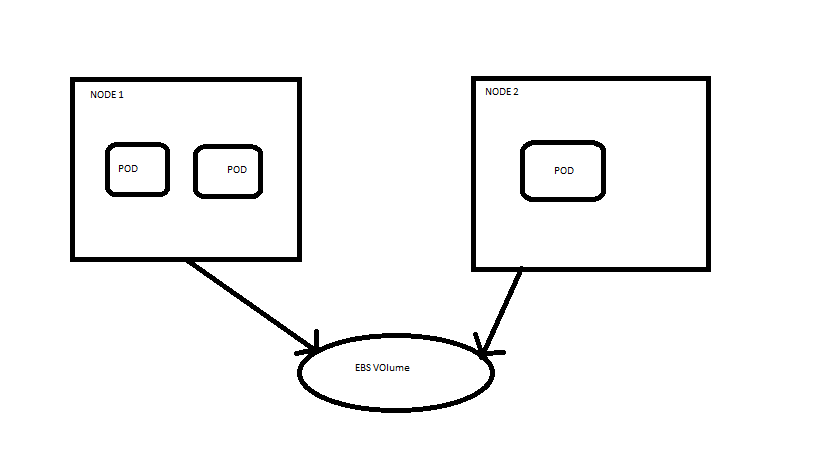
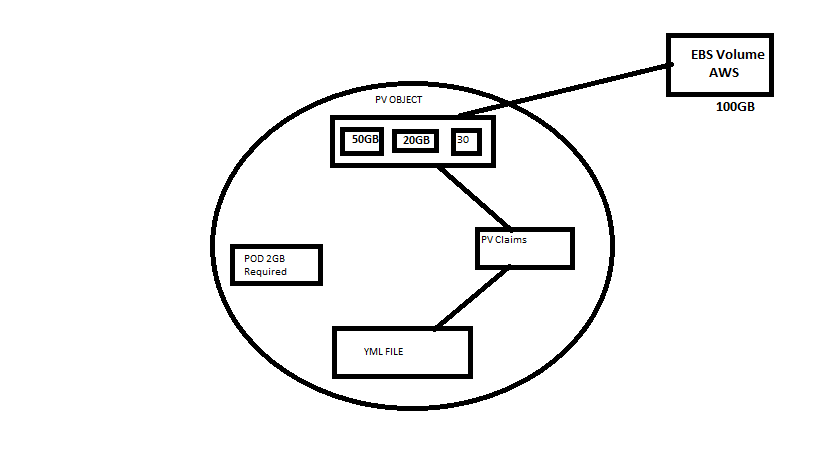
***Persistent Volume and LivenessProbe***

The concept of this topic is lastly we are create the volume in pods the container can easily share and store data in volume. which are mounted between the containers is specific path. If any containers fail and go down so new container is create and get the data from the define volume path in yml file.  
  
But what if the pods is fail and create in other node in cluster so it cannot get the same volume data easily accessible. Wo we learn in this topic how to mount pod volume if it’s created on other nodes. this method is called Persistent volume





***LABS***

***Note:***

* Create EBS on same availability zone where your instance is running.

1. **How to make PV on k8s**

* nano pv.yml

apiVersion: v1

kind: PersistentVolume

metadata:

name: myebsvol

spec:

capacity:

storage: 1Gi

accessModes:

- ReadWriteOnce

persistentVolumeReclaimPolicy: Recycle

awsElasticBlockStore:

volumeID: # YAHAN APNI EBS VOLUME ID DAALO

fsType: ext4

kubectl get pv // to check pv is created

1. **How to claim memory from PV**

//since you have create the pv now you want to claim some memory from the pv is called pv-claims

* nano mypvclaim.yml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: myebsvolclaim

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 1Gi

* kubectl apply -f mypvclaim.yml
* kubectl get pvc /to check pvc



1. **How to attached pv-claim to pod**

* nano pod\_pvc.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: pvdeploy

spec:

replicas: 1

selector: # tells the controller which pods to watch/belong to

matchLabels:

app: mypv

template:

metadata:

labels:

app: mypv

spec:

containers:

- name: shell

image: centos

command: ["bin/bash", "-c", "sleep 10000"]

volumeMounts:

- name: mypd

mountPath: "/tmp/persistent"

volumes:

- name: mypd

persistentVolumeClaim:

claimName: myebsvolclaim

* kubectl apply -f pod\_pvc.yml
* kubectl get deploy
* kubectl get rs
* kubectl get pods



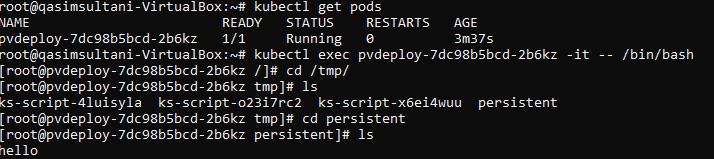
**//now go into the pod and create the file in the path where your pv mounted and delet the pod after creating file**

* kubectl get pods
* kubectl exec pvdeploy-7dc98b5bcd-sc9pq -it -- /bin/bash
* cd tmp/ persistent
* touch hello



//exit from pod & delet pod check the pod id because new pod is create with other id and the data is available on the same path because of replica to create new pods and pv for the data available on the path

* kubectl delete pod pvdeploy-7dc98b5bcd-sc9pq
* kubectl get pods
* kubectl exec pvdeploy-7dc98b5bcd-2b6kz -it -- /bin/bash
* cd tmp/ persistent
* ls



//Exit from pod

//delete pv.pvclaim & pod create to pv yml file

* kubectl delete -f pod\_pvc.yml
* kubectl delete -f pv.yml
* kubectl delete -f mypvclaim.yml

**Healthcheck/ LivenessProbe**

* It is used to check either our application is working fine in our pods through commands in ymls file.
* We do’not only check is your container is created and pods are ready. We also check that our application is also working.
* It’s work like if the given command is not work in application to check application is working or not it will automatically delete pod and create new pod.
* nano livness.yml

apiVersion: v1

kind: Pod

metadata:

labels:

test: liveness

name: mylivenessprobe

spec:

containers:

- name: liveness

image: ubuntu

args:

- /bin/sh

- -c

- touch /tmp/healthy; sleep 1000

livenessProbe:

exec:

command:

- cat

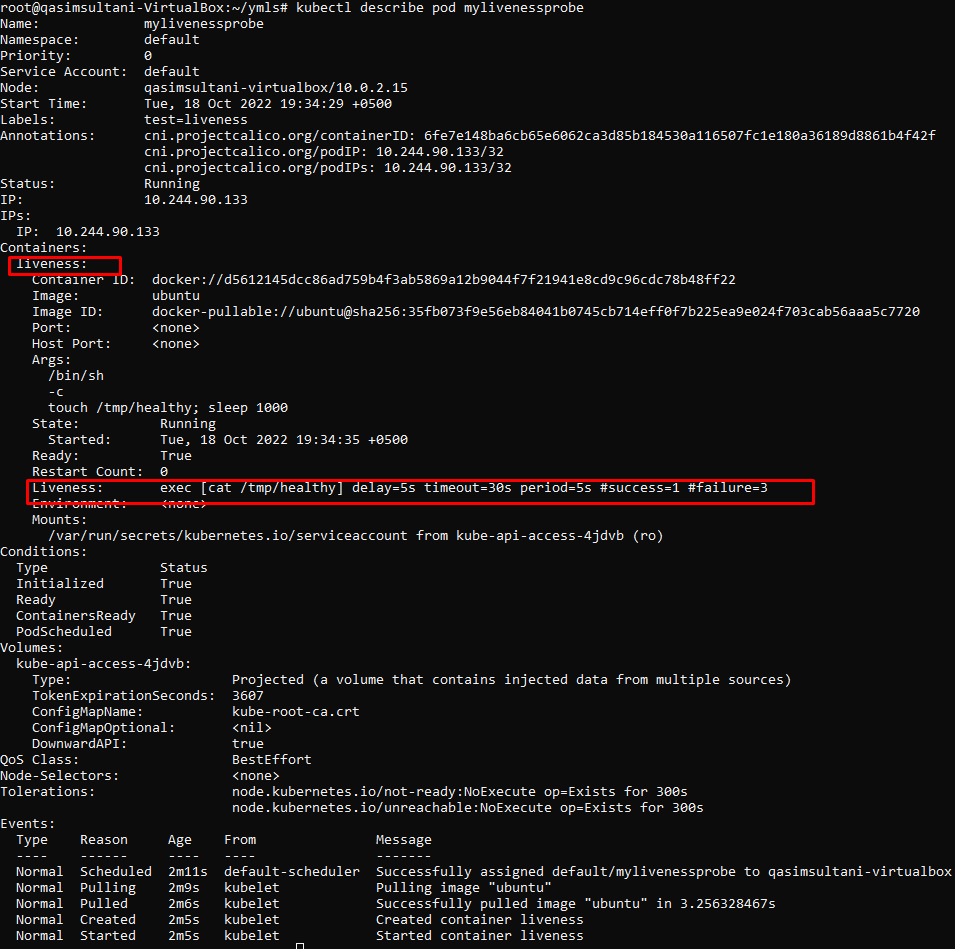
- /tmp/healthy

initialDelaySeconds: 5 //check 5 sec later when created pods & container run

periodSeconds: 5 // next check every 5 sec

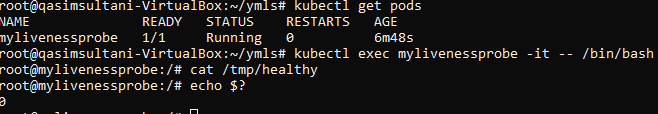
timeoutSeconds: 30 //if did not get any respons under 30 sec create new container/pod

* kubectl apply -f liveness.yml
* kubectl get pods
* kubectl describe pod mylivenessprobe



//failure=3 means “it will check 3 time if the specific command did not found continuously it will create new and delete pervious pod ”

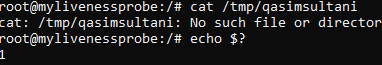
**//now enter in the pod and search file you make from command in yml file if the file is here it’s give output “0” if the file is not output “1” means wrong**



* kubectl get pods
* kubectl exec mylivenessprobe -it -- /bin/bash
* cat /tmp/healthy //this health file is in yml file
* echo $? /to check output

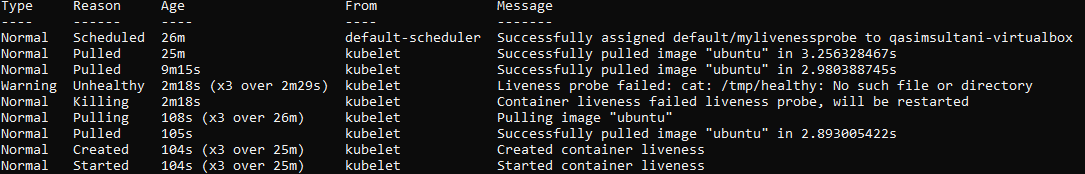
//now we give false file to check output

* cat /tmp/qasimsultani
* echo $?



// now we check if the output is not 0 or 1 it’s mean the file is delete or something wrong it’s give “non-zero & 1 value” output

* rm –rf /tmp/healthy
* cat /tmp/healthy
* echo $?
* kubectl describe pod mylivenessprobe



//you can see that pod is created and then Liveness pod failed due to not match command and then again created pod successfully