JAVA APPLICATION DEPLOYMENT IN MINIKUBE

**Linux System Setup and Java Installation**

sudo apt install fontconfig openjdk-17-jre

java -version

**Jenkins Installation and Management**

sudo service jenkins restart

sudo service jenkins status

For installation instructions: [Jenkins Installation Guide](https://www.jenkins.io/doc/book/installing/linux/#debianubuntu)

**Docker Installation and Commands**

sudo apt install docker.io -y

sudo service docker restart

sudo service docker status

sudo usermod -aG docker $USER

# Checking Docker Images and Containers

docker images

docker ps

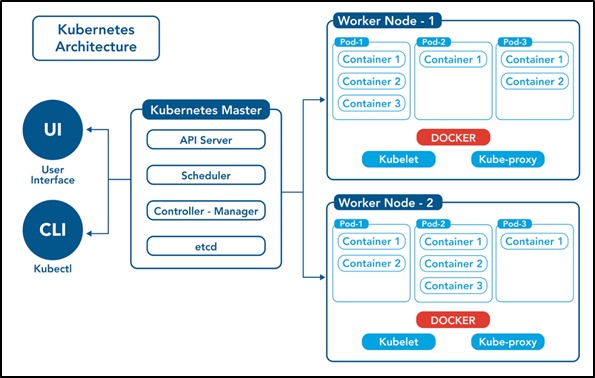
# Fixing permission issues

sudo chmod 666 /var/run/docker.sock

**Docker Compose Installation**

sudo apt install docker-compose -y

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

****

**Kubernetes (K8s) Installation and Commands**

**Installing kubectl**

curl -LO https://dl.k8s.io/release/v1.32.0/bin/linux/amd64/kubectl

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

chmod +x kubectl

mkdir -p ~/.local/bin

mv ./kubectl ~/.local/bin/kubectl

kubectl version --client

More details: [Install kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/)

**Installing Minikube**

curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

minikube start

minikube status

**Kubernetes Commands**

**Pod Management**

# Create a pod

kubectl run my-pod --image=nginx --port=80

# View all pods

kubectl get pods

kubectl get pods -A

kubectl get pods -n kube-system

# View pod details

describe pod <pod-name>

kubectl logs <pod-name>

kubectl exec <pod-name> -- <command>

**YAML Configuration for a Pod**

apiVersion: v1

kind: Pod

metadata:

name: my-pod

labels:

app: my-web-app

type: backend

spec:

containers:

- name: nginx-container

image: nginx

ports:

- containerPort: 80

**ReplicaSet Management**

# Create a ReplicaSet

kubectl create -f rs-test.yml

kubectl apply -f rs-test.yml

# View ReplicaSets

kubectl get replicasets

kubectl get rs -o wide

# Scale a ReplicaSet

kubectl scale replicaset <replicaset-name> --replicas=<desired-replica-count>

# Delete a ReplicaSet

kubectl delete rs <replicaset-name>

kubectl delete -f rs-test.yml

**ReplicaSet YAML Configuration**

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: my-rs

labels:

name: my-rs

spec:

replicas: 4

selector:

matchLabels:

apptype: web-backend

template:

metadata:

labels:

apptype: web-backend

spec:

containers:

- name: my-app

image: nginx

ports:

- containerPort: 8080

**Deployment Management**

# Create a deployment

kubectl create deployment webnginx2 --image=nginx:latest --replicas=1

# View deployments

kubectl get deployments

kubectl describe deploy <deployment-name>

# Scale a deployment

kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>

# Delete a deployment

kubectl delete deploy <deployment-name>

kubectl delete -f web-deploy.yml

**Deployment YAML Configuration**

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deploy

labels:

name: my-deploy

spec:

replicas: 4

selector:

matchLabels:

apptype: web-backend

strategy:

type: RollingUpdate

template:

metadata:

labels:

apptype: web-backend

spec:

containers:

- name: my-app

image: nginx

ports:

- containerPort: 7070

**Service Management**

# View services

kubectl get svc

# Create a service from YAML

kubectl create -f service.yml

# Delete a service

kubectl delete svc <service-name>

**Service YAML Configuration**

apiVersion: v1

kind: Service

metadata:

name: my-service

labels:

app: my-service

spec:

type: NodePort

ports:

- port: 9000

targetPort: 8080

nodePort: 30002

selector:

apptype: web-backend

**Namespace Management**

# Create a namespace

kubectl create namespace <namespace-name>

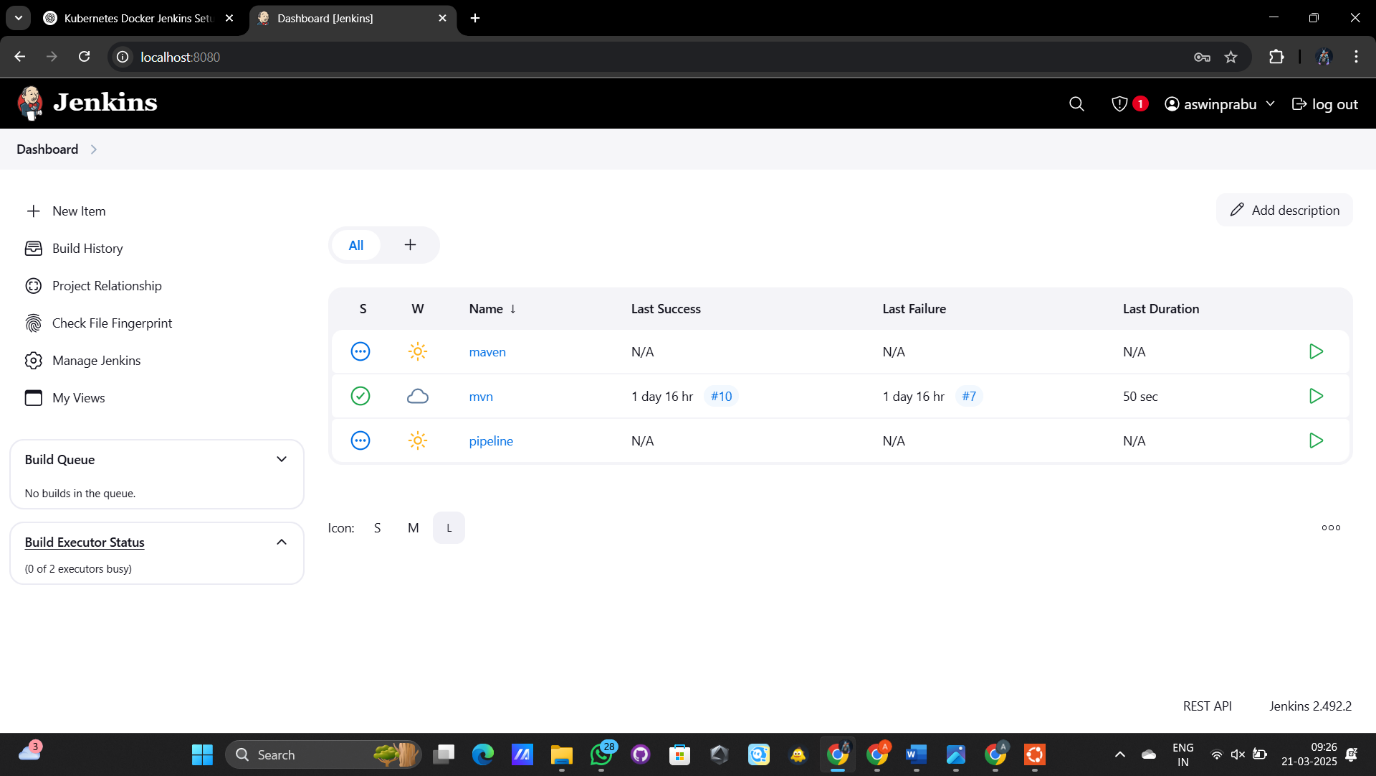
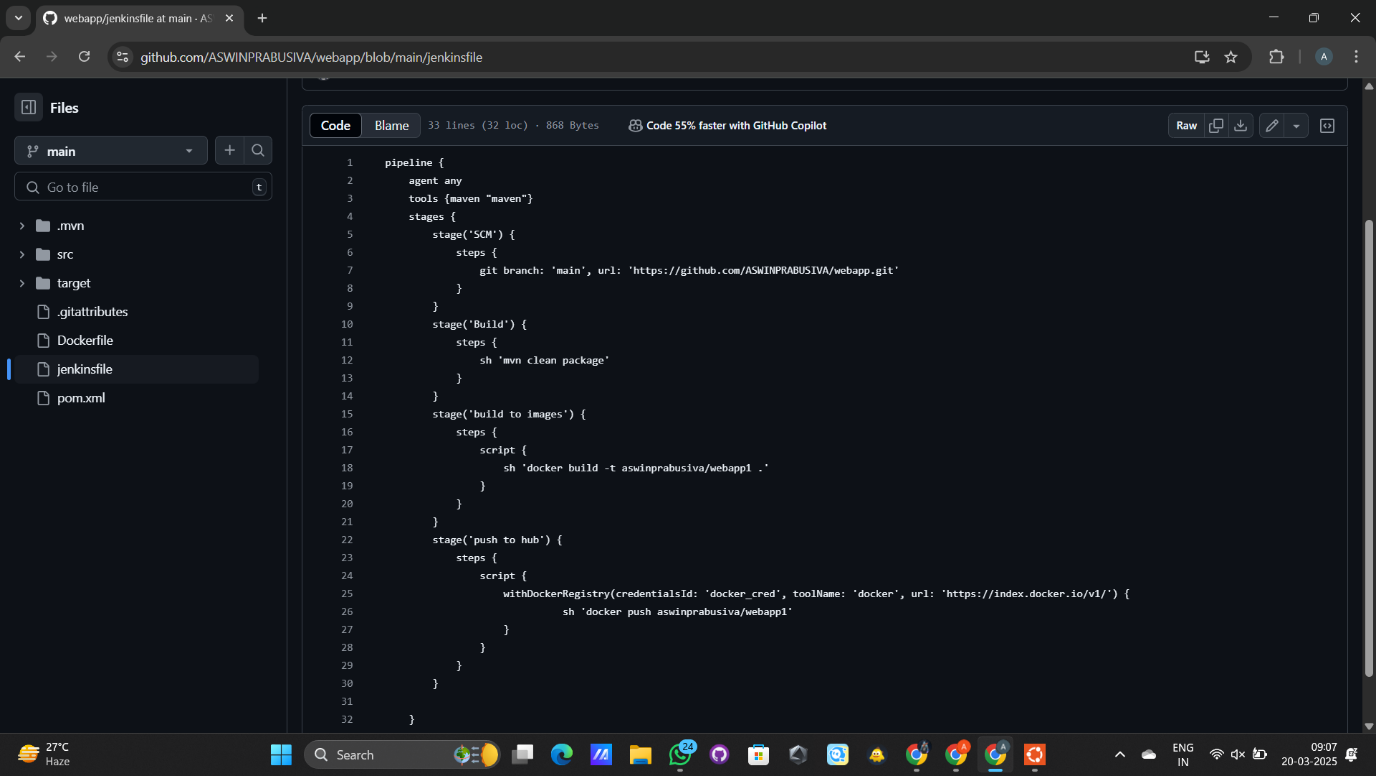
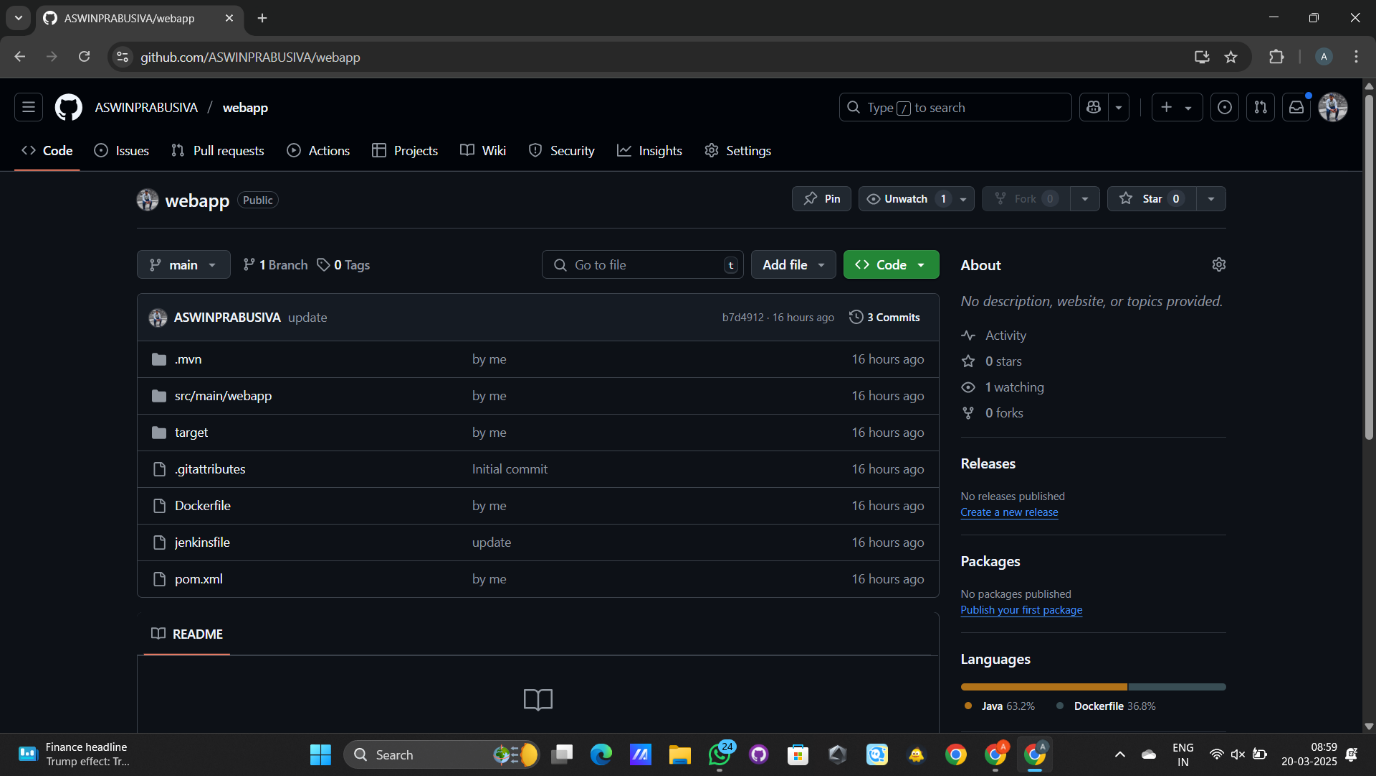
kubectl create ns my-bank

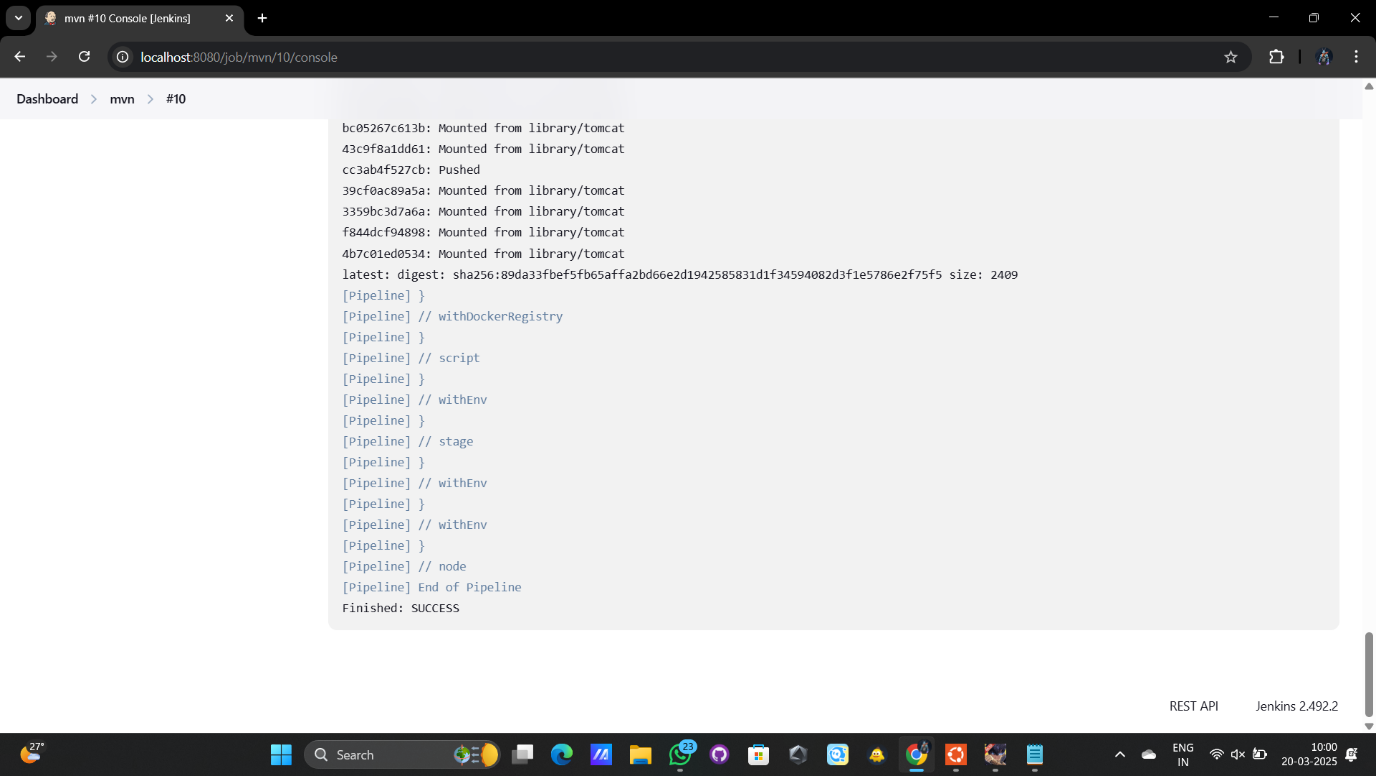
# View namespaces

kubectl get ns

# Switch to a namespace

kubectl config set-context --current --namespace=<namespace-name>

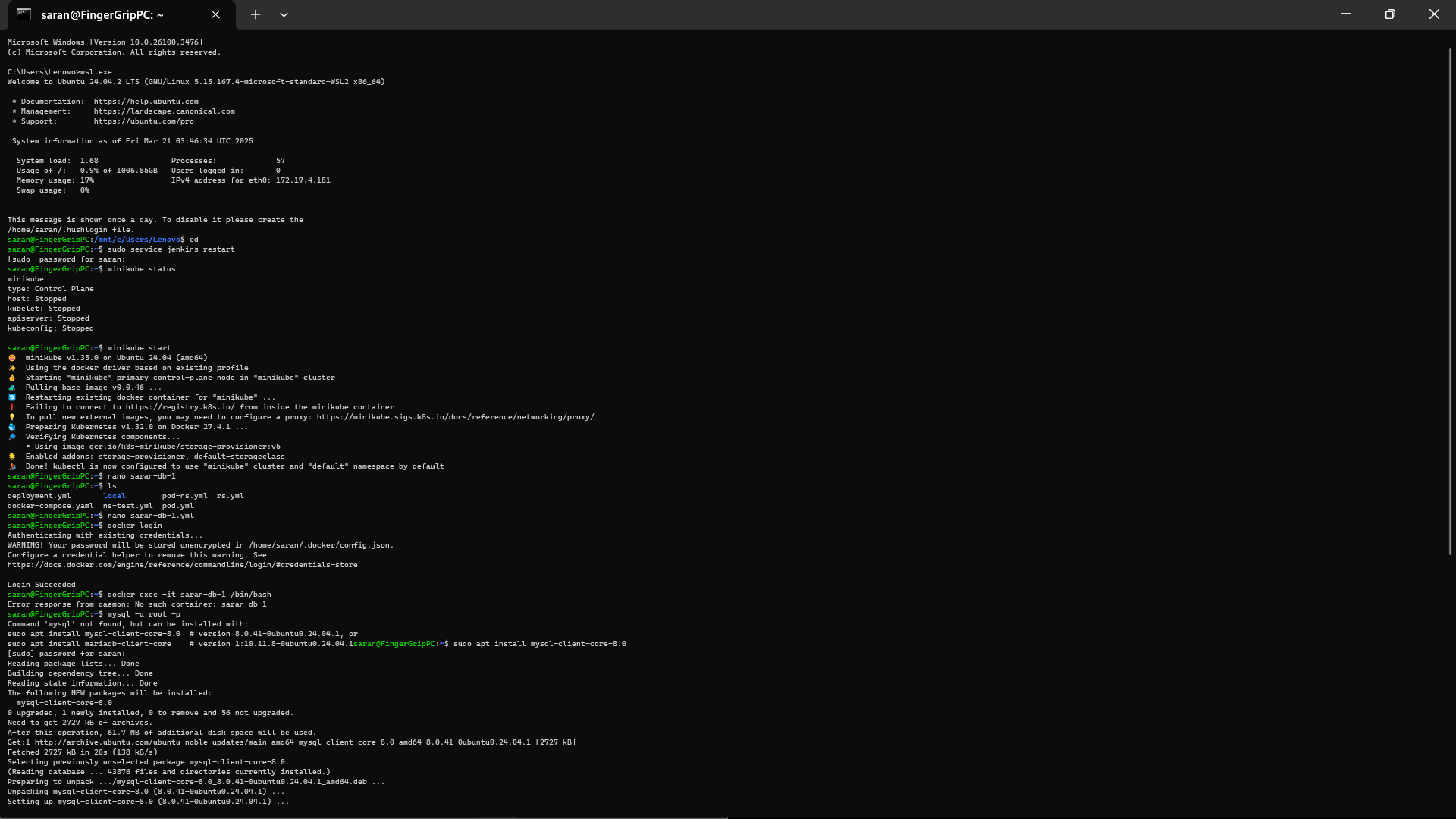
Automation using Jenkins push image from github to docker hub using jenkins automation:



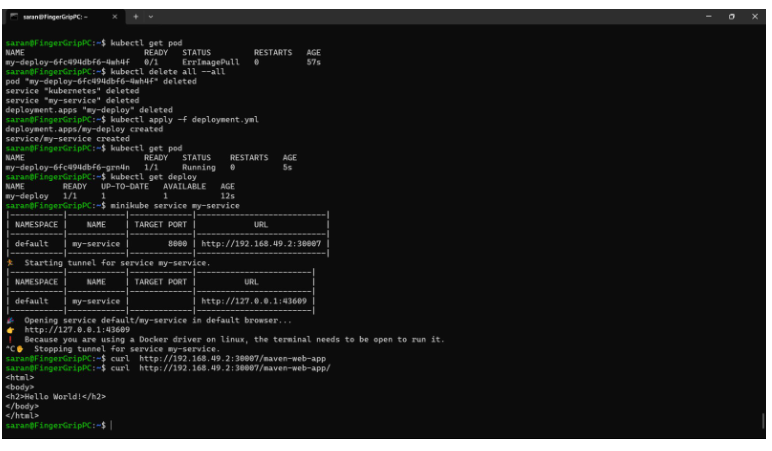
A screenshot of a computer

AI-generated content may be incorrect.

Minikube installation and mysql:



Kubernetes, Namespace:



Task 5:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.