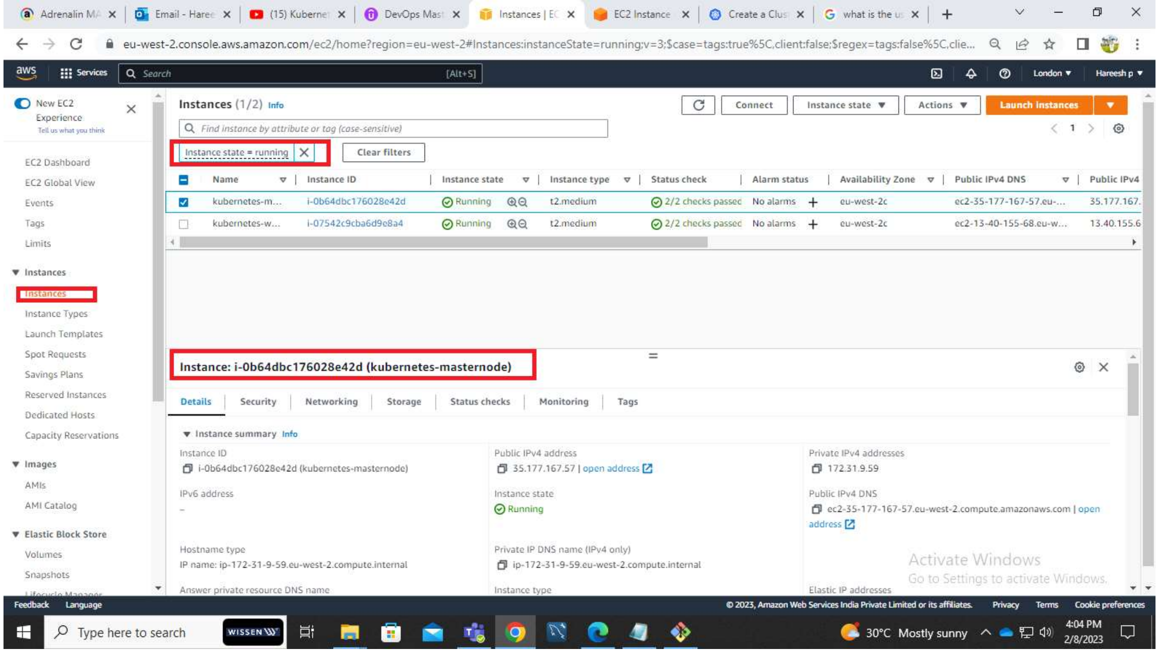
Setup Kubernetes master and node setup and able to establish the communication between master and node :

- login to aws console
- Navigate ec2 dashboard and launch the -2 instances



eu-west-2.console.aws.amazon.com/ec2/home?region=eu-west-2#Instances:instanceState=running;v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,clie...

Instances (1/2)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
☑ kubernetes-m...	i-0b64dbc176028e42d	Running	t2.medium	2/2 checks passed	No alarms	eu-west-2c	ec2-35-177-167-57.eu-...	35.177.167.
□ kubernetes-w...	i-07542c9cba6d9e8a4	Running	t2.medium	2/2 checks passed	No alarms	eu-west-2c	ec2-13-40-155-68.eu-w...	13.40.155.6

Instance: i-0b64dbc176028e42d (kubernetes-masternode)

Instance summary

Instance ID: i-0b64dbc176028e42d (kubernetes-masternode)

Public IPv4 address: 35.177.167.57 | open address

Private IPv4 addresses: 172.31.9.59

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-35-177-167-57.eu-west-2.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-9-59.eu-west-2.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-9-59.eu-west-2.compute.internal

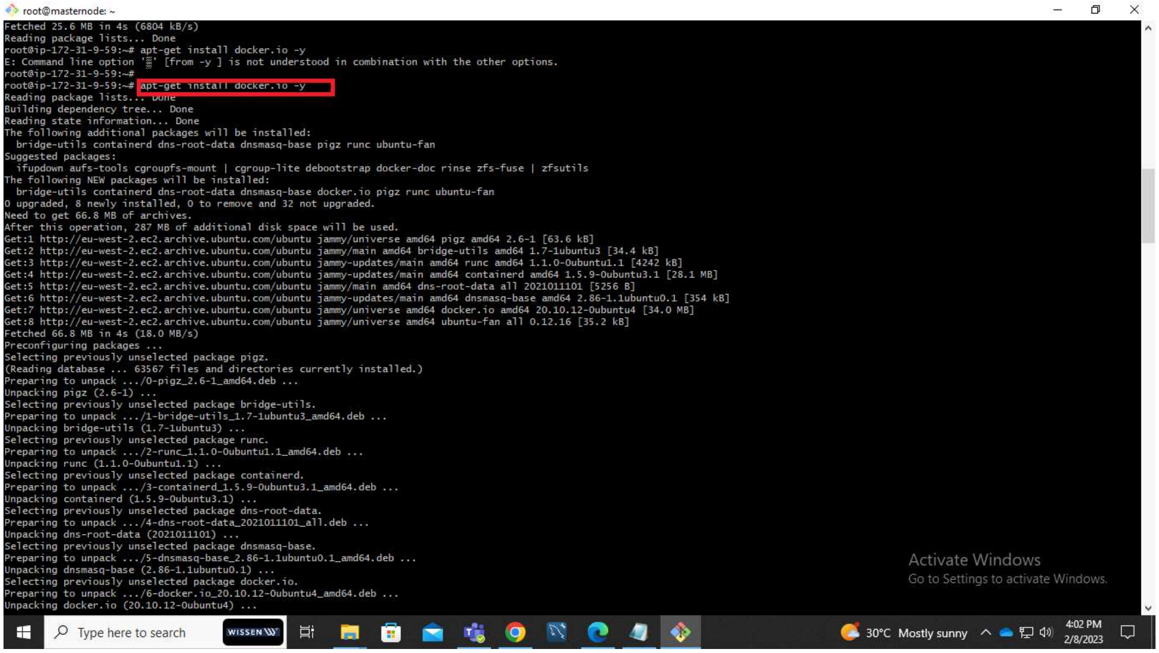
Answer private resource DNS name

Instance type

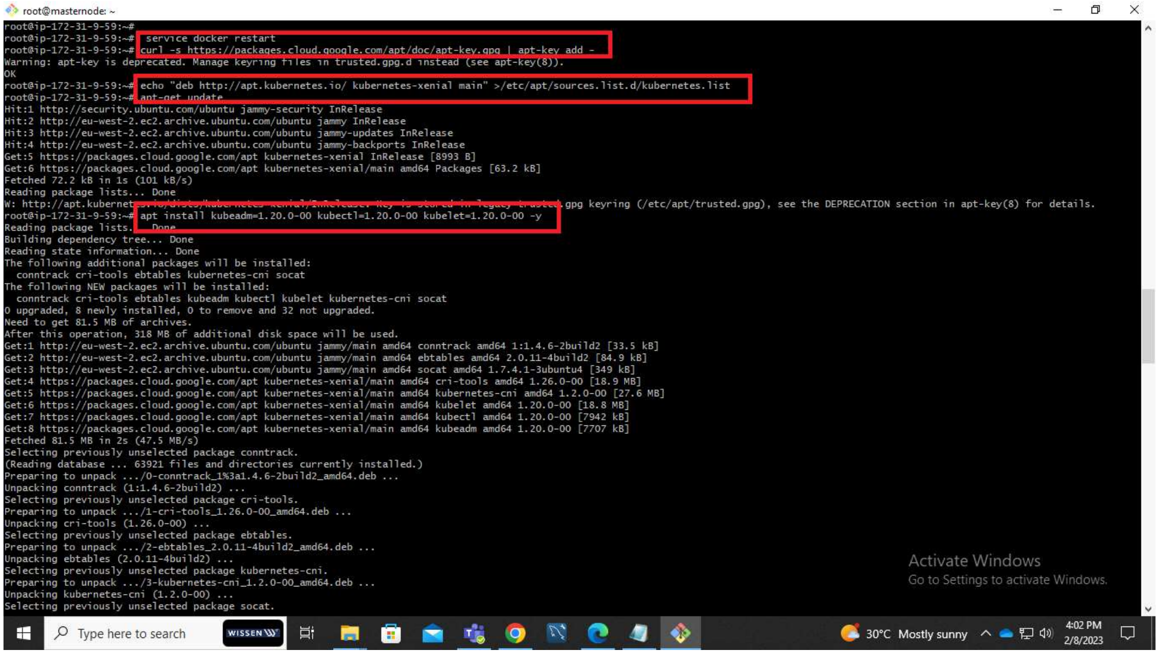
Elastic IP addresses

Instance-1 - (master) Apply the below commands:

- sudo su - (switch to root user).
- apt-get update (To keep all the packages up to date).
- apt-get install docker.io -y (Install docker)
- service docker restart (Start Docurl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add - (To add the key to authenticate the software packages). cker daemon).



```
Fetched 25.6 MB in 4s (6804 kB/s)
Reading package lists... Done
root@ip-172-31-9-59:~# apt-get install docker.io -y
E: Command line option '▒' [from -y ] is not understood in combination with the other options.
root@ip-172-31-9-59:~#
root@ip-172-31-9-59:~# apt-get install docker.io -y
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-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 32 not upgraded.
Need to get 66.8 MB of archives.
After this operation, 287 MB of additional disk space will be used.
Get:1 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.0-0ubuntu1.1 [4242 kB]
Get:4 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.5.9-0ubuntu3.1 [28.1 MB]
Get:5 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]
Get:6 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.86-1.1ubuntu0.1 [354 kB]
Get:7 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 docker.io amd64 20.10.12-0ubuntu4 [34.0 MB]
Get:8 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 66.8 MB in 4s (18.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 63567 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.0-0ubuntu1.1_amd64.deb ...
Unpacking runc (1.1.0-0ubuntu1.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.9-0ubuntu3.1_amd64.deb ...
Unpacking containerd (1.5.9-0ubuntu3.1) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../4-dns-root-data_2021011101_all.deb ...
Unpacking dns-root-data (2021011101) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.86-1.1ubuntu0.1_amd64.deb ...
Unpacking dnsmasq-base (2.86-1.1ubuntu0.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_20.10.12-0ubuntu4_amd64.deb ...
Unpacking docker.io (20.10.12-0ubuntu4) ...
```

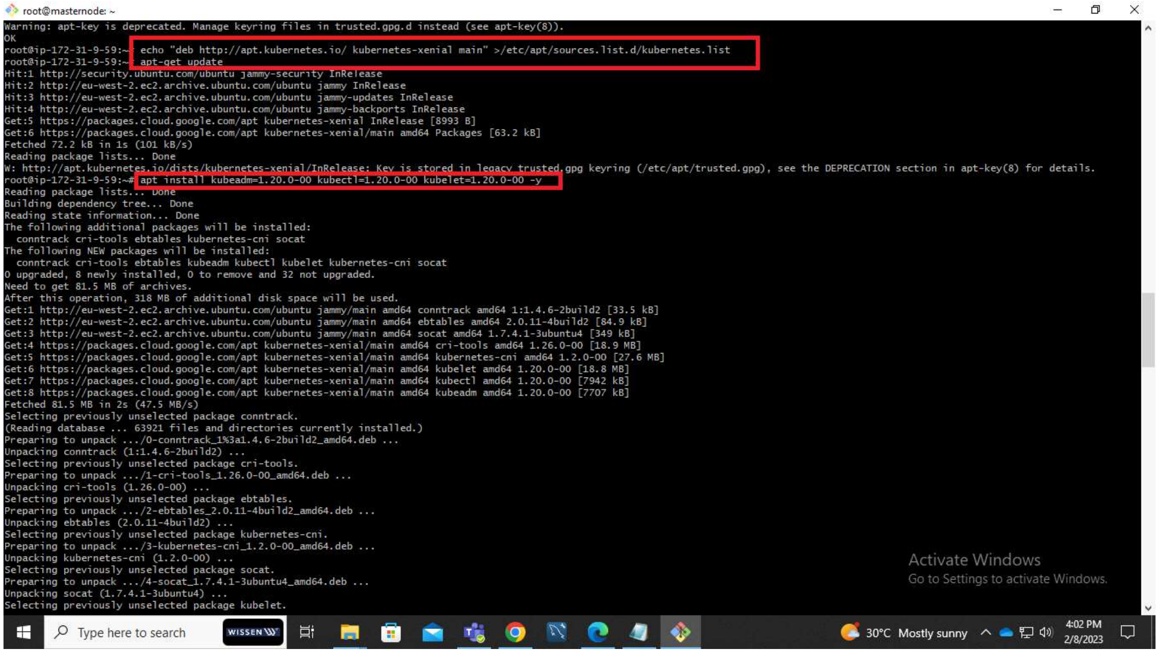


```
root@ip-172-31-9-59:~#
root@ip-172-31-9-59:~#  service docker restart
root@ip-172-31-9-59:~# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@ip-172-31-9-59:~# echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-9-59:~# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [63.2 kB]
Fetched 72.2 kB in 1s (101 kB/s)
Reading package lists... Done
W: http://apt.kubernetes.io/dists/kubernetes-xenial/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@ip-172-31-9-59:~# apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 32 not upgraded.
Need to get 81.5 MB of archives.
After this operation, 318 MB of additional disk space will be used.
Get:1 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ebtables amd64 2.0.11-4build2 [84.9 kB]
Get:3 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 socat amd64 1.7.4.1-3ubuntu4 [349 kB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.26.0-00 [18.9 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.2.0-00 [27.6 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.20.0-00 [18.8 MB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.20.0-00 [7942 kB]
Get:8 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.20.0-00 [7707 kB]
Fetched 81.5 MB in 2s (47.5 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 63921 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.6-2build2_amd64.deb ...
Unpacking conntrack (1:1.4.6-2build2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.26.0-00_amd64.deb ...
Unpacking cri-tools (1.26.0-00) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-4build2_amd64.deb ...
Unpacking ebtables (2.0.11-4build2) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_1.2.0-00_amd64.deb ...
Unpacking kubernetes-cni (1.2.0-00) ...
Selecting previously unselected package socat.
```

- echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list (Add the Kubernetes apt reposito

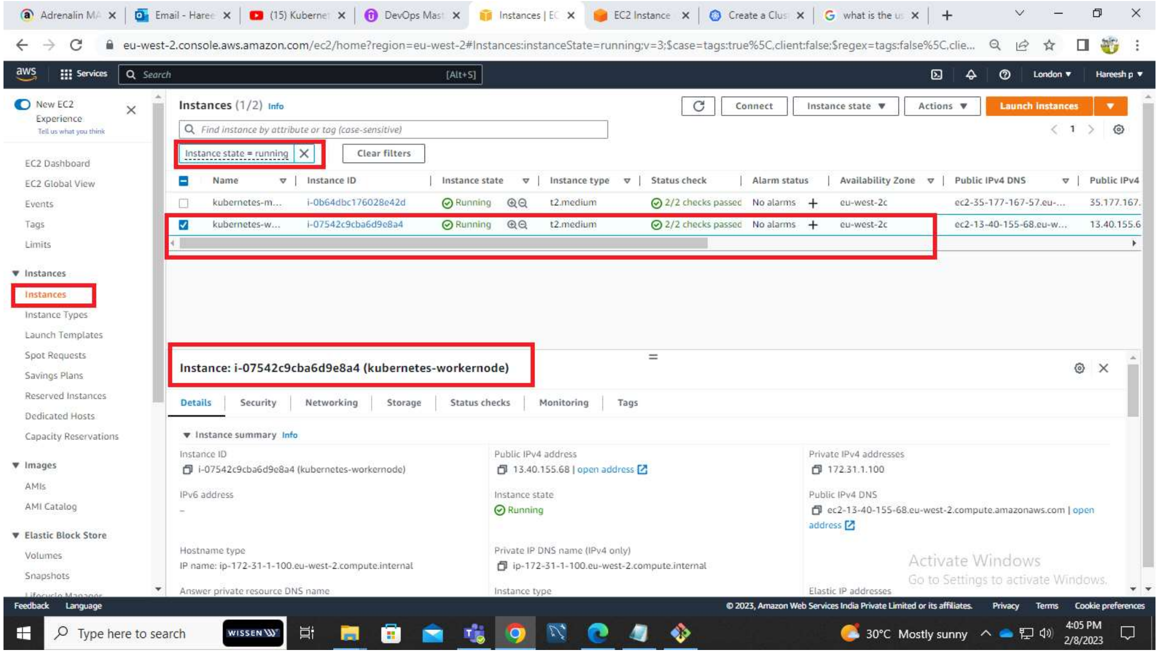
- apt-get update

- apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y (Install kubeadm,kubectl kubelet).



```
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@ip-172-31-9-59:~# echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-9-59:~# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [63.2 kB]
Fetched 72.2 kB in 1s (101 kB/s)
Reading package lists... Done
W: http://apt.kubernetes.io/dists/kubernetes-xenial/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
root@ip-172-31-9-59:~# apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 32 not upgraded.
Need to get 81.5 MB of archives.
After this operation, 318 MB of additional disk space will be used.
Get:1 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:2 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ebtables amd64 2.0.11-4build2 [84.9 kB]
Get:3 http://eu-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 socat amd64 1.7.4.1-3ubuntu4 [349 kB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.26.0-00 [18.9 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.2.0-00 [27.6 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.20.0-00 [18.8 MB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.20.0-00 [7942 kB]
Get:8 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.20.0-00 [7707 kB]
Fetched 81.5 MB in 2s (47.5 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 63921 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.6-2build2_amd64.deb ...
Unpacking conntrack (1:1.4.6-2build2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.26.0-00_amd64.deb ...
Unpacking cri-tools (1.26.0-00) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-4build2_amd64.deb ...
Unpacking ebtables (2.0.11-4build2) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_1.2.0-00_amd64.deb ...
Unpacking kubernetes-cni (1.2.0-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.4.1-3ubuntu4_amd64.deb ...
Unpacking socat (1.7.4.1-3ubuntu4) ...
Selecting previously unselected package kubelet.
```

Instance-2 - (Workingnode)



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
☐	kubernetes-m...	i-0b64dbc176028e42d	Running	t2.medium	2/2 checks passed	No alarms	eu-west-2c	ec2-35-177-167-57.eu-...	35.177.167.
☑	kubernetes-w...	i-07542c9cba6d9e8a4	Running	t2.medium	2/2 checks passed	No alarms	eu-west-2c	ec2-13-40-155-68.eu-w...	13.40.155.6

Instance: i-07542c9cba6d9e8a4 (kubernetes-workernode)

Instance summary Info

Instance ID
i-07542c9cba6d9e8a4 (kubernetes-workernode)

Public IPv4 address
13.40.155.68 | open address

Private IPv4 addresses
172.31.1.100

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-13-40-155-68.eu-west-2.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-1-100.eu-west-2.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-1-100.eu-west-2.compute.internal

Answer private resource DNS name

Instance type

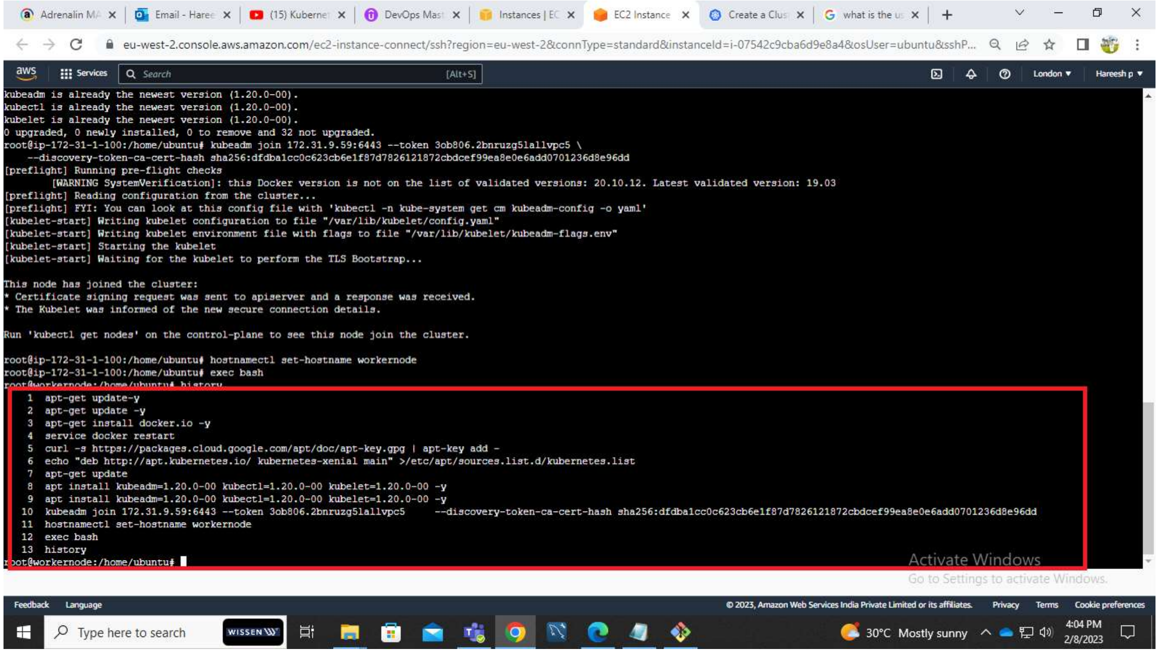
Elastic IP addresses

- Apply the below commands for working node also

- sudo su - (switch to root user).

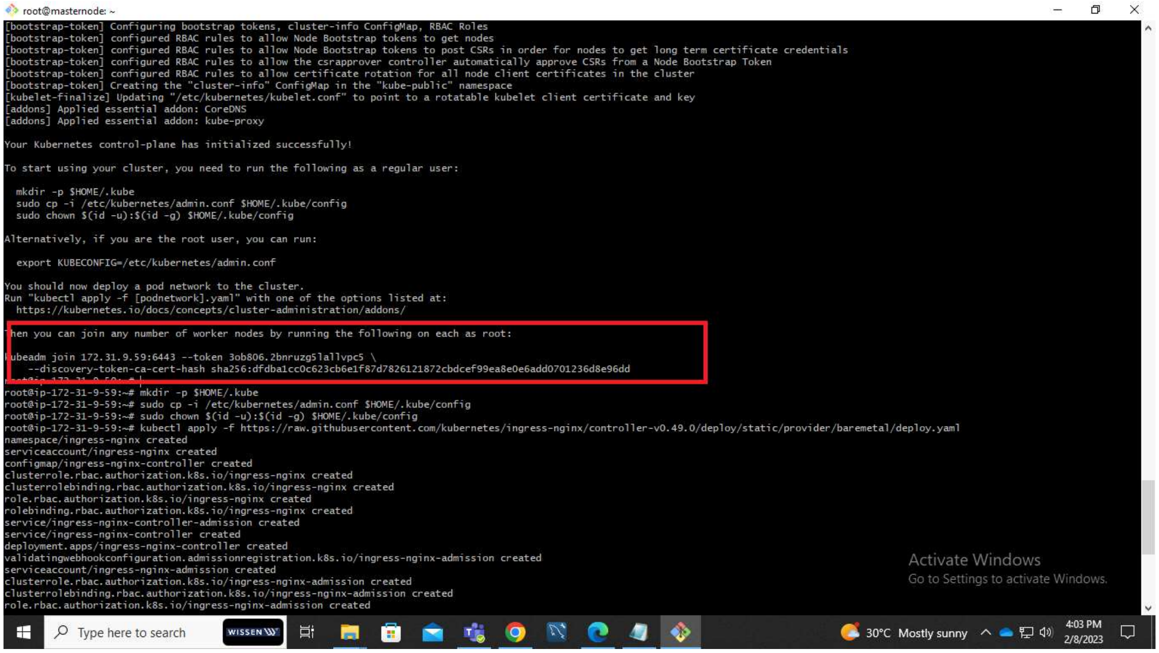
- * apt-get update (To keep all the packages up to date).

- * apt-get install docker.io -y (Install docker)
- * service docker restart (Start Docker daemon).
- * curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add - (To add the key to authenticate the software packages).
- * echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list (Add the Kubernetes apt repository)
- * apt-get update
- * apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y (Install kubeadm,kubectl kubelet).


Adrenalin MA
Email - Haree
(15) Kuberne
DevOps Mast
Instances | EC
EC2 Instance
Create a Clus
what is the u
eu-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?region=eu-west-2&connType=standard&instanceId=i-07542c9cba6d9e8a4&osUser=ubuntu&sshP...
aws
Services
Search
[Alt+S]
London ▼
Hareesh p ▼

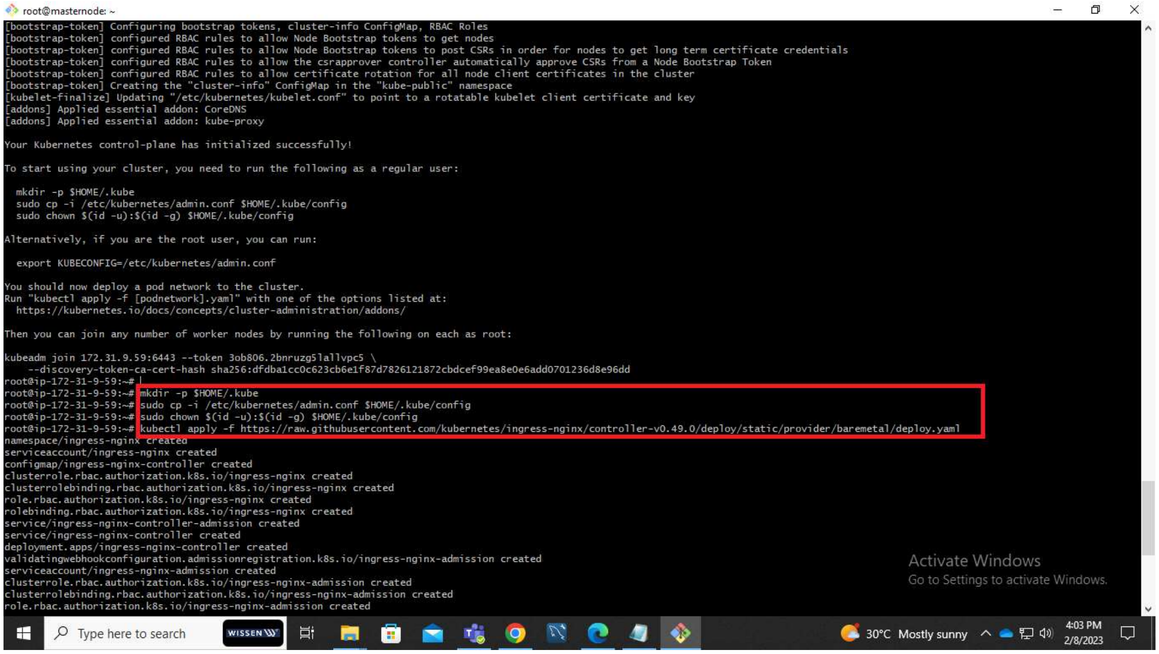
```
kubeadm is already the newest version (1.20.0-00).
kubectl is already the newest version (1.20.0-00).
kubelet is already the newest version (1.20.0-00).
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
root@ip-172-31-1-100:/home/ubuntu# kubeadm join 172.31.9.59:6443 --token 3ob806.2bnruzg5lallvpc5 \
    --discovery-token-ca-cert-hash sha256:dfdba1cc0c623cb6e1f87d7826121872cbdcef99ea8e0e6add0701236d8e96dd
[preflight] Running pre-flight checks
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 20.10.12. Latest validated version: 19.03
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-1-100:/home/ubuntu# hostnamectl set-hostname workernode
root@ip-172-31-1-100:/home/ubuntu# exec bash
root@workernode:/home/ubuntu# history
    1  apt-get update-y
    2  apt-get update -y
    3  apt-get install docker.io -y
    4  service docker restart
    5  curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
    6  echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list
    7  apt-get update
    8  apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y
    9  apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y
   10  kubeadm join 172.31.9.59:6443 --token 3ob806.2bnruzg5lallvpc5     --discovery-token-ca-cert-hash sha256:dfdba1cc0c623cb6e1f87d7826121872cbdcef99ea8e0e6add0701236d8e96dd
   11  hostnamectl set-hostname workernode
   12  exec bash
   13  history
root@workernode:/home/ubuntu# 
```

- Apply the master token in worker node to establishing communication



root@masternode: ~

```
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[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 172.31.9.59:6443 --token 3ob806.2bnruzg5lallvpc5 \
    --discovery-token-ca-cert-hash sha256:dfdba1cc0c623cb6e1f87d7826121872cbdcef99ea8e0e6add0701236d8e96dd
root@ip-172-31-9-59:~#
root@ip-172-31-9-59:~# mkdir -p $HOME/.kube
root@ip-172-31-9-59:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-9-59:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-9-59:~# kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
configmap/ingress-nginx-controller created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
service/ingress-nginx-controller-admission created
service/ingress-nginx-controller created
deployment.apps/ingress-nginx-controller created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
serviceaccount/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
```



root@masternode: ~

```
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[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 172.31.9.59:6443 --token 3ob806.2bnruzg5lallvpc5 \
    --discovery-token-ca-cert-hash sha256:dfdba1cc0c623cb6e1f87d7826121872cbdcef99ea8e0e6add0701236d8e96dd
root@ip-172-31-9-59:~#
root@ip-172-31-9-59:~# mkdir -p $HOME/.kube
root@ip-172-31-9-59:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-9-59:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-9-59:~# kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
configmap/ingress-nginx-controller created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
service/ingress-nginx-controller-admission created
service/ingress-nginx-controller created
deployment.apps/ingress-nginx-controller created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
serviceaccount/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
```

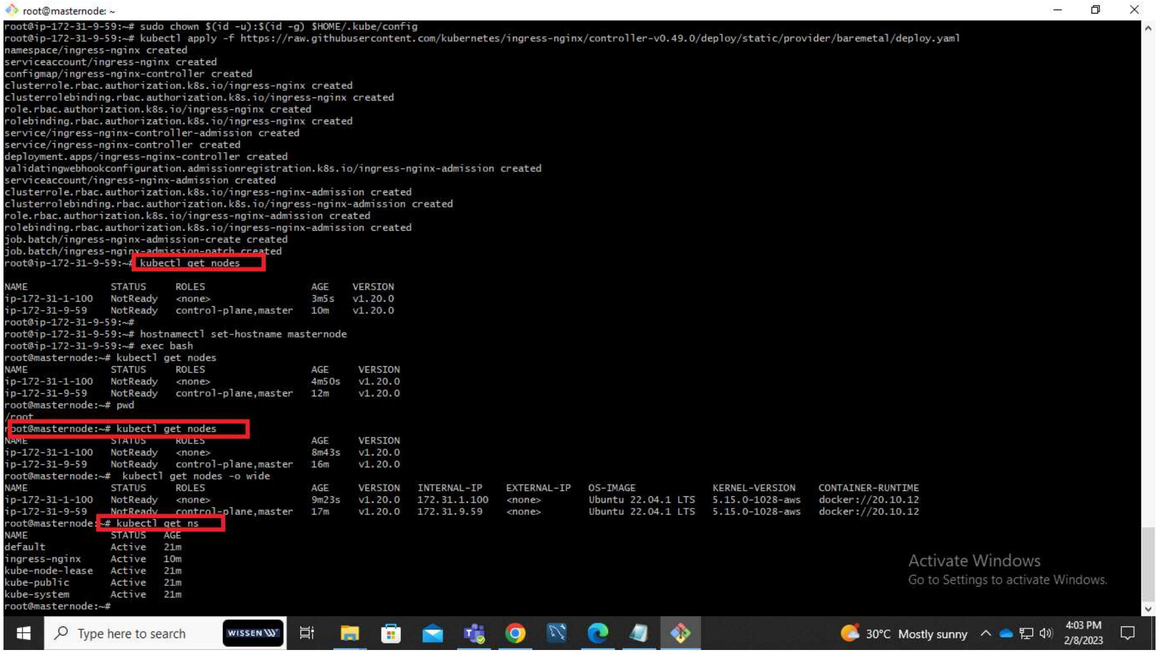
- In master

- apply - kubeadm init --pod-network-cidr=192.168.0.0/16 (Initialize kubadm and give the cidr).

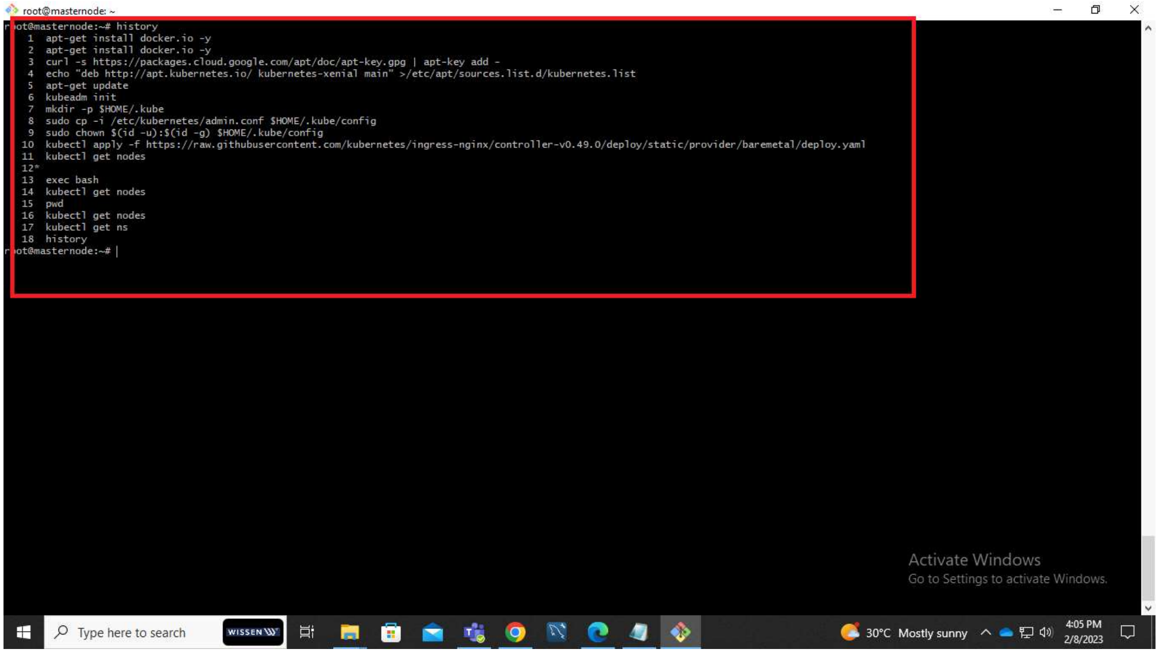
- mkdir -p $HOME/.kube (To Create the .kube subdirectory in your home directory).

- * sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config (To Create a copy of the Kubernetes admin.conf file in the .kube directory).

- * sudo chown $(id -u):$(id -g) $HOME/.kube/config (To change the permissions for the file).
- kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
- * kubectl get nodes
- * kubectl get nodes -o wide
- * kubectl get ns



```
root@ip-172-31-9-59:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-9-59:~# kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
configmap/ingress-nginx-controller created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
service/ingress-nginx-controller-admission created
service/ingress-nginx-controller created
deployment.apps/ingress-nginx-controller created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
serviceaccount/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
job.batch/ingress-nginx-admission-create created
job.batch/ingress-nginx-admission-patch created
root@ip-172-31-9-59:~# kubectl get nodes

NAME               STATUS     ROLES                  AGE    VERSION
ip-172-31-1-100    NotReady   <none>                 3m5s   v1.20.0
ip-172-31-9-59     NotReady   control-plane,master   10m    v1.20.0
root@ip-172-31-9-59:~#
root@ip-172-31-9-59:~# hostnamectl set-hostname masternode
root@ip-172-31-9-59:~# exec bash
root@masternode:~# kubectl get nodes
NAME               STATUS     ROLES                  AGE     VERSION
ip-172-31-1-100    NotReady   <none>                 4m50s   v1.20.0
ip-172-31-9-59     NotReady   control-plane,master   12m     v1.20.0
root@masternode:~# pwd
/root
root@masternode:~# kubectl get nodes
NAME               STATUS     ROLES                  AGE     VERSION
ip-172-31-1-100    NotReady   <none>                 8m43s   v1.20.0
ip-172-31-9-59     NotReady   control-plane,master   16m     v1.20.0
root@masternode:~#  kubectl get nodes -o wide
NAME               STATUS     ROLES                  AGE     VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
ip-172-31-1-100    NotReady   <none>                 9m23s   v1.20.0   172.31.1.100   <none>        Ubuntu 22.04.1 LTS   5.15.0-1028-aws    docker://20.10.12
ip-172-31-9-59     NotReady   control-plane,master   17m     v1.20.0   172.31.9.59    <none>        Ubuntu 22.04.1 LTS   5.15.0-1028-aws    docker://20.10.12
root@masternode:~# kubectl get ns
NAME               STATUS   AGE
default            Active   21m
ingress-nginx      Active   10m
kube-node-lease    Active   21m
kube-public        Active   21m
kube-system        Active   21m
root@masternode:~#
```



```
root@masternode:~# history
    1  apt-get install docker.io -y
    2  apt-get install docker.io -y
    3  curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
    4  echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list
    5  apt-get update
    6  kubeadm init
    7  mkdir -p $HOME/.kube
    8  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
    9  sudo chown $(id -u):$(id -g) $HOME/.kube/config
   10  kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
   11  kubectl get nodes
   12*
   13  exec bash
   14  kubectl get nodes
   15  pwd
   16  kubectl get nodes
   17  kubectl get ns
   18  history
root@masternode:~# |
```