Kubernetes Setup Using Kubeadm In AWS EC2 Ubuntu Servers

=======================================================

Prerequisite:

==========

3 - Ubuntu Serves

1 - Manager (4GB RAM , 2 Core) t2.medium

2 - Workers (1 GB, 1 Core) t2.micro

**Note:**Open Required Ports In AWS Security Groups. For now we will open All trafic.

==========COMMON FOR MASTER & SLAVES START ====  
  
# First, login as ‘root’ user because the following set of commands need to be executed with ‘sudo’ permissions.  
  
sudo su -  
  
# Install Required packages and apt keys.

apt-get update –y

apt-get update && apt-get install -y apt-transport-https curl

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

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

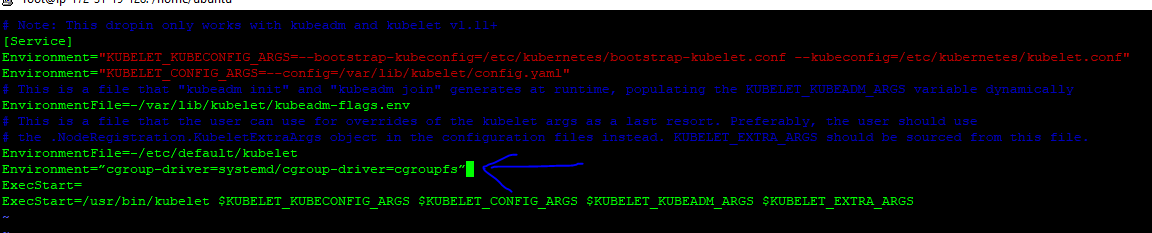
apt-get update –y

#Install kubeadm, Kubelet And Kubectl  
  
apt-get install -y kubelet kubeadm kubectl

vi /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

Then add the marked line above ExecStart=

Environment=”cgroup-driver=systemd/cgroup-driver=cgroupfs”



# Install And Enable Docker  
  
apt install docker.io –y  
systemctl status docker

Go to /etc/docker

Create daemon.json and add the below config.

{

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

}

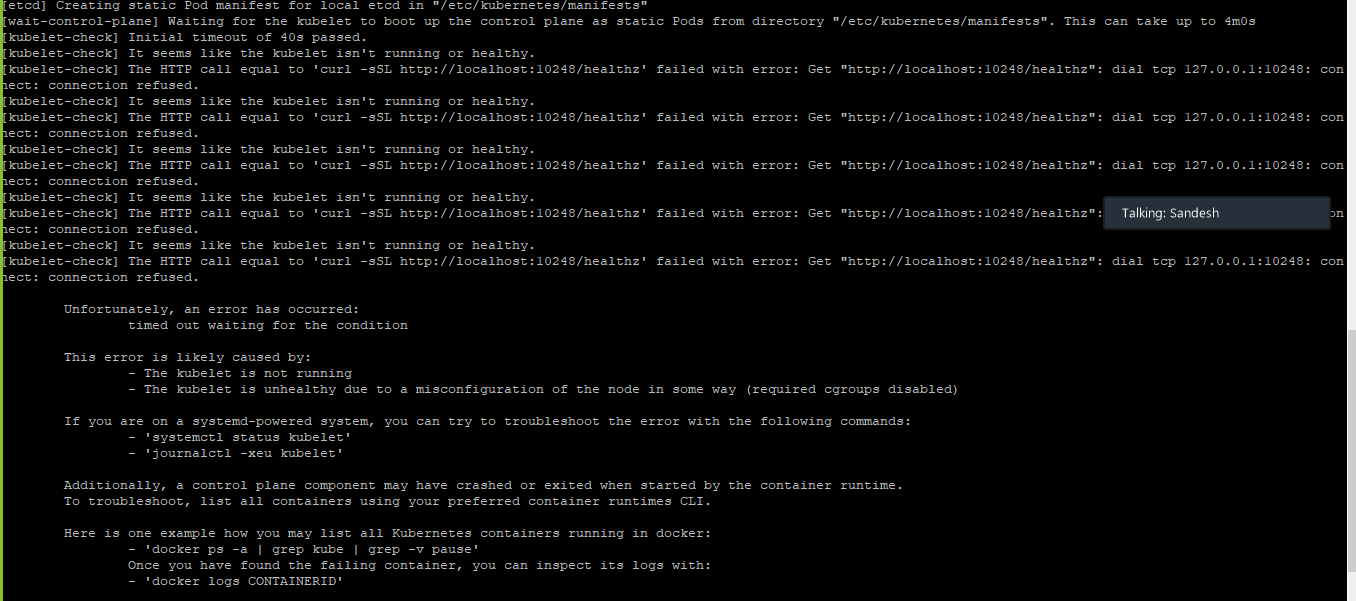
systemctl daemon-reloadsystemctl restart dockersystemctl restart kubelet

===========In Master Node Start====================

Syntax is : # kubeadm init --apiserver-advertise-address=<ip-address-of-kmaster-vm> --pod-network-cidr=x.x.0.0/16

kubeadm init --apiserver-advertise-address=172.31.19.128 --pod-network-cidr=172.31.0.0/16

**If you got the below error.**

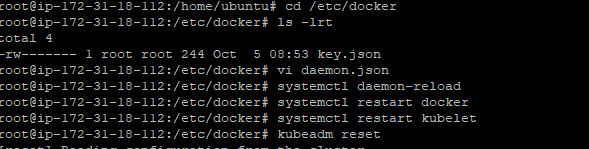


Go to /etc/docker

Create daemon.json and add the below config.

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

systemctl daemon-reloadsystemctl restart dockersystemctl restart kubelet



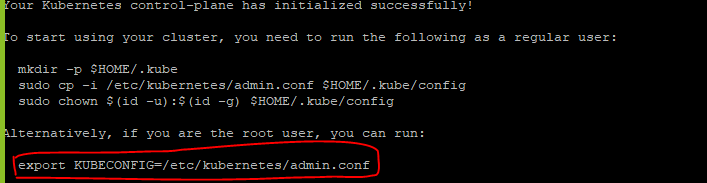
#reset kubeadm and create kubeadm init again

kubeadm reset

Syntax is : # kubeadm init --apiserver-advertise-address=<ip-address-of-kmaster-vm> --pod-network-cidr=x.x.0.0/16

kubeadm init --apiserver-advertise-address=172.31.19.128 --pod-network-cidr=172.31.0.0/16

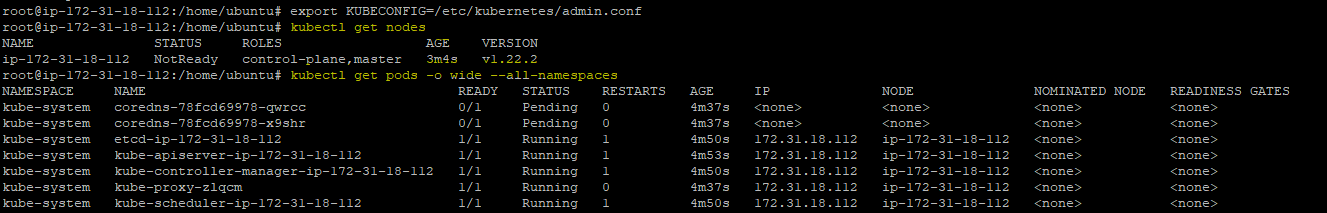
Execute the export command in root



# To verify, if kubectl is working or not, run the following command.  
  
kubectl get pods -o wide --all-namespaces

//verify master is ready or not

Kubectl get nodes



#You will notice from the previous command, that all the pods are running except one: ‘kube-dns’. For resolving this we will install a # pod network. To install the weave pod network, run the following command:

You should now deploy a pod network to the cluster.

Syntax: Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

Doc: <https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Commands:

**kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')**

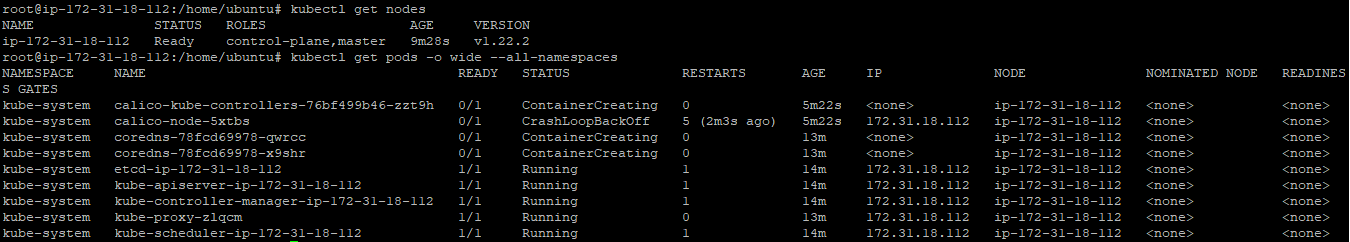
or for Calico pod network.

**kubectl apply -f** [**https://docs.projectcalico.org/v3.9/manifests/calico.yaml**](https://docs.projectcalico.org/v3.9/manifests/calico.yaml)

kubectl get pods -o wide --all-namespaces

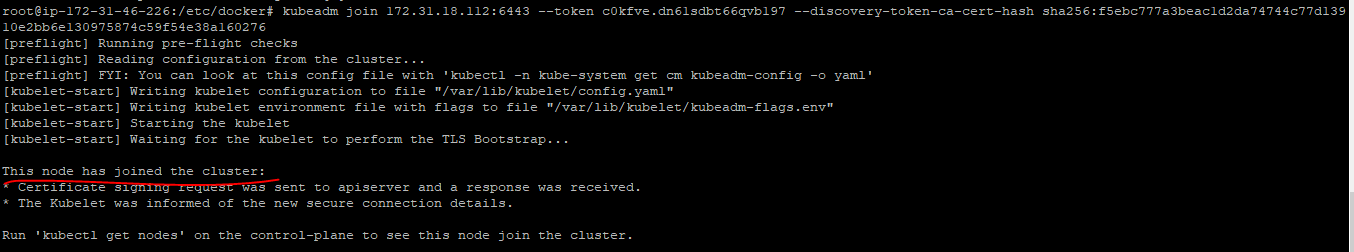
//verify master is ready

Kubectl get nodes

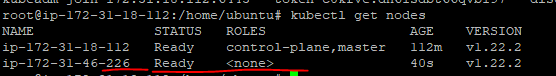
****

# Get token  
  
kubeadm token create --print-join-command

=========In Master Node End====================  
  
  
Add Worker Machines to Kubernates Master  
=========================================  
  
Copy kubeadm join token from and execute in Worker Nodes to join to cluste**r**

****

**Run kubectl get nodes to check worker node is joined and ready state**

****