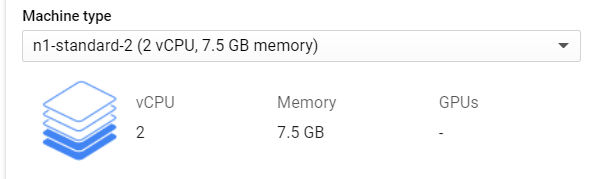
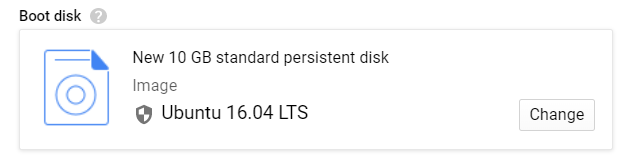
Setup Kubernetes Cluster Using Kubeadm on Ubuntu VM

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

1. Create Ubuntu VM on GCP, with name like k8s-master

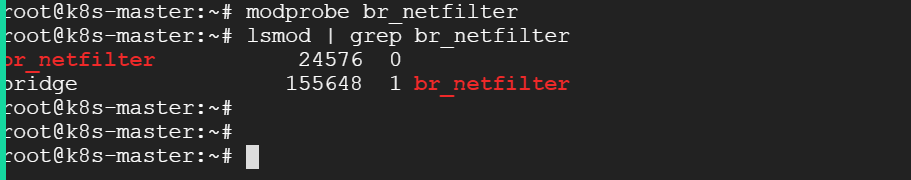




1. Create another Ubuntu VM on GCP with name like k8s-worker
2. Turn root and run below commands

modeporbe br\_netfilter

lsmod | grep br\_netfilter



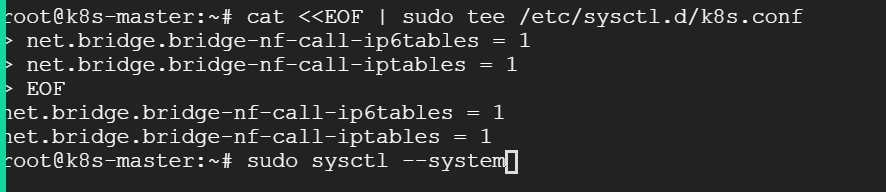
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

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

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

EOF

sudo sysctl --system



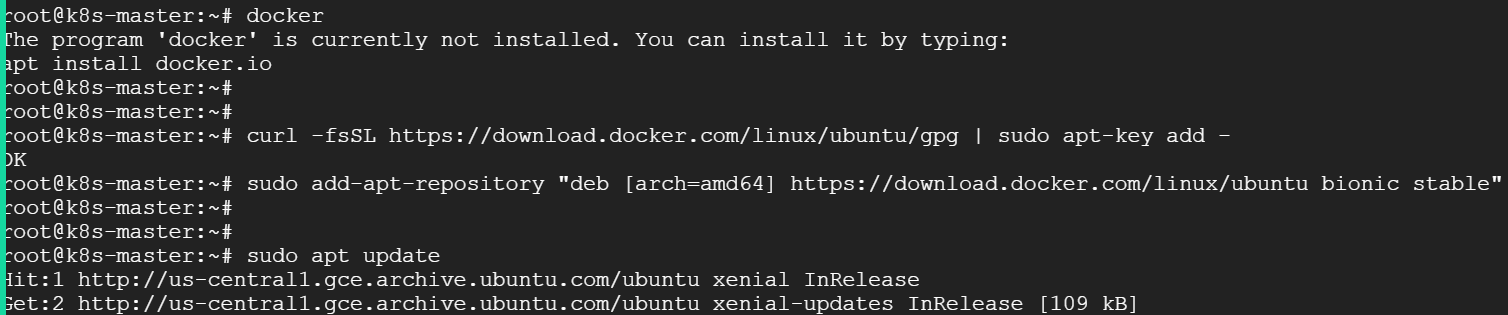
Install docker as your container run time

sudo sysctl --system

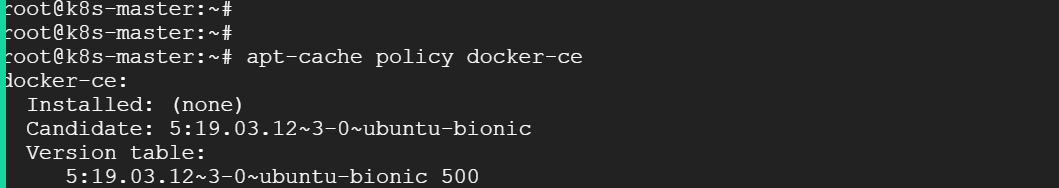
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

sudo apt update

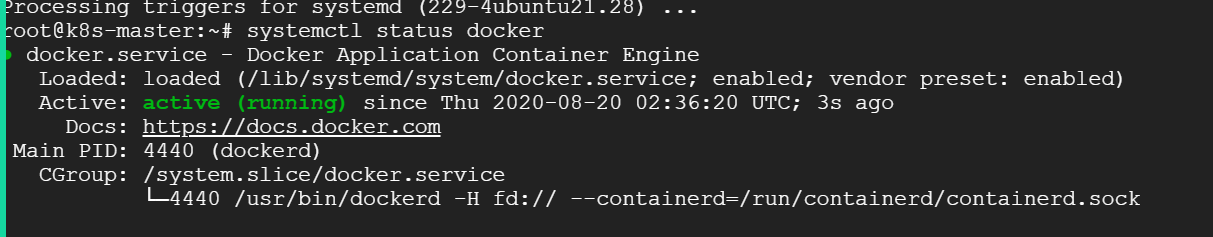


apt-cache policy docker-ce



sudo apt install docker-ce

sudo systemctl status docker



Install kubelet kubeadm and kubectl on Master node

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

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

apt-key add –

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

apt-get update

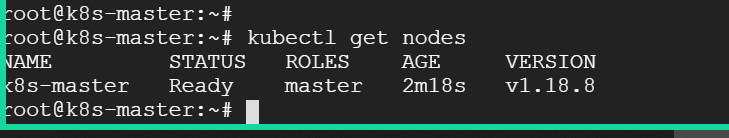
apt-get install -y kubelet kubeadm kubectl

apt-mark hold kubelet kubeadm kubectl

kubeadm init

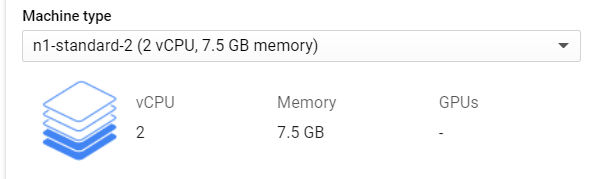


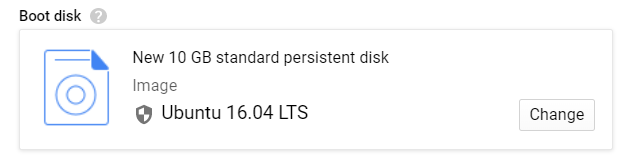
Kubectl get nodes will give you state of your master node



Setup the worker Node for your cluster

1. Create Ubuntu VM on GCP, with name like k8s-worker

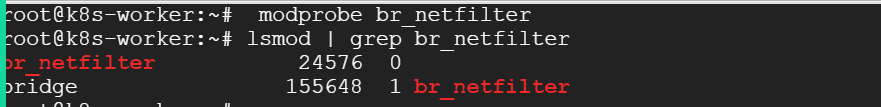


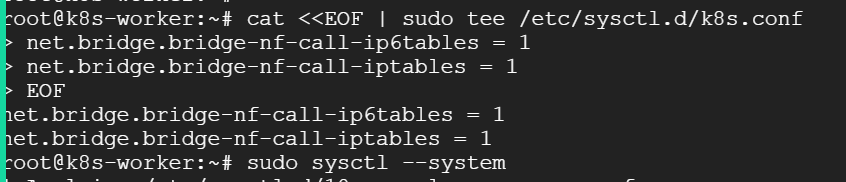


1. Create another Ubuntu VM on GCP with name like k8s-worker
2. Turn root and run below commands

modeporbe br\_netfilter

lsmod | grep br\_netfilter







apt-cache policy docker-ce

apt install docker-ce

systemctl status docker

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

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

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

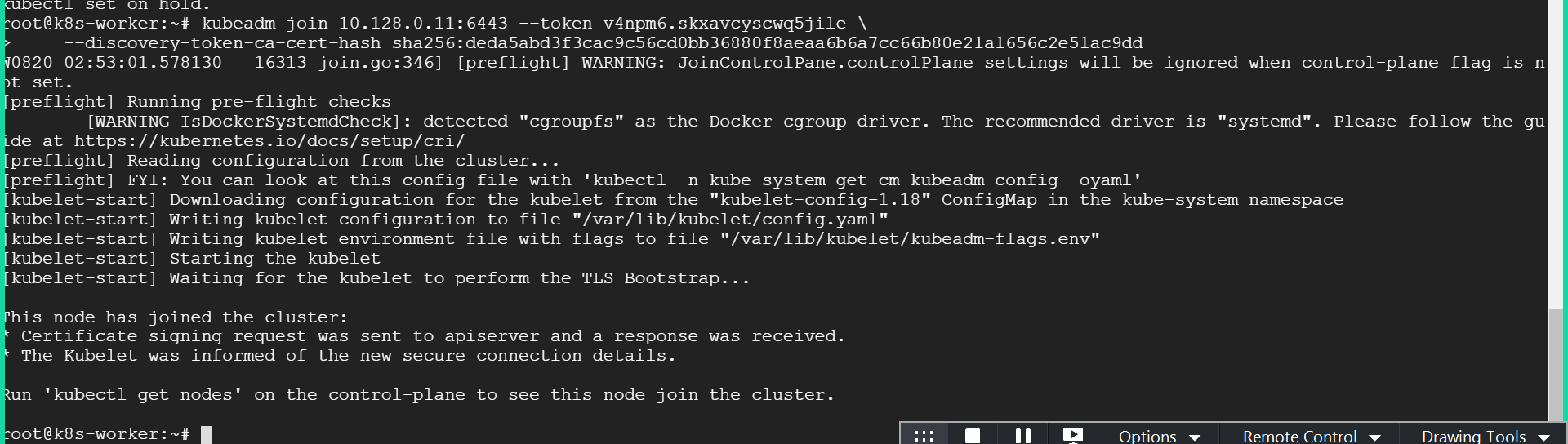
apt-get update

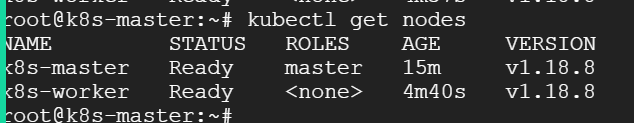
apt-get install -y kubelet kubeadm kubectl

apt-mark hold kubelet kubeadm kubectl

kubeadm join 10.128.0.11:6443 --token v4npm6.skxavcyscwq5jile \

--discovery-token-ca-cert-hash sha256:deda5abd3f3cac9c56cd0bb36880f8aeaa6b6a7cc66b80e21a1656c2e51ac9dd





Adding one More Node to Existing Cluster

Bootstap a new worker node in GCP Consle and install below

modprobe br\_netfilter

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

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

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

EOF

sudo sysctl --system

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

sudo apt update

apt-cache policy docker-ce

sudo apt install docker-ce

sudo systemctl status docker

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

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

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

apt-get update

apt-get install -y kubelet kubeadm kubectl

apt-mark hold kubelet kubeadm kubectl

kubeadm join 10.128.0.11:6443 --token v4npm6.skxavcyscwq5jile --discovery-token-ca-cert-hash sha256:deda5abd3f3cac9c56cd0bb36880f8aeaa6b6a7cc66b80e21a1656c2e51ac9dd

How to generate the Token kubeadm token create --print-join-command

