KMaster& KNode -🡪>Ubuntu 20.04

Perform all the commands as root user unless otherwise specified

**Disable Firewall**

ufw disable

**Disable swap**

swapoff -a; sed -i '/swap/d' /etc/fstab

**Update sysctl settings for Kubernetes networking**

cat >>/etc/sysctl.d/kubernetes.conf<<EOF

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

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

EOF

sysctl --system

**Install docker engine**

{

apt install -y apt-transport-https ca-certificates curl gnupg-agent software-properties-common

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

add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

apt update

apt install -y docker-ce=5:19.03.10~3-0~ubuntu-focal containerd.io

or

sudo apt-get install docker-ce docker-ce-cli containerd.io

}

**Kubernetes Setup**

**Add Apt repository**

{

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

echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" > /etc/apt/sources.list.d/kubernetes.list

or

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

}

**Install Kubernetes components**

apt update && apt install -y kubeadm=1.18.5-00 kubelet=1.18.5-00 kubectl=1.18.5-00

sudo apt update && apt-get install -y kubelet kubeadm kubectl

**In case you are using LXC containers for Kubernetes nodes**

Hack required to provision K8s v1.15+ in LXC containers

{

mknod /dev/kmsg c 1 11

echo '#!/bin/sh -e' >> /etc/rc.local

echo 'mknod /dev/kmsg c 1 11' >> /etc/rc.local

chmod +x /etc/rc.local

}

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

**On kmaster**

**Initialize Kubernetes Cluster**

Update the below command with the ip address of kmaster

1,kubeadm init --apiserver-advertise-address= 192.168.159.131 --pod-network-cidr=192.168.159.0/24 --ignore-preflight-errors=all

2,kubeadm init --apiserver-advertise-address=192.168.0.111 --pod-network-cidr=192.168.0.0/24 --ignore-preflight-errors=all

**Deploy Calico network**

3.kubectl --kubeconfig=/etc/kubernetes/admin.conf create -f <https://docs.projectcalico.org/v3.14/manifests/calico.yaml>

or

1.kubeadm init --pod-network-cidr=[10.244.0.0/16](http://10.244.0.0/16) --ignore-preflight-errors=all

2,kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

**Cluster join command**

kubeadm token create --print-join-command

**To be able to run kubectl commands as non-root user**

If you want to be able to run kubectl commands as non-root user, then as a non-root user perform these

mkdir -p $HOME/.kube

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

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

**On Kworker**

**Join the cluster**

Use the output from **kubeadm token create** command in previous step from the master server and run here.

**Verifying the cluster**

**Get Nodes status**

kubectl get nodes

**Get component status**

kubectl get cs