

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

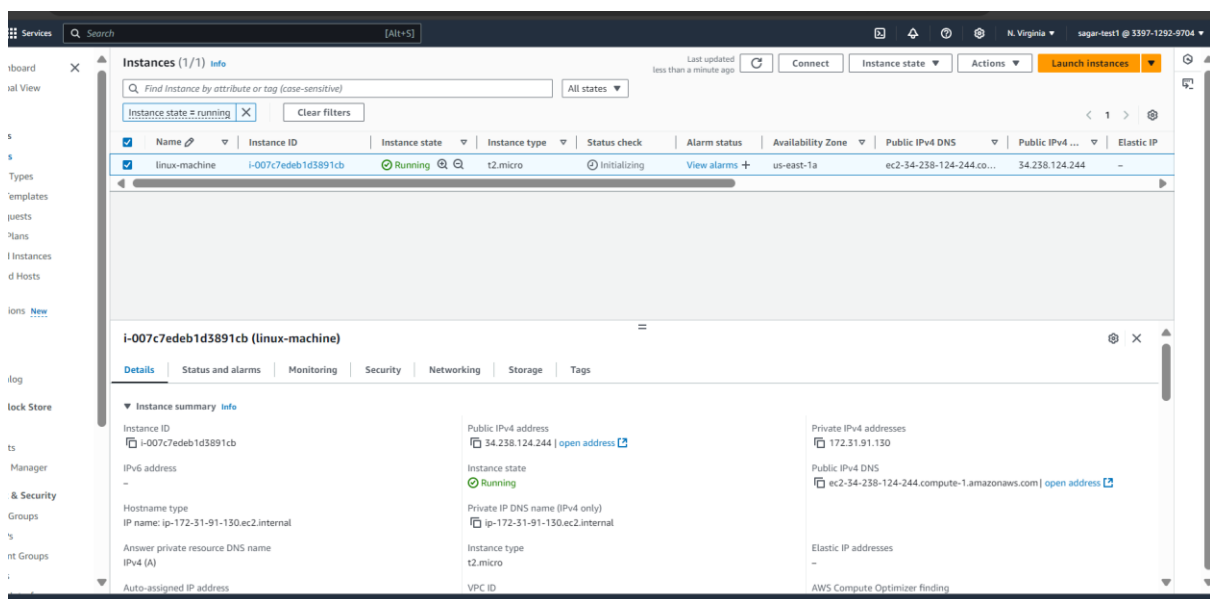
Step 6: Upload the document in Google form

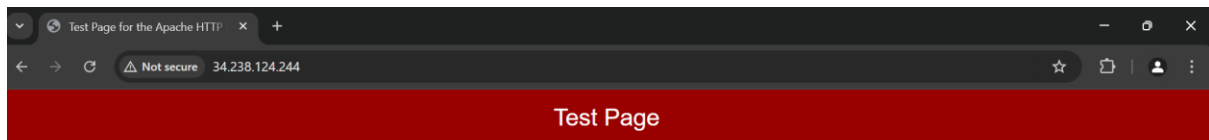
- **AWS Account is mandatory to complete the tasks.**
- **Include AWS account Name in your screenshots**
- **Resume / Copied and similar tasks will be rejected directly.**

Assignment 1.

Configure EC2 linux machine and install apache configuration

Upload the final output Screenshot





This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

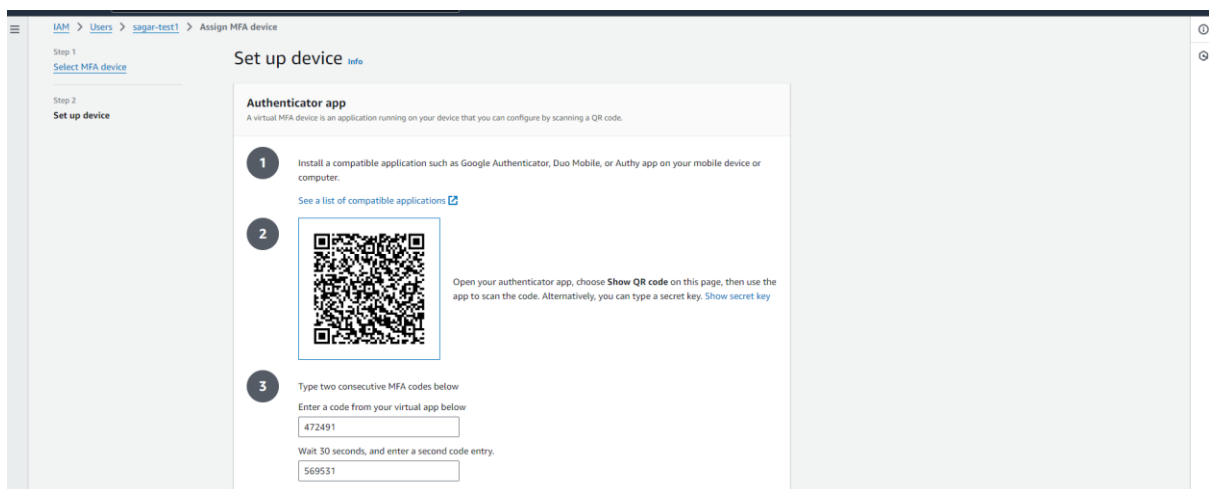
You are free to use the image below on web sites powered by the Apache HTTP Server:



Assignment 2.

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

Upload the final output Screenshot



sagar-test1

info

Delete

Summary

ARN
am:aws:iam::339712929704:user/sagar-test1

Created
October 26, 2024, 10:29 (UTC+05:30)

Console access
Enabled with MFA

Last console sign-in
Today

Access key 1
Create access key

Permissions

Groups

Tags

Security credentials

Last Accessed

Console sign-in

Manage console access

Console sign-in link
https://339712929704.signin.aws.amazon.com/console

Console password
Updated 21 minutes ago (2024-10-26 10:32 GMT+5:30)

Last console sign-in
23 minutes ago (2024-10-26 10:31 GMT+5:30)

Multi-factor authentication (MFA) (1)

Remove

Resync

Assign MFA device

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. [Learn more](#)

| Type | Identifier | Certifications | Created on |
|-------------------------------|---------------------------------------|----------------|-----------------|
| <input type="radio"/> Virtual | am:aws:iam::339712929704:mfa/iphone13 | Not Applicable | Sat Oct 26 2024 |

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.

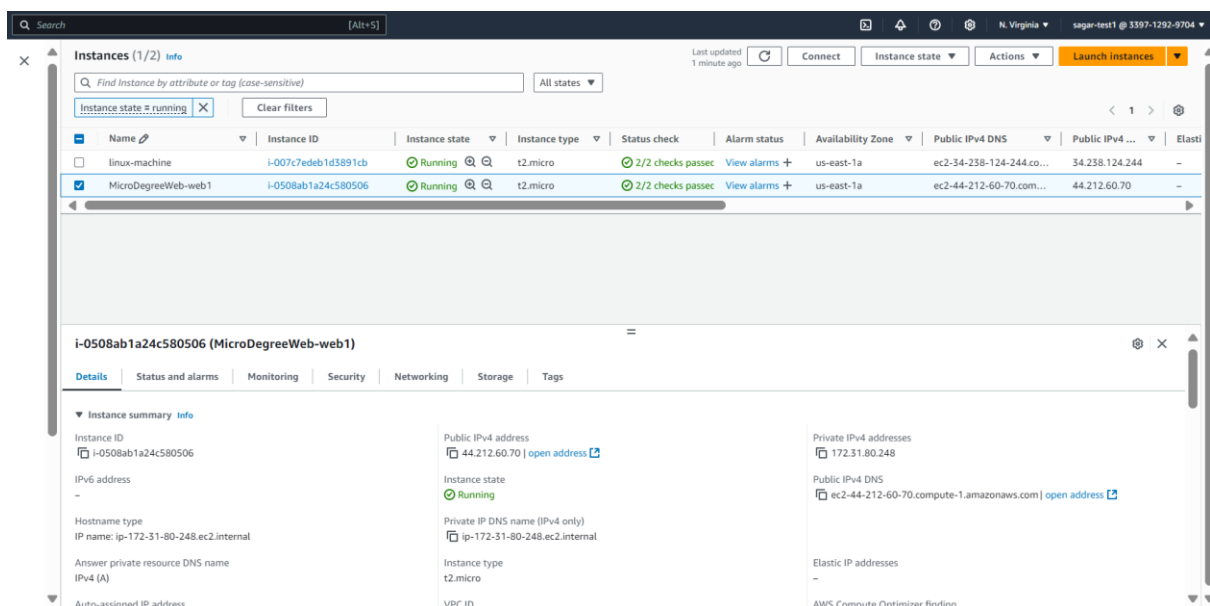
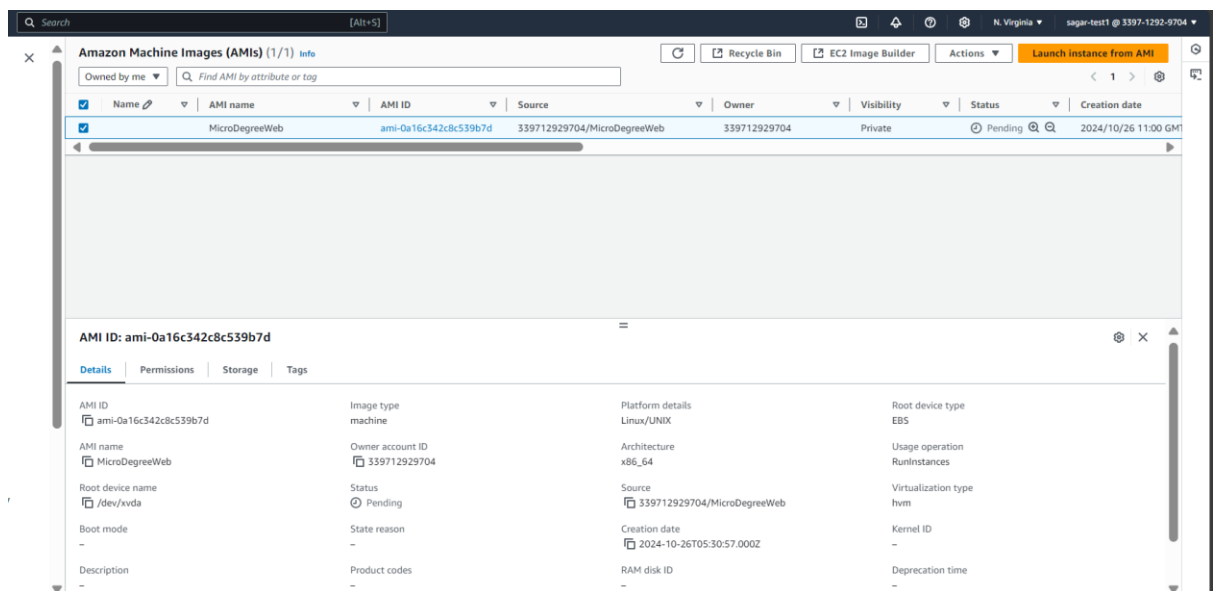
Upload the final output Screenshot

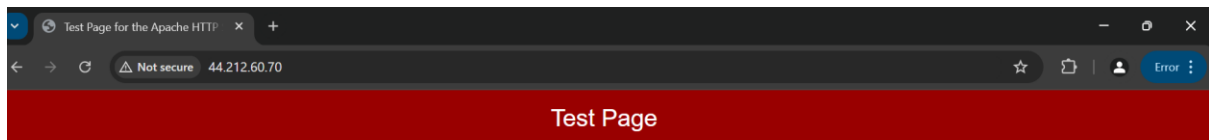
Assignment 4

**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





This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



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.

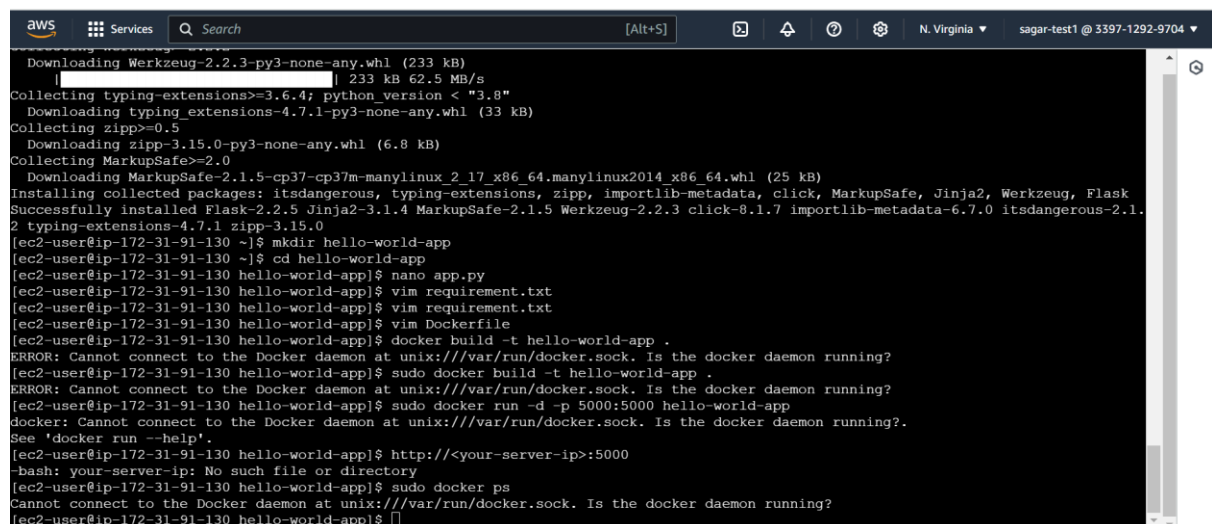
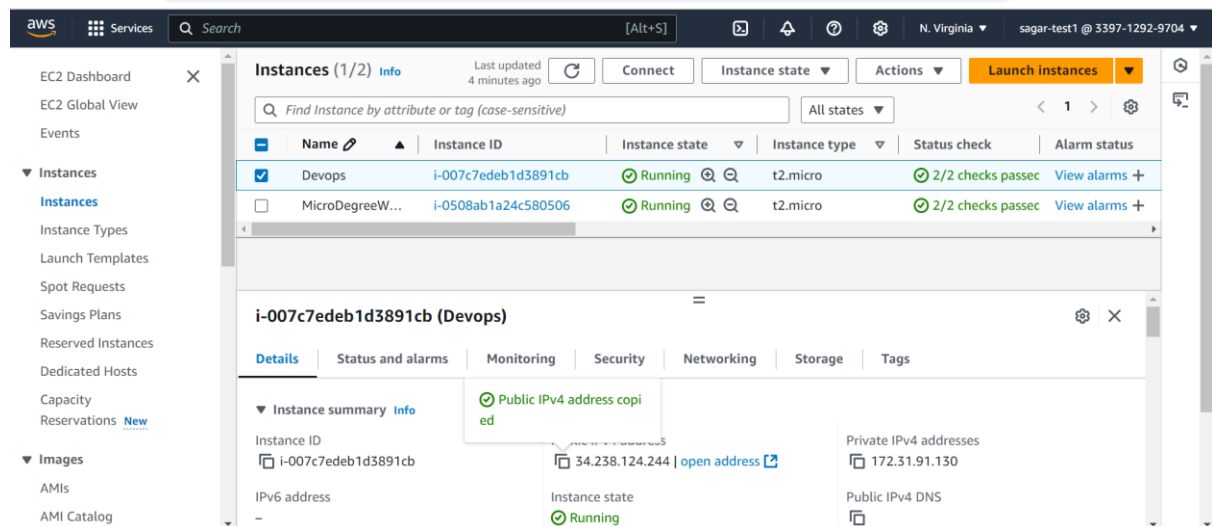
Upload the final output Screenshot

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

Instances | EC2 | us-east | EC2 Instance Connect | (no subject) - sagaritagalli | Meet - jvs-veqp-pog | New repository

github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

sagaritagalli / my-simple-repo

my-simple-repo is available.

Great repository names are short and memorable. Need inspiration? How about glowing-happiness ?

Description (optional)

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

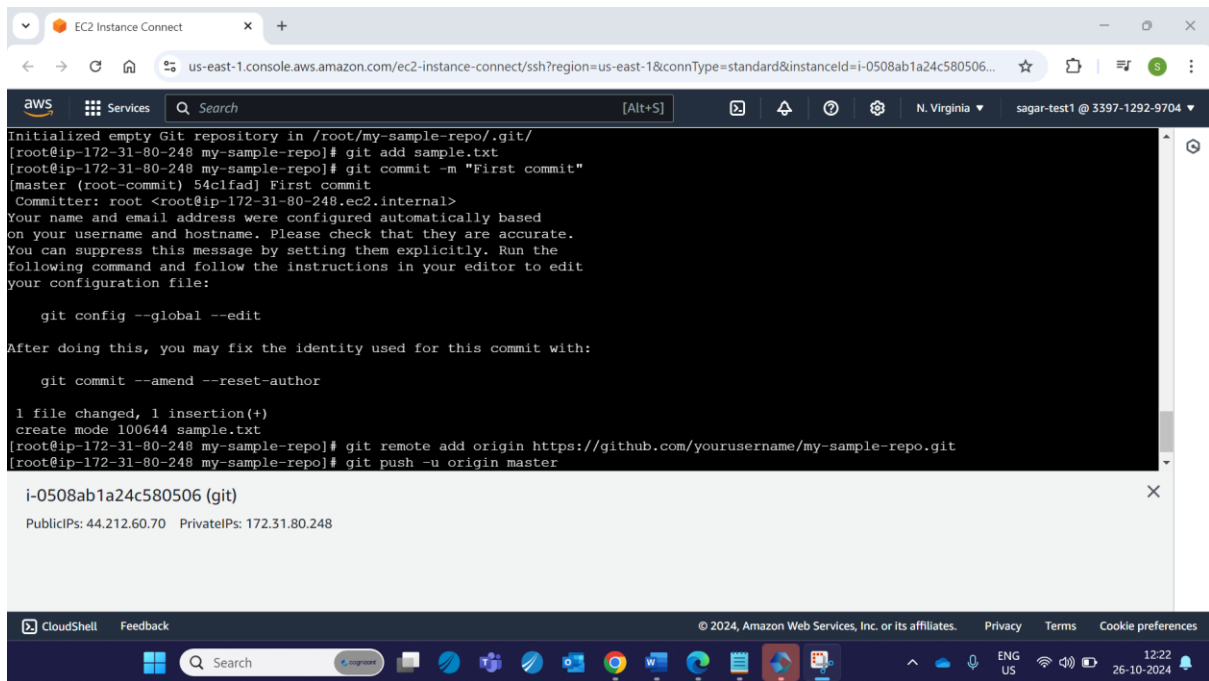
Add .gitignore
gitignore template: None

aws Services Search [Alt+S] N. Virginia sagar-test1 @ 3397-1292-9704

First Commit

i-0508ab1a24c580506 (git)

PublicIPs: 44.212.60.70 PrivateIPs: 172.31.80.248



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