```
docker –version
kind –version
kubectl version –client
```

Cloning the Github Repo -

```
git clone https://github.com/Arpit-commits/Arpit-CLO835-Assignment2.git

cd Arpit-CLO835-Assignment2
```

Branch -

```
git checkout version1
```

AWS Secrets

```
aws configure
nano ~/.aws/credentials

aws_access_key_id = ASIAR7I7HI6XYLDBDNKJ
aws_secret_access_key = NB4PcIU/C98NGbVmp7vZjbQtDa0LcBCiLmPrg2gC
aws_session_token =
IQoJb3JpZ2luX2VjEEcaCXVzLXdlc3QtMiJGMEQCICV5Zl4FpyuR6qjsG5gdTNAc076QV1ohQC5NT5ffylaHAiA/HOb0RO/1Apti
rlkVHA821bxyhHr8OSOX60EYfHeiaSq4AgiQ//////////8BEAIaDDEzNTg5MzgyOTU1MSIMATVvrLsv1opKLqCwKowCsyfiH2lV
c5HekxgwVhv>
region = us-east-1
```

To confirm the aws secrets

```
aws sts get-caller-identity
```

Create Cluster

```
kind create cluster --config k8s-manifests/kind-config.yaml --name clo835-cluster

kind create cluster --name clo835-cluster --config kind-config.yaml
```

Cluster Ip for report -

```
kubectl cluster-info
```

Create NameSpaces -

```
kubectl create namespace flask
kubectl create namespace sql
```

Creating Secret

Kubectl

```
kubectl create secret docker-registry ecr-secret --docker-server=135893829551.dkr.ecr.us-east-1.amazonaws.com --docker-username=AWS --docker-password=$(aws ecr get-login-password --region us-east-1) -n flask
```

Sql

```
kubectl create secret docker-registry ecr-secret --docker-server=135893829551.dkr.ecr.us-east-1.amazonaws.com --docker-username=AWS --docker-password=$(aws ecr get-login-password --region us-east-1) -n sql
```

Verify Secret

```
kubectl get secrets -n flask
kubectl get secrets -n sql
```

Create Images

```
docker build -t sql -f Dockerfile.flask .
docker build -t sql -f Dockerfile.sql .
```

Tagging the images

```
docker tag flask:latest 135893829551.dkr.ecr.us-east-1.amazonaws.com/flask:v1.0
docker tag sql:latest 135893829551.dkr.ecr.us-east-1.amazonaws.com/sql:v1.0
```

Login to ecr

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 135893829551.dkr.ecr.us-east-1.amazonaws.com
```

Pushing the images

```
docker push 135893829551.dkr.ecr.us-east-1.amazonaws.com/flask:v2.0
docker push 135893829551.dkr.ecr.us-east-1.amazonaws.com/sql:v1.0
```

List images

```
aws ecr list-images --repository-name flask --region us-east-1
aws ecr list-images --repository-name sql --region us-east-1
```

Applying Files

```
kubectl apply -f k8s-manifests/flask-service.yaml
kubectl apply -f k8s-manifests/sql-service.yaml
kubectl apply -f k8s-manifests/flask-deployment.yaml
kubectl apply -f k8s-manifests/mysql-deployment.yaml
kubectl apply -f k8s-manifests/flask-pod.yaml
kubectl apply -f k8s-manifests/mysql-pod.yaml
kubectl get pods --all-namespaces
kubectl get svc --all-namespaces

kubectl apply -f k8s-manifests/flask-pod.yaml
kubectl apply -f k8s-manik ge fests/mysql-pod.yaml
```

Go inside the pod to check curl

```
kubectl exec -it $(kubectl get pod -n flask -l app=employees -o jsonpath="{.items[0].metadata.name}") -n flask – sh
kubectl exec -it $(kubectl get pod -n flask -l app=employees -o jsonpath="{.items[0].metadata.name}") -n flask -- bash
```

Install Curl -

```
apt-get update && apt-get install -y curl
apt install -y curl
```

Curl the page

```
curl http://localhost:8080/
```

report

```
kubectl logs flask-pod -n flask
```

For version 2

```
git checkout version2
git reset --hard version2
```

Creating Image loggin in and pushing it

```
docker build -t 135893829551.dkr.ecr.us-east-1.amazonaws.com/flask:v2.0 -f Dockerfile.flask .

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin
135893829551.dkr.ecr.us-east-1.amazonaws.com

docker push 135893829551.dkr.ecr.us-east-1.amazonaws.com/flask:v2.0
```

Update and Deploy:

```
kubectl apply -f k8s-manifests/flask-deployment.yaml

kubectl rollout status deployment/flask-deployment -n flask

---------------------------------
kubectl get pods -n flask


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

curl http://localhost:8080/

###########################
nano k8s-manifests/flask-deployment.yaml
change the colour
kubectl apply -f k8s-manifests/flask-deployment.yaml

kubectl rollout status deployment/flask-deployment -n flask

 kubectl get pods -n flask
```

Terraform – to create instance

```
wget https://releases.hashicorp.com/terraform/1.9.5/terraform_1.9.5_linux_amd64.zip

unzip terraform_1.9.5_linux_amd64.zip

sudo mv terraform /usr/local/bin/

terraform –version

mkdir -p ~/environment/terraform

cd ~/environment/terraform

ssh-keygen -t rsa -b 4096 -f cloud9-key -N ""

chmod 400 cloud9-key

chmod 644 cloud9-key.pub

ls -l

nano main.tf

terraform init

terraform plan

terraform apply

ssh -i ~/environment/terraform/cloud9-key ec2-user@<ec2-public-ip>
```