Name : Sandeshkumar.M.Yadav Div : D15C Roll : 61

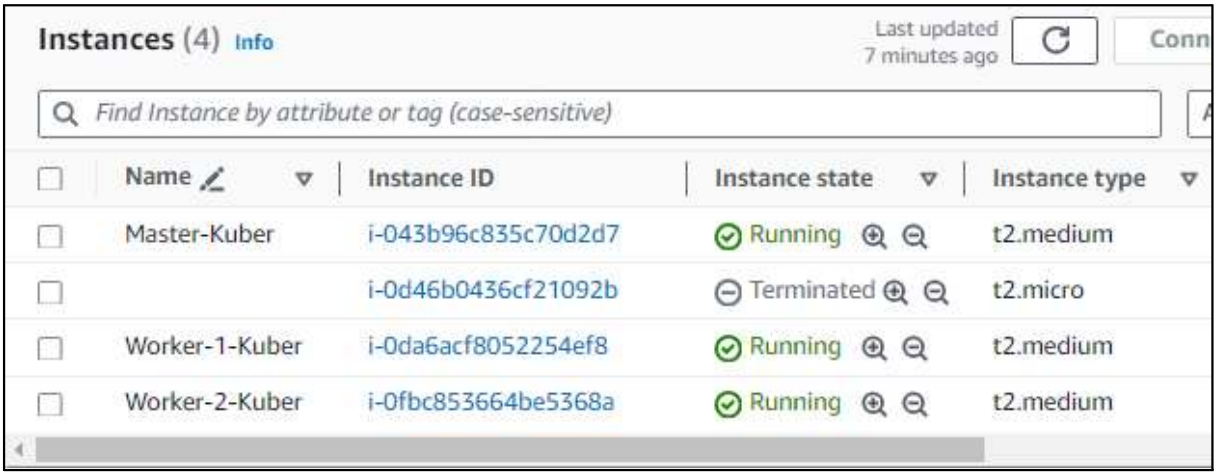
Experiment:3

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

Steps:

Create 3 EC2 Amazon Linux Instances on AWS:

- Created three instances (Master, worker-1, and worker-2) to set up the Kubernetes cluster.



1. From now on, until mentioned, perform these steps on all 3 machines.
   Install Docker on All Machines:

   Commands:

   sudo su
   yum update -y
   yum install docker -y
   service docker start
   systemctl enable docker
   docker --version

   Purpose: Docker is essential for running containers in Kubernetes. The installation commands ensure Docker is updated, installed, started, and set to start on boot.

Name : Sandeshkumar.M.Yadav Div : D15C Roll : 61

```
[root@ip-172-31-19-73 home]# yum update -y
Last metadata expiration check: 0:02:58 ago on Sun Sep 15 04:37:38 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-19-73 home]# yum install docker -y
Last metadata expiration check: 0:03:19 ago on Sun Sep 15 04:37:38 2024.
Package docker-25.0.6-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-19-73 home]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-19-73 home]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-172-31-19-73 home]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-19-73 home]#
```

i-043b96c835c70d2d7 (Master-Kuber)

```
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.20-1.amzn2023.0.1.x86_64                                                    1/10
  Verifying        : docker-25.0.6-1.amzn2023.0.2.x86_64                                                        2/10
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                  3/10
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                   4/10
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                        5/10
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                         6/10
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                  7/10
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                       8/10
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                             9/10
  Verifying        : runc-1.1.13-1.amzn2023.0.1.x86_64                                                         10/10

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-172-31-29-38 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-29-38 ec2-user]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-172-31-29-38 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-29-38 ec2-user]#
```

i-0da6acf8052254ef8 (Worker-1-Kuber)

PublicIPs: 3.92.32.91 PrivateIPs: 172.31.29.38

```
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.20-1.amzn2023.0.1.x86_64                                                    1/10
  Verifying        : docker-25.0.6-1.amzn2023.0.2.x86_64                                                        2/10
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                  3/10
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                   4/10
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                        5/10
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                         6/10
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                  7/10
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                       8/10
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                             9/10
  Verifying        : runc-1.1.13-1.amzn2023.0.1.x86_64                                                         10/10

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-172-31-28-21 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-28-21 ec2-user]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-172-31-28-21 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-28-21 ec2-user]#
```

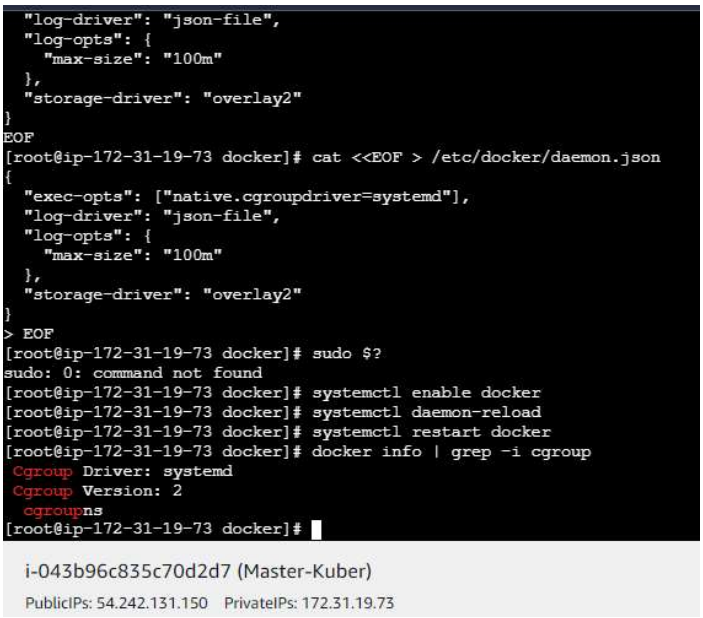
i-0fbc853664be5368a (Worker-2-Kuber)

PublicIPs: 50.19.34.228 PrivateIPs: 172.31.28.21

Then, configure cgroup in a daemon.json file.

```
cd /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
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Purpose: Configures Docker to use the systemd cgroup driver, which is recommended for Kubernetes, and ensures Docker logs are managed and storage is optimized.



```
[root@ip-172-31-29-38 ec2-user]# cd /etc/docker
[root@ip-172-31-29-38 docker]# cat <<EOF > /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
[root@ip-172-31-29-38 docker]# systemctl enable docker
[root@ip-172-31-29-38 docker]# systemctl daemon-reload
[root@ip-172-31-29-38 docker]# systemctl restart docker
[root@ip-172-31-29-38 docker]# docker info | grep -i cgroup
 Cgroup Driver: systemd
 Cgroup Version: 2
  cgroupns
[root@ip-172-31-29-38 docker]#
```

i-0da6acf8052254ef8 (Worker-1-Kuber)

PublicIPs: 3.92.32.91 PrivateIPs: 172.31.29.38

```
[root@ip-172-31-28-21 ec2-user]# cd /etc/docker
[root@ip-172-31-28-21 docker]# cat <<EOF > /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
[root@ip-172-31-28-21 docker]# systemctl enable docker
[root@ip-172-31-28-21 docker]# systemctl daemon-reload
[root@ip-172-31-28-21 docker]# systemctl restart docker
[root@ip-172-31-28-21 docker]# docker info | grep -i cgroup
 Cgroup Driver: systemd
 Cgroup Version: 2
  cgroupns
[root@ip-172-31-28-21 docker]#
```

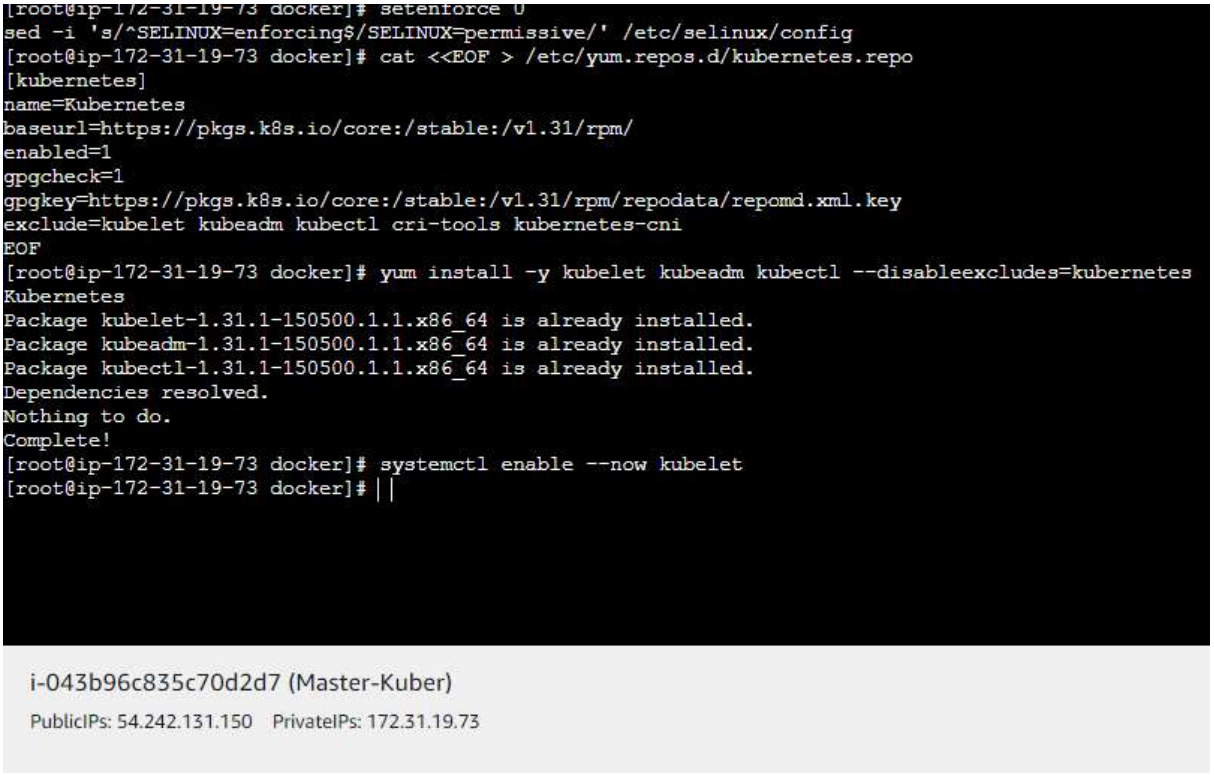
i-0fbc853664be5368a (Worker-2-Kuber)

PublicIPs: 50.19.34.228 PrivateIPs: 172.31.28.21

1. Set SELinux to permissive mode:
   These instructions are for Kubernetes 1.31.
   ```
   sudo setenforce 0
   sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
   ```
2. Add the Kubernetes yum repository. The exclude parameter in the repository definition ensures that the packages related to Kubernetes are not upgraded upon running yum update as there's a special procedure that must be followed for upgrading Kubernetes. Please note that this repository have packages only for Kubernetes 1.31; for other Kubernetes minor versions, you need to change the Kubernetes minor version in the URL to match your desired minor version (you should also check that you are reading the documentation for the version of Kubernetes that you plan to install).
   ```
   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

3. Install kubelet, kubeadm and kubectl:
   sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
4. (Optional) Enable the kubelet service before running kubeadm:
   sudo systemctl enable --now kubelet



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

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1.x
  kubectl-1.31.1-150500.1.1.x86_64                     kubelet-1.31.1-150500.1.1.x86
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   libnetfilter_cttimeout-1.0.0-

Complete!
[root@ip-172-31-29-38 docker]# systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/li
[root@ip-172-31-29-38 docker]#
```

i-0da6acf8052254ef8 (Worker-1-Kuber)

PublicIPs: 3.92.32.91 PrivateIPs: 172.31.29.38

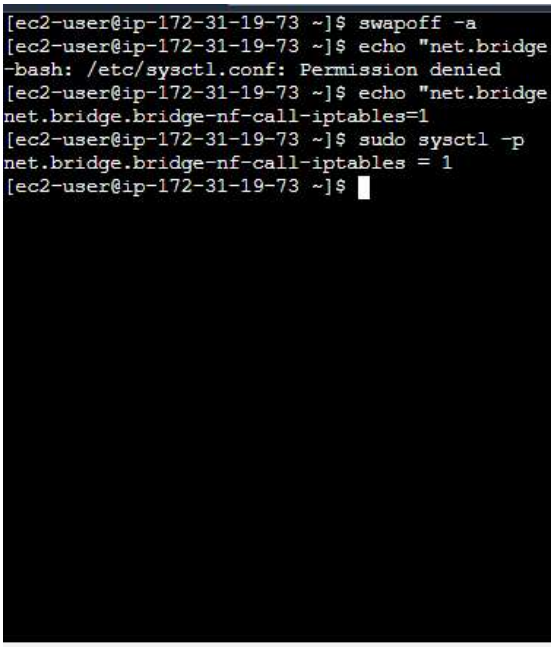
```
  Installing       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
  Installing       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  Installing       : kubelet-1.31.1-150500.1.1.x86_64
  Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64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

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64          cri-tools-1.31.1-150500.1.1
  kubectl-1.31.1-150500.1.1.x86_64                     kubelet-1.31.1-150500.1.1.x
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   libnetfilter_cttimeout-1.0.

Complete!
[root@ip-172-31-28-21 docker]# systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/
[root@ip-172-31-28-21 docker]#
```

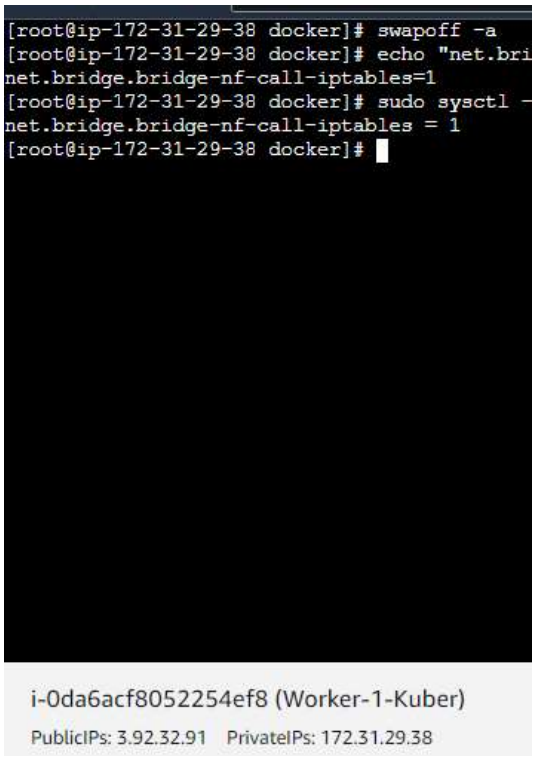
i-0fbc853664be5368a (Worker-2-Kuber)

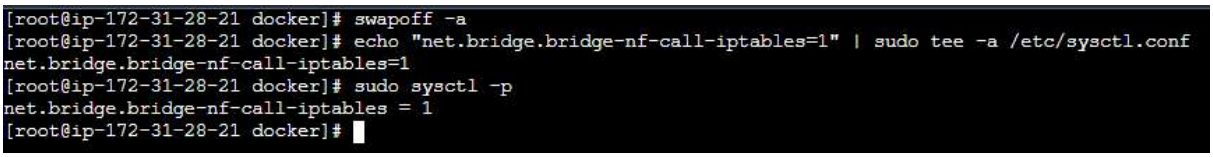
PublicIPs: 50.19.34.228 PrivateIPs: 172.31.28.21

5. 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

   Purpose: Disables swap memory and configures network settings to ensure proper networking and performance for Kubernetes.







6. Perform this ONLY on the Master machine
   kubeadm init first runs a series of prechecks to ensure that the machine is ready to run Kubernetes. These prechecks expose warnings and exit on errors. kubeadm init then downloads and installs the cluster control plane components. This may take several minutes. After it finishes you should see:

   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

   You should now deploy a Pod network to the cluster.

Run
"kubectl apply -f
https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml"
with one of the options listed at : /docs/concepts/cluster-administration/addons/

Purpose: Initialises the Kubernetes master, sets up the kubeconfig file for cluster access, and deploys the Flannel network plugin for pod networking.

You can now join any number of machines by running the following on each node
as root:

```
[root@ip-172-31-19-73 /]# mkdir -p $HOME/.kube
[root@ip-172-31-19-73 /]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
cp: overwrite '/root/.kube/config'? y
[root@ip-172-31-19-73 /]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-19-73 /]# kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[root@ip-172-31-19-73 /]# kubectl get pods
No resources found in default namespace.
[root@ip-172-31-19-73 /]#
[root@ip-172-31-19-73 /]# kubectl get pods --all-namespaces
NAMESPACE      NAME                                                READY   STATUS              RESTARTS          AGE
kube-flannel   kube-flannel-ds-lbgbd                               0/1     CrashLoopBackOff    3 (44s ago)       99s
kube-system    coredns-7c65d6cfc9-8vjkk                            0/1     ContainerCreating   0                 3m13s
kube-system    coredns-7c65d6cfc9-bl78b                            0/1     ContainerCreating   0                 3m13s
kube-system    etcd-ip-172-31-19-73.ec2.internal                   1/1     Running             19 (3m5s ago)     3m21s
kube-system    kube-apiserver-ip-172-31-19-73.ec2.internal         1/1     Running             20 (3m25s ago)    3m21s
kube-system    kube-controller-manager-ip-172-31-19-73.ec2.internal   1/1     Running             21 (3m5s ago)     3m21s
kube-system    kube-proxy-j9fdq                                    0/1     CrashLoopBackOff    2 (16s ago)       3m13s
kube-system    kube-scheduler-ip-172-31-19-73.ec2.internal         1/1     Running             21 (3m ago)       3m57s
[root@ip-172-31-19-73 /]# kubectl get nodes
NAME                           STATUS   ROLES           AGE     VERSION
ip-172-31-19-73.ec2.internal   Ready    control-plane   4m30s   v1.31.1
```

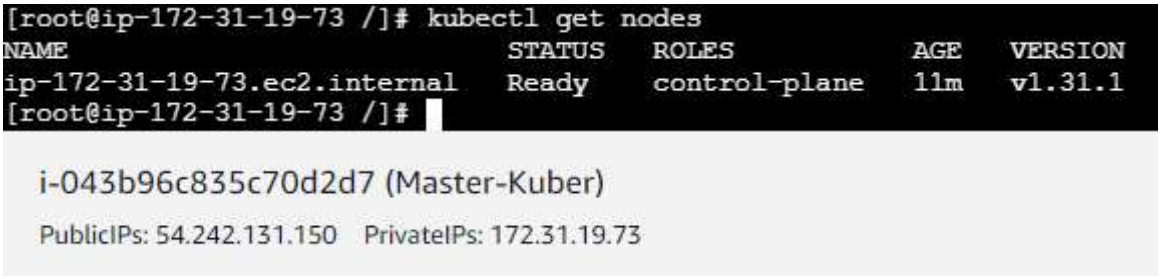
sudo yum install iproute-tc socat -y
sudo systemctl enable kubelet
sudo systemctl restart kubelet

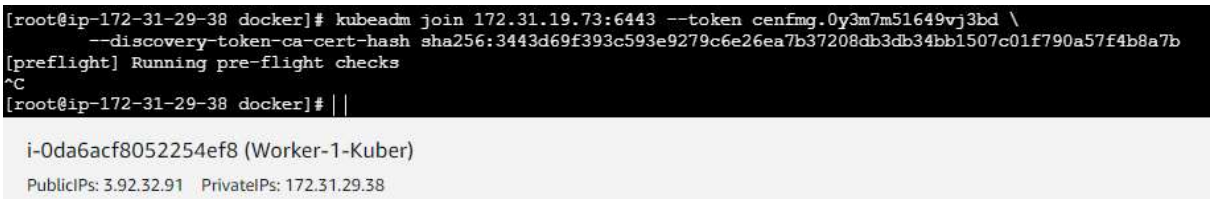
Purpose: Installs necessary packages on worker nodes, ensures `kubelet` starts, and joins the nodes to the Kubernetes cluster using the provided token and hash.

```
[root@ip-172-31-29-38 docker]# sudo yum install iproute-tc socat -y
sudo systemctl enable kubelet
sudo systemctl restart kubelet
Last metadata expiration check: 0:57:00 ago on Sun Sep 15 04:57:12 2024.
Dependencies resolved.
================================================================================
 Package                         Architecture                    Version
================================================================================
Installing:
 iproute-tc                      x86_64                          5.10.0-2.amzn
 socat                           x86_64                          1.7.4.2-1.amz

Transaction Summary
================================================================================
Install  2 Packages

Total download size: 758 k
Installed size: 2.0 M
Downloading Packages:
(1/2): iproute-tc-5.10.0-2.amzn2023.0.5.x86_64.rpm
(2/2): socat-1.7.4.2-1.amzn2023.0.2.x86_64.rpm
--------------------------------------------------------------------------------
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
```

Paste this to both worker node to join:

kubeadm join 172.31.19.73:6443 --token cenfmg.0y3m7m51649vj3bd \
--discovery-token-ca-cert-hash
sha256:3443d69f393c593e9279c6e26ea7b37208db3db34bb1507c01f790a57f4b8a7b





Conclusion :

In this experiment, we aimed to understand Kubernetes cluster architecture by setting up a Kubernetes cluster on AWS EC2 instances. We started by creating three Amazon Linux instances (Master, worker-1, and worker-2). Docker was installed and configured on all nodes, SELinux was set to permissive mode, and the Kubernetes repository was added. We then installed the Kubernetes packages and initialised the cluster on the master node. After configuring networking and deploying the Flannel plugin, we joined the worker nodes to the cluster. Despite following the steps, the setup did not complete successfully, indicating issues with establishing a fully functional Kubernetes cluster with all nodes connected.