**Create EKS Cluster & Use RDS as Database**

Create VM & add the root access keys & secret accesskeys of root using aws cli

**# Install AWS CLI & Add credentails**

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

**Clone below Repo & go inside it & run terraform init and after wards terraform apply –auto-approve**

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

**# 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

**#Install Ingress controller & Cert manager**

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**

---

apiVersion: v1

kind: Secret

metadata:

name: rds-mysql-secret

namespace: webapps

type: Opaque

stringData:

SPRING\_DATASOURCE\_USERNAME: root

SPRING\_DATASOURCE\_PASSWORD: Aditya#222

---

apiVersion: v1

kind: ConfigMap

metadata:

name: rds-mysql-config

namespace: webapps

data:

SPRING\_DATASOURCE\_URL: jdbc:mysql://rds-mysql:3306/bankappdb?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true

---

apiVersion: v1

kind: Service

metadata:

name: rds-mysql

namespace: webapps

spec:

type: ExternalName

externalName: database-1.cdqoqoe6gep3.ap-south-1.rds.amazonaws.com

ports:

- port: 3306

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: bankapp

namespace: webapps

spec:

replicas: 1

selector:

matchLabels:

app: bankapp

template:

metadata:

labels:

app: bankapp

spec:

initContainers:

- name: init-create-db

image: mysql:8

command:

- sh

- -c

- mysql -h rds-mysql -u root -p$MYSQL\_ROOT\_PASSWORD -e "CREATE DATABASE IF NOT EXISTS bankappdb;"

env:

- name: MYSQL\_ROOT\_PASSWORD

valueFrom:

secretKeyRef:

name: rds-mysql-secret

key: SPRING\_DATASOURCE\_PASSWORD

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

valueFrom:

configMapKeyRef:

name: rds-mysql-config

key: SPRING\_DATASOURCE\_URL

- name: SPRING\_DATASOURCE\_USERNAME

valueFrom:

secretKeyRef:

name: rds-mysql-secret

key: SPRING\_DATASOURCE\_USERNAME

- name: SPRING\_DATASOURCE\_PASSWORD

valueFrom:

secretKeyRef:

name: rds-mysql-secret

key: SPRING\_DATASOURCE\_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

**Test-Pod**

apiVersion: v1

kind: Pod

metadata:

name: mysql-client

spec:

containers:

- name: mysql

image: mysql:8

command: ["sleep"]

args: ["3600"]