Name: Shejal Nilesh Tiwari Div/Roll no: D15C/57

EXPERIMENT NO: 3

AIM :To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

STEPS :

1.We have created 3 instances

☐	Master	i-0767a02f53056b254	Running	t3.small	Initializing	View alarms +	us-east-1c	ec2-44-202-26-210.co.
☐	worker-1	i-0a98404682c2bf690	Running	t3.small	Initializing	View alarms +	us-east-1c	ec2-18-206-158-113.cc
☐	worker-2	i-0a9ea3e263873d151	Running	t3.small	Initializing	View alarms +	us-east-1c	ec2-3-89-36-106.comp

2. Now connect the three instances
And write the command on the three of the linux command promp.

sudo su
And yum install docker -y (To download docker in All three machines)

Master Worker1 Worker2

```
[ec2-user@ip-172-31-93-226 ~]$ sudo su
[root@ip-172-31-93-226 ec2-user]# yum install docker -y
Last metadata expiration check: 0:07:12 ago on Fri Sep 13 11:58:42 2024.
Dependencies resolved.
================================================================================
====================================
 Package                          Architecture          Version                   Repo
sitory                           Size
================================================================================
====================================
Installing:
 docker                           x86_64                25.0.6-1.amzn2023.0.2     amaz
onlinux                          44 M
Installing dependencies:
 containerd                       x86_64                1.7.20-1.amzn2023.0.1     amaz
onlinux                          35 M
 iptables-libs                    x86_64                1.8.8-3.amzn2023.0.2      amaz
onlinux                         401 k
 iptables-nft                     x86_64                1.8.8-3.amzn2023.0.2      amaz
onlinux                         183 k
```

3.Now to start docker write command
systemctl start docker

Master Worker1 Worker2

```
[root@ip-172-31-93-226 ec2-user]# systemctl start docker
[root@ip-172-31-93-226 ec2-user]#
```

Name: Shejal Nilesh Tiwari Div/Roll no: D15C/57

4.Now to install kubeadm

sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
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

sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
sudo systemctl enable --now kubelet

Master Worker1 Worker2

```
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64              cri-tools-1.31.1-150500.1.1.x86_64                      kubeadm-1.31.1-15
0500.1.1.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_queu
e-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-84-46 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
```

6. Now to confirm that we have got repository for kubernets we will write the command
yum repo list

Master Worker1 Worker2

```
[root@ip-172-31-84-46 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-172-31-84-46 ec2-user]#
```

Name: Shejal Nilesh Tiwari Div/Roll no: D15C/57

7.NOW IN THE MASTER NODE WE NEED TO INITIALIZE KUBEADM

Only in Master

```
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 172.31.84.46:6443 --token jl06ac.t7cdzxf0x5eddmsl \
        --discovery-token-ca-cert-hash sha256:4a152b913ee4b60dc2126d55f631b86d0dafb7d58132416c4f32f0668ac553be
[root@ip-172-31-84-46 ec2-user]#
```

In the screenshot you can see the commands written in the 3rd 4th and 5th line
Copy that command , this command is used to add right permission to the user
Also copy the 7th line , here it is the credential for the user .
Also copy the last 2 lines it is a link used to join the nodes

NOW IN THE MASTER WRITE THE COMMAND WHICH YOU COPIED IN FIRST AND SECOND TIME.

This is the join link which you need to enter in Worker1 and Worker2

kubeadm join 172.31.84.46:6443 --token jl06ac.t7cdzxf0x5eddmsl \
        --discovery-token-ca-cert-hash
sha256:4a152b913ee4b60dc2126d55f631b86d0dafb7d58132416c4f32f0668ac553be

Now in the master we will write the command
kubectl get node

This was the output i got in my master node

```
[root@ip-172-31-84-46 ec2-user]# kubectl get node
NAME                          STATUS     ROLES           AGE   VERSION
ip-172-31-84-46.ec2.internal   NotReady   control-plane   56s   v1.31.1
[root@ip-172-31-84-46 ec2-user]# kubectl get node
The connection to the server 172.31.84.46:6443 was refused - did you specify the right host or port?
[root@ip-172-31-84-46 ec2-user]# kubectl get node
NAME                          STATUS     ROLES           AGE    VERSION
ip-172-31-84-46.ec2.internal   NotReady   control-plane   5m1s   v1.31.1
[root@ip-172-31-84-46 ec2-user]#
```

Name: Shejal Nilesh Tiwari Div/Roll no: D15C/57

And this was the output when i tried connecting it with worker 1 and worker 2

So there was a failure in connecting the worker 1 and 2 with the master even if i provided the joining link.
The Cloud shell didn't proceeds further.

CONCLUSION :
We first created Three instances Worker 1 and 2 , Master .
Then after connecting it .
We installed DOCKER and KUBERNETES repository in all three machines.
Then we initialized Kubernetes control panel.
Provided join link to Worker 1 and Worker2 but it didn't successfully joined.
We checked the joined nodes and master but Worker 1 and 2 and we suspect the following issues while trouble shooting .

Connection Refused Errors: The Kubernetes components are failing to connect to the API server at `https://172.31.84.46:6443`, resulting in "connection refused" errors.
No Running Containers: The `docker ps -a` command shows no containers, which means the Kubernetes components are not running in Docker.
CrashLoopBackOff: There are errors indicating that the containers for Kubernetes components are restarting repeatedly but failing to start properly.
Network Plugin Not Ready: The error message "Network plugin returns error: cni plugin not initialized" suggests issues with the network plugin required for Kubernetes.