----------------------------------------------------------------

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

*# sysctl params required by setup, params persist across reboots*

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

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

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

net.ipv4.ip\_forward = 1

EOF

*# Apply sysctl params without reboot*

sudo sysctl --system

lsmod | grep br\_netfilter

lsmod | grep overlay

THEN GO FOR INSTALL DOCKER AS RUNTIME.

**Installation Steps**

1. First Update Software Repositories:

$ sudo apt-get update –y

2. Uninstall Old Versions of Docker ***(Optional:****Only if docker was already installed on this host and you want to configure it again****):***

$ sudo apt-get remove docker docker-engine docker.io containerd runc

3. Update the apt package index and install packages to allow apt to use a repository over HTTPS:

$ sudo apt-get update

$ sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

4. Add Docker’s official GPG key:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

5. Use the following command to set up the stable repository. To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below:

$ echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archivekeyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

6. Install Docker Engine:

$ sudo apt-get update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

7. Start and Enable Docker:

$ sudo systemctl start docker

$ sudo systemctl enable docker

8. Check Docker status:

$ sudo systemctl status docker

9. Check the Docker version and Docker Compose version:

$ docker --version

## Installing kubeadm, kubelet and kubectl

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

1. sudo curl -fsSLo /etc/apt/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
2. echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
3. sudo apt-get update
4. sudo apt-get install -y kubelet kubeadm kubectl
5. sudo apt-mark hold kubelet kubeadm kubectl

----------------------------------------------------------------------------

Master:----

kubeadm init --pod-network-cidr 192.168.0.0/16

output:----

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

You should now deploy a Pod network to the cluster.

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

/docs/concepts/cluster-administration/addons/

You can now join any number of machines by running the following on each node

as root:

kubeadm join <control-plane-host>:<control-plane-port> --token <token> --discovery-token-ca-cert-hash sha256:<hash>

master node:-----

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

worker node:

kubeadm join <control-plane-host>:<control-plane-port> --token <token> --discovery-token-ca-cert-hash sha256:<hash>

master node:------install network plugin

$ **kubectl apply -f** [**https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml**](https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml)

**$ kubectl get node**