\*\* Use AWS configure command to set up AWS CLI installation. AWS CLI prompts for four pieces of information

The following is the deployment and service YAML to deploy Jenkins in Kubernetes Cluster. Save it with the filename jenkins-eks-deployment.yml.

Execute the kubectl apply command to create deployment and service for Jenkins in the cicd namespace:

kubectl create -f jenkins-eks-deployment.yml

Run the following command to deploy the Amazon EFS CSI driver:

kubectl create -k "github.com/kubernetes-sigs/aws-efs-csi-driver/deploy/kubernetes/overlays/stable/?ref=master"

\*\* Jenkins Deployment with EKS

Here is the YAML for storage class

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: efs-sc

provisioner: efs.csi.aws.com

The following is the YAML for Persistent Volume:

apiVersion: v1

kind: PersistentVolume

metadata:

name: jenkins-efs-pv

spec:

capacity:

storage: 10Gi

volumeMode: Filesystem

accessModes:

- ReadWriteMany

persistentVolumeReclaimPolicy: Retain

storageClassName: jenkins-efs-sc

csi:

driver: efs.csi.aws.com

volumeHandle: fs-0d9e28d3eb84a72d0::fsap-09ed642ba1f93a7e7

Here is the YAML for PVC:

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: jenkins-efs-claim

spec:

accessModes:

- ReadWriteMany

storageClassName: jenkins-efs-sc

resources:

requests:

storage: 10Gi

The following is the YAML for Jenkins on EKS Deployment

apiVersion: apps/v1

kind: Deployment

metadata:

name: blueocean-deployment

labels:

app: blueocean

spec:

template:

metadata:

name: blueocean-pod

labels:

app: blueocean

spec:

securityContext:

fsGroup: 1000

runAsUser: 0

containers:

- name: blueocean

image: jenkinsci/blueocean

ports:

- name: http-port

containerPort: 8080

- name: jnlp-port

containerPort: 50000

volumeMounts:

- name: jenkins-home

mountPath: /var/jenkins\_home

volumes:

- name: jenkins-home

persistentVolumeClaim:

claimName: jenkins-efs-claim

replicas: 1

selector:

matchLabels:

app: blueocean

---

apiVersion: v1

kind: Service

metadata:

name: blueocean-service

labels:

app: blueocean

spec:

selector:

app: blueocean

ports:

- protocol: TCP

name: http

port: 8080

targetPort: 8080

- protocol: TCP

name: agent

port: 50000

targetPort: 50000

type: LoadBalancer

\*\* SonarQube Deployment

apiVersion: apps/v1

kind: Deployment

metadata:

name: sonarqube-deployment

labels:

app: sonarqube

spec:

template:

metadata:

name: sonarqube-pod

labels:

app: sonarqube

spec:

containers:

- name: sonarqube

image: sonarqube:latest

resources:

requests:

cpu: 500m

memory: 1024Mi

limits:

cpu: 2000m

memory: 2048Mi

ports:

- name: sonar-port

protocol: TCP

containerPort: 9000

replicas: 1

selector:

matchLabels:

app: sonarqube

---

apiVersion: v1

kind: Service

metadata:

name: sonarqube-service

labels:

app: sonarqube

spec:

selector:

app: sonarqube

ports:

- name: sonar-port

protocol: TCP

port: 9000

targetPort: 9000

type: LoadBalancer