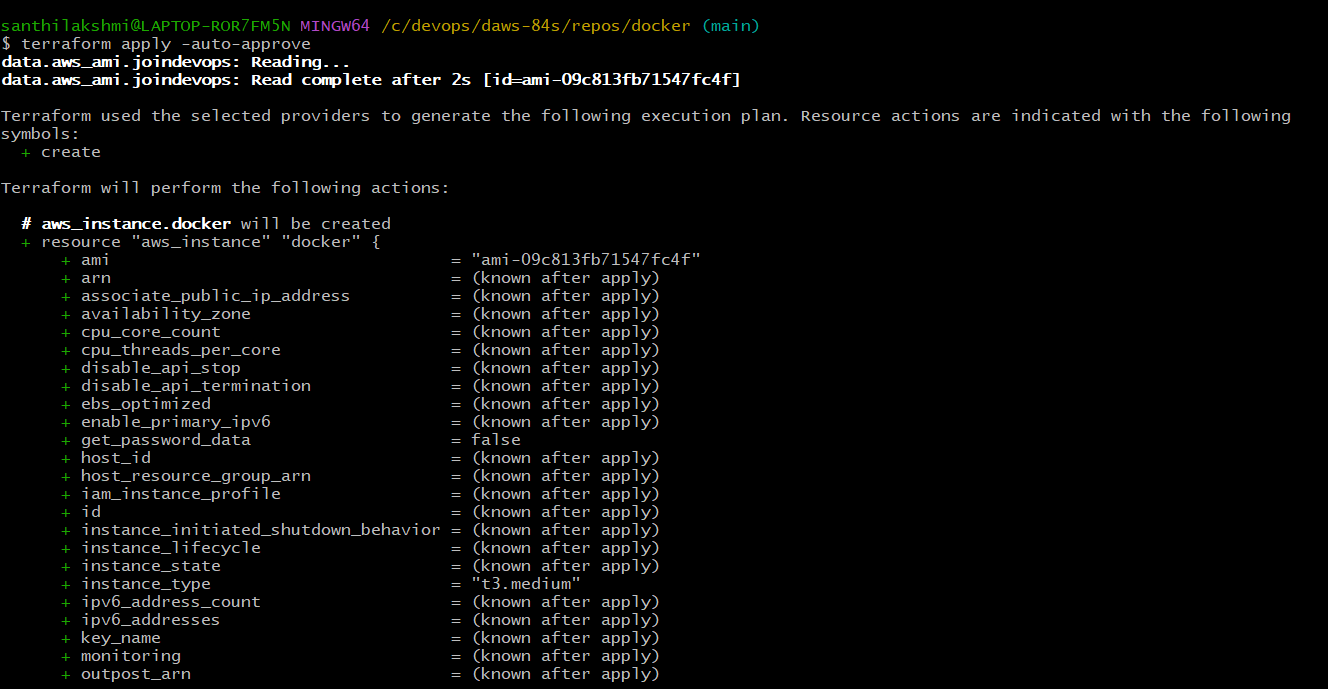
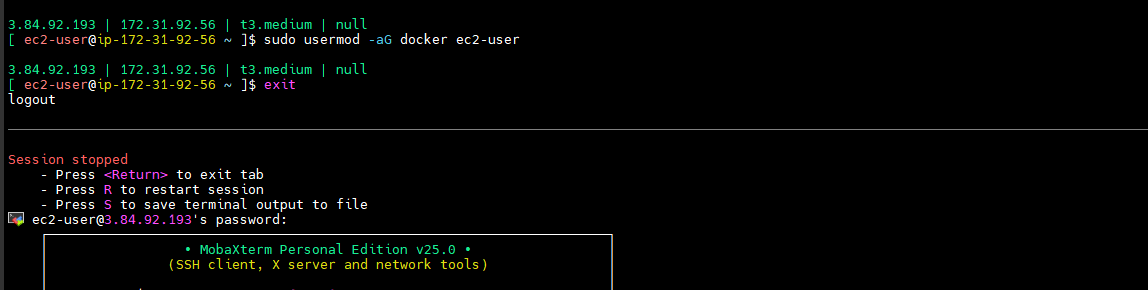
**SESSION - 61**

**--> cd /c/devops/daws-84s/repos/docker**

**--> terraform apply -auto-approve**



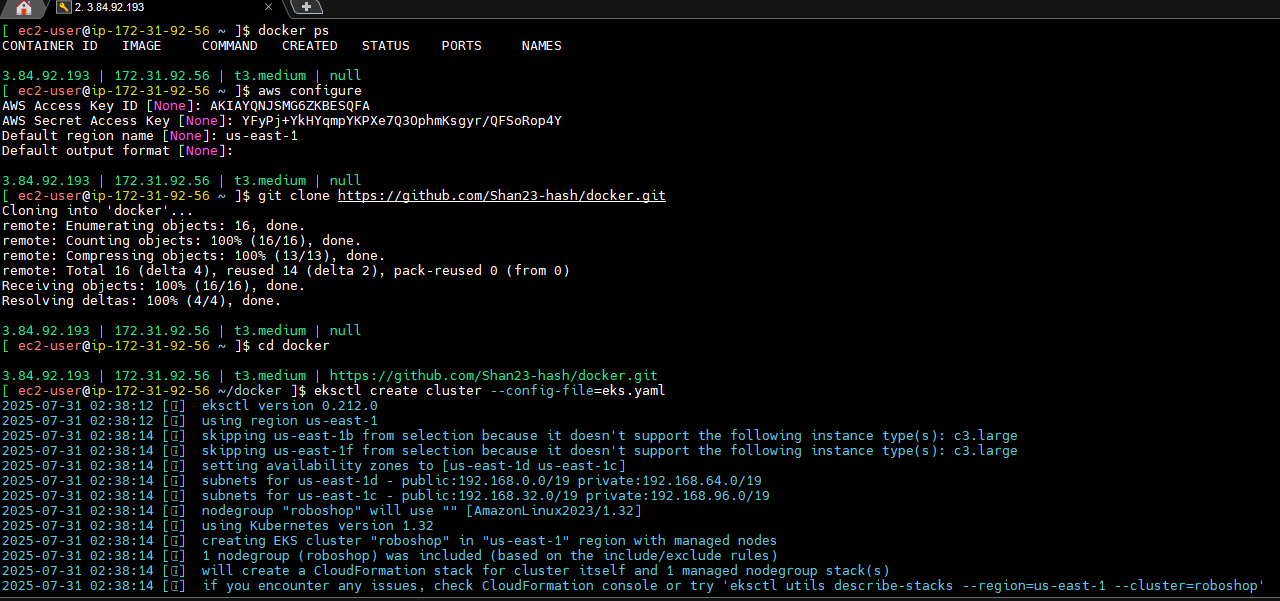
**--> sudo usermod -aG docker ec2-user**

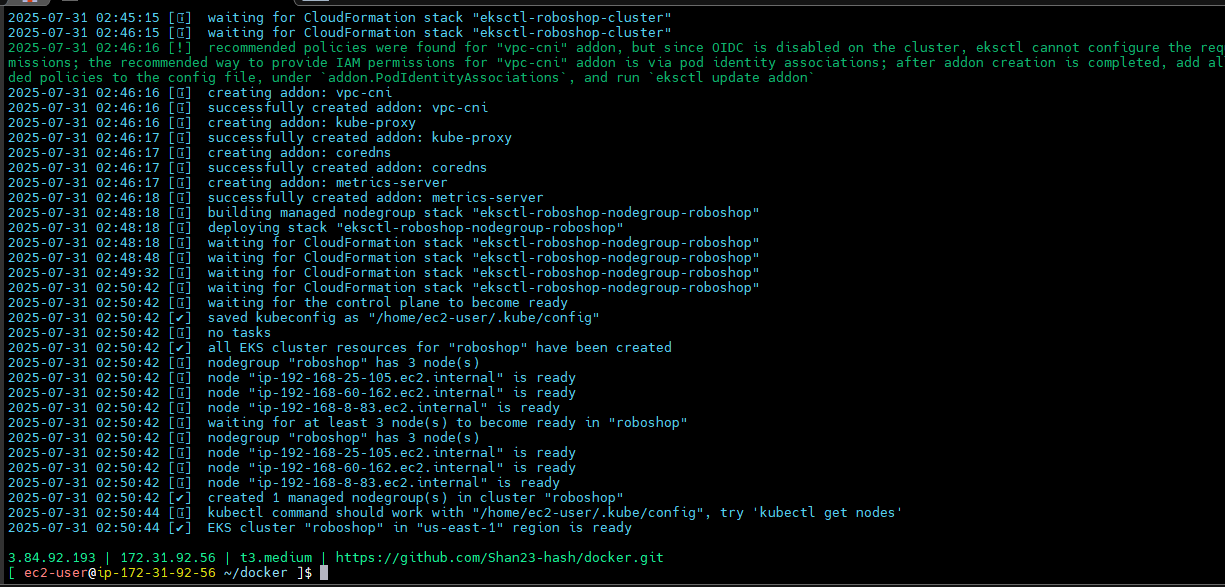


**--> aws configure**

**--> git clone <https://github.com/Shan23-hash/docker.git>**

**--> eksctl create cluster --config-file=eks.yaml**





**VOLUMES**

**--------------**

**-->**  Volume is nothing but data storage.

--> External HD (hard disk)

--> Google drives

**HD -- EBS (Elastic block storage)**

* Should be as near as possible to the system/server.
* Data transfer is very fast.
* Can store Operating system.
* Well suitable for databases
* Can be connected to only one device

**Drive -- EFS (Elastic File System)**

* Can be anywhere in the internet
* Can be used in multiple systems
* Can’t store OS and databases in drive
* Can store files (files means images and vedios…)
* Data transfer is slow compared to HD.

--> How we can store data in these two, will see

**Storage Administrator**

**------------------------------**

--> Every data they have to take data backup

--> Data security

--> Data Restore

--> Data Replication

Kubernates created rapars

--> PV,PVC, AND SC

--> Persistant volume, persistant volume claim, storage class

--> on behalf of us kubernates connect with this storage devices and perform the operations through this objects or resources.

**EBS (Elastic Block Storage)**

**=======================**

--> here we have two types of provisioning

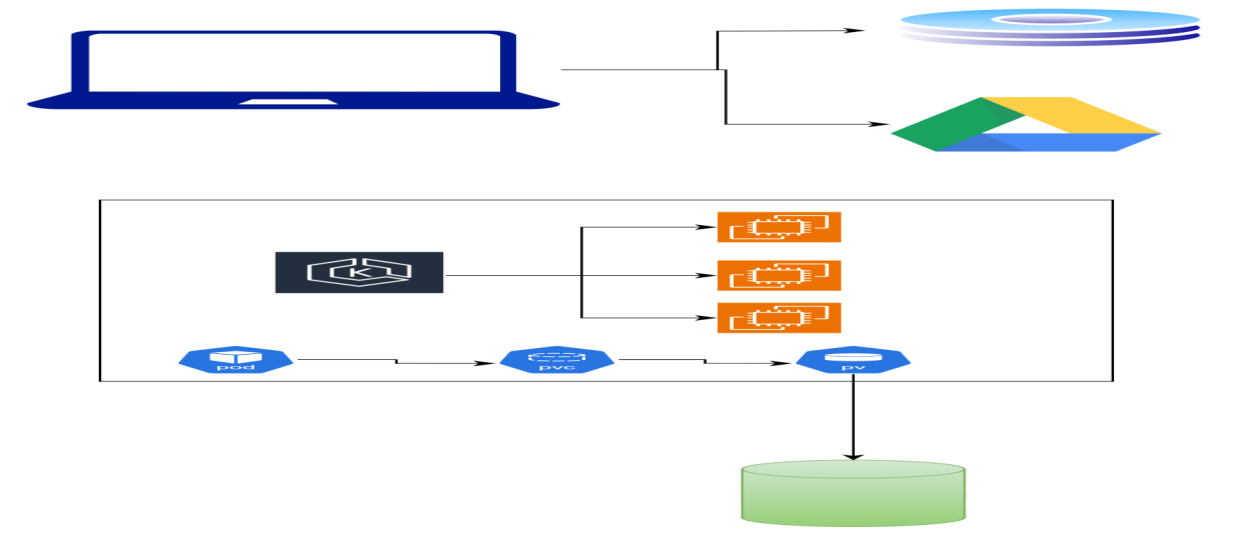
1. Static provisioning --> Need to create storage manually
2. Dynamic provisioning --> Disk will be created automatically

**EBS Static provisioning**

**====================**

1. Need to create disk
2. EBS disks should be in the same Availability zone of EC2 instance.
3. Install drivers
4. EC2 nodes should have permission to connect EBS disks.

--> First will see where is our servers.



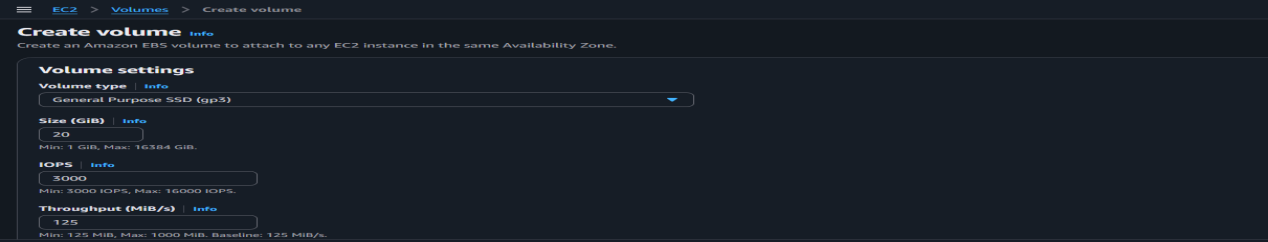
--> in 1a and 1c is there.

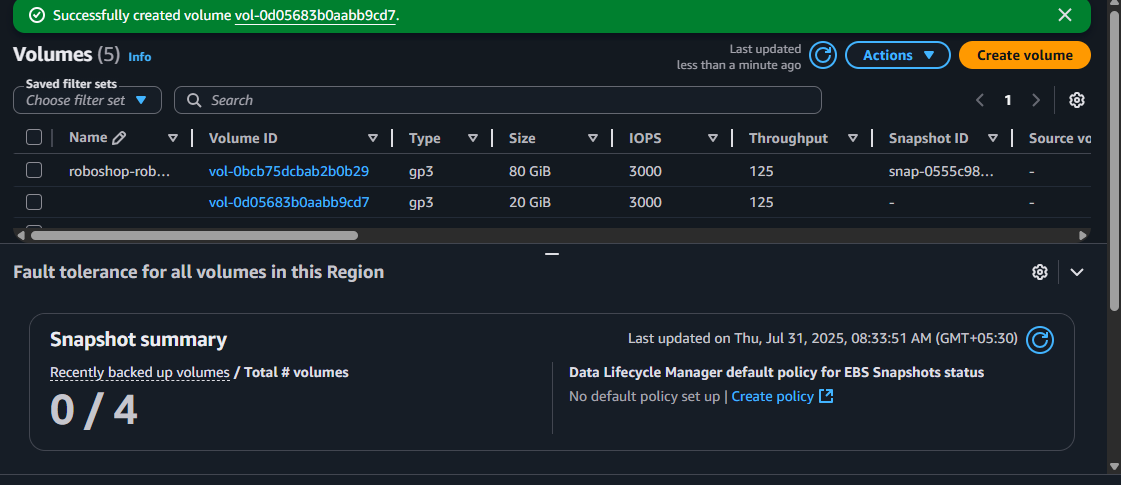
--> Create volume

--> Availability zone - 1c

--> SIze - 20GB

--> disk was created

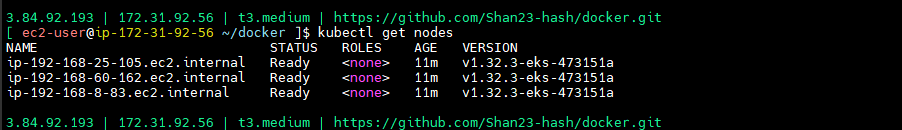




--> you want to miunt this one

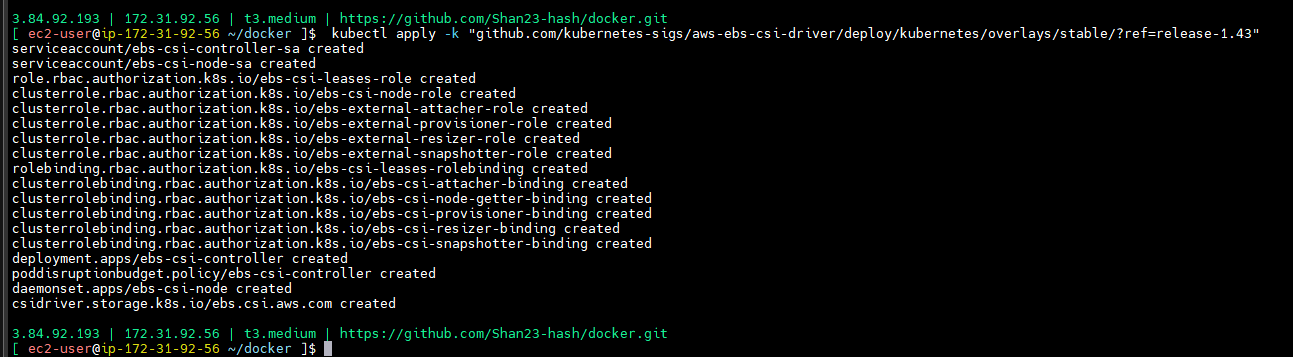
--> for connecting kubernates install deivers

**--> kubectl get nodes**



--> Kubernates ebs drivers -- check in google

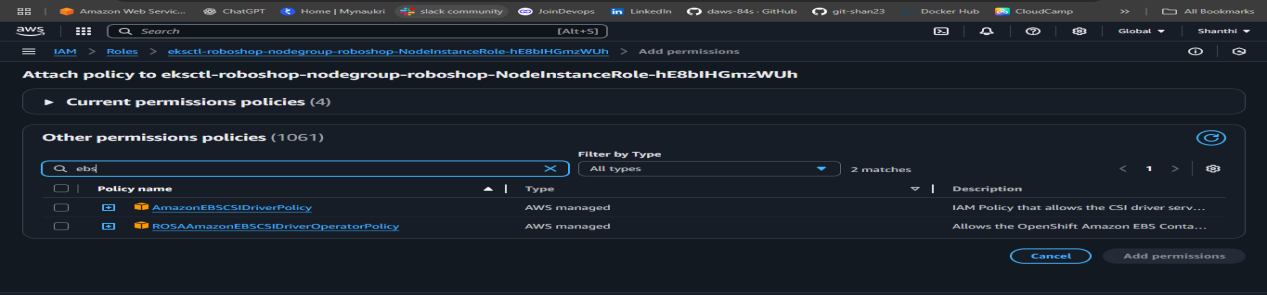
**--> kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.43"**



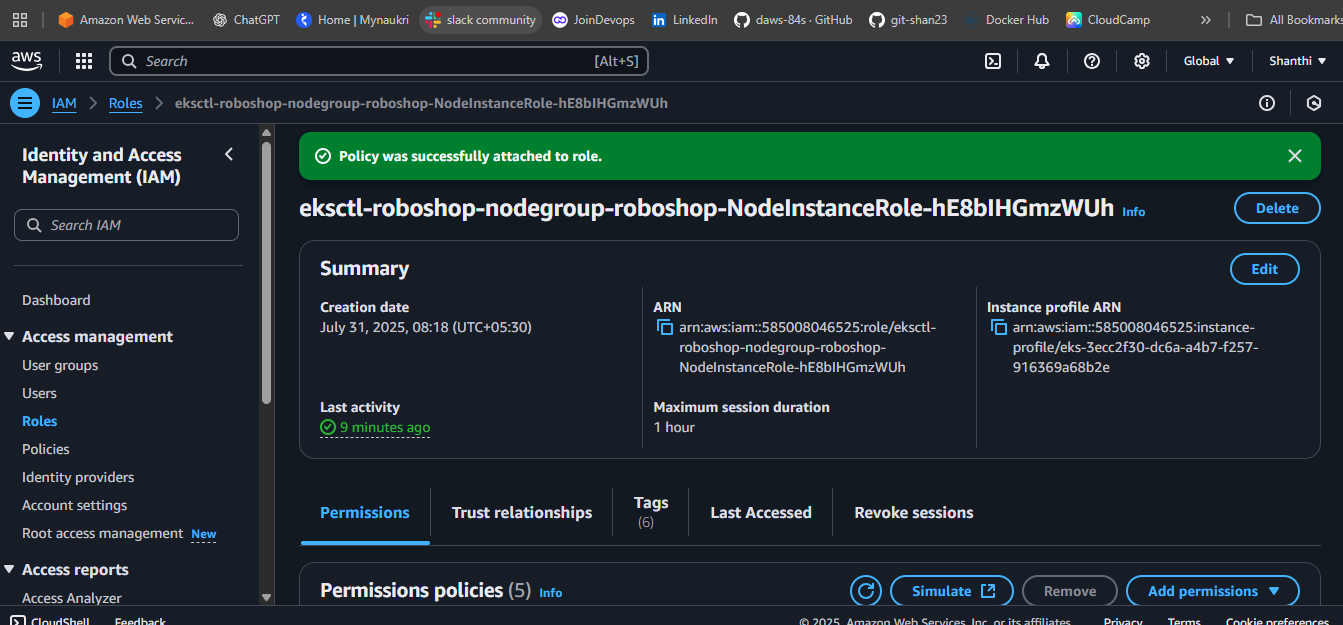
--> Run this above command.

--> in EC2 instance role have to give give that permission.

--> EC2 instance -->security --> IAM Role --> Add policies



--> AmazonEBSCSIDriverPolicy --> add permissions



--> That two steps are created.

**PV (Persistant volume)**

**---------------------------------**

--> K8 resource, It is the physical representation of the volume…

--> You can represent kubernates.

**volumes/01-ebs-static.yaml**

apiVersion: v1

kind: PersistentVolume

metadata:

name: ebs-static

spec:

accessModes:

- ReadWriteOnce

capacity:

storage: 20Gi

csi:

driver: ebs.csi.aws.com

fsType: ext4

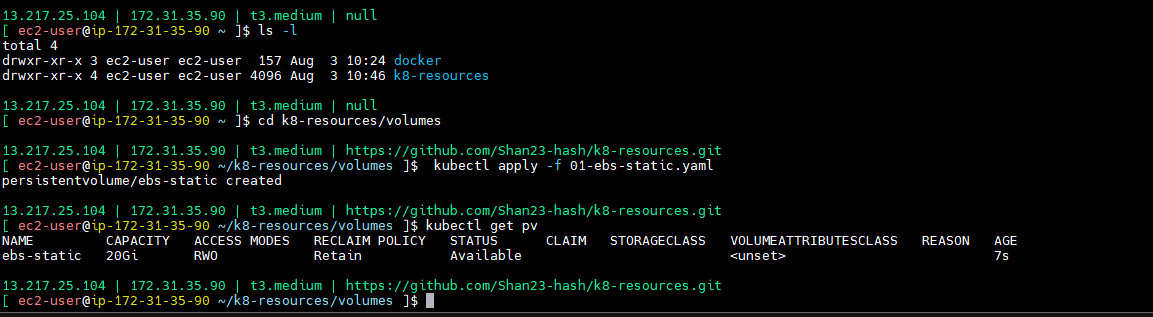
volumeHandle: vol-072a8f44214a23f64

**--> git clone https://github.com/Shan23-hash/k8-resources.git**

**--> cd volumes/**

**--> kubectl apply -f 01-ebs-static.yaml**

**--> kubectl get pv**



--> EBS can not be attached at a time to multiple instances.

### Access Modes

A PersistentVolume can be mounted on a host in any way supported by the resource provider. As shown in the table below, providers will have different capabilities and each PV's access modes are set to the specific modes supported by that particular volume. For example, NFS can support multiple read/write clients, but a specific NFS PV might be exported on the server as read-only. Each PV gets its own set of access modes describing that specific PV's capabilities.

The access modes are:

--> **ReadWriteOnce**

the volume can be mounted as read-write by a single node. ReadWriteOnce access mode still can allow multiple pods to access (read from or write to) that volume when the pods are running on the same node. For single pod access, please see ReadWriteOncePod.

**ReadOnlyMany**

the volume can be mounted as read-only by many nodes.

**ReadWriteMany**

the volume can be mounted as read-write by many nodes.

the volume can be mounted as read-write by a single Pod. Use ReadWriteOncePod access mode if you want to ensure that only one pod across the whole cluster can read that PVC or write to it.

**volumes/01-ebs-static.yaml**

apiVersion: v1

kind: PersistentVolume

metadata:

name: ebs-static

spec:

accessModes:

- ReadWriteOnce

capacity:

storage: 20Gi

csi:

driver: ebs.csi.aws.com

fsType: ext4

volumeHandle: vol-072a8f44214a23f64

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: ebs-static

spec:

storageClassName: "" # Empty string must be explicitly set otherwise default StorageClass will be set

volumeName: ebs-static

accessModes:

- ReadWriteOnce

resources:

requests:

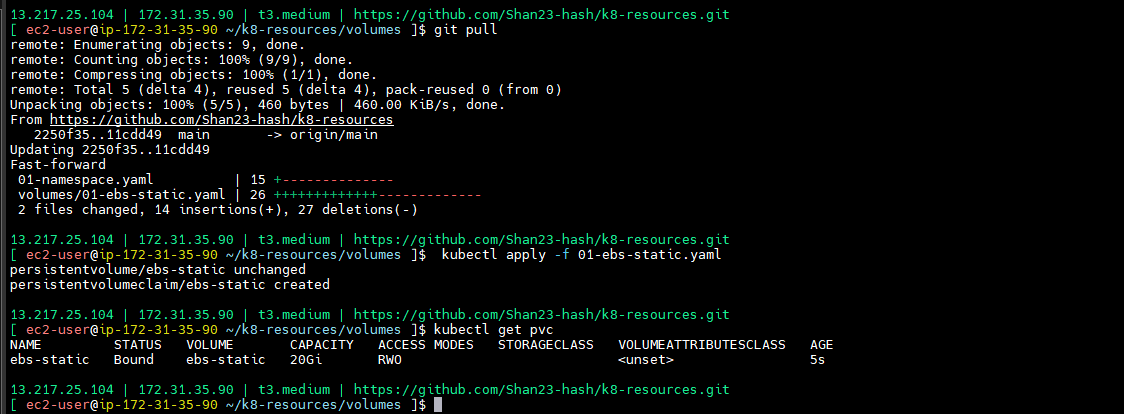
storage: 15Gi

**--> git clone https://github.com/Shan23-hash/k8-resources.git**

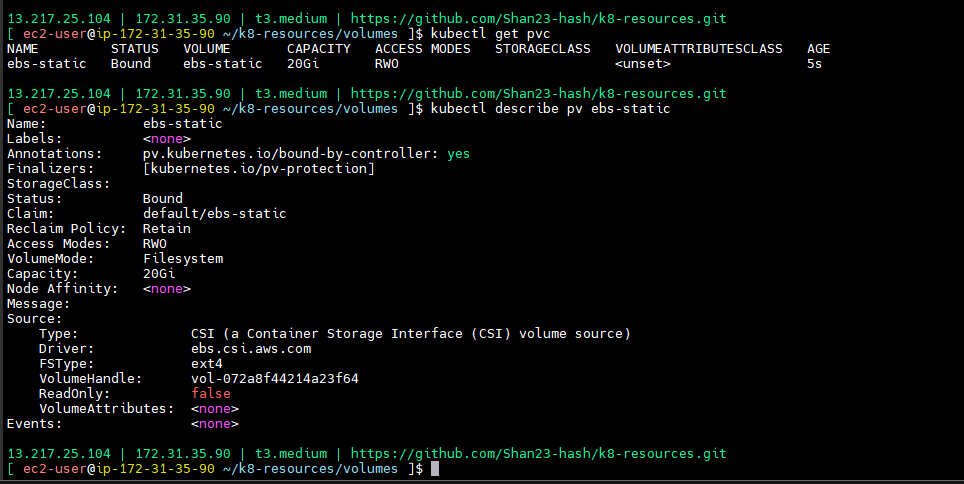
**--> cd volumes/**

**--> kubectl apply -f 01-ebs-static.yaml**

**--> kubectl get pv**



**--> kubectl describe pv ebs-static**



### Reclaim Policy

Current reclaim policies are:

* Retain -- manual reclamation
* Recycle -- basic scrub (rm -rf /thevolume/\*)
* Delete -- delete the volume

### Reclaiming

When a user is done with their volume, they can delete the PVC objects from the API that allows reclamation of the resource. The reclaim policy for a PersistentVolume tells the cluster what to do with the volume after it has been released of its claim. Currently, volumes can either be Retained, Recycled, or Deleted.

#### Retain

The Retain reclaim policy allows for manual reclamation of the resource. When the PersistentVolumeClaim is deleted, the PersistentVolume still exists and the volume is considered "released". But it is not yet available for another claim because the previous claimant's data remains on the volume. An administrator can manually reclaim the volume with the following steps.

1. Delete the PersistentVolume. The associated storage asset in external infrastructure still exists after the PV is deleted.
2. Manually clean up the data on the associated storage asset accordingly.
3. Manually delete the associated storage asset.

If you want to reuse the same storage asset, create a new PersistentVolume with the same storage asset definition.

#### Delete

For volume plugins that support the Delete reclaim policy, deletion removes both the PersistentVolume object from Kubernetes, as well as the associated storage asset in the external infrastructure. Volumes that were dynamically provisioned inherit the [reclaim policy of their StorageClass](https://kubernetes.io/docs/concepts/storage/persistent-volumes/" \l "reclaim-policy), which defaults to Delete. The administrator should configure the StorageClass according to users' expectations; otherwise, the PV must be edited or patched after it is created. See [Change the Reclaim Policy of a PersistentVolume](https://kubernetes.io/docs/tasks/administer-cluster/change-pv-reclaim-policy/).

--> you can change reclime pods.

**PVC**

**====**

it is claiming resource of the storage i.e PV

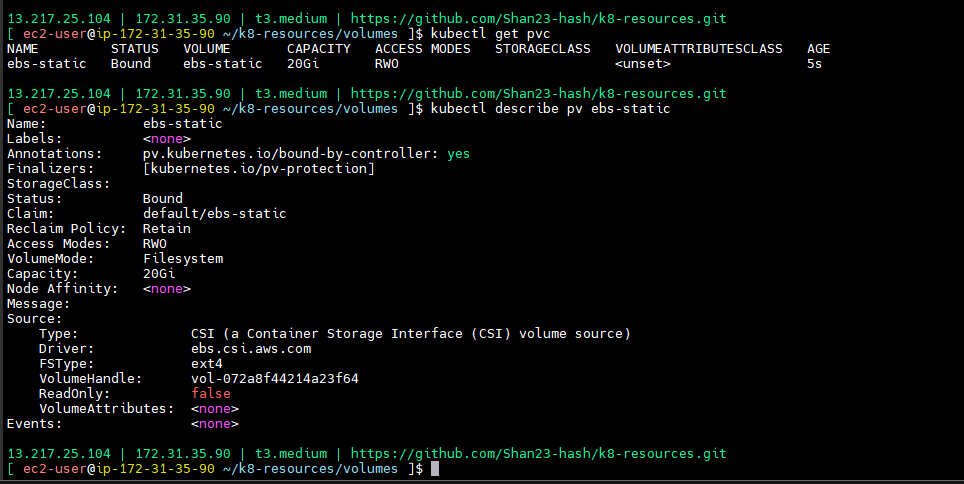
--> Maximum you cn clime 20GB.

--> push and pull the code

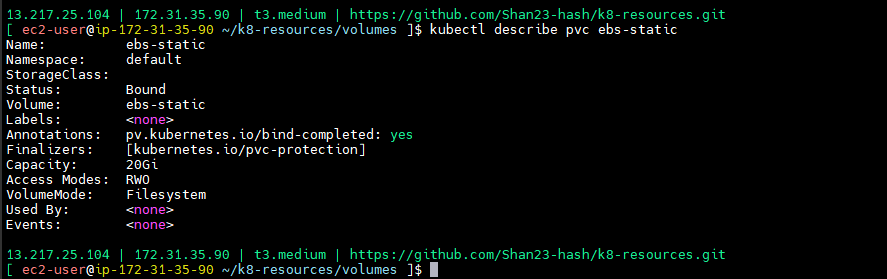
**--> kubectl apply -f 01-ebs-static.yaml**

**--> Kubectl get pvc**

**--> kubectl describe pv ebs-static**



--> **kubectl describe pvc ebs-static**



--> capacity is 20GB, but we are asked 15GB

--> In pod you have mount one container.

--> where it will store altimately

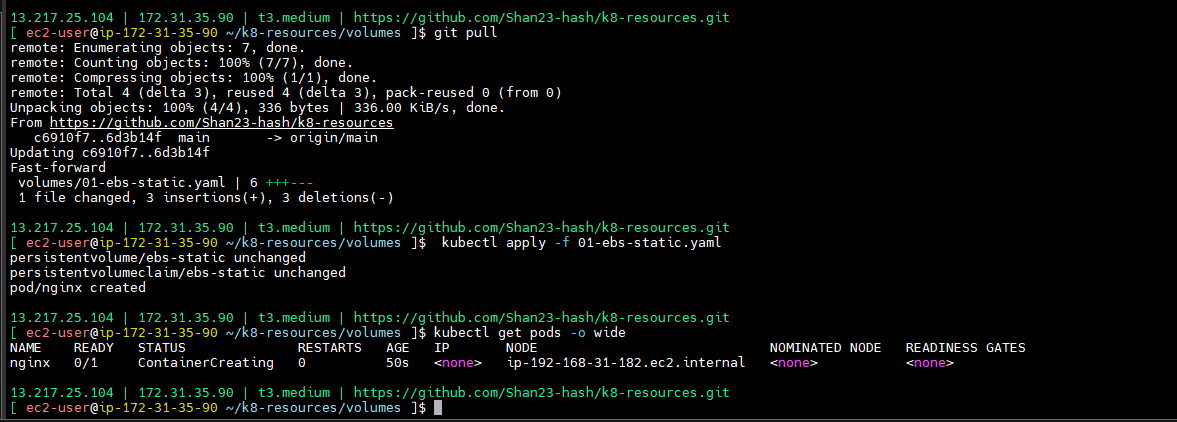
--> if pod go to 1c, pod will go then only it will attach.

--> if pod go to 1a, Pod will not attach, will try

--> Push and pull the code

**--> kubectl apply -f 01-ebs-static.yaml**

**--> kubectl get pods -o wide**



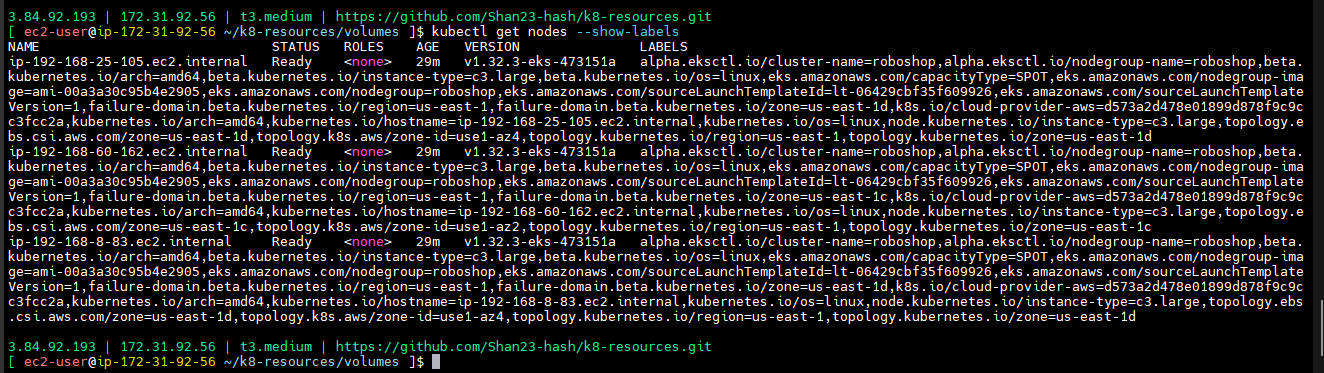
**-->** if you want to go particular nodes that is called select nodes

--> kubectl node selectros -- check in google

you can create labels to the nodes, use those labels as node selectors inside pod.

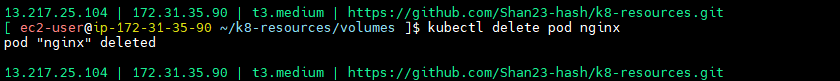
--> kubernates get nodes show labels -- check in google

**--> kubectl get nodes --show-labels**



**-->** copy labels and check

**--> kubectl delete pod nginx**

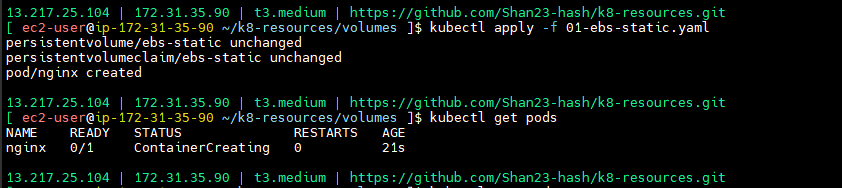


**-->** push the code

**--> kubectl apply -f 01-ebs-static.yaml**

--> we are selected pods so it will come 1c with running state.

**--> kubectl get pods**



--> I want to see nginx pods need to create one service that is load balancer.

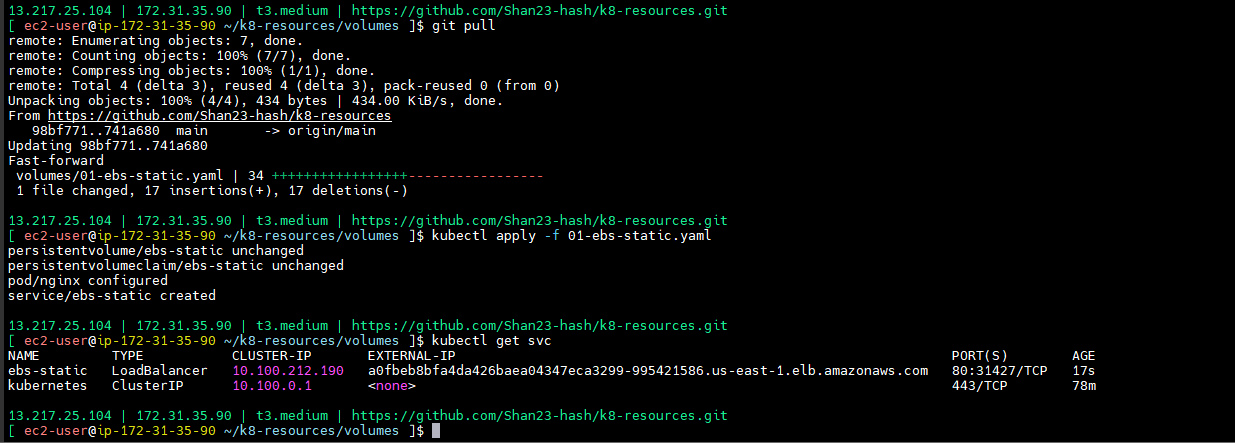
--> create labels also

--> This labels I’m giving to appversion and node selectros

--> git pull -- push and pull the code

**--> kubectl apply -f 01-ebs-static.yaml**

**--> kubectl get svc**



--> acccess the output -- External-ip -- not came output see the security services

--> response came but 403 got

--> **kubectl get svc**

**--> kubectl exec -it nginx -- bash**

**--> cd /usr/share/nginx/html/**

**-->echo ” <h1>Hi, I am from EBS static</h1>” index.html**

**--> kubectl delete pod nginx**

--> but it will not delete

--> See EBS volumes after deleting pod came to avilable ( avilable means no one using this pod)

--> if again created pod

**--> kubectl apply -f 01-ebs-static.yaml**

--> It should be in use and you successfully access with in a sec

**--> kubectl get pods**(came to running)

**--> kubectl exec -it nginx -- bash**

**--> cd /usr/share/nginx/html/**

**-->** index.html is there.

<https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/>

--> This is EBS static

--> Kubernates is only for state less application.

**EBS Dynamic**

**-----------------------**

--> no need to create disk manually.

--> There should be some resources in kubernates.

--> StorageClass is the resource that creates the disk automatically.

--> EKS administrator creates SC resource, he will make it available for us. we use this SC to create disk and PV.

**--> kubectl delete -f 01-ebs-static.yaml**

--> Remove manually created volume

--> kubectl get sc

--> Get storage class

--> EC2 is nothing but EBS, Reclaim policy we can give it as retail

**--> kubectl get sc gp2 -o yaml**



--> Here I can check dynamic provisioning - in provisioning example storage class storage class there.

**volumes/02-sc.yaml**

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: roboshop-ebs

reclaimPolicy: Retain

provisioner: ebs.csi.aws.com

volumeBindingMode: WaitForFirstConsumer # disk will be created when pod is getting created

--> push and pull the code

--> in ebs have to create dynamic automatically.

--> disk you are not created db also no need to create.

**volumes/03-ebs-dynamic.yaml**

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: ebs-dynamic

spec:

accessModes:

- ReadWriteOnce

storageClassName: roboshop-ebs

resources:

requests:

storage: 4Gi

---

apiVersion: v1

kind: Pod

metadata:

name: nginx

labels:

purpose: ebs-dynamic-demo

project: roboshop

spec:

nodeSelector:

topology.kubernetes.io/zone: us-east-1c

containers:

- name: nginx

image: nginx

volumeMounts: # docker run -v host-path:container-path

- name: persistent-storage

mountPath: /usr/share/nginx/html

volumes:

- name: persistent-storage

persistentVolumeClaim:

claimName: ebs-dynamic

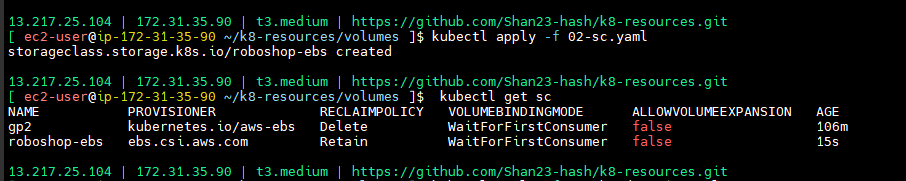
--> we can attch to pod

--> storage class one time creation

--> puah and pull the code

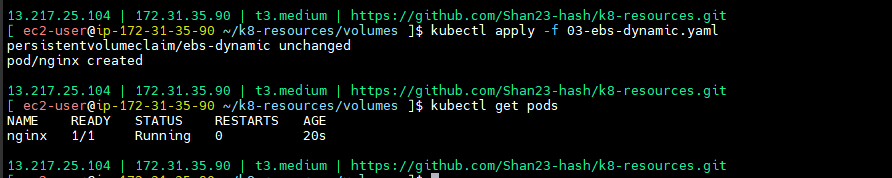
**--> kubectl apply -f 02-sc.yaml**

**--> kubectl get sc**



**--> kubectl apply -f 03-ebs-dynamic.yaml**

**--> kubectl get pods**



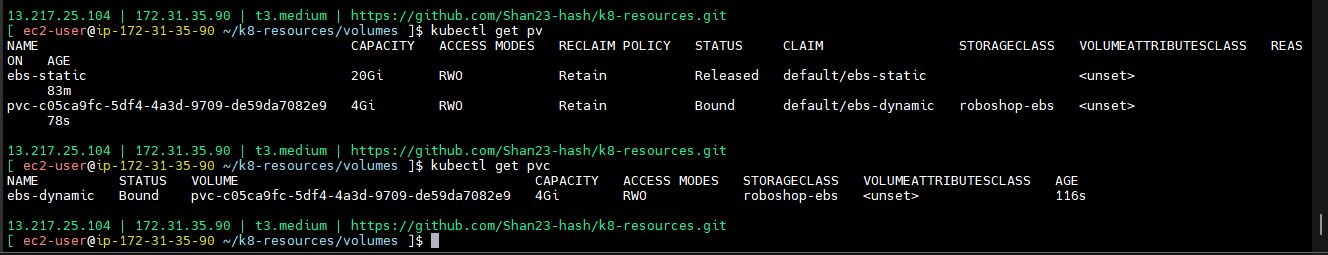
**-->** came from running inside.

--> output: See 4GB disk is created.

--> check pv, while creating disk pv also will create

**--> kubectl get pv --** pvc created dynamically

**--> kubectl get pvc --** name and volume it will come



**Quiz**

**---------**

1. **Which of below is true about docker?**

**Ans)) All of these**

1. **How do you list all Docker images on your system?**

**Ans)) Docker images**

1. **How can you view the logs of a docker container?**

**Ans)) docker logs**

1. **What a command is used to remove a docker container?**

**Ans)) docker rm**

1. **What command provides real-time resources usage statistics for docker containers?**

**Ans)) docker stats**

1. **How can you list Docker networks on your system?**

**Ans)) docker network ls**

1. **How can you execute a command inside a running docker container?**

**Ans)) docker exec**

1. **What option is used to map host ports to container ports to container ports when running a docker container?**

**Ans)) -p host-port:container-port**

1. **What is the first instruction you typically use in a docker file?**

**Ans)) FROM**

1. **What is the porpose of the EXPOSE instruction in a Dockerfile?**

**Ans)) To let know the exposed ports to the outside world**

1. **How do you push a docker image to a remote registry?**

**Ans)) docker push url/user-name/image:version**

1. **What is the best practice of containers?**

**Ans)) Running as non-root user**

1. **Which factor should be considered when selecting an AWS region for deploying resources?**

**Ans)) All the above**

1. **When CMD instruction executes?**

**Ans)) Run time**

dev and prod

dev

=====

roboshop-dev

roboshop-qa

prod

======

roboshop

cluster level -> admin activity

namespace level --> EKS user activity