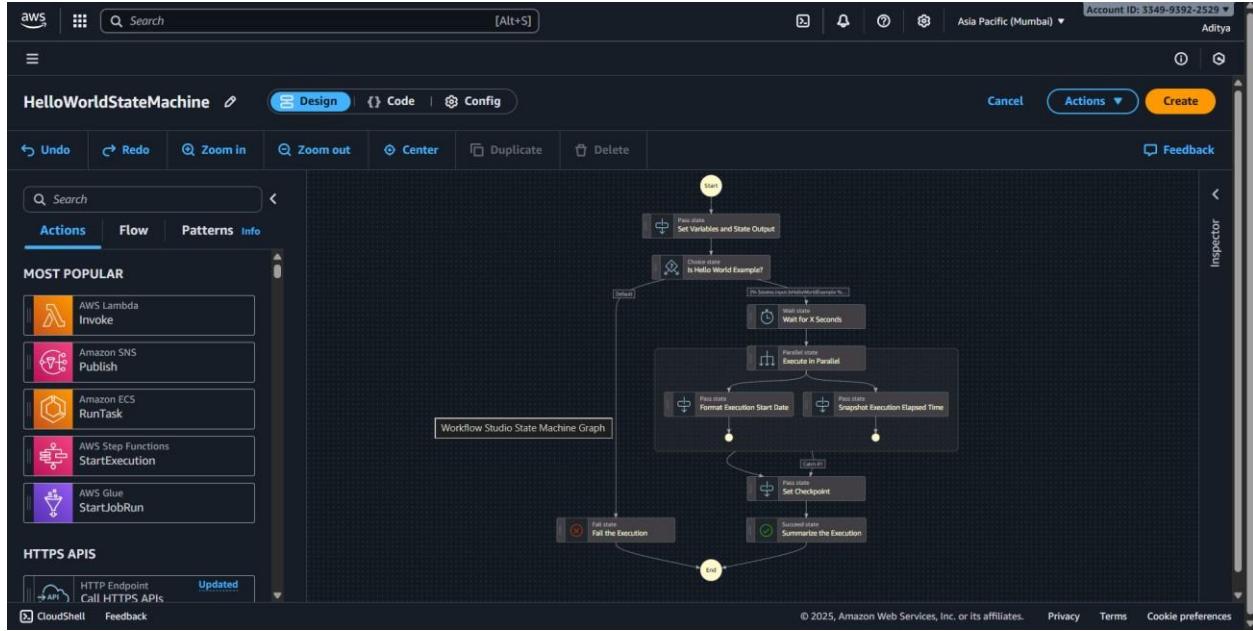
3.11.

- Use a small function that randomly succeeds or fails:
- ```
import random
def lambda_handler(event, context):
 if random.choice([True, False]):
 raise Exception("Simulated Error")
 return "Success"
```

The screenshot shows the AWS Lambda function details page for 'welcomeFunction-\_24'. A green success message at the top states: 'Successfully created the function welcomeFunction-\_24. You can now change its code and configuration. To invoke your function with a test event, choose "Test".'. The function ARN is listed as 'arn:aws:lambda:ap-south-1:334993922529:function:welcomeFunction-\_24'. The 'Code' tab is selected, showing the code source in the 'EXPLORER' panel with two files: 'lambda\_function.py' and 'lambda\_function.py'. There are tabs for 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'.

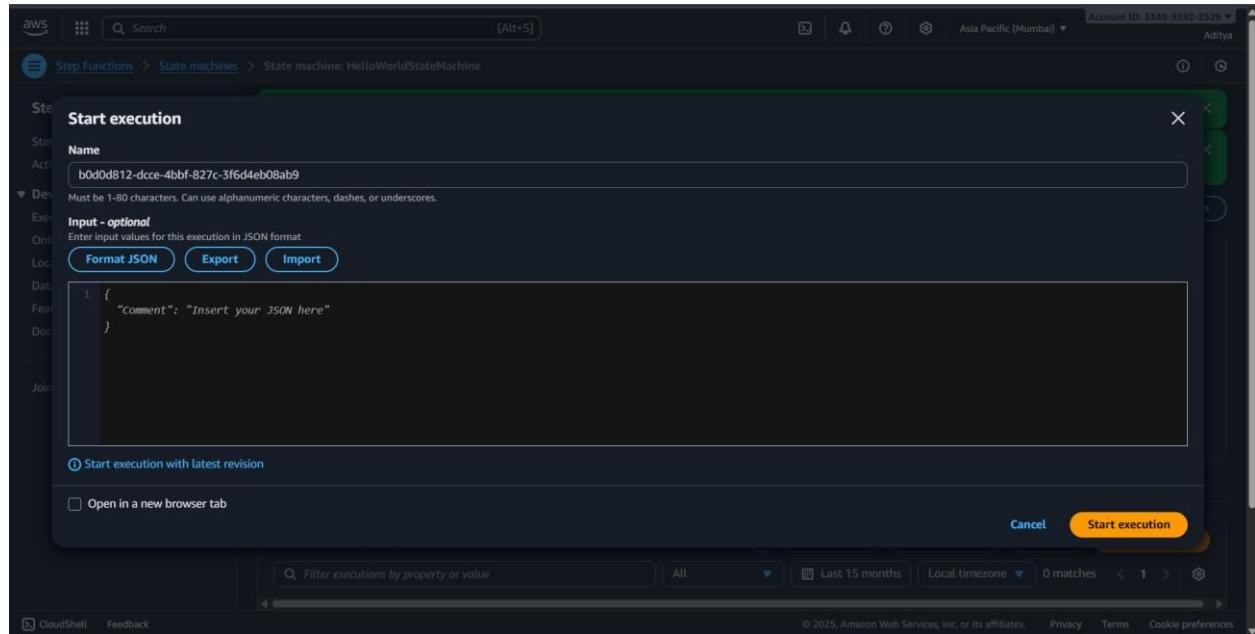
- Click Create function.

- 2. Open Step Functions**
- Go to AWS Console → Step Functions → Create state machine.



- Choose Standard Workflow → Design your workflow visually.

- 3. Add Workflow Steps**
- Add Task state → Choose Lambda function LabLambda.
  - Add Retry: ErrorEquals = ["States.ALL"], IntervalSeconds = 2, MaxAttempts = 2, BackoffRate = 2.0.
  - Add Catch: ErrorEquals = ["States.ALL"] → Next = FailState or error handling step.
  - Add Succeed state at the end.
- 4. Assign IAM Role**
- Use/create a role that allows Step Functions to invoke Lambda.
- 5. Start Execution**
- Click Start execution → Input {} (JSON).



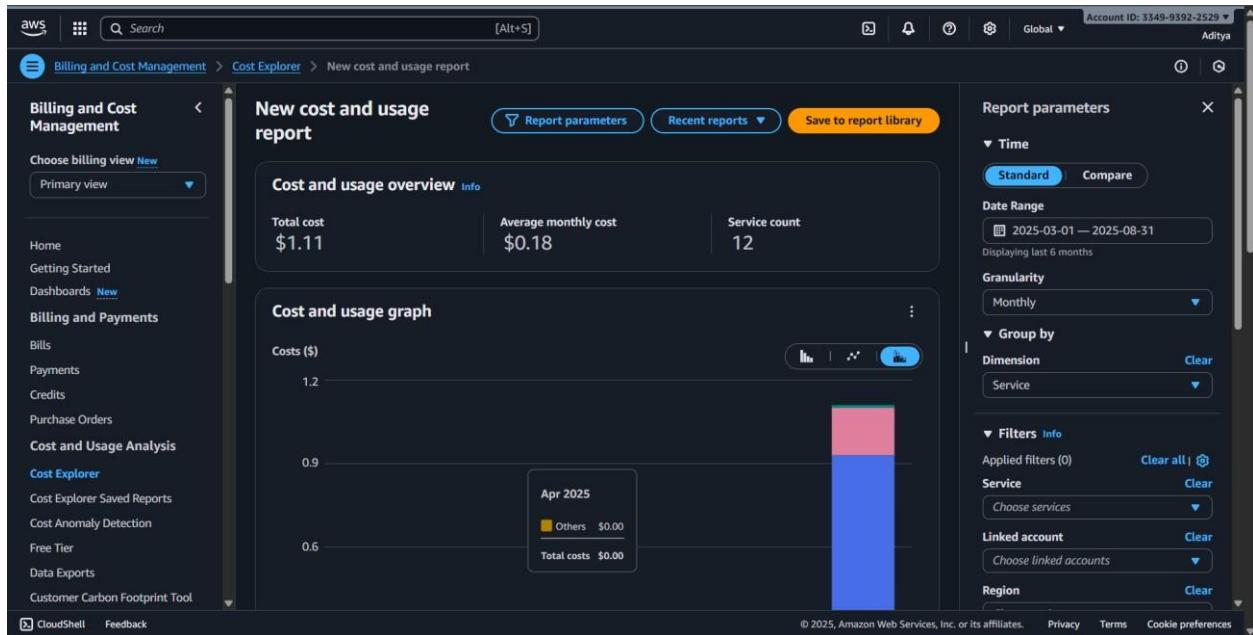
- Observe Execution Graph → green = success, red = error/retry.
- 6. Monitor Workflow**
  - Open CloudWatch → Logs → Check Lambda and Step Functions logs.
  - Optionally, create CloudWatch dashboard to track executions and success/failures.
- 7. Cleanup**
  - Delete Step Function state machine.
  - Delete Lambda function.
  - Delete CloudWatch log groups to reduce cost

## 10. Set up comprehensive cost monitoring. Create budgets and alerts. Analyze spending patterns. Implement cost optimization strategies

### 1. Set Up Comprehensive Cost Monitoring

1. Login to AWS Management Console.
2. Search for Cost Explorer in the search bar.
3. Enable Cost Explorer (if not already enabled).

4. This gives you detailed cost and usage reports for services, regions, and linked accounts.
5. Explore views such as:
  - o Monthly/Hourly Costs o Service-wise Cost Breakdown o Linked Account Usage



## 2. Create Budgets and Alerts

1. Go to AWS Budgets (in Billing & Cost Management).
2. Click Create Budget.

### 3. Select Cost budget → Next.

The screenshot shows the 'Create budget' page in the AWS Billing and Cost Management console. The 'Templates - new' section is displayed, with the 'Zero spend budget' option selected. This template creates a budget that notifies you once your spending exceeds \$0.01. Other options include 'Daily Savings Plans coverage budget', 'Monthly cost budget', and 'Daily reservation utilization budget'. Below the template section, there's a 'Zero spend budget - Template' configuration area with fields for 'Budget name' (set to 'My Zero-Spend Budget') and 'Email recipients' (with a note about a maximum of 10). The left sidebar lists various AWS services and tools related to billing and cost management.

### 4. Set:

- Period (Monthly/Quarterly/Yearly).
- Budgeted amount (e.g., \$10 for Free Tier experiments).

### 5. Configure alerts:

- Add email address.
- Set threshold (e.g., 80% of budget).

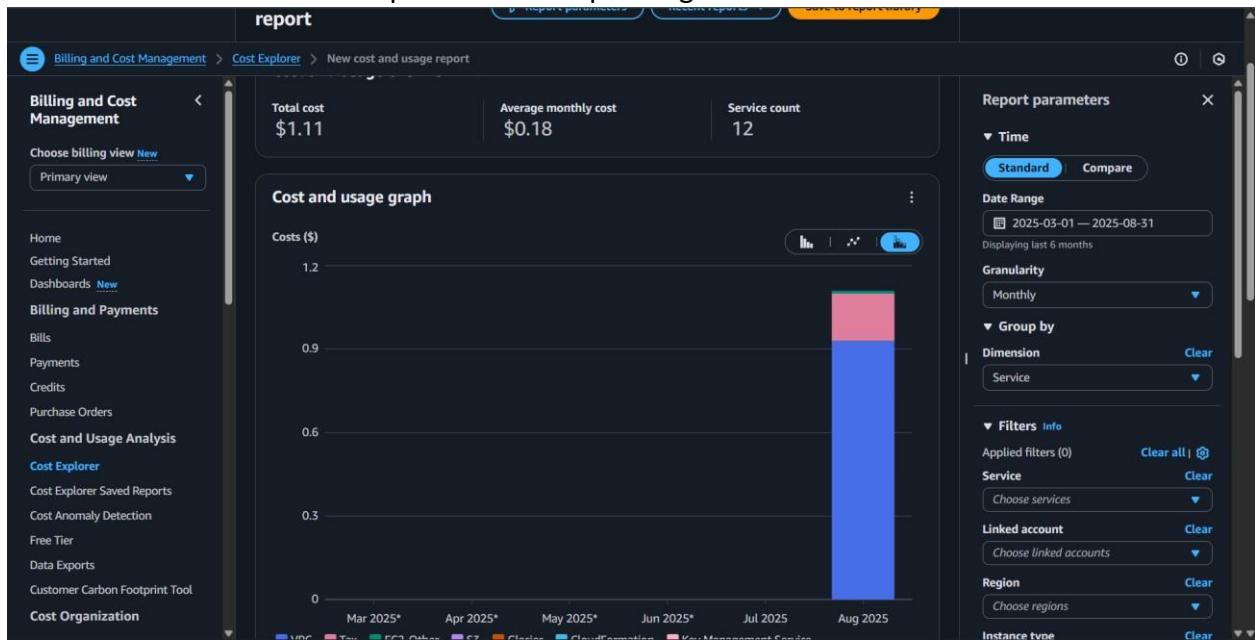
### 6. Review and click Create Budget.

Now you'll get an alert email if spending exceeds the threshold.

The screenshot shows the 'Overview' page in the AWS Billing and Cost Management console. A green success message at the top states 'Your budget My Zero-Spend Budget has been created successfully.' The main area displays a table titled 'Budgets (1)'. The table has columns for Name, Thresholds, Health status, Billing View, Budget, Amount..., and Forecast... The single entry in the table is 'My Zero-Spend Budget', which is marked as 'Exceeded (1)' and 'Healthy'. The 'Billing View' is set to 'Primary View' and the 'Budget' is '\$1.00'. The left sidebar shows the same navigation options as the previous screenshot.

### 3. Analyze Spending Patterns

1. Go to Cost Explorer → Reports.
2. Analyze usage by:
  - Service (EC2, S3, RDS, etc.). ○ Region (e.g., US East vs Asia Pacific).
  - Usage type (On-Demand, Reserved, Data transfer).
3. Identify which services cost the most.
4. Use Trends & Forecasts to predict future spending.



### 4. Implement Cost Optimization Strategies

1. EC2 Optimization
  - Use right-sizing recommendations (smaller instance types).
  - Stop or terminate unused instances.
  - Use Savings Plans or Reserved Instances for long-term workloads.
2. Storage Optimization
  - Move rarely used data to S3 Glacier or Infrequent Access.
  - Delete unused snapshots and EBS volumes.
3. Networking Optimization
  - Minimize data transfer across regions.
  - Use CloudFront CDN to reduce data transfer cost.
4. Automation
  - Use AWS Instance Scheduler to stop dev/test servers at night.
  - Use Trusted Advisor for cost-saving recommendations.

### 5. Monitoring and Continuous Improvement

1. Set regular reviews (weekly/monthly) in Cost Explorer.
2. Adjust budgets based on usage.
3. Continuously apply cost-saving best practices.

