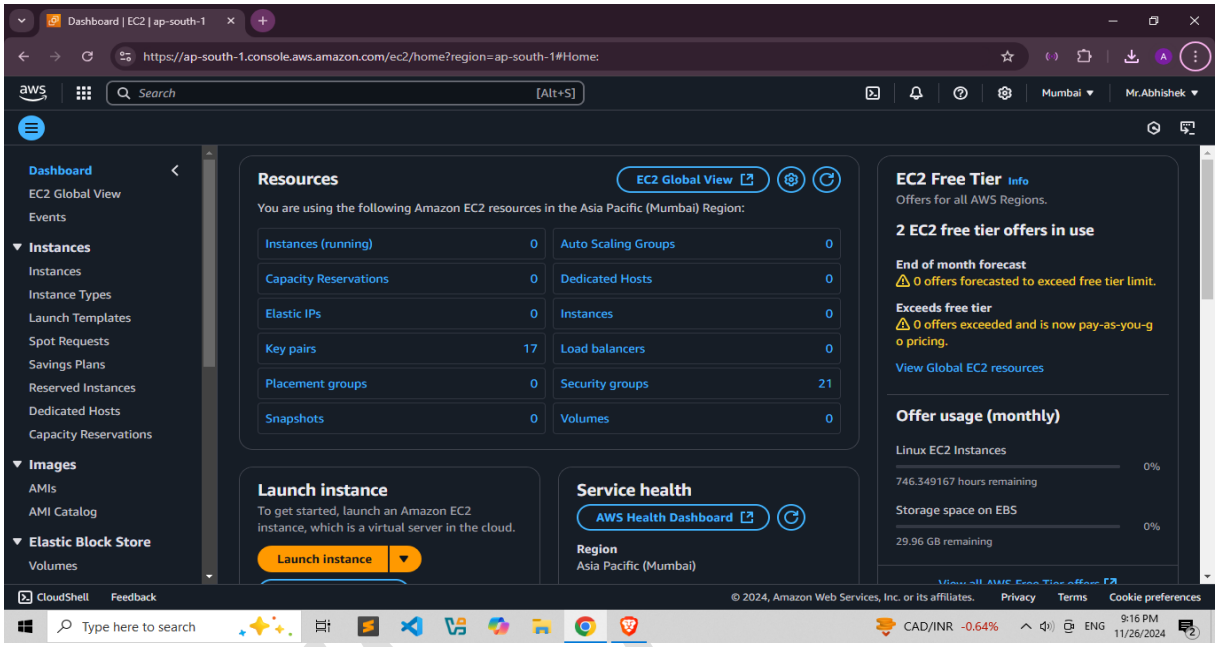
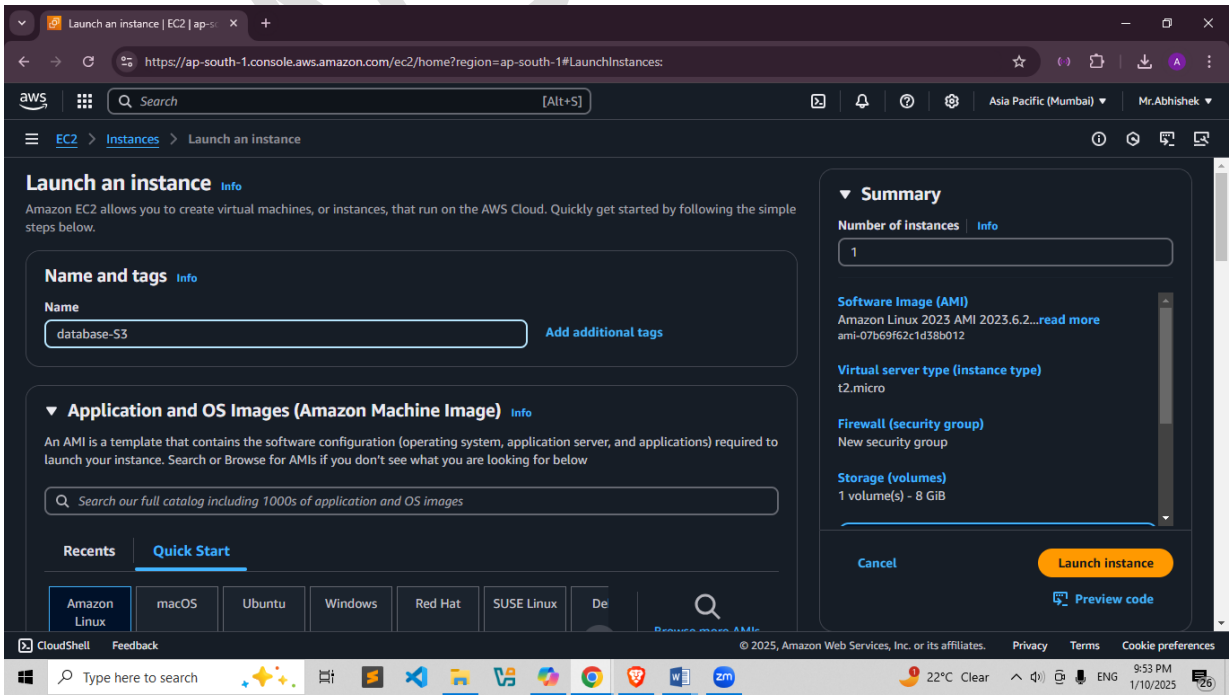
Install mariadb to the Linux instance and create DB. That DB backup Store in S3 Service (AWS Role).

Step 1: Add Name and Tags

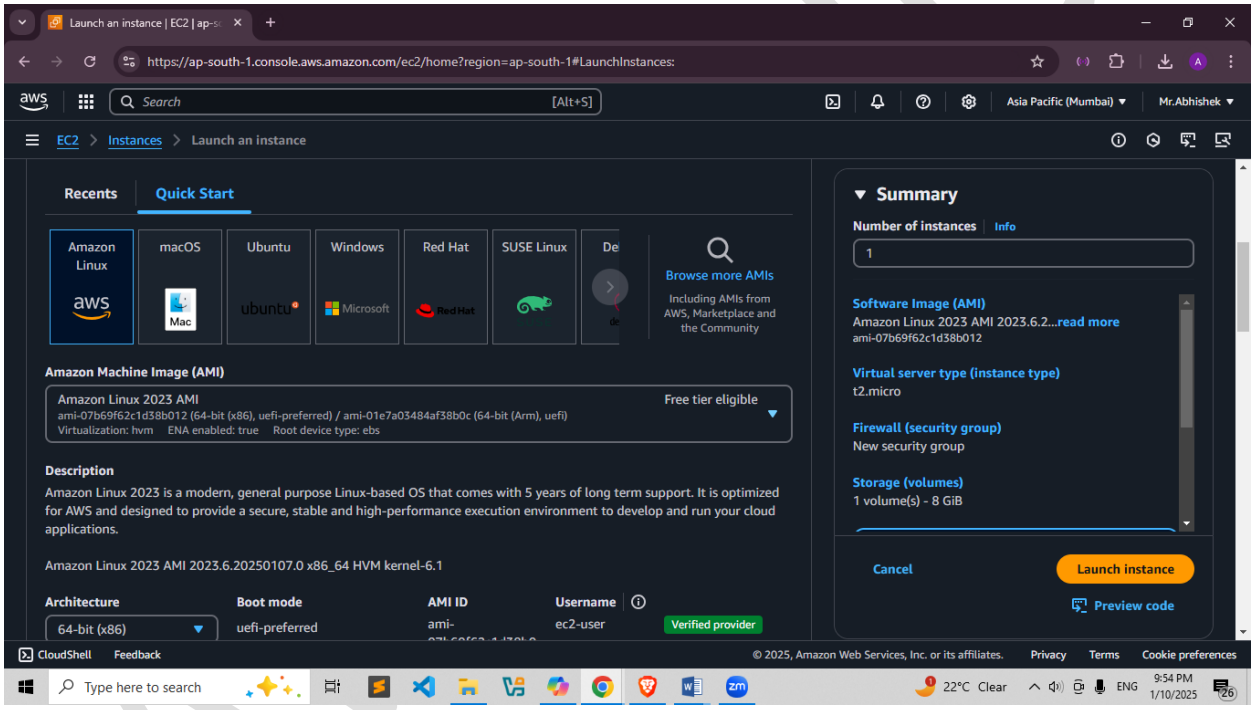
- Add tags to your instance for better management and identification.
  - Example: Key: Name, Value: MyInstance.





Step 2: Choose an Amazon Machine Image (AMI)

- Select an **AMI**, which is a pre-configured virtual machine template.
  - Examples:
    - **Amazon Linux 2 AMI** (Free-tier eligible).
    - **Ubuntu Server**.
    - **Windows Server** (if you need a Windows environment).
- Choose an AMI that suits your requirements for the operating system and software packages.

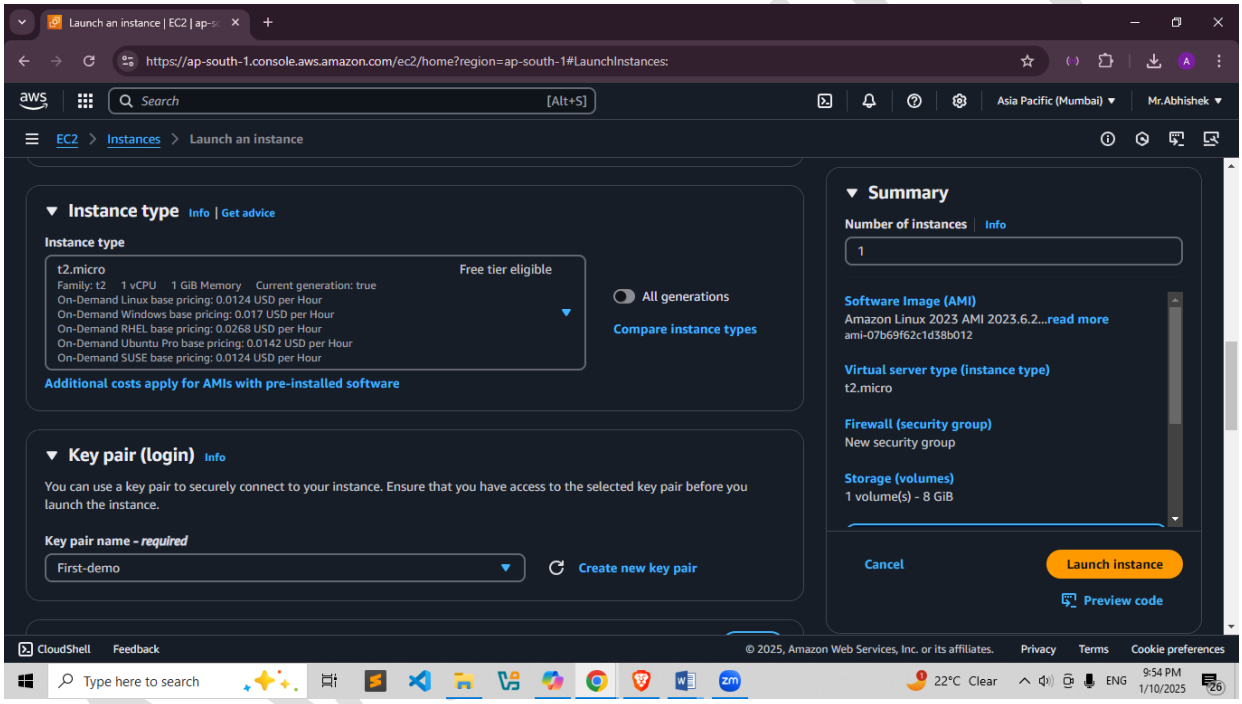


Step 3: Choose an Instance Type

Select the instance type based on your performance needs (CPU, memory, storage, etc.).

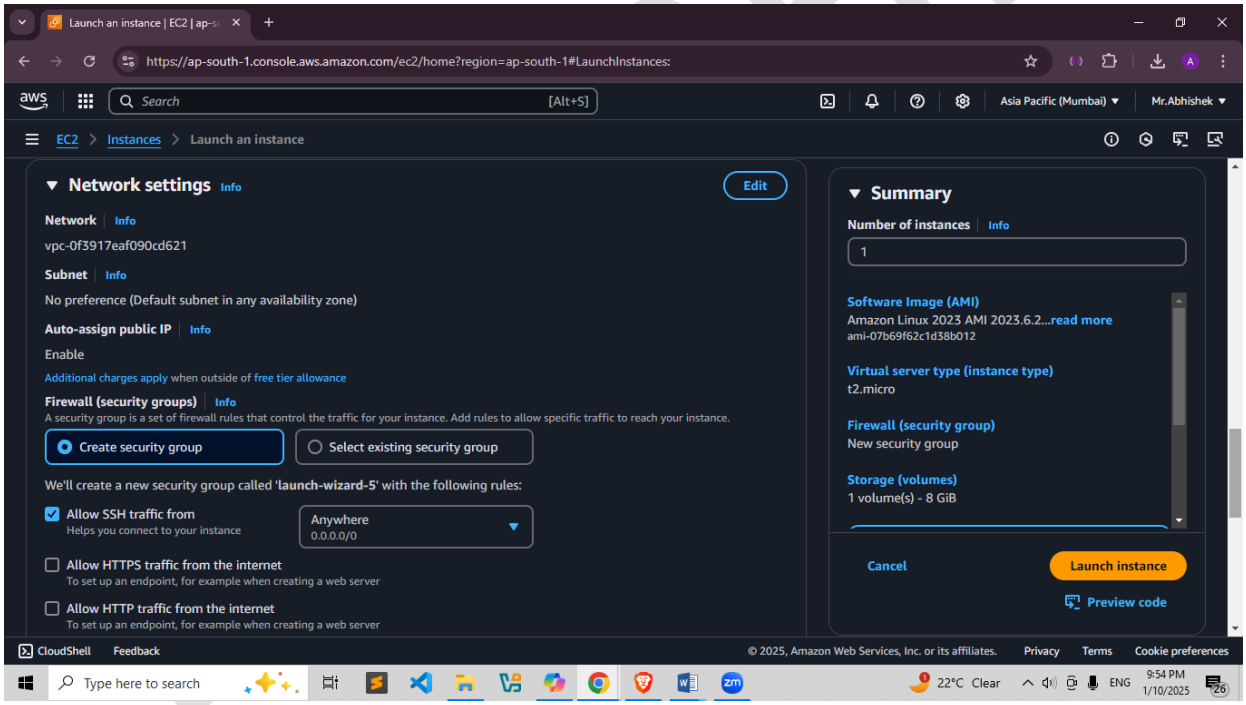
- For free-tier eligible users, choose **t2.micro** or **t3.micro**.

- When prompted, create a new key pair or use an existing one for SSH access:
  - Download the private key file (.pem) if creating a new key pair.



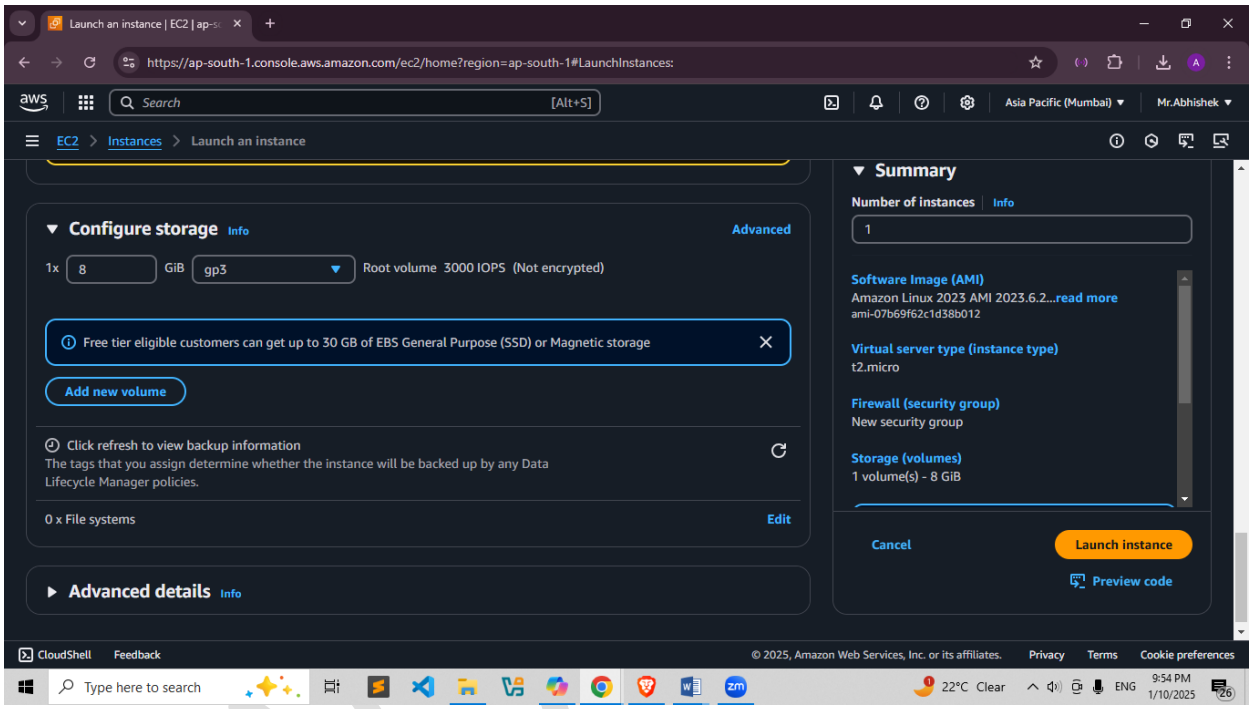
Step 4: Configure Instance Details

- Specify the details for your instance:
  - **Number of instances**: Default is 1.
  - **Network**: Select a Virtual Private Cloud (VPC).
  - **Subnet**: Choose a subnet for your instance.
  - **Auto-assign Public IP**: Enable this if you need internet access.
  - **IAM Role**: Assign an IAM role if necessary.
  - Advanced options: Configure placement groups, capacity reservations, etc.
- Click **Next** when done.



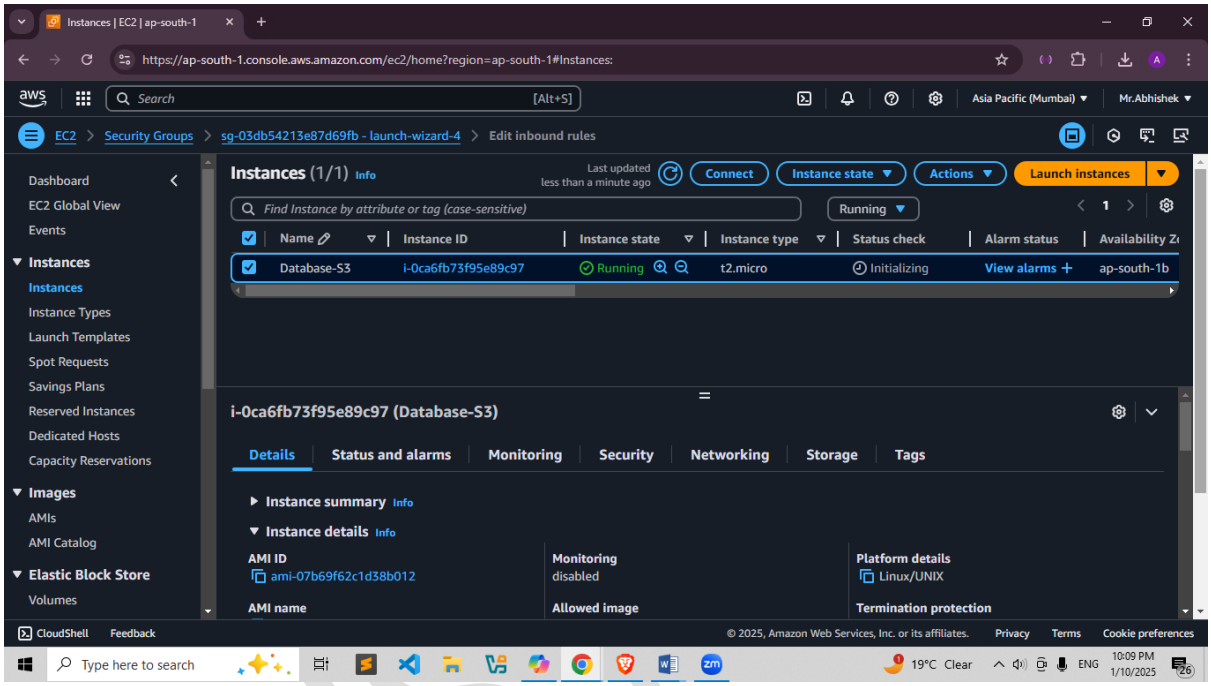
Step 5: Add Storage

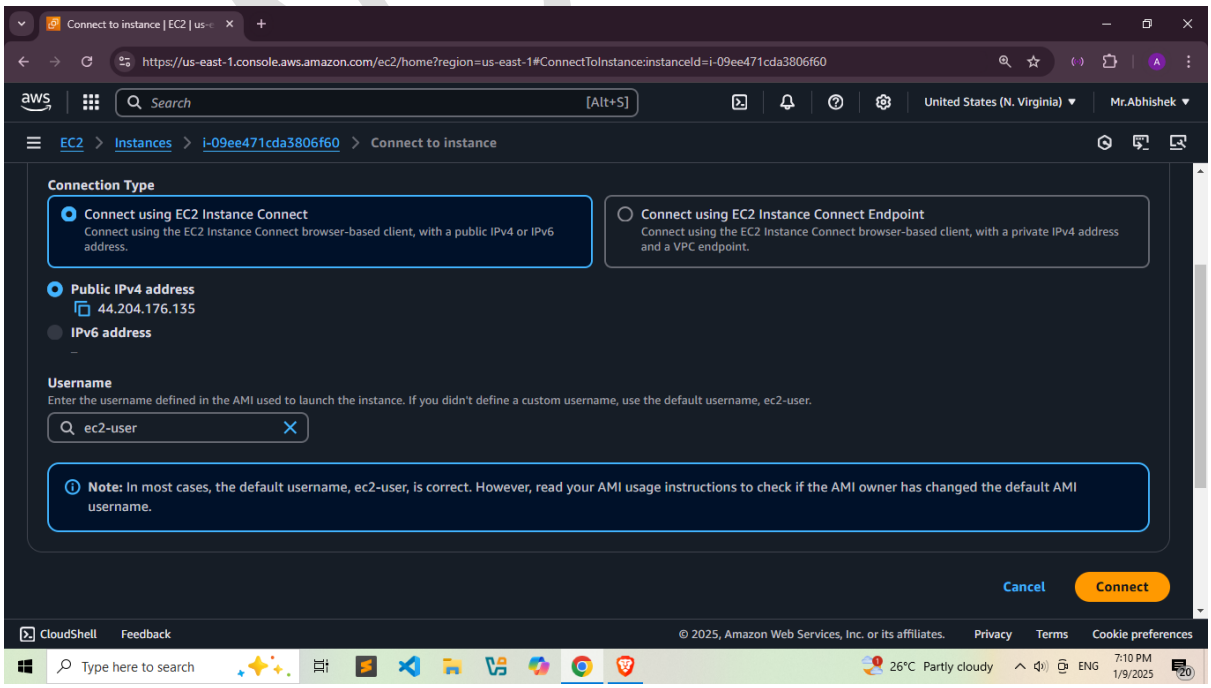
- Configure the storage for your instance:
  - Root volume size (default is 8 GiB for Amazon Linux).
  - Add additional volumes if required.
  - Choose the storage type (e.g., General Purpose SSD).



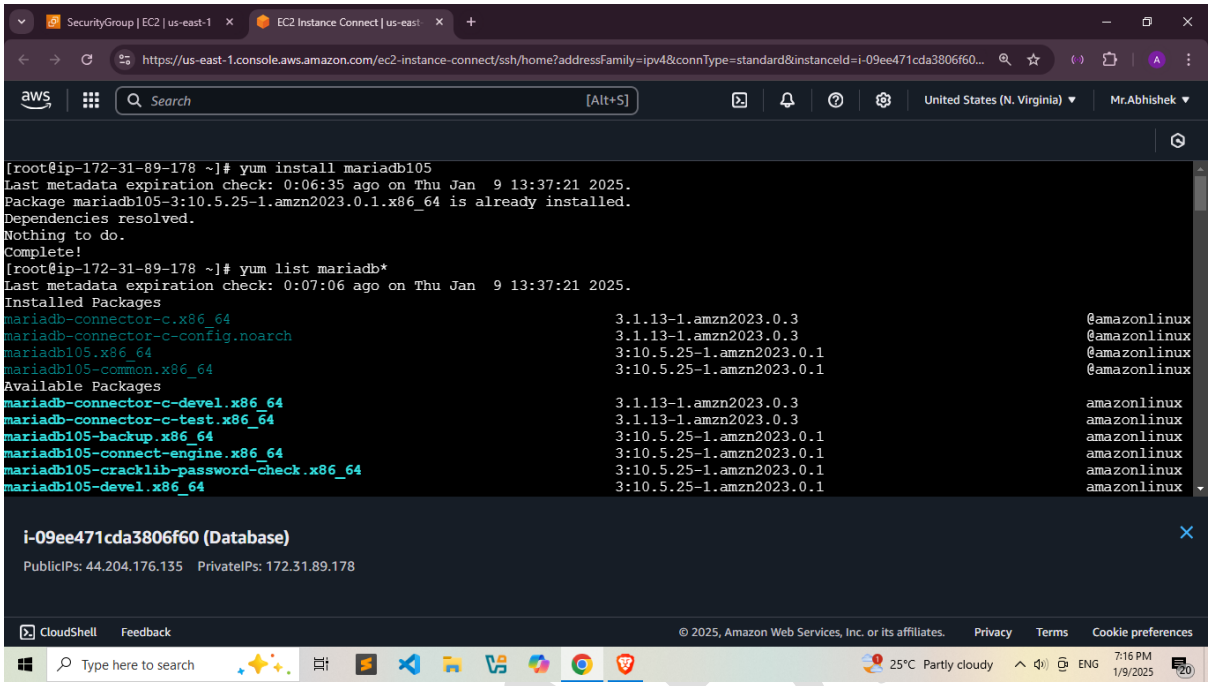
Step 6: Instance to Launch

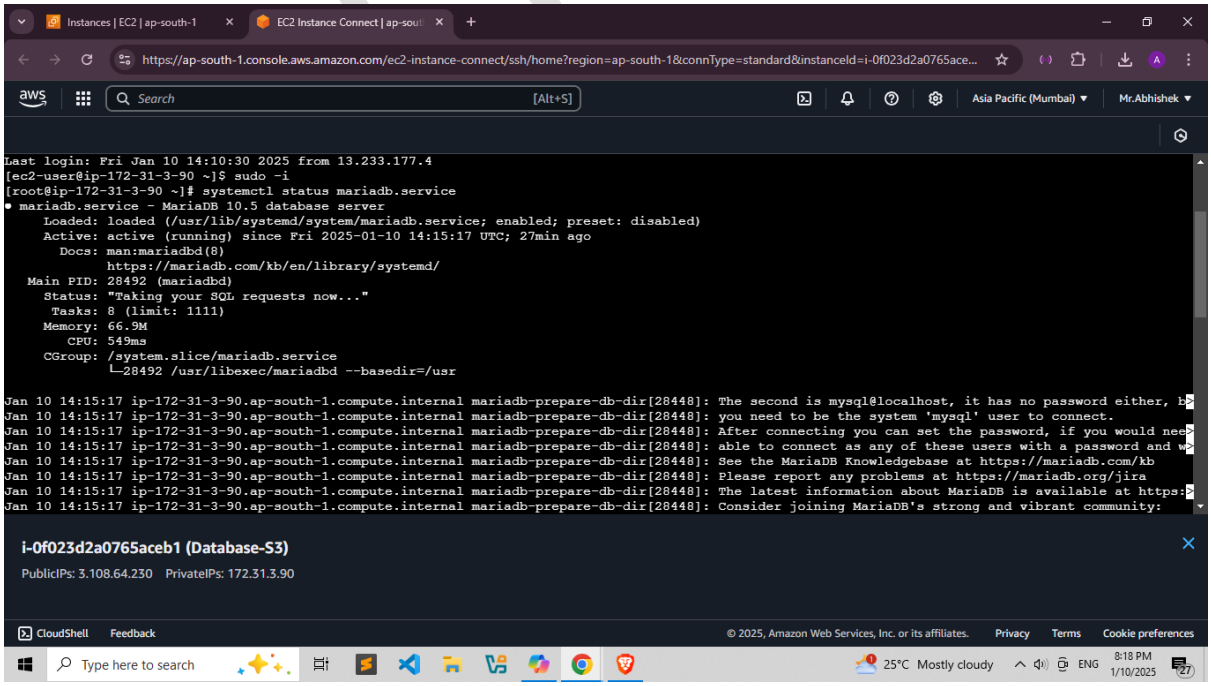
- You will be redirected to a confirmation page.
- Click **View Instances** to go to the EC2 Dashboard and monitor the status of your instance.



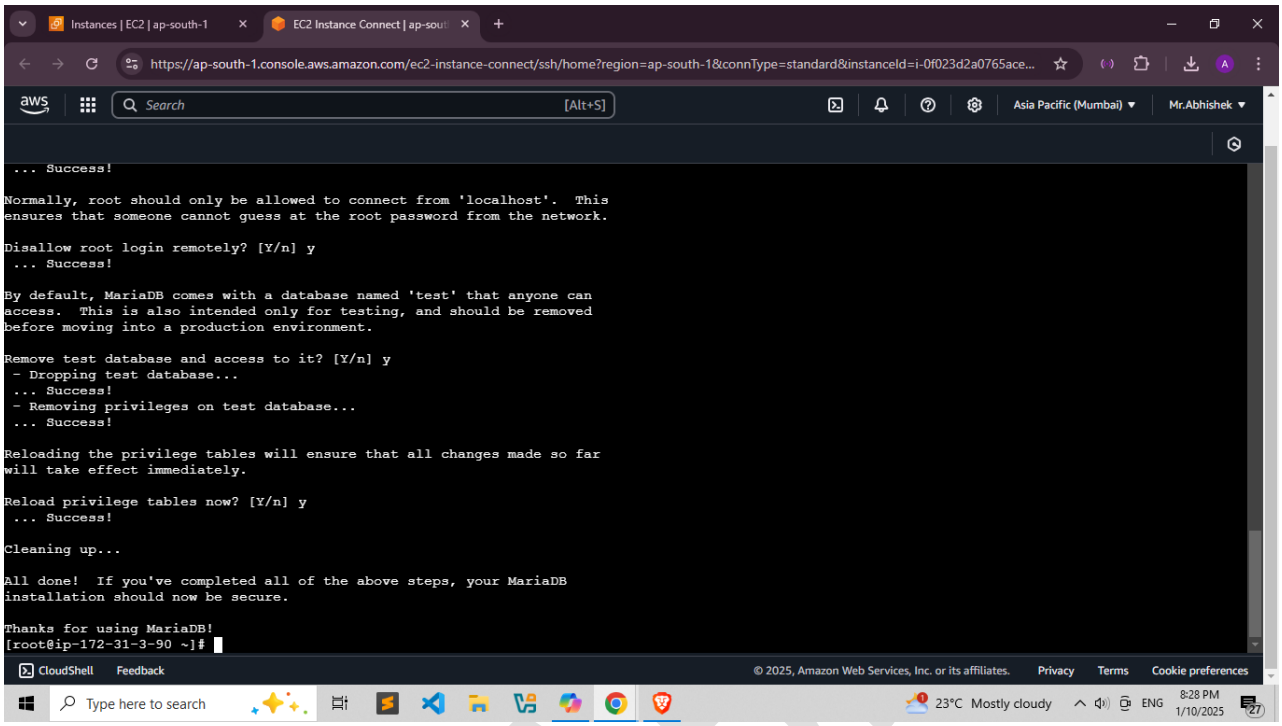


Install MariaDB: Install MariaDB using the package manager. On redhat-based systems like centos, use:

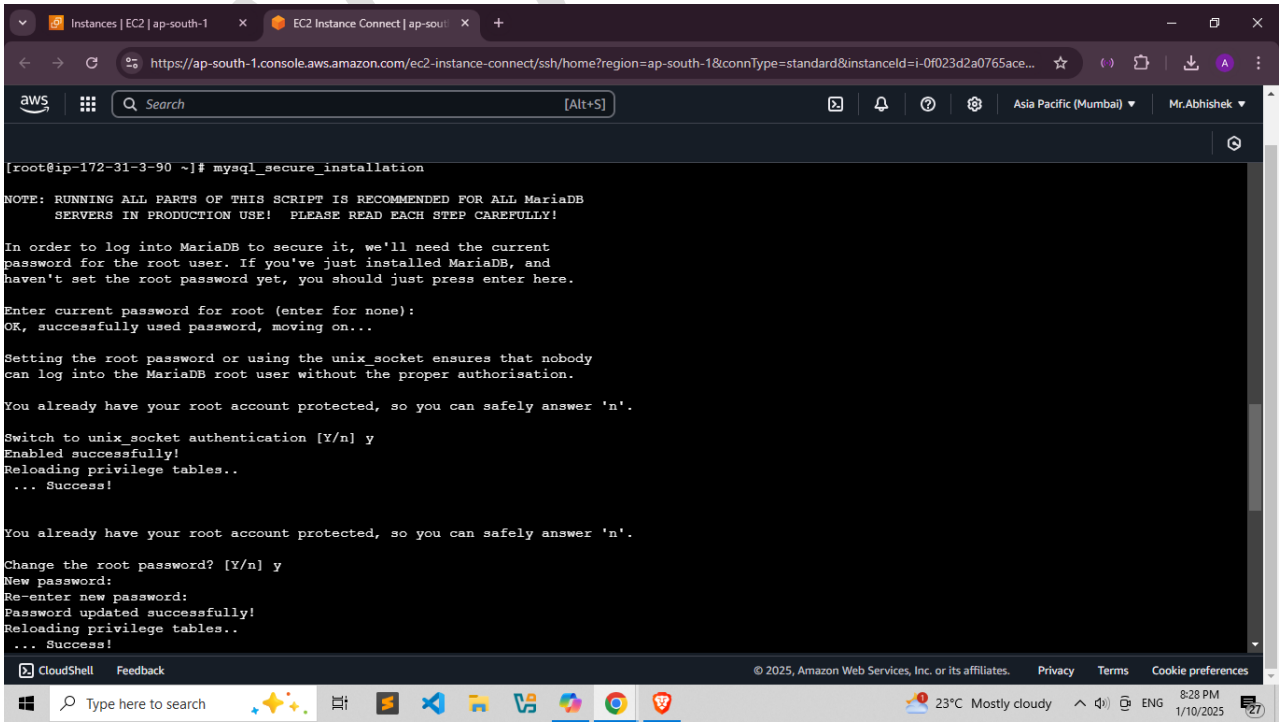




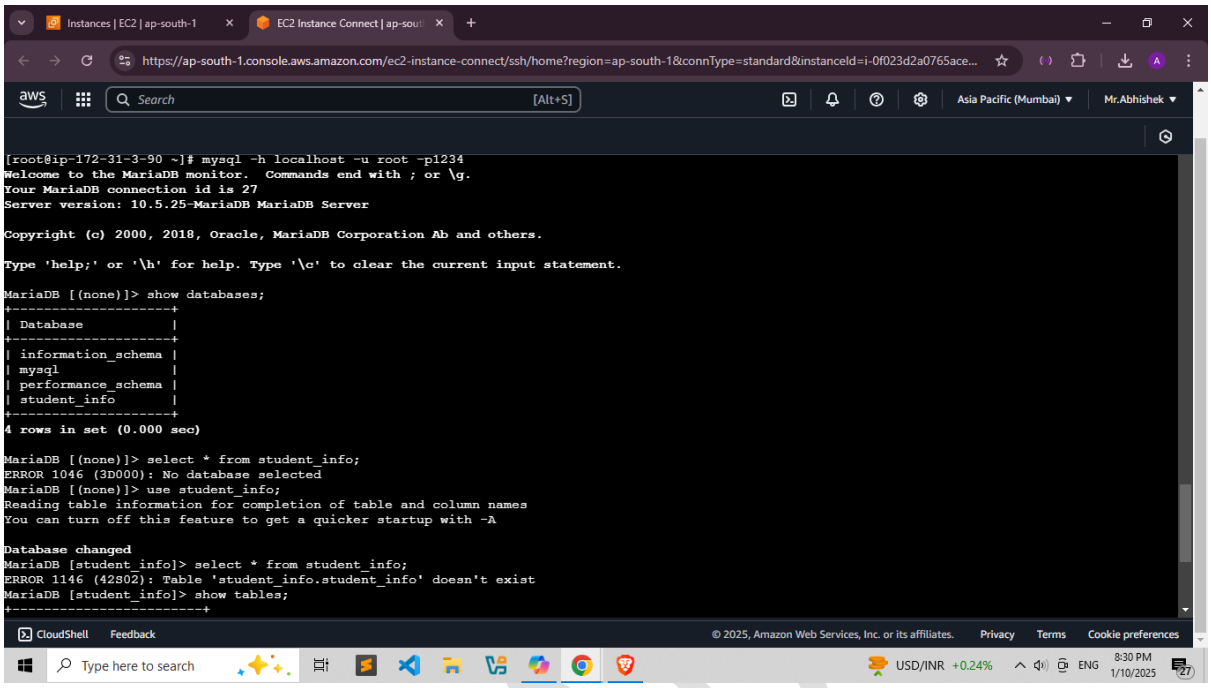
Secure MariaDB Installation: Run the security script to set up security options.



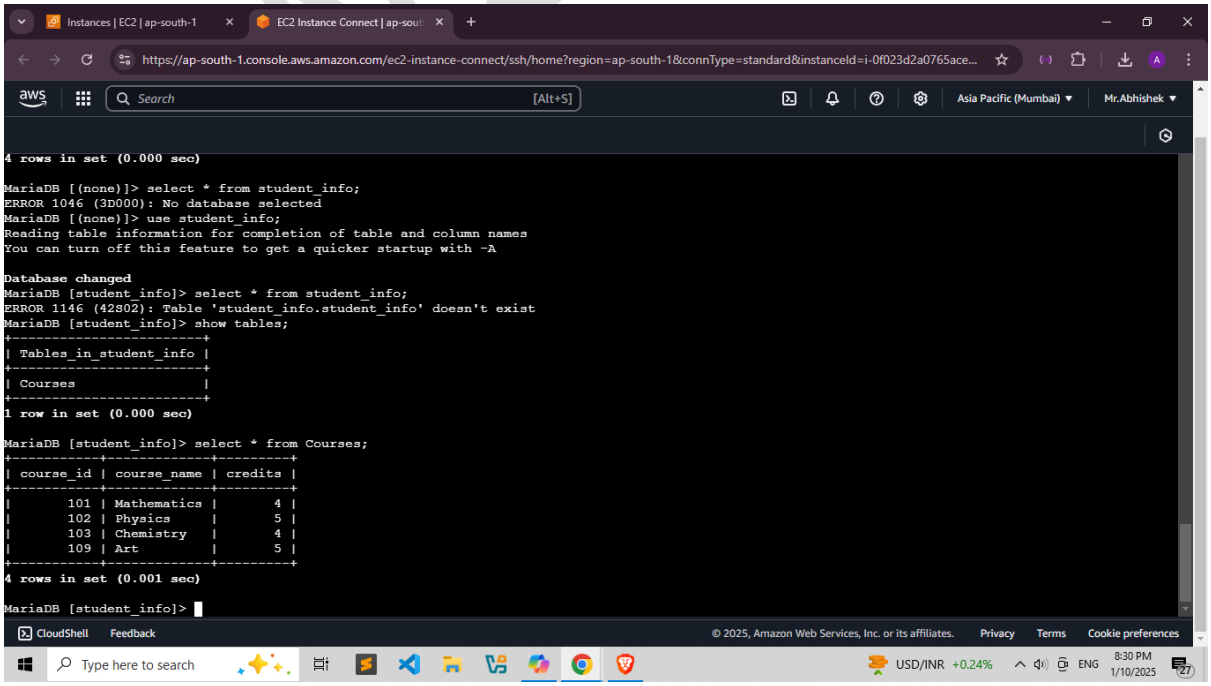
- **Change the Root Password**: You will be prompted to change the root password. Follow the prompts to set a new password:



- Switch to user **sudo mysql -u root –p1234.**
- Write **show databases;** command.

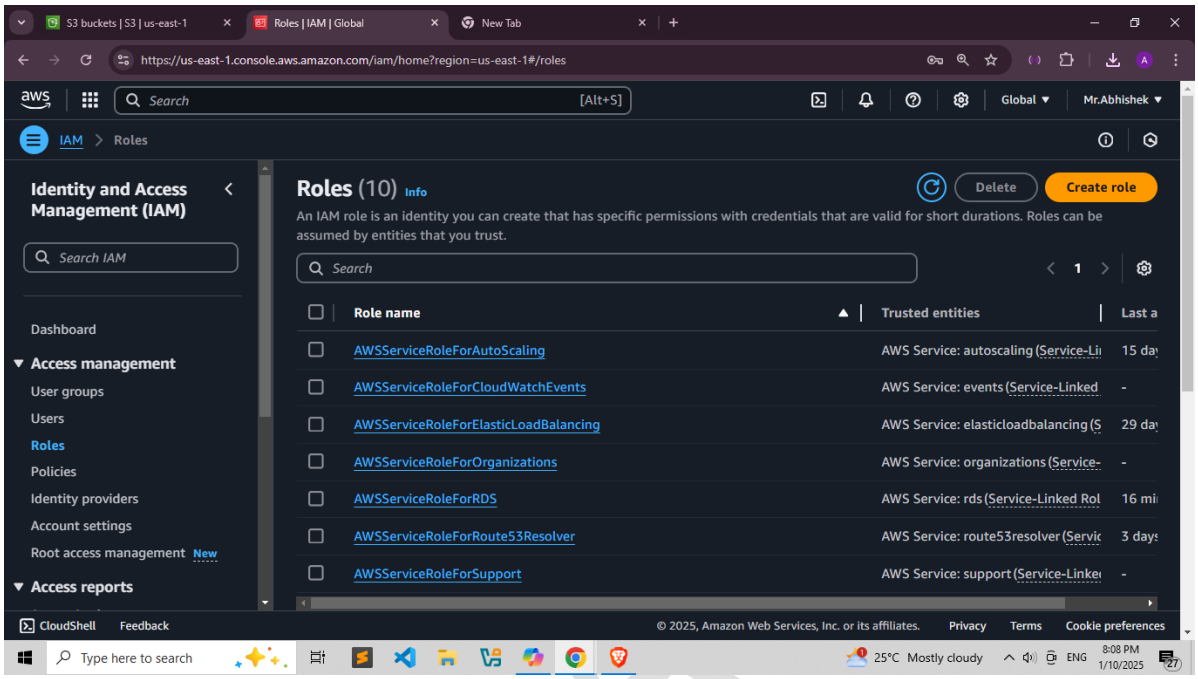


- To switch to a different database in MariaDB, you use the **(USE database_name;)**
- Create Table and Fields also using MySQL commands.



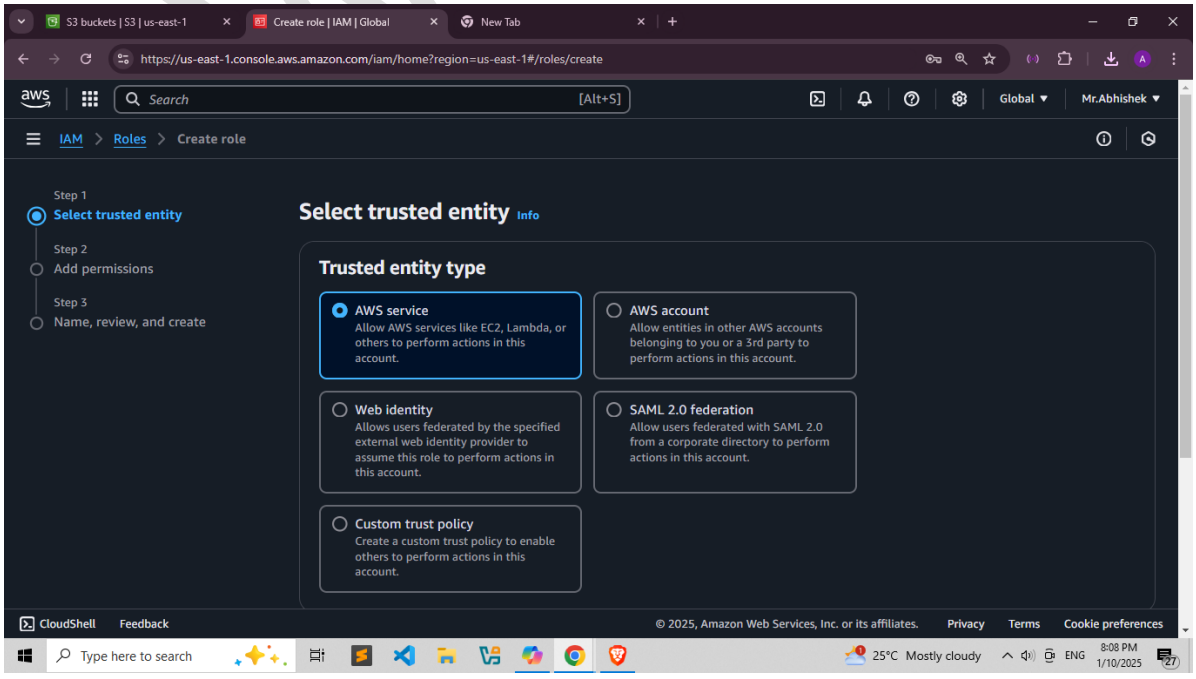
Step 1: Create the Role

1. **Sign in to the AWS Management Console**.
2. **Navigate to the IAM Console**: In the navigation pane, choose "Roles" and then "Create role."

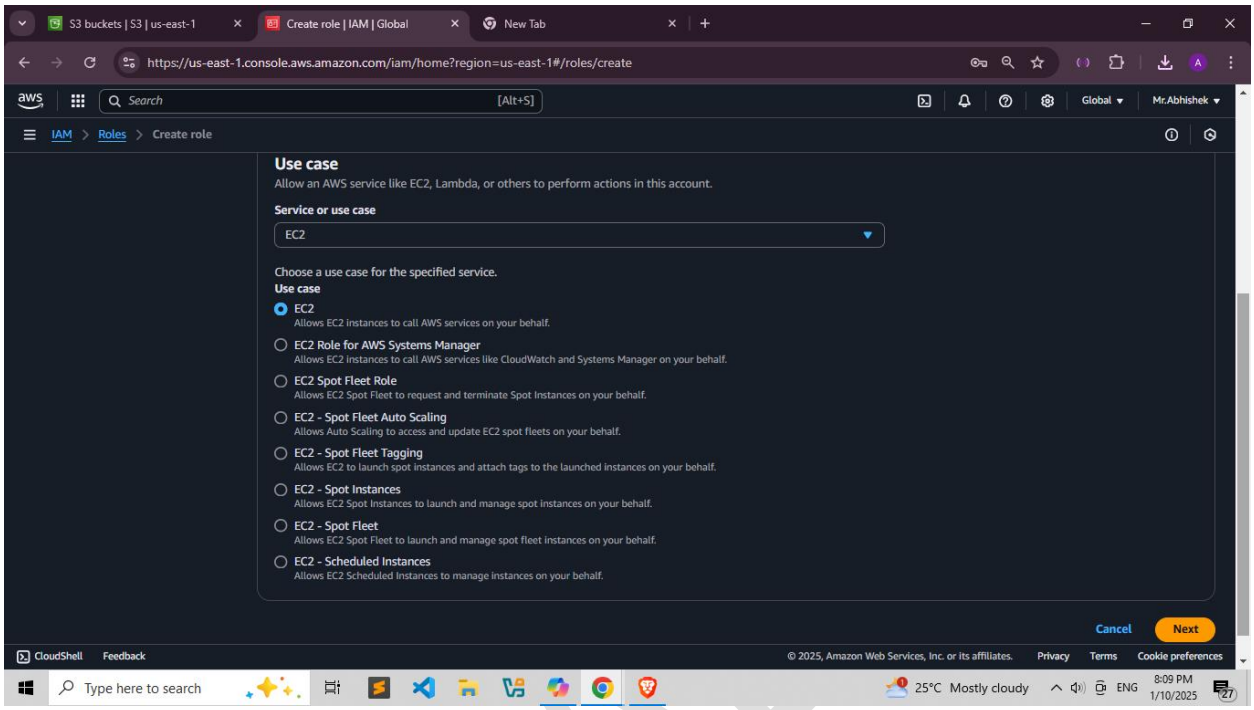


Select the Trusted Entity:

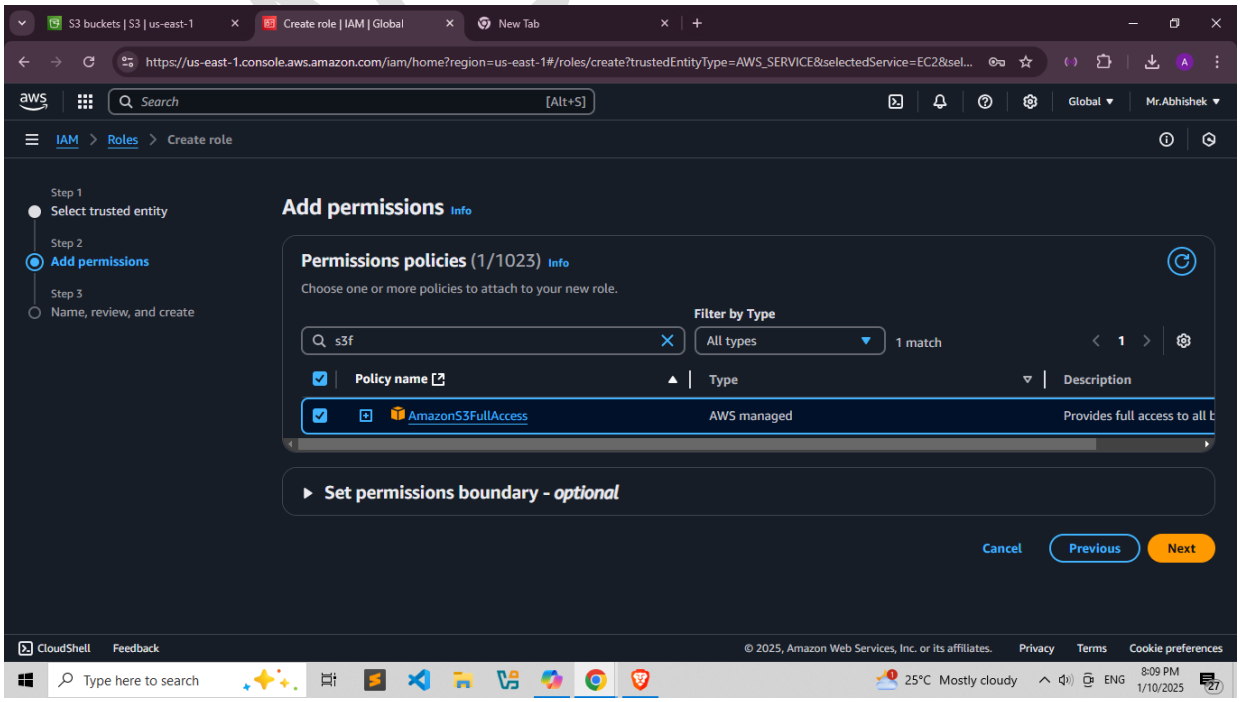
- Choose "AWS service."



- **Select "EC2" as the service that will use this role.**

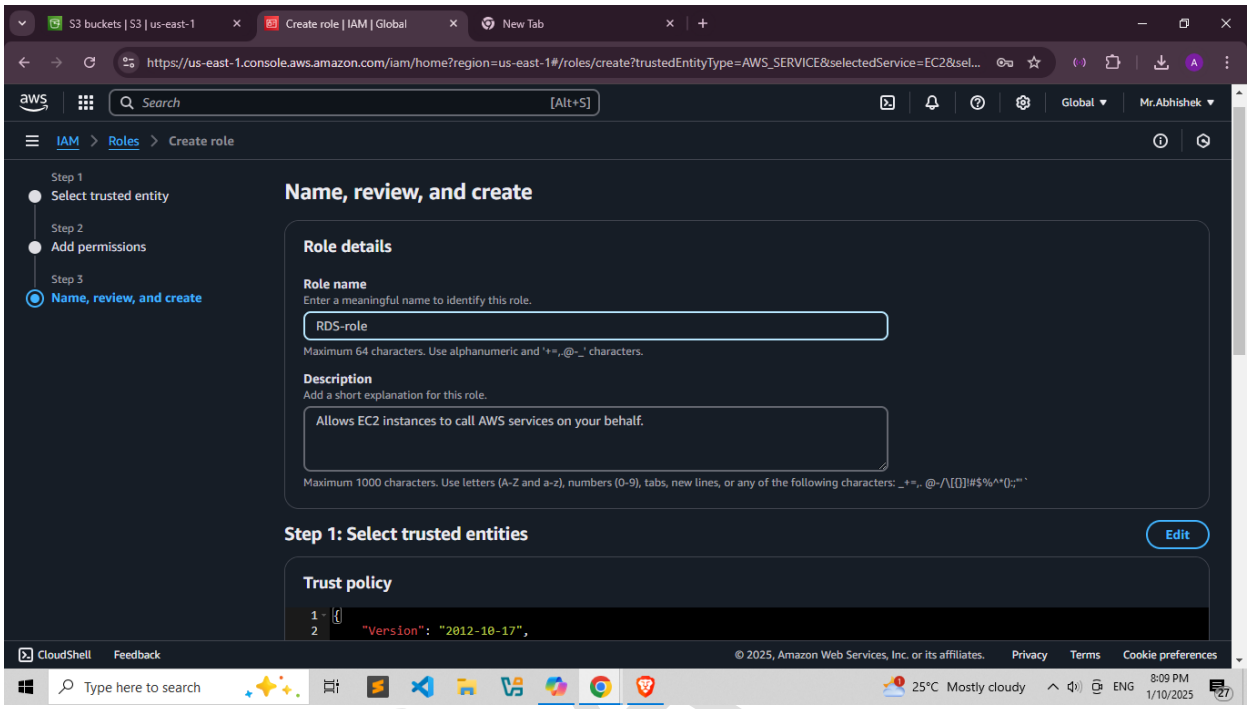


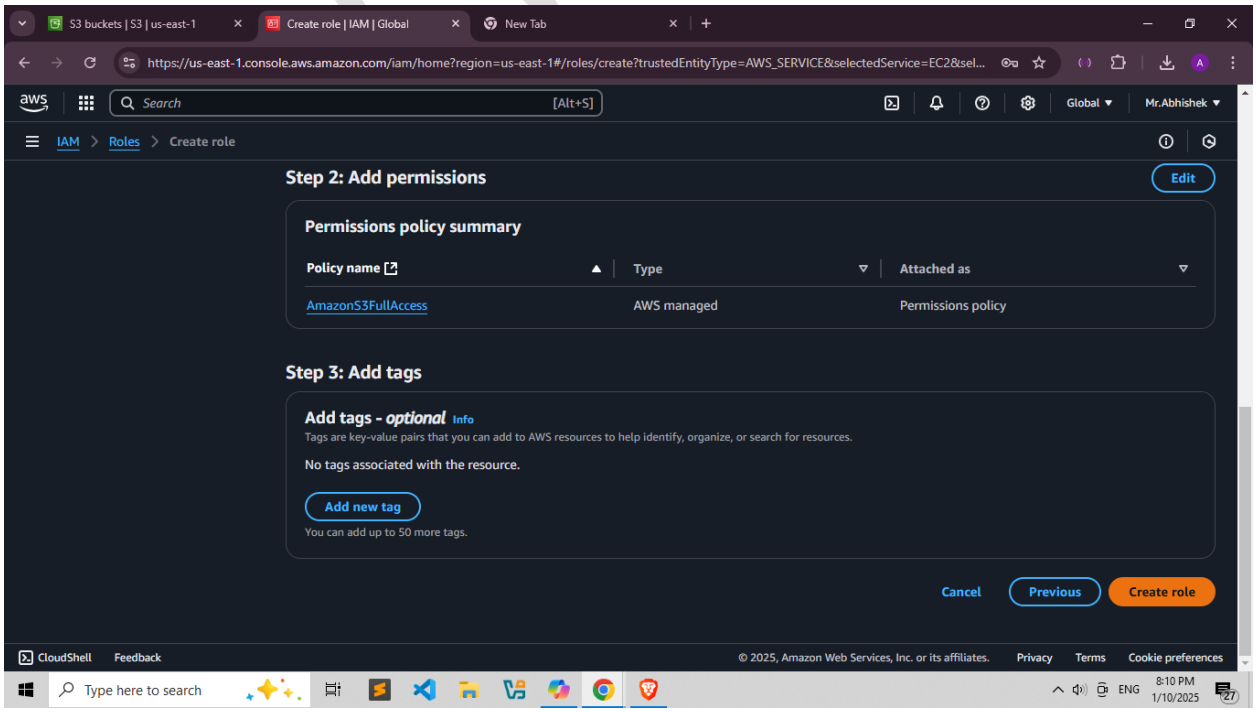
- **Click "Next: Permissions."**

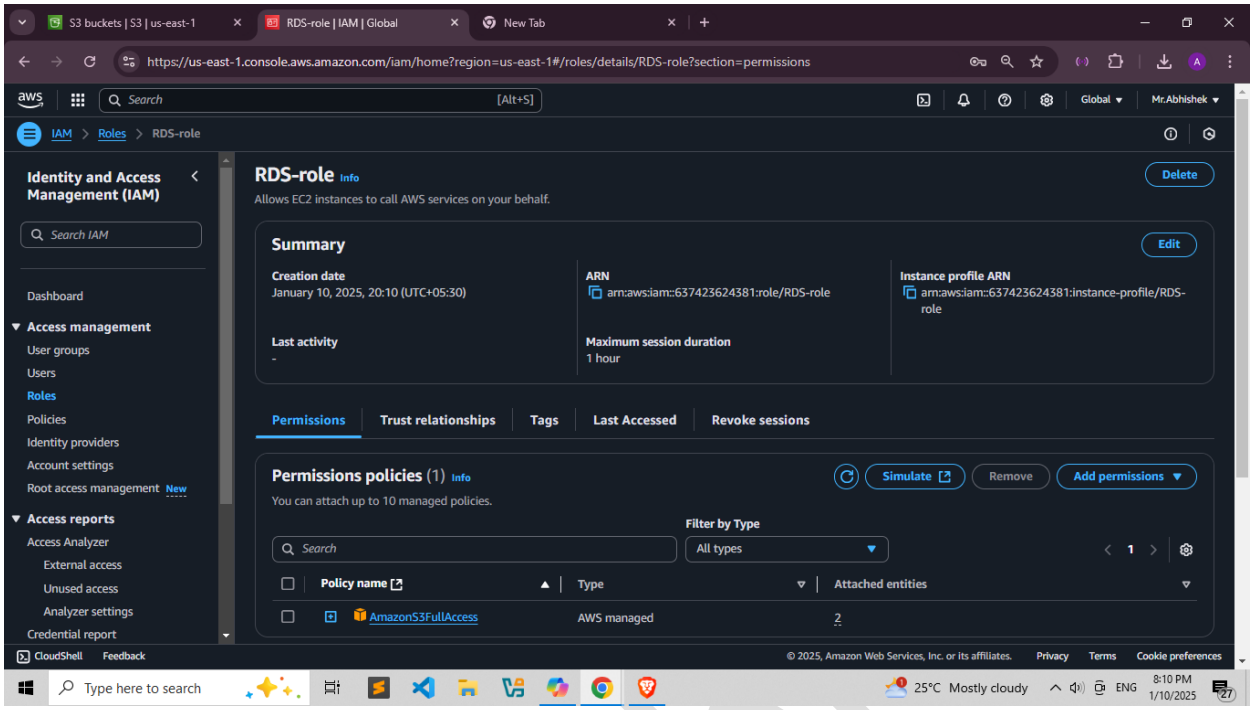


Add Tags (Optional):

- You can add tags to your role to help you organize and manage your resources.
- Click "Next: Review."
- Click On Create Role Button.



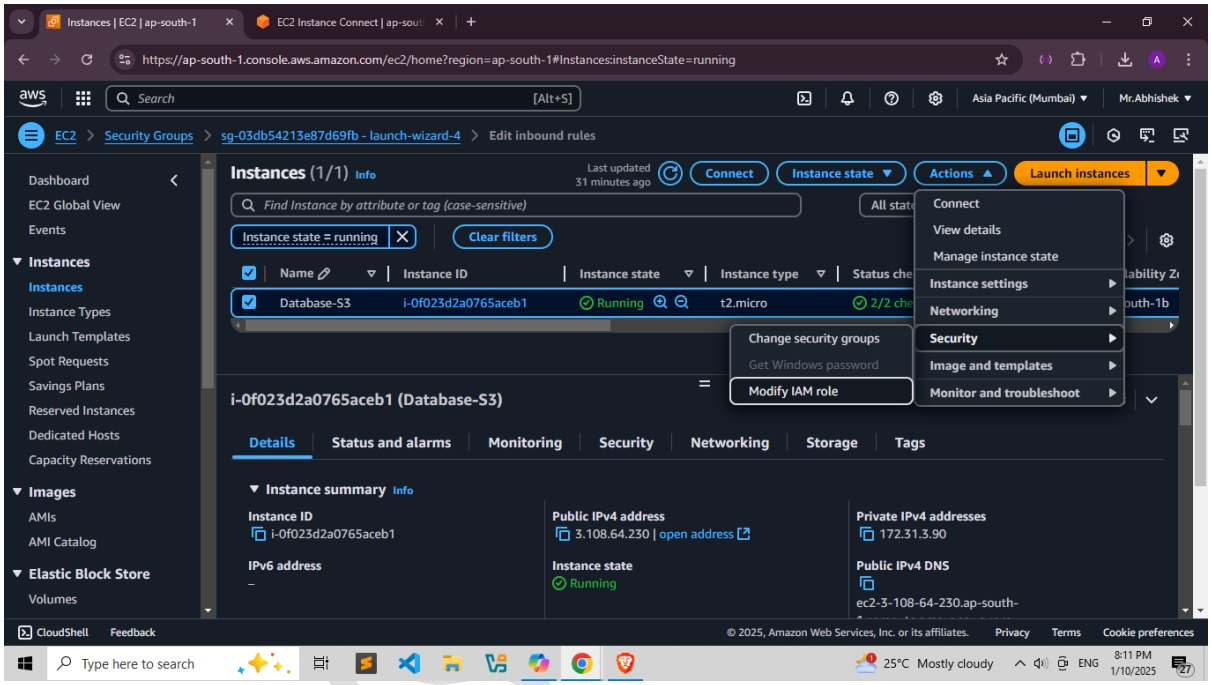


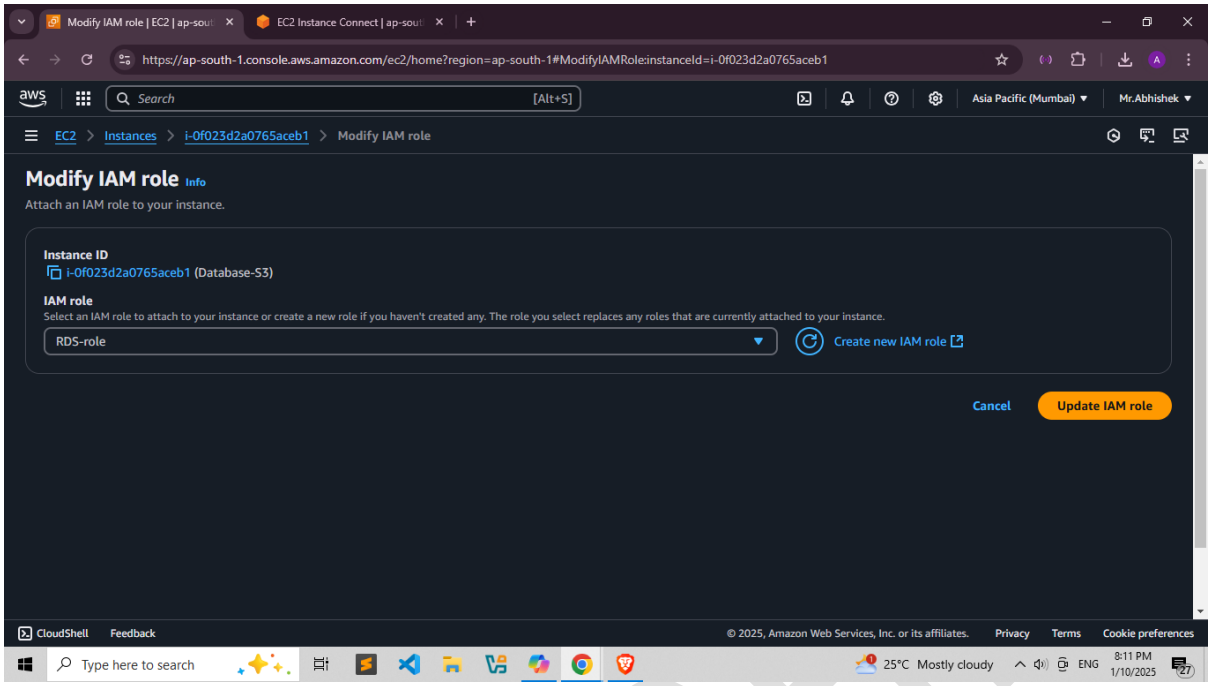

S3 buckets | S3 | us-east-1
RDS-role | IAM | Global
New Tab
https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/details/RDS-role?section=permissions
Search
[Alt+S]
Global
Mr.Abhishek
IAM > Roles > RDS-role
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
RDS-role Info
Allows EC2 instances to call AWS services on your behalf.
Delete
Summary
Edit
Creation date
January 10, 2025, 20:10 (UTC+05:30)
ARN
arn:aws:iam::637423624381:role/RDS-role
Instance profile ARN
arn:aws:iam::637423624381:instance-profile/RDS-role
Last activity
-
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (1) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached entities
AmazonS3FullAccess
AWS managed
2
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
25°C Mostly cloudy
ENG
8:10 PM
1/10/2025

Step 3: Attach the Role to an EC2 Instance

Attach IAM Role:

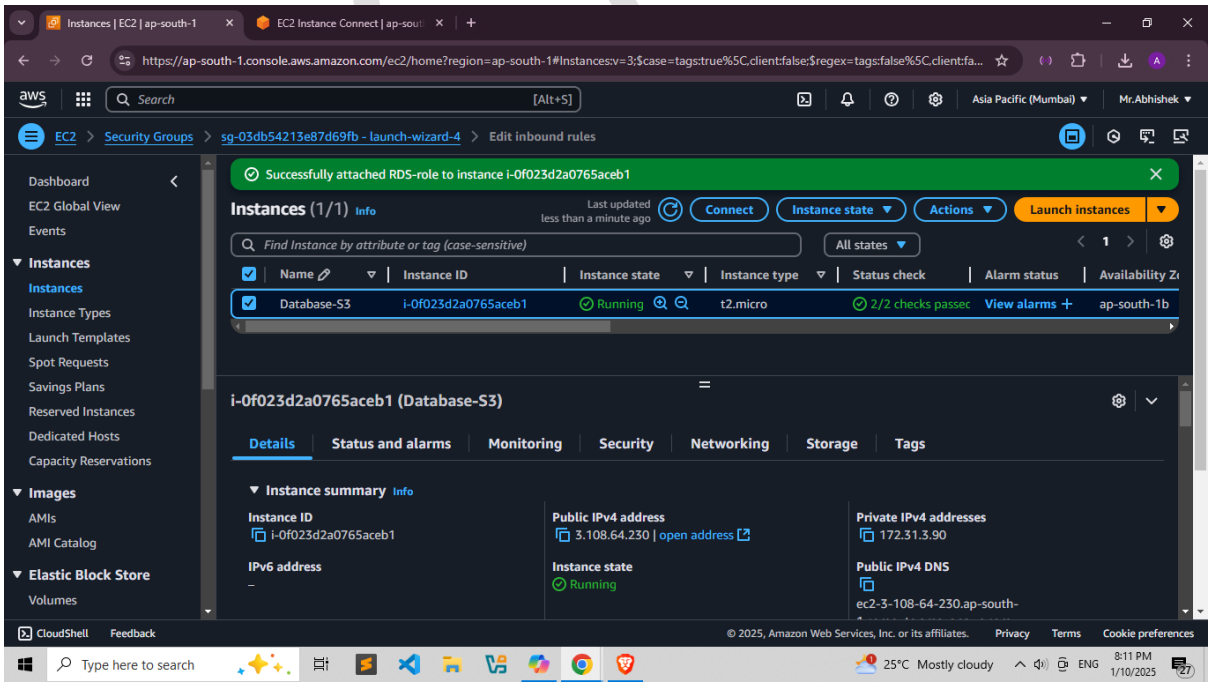
1. Choose "Actions" > "Security" > "Modify IAM Role."
2. In the "IAM role" drop-down menu, select the role you created (EC2_S3_FullAccess_Role).
3. Click "Update IAM role."





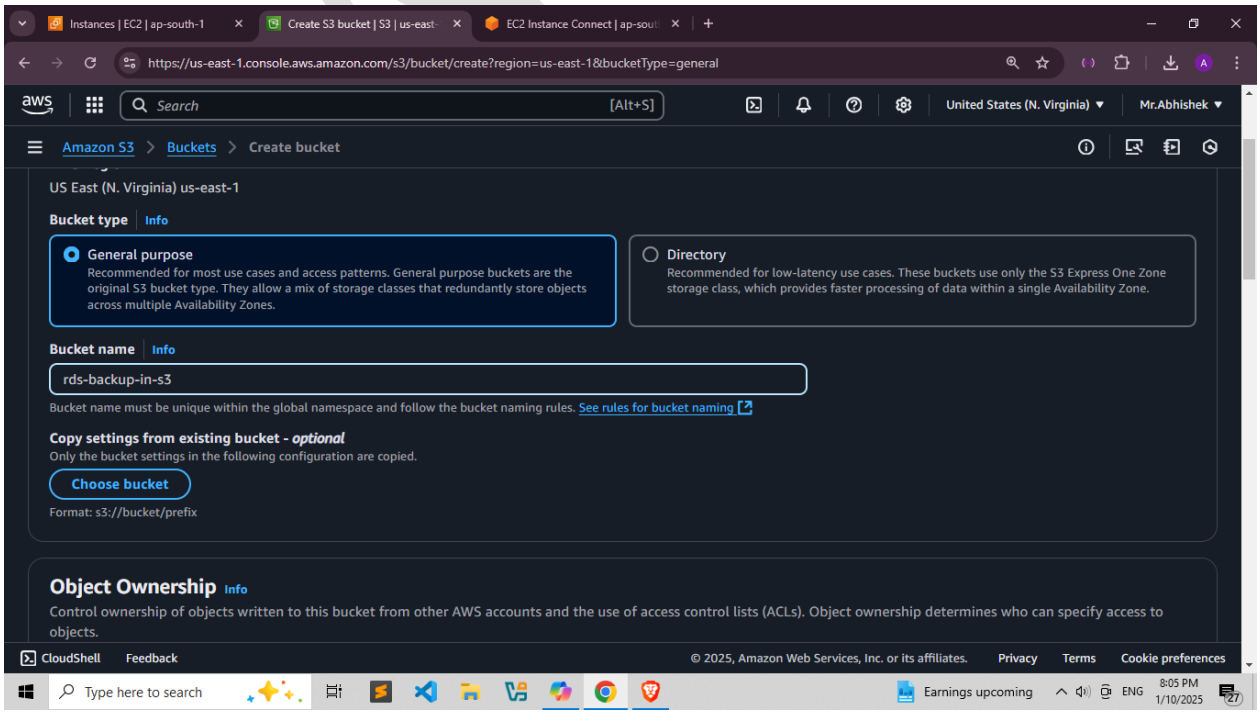
Step 4: Verify Role Attachment

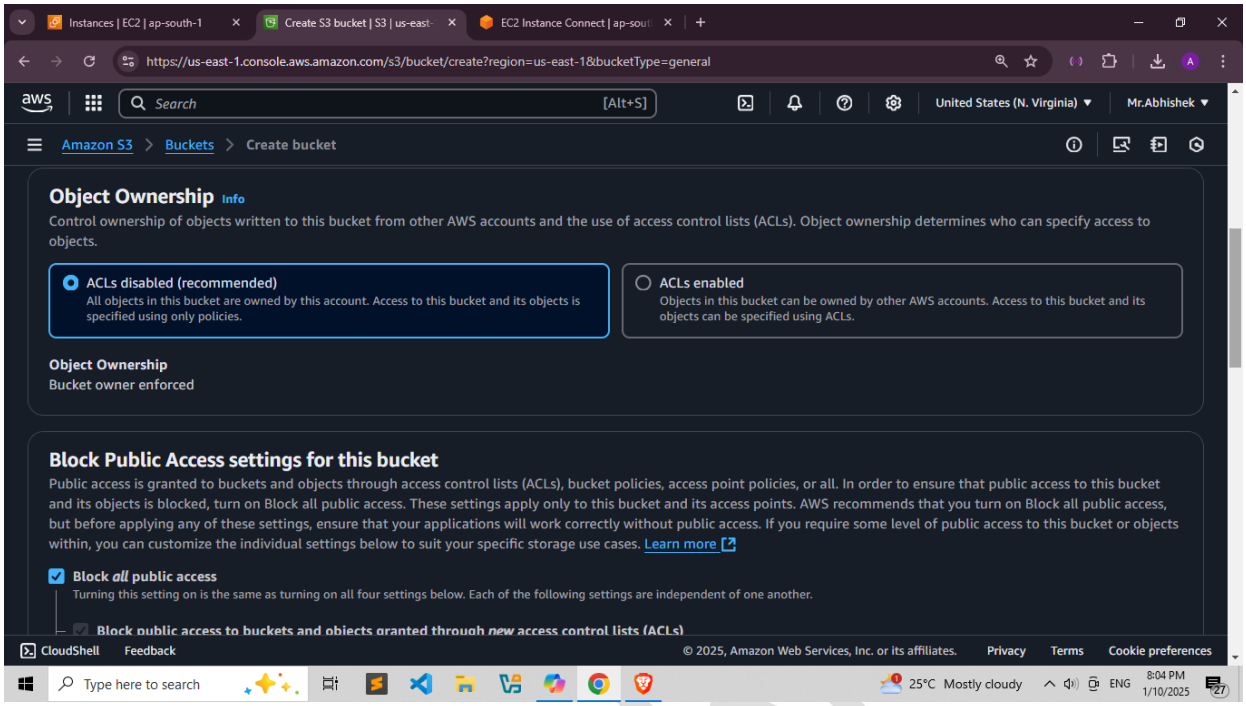
1. **Verify**: Make sure that the EC2 instance now has the appropriate role attached by checking the "IAM Role" field in the instance details.

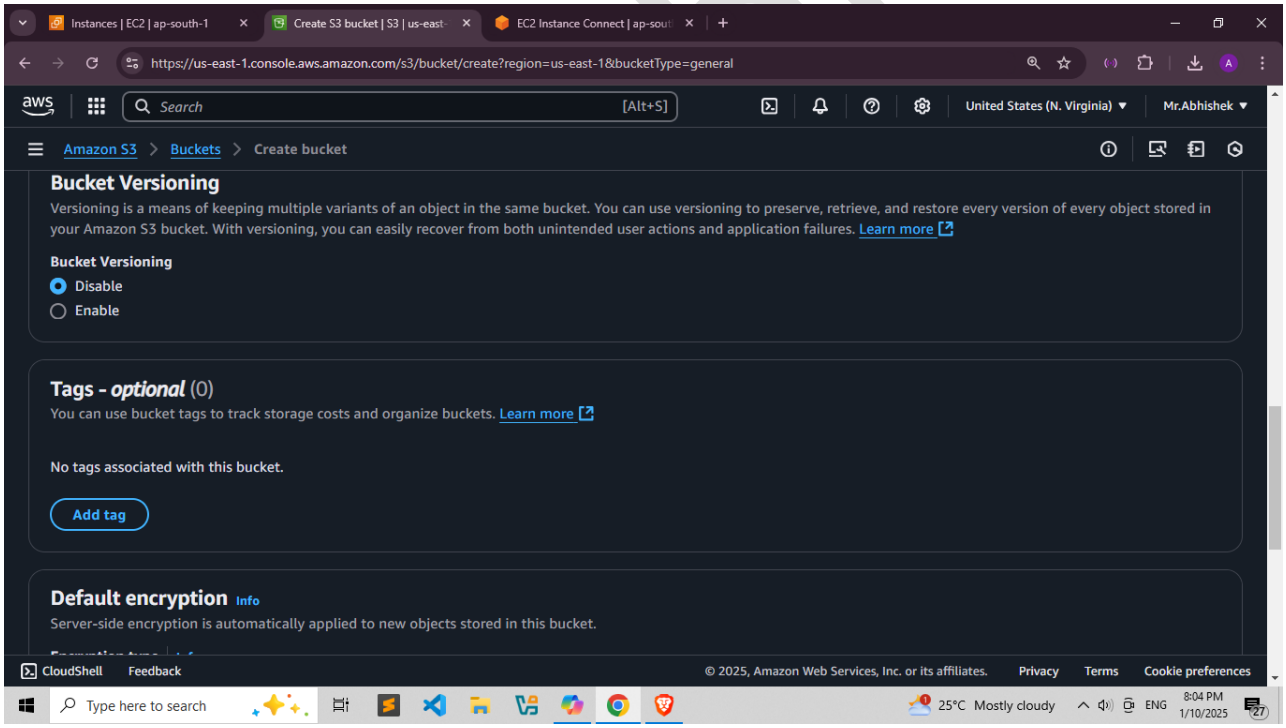


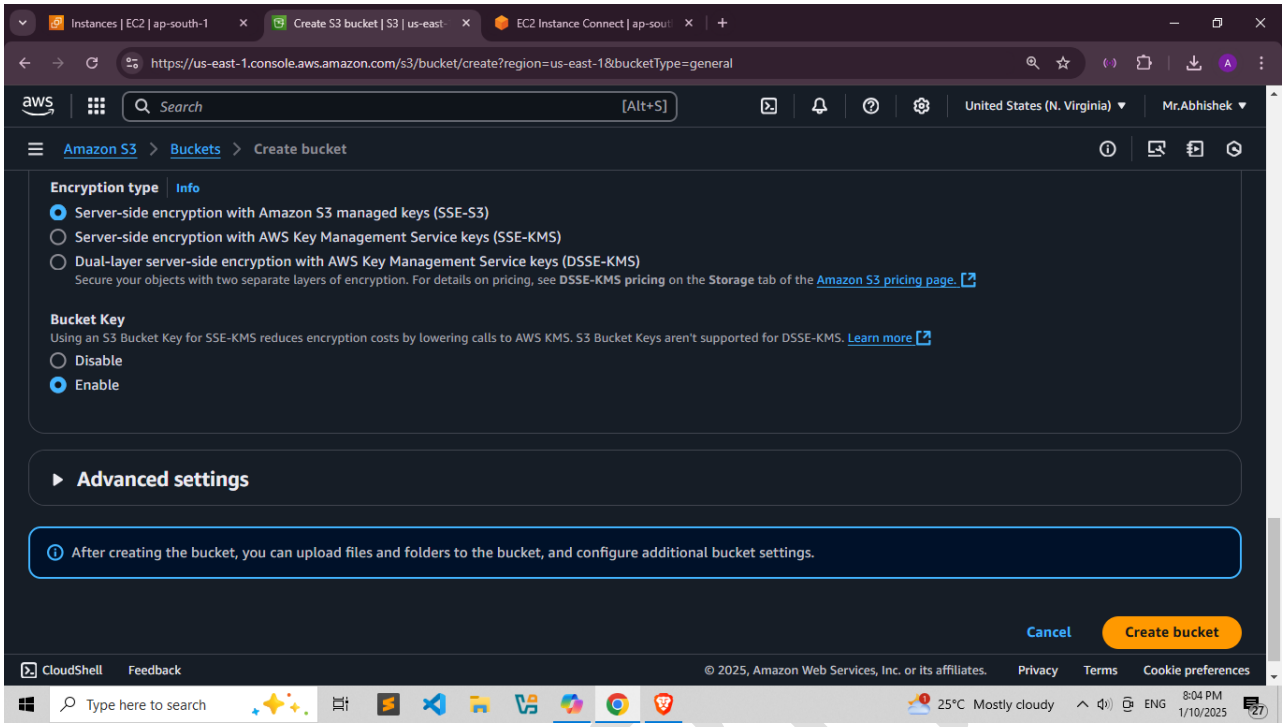
Step 1: Create a Bucket

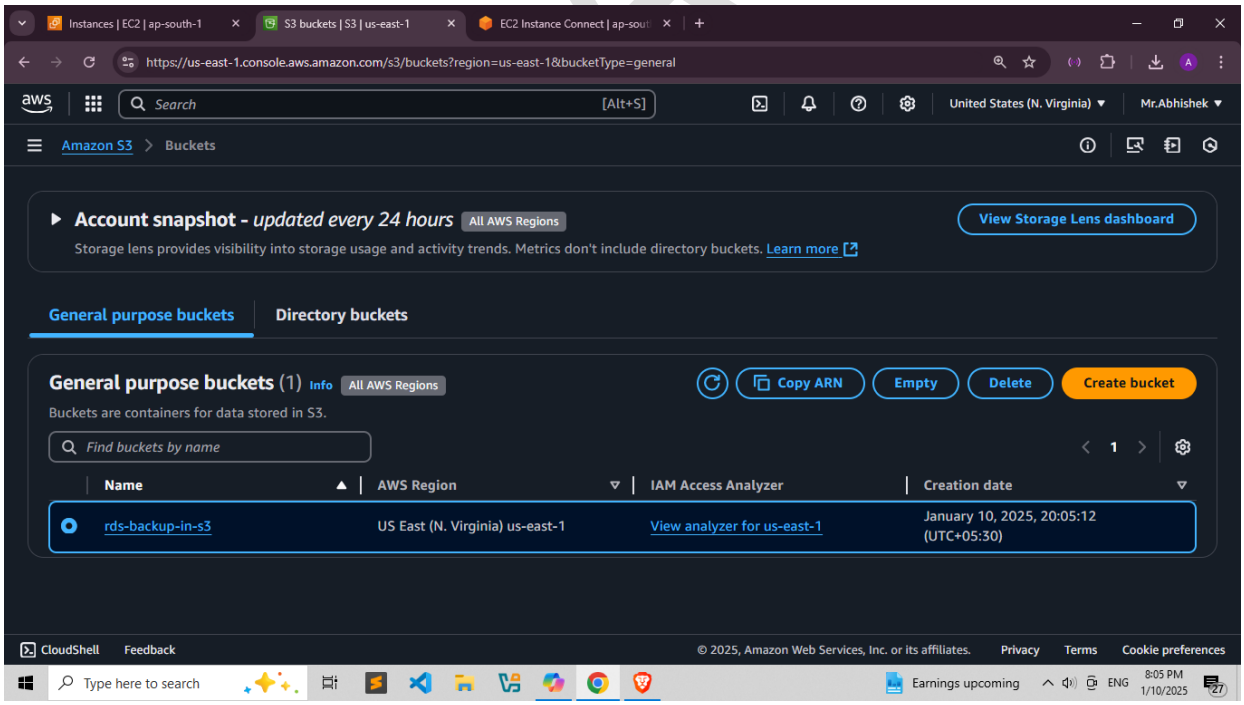
1. **Click on "Create bucket"**.
2. **Configure the Bucket**:
    - **Bucket name**: Enter a unique name for your bucket. Bucket names must be globally unique and must comply with certain naming conventions.
    - **Region**: Choose the AWS region where you want the bucket to be created.




Instances | EC2 | ap-south-1
Create S3 bucket | S3 | us-east-
EC2 Instance Connect | ap-sout
https://us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
Search
[Alt+S]
United States (N. Virginia)
Mr.Abhishek
Amazon S3 > Buckets > Create bucket
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
Object Ownership
Bucket owner enforced
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
25°C Mostly cloudy
ENG
8:04 PM
1/10/2025

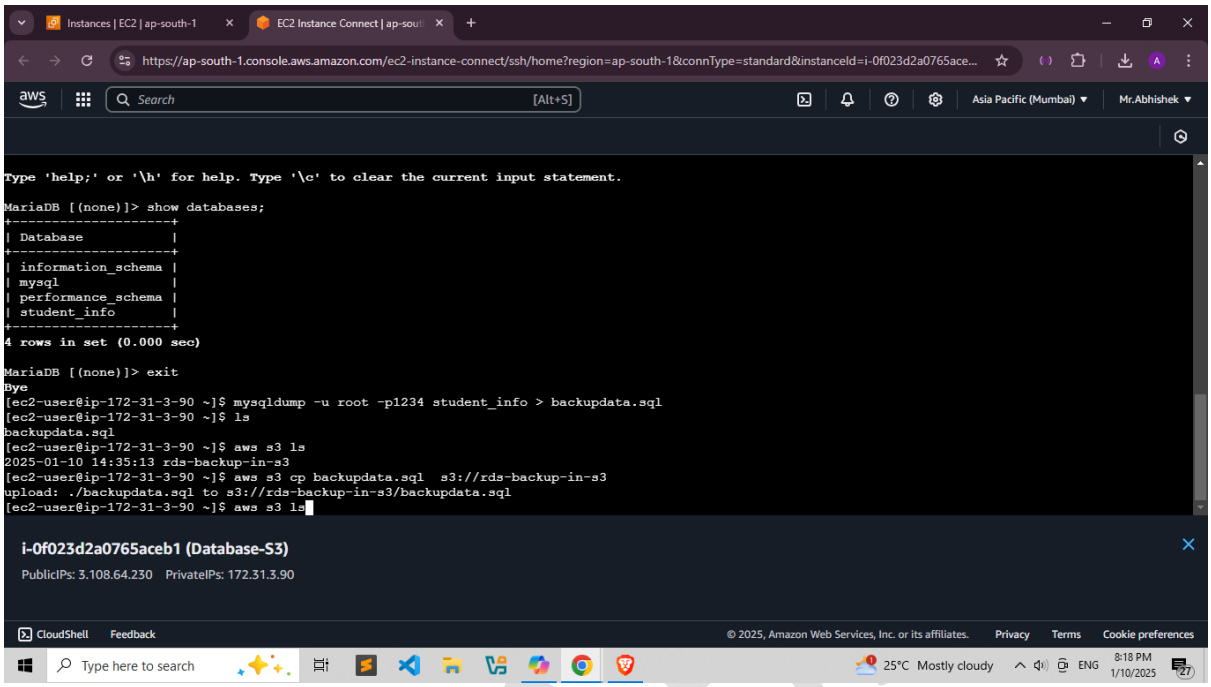

Instances | EC2 | ap-south-1
Create S3 bucket | S3 | us-east-
EC2 Instance Connect | ap-sout
https://us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
Search
[Alt+S]
United States (N. Virginia)
Mr.Abhishek
Amazon S3 > Buckets > Create bucket
Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable
Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
25°C Mostly cloudy
ENG
8:04 PM
1/10/2025



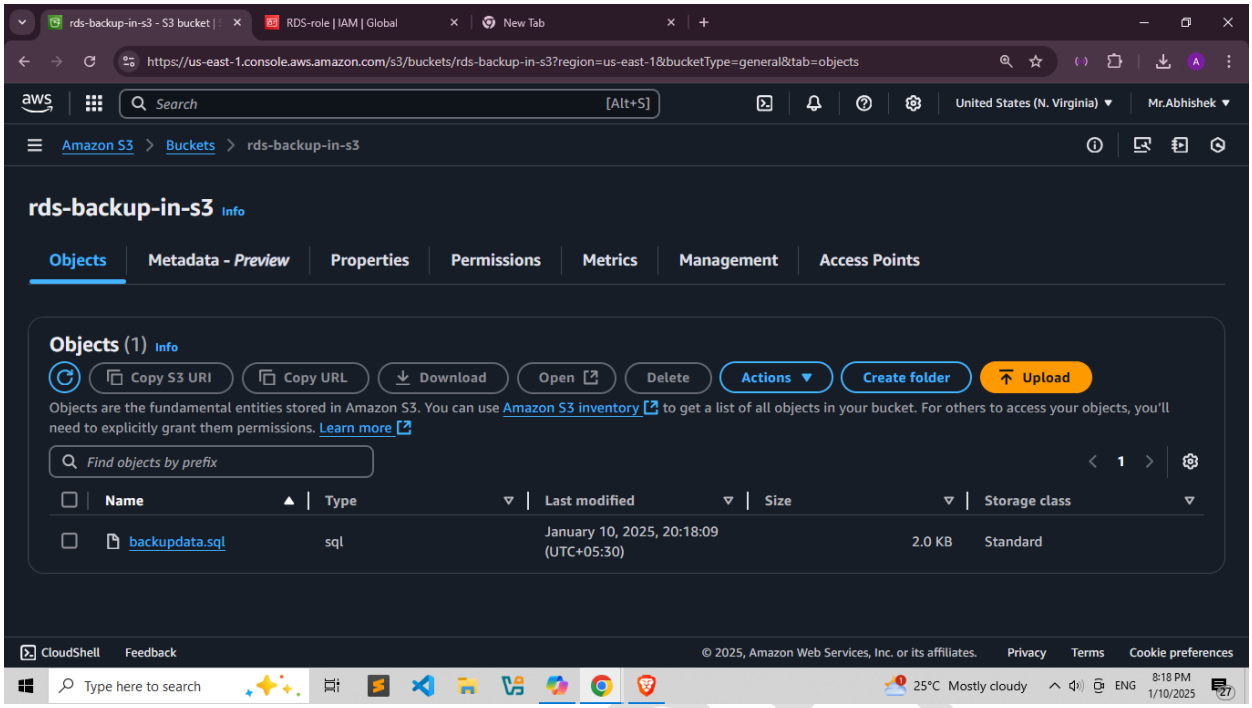


Copy this backup file to the S3 Bucket using:

- **aws s3 cp filename.sql s3://bucket_name**



Bucket View: Store backup file in Bucket through CLI.


rds-backup-in-s3 - S3 bucket
RDS-role | IAM | Global
New Tab
https://us-east-1.console.aws.amazon.com/s3/buckets/rds-backup-in-s3?region=us-east-1&bucketType=general&tab=objects
Search
[Alt+S]
United States (N. Virginia)
Mr.Abhishek
Amazon S3 > Buckets > rds-backup-in-s3
rds-backup-in-s3 Info
Objects
Metadata - Preview
Properties
Permissions
Metrics
Management
Access Points
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
backupdata.sql
sql
January 10, 2025, 20:18:09 (UTC+05:30)
2.0 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
25°C Mostly cloudy
ENG
8:18 PM
1/10/2025