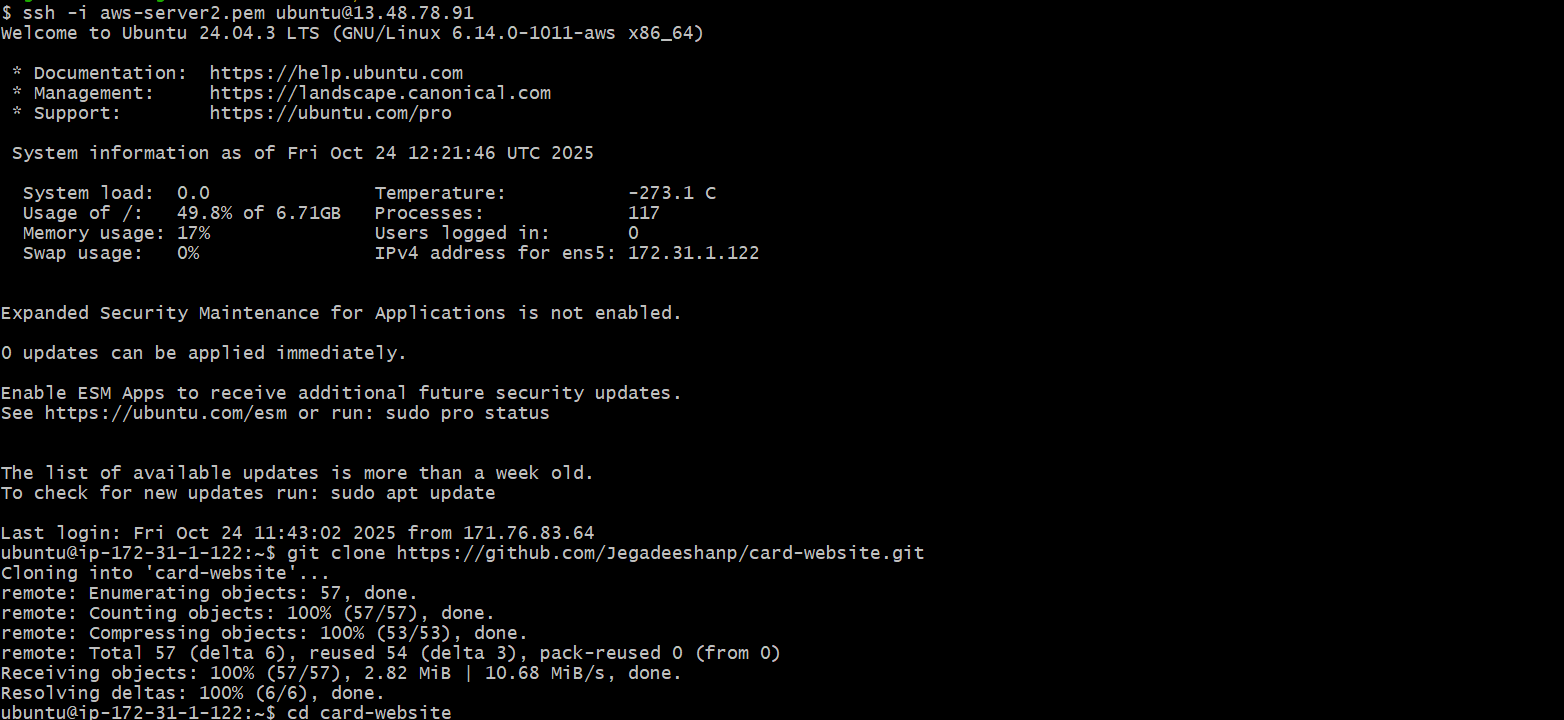
**AWS Task 5**

Clone the repository:

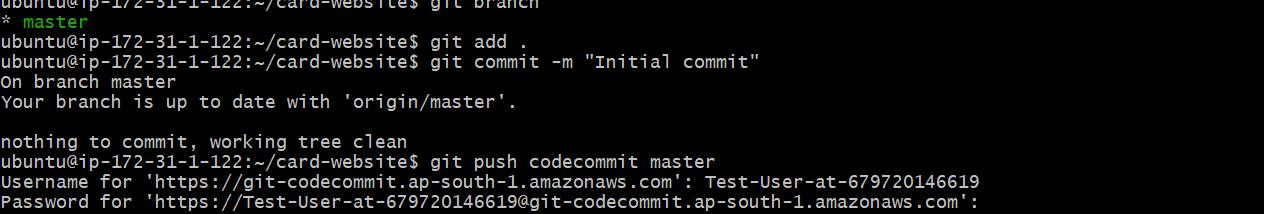


Push Code to AWS CodeCommit

1. Create CodeCommit repo:

* Go to AWS Console → CodeCommit → Create repository.
* Name: card-website-repo.

1. Connect local repo to CodeCommit:



**Note: I was not able to attached the screenshots as my CodeCommit access isn’t working. I tried contacting the AWS customer care but no use.**

**I have mentioned all the steps that I would execute as per the Task.**

Create an S3 Bucket (Optional)

Go to S3 → Create bucket

* Name: card-website-deploy
* Region: same as CodeCommit

**Create a CodeBuild Project**

1. Go to **CodeBuild → Create project**
2. Configure:
   * Name: card-website-build
   * Source: AWS CodeCommit → card-website-repo
   * Environment: Managed image → Ubuntu → Standard
   * Buildspec: Use buildspec.yml

Prepare for CodeDeploy

1.Create appspec.yml in repo root:

2.Set up EC2 instance:

sudo yum update -y

sudo yum install -y httpd

sudo systemctl start httpd

sudo systemctl enable httpd

sudo yum install ruby -y

cd /home/ec2-user

wget https://aws-codedeploy-ap-south-1.latest.s3.amazonaws.com/install

chmod +x ./install

sudo ./install auto

sudo systemctl start codedeploy-agent

3.Attach AmazonEC2RoleforAWSCodeDeploy role to EC2.

Create CodeDeploy Application

1.Create application:

Name: card-website-cd

Compute platform: EC2/On-premises

2.Create deployment group:

Name: card-website-deployment-group

Select EC2 instance / Deployment type: In-place

Service role: CodeDeployRole

Create CodePipeline

1.Go to CodePipeline → Create pipeline

2.Configure:

Name: card-website-pipeline

Service role: Let AWS create a new one

3.Add stages:

Source: CodeCommit → card-website-repo → main

Build: CodeBuild → card-website-build

Deploy: CodeDeploy → card-website-cd → card-website-deployment-group

Test Deployment

1.Push a change to CodeCommit:

echo "<!-- test update -->" >> index.html

git add .

git commit -m "Test pipeline deployment"

git push codecommit main

2.Pipeline will trigger automatically.

3.Open the EC2 public IP in browser to view the website