KUBERNETES[TASK-08]

**1)Create and Test a Kubernetes Pod with an EmptyDir Volume**

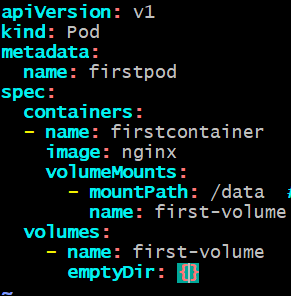
EmptyDir

*We create volume inside a pod to store data related to container.*

*If the container is killed for any reason, a new container will be created in the same pod and*

*the same volume will be attached to container.*

Created a script



-kubectl apply -f <filename>

1) Login to container and create some random files in /data location.

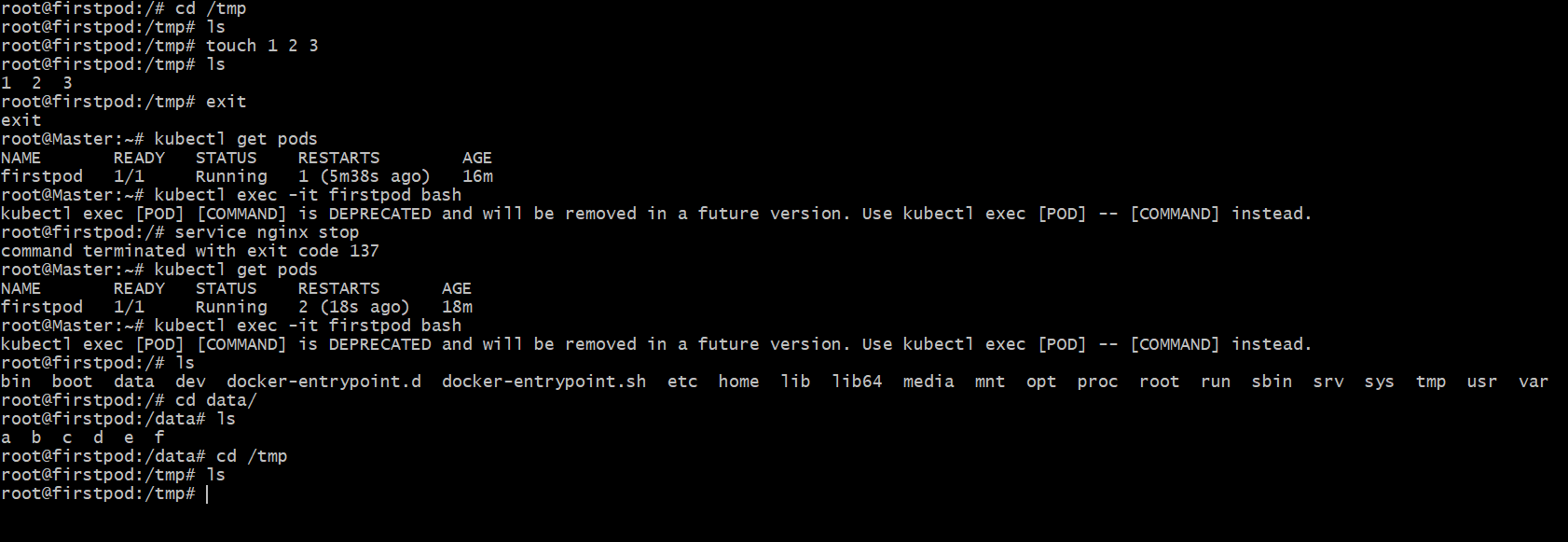
2) Create some random files in tmp location.

3) Stop nginx service to terminate your container.

4) Now you can see pod will create a new container and try to login into container

5) Validate the files in both the locations.

6) Files in directory /data will be visible.



**2) Configure a HostPath Volume in Kubernetes and Validate Data Persistence**

**Created a hostpath.yaml**

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

volumeMounts:

- mountPath: /data

name: first-volume

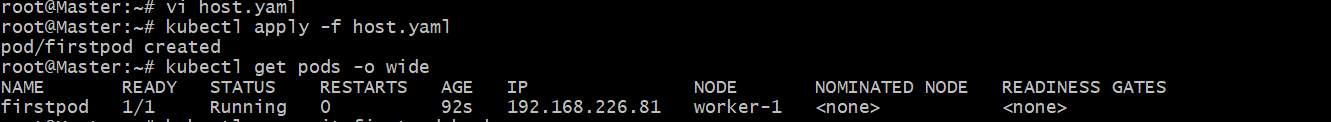
volumes:

- name: first-volume

hostPath:

path: /tmp/data

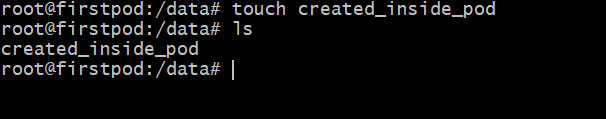
-kubectl apply -f hostpath.yaml[firstpod will be created]



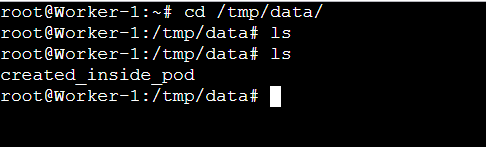
Hostpath got created in the worker-1



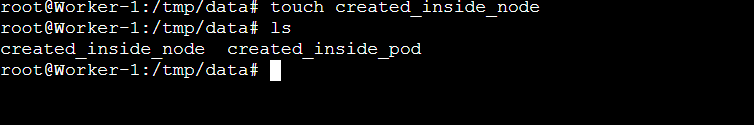
Now created a file in master node in the container

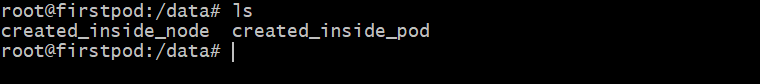


We can see that file will be stored in worker-1 node

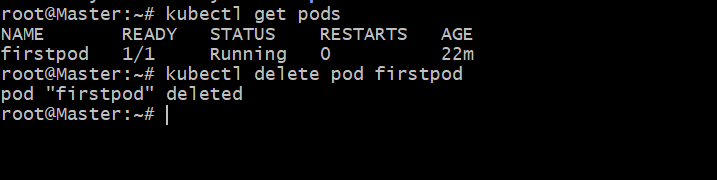


And also,if we created file in the node ,it will get stored in the master

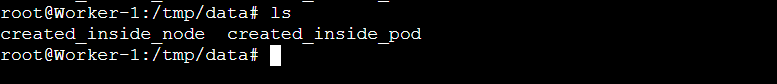




Deleted the pod



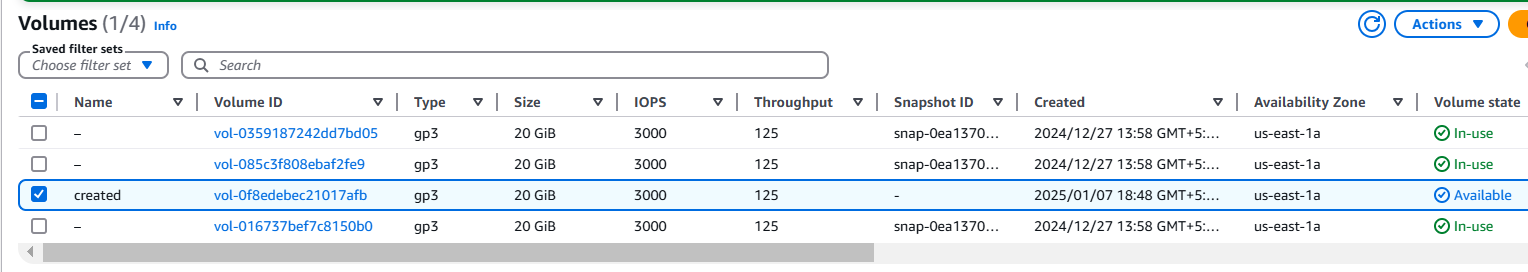
After deleting the pod ,we can see the volumes in the node[worker-1] machine



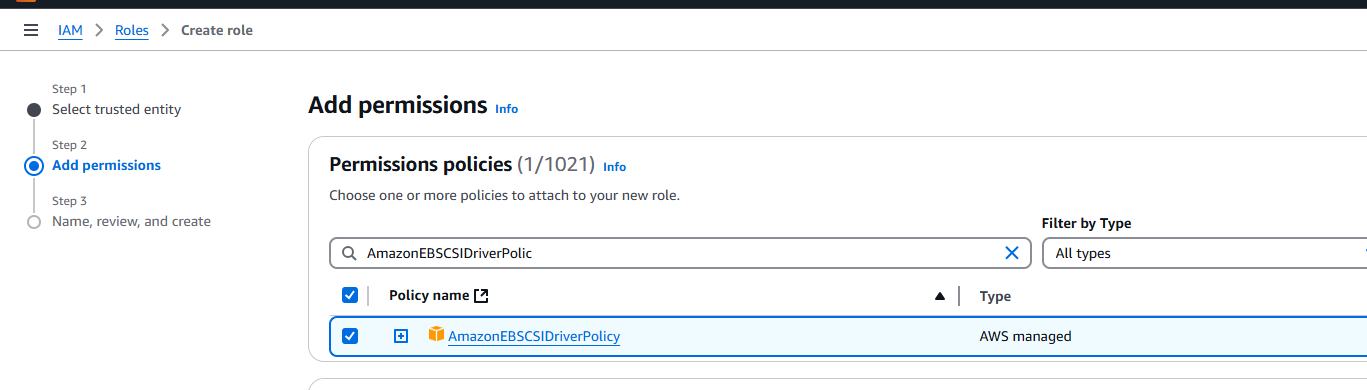
**3) Deploy an Amazon EBS Volume Using Persistent Volume and Persistent Volume Claim (PVC)**

Steps:

Created a volume

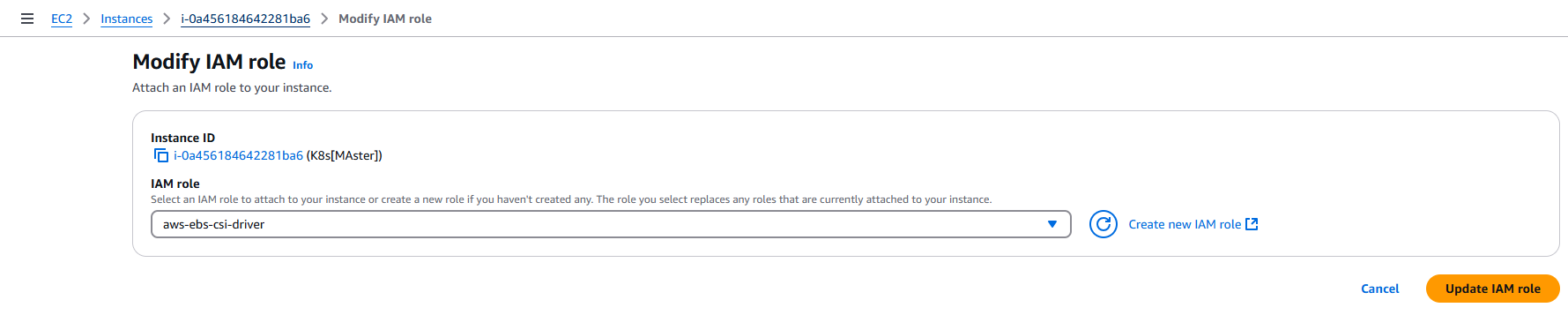


Created role

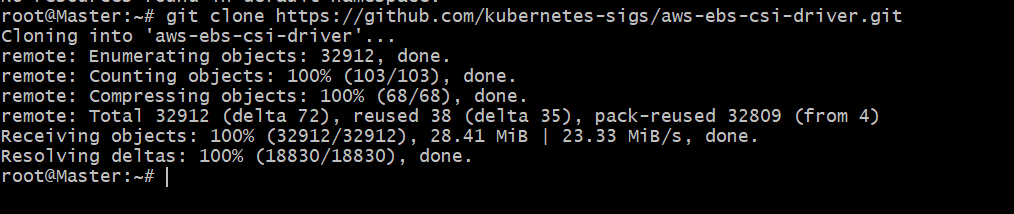


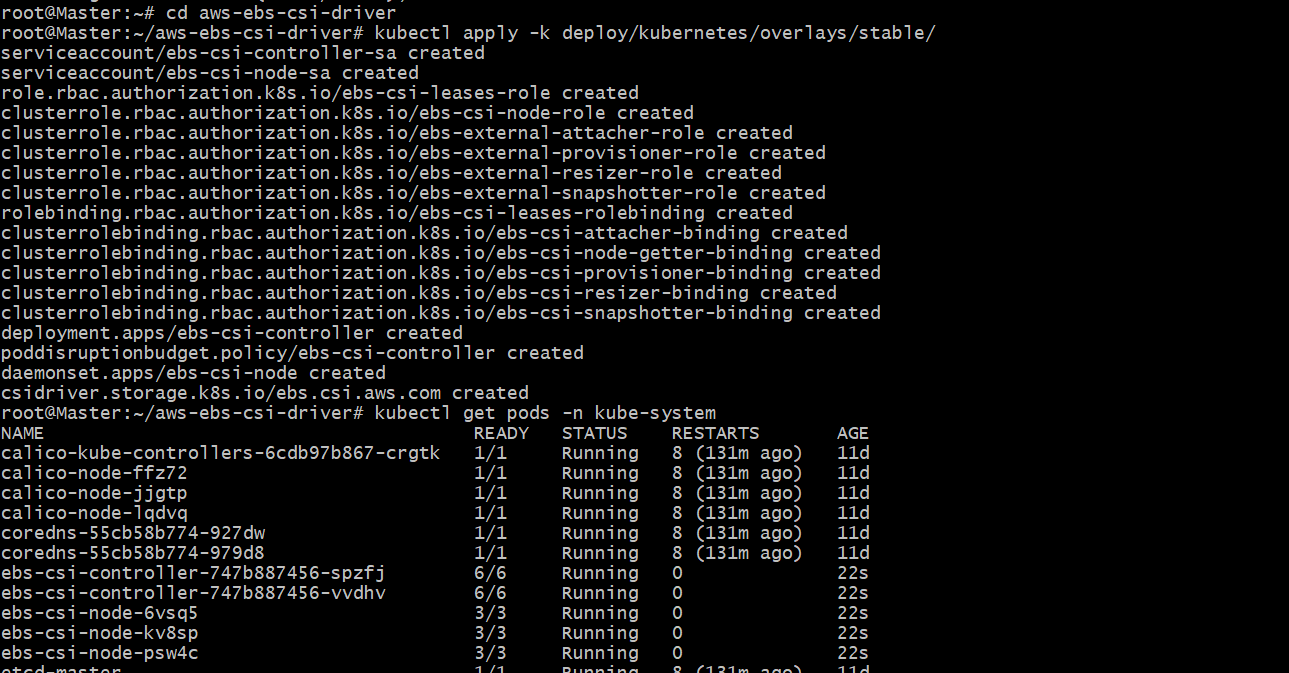
Next,

Modifying IAM role for all Nodes[Master,Worker-1,Worker-2]



Now ,in master machine





4) Now we need to create one Persistent volume for our EBS.

apiVersion: v1

kind: PersistentVolume

metadata:

name: my-ebs-pv

spec:

capacity:

storage: 10Gi

accessModes:

- ReadWriteOnce

persistentVolumeReclaimPolicy: Retain

storageClassName: aws-ebs

awsElasticBlockStore:

volumeID: <your-ebs-volume-id> #EBS volume ID

fsType: ext4

5) We need to create on claim for Persistent volume (PVC) for our EBS.

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: my-ebs-pvc

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 10Gi

storageClassName: aws-ebs

6) Create pod with PVC.

apiVersion: v1

kind: Pod

metadata:

name: my-pod

spec:

containers:

- name: my-container

image: nginx

volumeMounts:

- name: data-volume

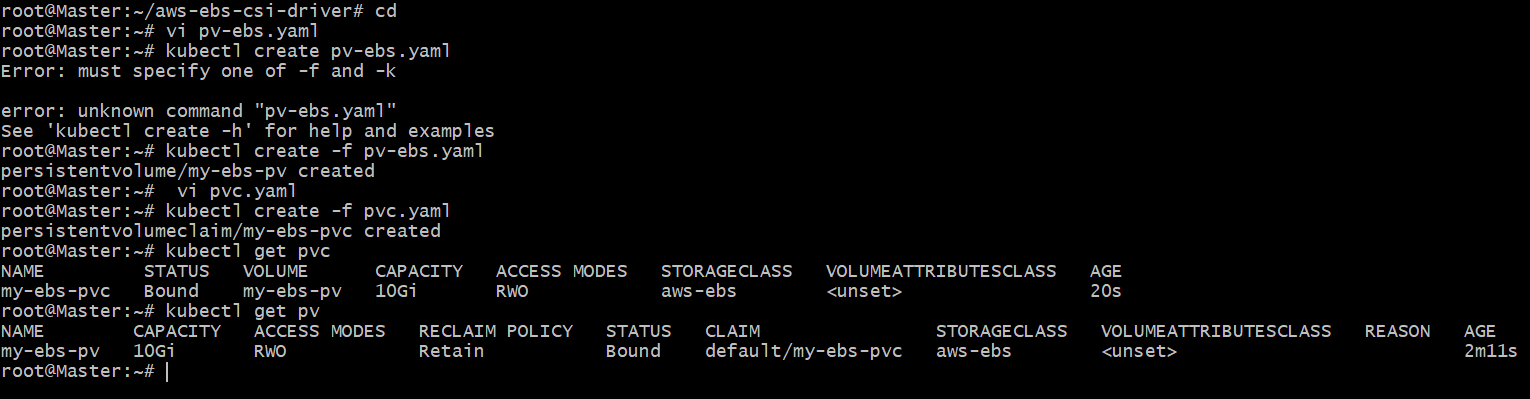
mountPath: /data #data where we can write data

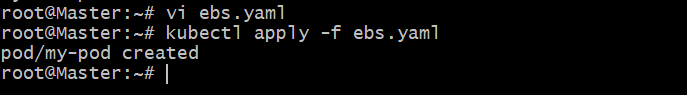
volumes:

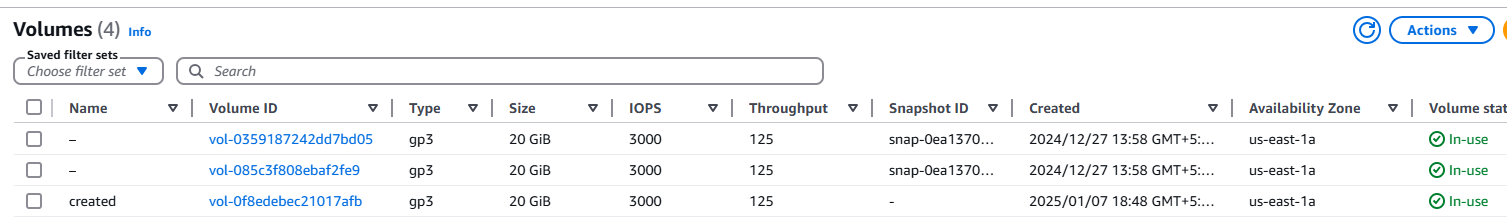
- name: data-volume

persistentVolumeClaim:

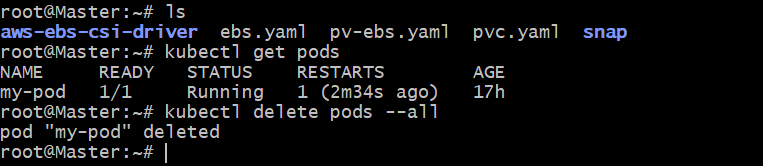
claimName: my-ebs-pvc



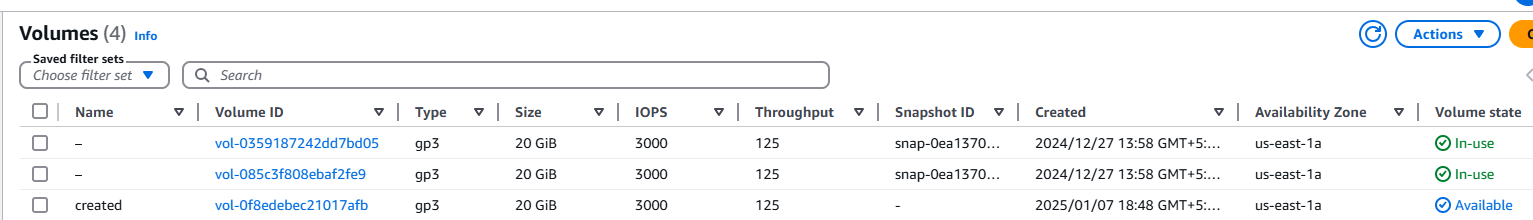




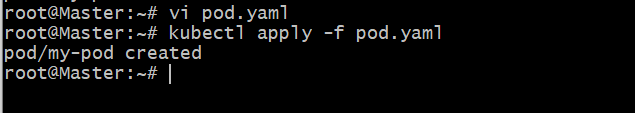
If we delete the pod

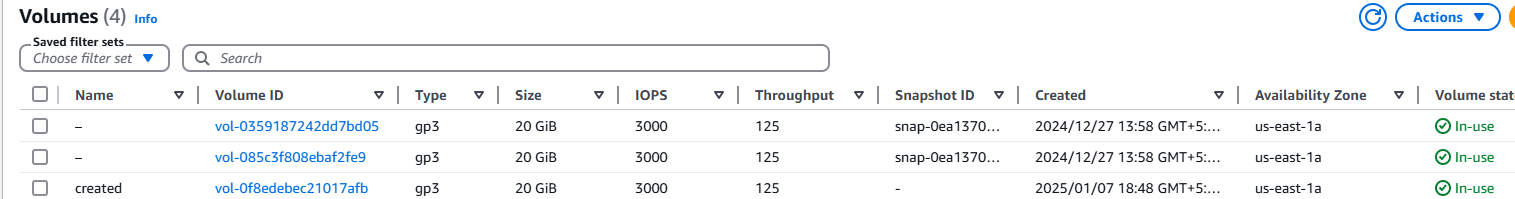


Then volume will be in “available”



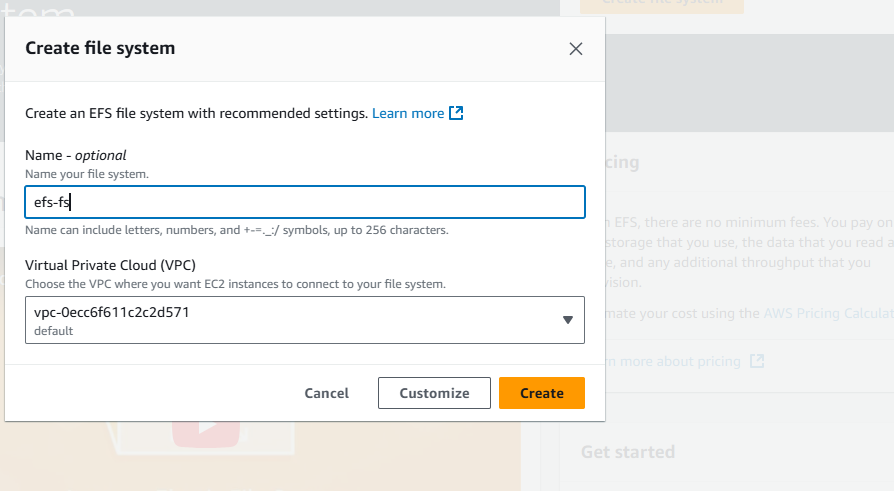
Create new pod again and this time if it is scheduled on another node then volume will be attached to that node.





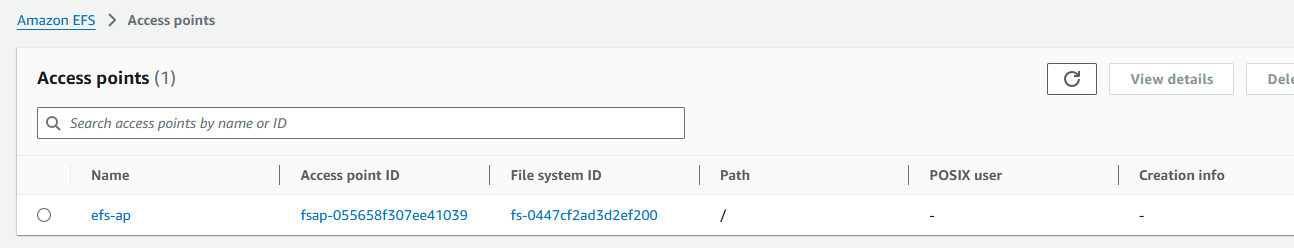
**4) Set Up an Amazon EFS Volume and Attach it to Multiple Pods**

Created a EFS



Step-2:

Creating access point



Step-3

3)Install AWS EFS CSI driver on k8s master node.

#) git clone https://github.com/kubernetes-sigs/aws-efs-csi-driver.git

Apply the CSI Driver Manifests: Navigate to the cloned repository and apply the

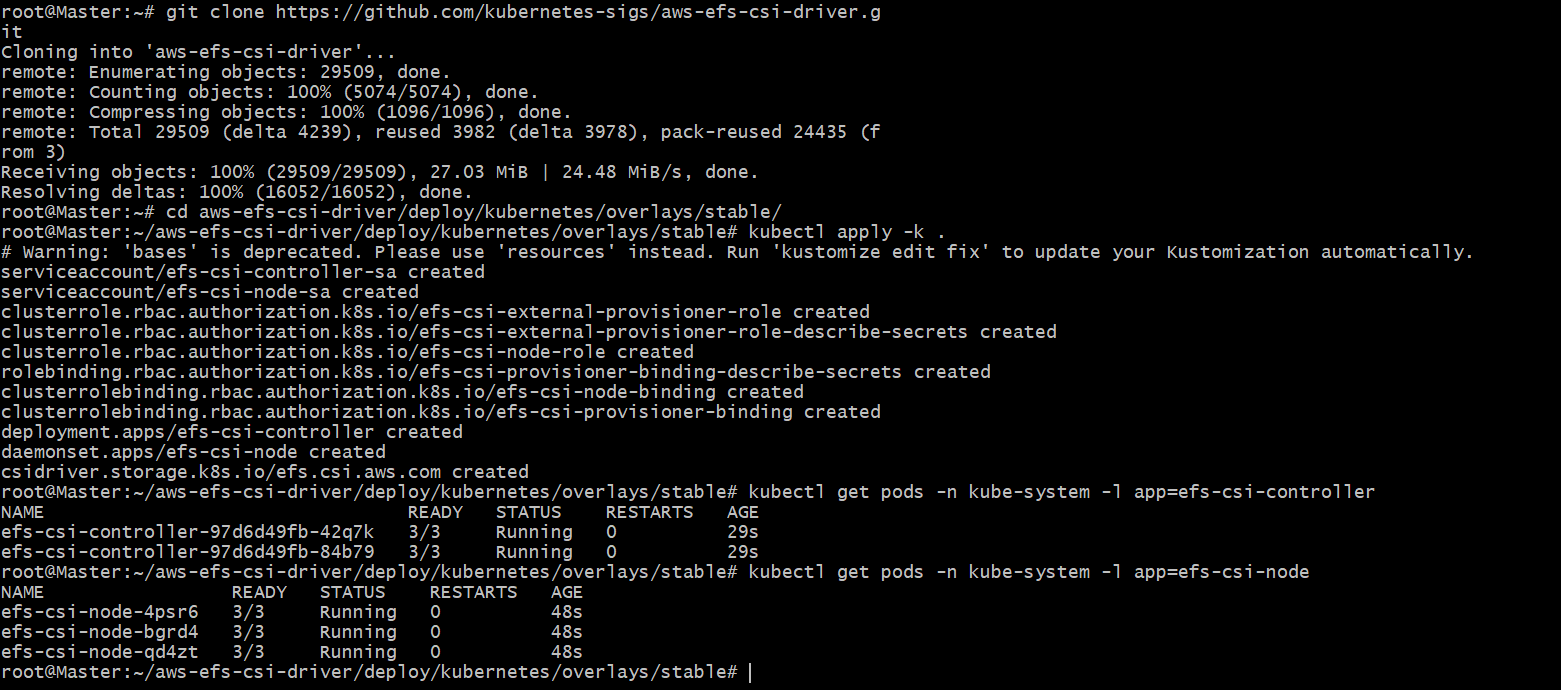
Kubernetes manifests using the kubectl apply command:

# cd aws-efs-csi-driver/deploy/kubernetes/overlays/stable/

# kubectl apply -k .

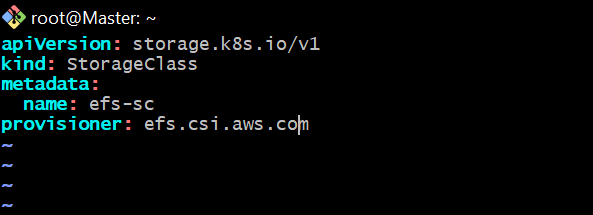
verify the installation.

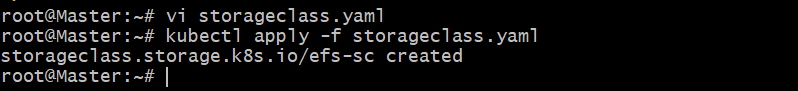
#kubectl get pods -n kube-system -l app=efs-csi-controller

#kubectl get pods -n kube-system -l app=efs-csi-node

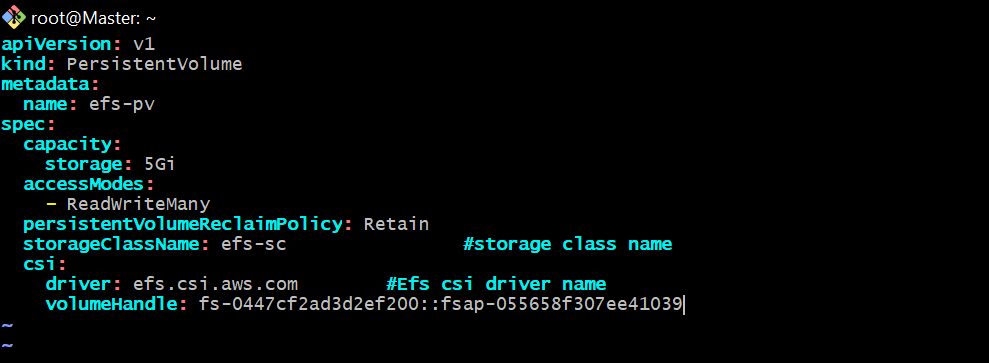
Create EFS Provisioner Storage Class: To dynamically provision EFS volumes,

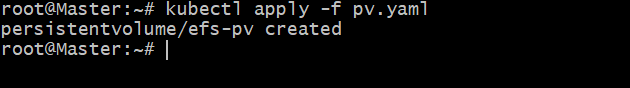
you need to create a StorageClass that uses the EFS CSI driver.



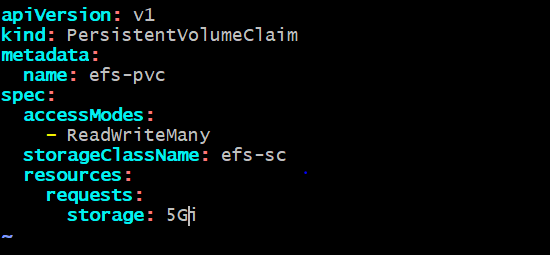


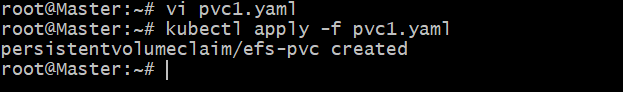
4) Now we need to create one Persistent volume for our EFS.



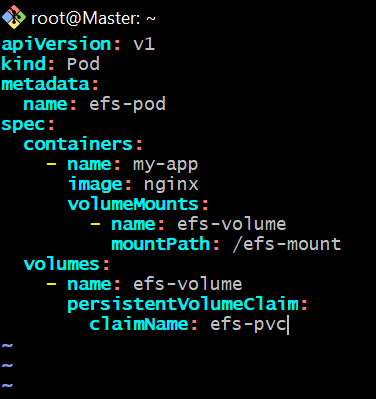


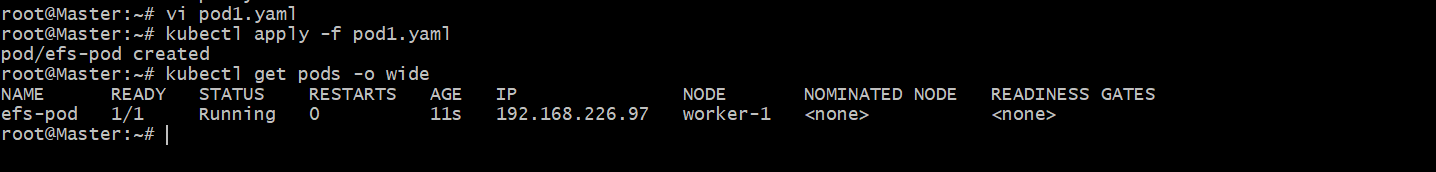
5) We need to create on claim for Persistent volume (PVC) for our EBS.





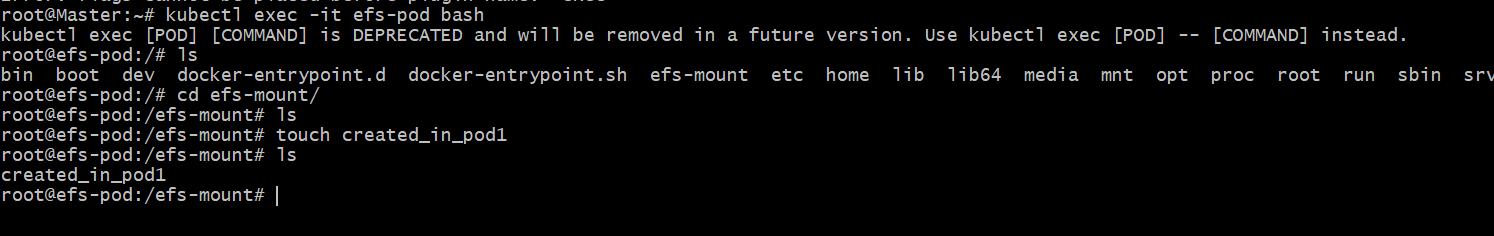
6) Create pod with PVC.



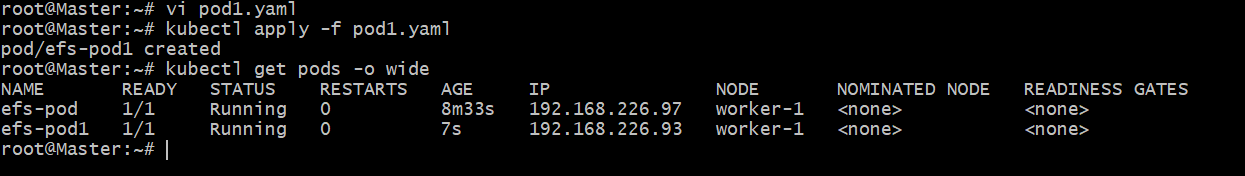


How to test:

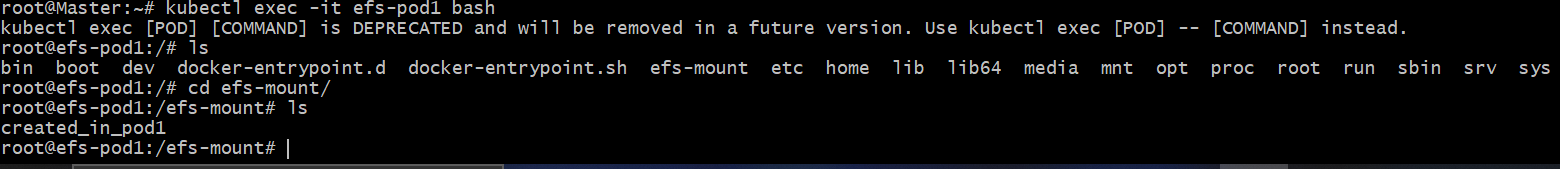
1) Create one pod and login to pod and create some random files in efs-mount directory.



2) Create one more pod and check if the files are available or not



The file created in pod1 is avilabe in pod2



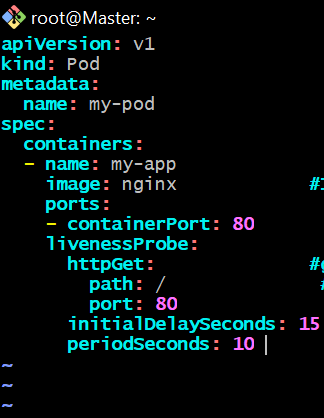
**5) Implement and Test Liveness and** **Readiness Probes in a Kubernetes Pod**

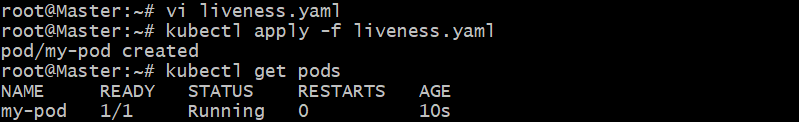
Liveness Probes

Liveness can use with pod configuration to check the health status of pod.

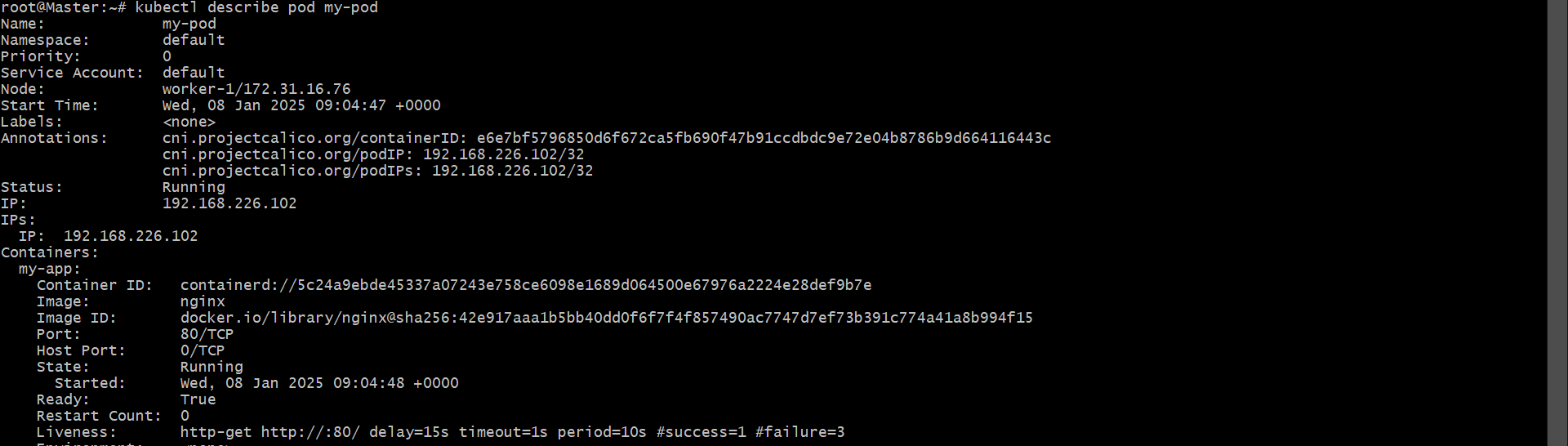
If pod in not healthy for any reason then Liveness will restart the pod.

Created a file[vi liveness.yaml]





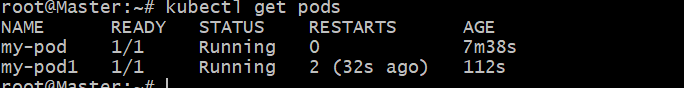
Using “kubectl describe pod my-pod”

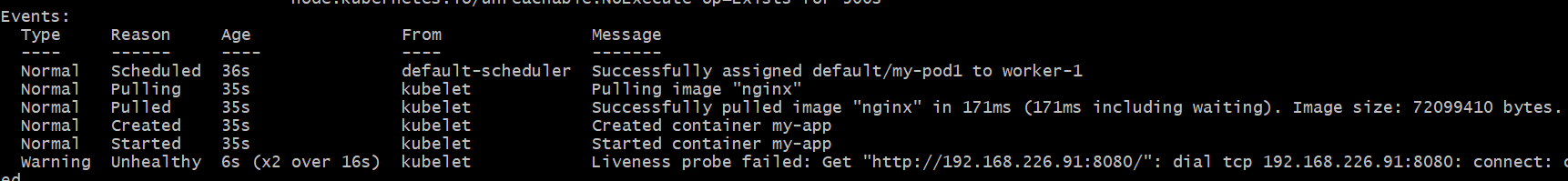


Testing

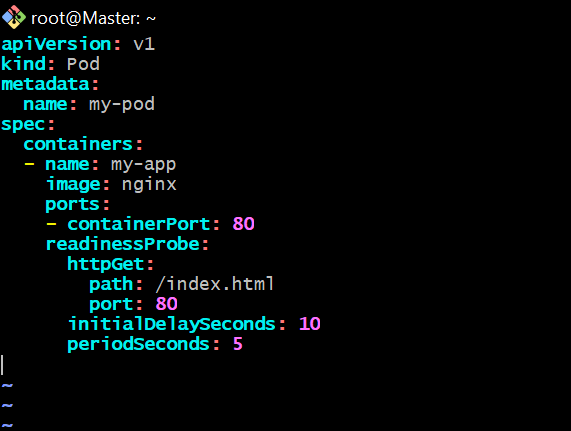
Create pod using above yaml and use describe to check the liveness.

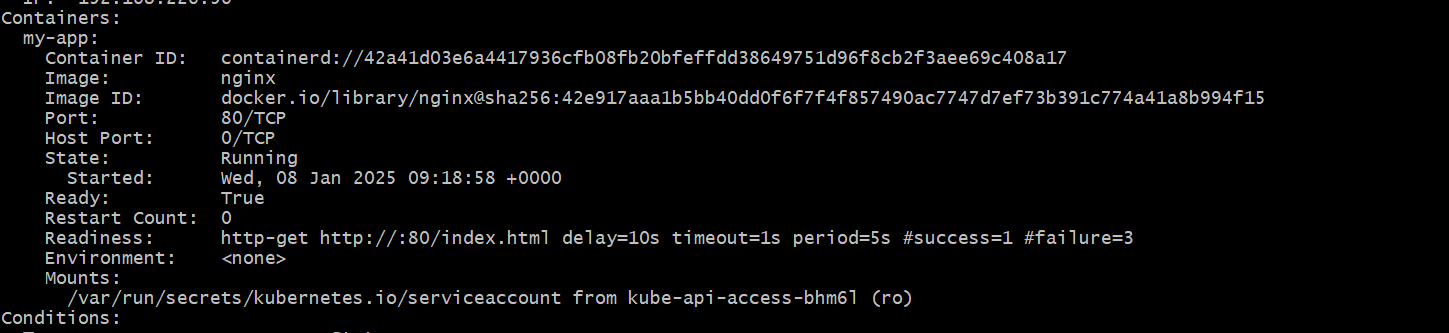
or create one yaml with wrong container ports.

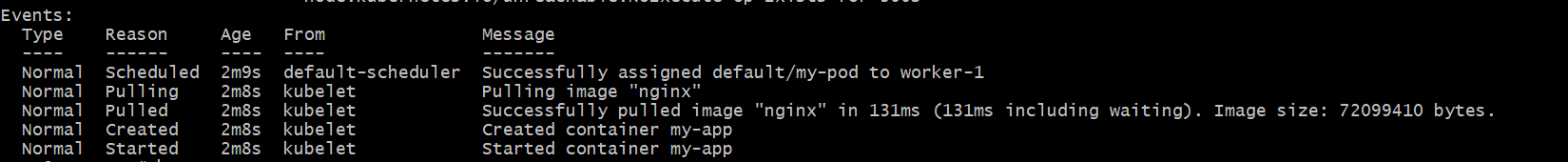




Readiness Probes





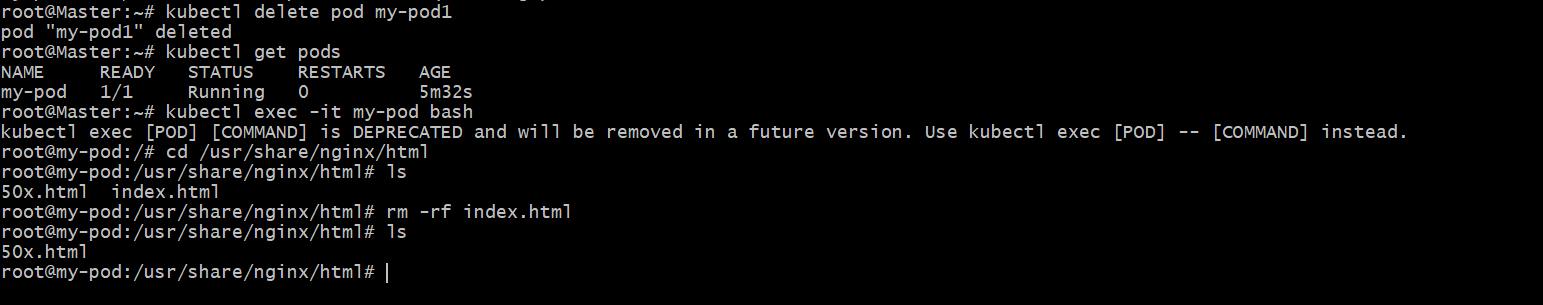


Test:

Login to the pod and delete the index.html file

Execute kubectl get pods -wait

We can now see pod is not ready status.



Kubectl describe pod my-pod

