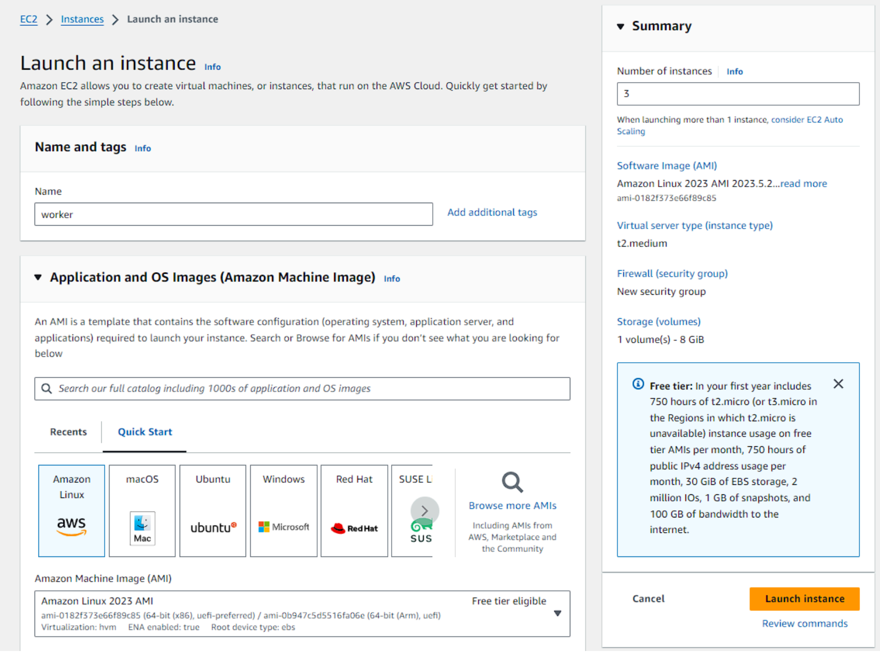
Name - Brijesh Sharma **Div - D15C** **Roll No. 48**

Experiment - 3

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

1) Create 3 EC-2 instances with all running on Amazon Linux as OS with inbound SSH allowed.
2) To efficient run kubernetes cluster, select instance type of at least t2.medium as kubernetes recommends at least 2 vCPU to run smoothly



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: worker

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0182f373e66f89c85 (64-bit (x86), uefi-preferred) / ami-0b947c5d5516fa06e (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances: 3

When launching more than 1 instance, consider EC2 Auto Scaling

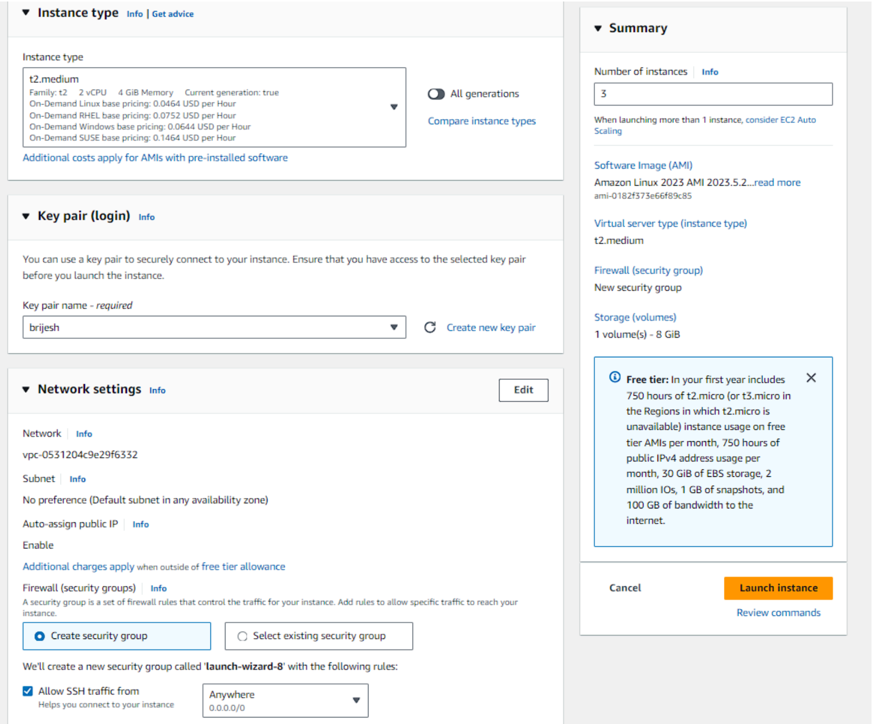
Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more
ami-0182f373e66f89c85

Virtual server type (instance type): t2.medium

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Instance type Info | Get advice

Instance type: t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: brijesh

Network settings Info

Network: vpc-0531204c9e29f6332

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-8**' with the following rules:

☑ Allow SSH traffic from: Anywhere 0.0.0.0/0
Helps you connect to your instance

Summary

Number of instances: 3

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more
ami-0182f373e66f89c85

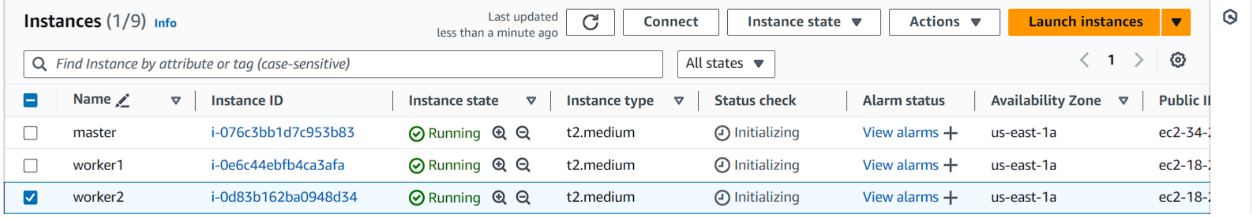
Virtual server type (instance type): t2.medium

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

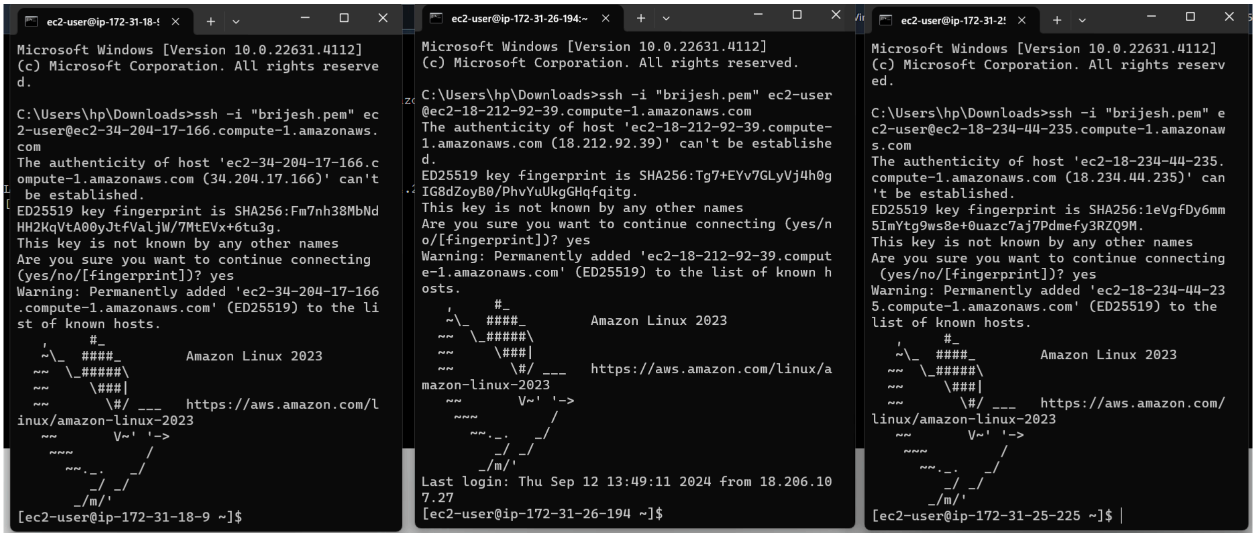
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

3) Three instance are ready - master, worker1, and worker2.


Instances (1/9) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

■	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public II
□	master	i-076c3bb1d7c953b83	✓ Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-34-
□	worker1	i-0e6c44ebfb4ca3afa	✓ Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-18-
☑	worker2	i-0d83b162ba0948d34	✓ Running	t2.medium	Initializing	View alarms +	us-east-1a	ec2-18-

4) Connect the instances to the local terminal using the SSH client.



```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserve
d.

C:\Users\hp\Downloads>ssh -i "brijesh.pem" ec
2-user@ec2-34-204-17-166.compute-1.amazonaws.
com
The authenticity of host 'ec2-34-204-17-166.c
ompute-1.amazonaws.com (34.204.17.166)' can't
 be established.
ED25519 key fingerprint is SHA256:Fm7nh38MbNd
HH2KqVtA00yJtfValjW/7MtEVx+6tu3g.
This key is not known by any other names
Are you sure you want to continue connecting
(yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-204-17-166
.compute-1.amazonaws.com' (ED25519) to the li
st of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/l
inux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-18-9 ~]$
```

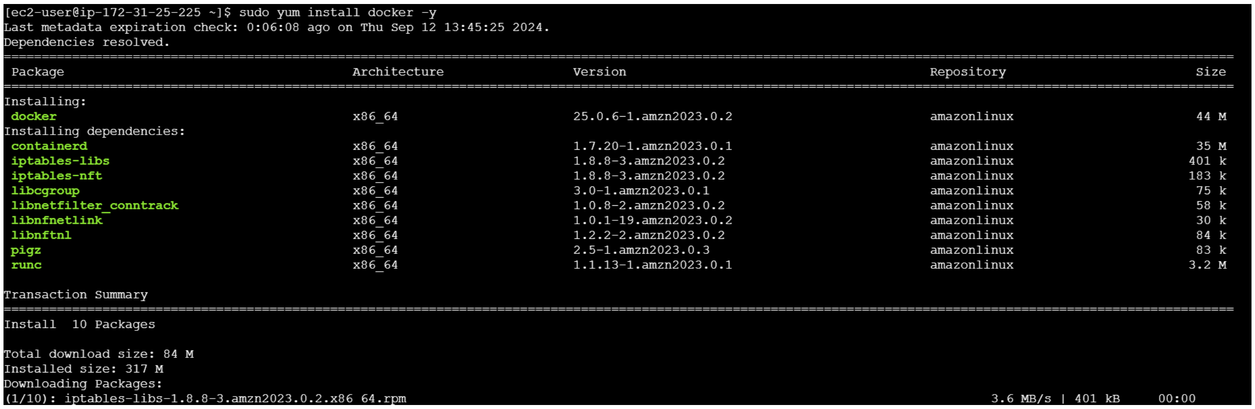
```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Downloads>ssh -i "brijesh.pem" ec2-user
@ec2-18-212-92-39.compute-1.amazonaws.com
The authenticity of host 'ec2-18-212-92-39.compute-
1.amazonaws.com (18.212.92.39)' can't be establishe
d.
ED25519 key fingerprint is SHA256:Tg7+EYv7GLyVj4h0g
IG8dZoyB0/PhvYuUkgGHqfqitg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/n
o/[fingerprint])? yes
Warning: Permanently added 'ec2-18-212-92-39.comput
e-1.amazonaws.com' (ED25519) to the list of known h
osts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/a
mazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Sep 12 13:49:11 2024 from 18.206.10
7.27
[ec2-user@ip-172-31-26-194 ~]$
```

```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserv
ed.

C:\Users\hp\Downloads>ssh -i "brijesh.pem" e
c2-user@ec2-18-234-44-235.compute-1.amazonaw
s.com
The authenticity of host 'ec2-18-234-44-235.
compute-1.amazonaws.com (18.234.44.235)' can
't be established.
ED25519 key fingerprint is SHA256:1eVgfDy6mm
5ImYtg9ws8e+0uazc7aj7Pdmefy3RZQ9M.
This key is not known by any other names
Are you sure you want to continue connecting
 (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-234-44-23
5.compute-1.amazonaws.com' (ED25519) to the
list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/
linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-25-225 ~]$
```

5) Run the following commands on all the machines.

Install Docker

a) sudo yum install docker -y



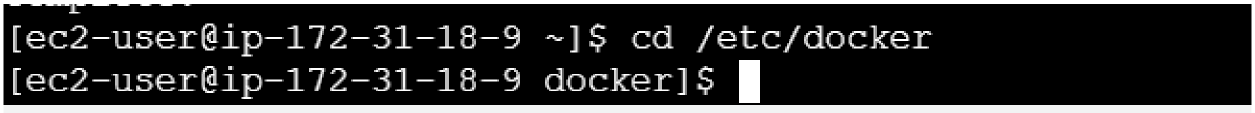
```
[ec2-user@ip-172-31-25-225 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:06:08 ago on Thu Sep 12 13:45:25 2024.
Dependencies resolved.
================================================================================================================================
 Package                          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
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm                                          3.6 MB/s | 401 kB     00:00
```

b) Then, configure cgroup in a daemon.json file by using following commands. This allows kubernetes to manage host more efficiently -

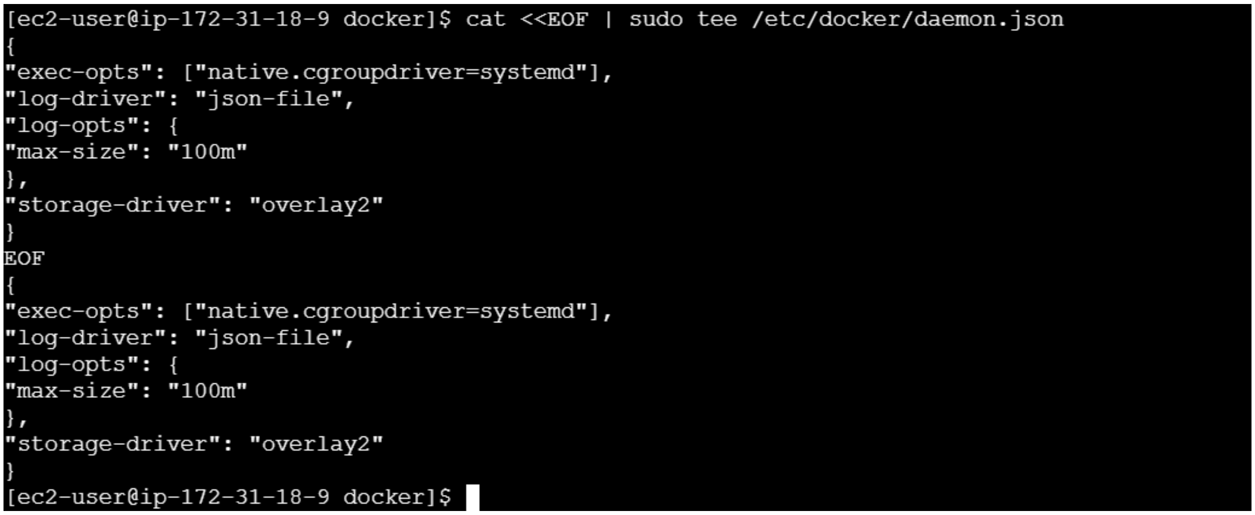
- cd /etc/docker

Run the scripts below -

```
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
```



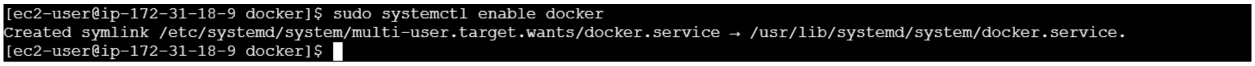
```
[ec2-user@ip-172-31-18-9 ~]$ cd /etc/docker
[ec2-user@ip-172-31-18-9 docker]$
```



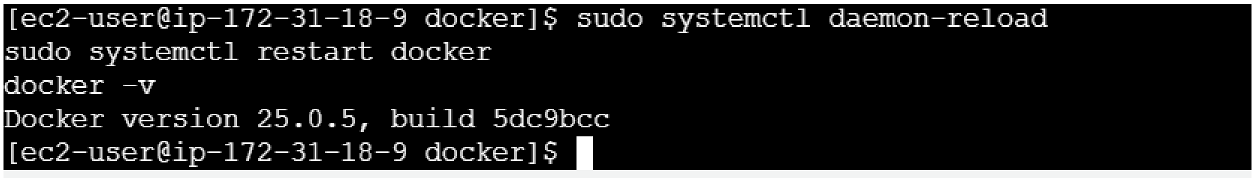
```
[ec2-user@ip-172-31-18-9 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
[ec2-user@ip-172-31-18-9 docker]$
```

c) After configuring restart docker service service :

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



```
[ec2-user@ip-172-31-18-9 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-18-9 docker]$
```



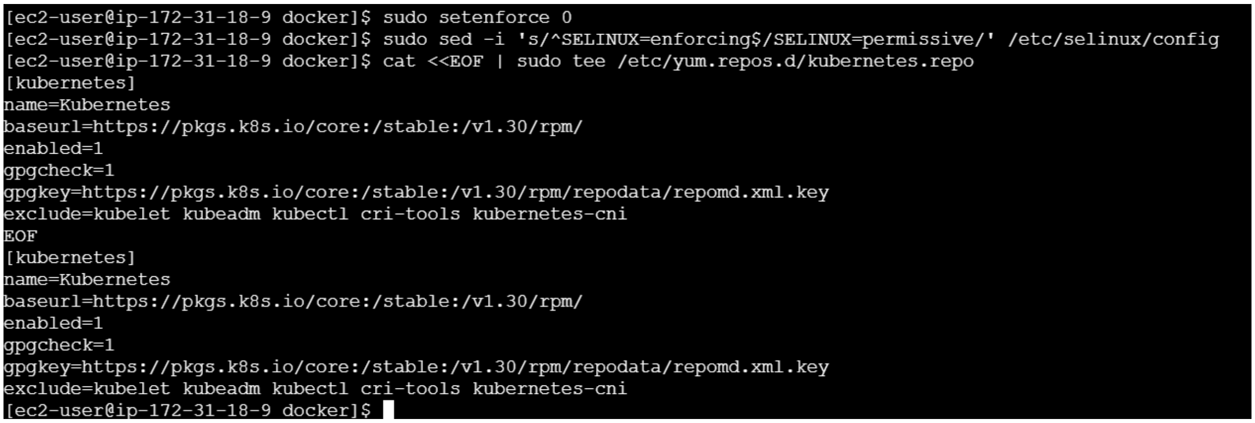
```
[ec2-user@ip-172-31-18-9 docker]$ sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-18-9 docker]$
```

Install Kubernetes

a) SELinux needs to be disabled before configuring kubelet to avoid interference with kubernetes api server

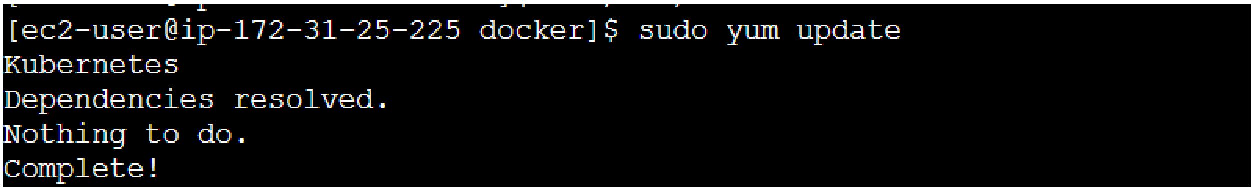
- sudo setenforce 0
- sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
- Add kubernetes repository (paste in terminal)

  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/repomd.xml.key
  exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
  EOF



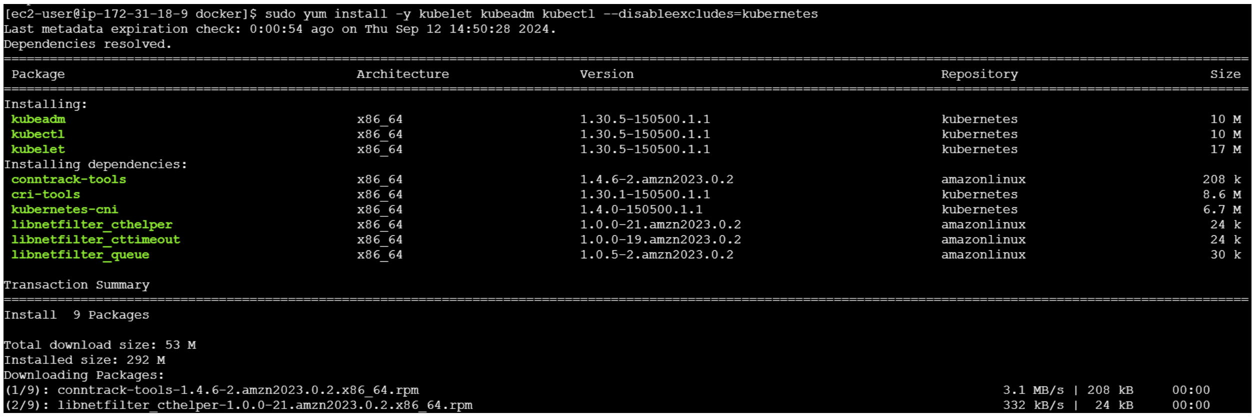
```
[ec2-user@ip-172-31-18-9 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-18-9 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-18-9 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[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/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[ec2-user@ip-172-31-18-9 docker]$
```

b) sudo yum update



```
[ec2-user@ip-172-31-25-225 docker]$ sudo yum update
Kubernetes
Dependencies resolved.
Nothing to do.
Complete!
```

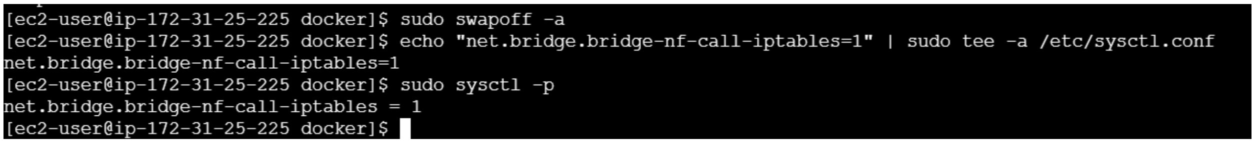
c) sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes



```
[ec2-user@ip-172-31-18-9 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:54 ago on Thu Sep 12 14:50:28 2024.
Dependencies resolved.
==============================================================================================================================================================
 Package                                      Architecture                 Version                                         Repository                    Size
==============================================================================================================================================================
Installing:
 kubeadm                                      x86_64                       1.30.5-150500.1.1                               kubernetes                    10 M
 kubectl                                      x86_64                       1.30.5-150500.1.1                               kubernetes                    10 M
 kubelet                                      x86_64                       1.30.5-150500.1.1                               kubernetes                    17 M
Installing dependencies:
 conntrack-tools                              x86_64                       1.4.6-2.amzn2023.0.2                            amazonlinux                  208 k
 cri-tools                                    x86_64                       1.30.1-150500.1.1                               kubernetes                   8.6 M
 kubernetes-cni                               x86_64                       1.4.0-150500.1.1                                kubernetes                   6.7 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
==============================================================================================================================================================
Install  9 Packages

Total download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm                                                                 3.1 MB/s | 208 kB     00:00
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm                                                          332 kB/s |  24 kB     00:00
```

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

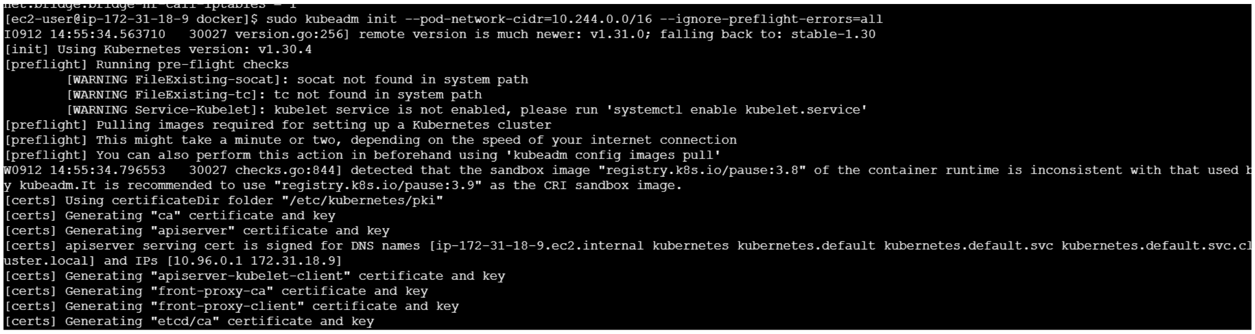
- sudo swapoff -a
- echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
- sudo sysctl -p



```
[ec2-user@ip-172-31-25-225 docker]$ sudo swapoff -a
[ec2-user@ip-172-31-25-225 docker]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-25-225 docker]$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-25-225 docker]$ 
```

6) To perform only on Master machine

a) Initialize kubernetes by typing below command
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all



```
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-18-9 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
I0912 14:55:34.563710   30027 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0912 14:55:34.796553   30027 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used b
y kubeadm.It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-18-9.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cl
uster.local] and IPs [10.96.0.1 172.31.18.9]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
```

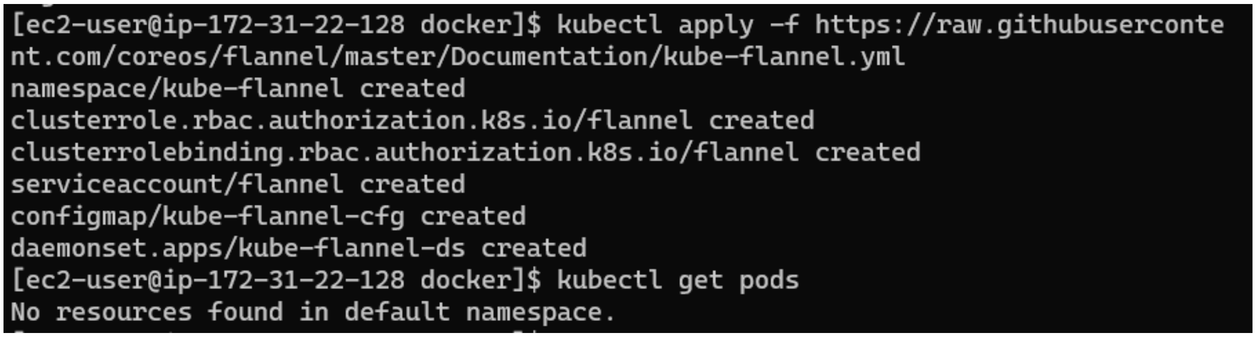
b) Copy this join link

kubeadm join 172.31.22.128:6443 --token 2nzclk.1ek0i93tsqnednb9 \
    --discovery-token-ca-cert-hash
sha256:e7c55b0579b7e928431704c459e9c9c521c4af034e3d346f3418e1afc672928d

c) Run the below command -

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

d) 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

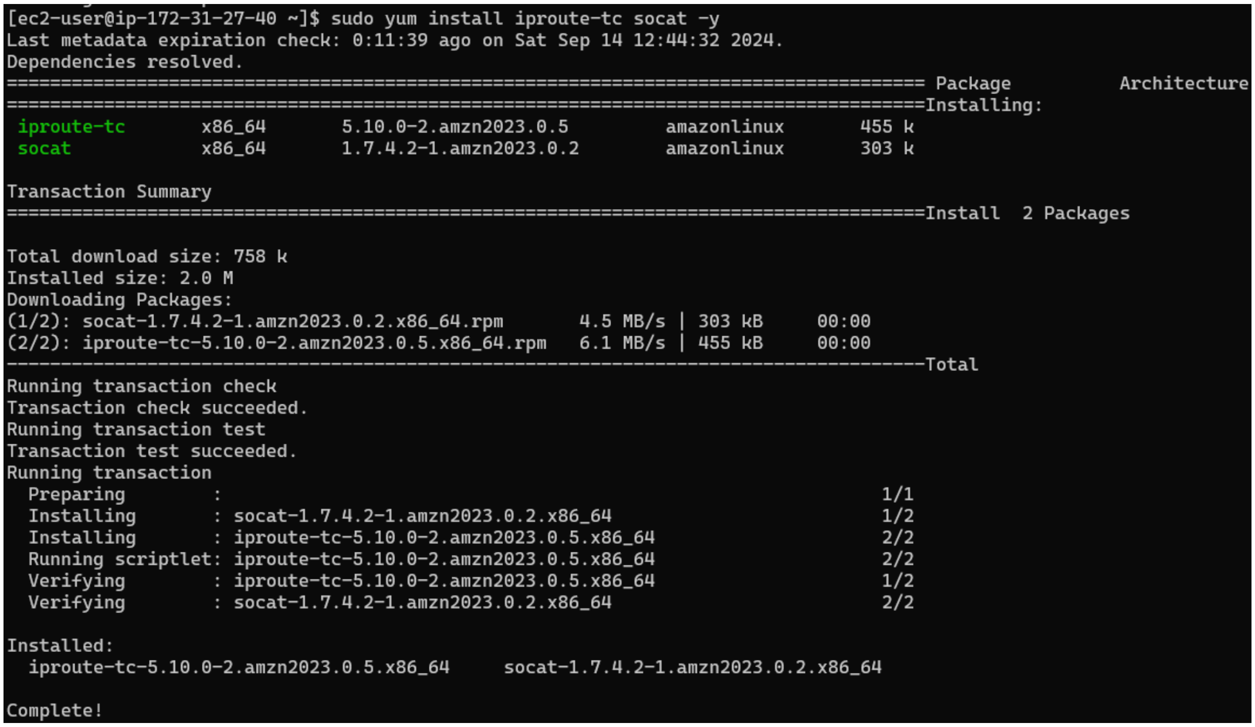


```
[ec2-user@ip-172-31-22-128 docker]$ kubectl apply -f https://raw.githubuserconte
nt.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-22-128 docker]$ kubectl get pods
No resources found in default namespace.
```

e) Check the created pod using this command
- kubectl get pods

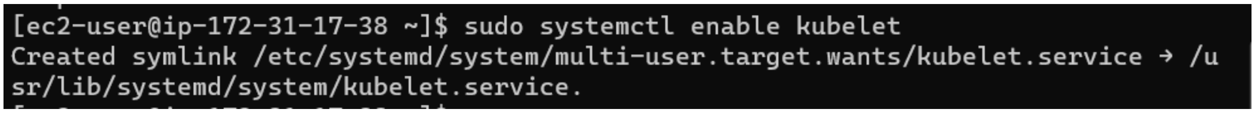
7) To perform on both worker machine -

a) sudo yum install iproute-tc socat -y



```
[ec2-user@ip-172-31-27-40 ~]$ sudo yum install iproute-tc socat -y
Last metadata expiration check: 0:11:39 ago on Sat Sep 14 12:44:32 2024.
Dependencies resolved.
================================================================================ Package         Architecture
================================================================================Installing:
 iproute-tc       x86_64       5.10.0-2.amzn2023.0.5         amazonlinux       455 k
 socat            x86_64       1.7.4.2-1.amzn2023.0.2        amazonlinux       303 k

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

Total download size: 758 k
Installed size: 2.0 M
Downloading Packages:
(1/2): socat-1.7.4.2-1.amzn2023.0.2.x86_64.rpm       4.5 MB/s | 303 kB     00:00
(2/2): iproute-tc-5.10.0-2.amzn2023.0.5.x86_64.rpm   6.1 MB/s | 455 kB     00:00
--------------------------------------------------------------------------------Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : socat-1.7.4.2-1.amzn2023.0.2.x86_64                    1/2
  Installing       : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64                2/2
  Running scriptlet: iproute-tc-5.10.0-2.amzn2023.0.5.x86_64                2/2
  Verifying        : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64                1/2
  Verifying        : socat-1.7.4.2-1.amzn2023.0.2.x86_64                    2/2

Installed:
  iproute-tc-5.10.0-2.amzn2023.0.5.x86_64     socat-1.7.4.2-1.amzn2023.0.2.x86_64

Complete!
```

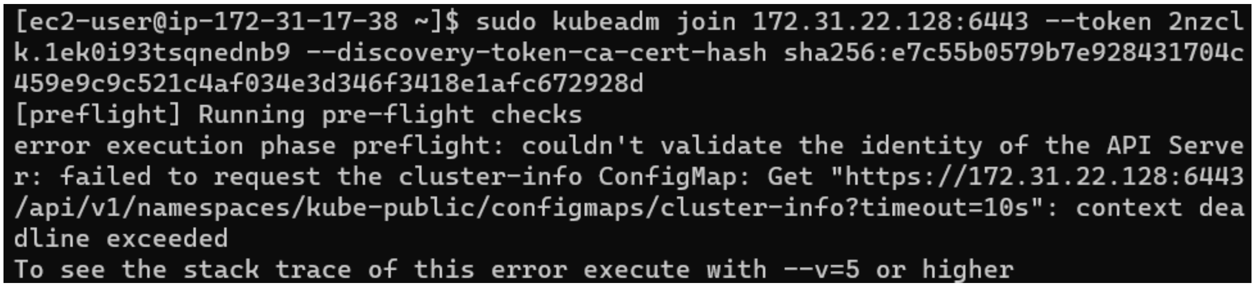
b) sudo systemctl enable kubelet



```
[ec2-user@ip-172-31-17-38 ~]$ sudo systemctl enable kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /u
sr/lib/systemd/system/kubelet.service.
```

c) sudo systemctl restart kubelet

d) kubeadm join 172.31.22.128:6443 --token 2nzclk.1ek0i93tsqnednb9 \
--discovery-token-ca-cert-hash
sha256:e7c55b0579b7e928431704c459e9c9c521c4af034e3d346f3418e1afc672928d



```
[ec2-user@ip-172-31-17-38 ~]$ sudo kubeadm join 172.31.22.128:6443 --token 2nzcl
k.1ek0i93tsqnednb9 --discovery-token-ca-cert-hash sha256:e7c55b0579b7e928431704c
459e9c9c521c4af034e3d346f3418e1afc672928d
[preflight] Running pre-flight checks
error execution phase preflight: couldn't validate the identity of the API Serve
r: failed to request the cluster-info ConfigMap: Get "https://172.31.22.128:6443
/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context dea
dline exceeded
To see the stack trace of this error execute with --v=5 or higher
```

If it gives error or refusing connection, restart the master server using sudo systemctl restart kubelet and try to connect.

With the help of command the worker nodes are connected master node and is ready to do task assigned by master node.

Conclusion -

In this experiment, we connected master nodes from Kubernetes to the worker nodes successfully First, we created the instances and connected with Kubernetes, while installing and configuring, there were some packages, that were needed to be installed separately. Even while connecting the nodes, error occurs and hence, system needs to be restarted and connected properly..