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

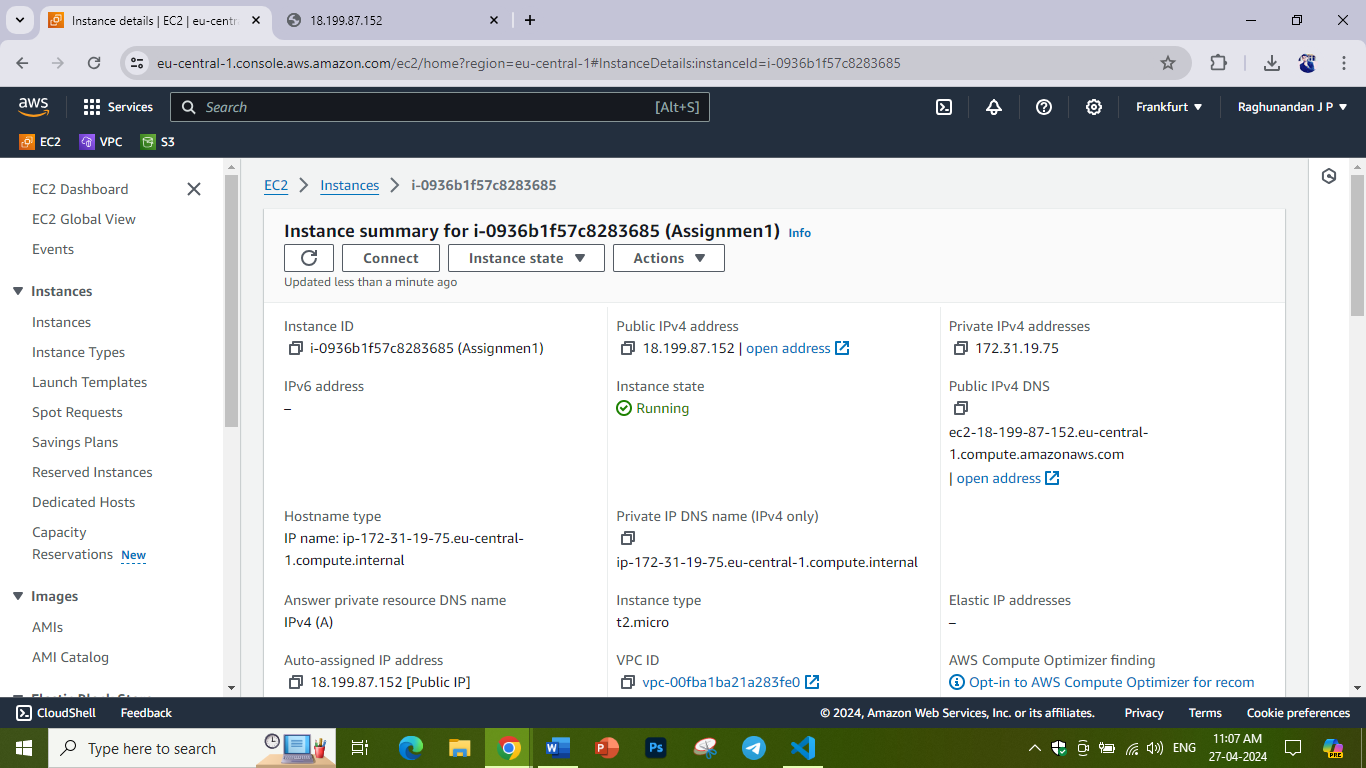
Step 6: Upload the document in Google form

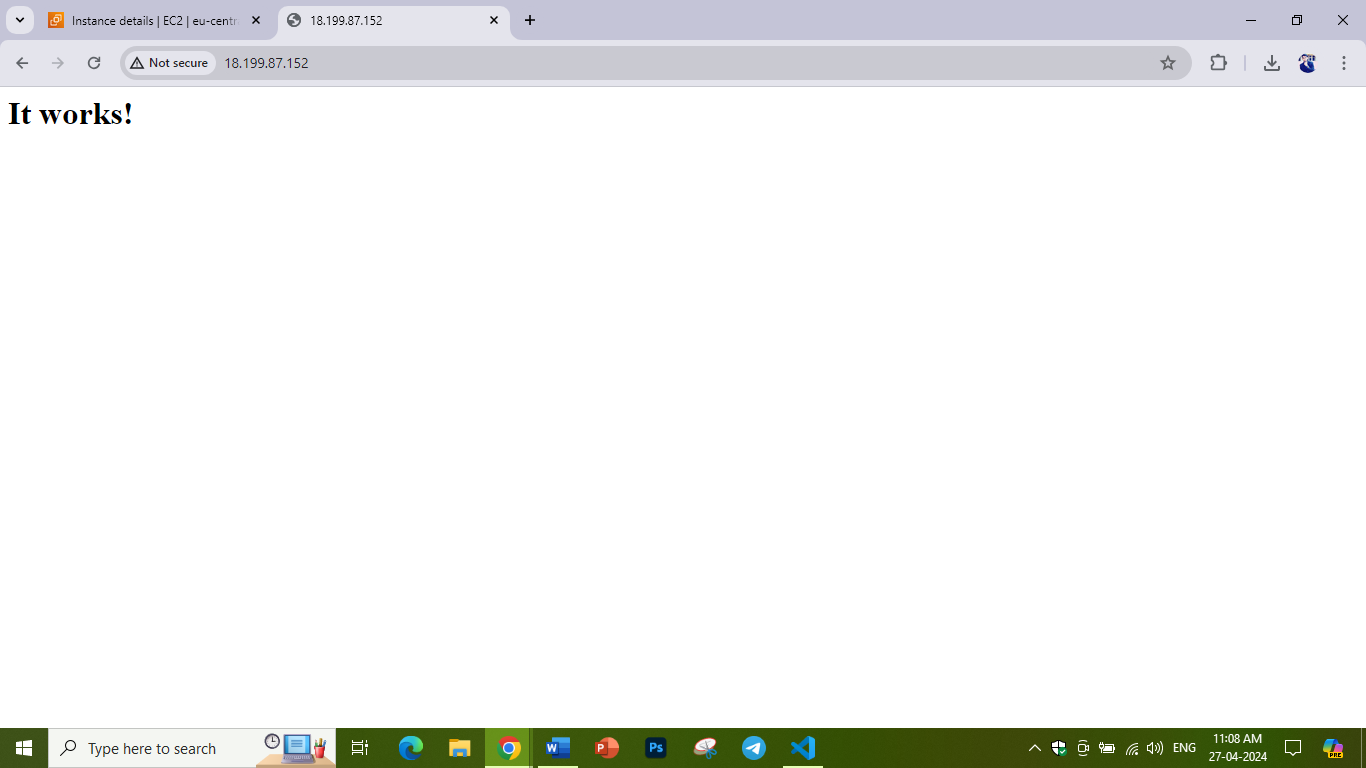
* **AWS Account is mandatory to complete the tasks**.
* **Include AWS account Name in your screenshots**
* **Resume / Copied and similar tasks will be rejected directly.**

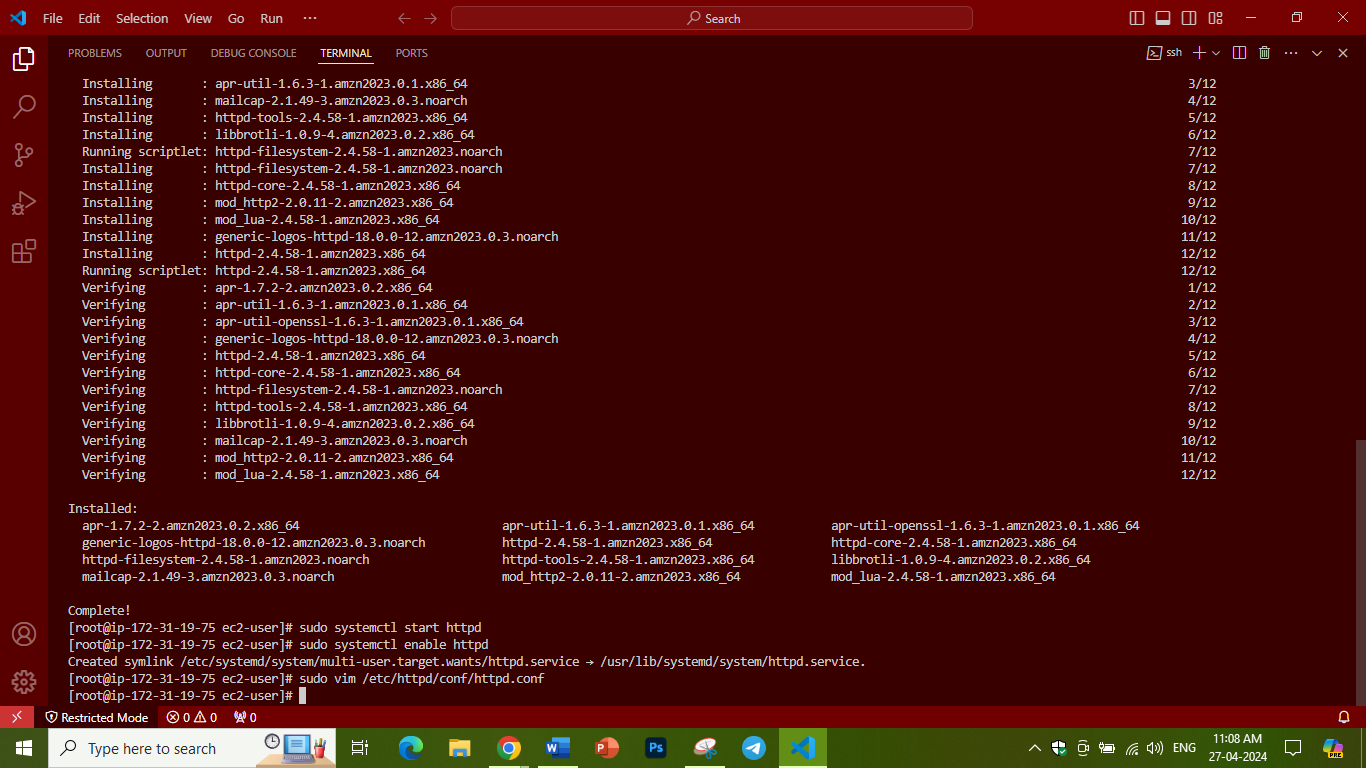
**Assignment 1.**

**Configure ec2 linux machine and install apache configuration**

**Upload the final output Screenshot**







**Assignment 2.**

**Associate an Elastic IP to an EC2 Instance using Terraform**

**- Create an ec2 instance using terraform workflow**

**- associate an elastic IP**

**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**

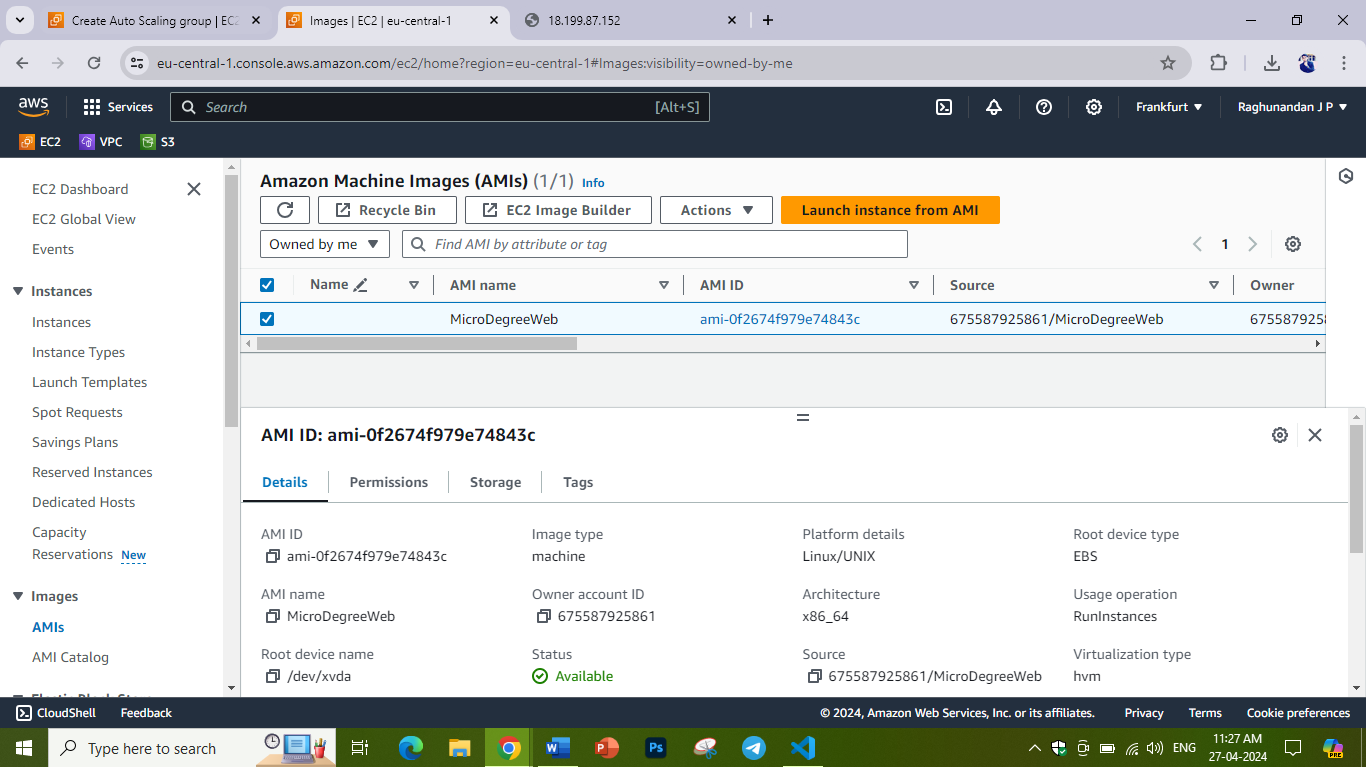
**- Install http on the new instance, enable the http service to start at boot.**

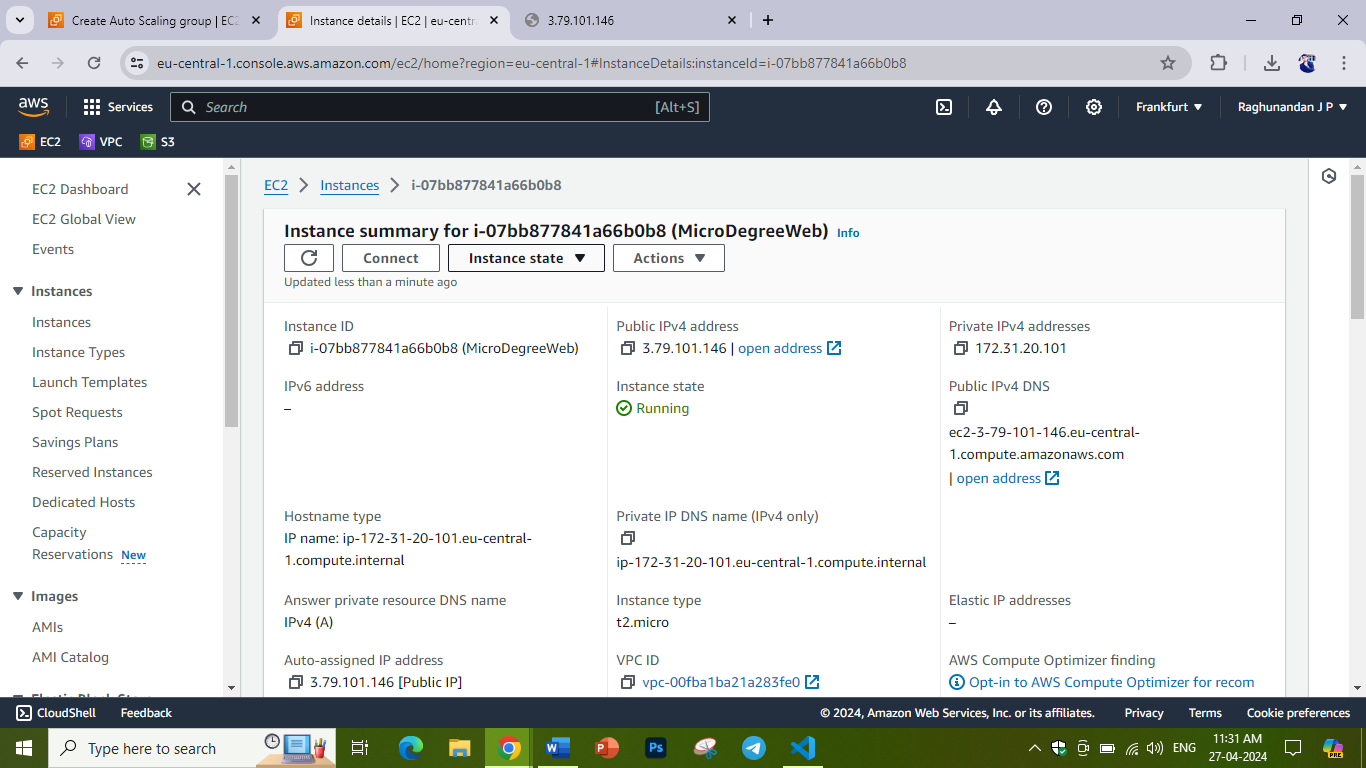
**- Create a New AMI from customised instance and name the AMI MicroDegreeWeb**

**- Launch a New Instance Using the Custom AM**

**- Verify that http is running.**

**Upload the final output Screenshot**

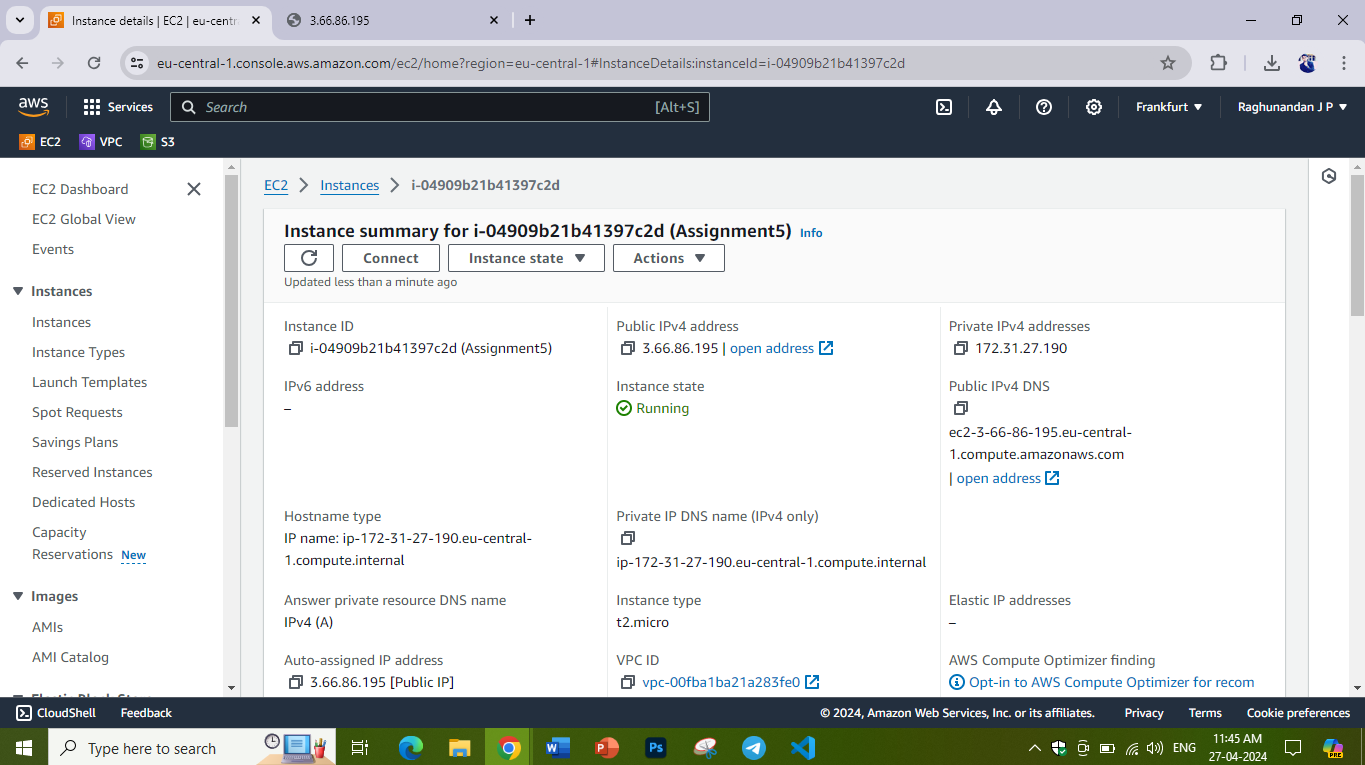
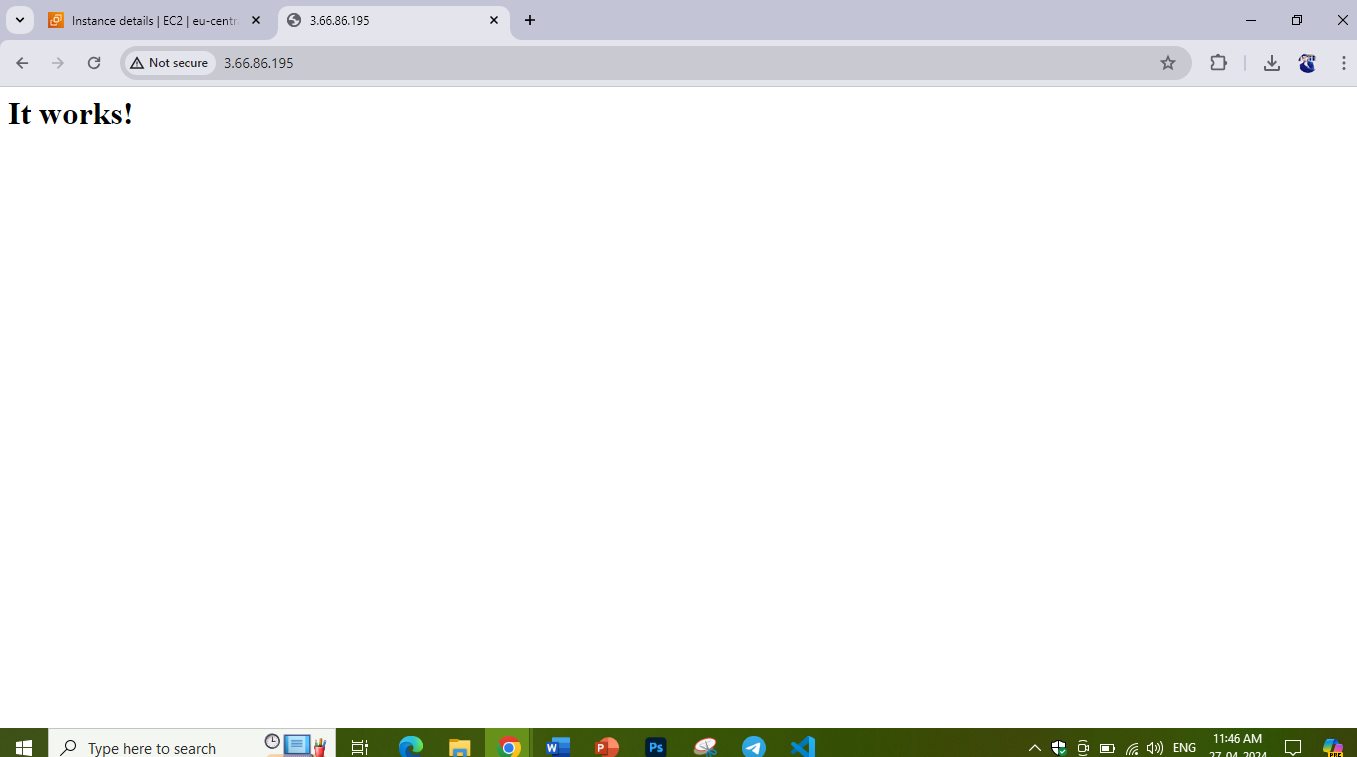


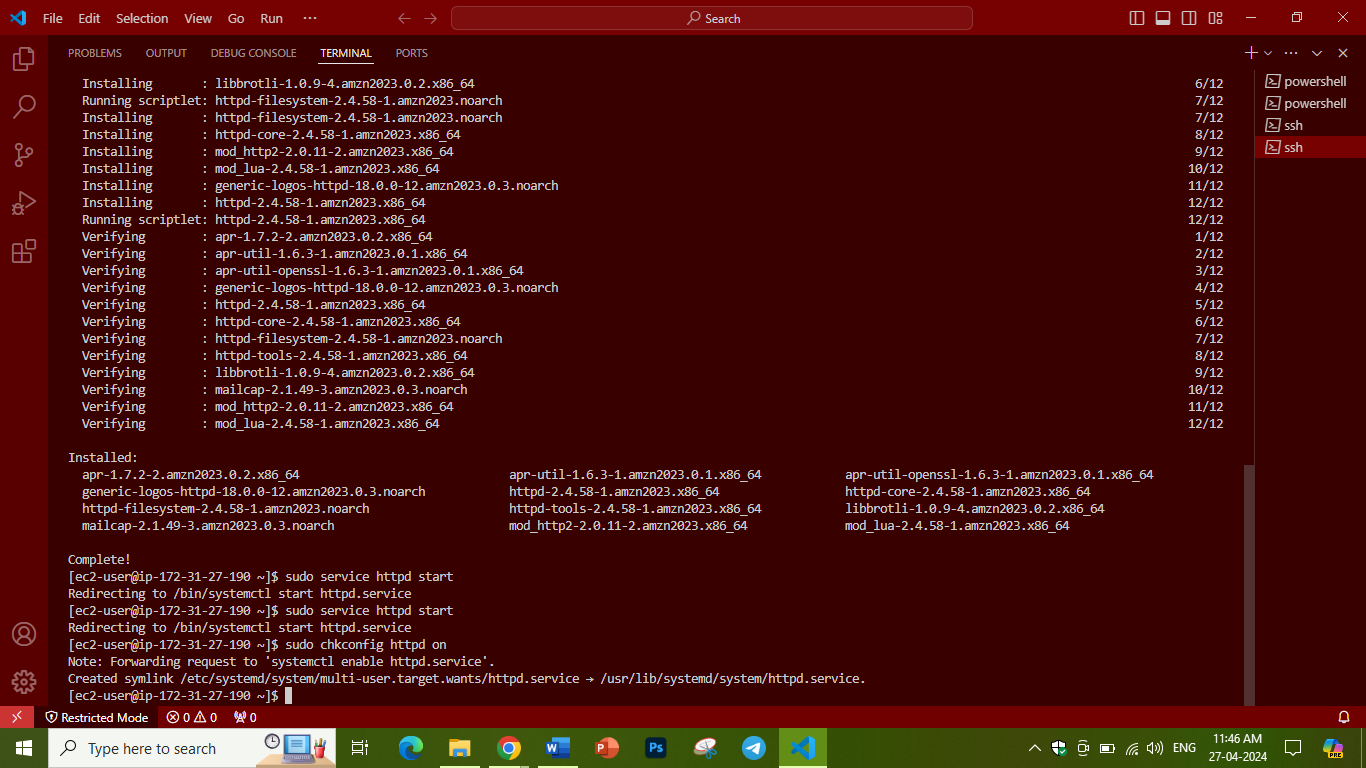
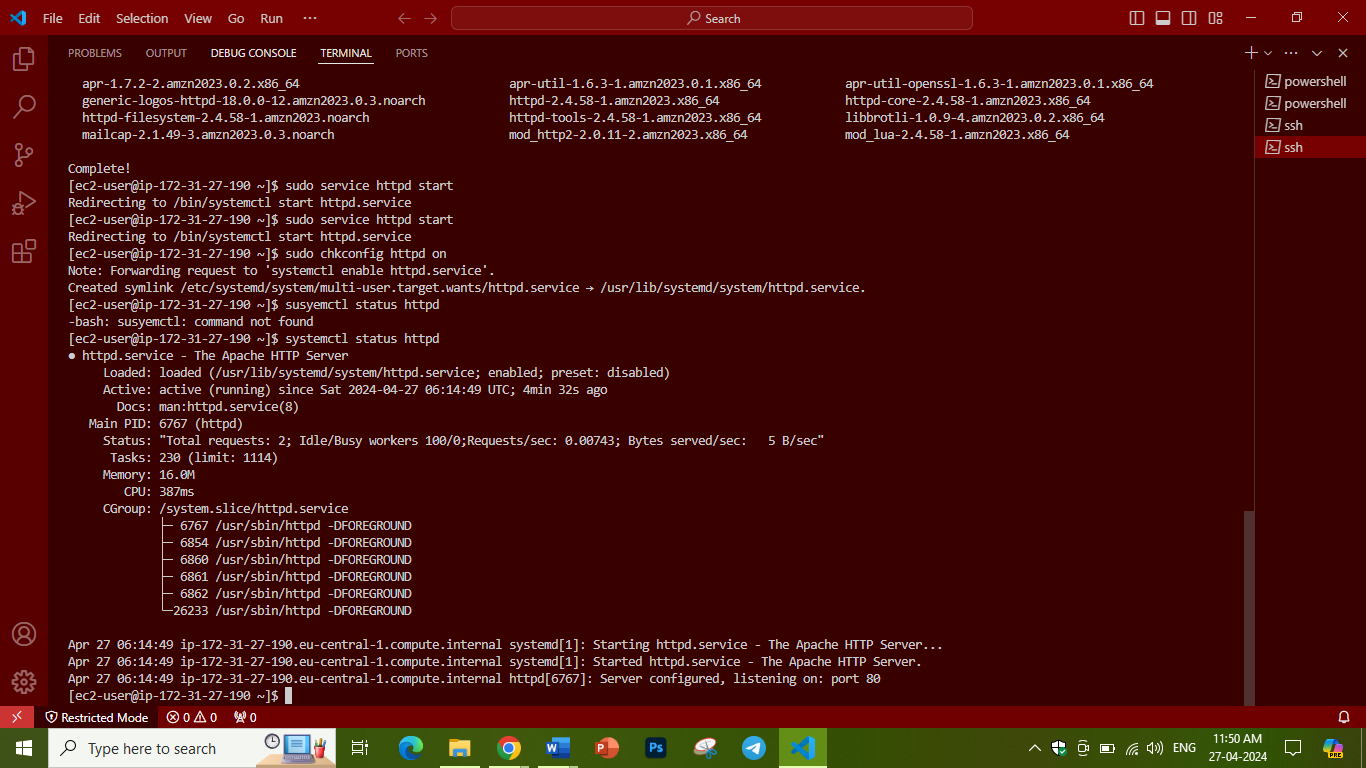


**Assignment 5**

Set up 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

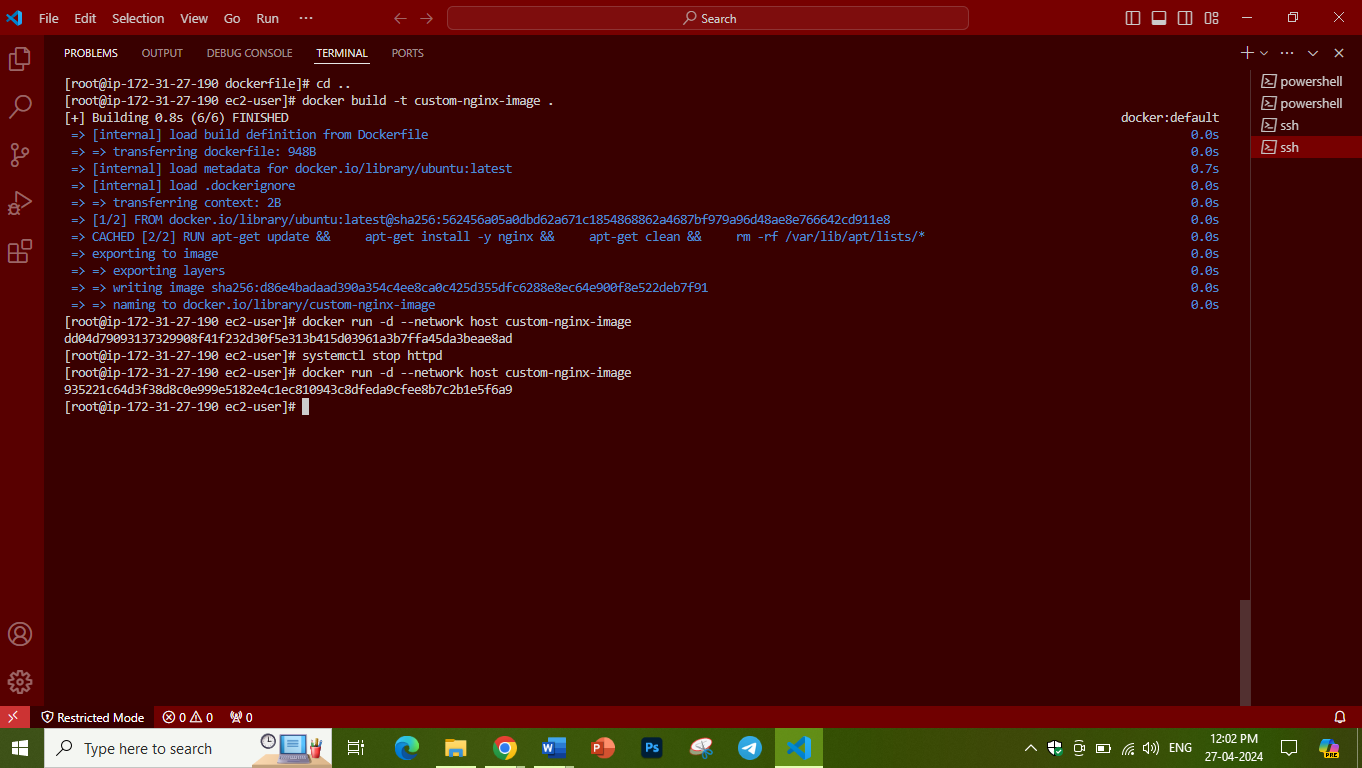
  


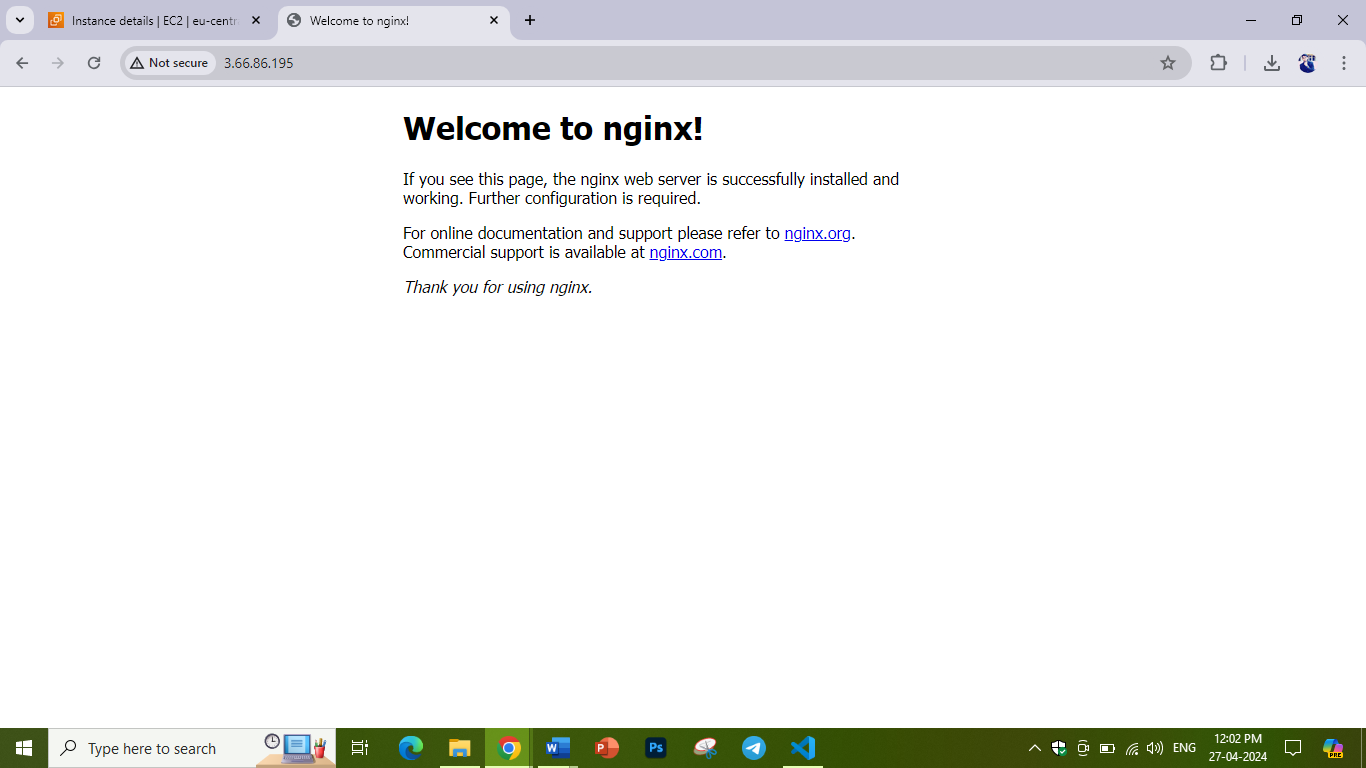
**Assignment 6**

Open putty, connect public IP and run some commands and show the output. Create Empty file, add some content and edit it in Vim.

Upload the final output Screenshot

**Assignment 7**

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  




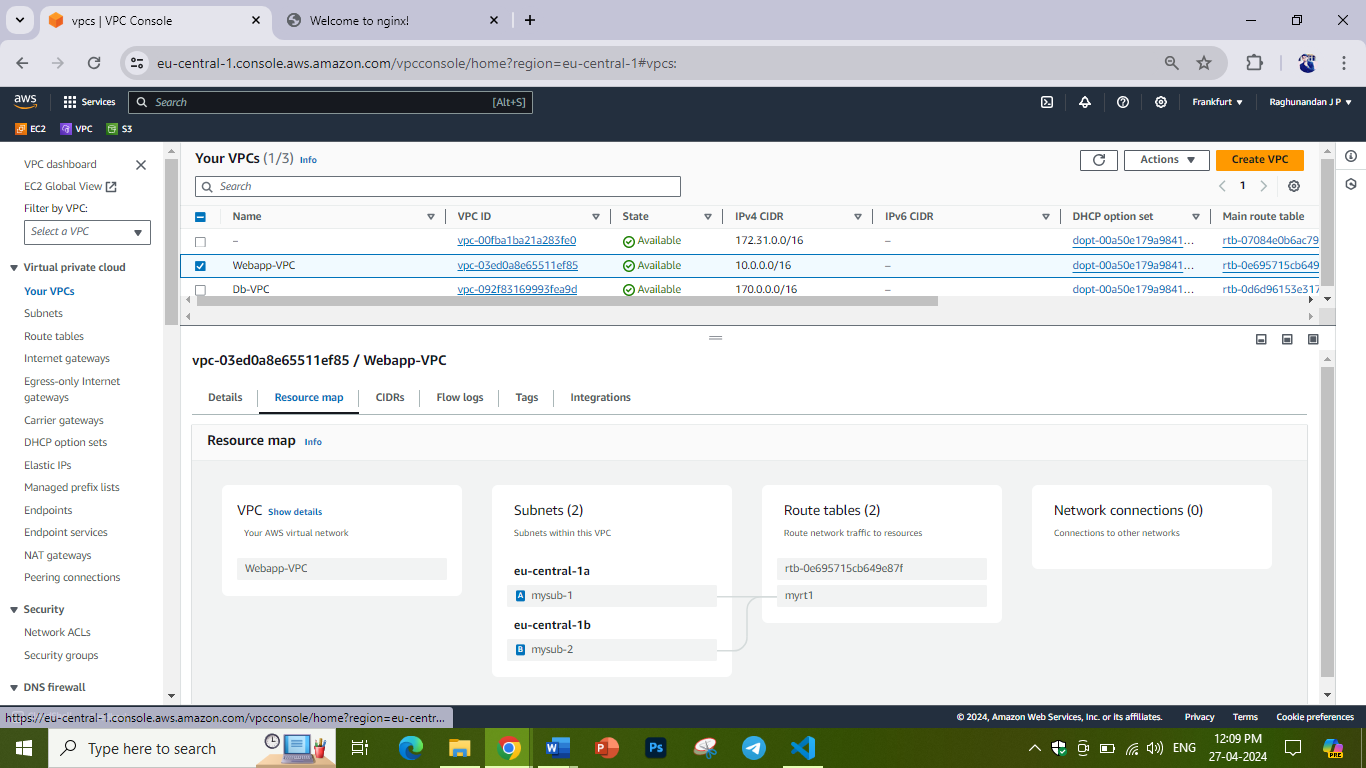
Upload the final output Screenshot

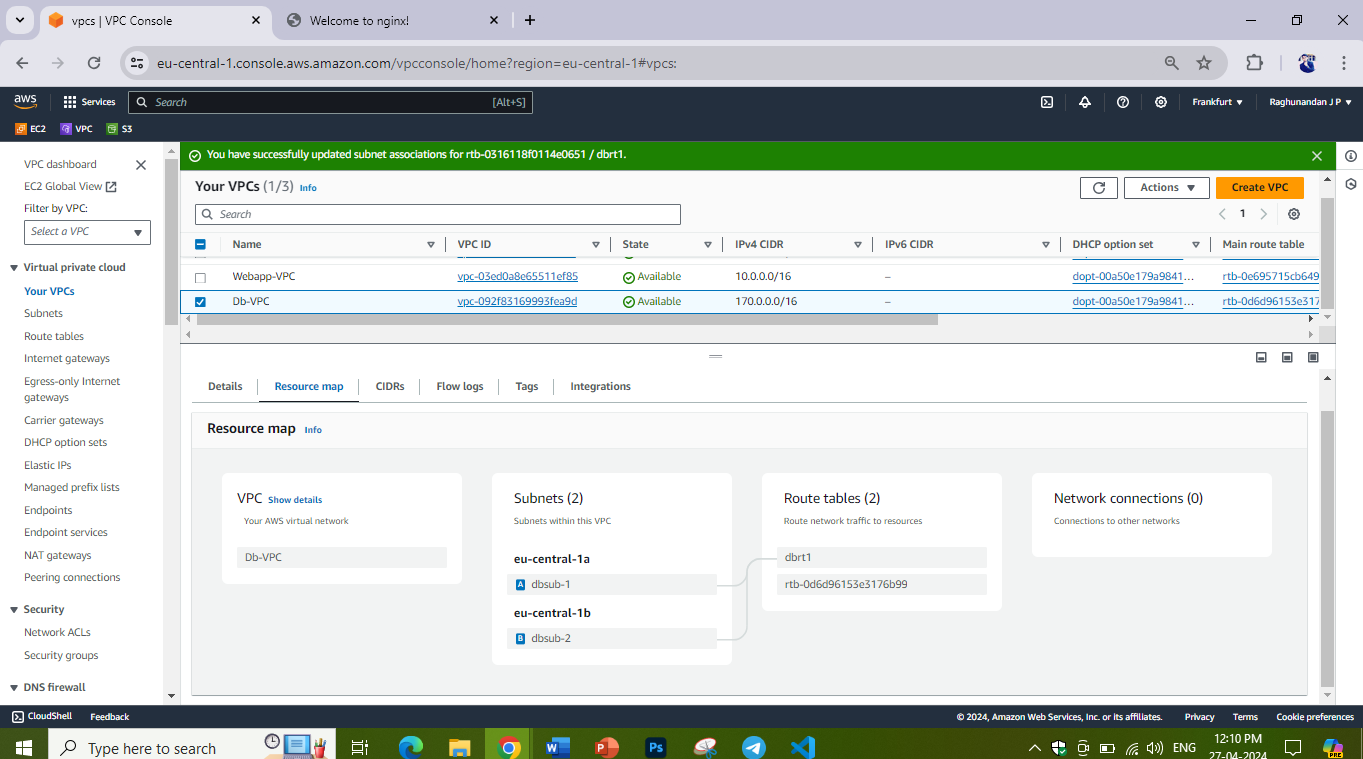
**Assignment 8**

Create 2 VPC's Named "Webapp-VPC" & "Db-VPC"

It should have 2 Subnets each, one with Class A IPv4 CIDR and Class B IPv4 CIDR, and 255 ports in each subnet.

Upload the final output Screenshot

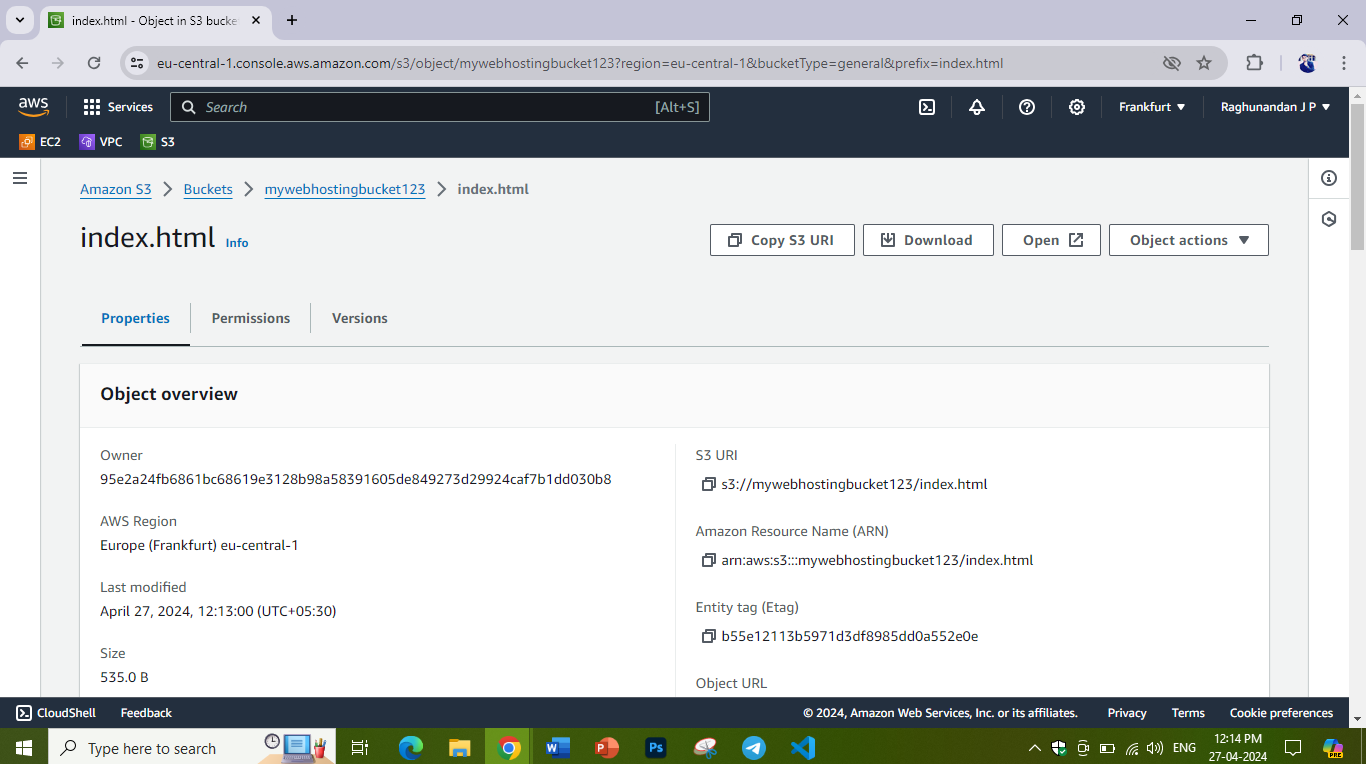


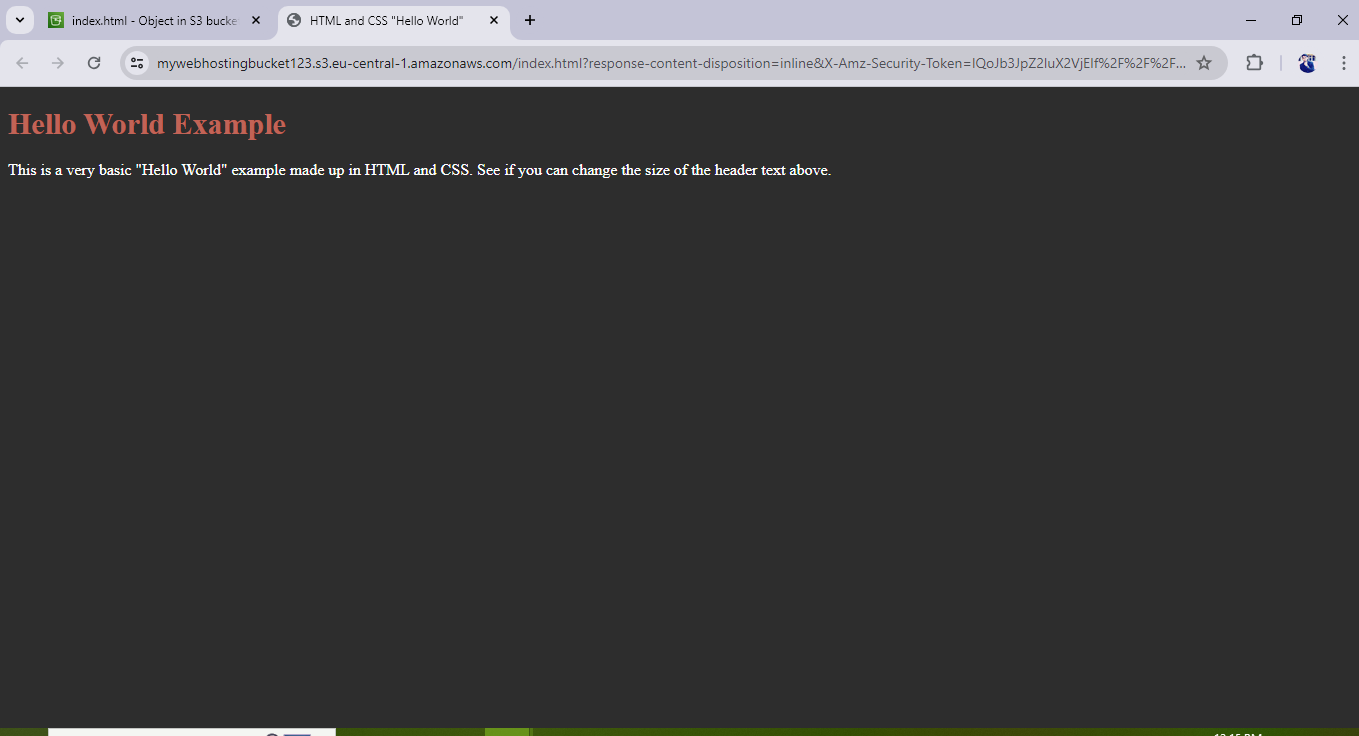


**Assignment 9**

Create and configure an S3 bucket: Create a new Amazon Simple Storage Service (S3) bucket and configure it for static website hosting. Then, upload a sample HTML file and ensure that it can be accessed through a web browser.

Upload the final output Screenshot





**Assignment 10**

**Creating a Docker Compose Configuration:**

**Write a Docker Compose YAML file (docker-compose.yml) to define a multi-container application setup.**

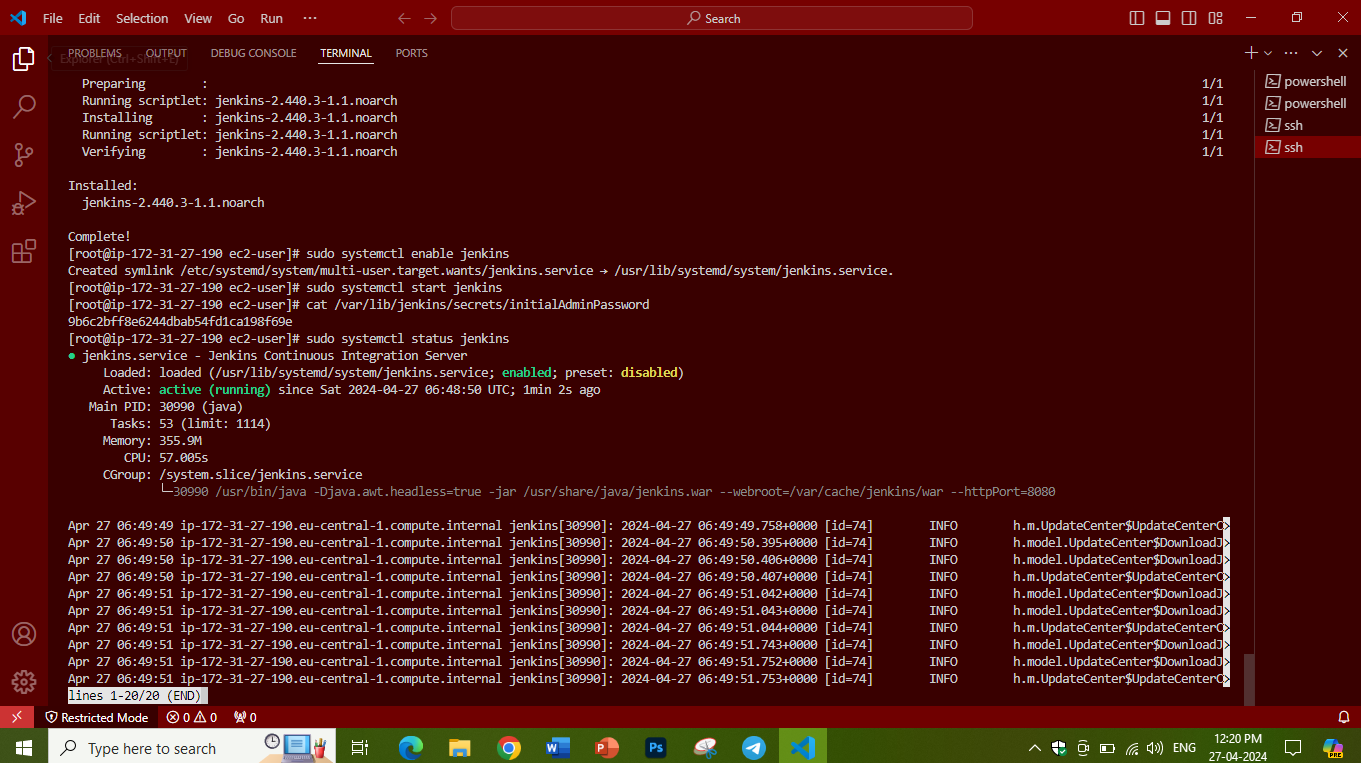
**Specify the services, their configurations, and any necessary links or dependencies between containers.**

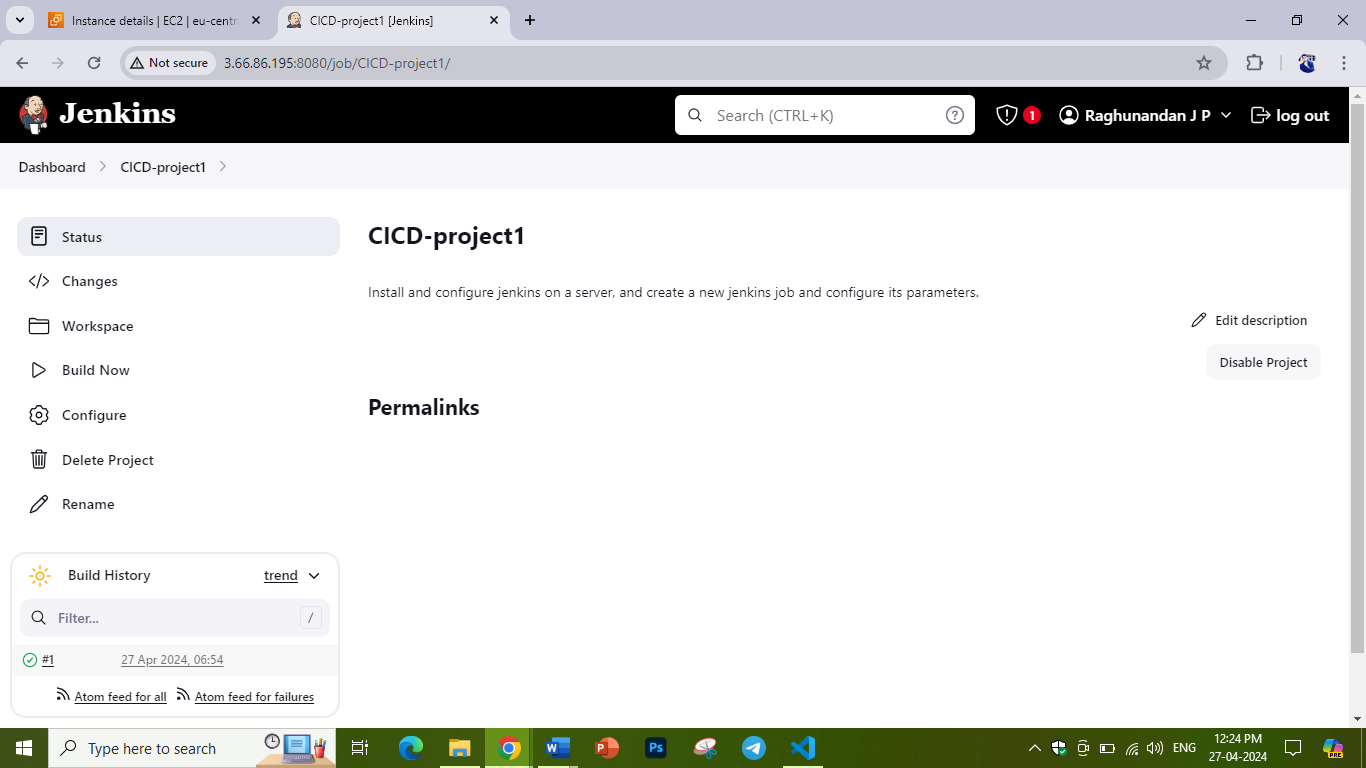
**Use the docker-compose up command to start the containers defined in the Docker**

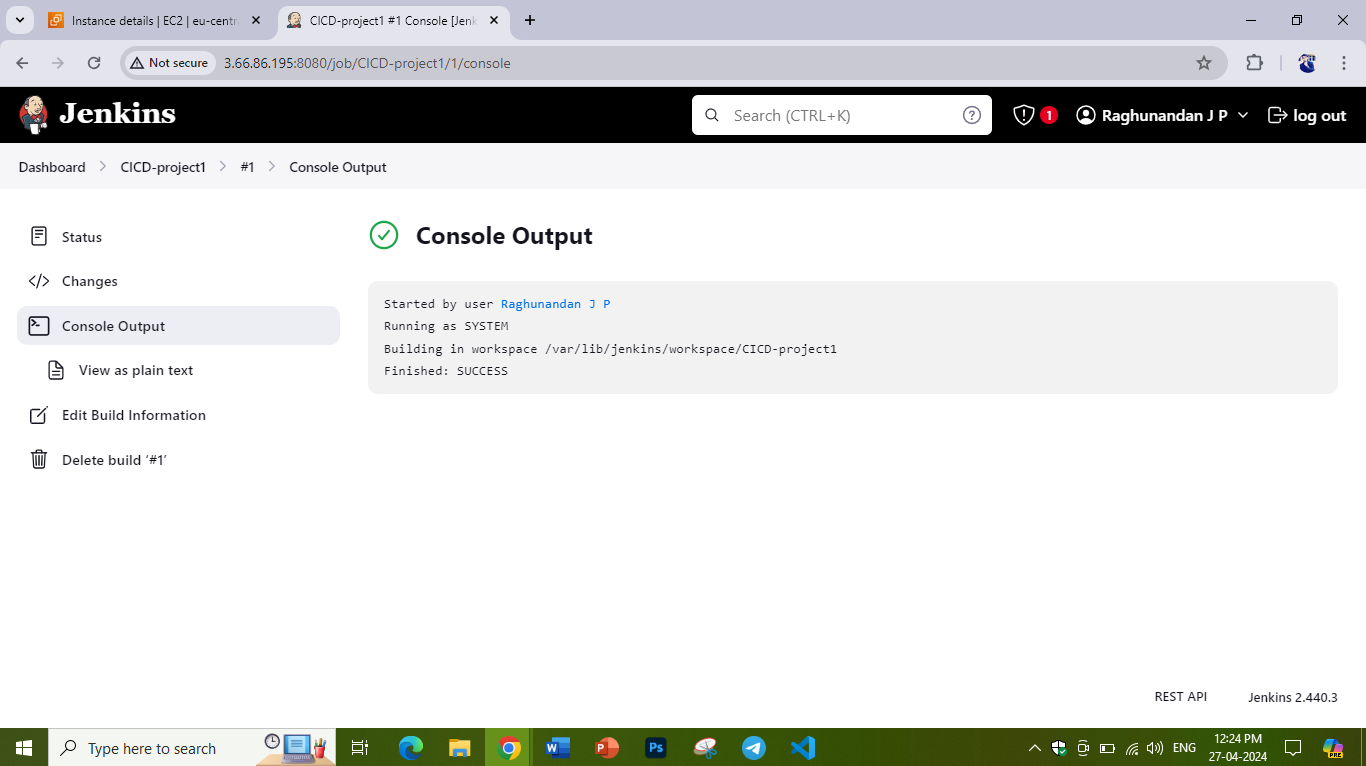
**Upload the final output Screenshot**

**Assignment 11**

**Install and configure jenkins on a server, and create a new jenkins job and configure its parameters.**







**Assignment 12**

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

**Assignment 13**

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 14**

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

Upload the final output Screenshot

**Assignment 15**

Scenario:A user reports that they are unable to access a website by its domain name, but other websites are working fine. What steps would you take to troubleshoot and resolve this issue from the DNS perspective?

Explain in 300 words

When a user cannot access a website by its domain name while other websites are working fine, it could indicate a DNS-related issue. Initially, we need to check the DNS connection and its configuration. Check for any cached data present in the browser, clear all cookies and cache data, and then check the connection. If it fails again, check with another DNS domain to verify whether the other DNS is working fine or not. If the issue still persists, we need to contact the DNS provider for further assistance.