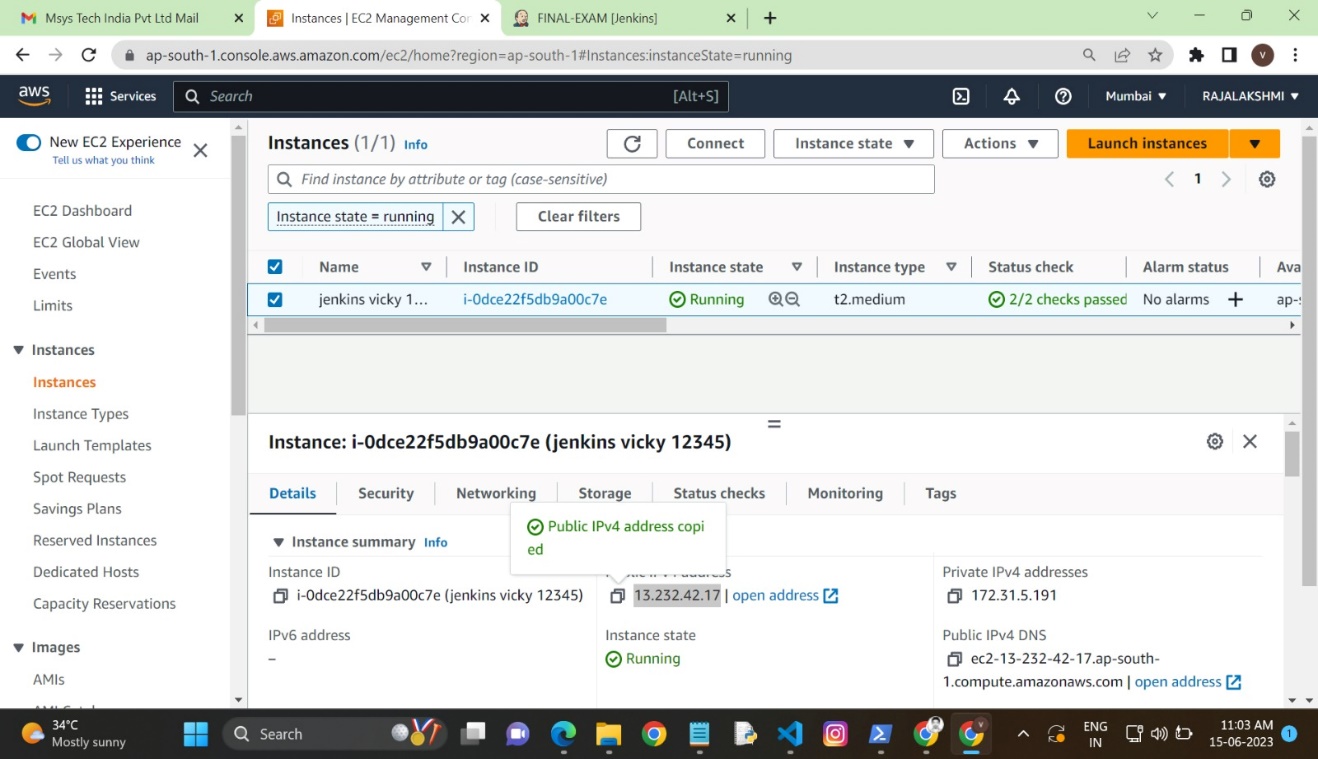
**Final Assessment**

**Create Build and Deploy Pipeline for K8s application deployment.**

1. **Create an EC2 instance in the AWS console:**



EC2 instance created

1. **Steps for installing Jenkins in the EC2 instance:**

sudo apt update -y

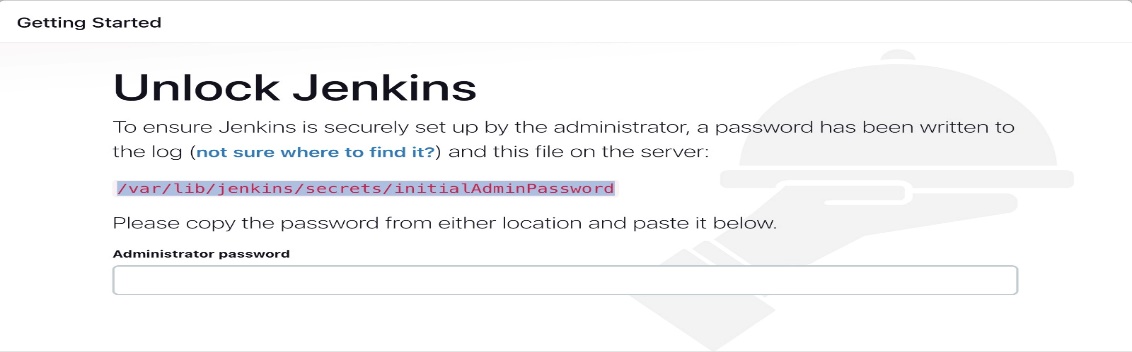
sudo apt install openjdk-8-jdk -y

#wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

#sudo apt-get update -y

#sudo apt-get install jenkins -y



Administrator password get with the help of cat cmd

A screenshot of a computer

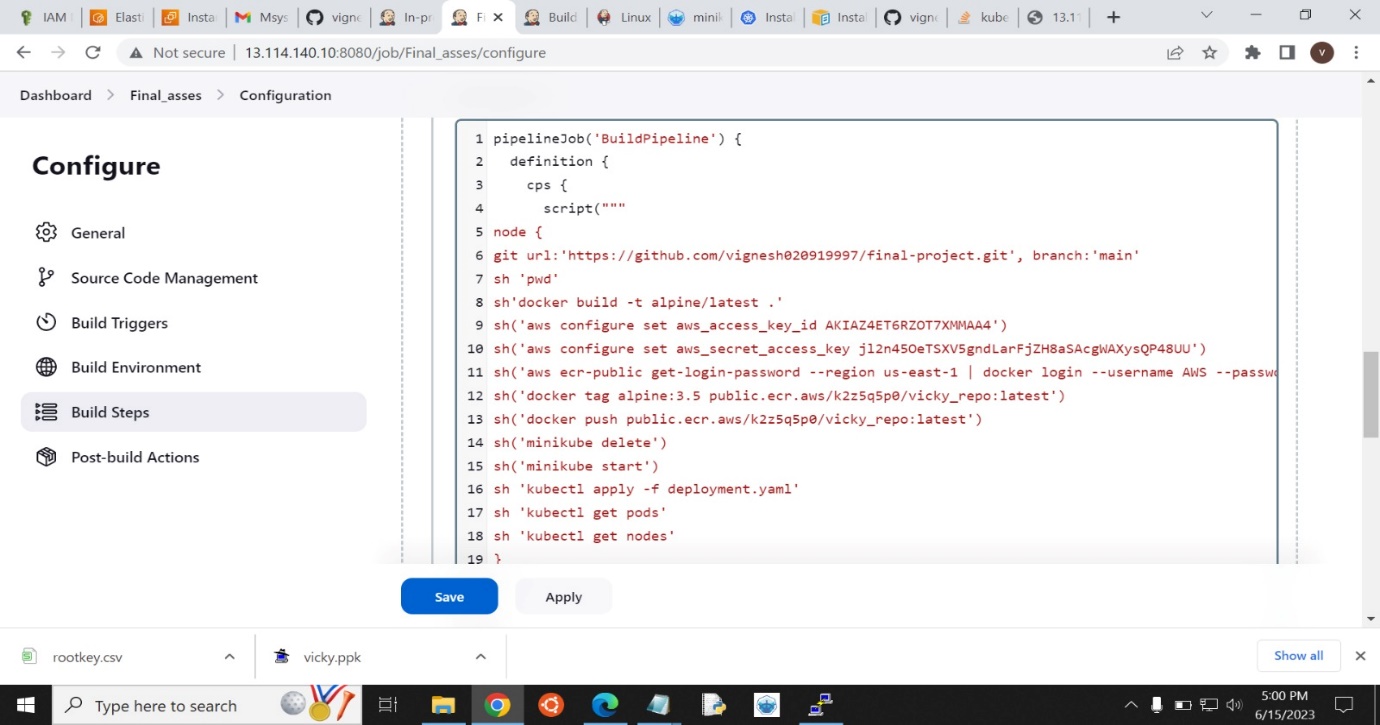
Description automatically generated with medium confidence

Create a new Jenkins pipeline:

1. **Create Free Style Job to Run Job DSL code to create above pipeline jobs.**

Jenkins, create a new pipeline job and configure it with the Git repository URL for the Java application.

Add a Jenkins file to the Git repository to define the pipeline stages.



DSL pipeline

1. **Install Minikube in the local system:**
2. After installing kubectl
3. Then minikube addon enable ingress
4. Using the kubectl cmd to deploy the yaml file

apiVersion: apps/v1

kind: Deployment # Kubernetes resource kind we are creating

metadata:

name: pyapp-k8s-deployment

spec:

selector:

matchLabels:

app: pyapp-k8s

replicas: 2 # Number of replicas that will be created for this deployment

template:

metadata:

labels:

app: pyapp-k8s

spec:

containers:

- name: pyapp-k8s

image: nagarajansam/pythonapp:latest

imagePullPolicy: IfNotPresent

ports:

- containerPort: 5000 # The port that the container is running on in the cluster

---

apiVersion: v1 # Kubernetes API version

kind: Service # Kubernetes resource kind we are creating

metadata: # Metadata of the resource kind we are creating

name: pyapp-k8ssvc

spec:

selector:

app: pyapp-k8s

ports:

- protocol: "TCP"

port: 8081 # The port that the service is running on in the cluster

targetPort: 5000 # The port exposed by the service

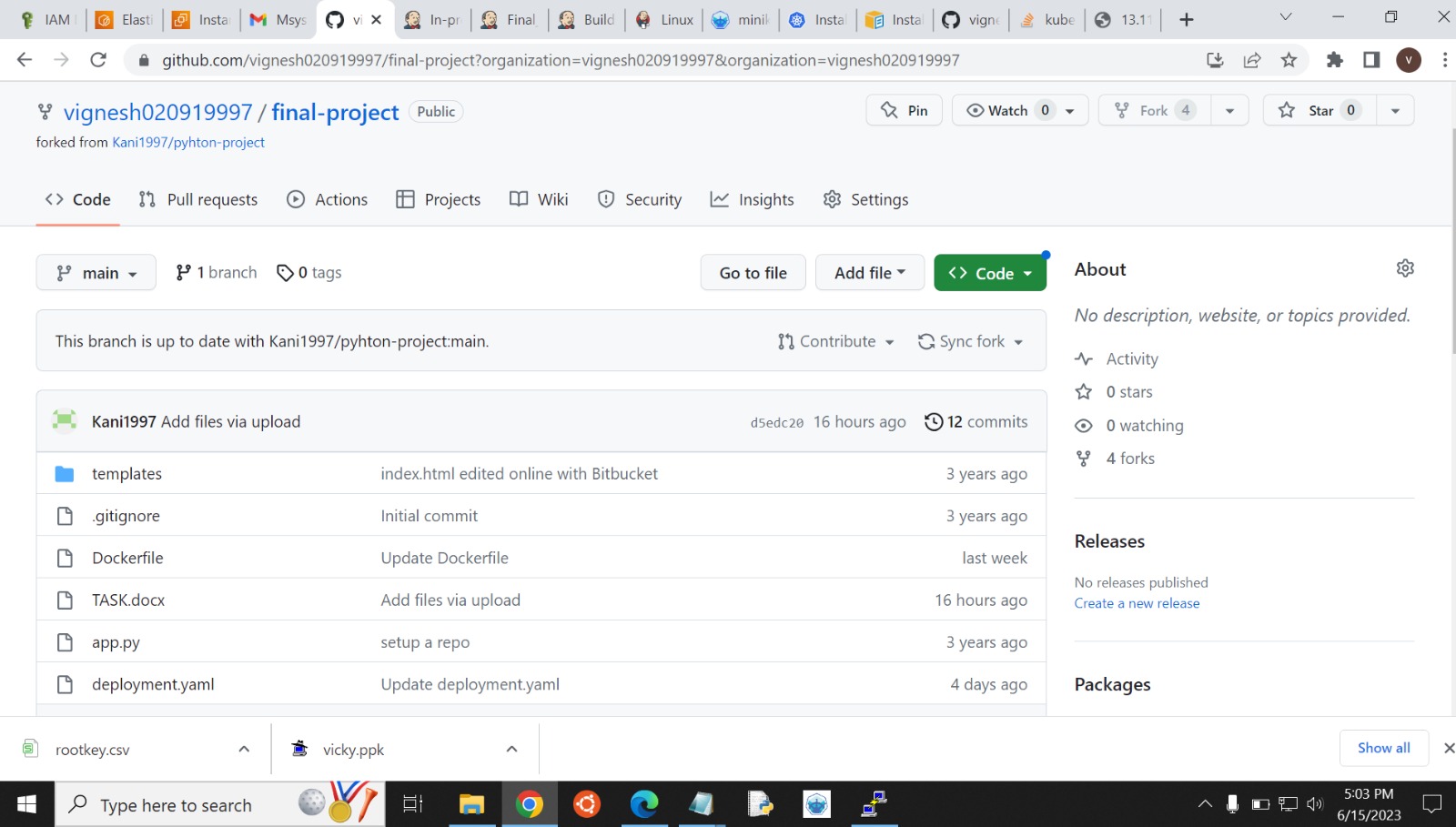
type: NodePort # type of the service

vi) Commands:

kubectl apply -f deployment.yaml

kubectl port-forward --address 0.0.0.0 svc/pyapp-k8ssvc 8081:8081

1. **Gitup Repository**



Reference: https://github.com/vignesh020919997/final-project

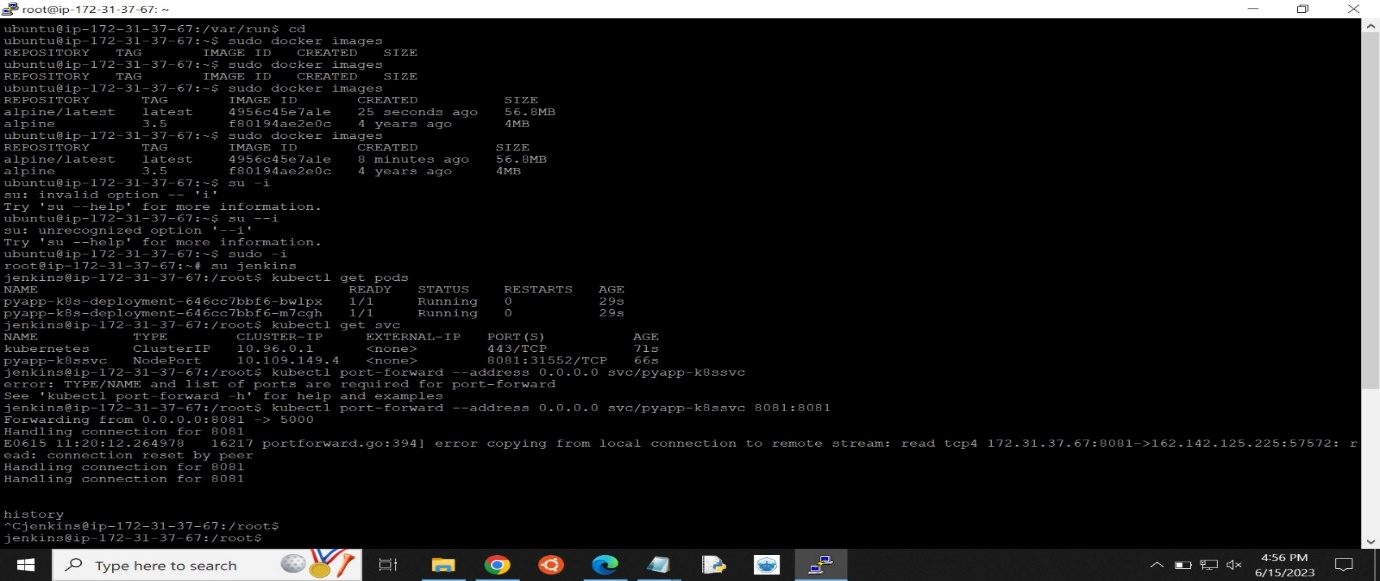
1. **After that I Hit the IP with the port number:**

A screenshot of a computer

Description automatically generated

**It works** Run http://<ip\_address>:<port> IP: 13.114.140.10 , Port number: 8081

<http://13.114.140.10:8081> Finally application deploys the K8



1. **The code stuff in a git repository:**

**Git-repositorylink:** https://github.com/vignesh020919997/final-project