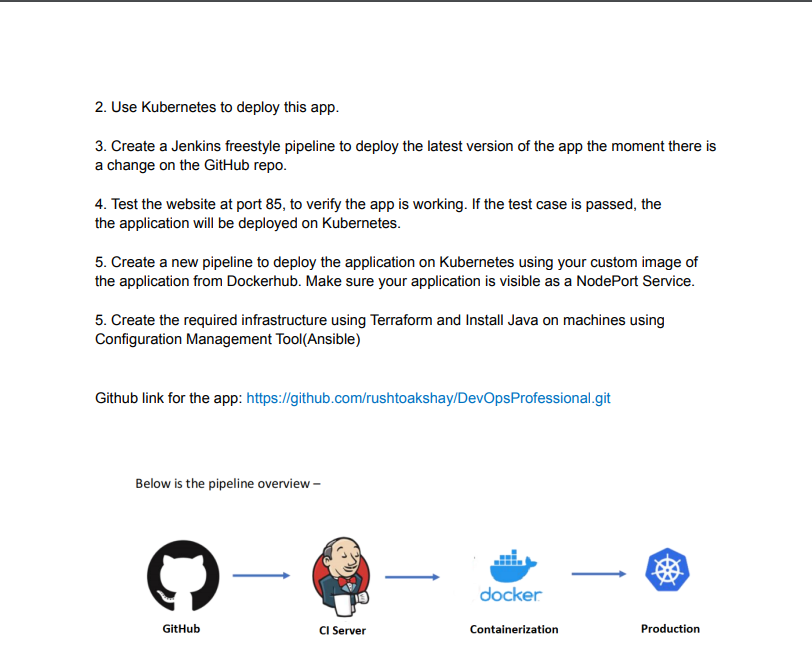
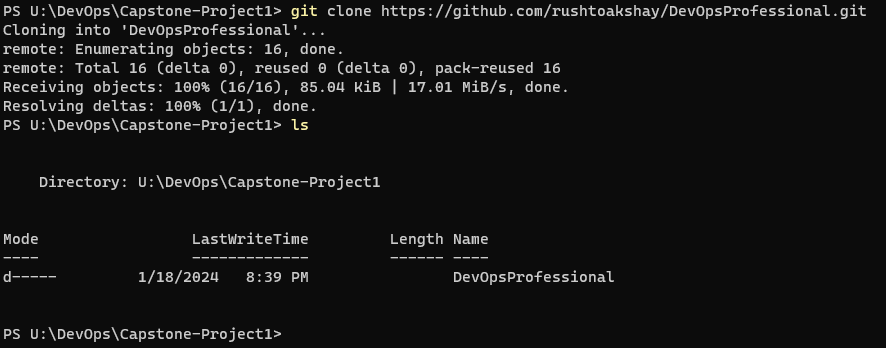
**Capstone Project – 1**

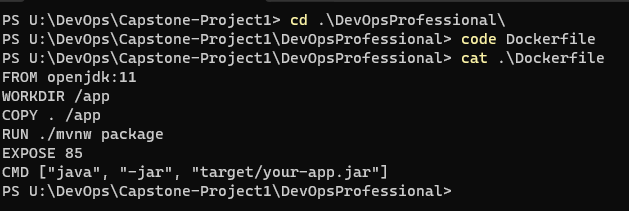
****

****

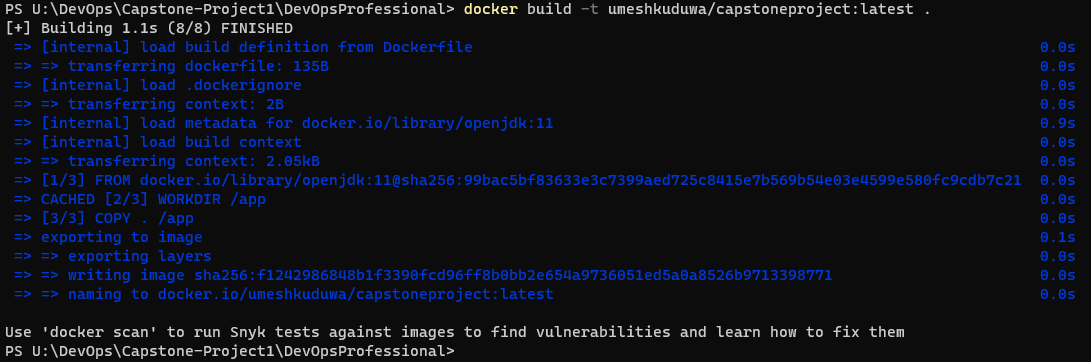
1. Proceed with Cloning the GitHub Repo to local machine.



1. Now, move inside the DevOpsProfessional folder and create a Dockerfile to dockerize the app.



1. Now, build the docker image.



1. Now create a deployment.yaml for Kube.

apiVersion: apps/v1

kind: Deployment

metadata:

name: myapp-deployment

spec:

replicas: 3

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec:

containers:

- name: myapp

image: umeshkuduwa/capstoneproject:latest

ports:

- containerPort: 85

1. Apply the deployment.yaml by passing : kubectl apply -f deployment.yaml
2. Create the Jenkins Freestyle Pipeline. Also a script : test\_script.sh to validate the testing.

**test\_script.sh**

#!/bin/bash

APP\_URL="http:// 24.151.201.206:85"

check\_application() {

local response\_code

response\_code=$(curl -s -o /dev/null -w "%{http\_code}" $APP\_URL)

if [ "$response\_code" == "200" ]; then

echo "Application test passed! Deploying to Kubernetes..."

return 0 # Success

else

echo "Application test failed. HTTP Status Code: $response\_code"

return 1 # Failure

fi

}

check\_application

**Jenkins Pipeline :**

pipeline {

agent any

stages {

stage('Checkout') {

steps {

git 'https://github.com/rushtoakshay/DevOpsProfessional.git'

}

}

stage('Build, Test, and Deploy') {

steps {

script {

// Build Docker image

dir('DevOpsProfessional') {

docker build -t umeshkuduwa/capstoneproject:latest .

docker push umeshkuduwa/capstoneproject:latest

}

sh 'bash test\_script.sh'

if [ $? -eq 0 ]; then

echo "Test passed. Proceeding with deployment."

// Deploy to Kubernetes

def kubeConfig = readFile 'deployment.yaml'

sh "echo '$kubeConfig' | kubectl apply -f -"

sh 'kubectl rollout status deployment/myapp-deployment'

sh 'kubectl expose deployment myapp-deployment --type=NodePort --name=myapp-service --port=85'

} else {

echo "Test failed. Skipping deployment."

}

}

}

}

}

}

1. After testing, create a new pipeline :

pipeline {

agent any

stages {

stage('Checkout') {

steps {

git 'https://github.com/rushtoakshay/DevOpsProfessional.git'

}

}

stage('Build and Deploy') {

steps {

script {

dir('DevOpsProfessional') {

docker pull umeshkuduwa/capstoneproject:latest

def kubeConfig = readFile 'deployment\_custom\_image.yaml'

sh "echo '$kubeConfig' | kubectl apply -f -"

sh 'kubectl rollout status deployment/myapp-deployment-custom'

sh 'kubectl expose deployment myapp-deployment-custom --type=NodePort --name=myapp-service-custom --port=85'

}

}

}

}

}

}

1. Create the rest of required infrastructure with Terraform and Ansible :

**Terraform :**

provider "aws" {

region = "ap-south-1"

access\_key = ""

secret\_key = ""

}

resource "aws\_instance" "capstoneInstance" {

ami = "ami-0c55b159cbfafe1f0"

instance\_type = "t2.micro"

key\_name = "MumbaiKeyPair"

}

**Ansible :**

---

- name: Install Java

hosts: capstone-server

tasks:

- name: Update apt cache

apt:

update\_cache: yes

- name: Install Java

apt:

name: openjdk-11-jdk

state: present