Dear Candidate,   
  
  
Please find the online tasks for the shortlisting process. Read the following instructions carefully and complete the task within a given period of time.

* Complete minimum 10 tasks to get shortlisted for interview round
* This round will test your practical hands on skills w.r.t AWS and DevOps.
* Idea is to test your strengths rather than find your weaknesses. You can try any of the tasks. Need not be in any particular order.
* Once the task is completed kindly upload a final output screenshot after each question.
* Complete the task before 1 PM. No task will be accepted once the time is over.
* Only one task can be submitted by one person, so before submitting please review your task.

How to submit the 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

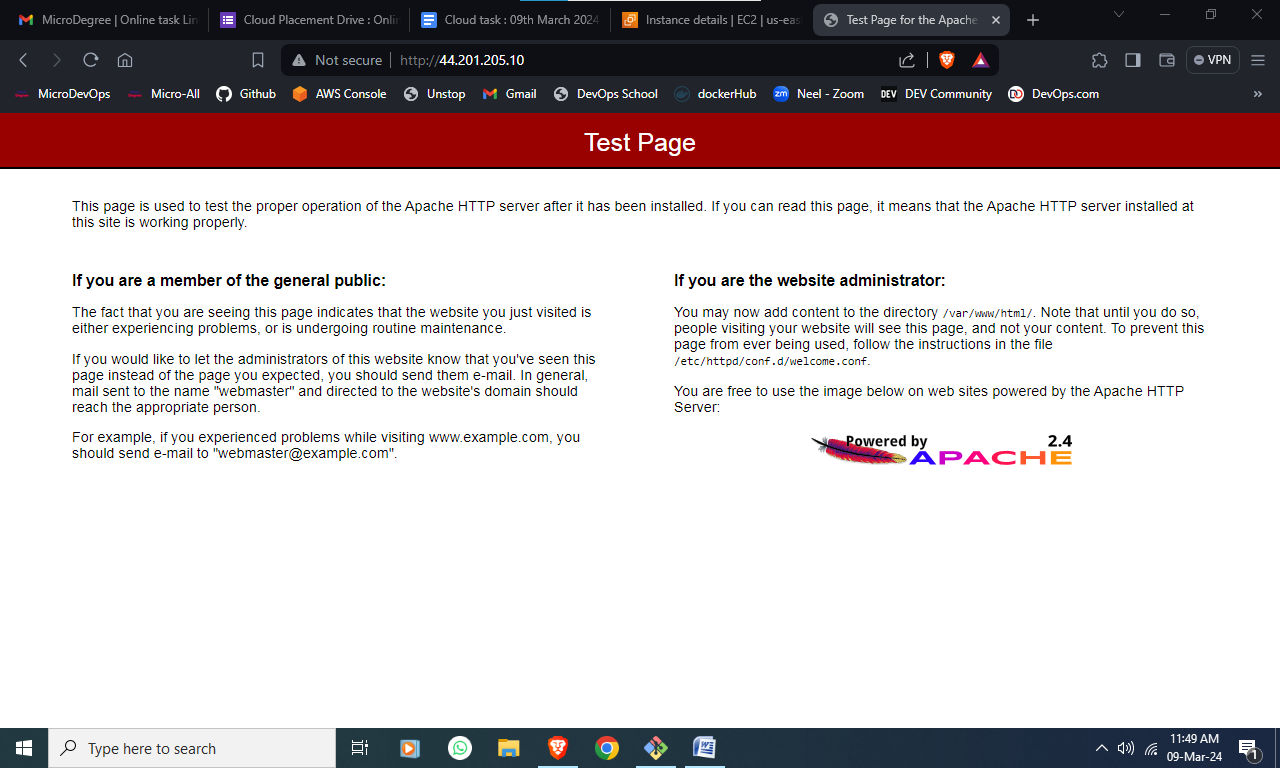
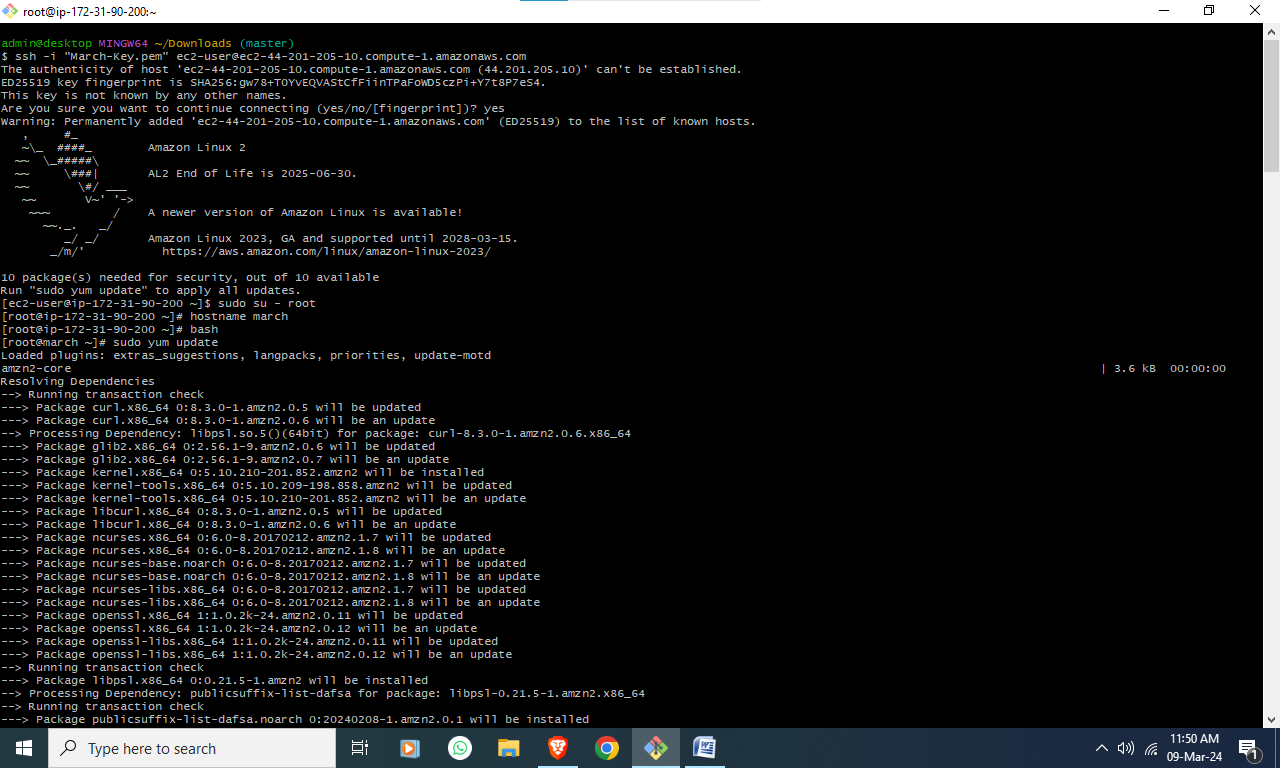
Step 5: Save the document

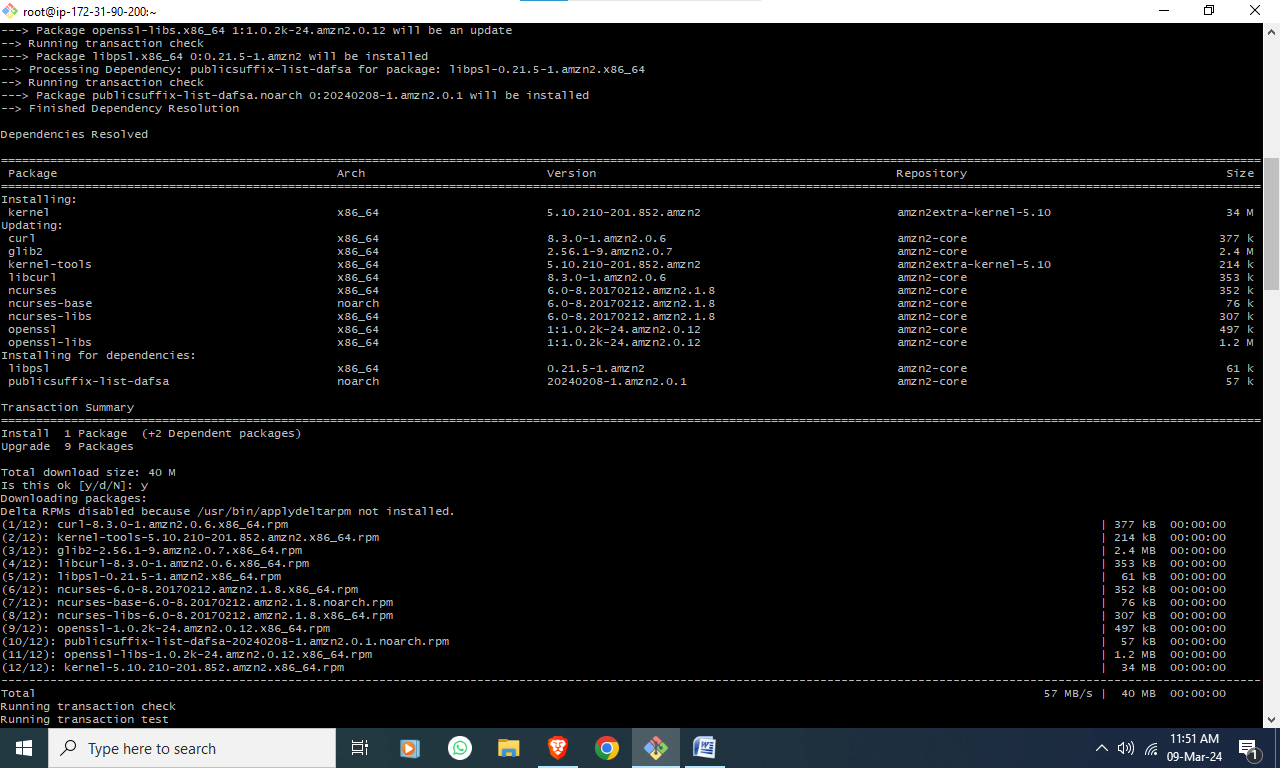
Step 6: Upload the document in Google form

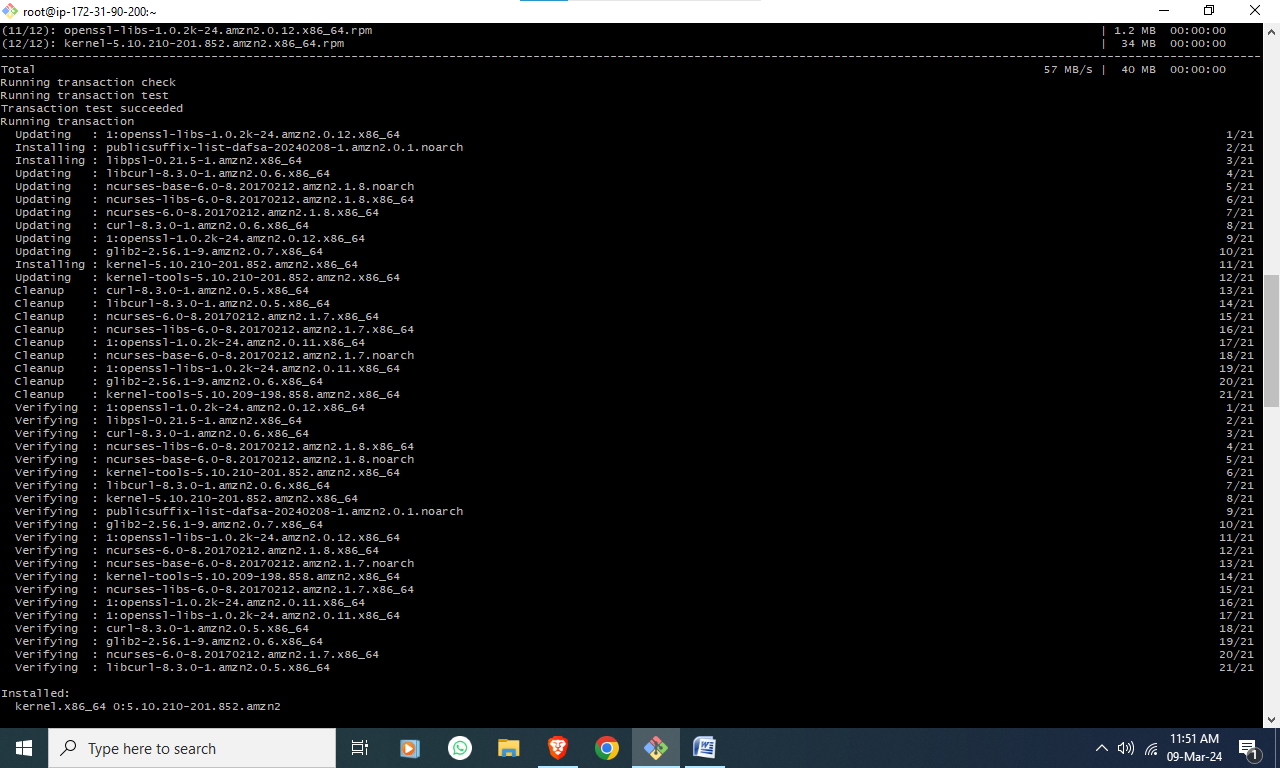
* **AWS Account is mandatory to complete the tasks**.
* **Include AWS account Name in your screenshots**

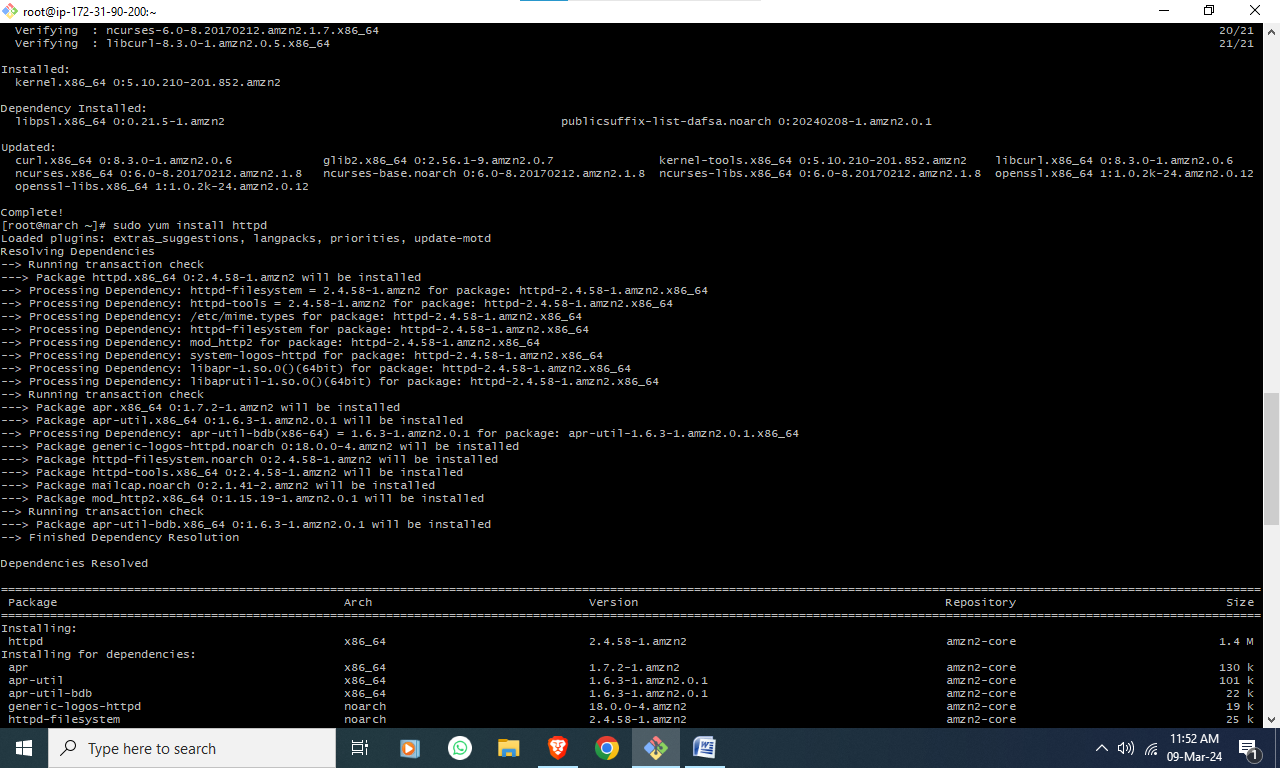
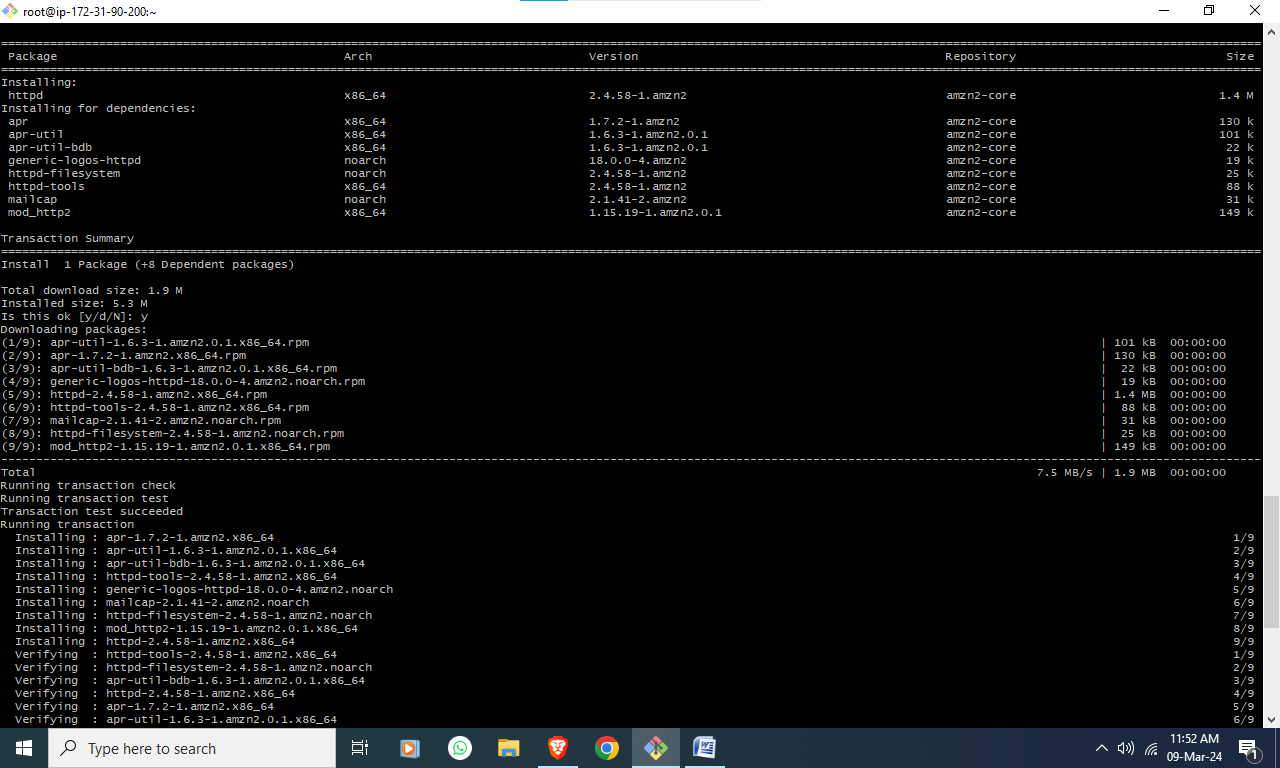
**Resume / Copied and similar tasks will be rejected directly.**

**Task 1** : Configure ec2 linux machine and install apache configuration  
Screenshot:





Installed:

httpd.x86\_64 0:2.4.58-1.amzn2

Dependency Installed:

apr.x86\_64 0:1.7.2-1.amzn2 apr-util.x86\_64 0:1.6.3-1.amzn2.0.1 apr-util-bdb.x86\_64 0:1.6.3-1.amzn2.0.1 generic-logos-httpd.noarch 0:18.0.0-4.amzn2

httpd-filesystem.noarch 0:2.4.58-1.amzn2 httpd-tools.x86\_64 0:2.4.58-1.amzn2 mailcap.noarch 0:2.1.41-2.amzn2 mod\_http2.x86\_64 0:1.15.19-1.amzn2.0.1

Complete!

[root@march ~]# sudo service httpd statrt

The service command supports only basic LSB actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.

[root@march ~]# sudo service httpd start

Redirecting to /bin/systemctl start httpd.service

Task 2 : Associate an Elastic IP to an EC2 Instance using Terraform

- Create an ec2 instance using terraform workflow

- associate an elastic IP

Screenshot

|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

**Task 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.

Screenshot

|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

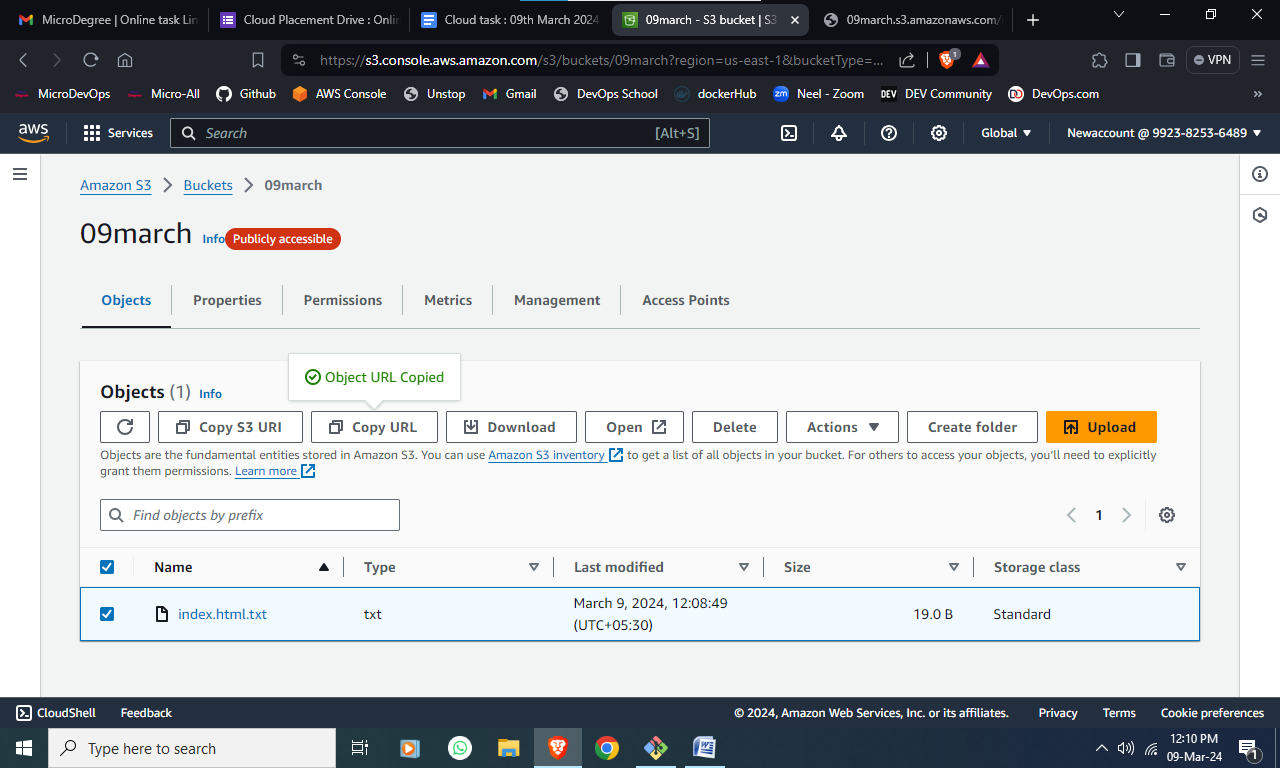
**Task 4: Setting Up Continuous Integration and Deployment (CI/CD):**

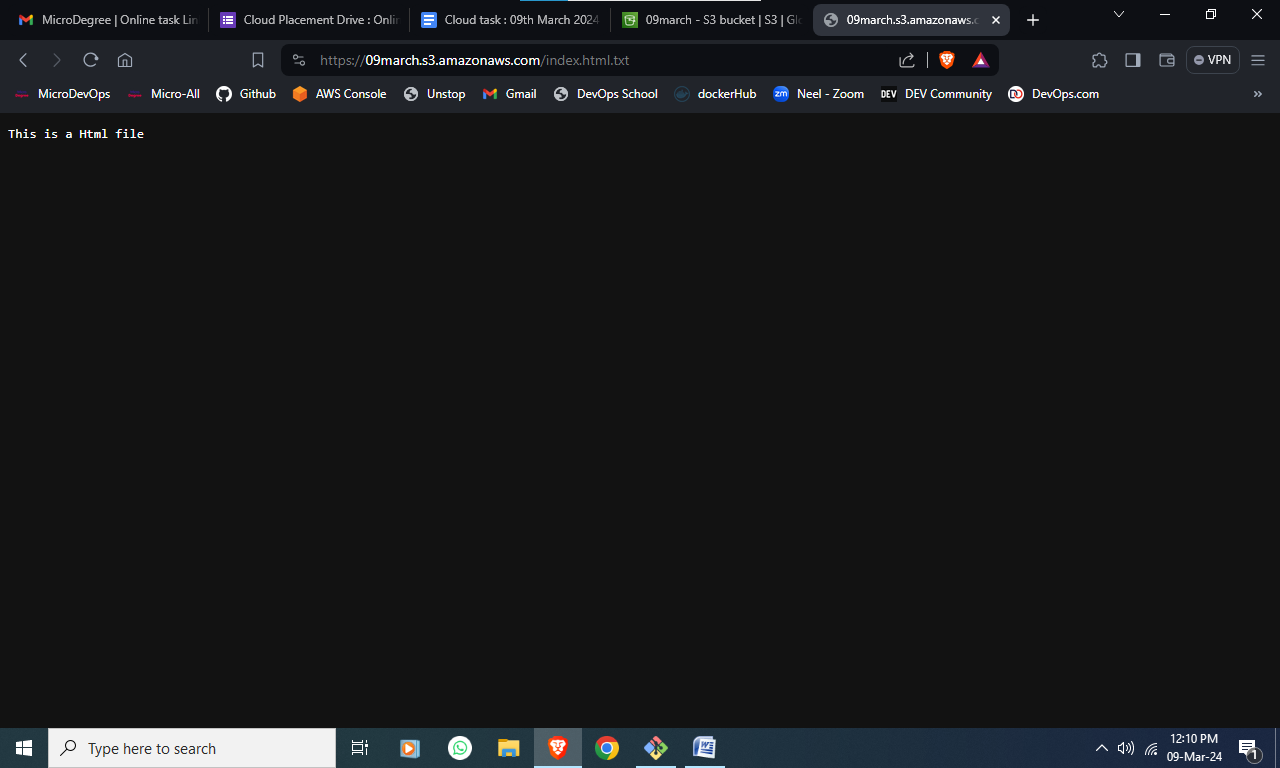
**Jenkins is often used for implementing CI/CD pipelines to automate the build, test, and deployment processes.Create a pipeline job using Jenkins Pipeline DSL (declarative or scripted) or a Jenkins file.**

**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.**

|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

**Task 5: C**reate 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.



  
  
­

**Task 6 :**Host a static website in S3

-> create a s3 bucket with private access ..

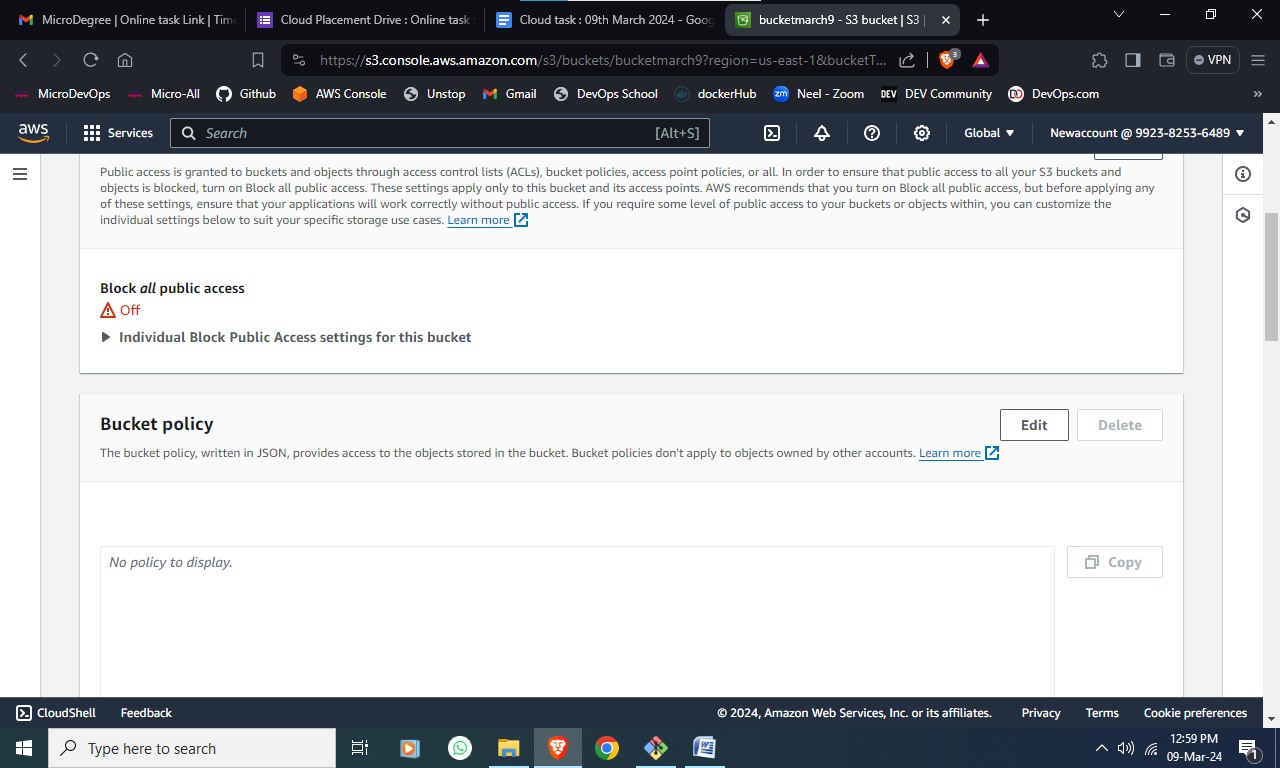
-> Note : bucket should not be Public access

-> ADD the objects -> like front-end code eg: index.html

-> create the Cloud Front (CDN- Content Delivery Network )

-> now you need to Redirect the traffic from Cloud-Front to s3 bucket

-> copy the distribution id of the cloud-front . check in any Browser . website should appear.



|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

**Task 7 :** Working with Docker Images

- Pull the latest `httpd` image.

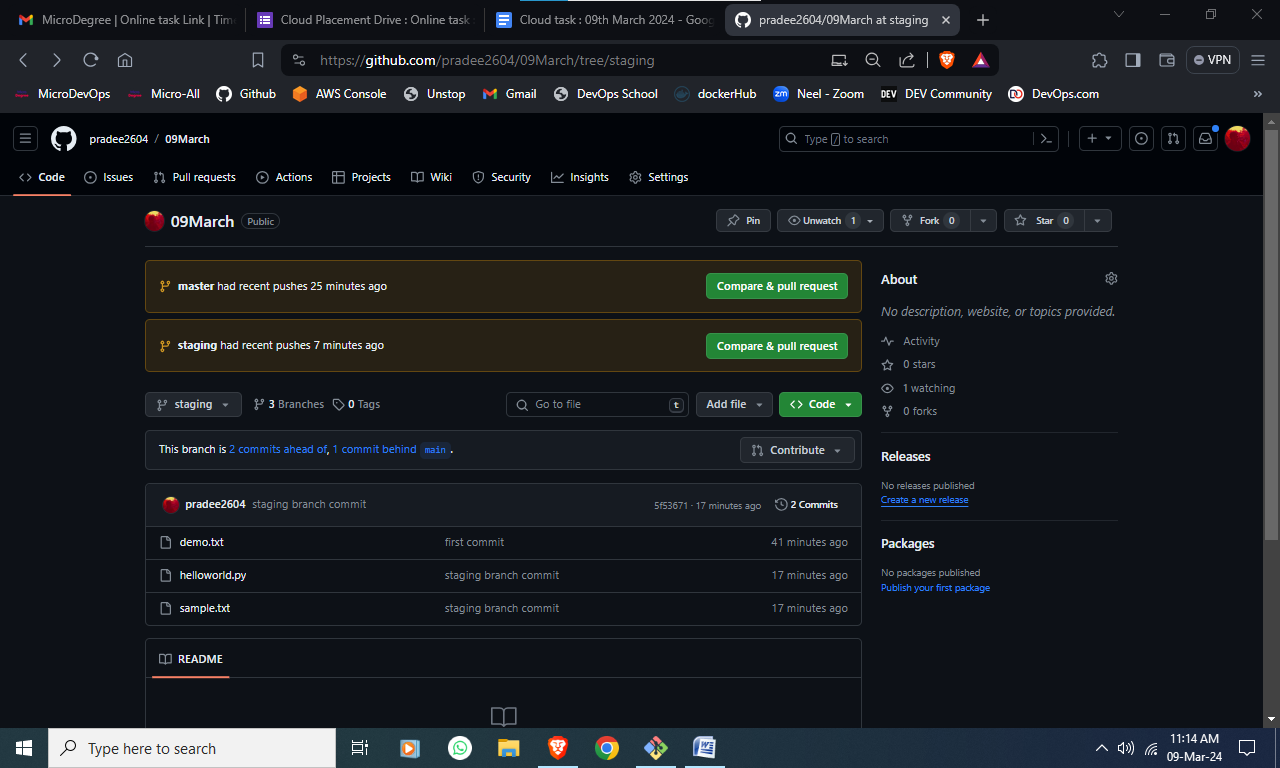
- Pull the latest `alpine` image.

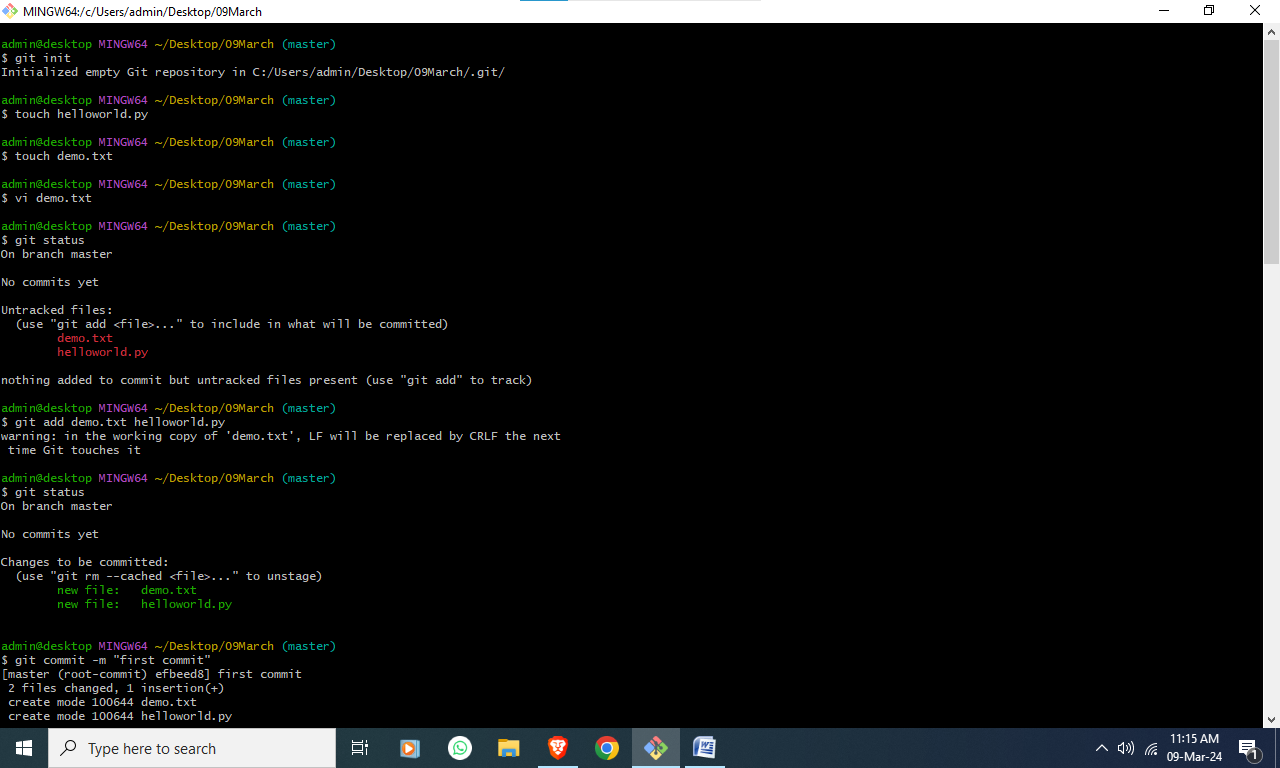
- verify images pulled and create 2 containers in each serverUpload the final output Screenshot

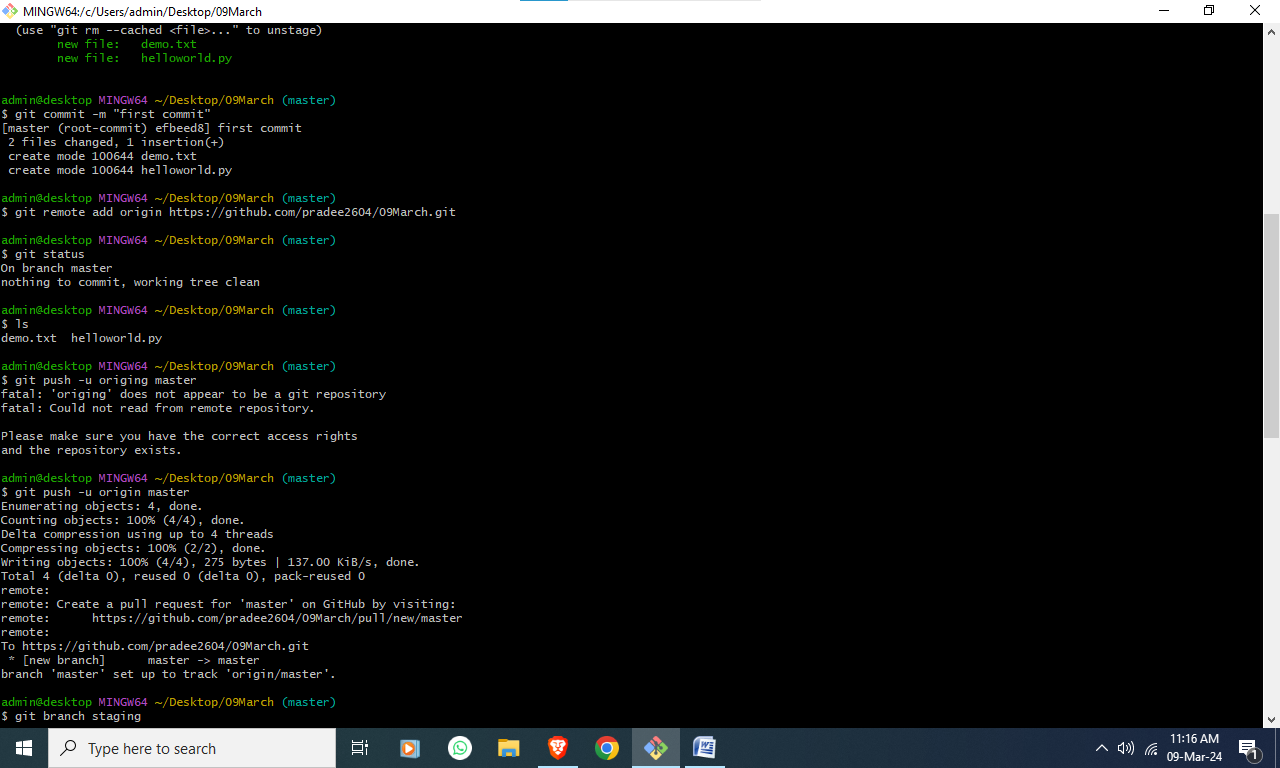
|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

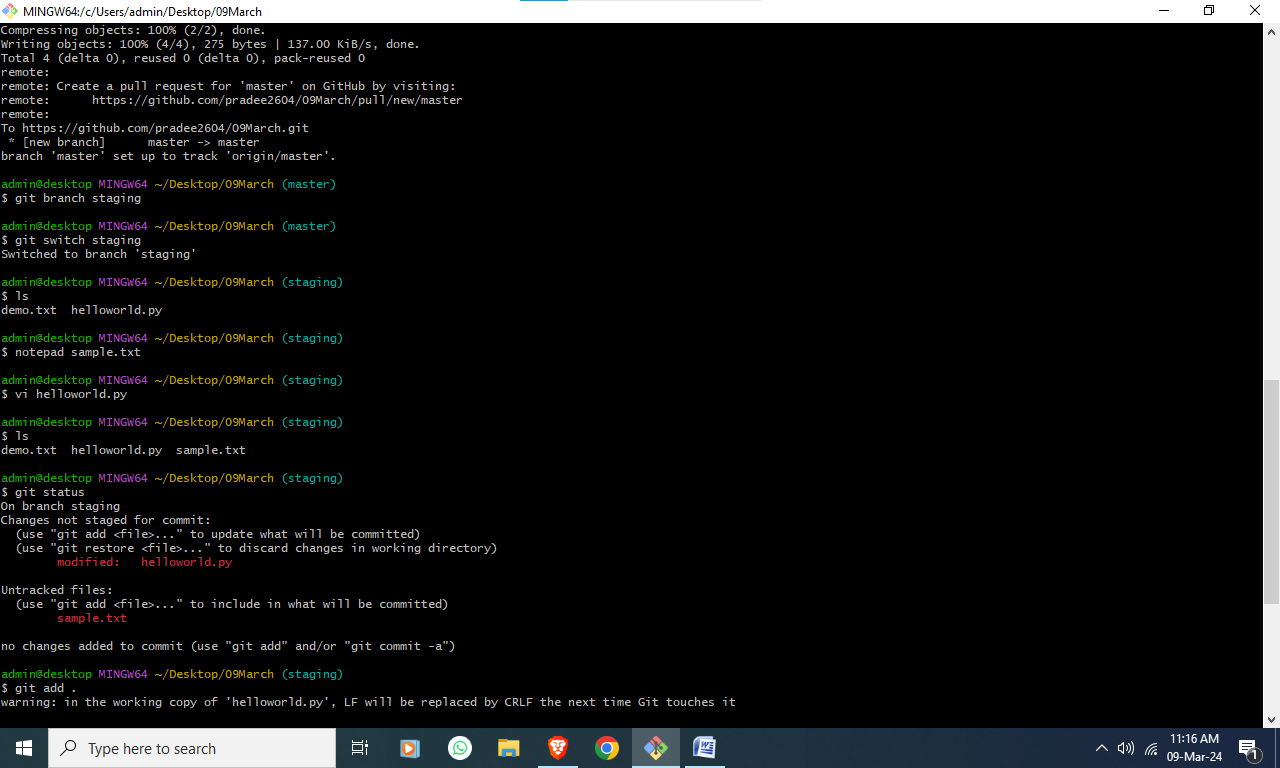
**Task : 8** Configure a Load Balancer: Set up an Elastic Load Balancer (ELB) that distributes incoming traffic to the EC2 instances in your Auto Scaling group. Configure health checks to ensure that the ELB only forwards traffic to healthy instances.  
Screenshot :

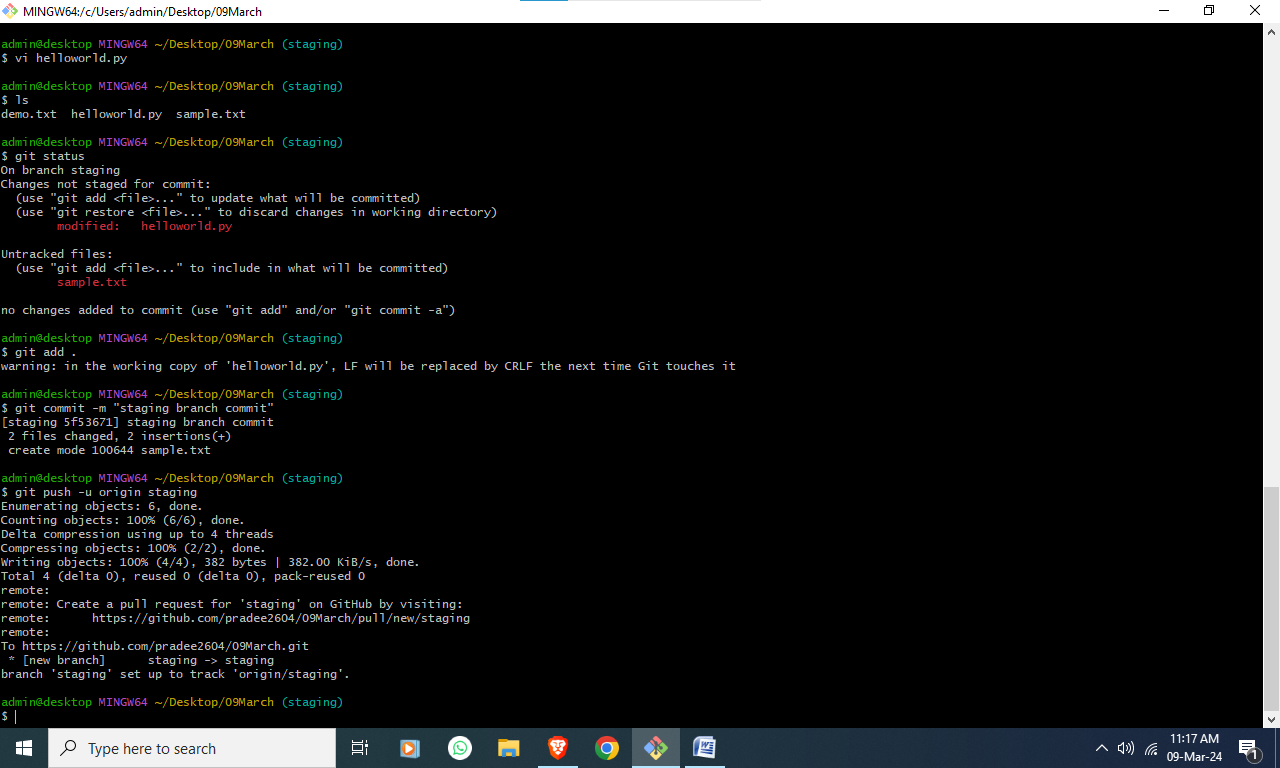
|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

Task 9:  
Create a Staging branch in GitHub and push code from the local repository to the Remote and share the full commands screen  
  
  




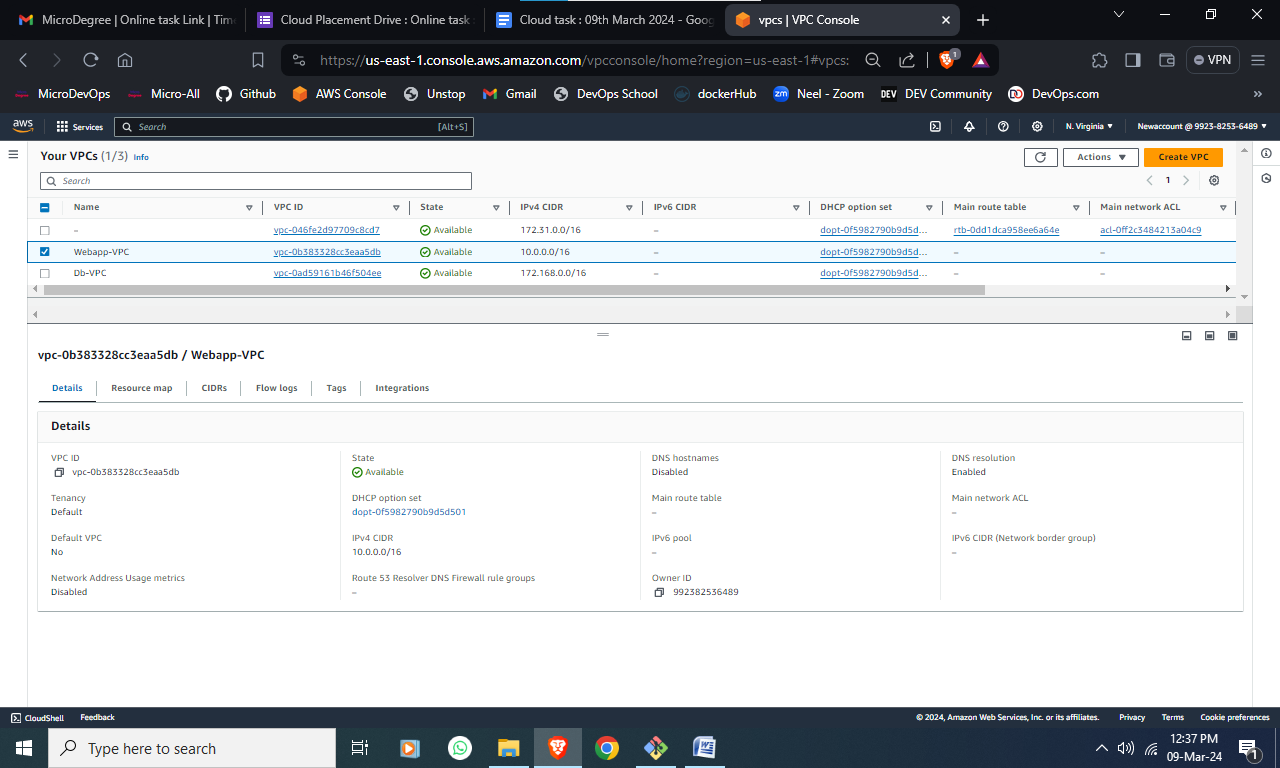


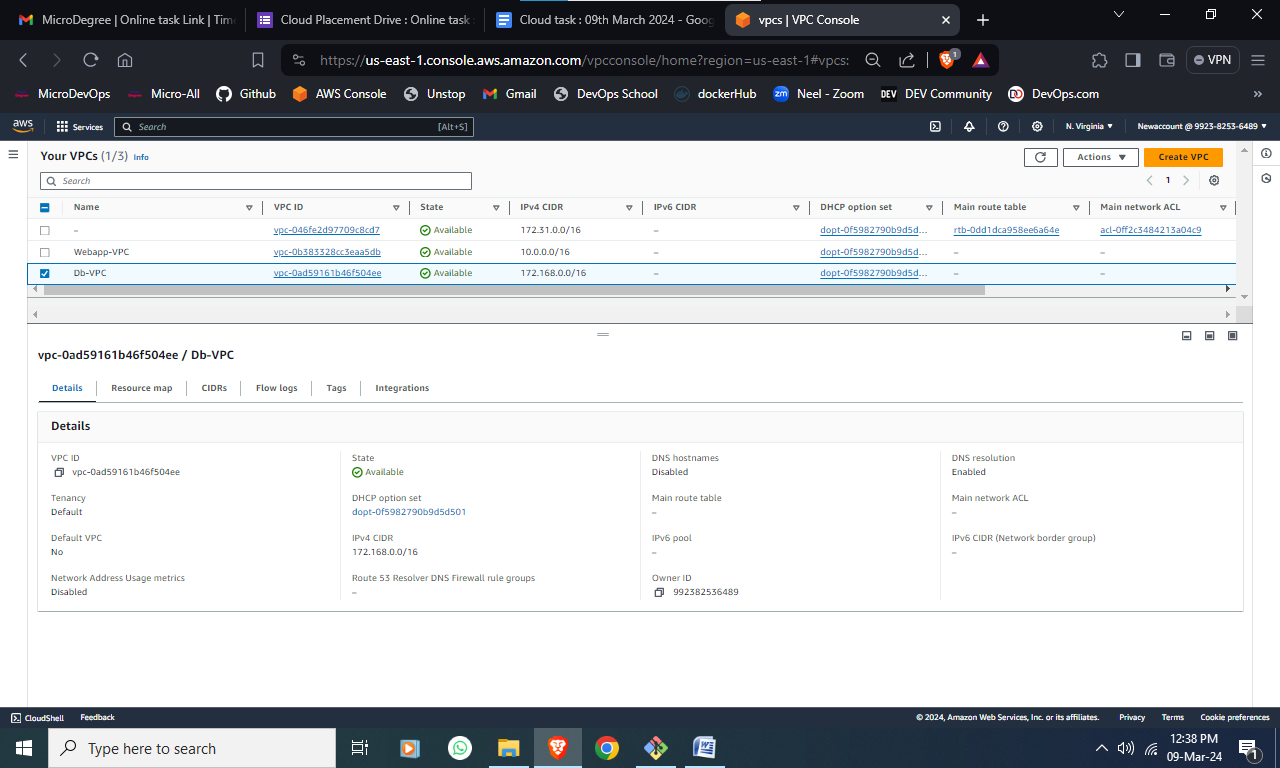


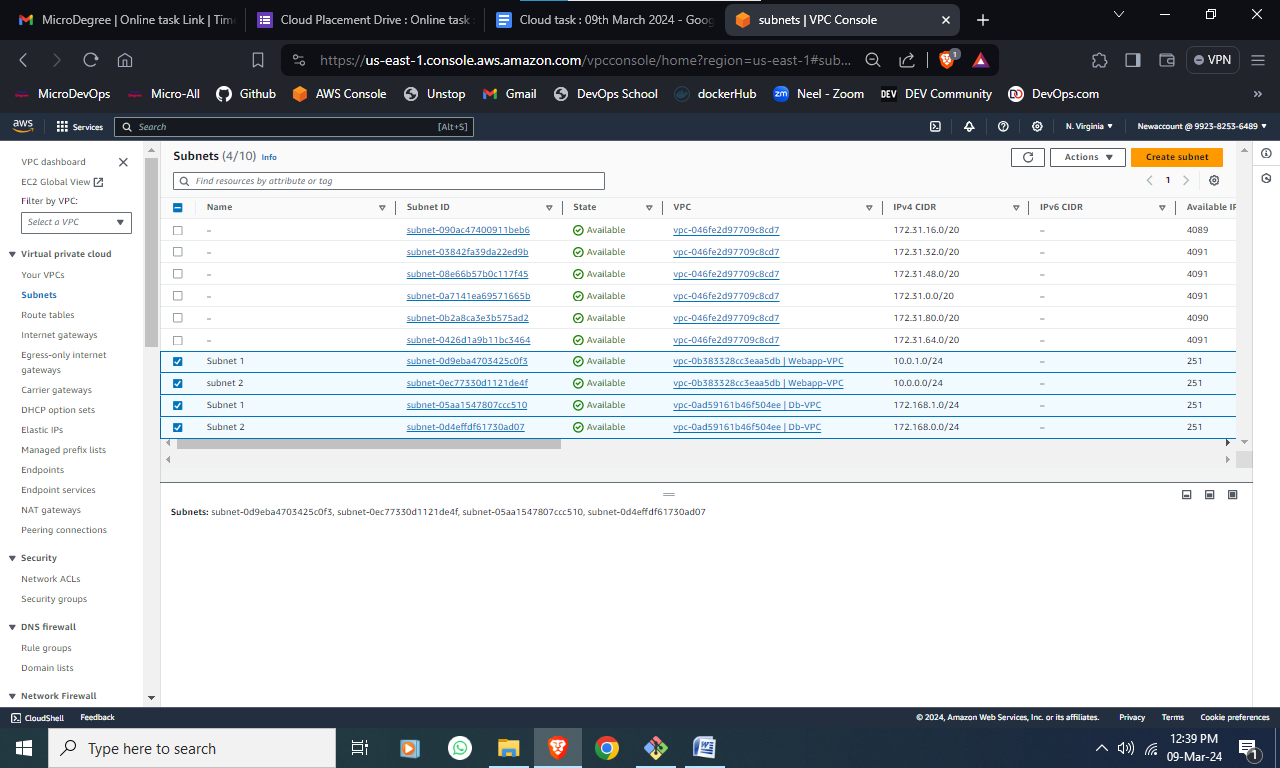


**Task 10:** Create 2 VPC's Named "Webapp-VPC" & "Db-VPC"

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







**Task 11:** 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

Screenshot:

|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

**Task 12:** Create a basic deployment pipeline that deploys your application to different environments, such as development, staging, and production. Implement testing and approval steps for each environment.

Screenshot:

|  |
| --- |
| Insert >> Image >> Upload from computer >> Upload task result screenshot |

**Scenario based questions:**

**Task 13** :Scenario: You have a running process that is consuming a lot of system resources and needs to be terminated. How would you find the process ID (PID) of the process and terminate it gracefully?  
  
Explain in 300 words

Task 14: Scenario: You are using a computer with a single-core processor. You have multiple applications open, including a web browser, a text editor, and a media player. The media player freezes and becomes unresponsive. Explain the potential reasons behind this issue and suggest a possible solution.

Single processor will not work if the load is more . it will work very slowly and sometimes it will not work. it's like playing a big game in a small laptops which has only basic features.   
so we should have proper resource else we should make the present machine by upgrading.  
in worst condition we can use the application by closing all other applications and providing all necessary things which are required for the application to run.  
  
also we can run the application using virtual machines and using the image file which is available in docker container   
  
with these steps somehow we can run the application   
still upgrading the machine or high-end machines performance is still better.   
  
  
**Task15:** 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