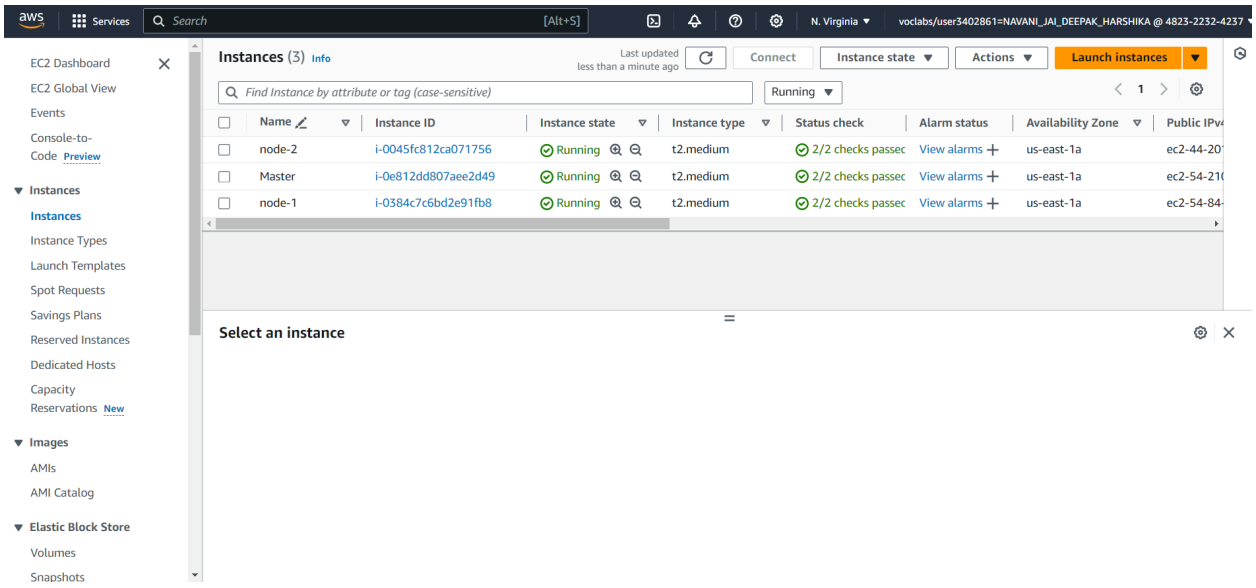
Experiment no:3
Advance-devops
Jai navani
D15-A

Step 1: create instances (1 master and 2 nodes)



Connect the instances:

Install docker in all instances

```
[root@ip-172-31-88-48 ec2-user]# yum install docker -y
Last metadata expiration check: 0:04:33 ago on Sun Sep 22 07:37:42 2024.
Dependencies resolved.
================================================================================================================
 Package                     Architecture        Version                          Repository              Size
================================================================================================================
Installing:
 docker                      x86_64              25.0.6-1.amzn2023.0.2            amazonlinux             44 M
Installing dependencies:
 containerd                  x86_64              1.7.20-1.amzn2023.0.1            amazonlinux             35 M
 iptables-libs               x86_64              1.8.8-3.amzn2023.0.2             amazonlinux            401 k
 iptables-nft                x86_64              1.8.8-3.amzn2023.0.2             amazonlinux            183 k
 libcgroup                   x86_64              3.0-1.amzn2023.0.1               amazonlinux             75 k
 libnetfilter_conntrack      x86_64              1.0.8-2.amzn2023.0.2             amazonlinux             58 k
 libnfnetlink                x86_64              1.0.1-19.amzn2023.0.2            amazonlinux             30 k
 libnftnl                    x86_64              1.2.2-2.amzn2023.0.2             amazonlinux             84 k
 pigz                        x86_64              2.5-1.amzn2023.0.3               amazonlinux             83 k
 runc                        x86_64              1.1.13-1.amzn2023.0.1            amazonlinux            3.2 M

Transaction Summary
================================================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
```

After installation start the docker:

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64       docker-25.0.6-1.amzn2023.0.2.x86_64       iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64      libcgroup-3.0-1.amzn2023.0.1.x86_64       libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64     libnftnl-1.2.2-2.amzn2023.0.2.x86_64      pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-88-48 ec2-user]# systemctl start docker
[root@ip-172-31-88-48 ec2-user]#
[root@ip-172-31-88-48 ec2-user]#
[root@ip-172-31-88-48 ec2-user]#
[root@ip-172-31-88-48 ec2-user]# systemctl start docker
```

Code for installation of kubernete:

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/

enabled=1

gpgcheck=1

gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key

exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni

EOF

# Set SELinux in permissive mode (effectively disabling it)

sudo setenforce 0
```

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

```
sudo systemctl enable --now kubelet
```

Output (in all instances):

```
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x86_64                  kubeadm-1.31.1-150500.1.1.x86_64
  kubectl-1.31.1-150500.1.1.x86_64                     kubelet-1.31.1-150500.1.1.x86_64                    kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-88-48 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
```

```
Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-88-241 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
[root@ip-172-31-88-241 ec2-user]#
```

i-0384c7c6bd2e91fb8 (node-1)

PublicIPs: 54.84.57.3 PrivateIPs: 172.31.88.241

```
Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-89-83 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
[root@ip-172-31-89-83 ec2-user]#
```

i-0045fc812ca071756 (node-2)

PublicIPs: 44.201.84.150 PrivateIPs: 172.31.89.83

Kubeadm init:
(after intialization):

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.88.48:6443 --token 2nrq25.jqlp7gerx8nty6ys \
        --discovery-token-ca-cert-hash sha256:d969d1bb086d72a8c952b2b6904c7c6f8c7e42a1d12f6ef1a82a46935363e411
```

```
[root@ip-172-31-88-241 ec2-user]# sudo yum install iproute
Last metadata expiration check: 0:29:35 ago on Sun Sep 22 07:54:40 2024.
Package iproute-5.10.0-2.amzn2023.0.5.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-88-241 ec2-user]# kubeadm join 172.31.88.48:6443 --token 2nrq25.jqlp7gerx8nty6ys --discovery-token-ca-cert-hash sha256:d969d1bb086d72a8c952b2b6904
c7c6f8c7e42a1d12f6ef1a82a46935363e411
[preflight] Running pre-flight checks
        [WARNING FileExisting-tc]: tc not found in system path
```