#### Prerequisites

* Minimal install Ubuntu 22.04
* Minimum 2GB RAM or more
* Minimum 2 CPU cores / or 2 vCPU
* 20 GB free disk space on /var or more

1) Set hostname on Each Node

sudo hostnamectl set-hostname "k8smaster.example.net"

exec bash

On the worker nodes, run

sudo hostnamectl set-hostname "k8sworker1.example.net"   // 1st worker node

sudo hostnamectl set-hostname "k8sworker2.example.net"   // 2nd worker node

++$ exec bash-

Add the following entries in /etc/hosts file on each node

192.168.1.173   k8smaster.example.net k8smaster

192.168.1.174   k8sworker1.example.net k8sworker1

192.168.1.175   k8sworker2.example.net k8sworker2

## 2) Disable swap & Add kernel Parameters

Execute beneath swapoff and [sed command](https://www.linuxtechi.com/20-sed-command-examples-linux-users/" \t "_blank" \o "20 Sed (Stream Editor) Command Examples for Linux Users) to disable swap. Make sure to run the following commands on all the nodes.

sudo swapoff -a

sudo sed -i '/ swap / s/^\(.\*\)$/#\1/g' /etc/fstab

Load the following kernel modules on all the nodes,

sudo tee /etc/modules-load.d/containerd.conf <<EOF

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

Set the following Kernel parameters for Kubernetes, run beneath [tee command](https://www.linuxtechi.com/tee-command-examples-in-linux/" \t "_blank" \o "9 tee Command Examples in Linux)

sudo tee /etc/sysctl.d/kubernetes.conf <<EOF

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

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

net.ipv4.ip\_forward = 1

EOF

Reload the above changes, run

sudo sysctl --system

## 3) Update system and install basic utilities for further installation

sudo apt-get update

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

### 4) Download the Google Cloud public signing key

sudo mkdir /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

### 5) Step 6: Add the Kubernetes apt repository

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

### 6) Step 7: Update apt package index, install kubelet, kubeadm, kubectl and docker.io, and pin their version

sudo apt-get update

sudo apt-get install -y kubelet=1.28.1-1.1 kubeadm=1.28.1-1.1 kubectl=1.28.1-1.1 docker.io

sudo apt-mark hold kubelet kubeadm kubectl docker.io

### 7) Step 8: Set the group driver for runc to systemd required for the kubelet

sudo mkdir /etc/containerd

sudo containerd config default > /etc/containerd/config.toml

sudo sed -i 's/ SystemdCgroup = false/ SystemdCgroup = true/' /etc/containerd/config.toml

sudo systemctl restart containerd

sudo systemctl restart kubelet

### 8) Step 9: Initialize k8s cluster with user defined network (Network must be /16, and must NOT match underlying LAN CIDR)

ONLY IN MASTER SERVER

kubeadm config images pull

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

Note: Please capture the worker node token as mentioned below)

(kubeadm join 172.31.8.58:6443 --token 1wkfhz.afxntvixlpms8xv2 \

--discovery-token-ca-cert-hash sha256:3f3bf9a068159afcaaba3306001d6755f48f092f861a4df978ec7694e12dcb50)

### 9) Step 10: Setup kubectl for user on 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

### 10) Step 11: Setup Calico SDN

### if pod network in "kubeadm init" is not =192.168.0.0/16, then edit downloaded custom-resources.yaml file accordingly (edit cidr= entry, default is 192.168.0.0/16)

kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/tigera-operator.yaml

curl https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/custom-resources.yaml -O

kubectl create -f custom-resources.yaml

Only in worker:

### Step 12: Join worker node(s):

kubeadm join 172.31.8.58:6443 --token 1wkfhz.afxntvixlpms8xv2 \

--discovery-token-ca-cert-hash sha256:3f3bf9a068159afcaaba3306001d6755f48f092f861a4df978ec7694e12dcb50

**Validation:**

watch -n5 kubectl get po -A

kubectl get nodes

