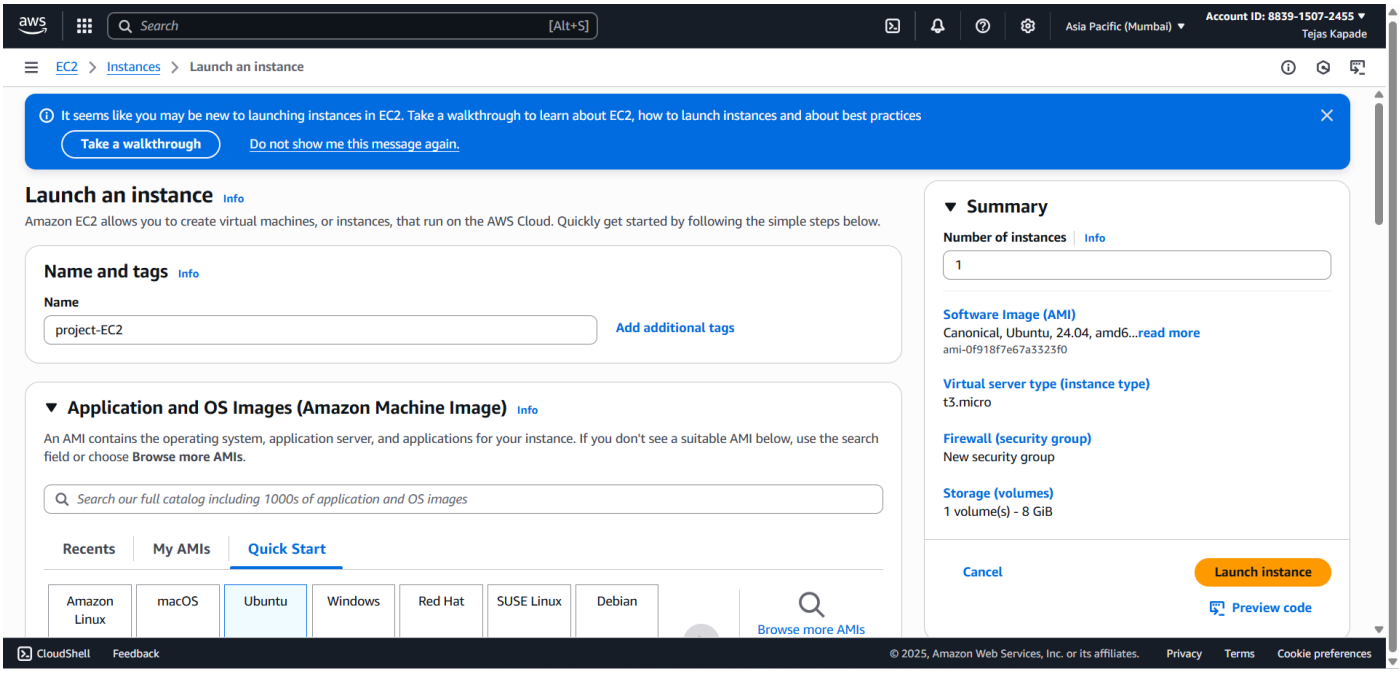
AWS, Project 01.

EC2- Based Web Server with S3 Backup

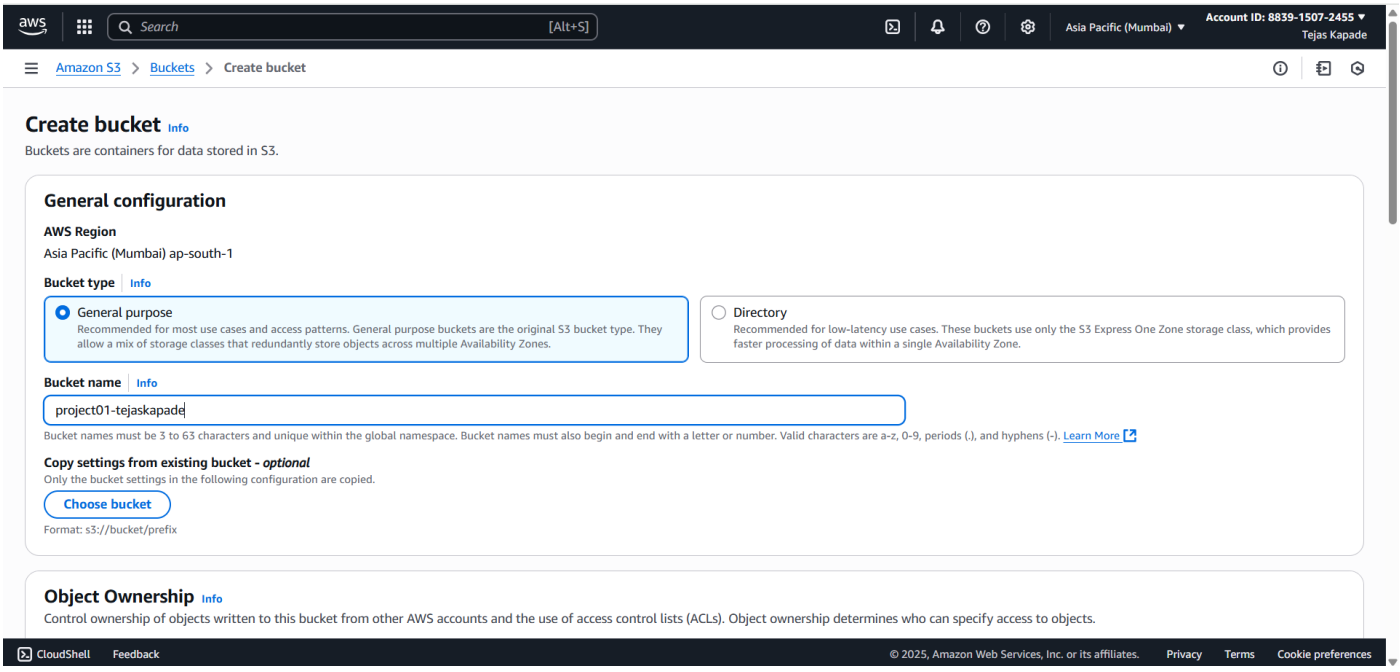
12-08-2025

Tejas Kapade.

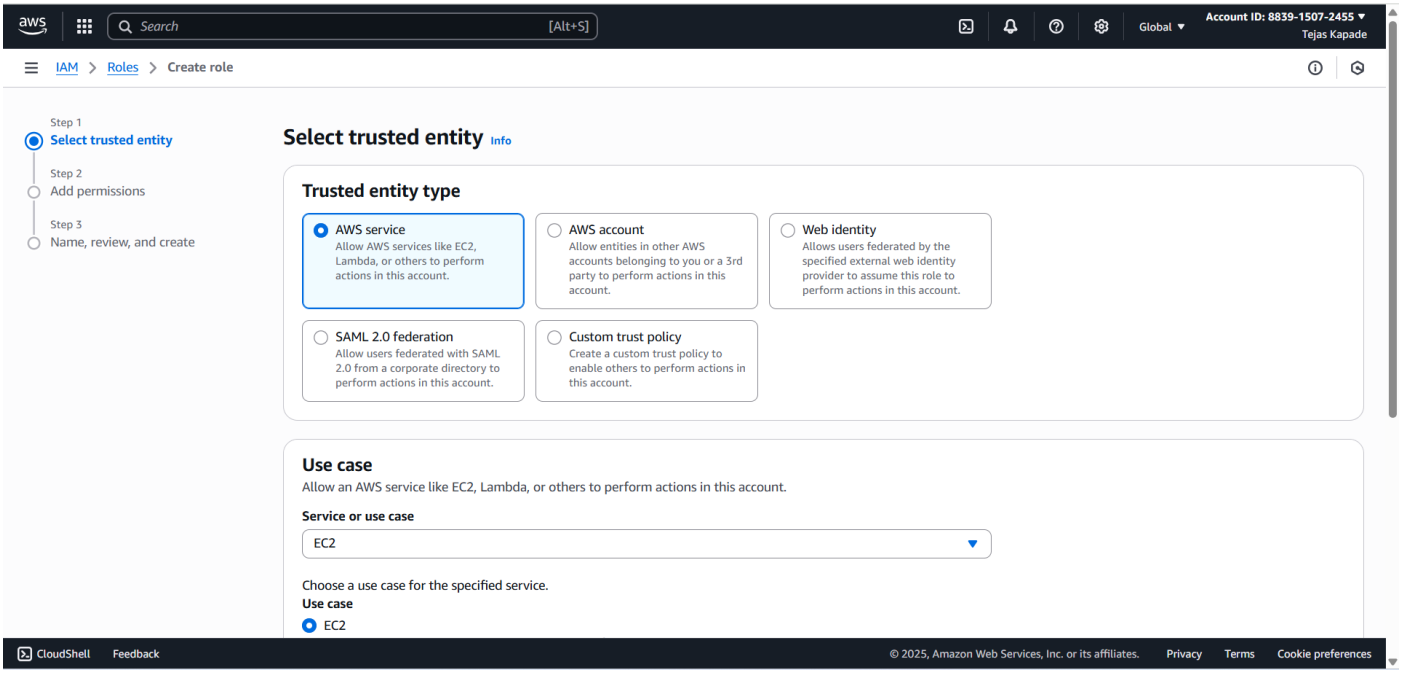
1. Launch EC2 Instance,

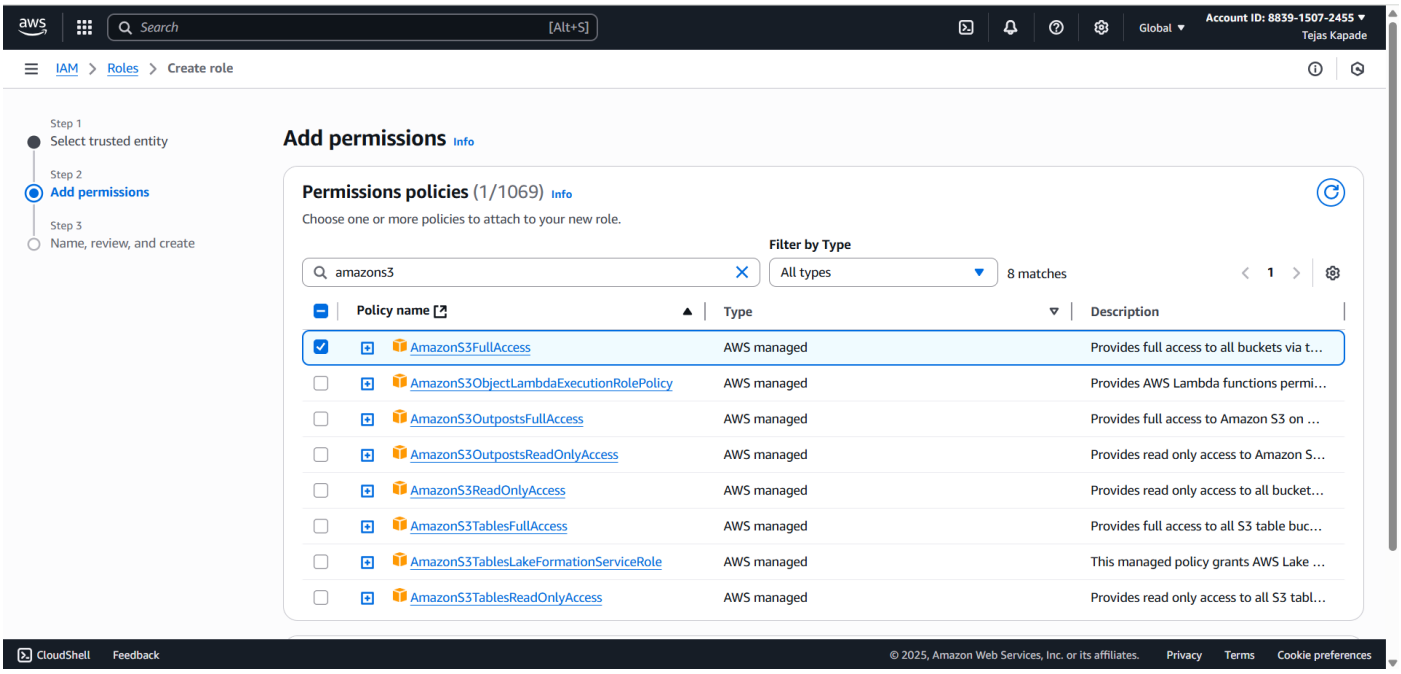


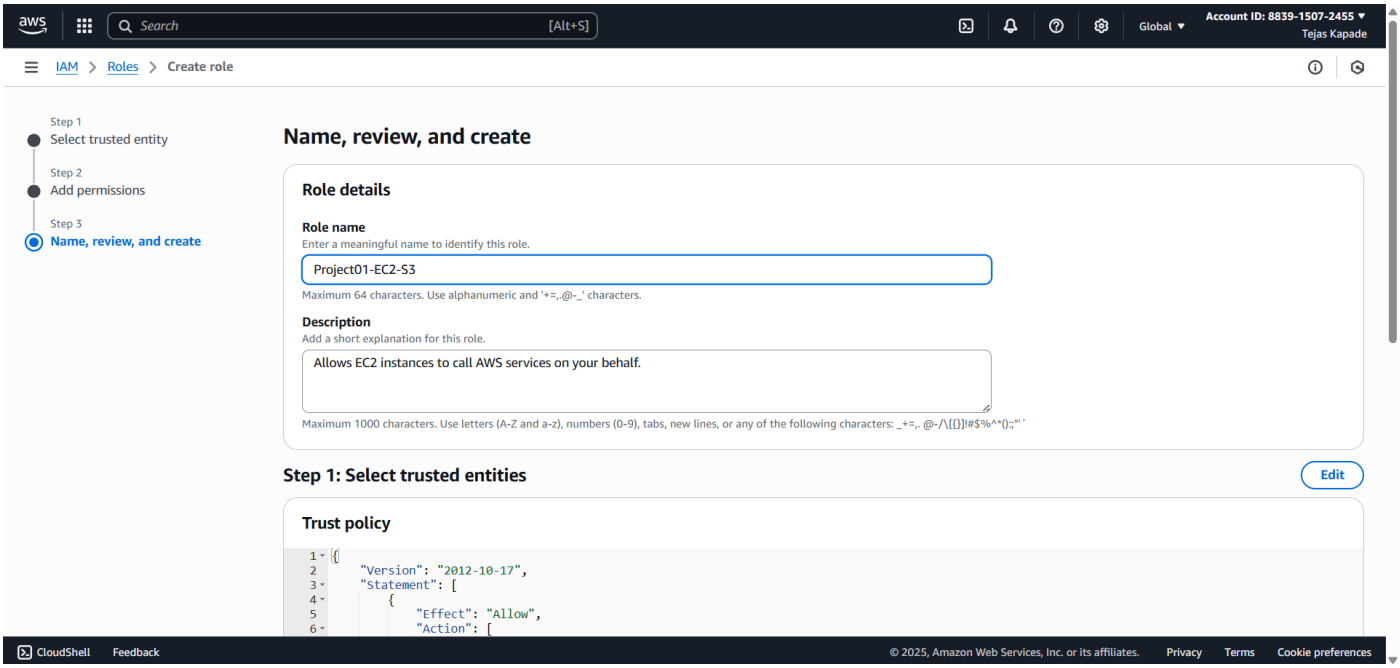
2. Create S3 Bucket,

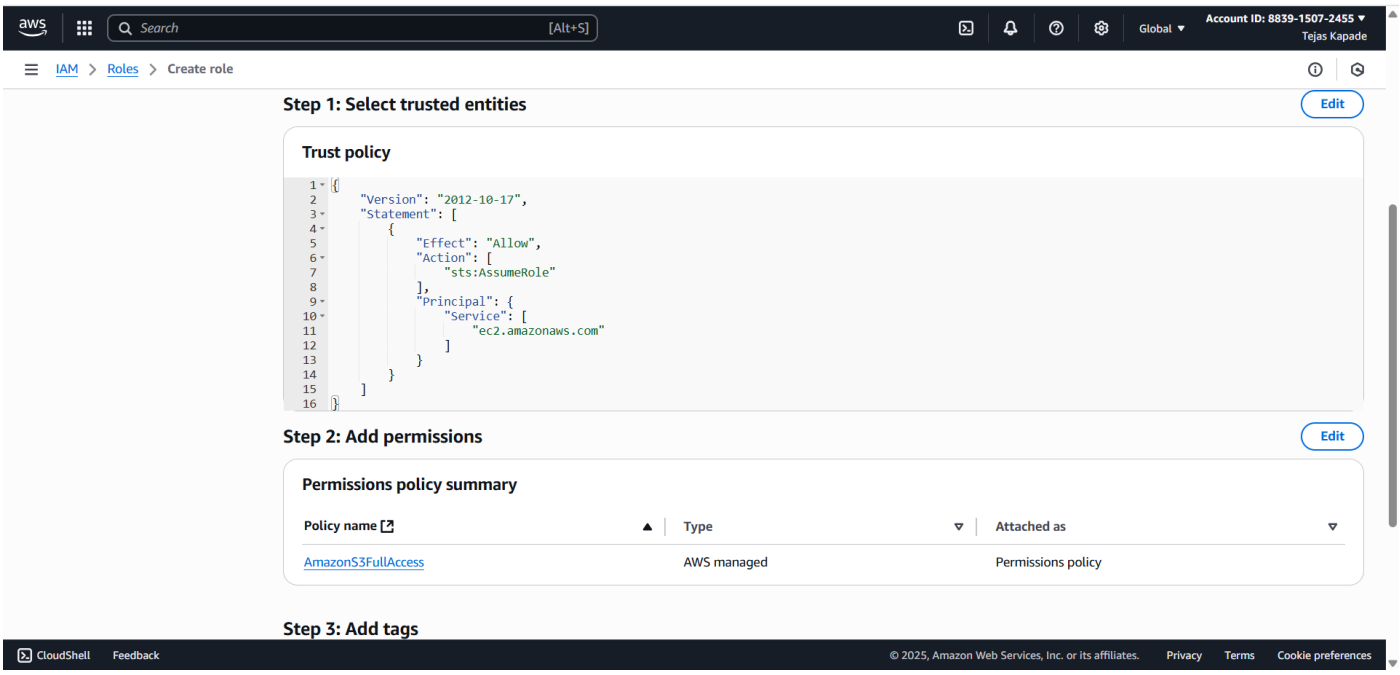


3. Create an IAM Role for EC2, (We need EC2 to access S3 without storing keys)

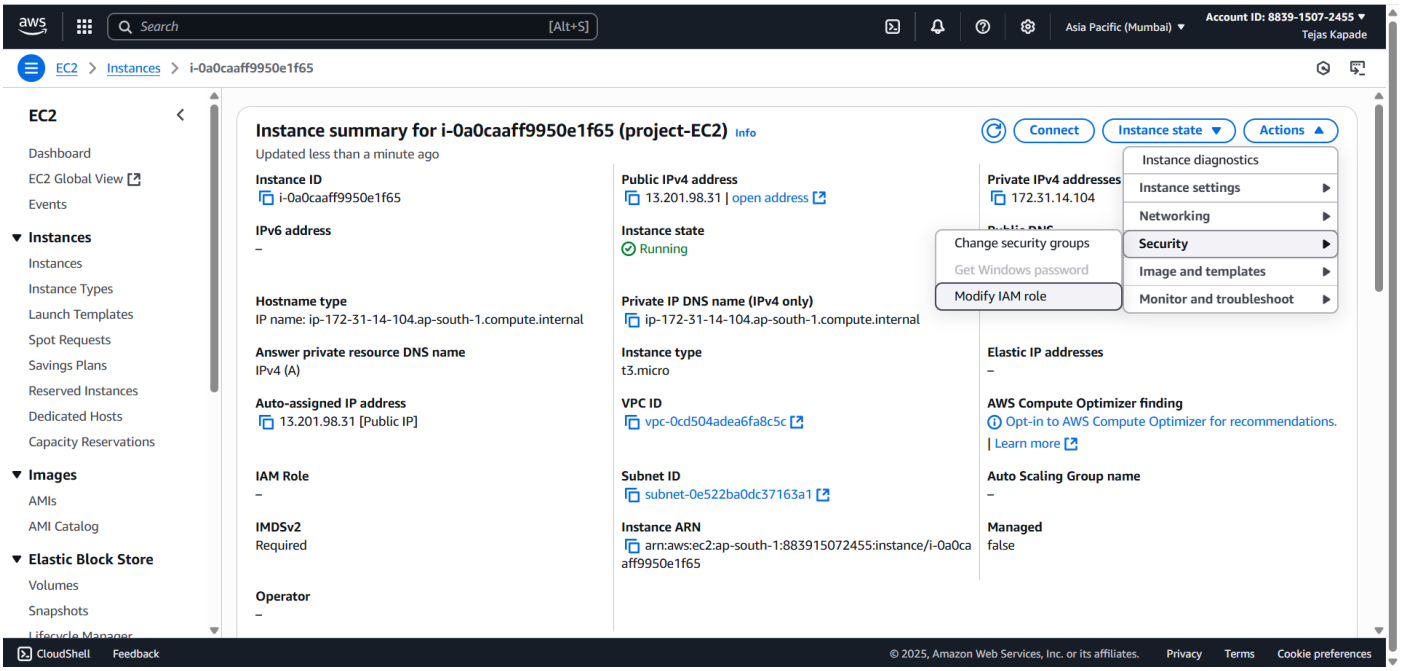


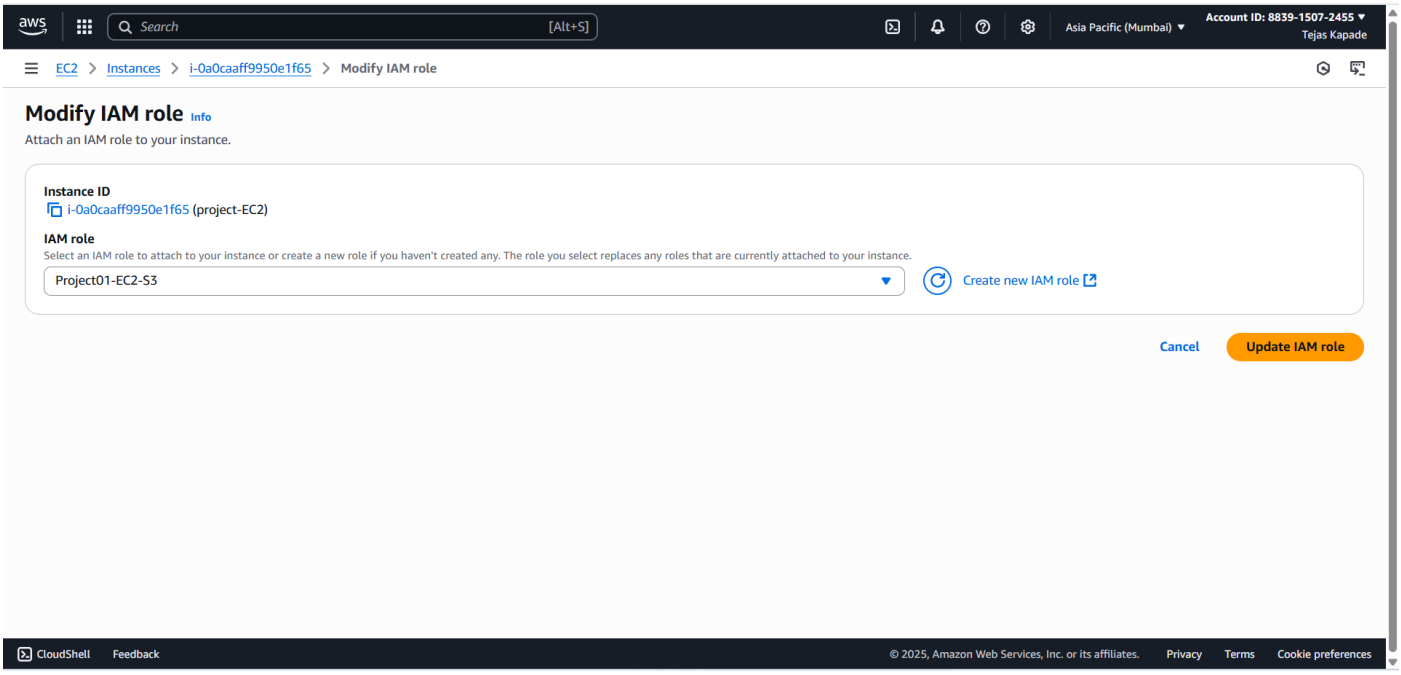


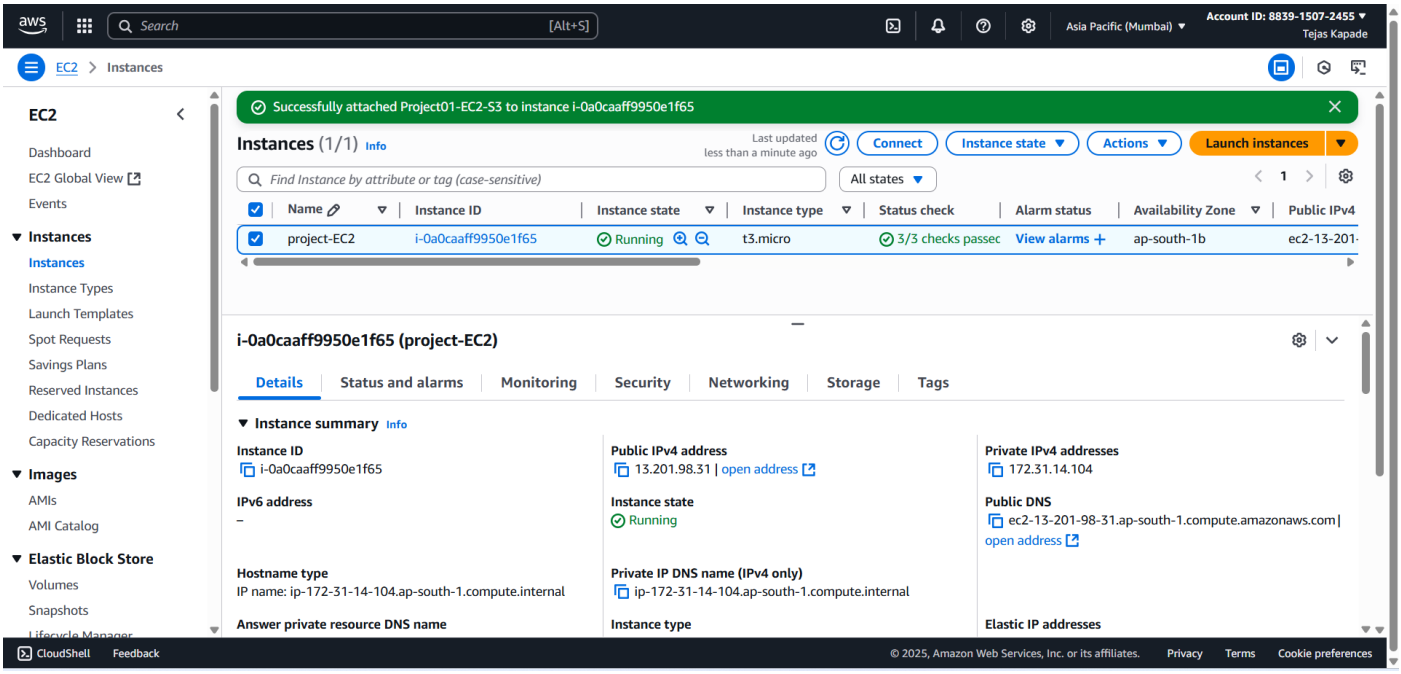

aws
Search
[Alt+S]
Global
Account ID: 8839-1507-2455
Tejas Kapade
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
Project01-EC2-S3
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'`
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Search
[Alt+S]
Global
Account ID: 8839-1507-2455
Tejas Kapade
IAM > Roles > Create role
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"sts:AssumeRole"
],
"Principal": {
"Service": [
"ec2.amazonaws.com"
]
}
}
]
}
Step 2: Add permissions
Edit
Permissions policy summary
Policy name
Type
Attached as
AmazonS3FullAccess
AWS managed
Permissions policy
Step 3: Add tags
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

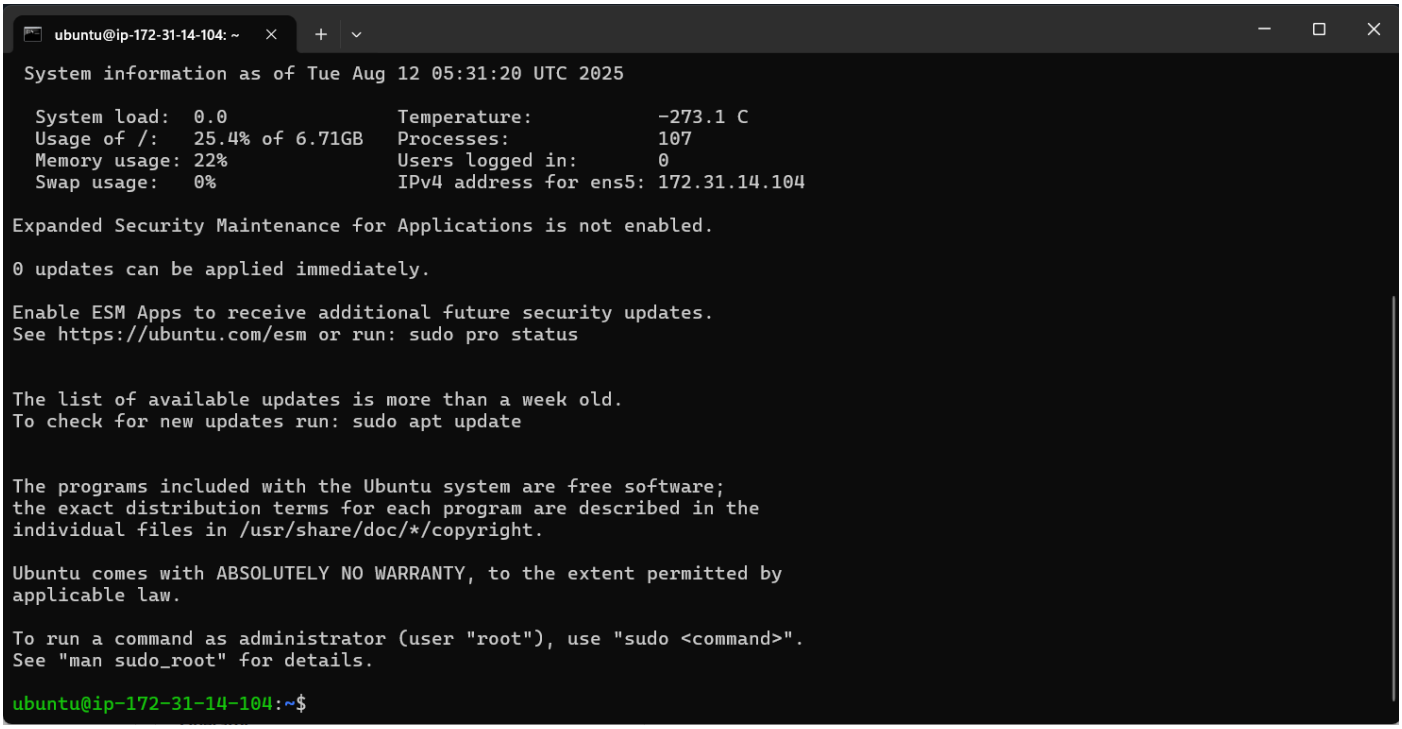
4. Attach IAM Role to EC2 Instance,

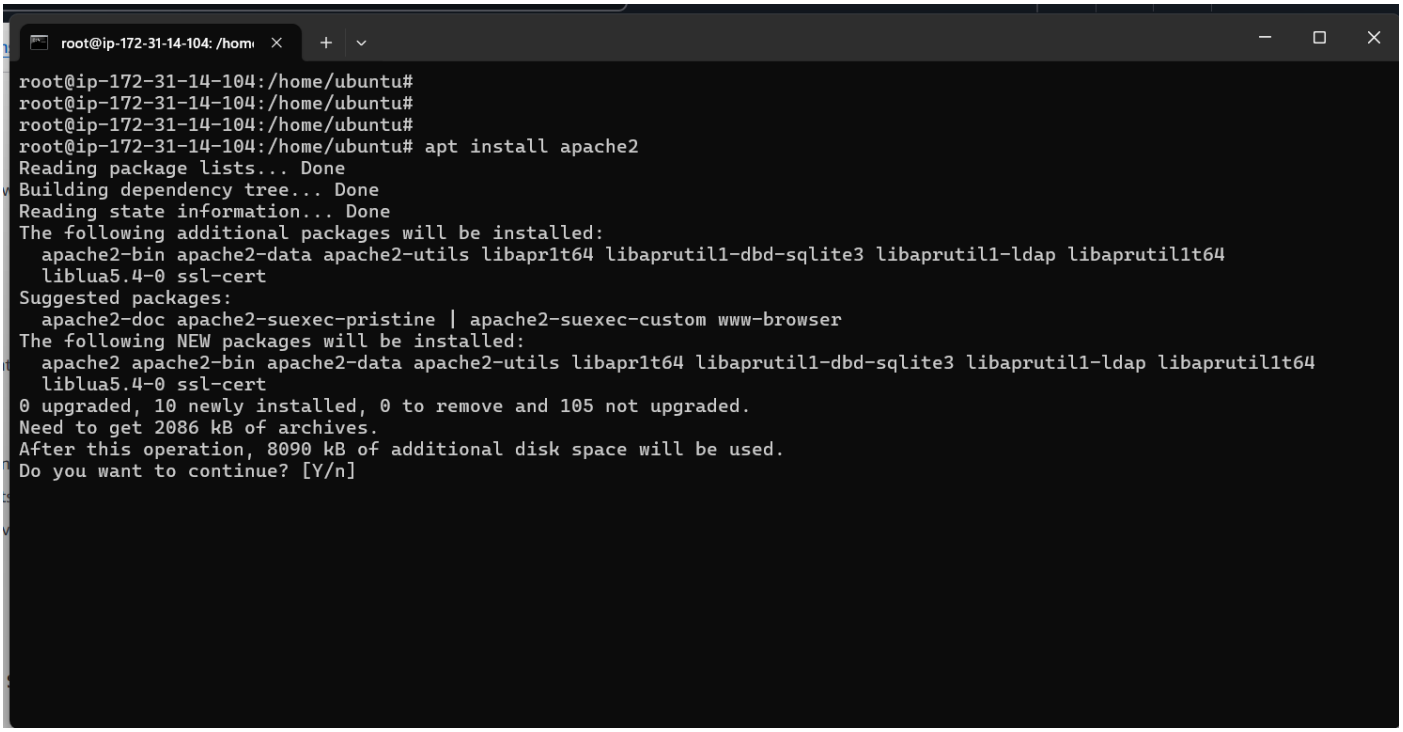




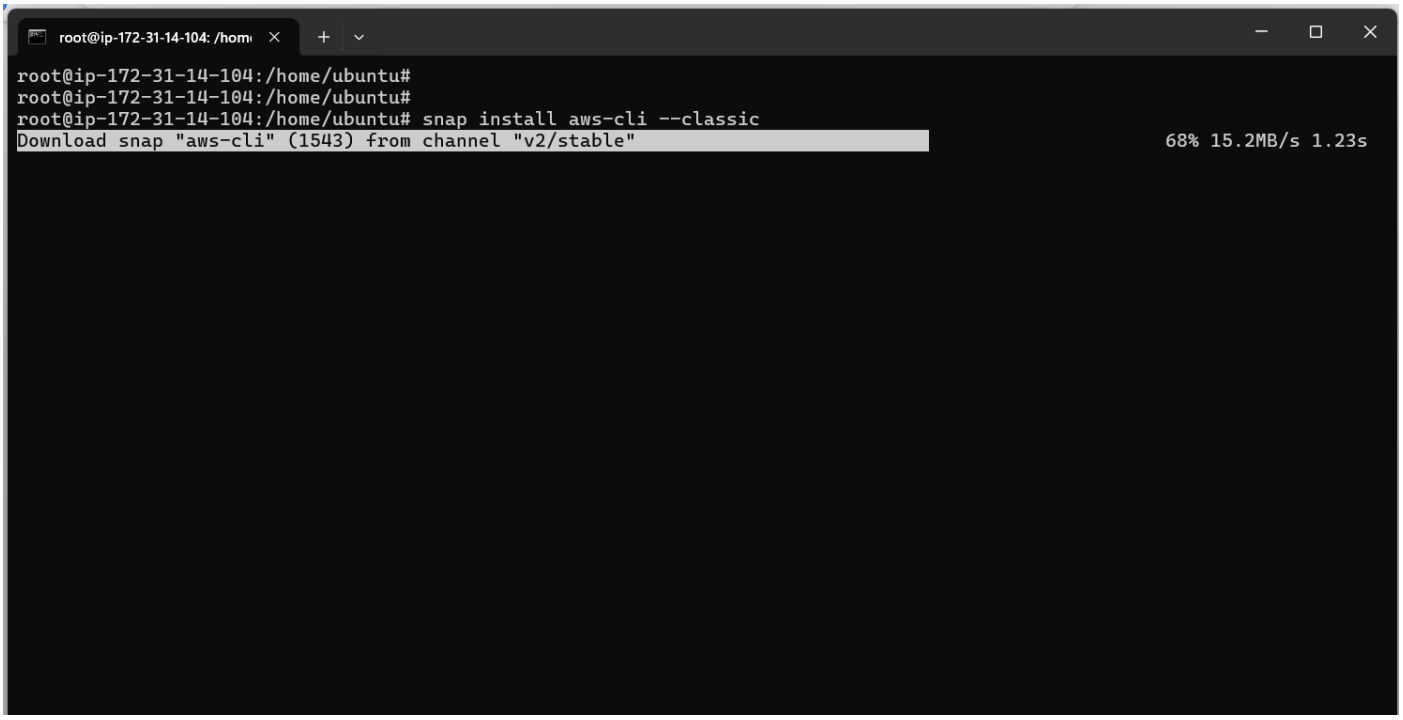


5. Install Apache on EC2,



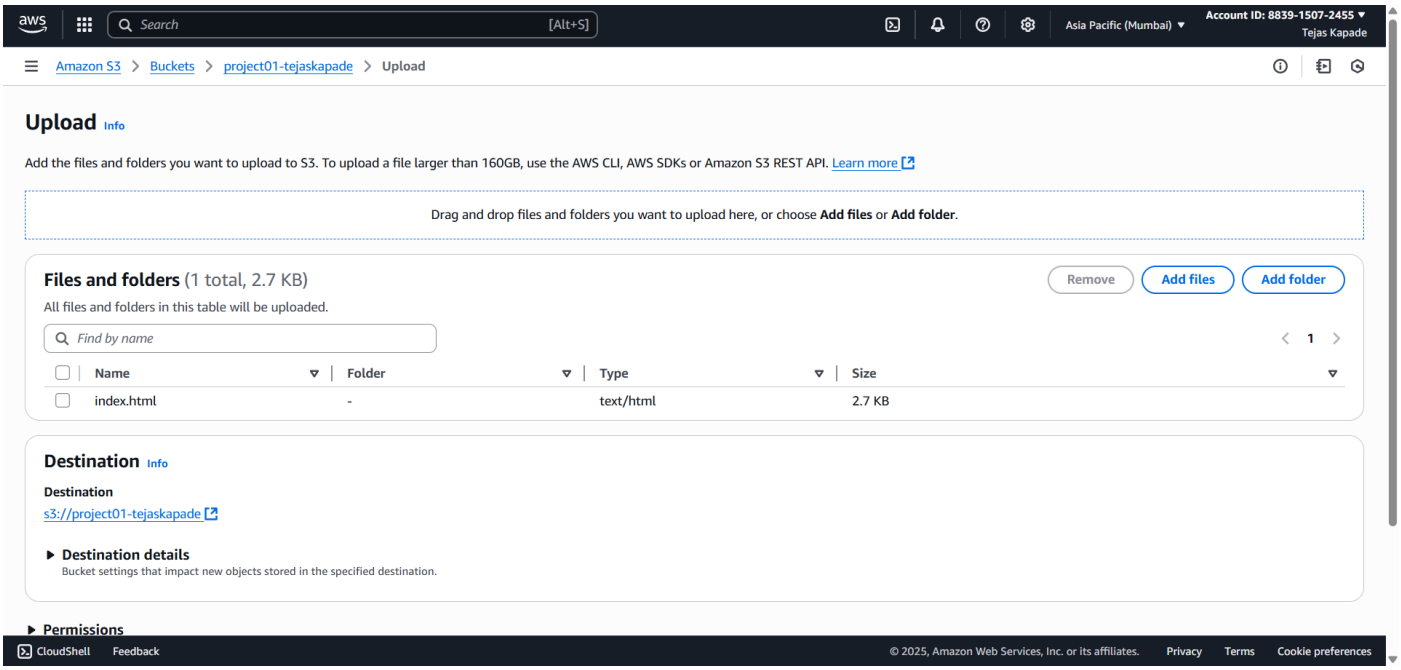


6. Install AWS CLI on EC2 and test it,

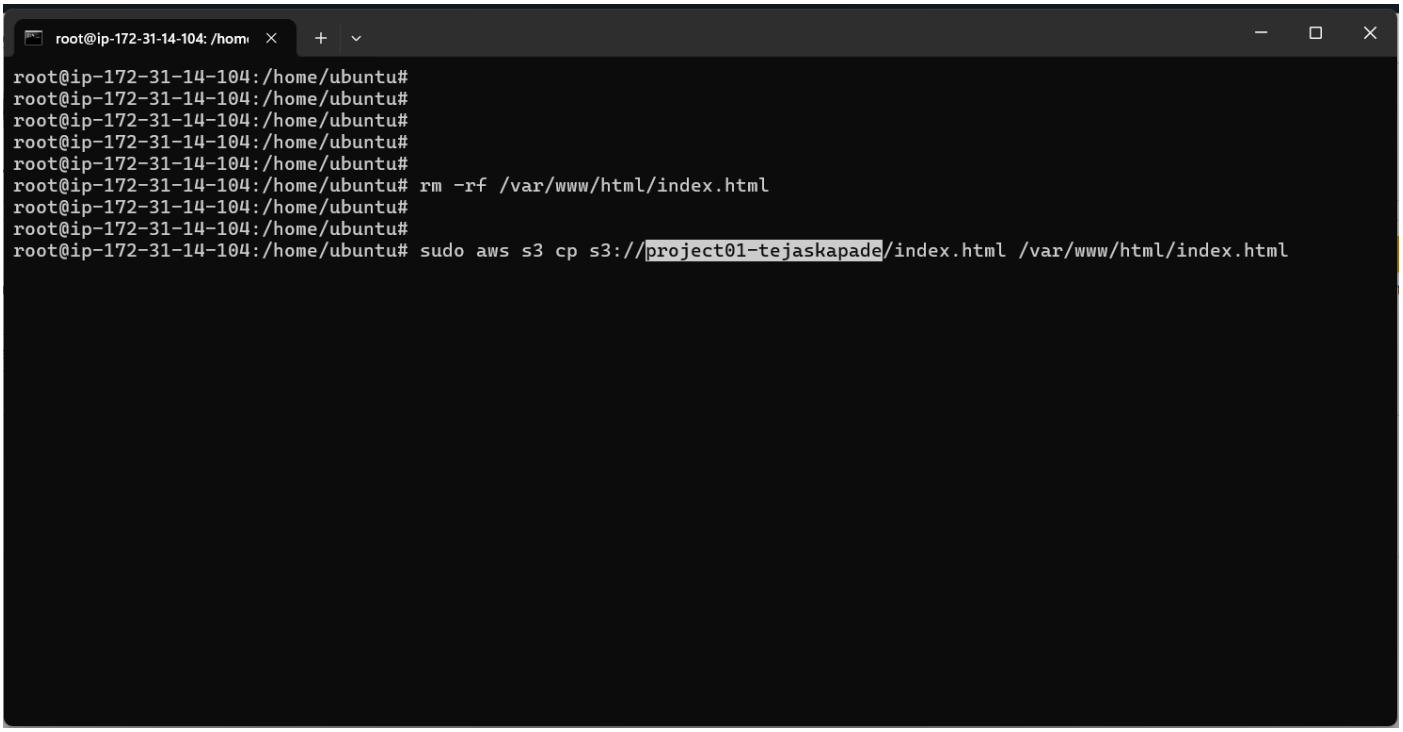


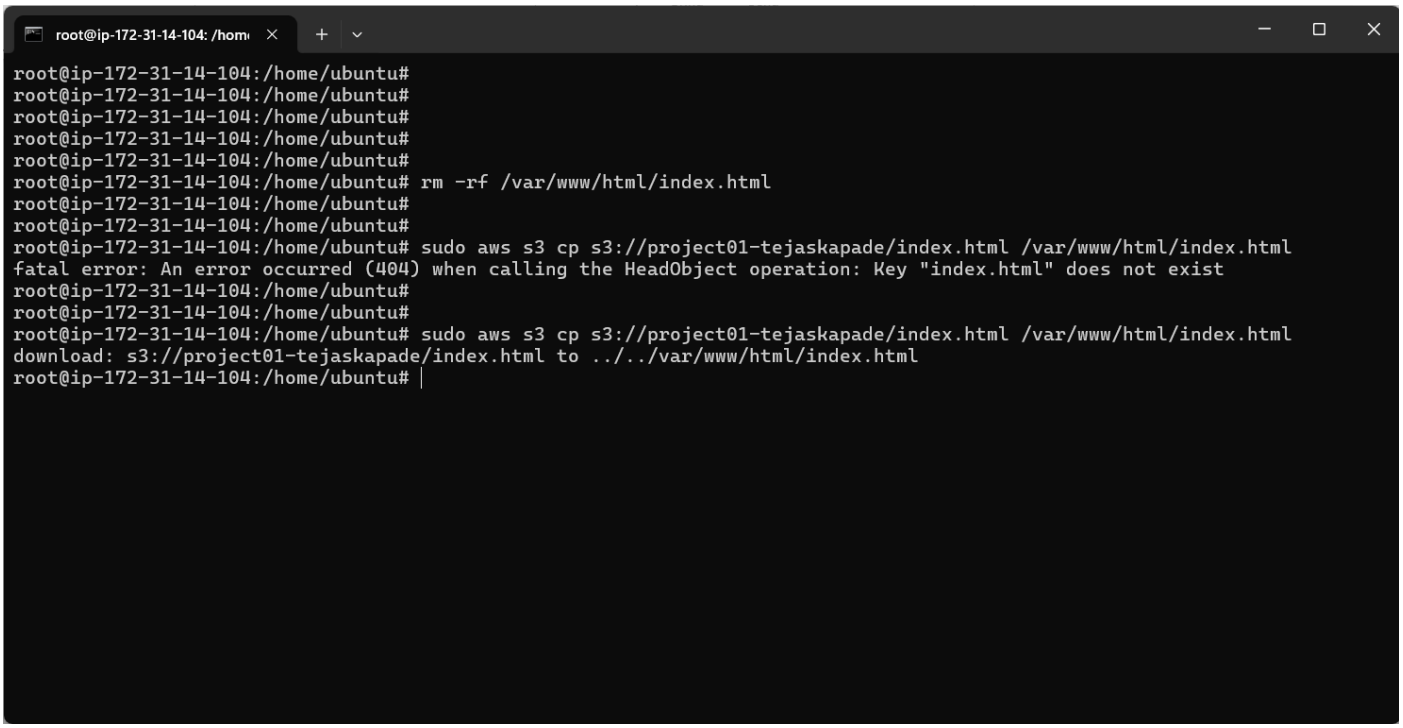
```
root@ip-172-31-14-104: /hom
root@ip-172-31-14-104:/home/ubuntu#
root@ip-172-31-14-104:/home/ubuntu#
root@ip-172-31-14-104:/home/ubuntu# snap install aws-cli --classic
aws-cli (v2/stable) 2.27.53 from Amazon Web Services (aws✓) installed
root@ip-172-31-14-104:/home/ubuntu#
root@ip-172-31-14-104:/home/ubuntu#
root@ip-172-31-14-104:/home/ubuntu#
root@ip-172-31-14-104:/home/ubuntu# aws sts get-caller-identity
{
    "UserId": "AROA43TLRUPD4535J76IB:i-0a0caaff9950e1f65",
    "Account": "883915072455",
    "Arn": "arn:aws:sts::883915072455:assumed-role/Project01-EC2-S3/i-0a0caaff9950e1f65"
}
root@ip-172-31-14-104:/home/ubuntu#
```

7. Upload Files/Backups to S3,

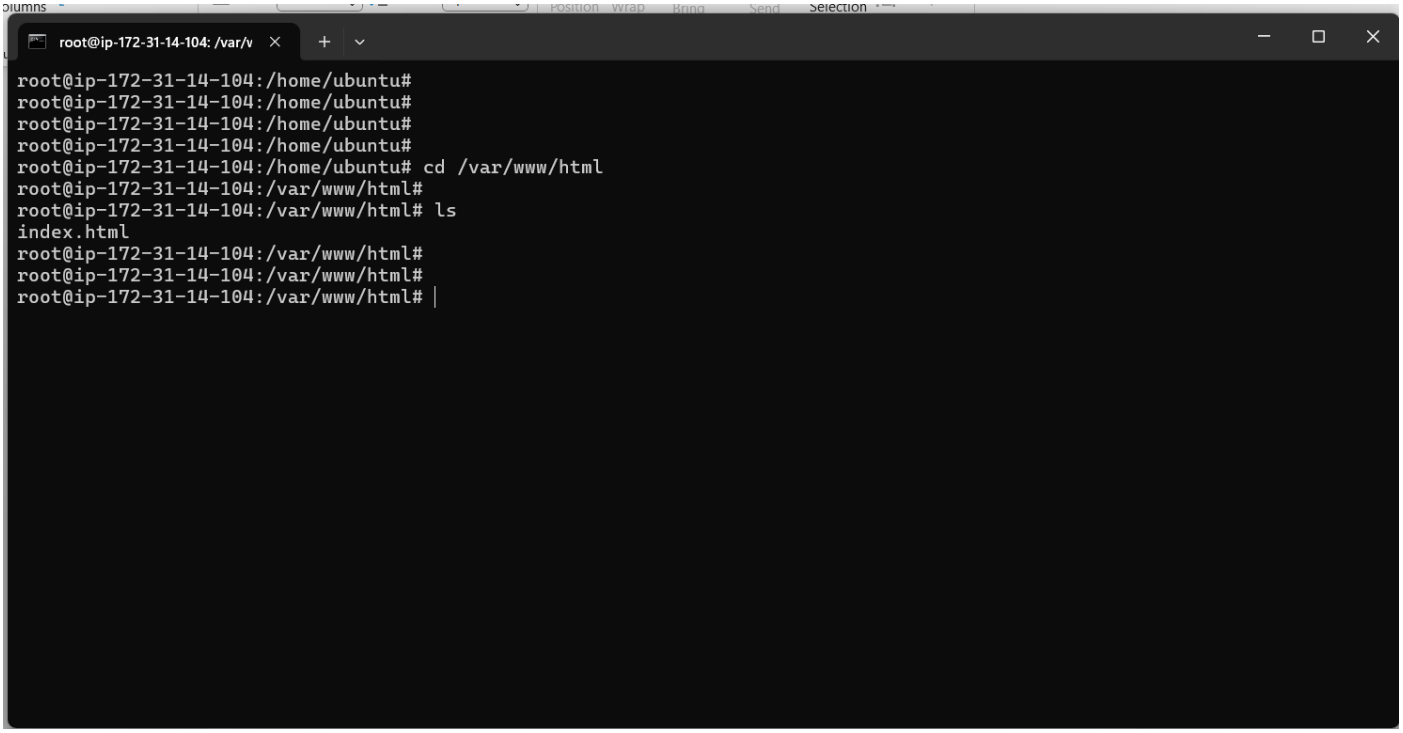


8. We will remove default index.html of apache, and sync our S3 bucket to that file path which is /var/www/html/ and sync our index.html file which is created in S3 bucket.





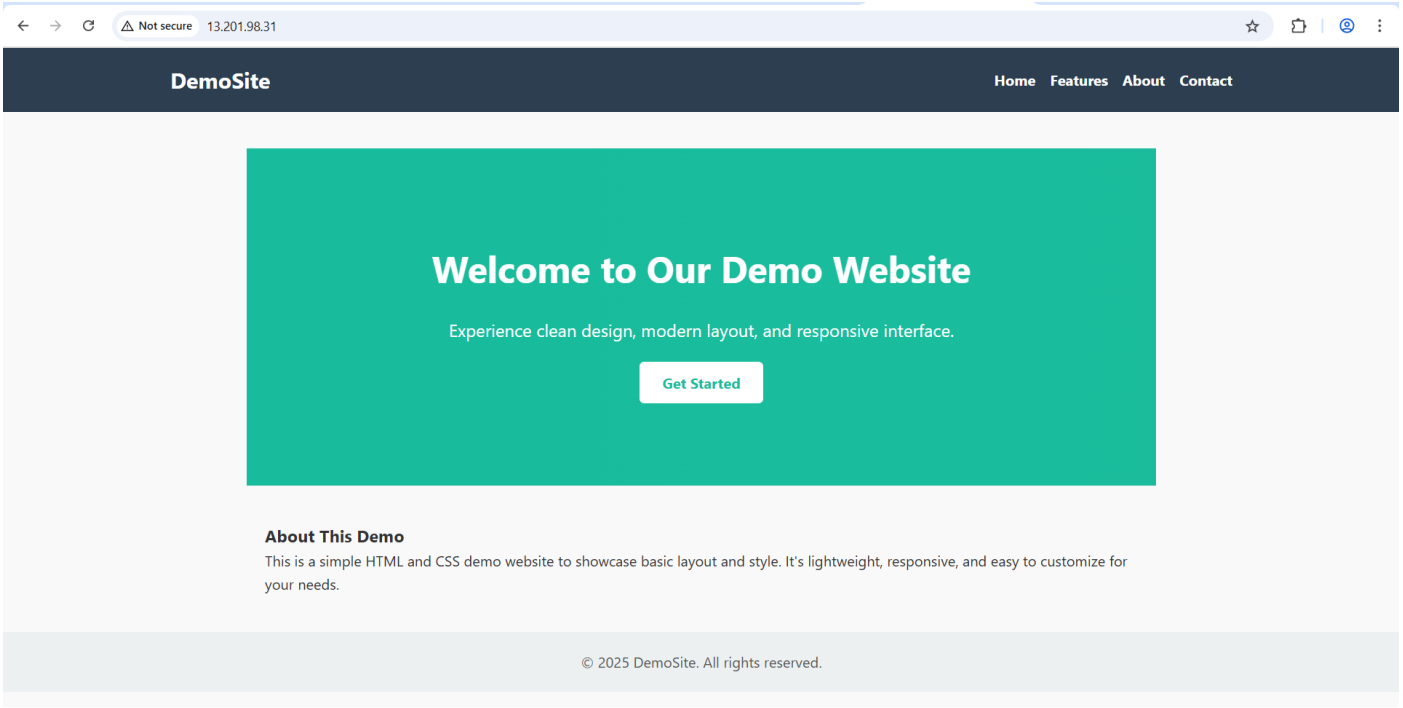
Download success



We can see index.html is now showing.

It was downloaded from our S3 bucket directly, using AWS CLI.

9. Verify: Access EC2 Public IP in Brows er -> Website should load



We can see now webpage is available.

What We Done:

1. Created EC2 instance, S3 bucket, Created IAM role for EC2 to give full access to S3 bucket.
2. Installed apache in EC2 removed default webpage, uploaded index.html to S3 bucket.
3. Installed AWS CLI in EC2 instance and through AWS CLI Commands we have done sync/download index.html from S3 bucket directly to out EC2 instance and checked it was visible using Public IP.