**Akhila Sulake**

**Linux:**   
**1. Provide steps to create a directory inside a directory where the parent directory does not exist.**

**.**mkdir -p

mkdir -p /path/to/parent-dir/sub-dir

By using this command, you can create a directory inside a directory that doesn't exist yet, and the parent directory will be created automatically.

**2. How to install a package on a Linux server when there is no internet connection?**

Download the package (.deb or .rpm) on a machine with internet access, then transfer it to the Linux server and install it using the package manager (e.g., dpkg or rpm).  
**3. How to access specific folders of Server A from Server B and Server C?**

use NFS (Network File System) sharing: On Server A, export the specific folders using NFS, and on Server B and Server C, mount the exported folders using NFS.

**4. How to check all the running processes from a server?**

ps -ef  
**5. Provide the command to delete all the files older than X days inside a specific directory.**

find /path/to/directory -type f -mtime +X -delete

example -find/downloads/keys -type f -mtime +10 -delete (from local machine)

**6. Create a shell script to identify the process ID a. script should as a user input for process ID b. If the process exists, the script should print the process ID and exit c. If the process doesn't exist script should print the process doesn't exist and asks for another input**

**bash**

#!/bin/bash

while true

do

read -p "Enter process ID: " pid

if [ -n "$pid" ]; then

if kill -0 $pid > /dev/null 2>&1; then

echo "Process ID $pid exists"

exit 0

else

echo "Process $pid does not exist"

fi

else

echo "Please enter a process ID"

fi

done

**I copied this from google, but I can analyse the bash and shell script and I can write basic shell scripts.**

**Docker**

2.What is docker and why do we need it?

. Docker is about packing and shipping application in container, once the application is in a container you can easily share it, like docker images. container it can be started and stopped by using docker commands.

. if we want to deploy our application without docker we would have to worry about installing all the dependencies and configuring the environment on the server. But with Docker, you can package the application and its dependencies into a container, and then deploy that container on the server. The container will have everything it needs to run the application, and it will run the same way it did on your laptop.

Docker makes it easy to develop, test, and deploy applications, and it ensures that the application will run consistently across different environments.

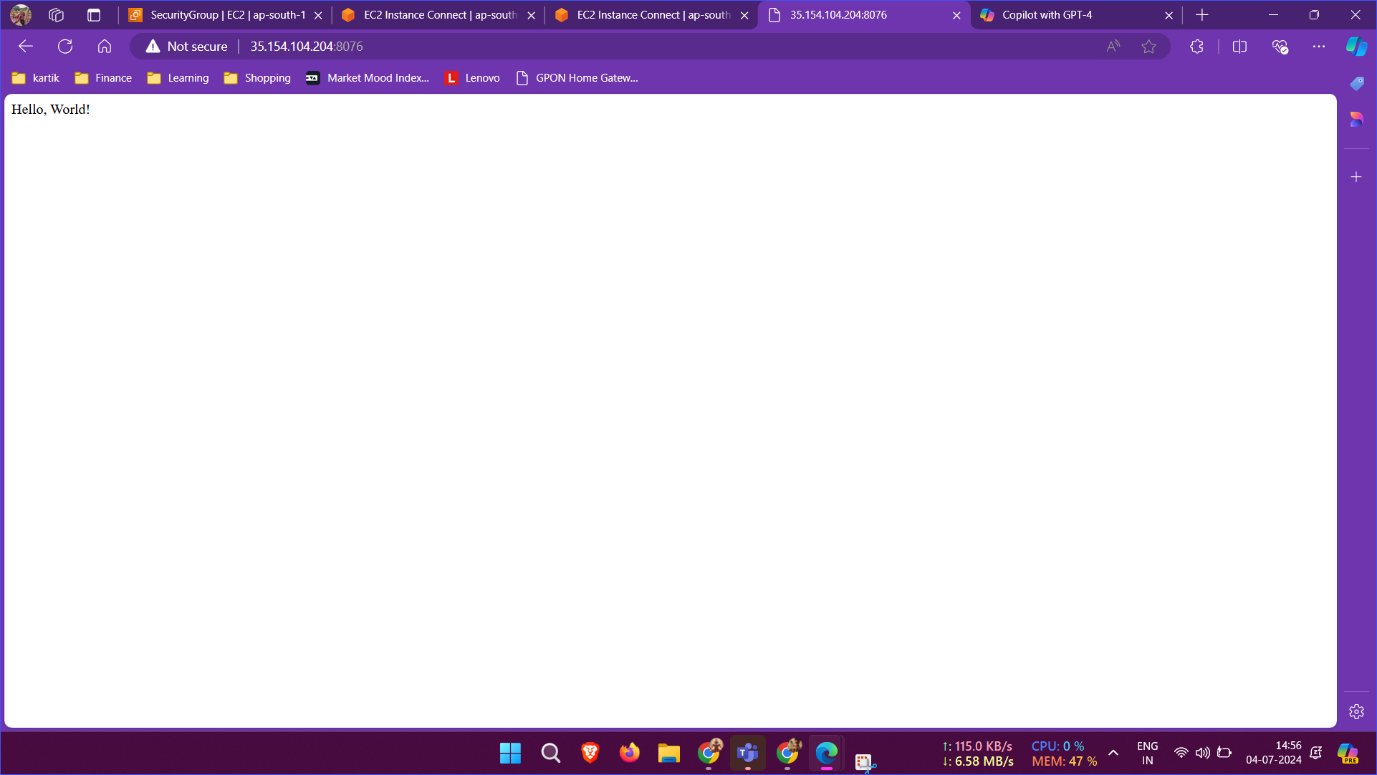
2. Write a docker file for a sample Java/python application.

FROM python:3.9

COPY app.py /app/

WORKDIR /app

CMD ["python", "app.py"]



Executed command –to create image ---docker build -t apiwiz .

Executed command ---docker run -it apiwiz

Docker run -d -p 8076:8080 apiwiz

Docker images to check **docker images** **docker ps** to check the running containers **docker ps -a**

**To check all the containers**

**3. What is the docker lifecycle?**

Docker file ------------create a docker image--------docker container----Running a container-----docker pause----docker stop

. Create: A container is created from a Docker image.

. Running: The container is started and executes the commands specified in the image.

. Pause: The container's execution is temporarily halted.

. Stop: The container is shut down and its execution is terminated.

. Delete: The container is removed from the system, freeing up resources

**4.What is the difference between an image and a container?**

Docker image is read only template

Docker container is run time instance of image

Docker image is a lightweight, standalone, and executable package that includes everything an application needs to run, such as code, libraries, dependencies, and settings.

Docker images are immutable, meaning they cannot be changed once created.

Docker Containers are created from images and can be started, stopped, paused, and deleted.

Docker Containers are mutable, meaning they can be modified while running.

5.How to check docker container logs? Provide the command for the same

.docker logs my\_container-id or name

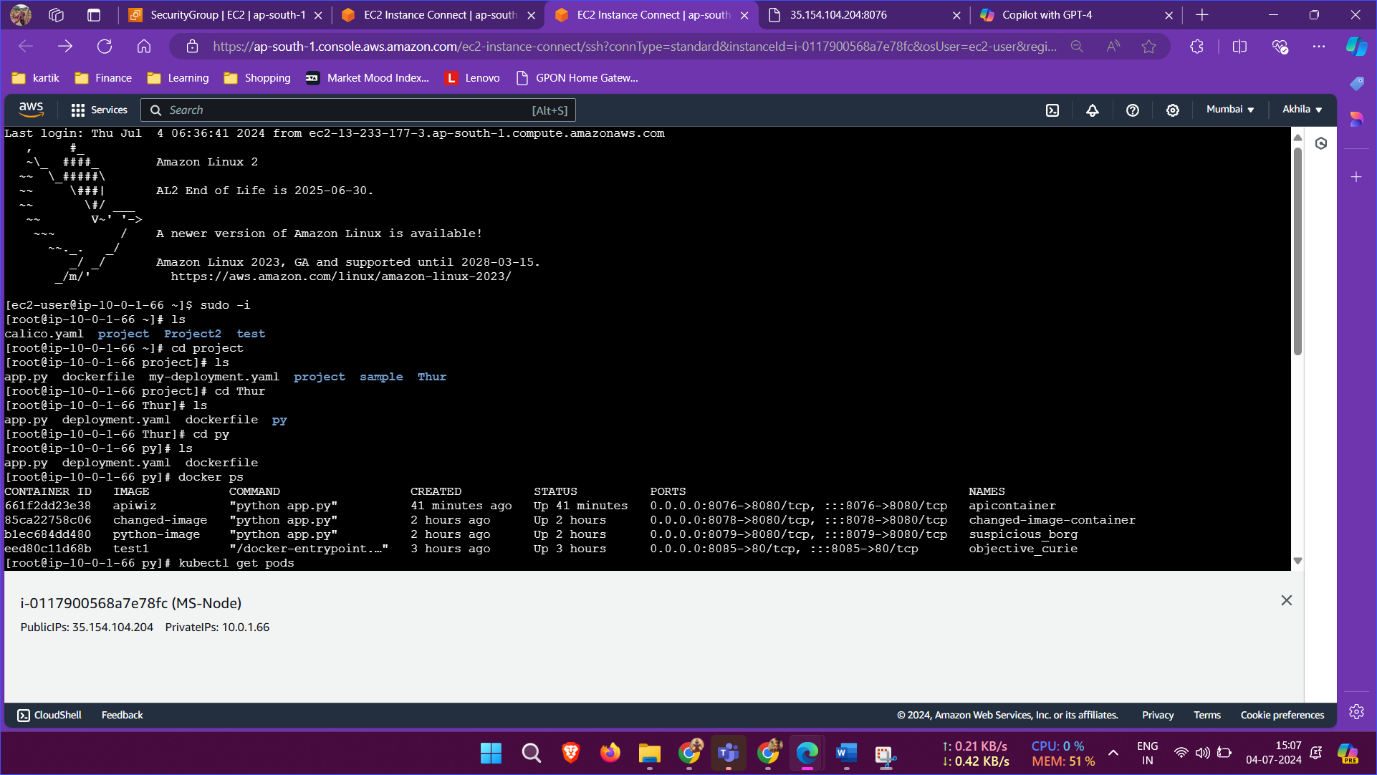
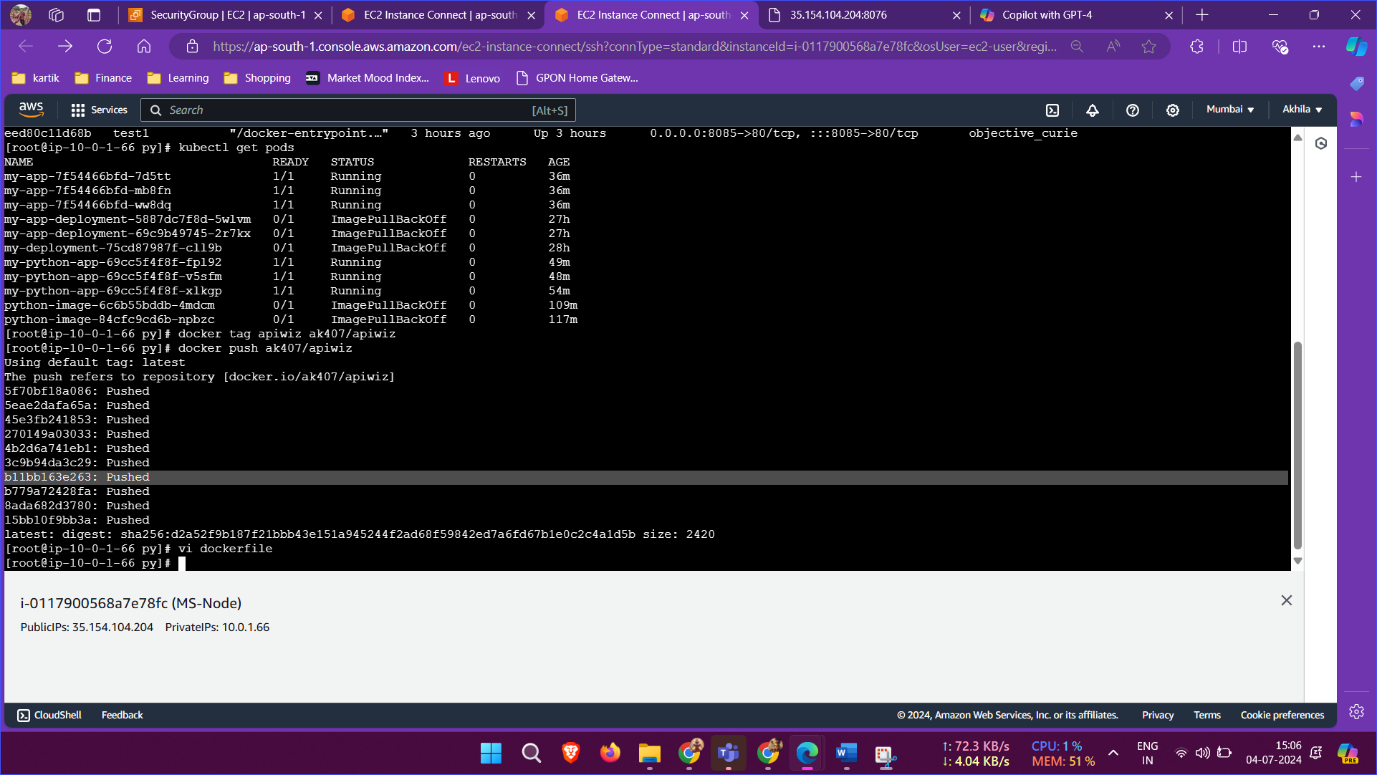
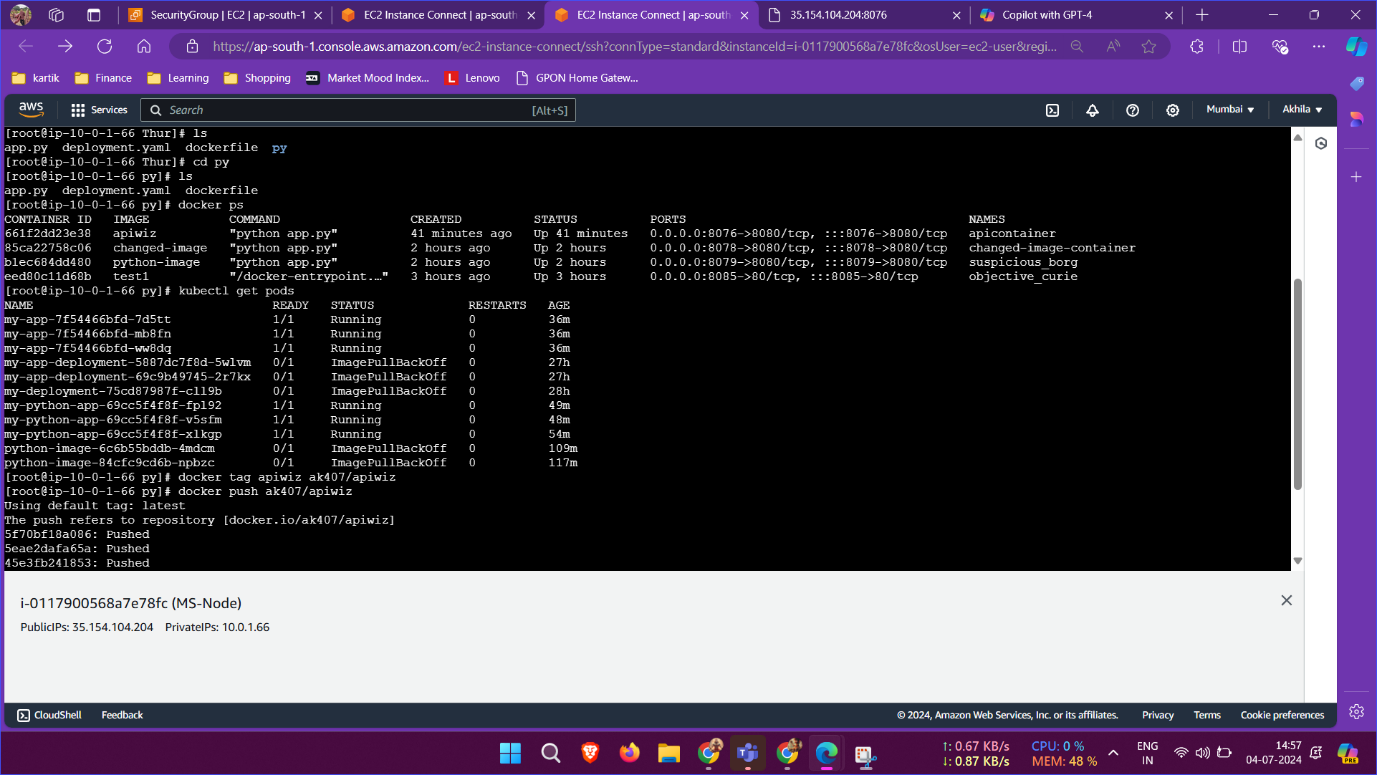
**3. Kubernetes:**   
1. What are different types of services?

Ingress: This is not a Service type, but it acts as the entry point for your cluster.

Gateway: Provides extra capabilities beyond Ingress and Service.

**2. What is a pod?**

Pods are smallest units of execution in k8s that contains one or more containers each with it one or more application and its binaries.

**3. Create a pod with the above created custom image when a pod dies k8s should automatically restart **

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-app

spec:

replicas: 3

selector:

matchLabels:

app: my-app

template:

metadata:

labels:

app: my-app

spec:

containers:

- name: python-container

image: apiwiz

ports:

- containerPort: 8080Tested by deleting pod kubectl delete pod <Podnamewith ID>

It automaticalled restarted

**Applies kubectl apply -f deployment.yaml**

**Kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**my-app-7f54466bfd-7d5tt 1/1 Running 0 56m**

**my-app-7f54466bfd-mb8fn 1/1 Running 0 56m**

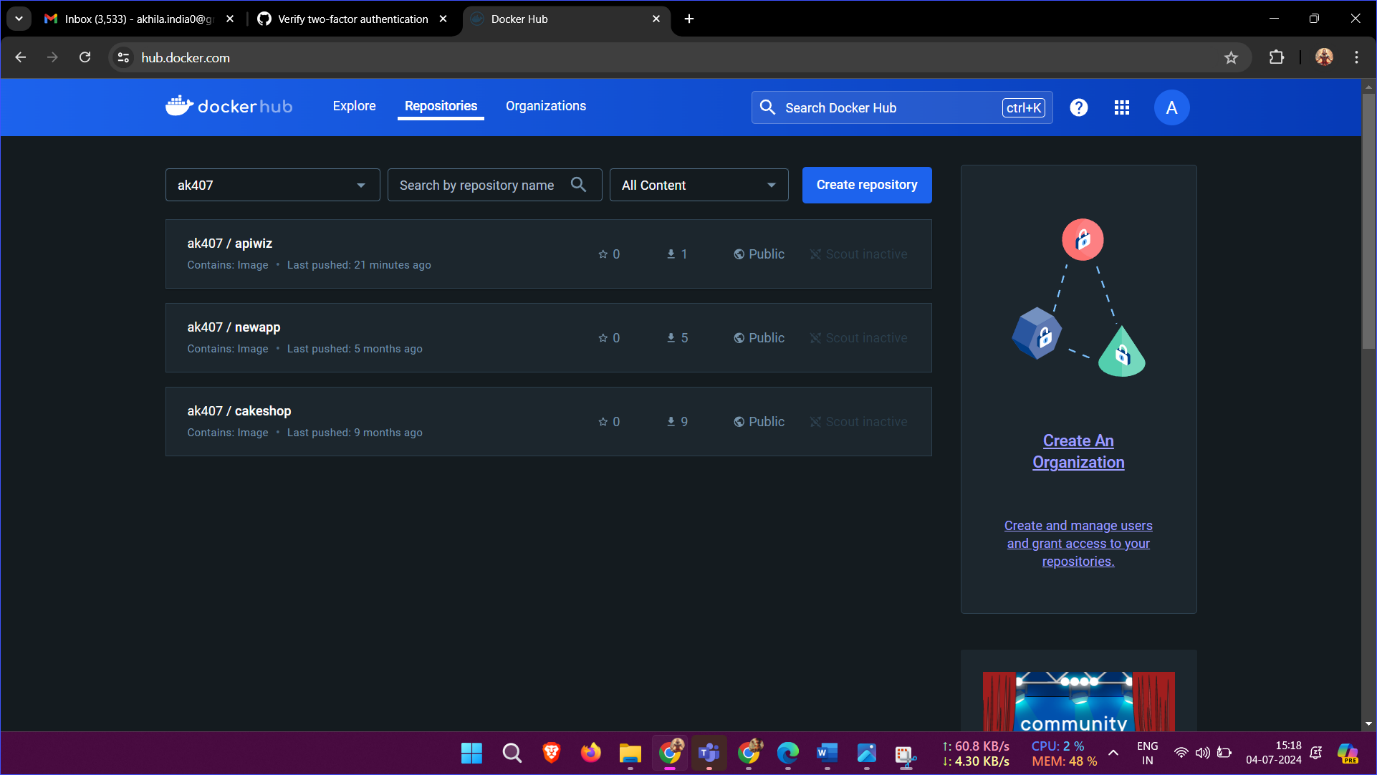
**my-app-7f54466bfd-ww8dq 1/1 Running 0 56m**

**login to the docker account by credentials**

**docker tag apiwiz al407/apiwiz**

**docker push ak407/apiwiz**

**docker pull ak407/apiwiz from dockerhub**



**4.** **How to access the custom application with a specific port?**

By writing the manifesting file ,

1. Expose the application as a Service

2. Specify the port number in the Service definition

3. Use a Service type that allows external access (LoadBalancer or NodePort)

4. Get the Service URL or node IP address and NodePort number

5. Use the Service URL or node IP address and NodePort number to access the application

apiVersion: v1

kind: Service

metadata:

name: my-app

spec:

selector:

app: my-app

port: 80

targetPort: 8080

type: LoadBalancer

**4. CI/CD:**   
1. Set up a pipeline (Github actions/Gitlab runner/ Jenkins or any open source tool) to build, test, create a docker image, publish and deploy to k8s. Use the application present in this public repo <https://github.com/apiwizlabs/wizdesk>.

pipeline {

agent any

stages {

stage('Git Clone') {

steps {

git '(git@github.com:akhila40/APIwiz.git)'

}

}

stage('Build Docker Image') {

steps {

dir('APIwiz') {

sh 'docker build -t my-image:latest .'

}

}

}

stage('Publish Docker Image') {

steps {

sh 'docker tag my-image:latest ak407/my-image:latest'

sh 'docker push ak407/my-image:latest'

}

}

}

}

Tried but I did not get the exact o/p

**K8’s YAML file**

apiVersion: apps/v1

kind: Deployment

metadata:

name: image-name

spec:

replicas: 3

selector:

matchLabels:

app: image-name

template:

metadata:

labels:

app: image-name

spec:

containers:

- name: image-name

image: your-docker-hub-ak407/image-name:latest

ports:

- containerPort: 80

2. Automate to spin up a network and virtual machines. Install the Nginx package and start the service(any cloud) tried but did not got output

provider "aws" {

  region = "ap-south-1"

  access\_key = "AKIAU6GDYOO4JCFJOGCO"

  secret\_key = "U9vacA66V9enFeyGetkZJBewXP7gwHEcSMOp3le/"

}

module "aws\_instance" {

  source = "terraform-aws-modules/ec2-instance/aws"

  name = "my-ec2"

  ami = "ami-01376101673c89611"

  instance\_type = "t2.micro"

  key\_name = "Anvith.pem"

  monitoring = false

  subnet\_id = "subnet-0d04dc28408d745f3"

  vpc\_security\_group\_ids = ["sg-09916bcf961478b20"]

  associate\_public\_ip\_address = true

}

resource "null\_resource" "example" {

  connection {

    type = "ssh"

    host = module.aws\_instance.public\_ip

    user = "ubuntu"

  private\_key = file("C:\\Users\\Sulake\\Desktop\\PREP\\terraform1")

  }

  provisioner "remote-exec" {

    inline = [

      "sudo apt update",

      "sudo apt install nginx -y",

      "sudo systemctl start nginx",

      "sudo systemctl enable nginx"

    ]

  }

}