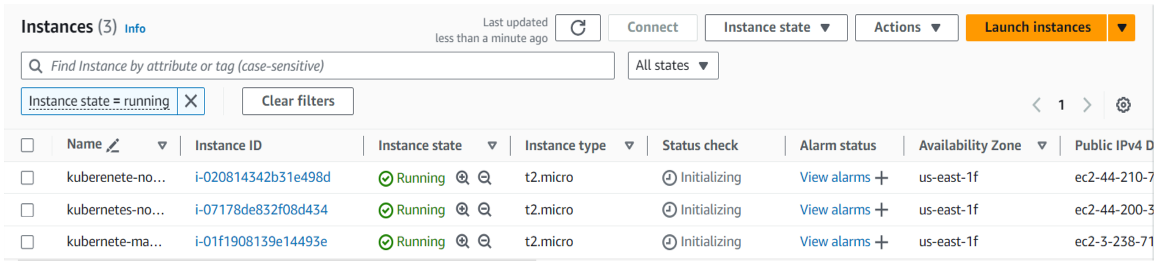
Name: Sahil Motiramani Div:D15C Roll No:35

EXP No.:3

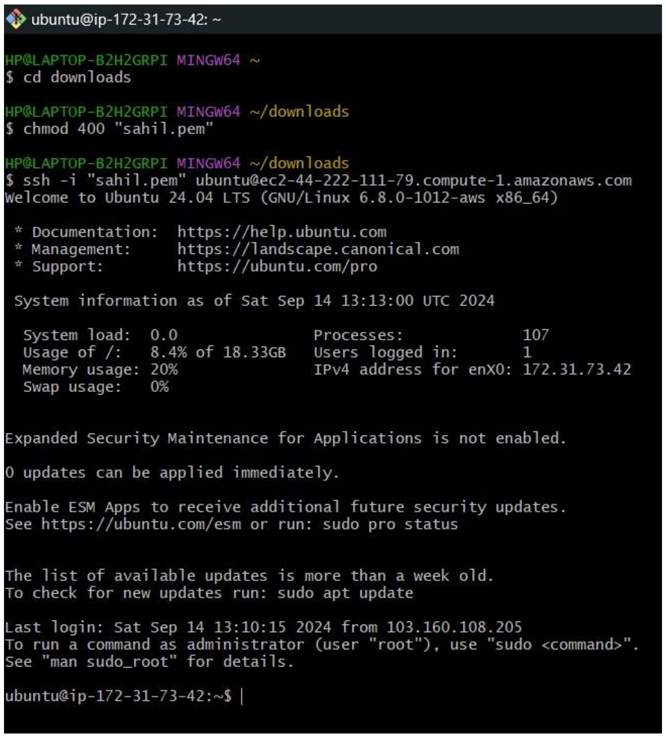
Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud

Step 1:Create 3 EC-2 instances with all running on Amazon Linux as OS Kubernete-master,kuberenete-node1,kubernete-node2.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
kuberenete-no...	i-020814342b31e498d	Running	t2.micro	Initializing	View alarms +	us-east-1f	ec2-44-210-7
kubernetes-no...	i-07178de832f08d434	Running	t2.micro	Initializing	View alarms +	us-east-1f	ec2-44-200-3
kubernete-ma...	i-01f1908139e14493e	Running	t2.micro	Initializing	View alarms +	us-east-1f	ec2-3-238-71

Step 2:SSH into all 3 machines each in separate terminal for each instance,

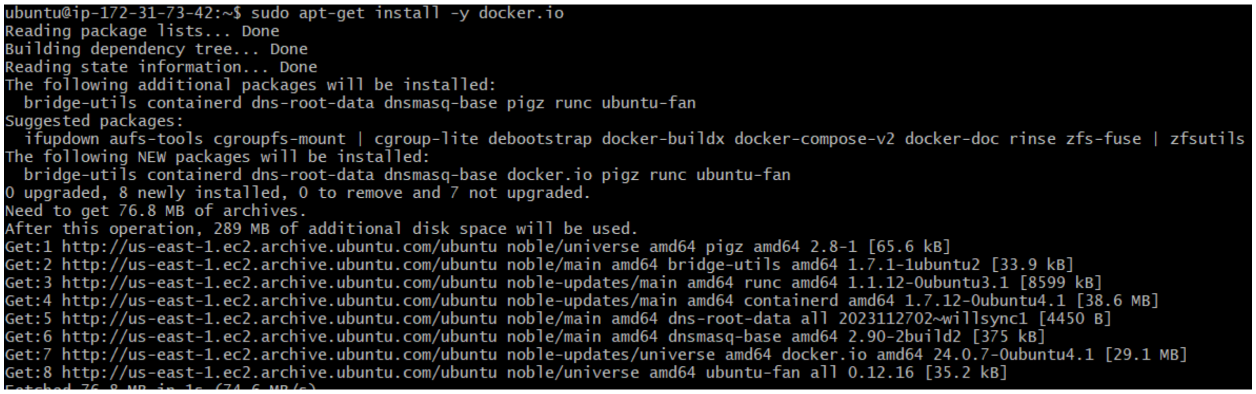


```
ubuntu@ip-172-31-73-42: ~

HP@LAPTOP-B2H2GRPI MINGW64 ~
$ cd downloads

HP@LAPTOP-B2H2GRPI MINGW64 ~/downloads
$ chmod 400 "sahil.pem"

HP@LAPTOP-B2H2GRPI MINGW64 ~/downloads
$ ssh -i "sahil.pem" ubuntu@ec2-44-222-111-79.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat Sep 14 13:13:00 UTC 2024

  System load:  0.0               Processes:             107
  Usage of /:   8.4% of 18.33GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for enX0: 172.31.73.42
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Sep 14 13:10:15 2024 from 103.160.108.205
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-73-42:~$ |
```

Name: Sahil Motiramani Div:D15C Roll No:35

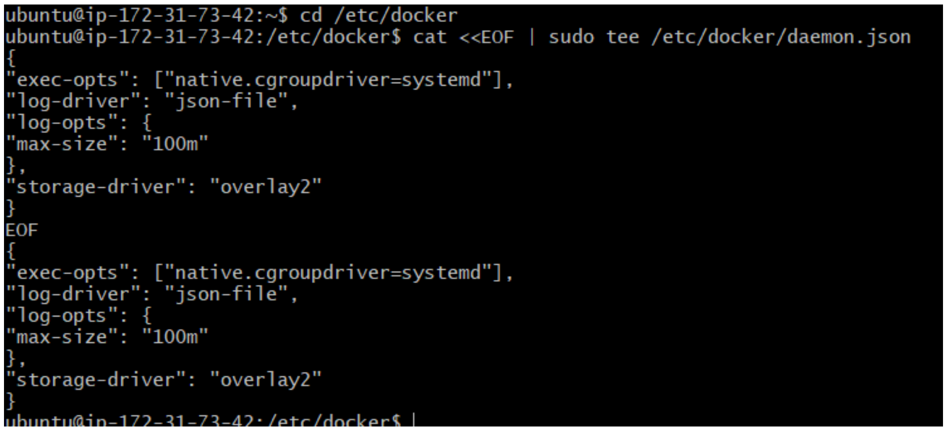
Step 3:Now install docker in all 3 instances;

sudo apt-get intall-y docker.io



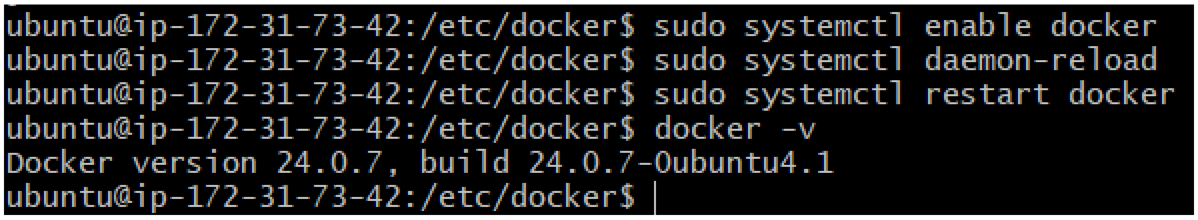
```
ubuntu@ip-172-31-73-42:~$ sudo apt-get install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 7 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702~willsync1 [4450 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
```

Then, Configure File daemon.json;



```
ubuntu@ip-172-31-73-42:~$ cd /etc/docker
ubuntu@ip-172-31-73-42:/etc/docker$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
```

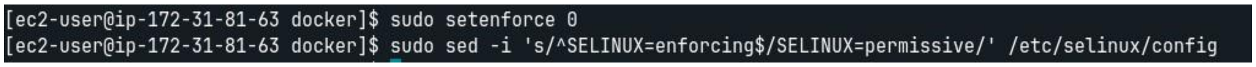
- sudo systemctl enable docker
- sudo systemctl daemon-reload
- sudo systemctl restart docker
- docker-v



```
ubuntu@ip-172-31-73-42:/etc/docker$ sudo systemctl enable docker
ubuntu@ip-172-31-73-42:/etc/docker$ sudo systemctl daemon-reload
ubuntu@ip-172-31-73-42:/etc/docker$ sudo systemctl restart docker
ubuntu@ip-172-31-73-42:/etc/docker$ docker -v
Docker version 24.0.7, build 24.0.7-0ubuntu4.1
ubuntu@ip-172-31-73-42:/etc/docker$
```

Name: Sahil Motiramani Div:D15C Roll No:35

Step 4:Install Kubernetes in all three instances:

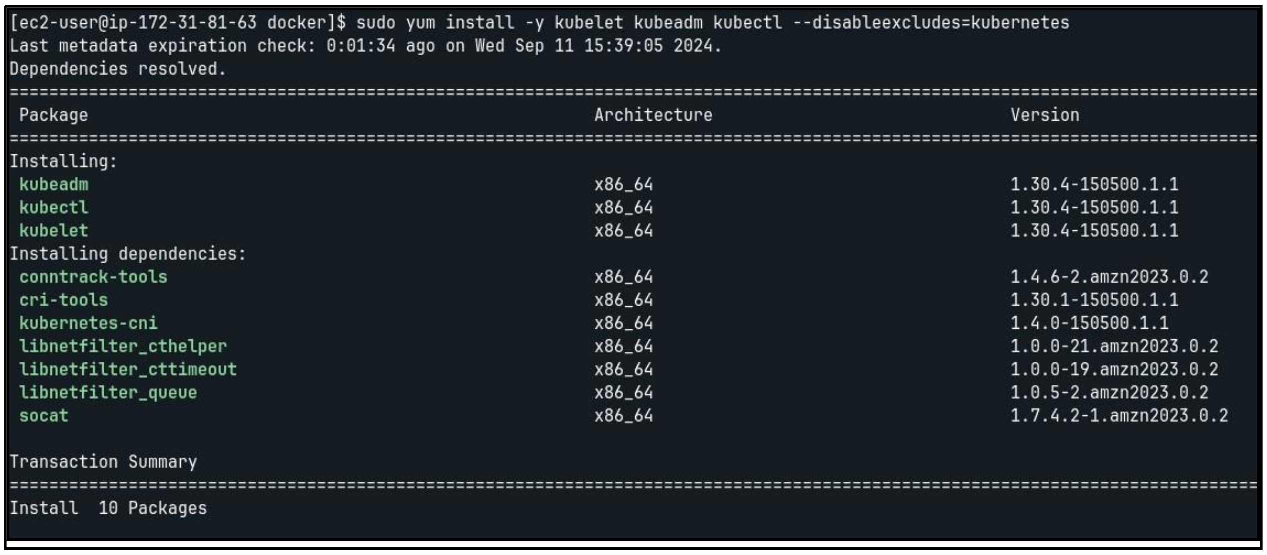


```
[ec2-user@ip-172-31-81-63 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-81-63 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

Add kubernetes repository (paste in terminal);

cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes] name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/ enabled=1 gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/r epomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni EOF

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



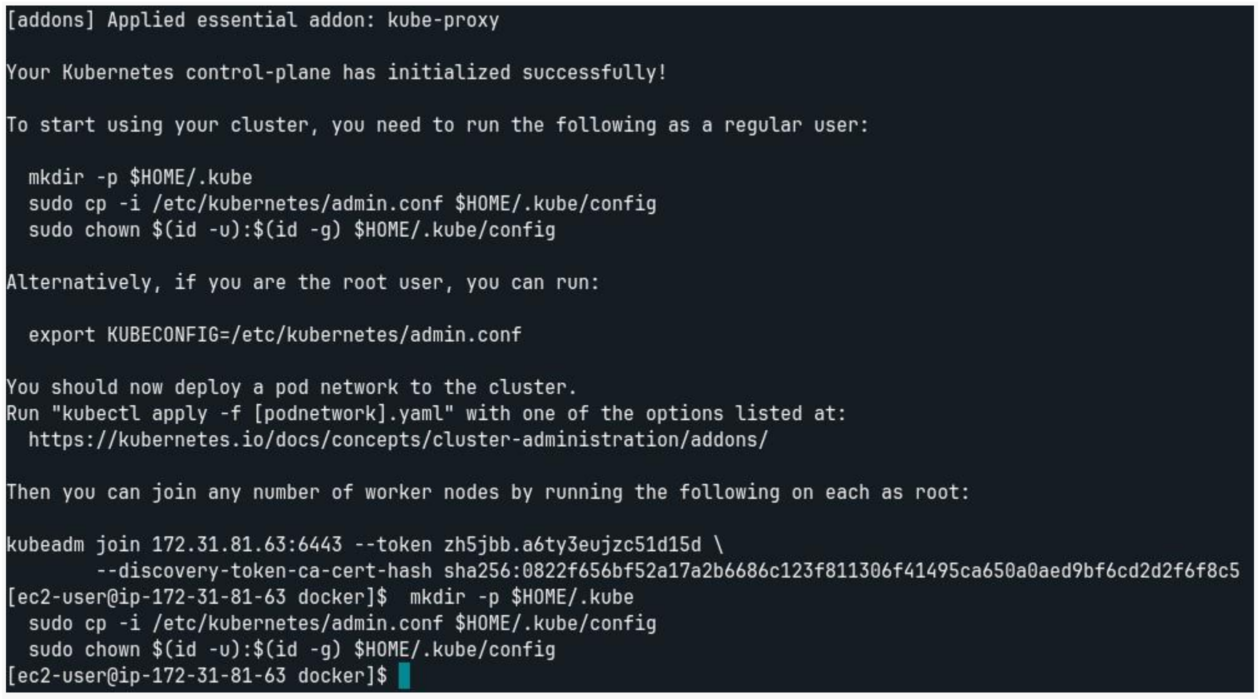
```
[ec2-user@ip-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
==============================================================================================================================
 Package                                                  Architecture                          Version
==============================================================================================================================
Installing:
 kubeadm                                                  x86_64                                1.30.4-150500.1.1
 kubectl                                                  x86_64                                1.30.4-150500.1.1
 kubelet                                                  x86_64                                1.30.4-150500.1.1
Installing dependencies:
 conntrack-tools                                          x86_64                                1.4.6-2.amzn2023.0.2
 cri-tools                                                x86_64                                1.30.1-150500.1.1
 kubernetes-cni                                           x86_64                                1.4.0-150500.1.1
 libnetfilter_cthelper                                    x86_64                                1.0.0-21.amzn2023.0.2
 libnetfilter_cttimeout                                   x86_64                                1.0.0-19.amzn2023.0.2
 libnetfilter_queue                                       x86_64                                1.0.5-2.amzn2023.0.2
 socat                                                    x86_64                                1.7.4.2-1.amzn2023.0.2

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

After installing Kubernetes, we need to configure internet options to allow bridging.

- sudo swapoff-a
- echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
- sudo sysctl-p

Step 5:.Perform this ONLY on the Master machine Initialize kubernetes by typing below command

- sudo kubeadm init--pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all



```
[addons] Applied essential addon: kube-proxy

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 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \
        --discovery-token-ca-cert-hash sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8c5
[ec2-user@ip-172-31-81-63 docker]$  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-63 docker]$
```

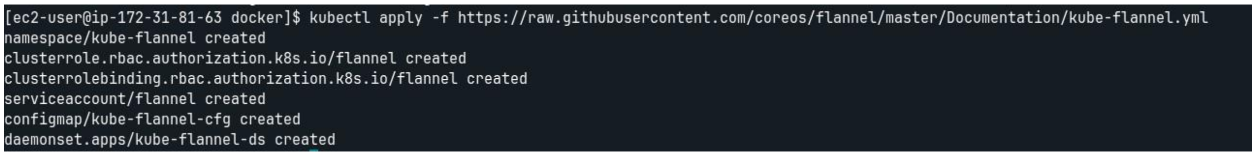
Name: Sahil Motiramani Div:D15C Roll No:35

Copy the mkdir and chown commands from the top and execute them
mkdir-p $HOME/.kube sudo cp-i /etc/kubernetes/admin.conf $HOME/.kube/config sudo chown $(id-u):$(id-g) $HOME/.kube/config

Copy this join link and save it in clipboard (copy from your output as it different for each instance)

kubeadm join 172.31.81.63:6443--token zh5jbb.a6ty3eujzc51d15d \
--discovery-token-ca-cert-hash sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6c d2d2f6f8c5
Then, add a common networking plugin called flammel file as mentioned in the code.

kubectl apply-f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml



```
[ec2-user@ip-172-31-81-63 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

Check the created pod using this command

- kubectl get pods

Step:6. For nodes only;

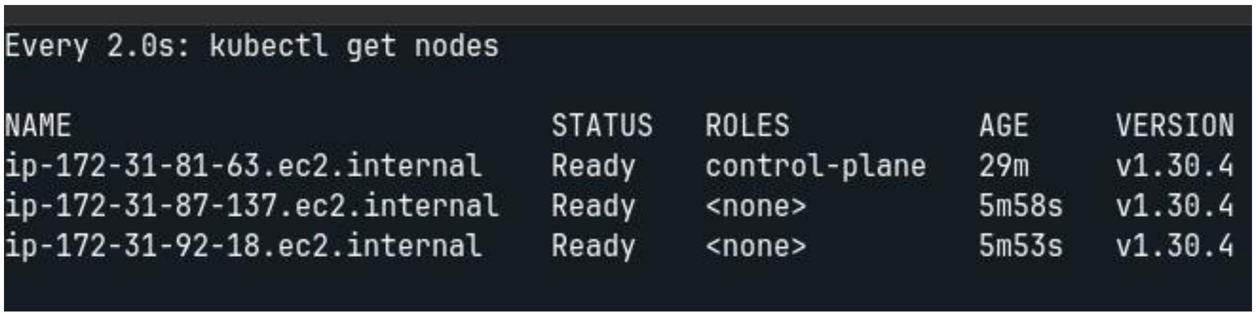
Use the below command on all 2 node machines

- Sudo yum install iproute-tc-y
- sudo systemctl enable kubelet
- sudo systemctl restart kubelet

- kubeadm join 172.31.81.63:6443--token zh5jbb.a6ty3eujzc51d15d \

--discovery-token-ca-cert-hash
sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8 c5

Master control nodes;



```
Every 2.0s: kubectl get nodes

NAME                            STATUS   ROLES           AGE     VERSION
ip-172-31-81-63.ec2.internal    Ready    control-plane   29m     v1.30.4
ip-172-31-87-137.ec2.internal   Ready    <none>          5m58s   v1.30.4
ip-172-31-92-18.ec2.internal    Ready    <none>          5m53s   v1.30.4
```