**EKS Setup & Deploy A Full Stack Application Using Manifest**

Terraform Repo: <https://github.com/jaiswaladi246/Mega-Project-Terraform.git>

========================================================

# Install AWS CLI

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

sudo apt install unzip

unzip awscliv2.zip

sudo ./aws/install

aws configure

-------------------------------------------------------------------------------------

#Install Terraform

sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl

curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list

sudo apt-get update && sudo apt-get install terraform -y

terraform -version

-------------------------------------------------------------------------------------------

# Kubeconfig

aws eks --region ap-south-1 update-kubeconfig --name devopsshack-cluster

---------------------------------------------------------------------------------

# Kubectl

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"

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

kubectl version --client

------------------------------------------------------------------------------------------

curl -sLO "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz"

tar -xzf eksctl\_$(uname -s)\_amd64.tar.gz

sudo mv eksctl /usr/local/bin

eksctl version

===============================================================================================

eksctl utils associate-iam-oidc-provider --region ap-south-1 --cluster devopsshack-cluster --approve

-----------------------------------------------------------------------------------------------

eksctl create iamserviceaccount \

--region ap-south-1 \

--name ebs-csi-controller-sa \

--namespace kube-system \

--cluster devopsshack-cluster \

--attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \

--approve \

--override-existing-serviceaccounts

-----------------------------------------------------------------------------------------------

kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/ecr/?ref=release-1.11"

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/cloud/deploy.yaml>

kubectl apply -f <https://github.com/cert-manager/cert-manager/releases/download/v1.12.0/cert-manager.yaml>

Manifest File Added Below

---

apiVersion: v1

kind: Secret

metadata:

name: mysql-secret

namespace: webapps

type: Opaque

data:

MYSQL\_ROOT\_PASSWORD: VGVzdEAxMjM=

---

apiVersion: v1

kind: ConfigMap

metadata:

name: mysql-config

namespace: webapps

data:

MYSQL\_DATABASE: bankappdb

---

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: ebs-sc

provisioner: ebs.csi.aws.com

parameters:

type: gp3

fsType: ext4

reclaimPolicy: Retain

volumeBindingMode: WaitForFirstConsumer

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: mysql-pvc

namespace: webapps

spec:

accessModes:

- ReadWriteOnce

storageClassName: ebs-sc

resources:

requests:

storage: 5Gi

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: mysql

namespace: webapps

spec:

selector:

matchLabels:

app: mysql

strategy:

type: Recreate

template:

metadata:

labels:

app: mysql

spec:

containers:

- image: mysql:8

name: mysql

resources:

requests:

memory: 512Mi

cpu: 500m

limits:

memory: 1Gi

cpu: "1"

env:

- name: MYSQL\_ROOT\_PASSWORD

valueFrom:

secretKeyRef:

name: mysql-secret

key: MYSQL\_ROOT\_PASSWORD

- name: MYSQL\_DATABASE

valueFrom:

configMapKeyRef:

name: mysql-config

key: MYSQL\_DATABASE

ports:

- containerPort: 3306

name: mysql

volumeMounts:

- mountPath: /var/lib/mysql

name: mysql-data

livenessProbe:

tcpSocket:

port: 3306

initialDelaySeconds: 30

periodSeconds: 10

readinessProbe:

exec:

command:

- mysqladmin

- ping

- -h

- 127.0.0.1

initialDelaySeconds: 30

periodSeconds: 10

volumes:

- name: mysql-data

persistentVolumeClaim:

claimName: mysql-pvc

---

apiVersion: v1

kind: Service

metadata:

name: mysql-service

namespace: webapps

spec:

ports:

- port: 3306

targetPort: 3306

selector:

app: mysql

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: bankapp

namespace: webapps

spec:

replicas: 1

selector:

matchLabels:

app: bankapp

template:

metadata:

labels:

app: bankapp

spec:

containers:

- name: bankapp

image: adijaiswal/bankapp:v6

resources:

requests:

memory: 256Mi

cpu: 250m

limits:

memory: 512Mi

cpu: 500m

ports:

- containerPort: 8080

env:

- name: SPRING\_DATASOURCE\_URL

value: jdbc:mysql://mysql-service:3306/bankappdb?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true

- name: SPRING\_DATASOURCE\_USERNAME

value: root

- name: SPRING\_DATASOURCE\_PASSWORD

valueFrom:

secretKeyRef:

name: mysql-secret

key: MYSQL\_ROOT\_PASSWORD

livenessProbe:

httpGet:

path: /login

port: 8080

initialDelaySeconds: 60

timeoutSeconds: 5

periodSeconds: 10

failureThreshold: 3

readinessProbe:

httpGet:

path: /login

port: 8080

initialDelaySeconds: 60

timeoutSeconds: 5

periodSeconds: 10

failureThreshold: 3

---

apiVersion: v1

kind: Service

metadata:

name: bankapp-service

namespace: webapps

spec:

type: LoadBalancer

ports:

- port: 80

targetPort: 8080

selector:

app: bankapp