#########RBAC FOR THE EFS ACCESS##########################

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: efs-provisioner-runner

rules:

- apiGroups: [""]

resources: ["persistentvolumes"]

verbs: ["get", "list", "watch", "create", "delete"]

- apiGroups: [""]

resources: ["persistentvolumeclaims"]

verbs: ["get", "list", "watch", "update"]

- apiGroups: ["storage.k8s.io"]

resources: ["storageclasses"]

verbs: ["get", "list", "watch"]

- apiGroups: [""]

resources: ["events"]

verbs: ["create", "update", "patch"]

- apiGroups: [""]

resources: ["endpoints"]

verbs: ["get", "list", "watch", "create", "update", "patch"]

---

kind: ClusterRoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: run-efs-provisioner

subjects:

- kind: ServiceAccount

name: efs-provisioner

# replace with namespace where provisioner is deployed

namespace: default

roleRef:

kind: ClusterRole

name: efs-provisioner-runner

apiGroup: rbac.authorization.k8s.io

---

kind: Role

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: leader-locking-efs-provisioner

rules:

- apiGroups: [""]

resources: ["endpoints"]

verbs: ["get", "list", "watch", "create", "update", "patch"]

---

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: leader-locking-efs-provisioner

subjects:

- kind: ServiceAccount

name: efs-provisioner

# replace with namespace where provisioner is deployed

namespace: default

roleRef:

kind: Role

name: leader-locking-efs-provisioner

apiGroup: rbac.authorization.k8s.io

---

apiVersion: v1

kind: ServiceAccount

metadata:

name: efs-provisioner

namespace: default

################EFS-PROVISIONER#############################

---

apiVersion: v1

kind: ConfigMap

metadata:

name: efs-provisioner

data:

file.system.id: fs-7e4708ca

aws.region: us-east-1

provisioner.name: example.com/aws-efs

dns.name: fs-7e4708ca.efs.us-east-1.amazonaws.com

---

kind: Deployment

apiVersion: apps/v1

metadata:

name: efs-provisioner

spec:

replicas: 1

selector:

matchLabels:

app: efs-provisioner

strategy:

type: Recreate

template:

metadata:

labels:

app: efs-provisioner

spec:

serviceAccount: efs-provisioner

containers:

- name: efs-provisioner

image: quay.io/external\_storage/efs-provisioner:latest

env:

- name: FILE\_SYSTEM\_ID

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: file.system.id

- name: AWS\_REGION

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: aws.region

- name: DNS\_NAME

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: dns.name

optional: true

- name: PROVISIONER\_NAME

valueFrom:

configMapKeyRef:

name: efs-provisioner

key: provisioner.name

volumeMounts:

- name: pv-volume

mountPath: /persistentvolumes

volumes:

- name: pv-volume

nfs:

server: fs-7e4708ca.efs.us-east-1.amazonaws.com

path: /

---

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: aws-efs

provisioner: example.com/aws-efs

#######################PVC##################################

---

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: efs-1

spec:

storageClassName: aws-efs

accessModes:

- ReadWriteMany

resources:

requests:

storage: 1Gi

#####################DEPLOYMENT##############################

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: sreeharshav/rollingupdate:v3

ports:

- containerPort: 80

apiVersion: v1

kind: Service

metadata:

labels:

app: nginx

name: nginx-deployment

annotations:

service.beta.kubernetes.io/aws-load-balancer-type: "nlb"

spec:

ports:

- port: 80

protocol: TCP

targetPort: 80

selector:

app: nginx

type: LoadBalancer

################################################################

<https://github.com/kubernetes-incubator/external-storage/tree/master/aws>

https://thenewstack.io/overcome-stuck-ebs-volumes-running-stateful-containers-aws/