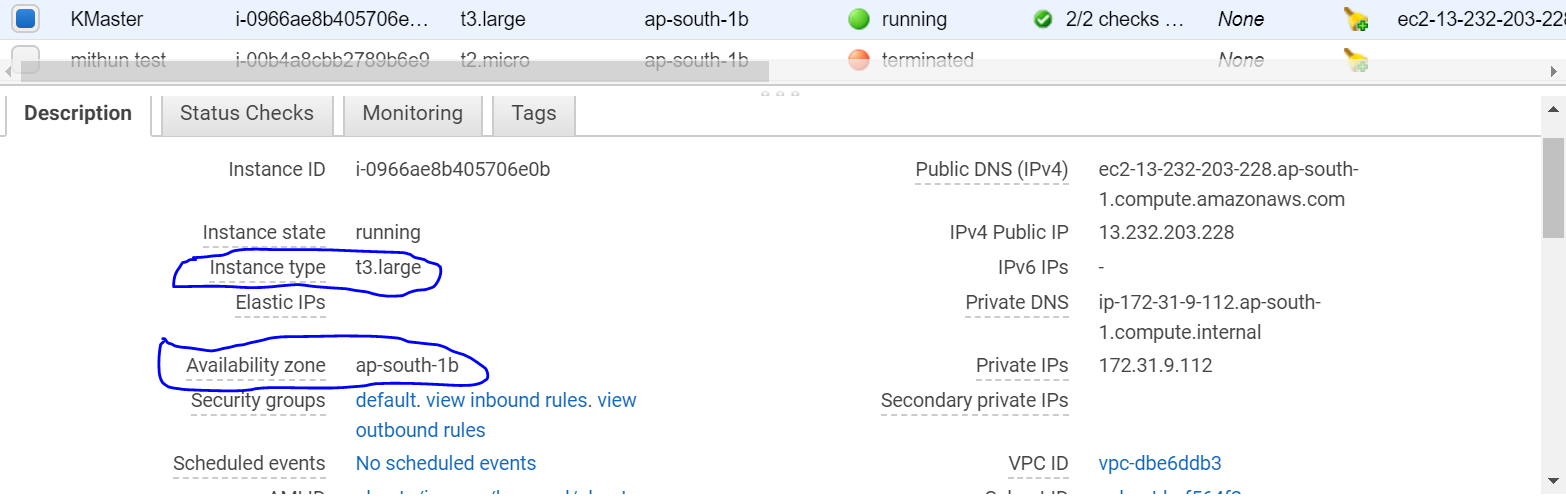
**Setting up Kubernetes Master Server**

Launch EC2 instance with Instance type ‘**t3.large’** & AZ should be **ap-soutj-1b** availability zone.



**Login as root.**

1 apt-get update

2 apt-get install docker.io -y || **To install Docker**

3 apt-get update

4 systemctl enable docker.service | **To enable docker services**

5 systemctl start docker.service | **To start doker services**

**Prerequisite for kube environment:**

6 apt-get update && apt-get install -y apt-transport-https -> To install https

7 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg|apt-key add -

8 cat <<EOF >/etc/apt/sources.list.d/kubernetes.list  
deb http://apt.kubernetes.io/ kubernetes-xenial main  
EOF

9 apt-get update

10 apt-get install -y kubeadm kubectl kubelet

**11.**

**java –version  TO check Java version**

**apt install default-jre  
apt install default-jre  
apt install openjdk-11-jre-headless  
apt install openjdk-8-jre-headless  
apt install openjdk-9-jre-headless**

12.apt-get update

13.kubeadm init | **Only to run if you want to make your server as a master (Only master node)**

**(It can take some time)**

**if it fails then:**

**cd /*etc/docker then create file daemon.json and add content***

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

***then* sudo systemctl daemon-reload**

**sudo systemctl restart docker**

**sudo systemctl restart kubelet**

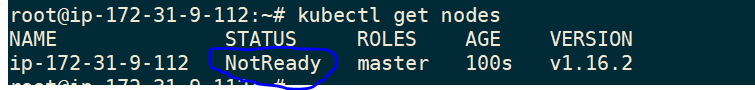
**now** run kubeadm reset(so that its deletes all the port no. currently used by kubeadm init) first to undo all of the changes from the first time you ran it.Then run systemctl restart kubelet

**Now again kubeadm init**

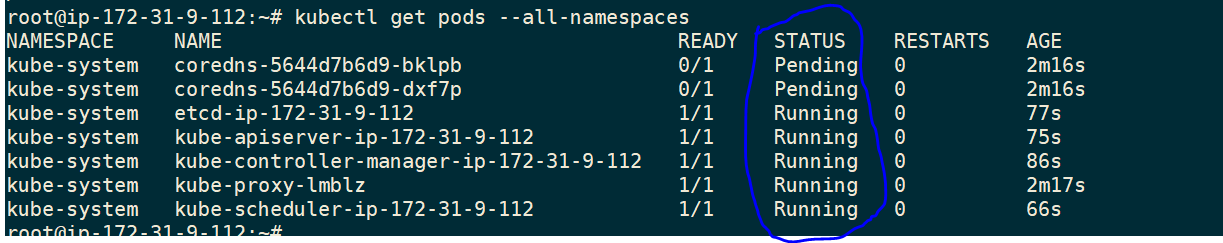
**14. To start using your cluster, you need to run the following as a regular user:**

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

15. **kubectl get nodes**



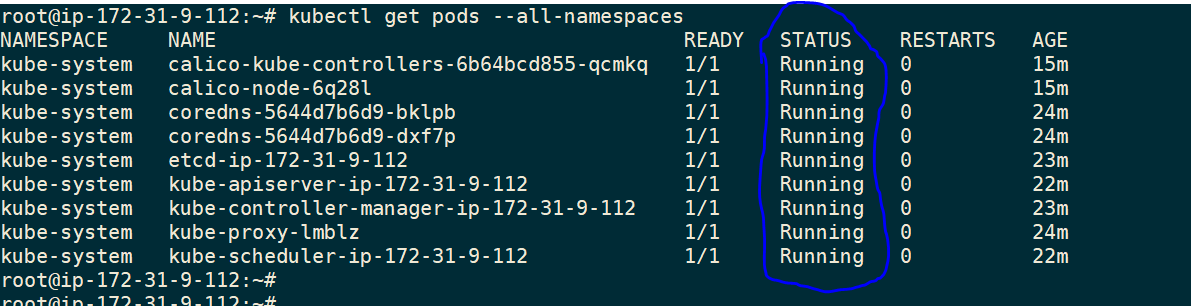
16.**kubectl get pods --all-namespaces** **|s All showing pending status because there is no pod network has been defined.**



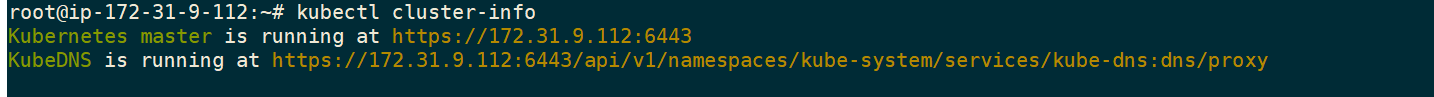
17.**kubectl apply -f https://docs.projectcalico.org/v3.10/manifests/calico.yaml**

18.**apt-get update**

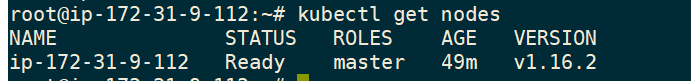
19.**kubectl get pods --all-namespaces**



20.**kubectl cluster-info**



21.**kubectl get nodes**

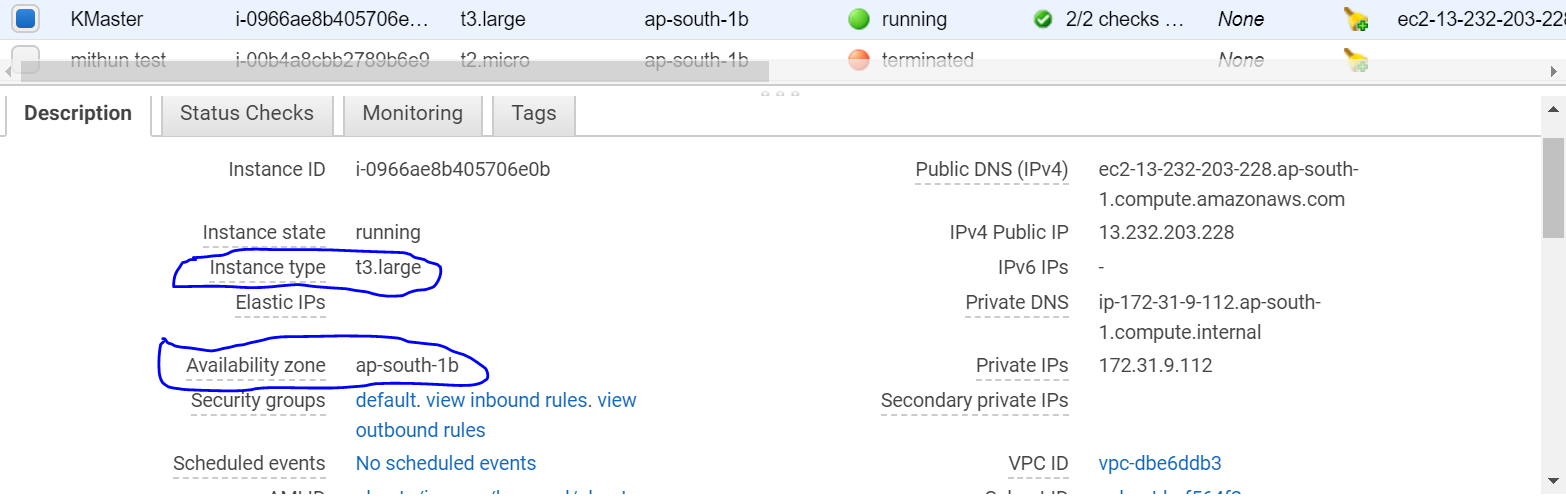


22.root@ip-172-31-9-112:~# **kubeadm token create --print-join-command**

kubeadm join 172.31.9.112:6443 --token jnhkbe.3835g4rixwiathz7 --discovery-token-ca-cert-hash sha256:2096c9cd3d82227a4da6139254df6e31dfd997ff0dfa503f6dba805b8e90579d

**Steps to setup Nodes:**

Launch EC2 instance with Instance type ‘**t3.large’** & AZ should be **ap-soutj-1b** availability zone.



**Login as root.**

1 apt-get update

2 apt-get install docker.io -y | **To install Docker**

3 apt-get update

4 systemctl enable docker.service | **To enable docker services**

5 systemctl start docker.service | **To start doker services**

**Prerequisite for kube environment:**

6 apt-get update && apt-get install -y apt-transport-https -> To install https

7 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg|apt-key add -

8 cat <<EOF >/etc/apt/sources.list.d/kubernetes.list  
deb http://apt.kubernetes.io/ kubernetes-xenial main  
EOF

9. apt-get update

10. apt-get install -y kubeadm kubectl kubelet

11 apt-get update

**12.**

**java –version  TO check Java version**

sudo apt install default-jre  
apt install default-jre  
apt install openjdk-11-jre-headless  
apt install openjdk-8-jre-headless  
apt install openjdk-9-jre-headless

13. apt-get update

**Run the token as copied from K8server:**

kubeadm join 172.31.9.112:6443 --token jnhkbe.3835g4rixwiathz7 --discovery-token-ca-cert-hash sha256:2096c9cd3d82227a4da6139254df6e31dfd997ff0dfa503f6dba805b8e90579d

**How to make ssh connection between two EC2 instances:**

**Note: Please make sure you have both ip entries in their /etc/hosts file.**

**On Server end:**

ubuntu@ip-172-31-33-61:~$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/ubuntu/.ssh/id\_rsa):

cat .ssh/id\_rsa.pub

**On Node:**

Go to authorized\_keys file and paste output of **id\_rsa.pub** which we have copied form servers **id\_rsa.pub.**

ubuntu@ip-172-31-13-243:~$ cat .ssh/authorized\_keys

<https://myopswork.com/how-to-install-kubernetes-k8-in-rhel-or-centos-in-just-7-steps-2b78331174a5>

<https://myopswork.com/how-to-install-kubernetes-k8-in-rhel-or-centos-in-just-7-steps-2b78331174a5>

[root@ip-172-31-42-23 ~]# history

1 pwd

2 clear

3 setenforce 0

4 sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

5 firewall-cmd --permanent --add-port=6443/tcp

6 echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables

7 cat <<EOF > /etc/yum.repos.d/centos.repo

[centos]

name=CentOS-7

baseurl=http://ftp.heanet.ie/pub/centos/7/os/x86\_64/

enabled=1

gpgcheck=1

gpgkey=http://ftp.heanet.ie/pub/centos/7/os/x86\_64/RPM-GPG-KEY-CentOS-7

#additional packages that may be useful

[extras]

name=CentOS-$releasever - Extras

baseurl=http://ftp.heanet.ie/pub/centos/7/extras/x86\_64/

enabled=1

gpgcheck=0

EOF

8 yum -y update

9 yum -y install docker

10 cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

11 setenforce 0

12 sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

13 sudo yum install -y yum-utils device-mapper-persistent-data lvm2

14 sudo yum-config-manager --enable rhel-7-server-extras-rpms

15 sudo -E yum-config-manager --add-repo "$DOCKERURL/rhel/docker-ee.repo"

16 sudo yum -y install docker-ee docker-ee-cli containerd.io

17 sudo yum install /path/to/package.rpm

18 sudo dnf repolist -v

19 dnf list docker-ce --showduplicates | sort -r

20 sudo dnf install https://download.docker.com/linux/centos/7/x86\_64/stable/Packages/containerd.io-1.2.6-3.3.el7.x86\_64.rpm

21 sudo dnf install docker-ce

22 sudo systemctl enable --now docker

23 sudo dnf install docker-ce

24 systemctl is-enabled docker

25 curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)" -o docker-compose

26 systemctl is-enabled docker

27 sudo dnf install docker-ce

28 top

29 $ sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

30 sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

31 sudo dnf repolist -v

32 dnf list docker-ce --showduplicates | sort -r

33 sudo dnf install docker-ce-3:18.09.1-3.el7

34 sudo dnf install --nobest docker-ce

35 sudo dnf install https://download.docker.com/linux/centos/7/x86\_64/stable/Packages/containerd.io-1.2.6-3.3.el7.x86\_64.rpm

36 sudo dnf install docker-ce

37 sudo systemctl disable firewalld

38 sudo systemctl enable --now docker

39 systemctl is-active docker

40 systemctl is-enabled docker

41 curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)" -o docker-compose

42 sudo mv docker-compose /usr/local/bin && sudo chmod +x /usr/local/bin/docker-compose

43 sudo dnf install python3-pip

44 pip3.6 install docker-compose --user

45 yum -y update

46 systemctl status docker

47 cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

48 setenforce 0

49 vi /etc/selinux/config

50 yum -y install kubelet kubeadm kubectl

51 systemctl start kubelet

52 systemctl enable kubelet

53 cat <<EOF > /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

EOF

54 sysctl --system

55 echo 1 > /proc/sys/net/ipv4/ip\_forward

56 kubeadm init --pod-network-cidr=10.244.0.0/16

57 kubectl get nodes

58 kubeadm init

ec2user@agv-master$ mkdir -p $HOME/.kube  
ec2user@agv-master$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/configec2user@agv-master$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

59 pwd

60 history

[root@ip-172-31-42-23 ~]#

On Linux:

Download the latest release with the command:

curl -LO https://github.com/kubernetes/kops/releases/download/**$(**curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4**)**/kops-linux-amd64

To download a specific version, replace the

**$(**curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4**)**

portion of the command with the specific version.

For example, to download kops version v1.15.0 type:

curl -LO https://github.com/kubernetes/kops/releases/download/1.15.0/kops-linux-amd64

Make the kops binary executable

chmod +x kops-linux-amd64

Move the kops binary in to your PATH.

sudo mv kops-linux-amd64 /usr/local/bin/kops

6 apt-get update && apt-get install -y apt-transport-https -> To install https

7 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg|apt-key add -

8 cat <<EOF >/etc/apt/sources.list.d/kubernetes.list  
deb http://apt.kubernetes.io/ kubernetes-xenial main  
EOF

9 apt-get update

10 apt-get install -y kubeadm kubectl kubelet