CAPSTONE_PROJECT 2
DEVOPS

PROJECT_LINK: https://github.com/whiteblaze1/website_Project_Devops.git

Used 3 Ec2 Instances:
1. Jenkins_master
2. k8_master
3. k8_slave

TERRAFORM FILE:

```
provider "aws" {
  access_key = ""
  secret_key = ""
  region = "us-east-1"

}
resource "aws_instance" "Jenkins_Master" {
  ami = "ami-0e86e20dae9224db8"
  instance_type = "t2.medium"
  key_name = "windows_key"

  tags = {
   Name = "Jenkins_Master"
  }
}

resource "aws_instance" "K8Master" {
  ami = "ami-0e86e20dae9224db8"
  instance_type = "t2.medium"
  key_name = "windows_key"

  tags = {
```

```
    Name = "KubsMaster"
   }
}

resource "aws_instance" "K8Slave" {
   ami = "ami-0e86e20dae9224db8"
   instance_type = "t2.micro"
   key_name = "windows_key"

   tags = {
    Name = "KubsSlave"
   }
}
output "jenkins_master_ip" {
  value = aws_instance.Jenkins_Master.public_ip
}

output "k8_master_ip" {
  value = aws_instance.K8Master.public_ip
}

output "k8_slave_ip" {
  value = aws_instance.K8Slave.public_ip
}
```

STEPS:

1.Install Ansible on main machine()

```
ubuntu@ip-172-31-83-108:/etc/ansible$ ansible-playbook ansi.yaml

PLAY [Installation on MainMachine] *********************************

TASK [Gathering Facts] *********************************************
ok: [localhost]

TASK [Installing Dependencies] *************************************
changed: [localhost]

PLAY [Installation on K8MASTER(JenkinsSlave)] **********************

TASK [Gathering Facts] *********************************************
ok: [52.5.63.57]

TASK [Installation of Dependencies] ********************************
changed: [52.5.63.57]

PLAY [Installation K8 slave] ***************************************

TASK [Gathering Facts] *********************************************
ok: [44.206.252.55]

TASK [Installation of Dependencies] ********************************
changed: [44.206.252.55]

PLAY RECAP *********************************************************
44.206.252.55              : ok=2    changed=1    unreachable=0
52.5.63.57                 : ok=2    changed=1    unreachable=0
localhost                  : ok=2    changed=1    unreachable=0

ubuntu@ip-172-31-83-108:/etc/ansible$
```

ANSIBLE-FILE

```
---
- name: Installation on MainMachine
  hosts: localhost
  become: true
  tasks:
  - name: Installing Dependencies
    script: local.sh
- name: Installation on K8MASTER(JenkinsSlave)
  hosts: k8m
  become: true
```

```
  tasks:
  - name: Installation of Dependencies
    script: master.sh
- name: Installation on K8Slave
  hosts: k8s
  become: true
  tasks:
  - name: Installation of Dependencies
script: slave.sh
```

local.sh

```
#!/bin/bash
sudo apt update
sudo apt install openjdk-17-jre-headless -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
echo $PATH
```

2.Install Java,jenkins on Main machine

3. Docker on k8_master

4.Docker on k8_slave

5.Create Dockerfile on MainMachine

DOCKER FILE—>

```
FROM ubuntu
RUN apt update
RUN apt install apache2 -y
```

```
ADD . /var/www/html/
ENTRYPOINT apachectl -D FOREGROUND
```

6.Create deploy.yaml for deploying the image created through docker file should be same name as should be used afterwards

DEPLOY.YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: devops-deploy
  labels:
    app: custom
spec:
  replicas: 2
  selector:
    matchLabels:
      app: custom
  template:
    metadata:
      labels:
        app: custom
    spec:
      containers:
        - name: devo_p
          image: whiteblaze098/devop
          ports:
            - containerPort: 80
```

SERVICE.YAML

```
apiVersion: v1
kind: Service
metadata:
  name: devops-deploy
  labels:
    app: custom
spec:
  type: NodePort
  selector:
    app: custom
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
    nodePort: 30008
```

7.Install k8s on k8s_master (Different file for that)

8. Use k8_slave as 2nd node for kubernetes

```
ubuntu@ip-172-31-81-36:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-81-36    Ready    control-plane   8h    v1.29.0
ip-172-31-84-197   Ready    <none>          8h    v1.29.0
ubuntu@ip-172-31-81-36:~$
```

9.Create jenkins pipeline with following work:

1. Adding Github project or downloading the project
2. Build Docker file as needed for the image
3. use Deploy.yaml and Service.yaml with 2 replica sets and no deport at 30008
4. Check the Link in Browser with k8s_master_ip:30008

Stage 'Kubernetes'

Started 2 min 6 sec ago

Queued 0 ms

Took 0.77 sec

Success

View as plain text

kubectl apply -f deploy.yaml
Shell Script — 0.3 sec

kubectl apply -f service.yaml
Shell Script — 0.3 sec

```
0  + kubectl apply -f service.yaml
1  service/devops-deploy unchanged
```

```
0  The recommended git tool is: NONE
1  No credentials specified
2  Fetching changes from the remote Git repository
3  Checking out Revision e56c453b501206be0d610d15906dbedbe55babf1 (refs/remotes/origin/master)
4  Commit message: "Update deploy.yaml"
5   > git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/create_pipeline/.git # timeout=10
6   > git config remote.origin.url https://github.com/whiteblaze1/website_Project_Devops.git # timeout=10
7  Fetching upstream changes from https://github.com/whiteblaze1/website_Project_Devops.git
8   > git --version # timeout=10
9   > git --version # 'git version 2.43.0'
10  > git fetch --tags --force --progress -- https://github.com/whiteblaze1/website_Project_Devops.git
   +refs/heads/*:refs/remotes/origin/* # timeout=10
11  > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
12  > git config core.sparsecheckout # timeout=10
13  > git checkout -f e56c453b501206be0d610d15906dbedbe55babf1 # timeout=10
14  > git branch -a -v --no-abbrev # timeout=10
15  > git branch -D master # timeout=10
16  > git checkout -b master e56c453b501206be0d610d15906dbedbe55babf1 # timeout=10
```

Scroll to Bottom

Stage 'Docker Stage'

Started 54 sec ago

Queued 0 ms

Took 3.7 sec

Success

View as plain text

sudo docker build -t whiteblaze098/devop .
Shell Script — 0.58 sec

sudo echo $DOC_PSW | sudo docker login -u $DOC_USR --password-stdin
Shell Script — 0.31 sec

sudo docker push whiteblaze098/devop
Shell Script — 2.6 sec

```
0  + sudo docker push whiteblaze098/devop
1  Using default tag: latest
2  The push refers to repository [docker.io/whiteblaze098/devop]
3  5d6abee99c8c: Preparing
4  bfc2f84eed15: Preparing
5  bc6817d54e24: Preparing
6  f36fd4bb7334: Preparing
7  f36fd4bb7334: Layer already exists
8  bfc2f84eed15: Layer already exists
9  bc6817d54e24: Layer already exists
10 5d6abee99c8c: Pushed
11 latest: digest: sha256:fce6cec5e8151b5c930cca46c75327e5221ba3bc9b494053cfa55e58b9e263e9 size: 1164
```

PIPELINE SCRIPT:

```
pipeline {
    agent none
```

```
    environment {
        DOC = credentials('2c12c27d-3822-4d40-beee-b91b98535db9')
    }
    stages {
        stage("Git Work") {
            agent {
                label 'k8smaster'
            }
            steps {
                git url: 'https://github.com/whiteblaze1/
website_Project_Devops.git', branch: 'master'
            }
        }
        stage('Docker Stage') {
            agent {
                label 'k8smaster'
            }
            steps {

                sh 'sudo docker build -t whiteblaze098/devop .'

                // Login to Docker using credentials stored in Jenkins
                sh ' sudo echo $DOC_PSW | sudo docker login -u
$DOC_USR --password-stdin '


// Push the Docker image
                sh 'sudo docker push whiteblaze098/devop'
            }
        }
        stage('Kubernetes') {
            agent {
                label 'k8smaster'
            }
            steps {
```

```
                // Apply the Kubernetes deployment and service
                sh 'kubectl apply -f deploy.yaml'
                sh 'kubectl apply -f service.yaml'
            }
        }
    }
}
```