Use AWS EFS for Kubernetes

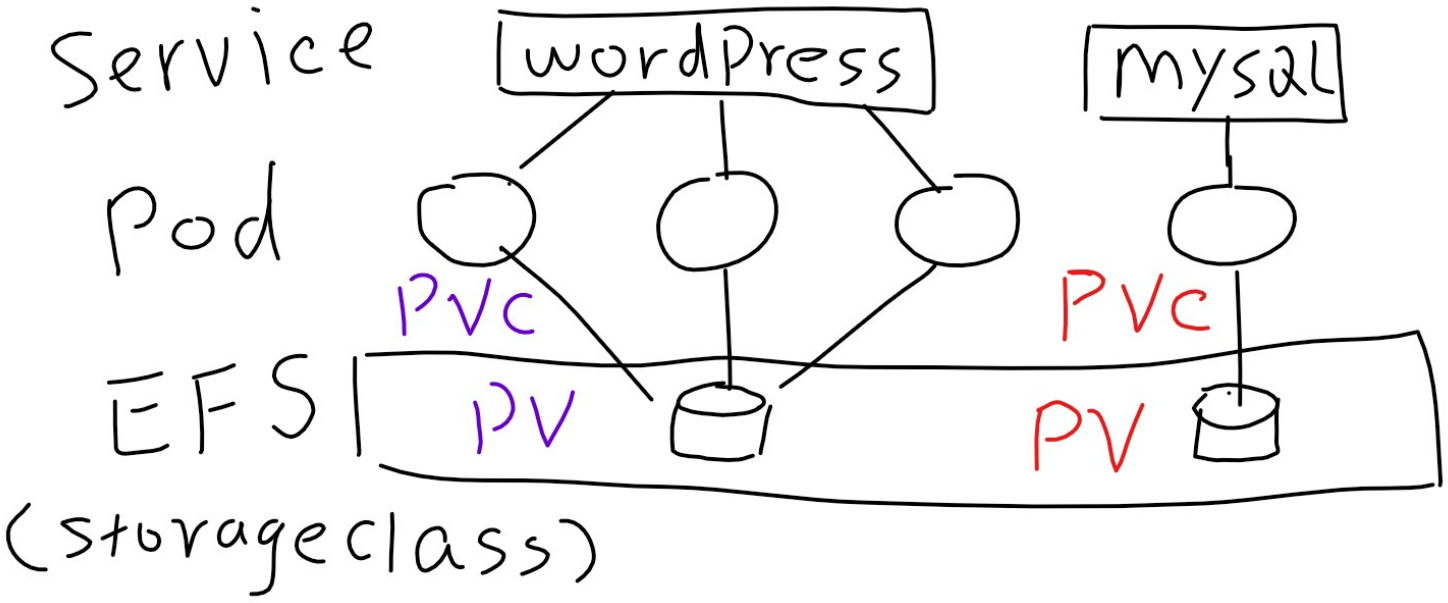
Jackie Chen Cloud, Storage May 1, 2018May 1, 2018 2 Minutes

i
1 Vote

Before introducing how to use AWS EFS for Kubernetes, let me recap some terms of Persistent Volumes (https://kubernetes.io/docs/concepts/storage/persistent-volumes/).

- PV (Persistent Volume): PV is a piece of storage, it can be NFS, iSCSI, EBS, EFS… The purpose of having PV is to decouple the storage from pod's lifecycle.
- PVC (Persistent Volume Claim): PVC provides the method for pods to use PV, it includes the request storage size and access mode. If the PV supports ReadWriteMany access mode, then its PVC can be used by multiple pods.
- Storage Class: This is the abstract layer of PV which hides the implementation of PV from end users.

With the above 3 points in your mind, lets have a look the architecture.



Basically, there are four steps:

1. Create EFS (only the first time): It includes the tasks: create the EFS in the right subnets, setup the security groups to allow Kubernetes nodes to access and enable DNS support/resolution in your VPC.
2. Create StorageClass for EFS via efs-provisioner (https://github.com/kubernetes-incubator/external-storage/tree/master/aws/efs) (only the first time). efs-provisioner runs as a container which plays the role of EFS broker. It allows other pods to mount EFS as the persistent volumes. Just be aware of that EFS is built on top of NFS4, so you need to have nfs-common packages installed in your Kubernetes nodes
3. Create a PVC to use the StorageClass for EFS. Just note that EFS has unlimited storage, so the storage size request actually does not take any effects here. But you still have to keep it to pass the syntax check.
4. Create a volume for the PVC, then mount the volume inside the pod.

Here is how the PVC and PV look. And I also attached the sample code as below.

```
vagrant@xenial:~$ kubectl get pvc
NAME                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
efs-web-0           Bound     pvc-0b72112c-4c2f-11e8-bfb3-0605599b3d50   1Mi        RWO            aws-efs        1d
efs-wordpress       Bound     pvc-e351c723-4ce8-11e8-bc93-02305208901e   1Gi        RWX            aws-efs        8h
efs-wordpress-db-0  Bound     pvc-aa420396-4ce8-11e8-a8eb-06522142c018   1Gi        RWX            aws-efs        8h
efs-wordpress-db-1  Bound     pvc-b444b5bd-4ce8-11e8-a8eb-06522142c018   1Gi        RWX            aws-efs        8h
vagrant@xenial:~$ kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS    CLAIM                          STORAGECLASS   REASON    AGE
pvc-0b72112c-4c2f-11e8-bfb3-0605599b3d50   1Mi        RWO            Delete           Bound     default/efs-web-0              aws-efs                  1d
pvc-aa420396-4ce8-11e8-a8eb-06522142c018   1Gi        RWX            Delete           Bound     default/efs-wordpress-db-0     aws-efs                  8h
pvc-b444b5bd-4ce8-11e8-a8eb-06522142c018   1Gi        RWX            Delete           Bound     default/efs-wordpress-db-1     aws-efs                  8h
pvc-e351c723-4ce8-11e8-bc93-02305208901e   1Gi        RWX            Delete           Bound     default/efs-wordpress          aws-efs                  8h
```

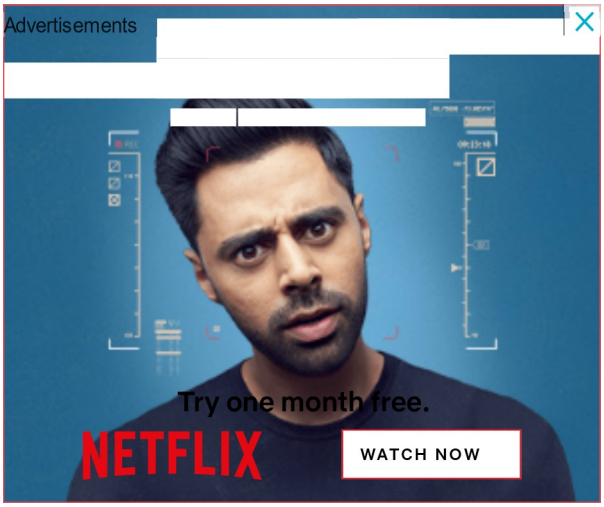
(https://jackiechendotorg.files.wordpress.com/2018/05/pv1.png)

```
---
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: aws-efs
provisioner: example.com/aws-efs
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: wordpress
data:
  database: wordpress-db
  password: ************
---
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
  - port: 80
  selector:
    app: wordpress
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: efs-wordpress
  annotations:
    volume.beta.kubernetes.io/storage-class: "aws-efs"
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
```

```
spec:
  replicas: 2
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
      - image: wordpress:latest
        name: wordpress
        env:
        - name: WORDPRESS_DB_HOST
          valueFrom:
            configMapKeyRef:
             name: wordpress
             key: database
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            configMapKeyRef:
              name: wordpress
              key: password
        ports:
        - containerPort: 80
          name: wordpress
        volumeMounts:
        - name: wordpress-pv
          mountPath: /var/www/html
      volumes:
      - name: wordpress-pv
        persistentVolumeClaim:
          claimName: efs-wordpress
---
```

AdChoices

ADVERTISING



REPORT THIS AD

REPORT THIS AD

Tagged:
AWS,
EFS,
Kubernetes

Published by Jackie Chen

View all posts by Jackie Chen

Powered by WordPress.com.

REPORT THIS AD


Jackie Chen
WorkShop