kubectl create secret generic db-user-literal --from-literal=username=devuser

kubectl create secret generic db-pass-literal --from-literal=password='S!B\\*d$zDsb='

kubectl get secret db-user-literal -o jsonpath="{.data.username}" | base64 --decode

kubectl get secret db-pass-literal -o jsonpath="{.data.password}" | base64 --decode

CONVERT THE PASSOWRDS TO BASE64 FORMAT FIRST.

echo -n AABBCCCDDDACCESSKEY | base64

echo -n AABBCCCDDDSECRETKEY | base64

apiVersion: v1

kind: Secret

metadata:

name: aws-access-keys

type: Opaque

data:

AWS\_ACCESS\_KEY\_ID: QUtJQVFFTUlKQllGNVNSWVhaSUgK

AWS\_SECRET\_ACCESS\_KEY: RUh5MTkzTjBoV2oxMmxqaFduZFBoeW0xU1E1SVp2ZEtaL002U1hBRw==

kubectl create configmap awsregion --from-literal=AWS\_DEFAULT\_REGION='us-east-1'

kubectl create secret generic aws-access-key --from-literal=key1=AKIAQEMIJBYFUZQBCVUT

kubectl create secret generic aws-secret-key --from-literal=key1=NTZvZ0FuOHZkRkpkcGg5emFhdWtMWEZ1d3RCRi9sR0dXajNRUWRTcA==

kubectl create secret generic aws-region --from-literal=key1=us-east-1

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

echo -n AKIAQEMIJBYFUZQBCVUT | base64

QUtJQVFFTUlKQllGVVpRQkNWVVQ=

echo -n 56ogAn8vdFJdph9zaaukLXFuwtBF/lGGWj3QQdSp | base64

NTZvZ0FuOHZkRkpkcGg5emFhdWtMWEZ1d3RCRi9sR0dXajNRUWRTcA==

echo -n us-east-1 | base64

dXMtZWFzdC0x

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

apiVersion: apps/v1

kind: Deployment

metadata:

name: awscli-deployment

labels:

app: awscli

spec:

replicas: 1

selector:

matchLabels:

app: awscli

template:

metadata:

labels:

app: awscli

spec:

containers:

- name: awscli

image: sreeharshav/utils:latest

env:

- name: AWS\_ACCESS\_KEY\_ID

valueFrom:

secretKeyRef:

name: aws-access-keys

key: AWS\_ACCESS\_KEY\_ID

- name: AWS\_SECRET\_ACCESS\_KEY

valueFrom:

secretKeyRef:

name: aws-access-keys

key: AWS\_SECRET\_ACCESS\_KEY

- name: AWS\_DEFAULT\_REGION

valueFrom:

secretKeyRef:

name: aws-aws-region-new

key: AWS\_DEFAULT\_REGION

volumeMounts:

- name: one

mountPath: "/tmp/one"

- name: two

mountPath: "/tmp/two"

- name: three

mountPath: "/tmp/three"

volumes:

- name: one

secret:

secretName: aws-access-keys

items:

- key: AWS\_ACCESS\_KEY\_ID

path: AWS\_ACCESS\_KEY\_ID

- name: two

secret:

secretName: aws-access-keys

items:

- key: AWS\_SECRET\_ACCESS\_KEY

path: AWS\_SECRET\_ACCESS\_KEY

- name: three

secret:

secretName: aws-aws-region-new

items:

- key: AWS\_DEFAULT\_REGION

path: AWS\_DEFAULT\_REGION

apiVersion: apps/v1

kind: Deployment

metadata:

name: awscli-deployment

labels:

app: awscli

spec:

replicas: 1

selector:

matchLabels:

app: awscli

template:

metadata:

labels:

app: awscli

spec:

containers:

- name: awscli

image: banst/awscli

#command: ["s3 ls"] #ENTRYPOINT in Dockerfile

args: ["ec2", "describe-instances"] #CMD in Dockerfile

ports:

- containerPort: 3306

env:

- name: AWS\_ACCESS\_KEY\_ID

valueFrom:

secretKeyRef:

name: aws-access-keys

key: AWS\_ACCESS\_KEY\_ID

- name: AWS\_SECRET\_ACCESS\_KEY

valueFrom:

secretKeyRef:

name: aws-access-keys

key: AWS\_SECRET\_ACCESS\_KEY

- name: AWS\_DEFAULT\_REGION

valueFrom:

configMapKeyRef:

name: awsregion

key: AWS\_DEFAULT\_REGION

nano username

nano password

kubectl create secret generic db-user-pass --from-file=./username --from-file=./password

kubectl create secret generic db-user-pass-literal --from-literal=username=devuser --from-literal=password='S!B\\*d$zDsb='