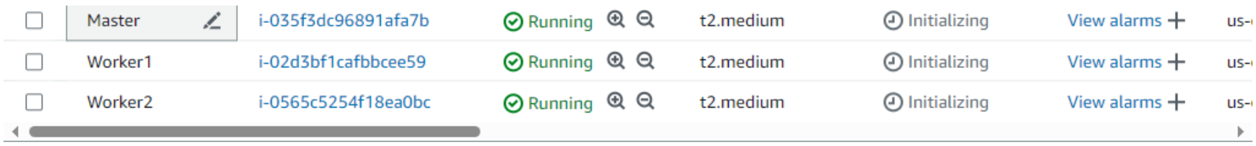
**Veydant Sharma D15C 50 AdvDevops EXP3**

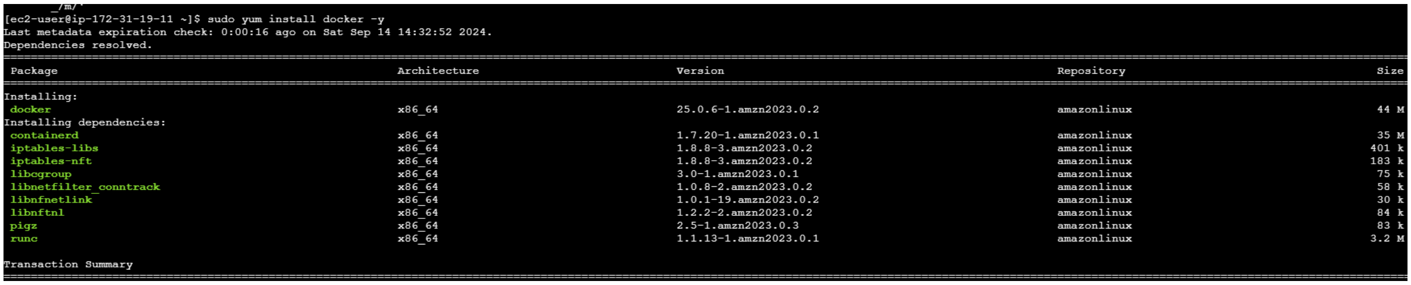
**Aim:** To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machine

1. Create 3 ec2 instances with an OS as Amazon Linux. Select the instance type as t2.medium.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| □ | Master | i-035f3dc96891afa7b | Running | t2.medium | Initializing | View alarms + | us- |
| □ | Worker1 | i-02d3bf1cafbbcee59 | Running | t2.medium | Initializing | View alarms + | us- |
| □ | Worker2 | i-0565c5254f18ea0bc | Running | t2.medium | Initializing | View alarms + | us- |

2. Install docker with the command:
   sudo yum install docker -y

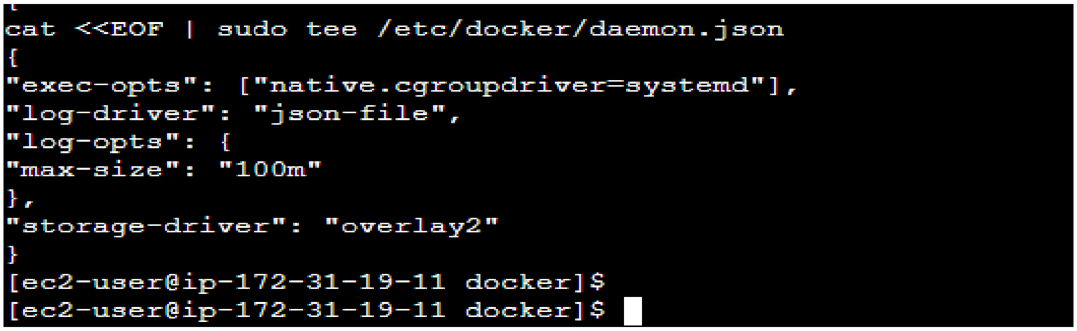


```
[ec2-user@ip-172-31-19-11 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:00:16 ago on Sat Sep 14 14:32:52 2024.
Dependencies resolved.
==========================================================================================================
 Package                        Architecture        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
==========================================================================================================
```

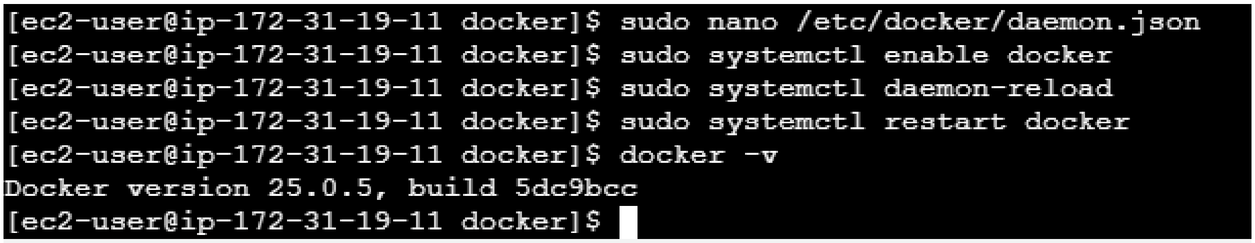
3. Run the following command

```
cd /etc/docker
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"
}
```



```
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"
}
[ec2-user@ip-172-31-19-11 docker]$
[ec2-user@ip-172-31-19-11 docker]$
```

4. Enable docker
   - sudo systemctl enable docker
   - sudo systemctl daemon-reload
   - sudo systemctl restart docker

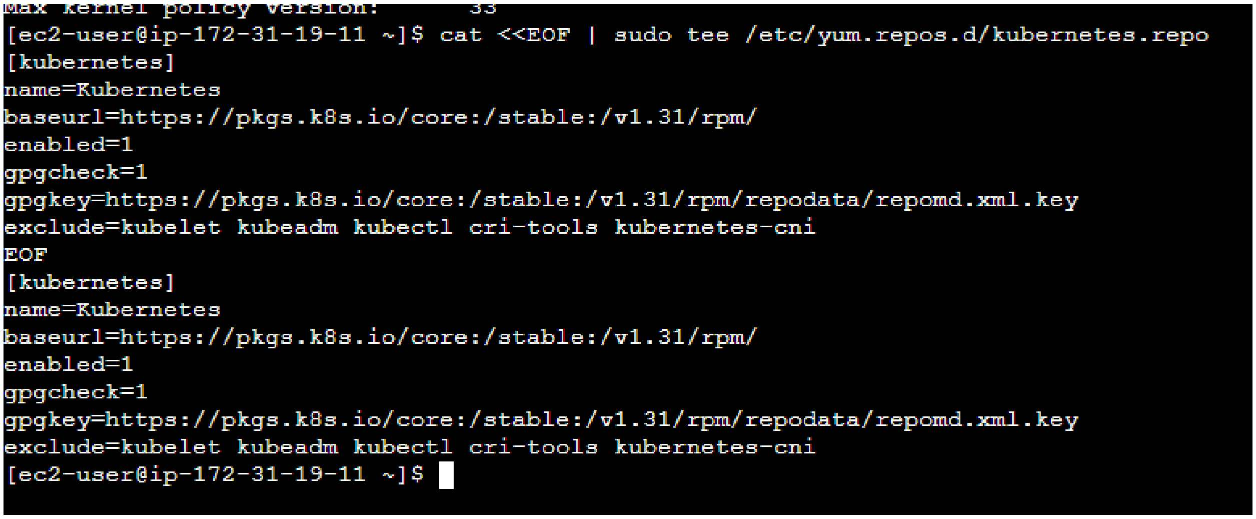


```
[ec2-user@ip-172-31-19-11 docker]$ sudo nano /etc/docker/daemon.json
[ec2-user@ip-172-31-19-11 docker]$ sudo systemctl enable docker
[ec2-user@ip-172-31-19-11 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-19-11 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-19-11 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-19-11 docker]$
```

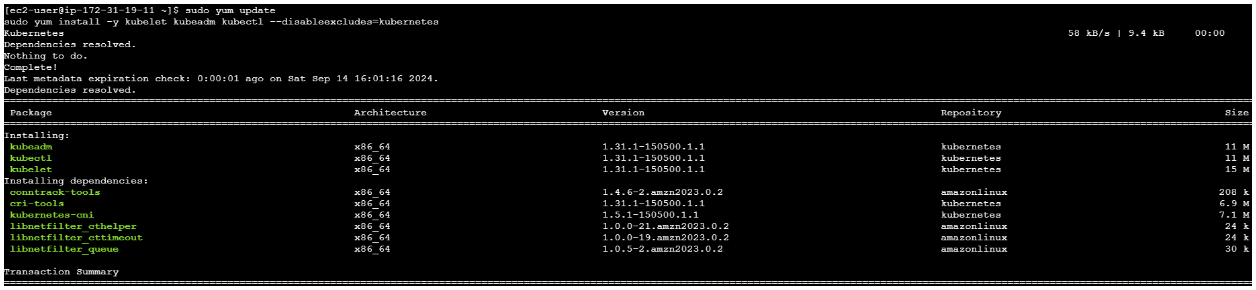
5. Install Kubernetes

   Add kubernetes repository

```
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
```



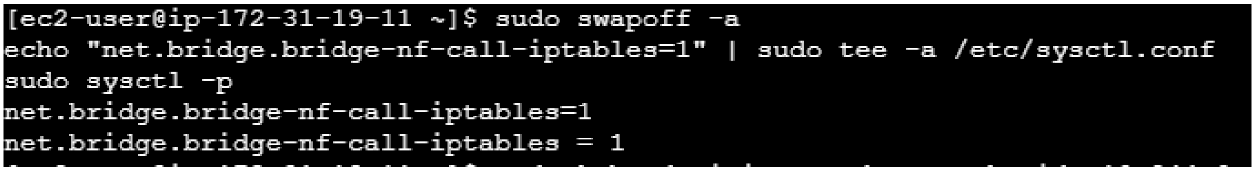
```
Max kernel policy version:      33
[ec2-user@ip-172-31-19-11 ~]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[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
[ec2-user@ip-172-31-19-11 ~]$
```



```
[ec2-user@ip-172-31-19-11 ~]$ sudo yum update
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes                                                                    58 kB/s | 9.4 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:01 ago on Sat Sep 14 16:01:16 2024.
Dependencies resolved.
===========================================================================================================
 Package                     Architecture     Version                       Repository                Size
===========================================================================================================
Installing:
 kubeadm                     x86_64           1.31.1-150500.1.1             kubernetes                11 M
 kubectl                     x86_64           1.31.1-150500.1.1             kubernetes                11 M
 kubelet                     x86_64           1.31.1-150500.1.1             kubernetes                15 M
Installing dependencies:
 conntrack-tools             x86_64           1.4.6-2.amzn2023.0.2          amazonlinux              208 k
 cri-tools                   x86_64           1.31.1-150500.1.1             kubernetes               6.9 M
 kubernetes-cni              x86_64           1.5.1-150500.1.1              kubernetes               7.1 M
 libnetfilter_cthelper       x86_64           1.0.0-21.amzn2023.0.2         amazonlinux               24 k
 libnetfilter_cttimeout      x86_64           1.0.0-19.amzn2023.0.2         amazonlinux               24 k
 libnetfilter_queue          x86_64           1.0.5-2.amzn2023.0.2          amazonlinux               30 k

Transaction Summary
===========================================================================================================
```

6. Configure Internet options

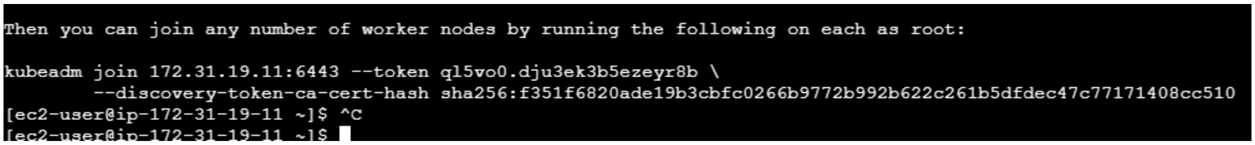


```
[ec2-user@ip-172-31-19-11 ~]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
```

7. Perform the following only on master machine

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

Save the join command



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

kubeadm join 172.31.19.11:6443 --token ql5vo0.dju3ek3b5ezeyr8b \
        --discovery-token-ca-cert-hash sha256:f351f6820ade19b3cbfc0266b9772b992b622c261b5dfdec47c77171408cc510
[ec2-user@ip-172-31-19-11 ~]$ ^C
[ec2-user@ip-172-31-19-11 ~]$
```

Add the networking plugin (Flammel file) using the following command.

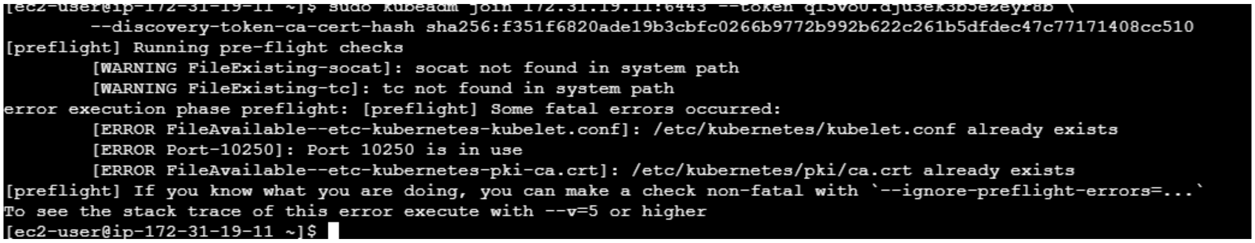
kubectl apply -f
https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml

8. Perform the following commands in worker node only.

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

**Veydant Sharma D15C 50 AdvDevops EXP3**

Run the join command saved in notepad previously.



```
[ec2-user@ip-172-31-19-11 ~]$ sudo kubeadm join 172.31.19.11:6443 --token qi5vo0.dju3ek3b5ezeyr8b \
        --discovery-token-ca-cert-hash sha256:f351f6820ade19b3cbfc0266b9772b992b622c261b5dfdec47c77171408cc510
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR FileAvailable--etc-kubernetes-kubelet.conf]: /etc/kubernetes/kubelet.conf already exists
        [ERROR Port-10250]: Port 10250 is in use
        [ERROR FileAvailable--etc-kubernetes-pki-ca.crt]: /etc/kubernetes/pki/ca.crt already exists
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
[ec2-user@ip-172-31-19-11 ~]$
```

The command was successfully run but the execution was not completed. This could be due to misconfiguration of clusters or some network connectivity issue.

**Conclusion:**

In this experiment, we set up a Kubernetes cluster across three Amazon Linux EC2 instances and installed the necessary Kubernetes components on each. The master node was configured using `kubeadm`, and we deployed the Flannel network plugin to manage pod networking. Worker nodes were then added to the cluster using a join command created during the master node setup. While we successfully established the Kubernetes cluster on EC2 instances and gained a clear understanding of the process, we encountered a delay in connecting the worker nodes to the master. This delay may have been due to network connectivity issues or configuration errors.