Read the following instructions before completing the given task.

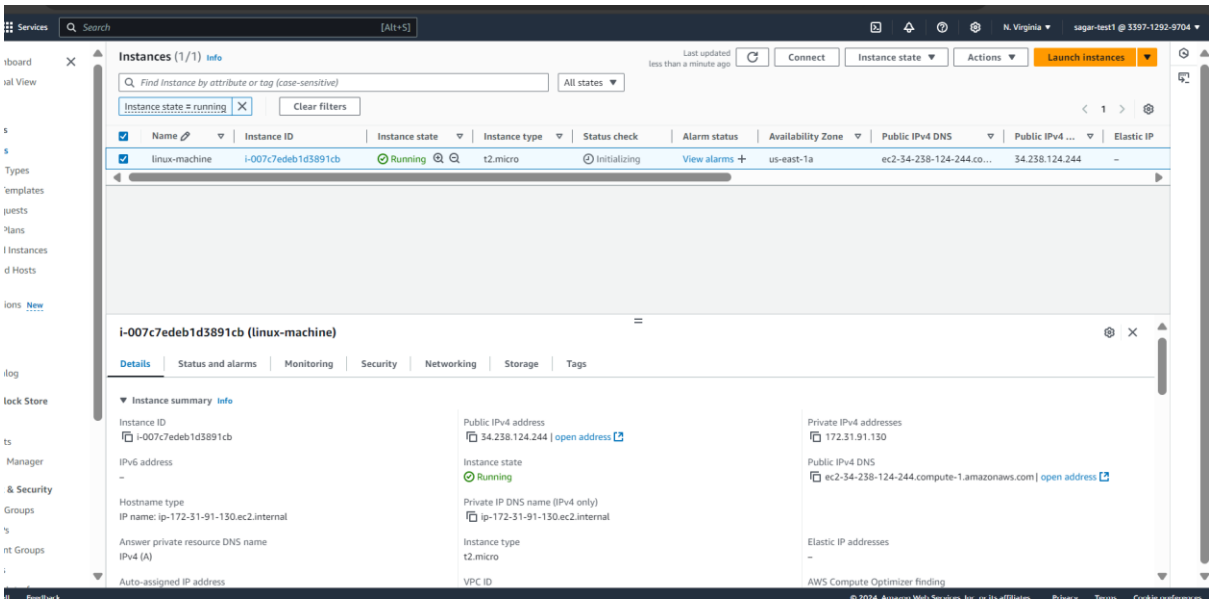
Step 1: Click on the Online Task Link
Step 2: Download the document
Step 3: Check the online questions
Step 4: Complete the task and take a screenshot for each and every task and upload
Step 5: Save the document
Step 6: Upload the document in Google form

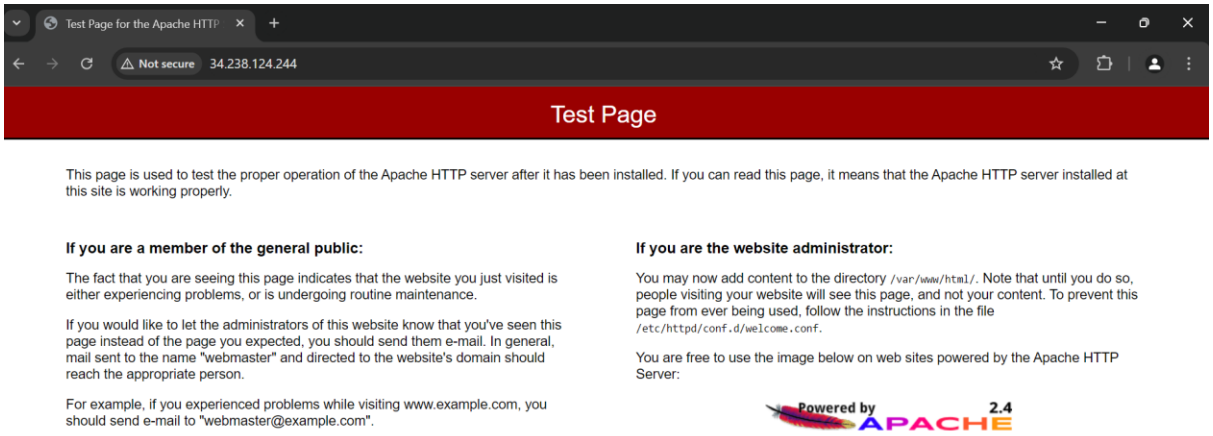
- **AWS Account is mandatory to complete the tasks**.
- **Include AWS account Name in your screenshots**
- **Resume / Copied and similar tasks will be rejected directly.**

Assignment 1.

Configure EC2 linux machine and install apache configuration

Upload the final output Screenshot

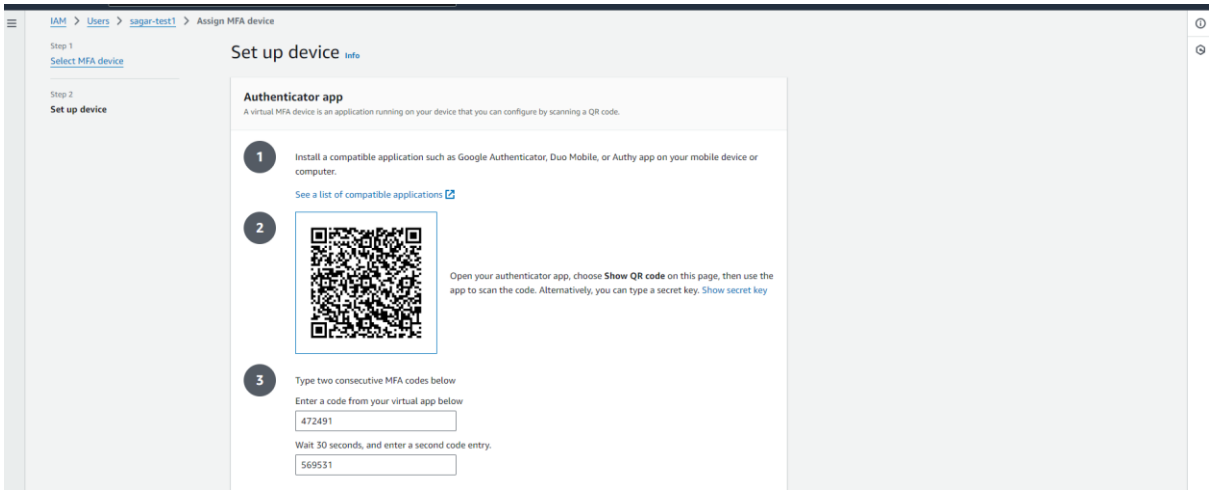


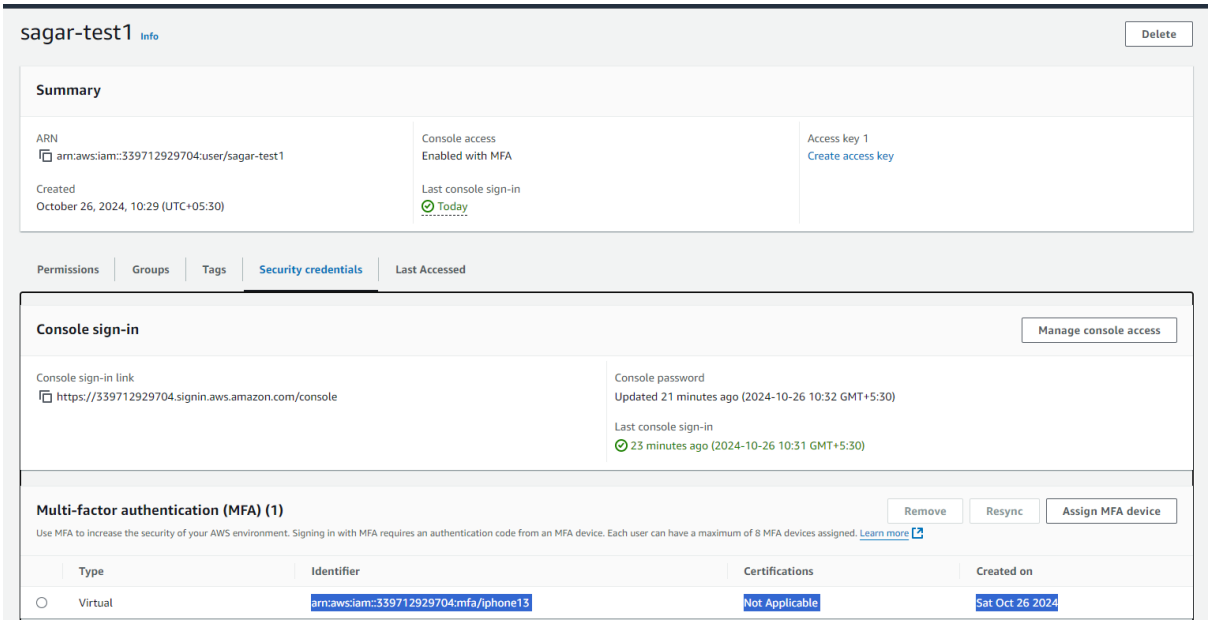


Assignment 2.

Create an enforced MFA policy, apply it to the IAM user,

Upload the final output Screenshot





Assignment 3.

Implement Auto Scaling: Create an Auto Scaling group that automatically launches new EC2 instances based on predefined rules. You can use the EC2 instance that you created in Task 1 as the base instance for the Auto Scaling group. Test the Auto Scaling group by simulating a surge in traffic to the web server.

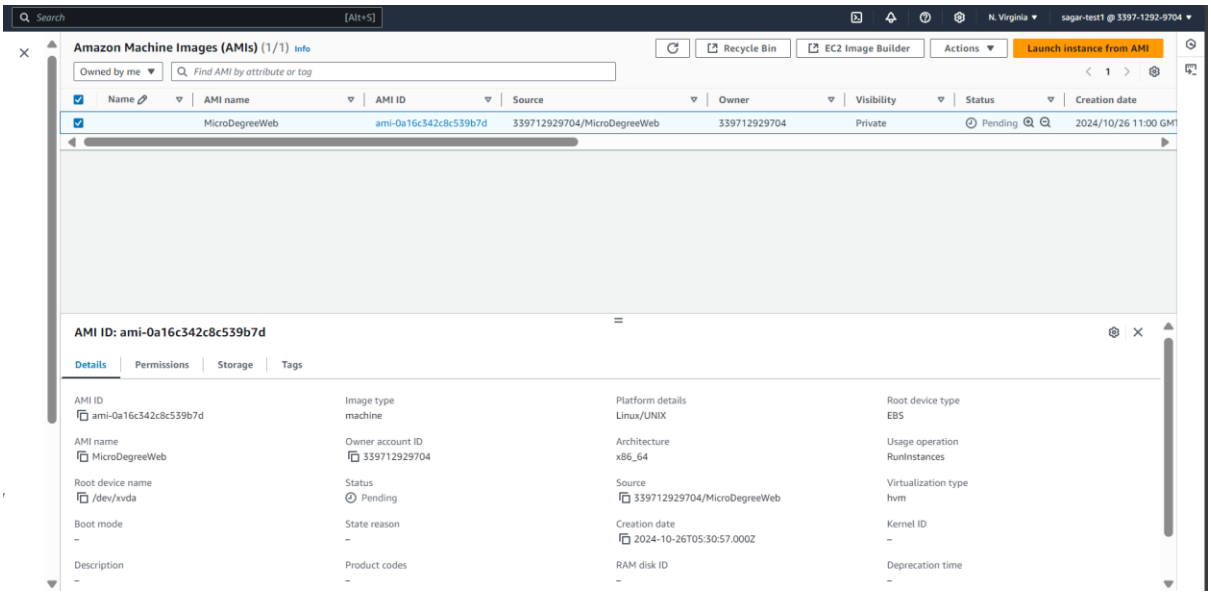
Upload the final output Screenshot

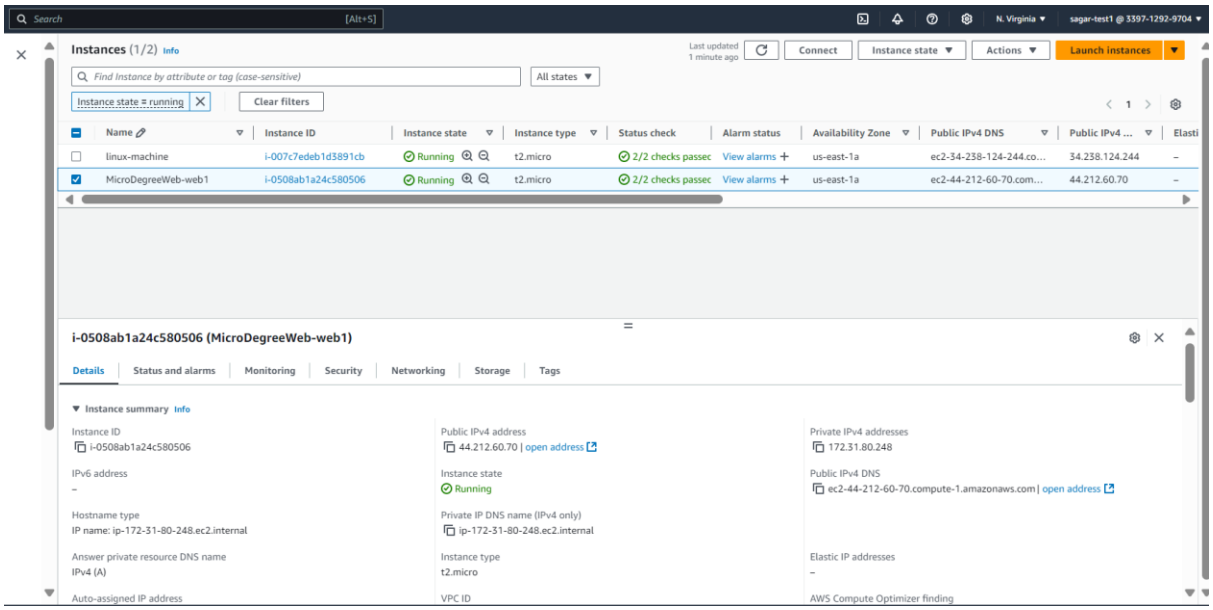
Assignment 4

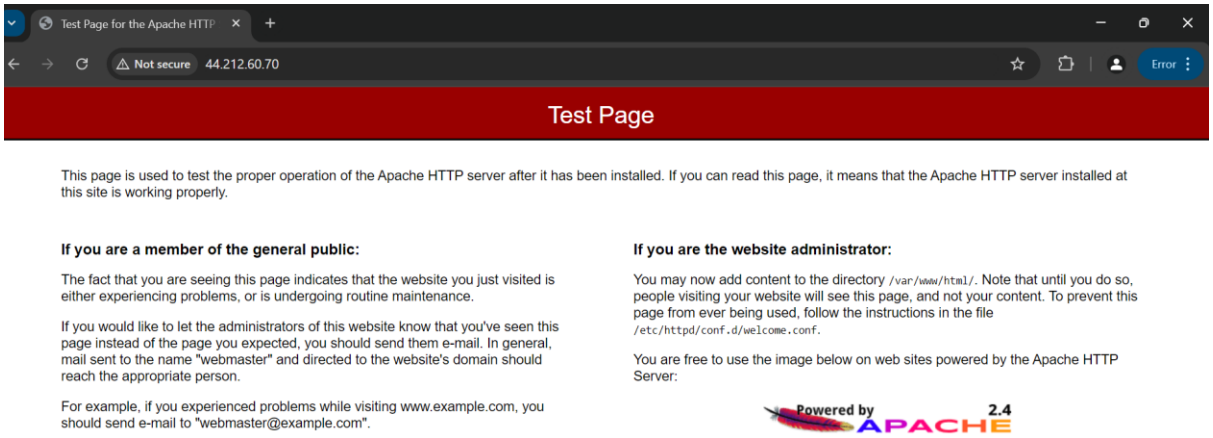
Creating a Custom Amazon Machine Image (AMI)
- Launch a New EC2 Instance

- Install http on the new instance, enable the http service to start at boot.
- Create a New AMI from customised instance and name the AMI MicroDegreeWeb
- Launch a New Instance Using the Custom AM
- Verify that http is running.

Upload the final output Screenshot







Assignment 5

Set up Cross-Region S3 Bucket Replication

- Create an S3 Bucket and Enable Replication
- Test Replication and Observe Results

Upload the final output Screenshot

Assignment 6

Configure a VPC peering connection between VPCs in different regions and please share the configuration details with Screenshot.

Upload the final output Screenshot

Assignment 7

Set up a basic EC2 instance: Create an Amazon Elastic Compute Cloud (EC2) instance and configure it to run a web server. You can choose any Linux-based operating system and web server software of your choice. Once the instance is up and running, access it using SSH and verify that the web server is serving web pages.

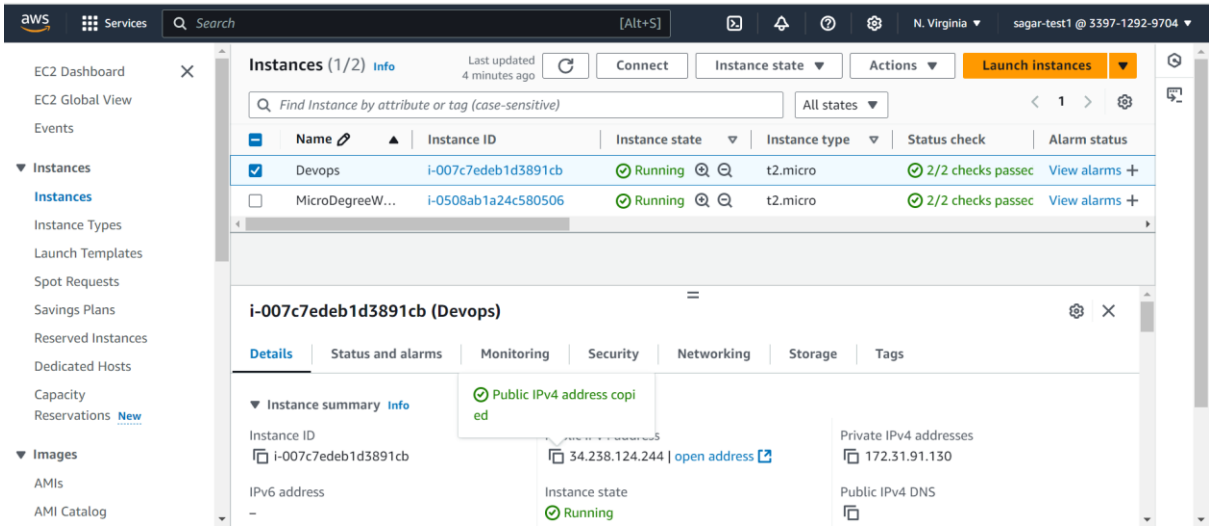
Upload the final output Screenshot

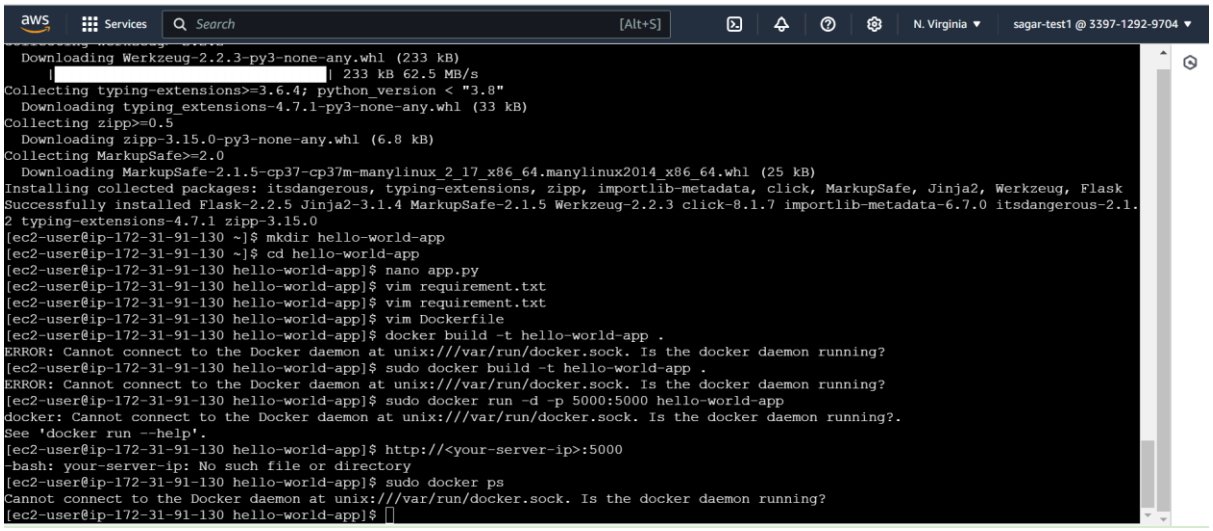
Assignment 8

Web Application:

- **Develop a simple "Hello World" web application (you can use any programming language or framework of your choice).**
- **Package the application into a deployable artifact (e.g., a JAR file, Docker image).**

Upload the final output Screenshot

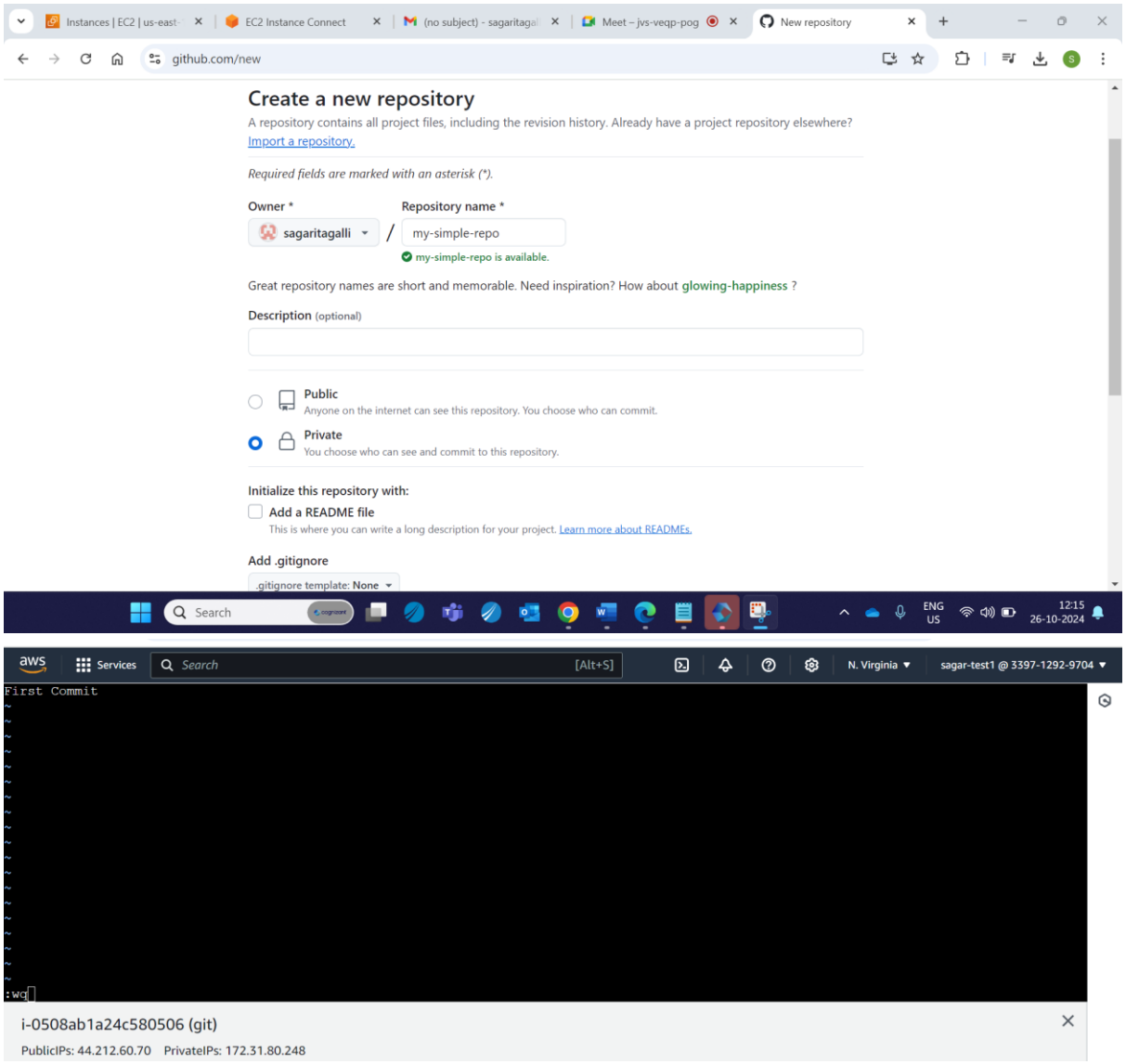


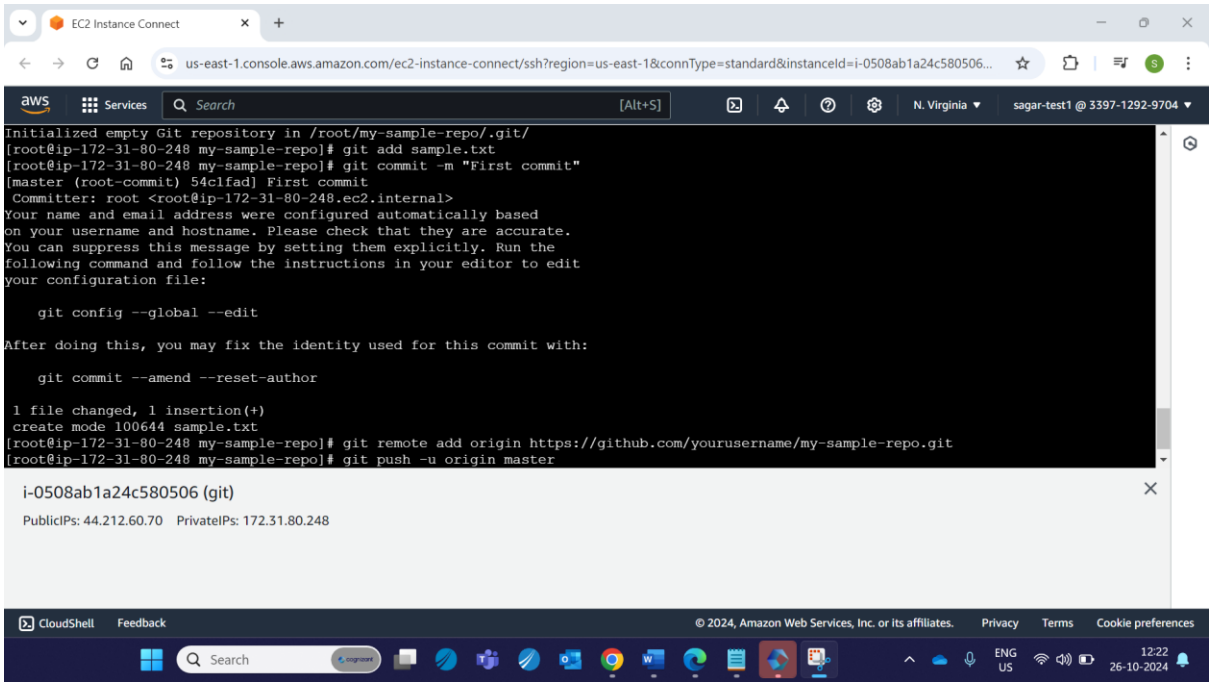


Assignment 9

Create a new GitHub repository, push sample files from local to remote, with a commit message of 'First commit'. Please note that the repository should be set to private.

Upload the final output Screenshot





Assignment 10

Create a Jenkins freestyle project, configure a sample build with the Maven plugin, and please share the output of the build.

Upload the final output Screenshot

Assignment 11

Configure email notifications
Description: Configure Jenkins to send email notifications when a build fails.
Instructions:

Upload the final output Screenshot

Assignment 12

Build a custom docker image using Ubuntu as a base docker image and run the nginx application. - this docker image should be built using a Dockerfile. Once the docker image is build, start the docker image using the host network and make it accessible on Public IP

Upload the final output Screenshot

Assignment 13

Create a Staging branch in GitHub and push code from the local repository to the Remote and share the full commands screen

Assignment 14

Working with Docker Images
- Pull the latest `httpd` image.
- Pull the latest `alpine` image.
- verify images pulled and create 2 containers in each server

Upload the final output Screenshot

Assignment 15

Setting Up Continuous Integration and Deployment (CI/CD):
Jenkins is often used for implementing CI/CD pipelines to automate the build, test, and deployment processes. Create a pipeline job using Jenkins Pipeline DSL (declarative or scripted) or a Jenkins file.
Define the stages of your pipeline, including building, testing, code analysis, and deployment.
Configure Jenkins to trigger the pipeline based on code changes, commits, or other events.

Upload the final output Screenshot