**Kubernetes 1.30.2 Cluster on Ubuntu 22.04 LTS**

# Prerequisites

* Ubuntu 22.04.4 LTS installed on all nodes.
* User with sudo privileges.
* Kubernetes 1.30.2

## Step-by-Step Installation

## Step1: Disable Swap on All Nodes

swapoff -a

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

## Step2: Enable IPv4 Packed Forwarding

**( sysctl params required by setup, params persist across reboots )**

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

net.ipv4.ip\_forward = 1

EOF

**( Apply sysctl params without reboot )**

sudo sysctl –system

## Step3: Verify IPv4 Packet Forwarding

sysctl net.ipv4.ip\_forward

## Step 4: Install containerd

**# Add Docker's official GPG key:**

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

**# Add the repository to Apt sources:**

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update && sudo apt-get install containerd.io && systemctl enable --now containerd

**Step 5: Install CNI Plugin**

wget https://github.com/containernetworking/plugins/releases/download/v1.4.0/cni-plugins-linux-amd64-v1.4.0.tgz

mkdir -p /opt/cni/bin

tar Cxzvf /opt/cni/bin cni-plugins-linux-amd64-v1.4.0.tgz

**Step 6: Forward IPv4 and Configure iptables**

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

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

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

sudo sysctl --system

sysctl net.bridge.bridge-nf-call-iptables net.bridge.bridge-nf-call-ip6tables net.ipv4.ip\_forward

modprobe br\_netfilter

sysctl -p /etc/sysctl.conf

**Step 7: Modify containerd Configuration for systemd Support**

sudo nano /etc/containerd/config.toml

**#Fetch config.toml configuration from the below link and save it on your local config.toml file.**

https://github.com/Musaele/Public-code.git

**Step 8: Restart containerd and Check the Status**

sudo systemctl restart containerd && systemctl status containerd

**Step 9: Install kubeadm, kubelet, and kubectl**

sudo apt-get update

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

sudo mkdir -p -m 755 /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

## **Following commands should be Executed on Master Node**

**Step 10: Initialize the Cluster and Install CNI**

sudo kubeadm config images pull

sudo kubeadm init

**#After Initialzing the Cluster Connect to it and apply the CNI yaml.**

**#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

**#Alternatively, if you are the root user, you can run:**

export KUBECONFIG=/etc/kubernetes/admin.conf

**Step11: YAML configuration to your Kubernetes cluster**

Nano net.yaml

**#paste the following content of net.yaml from the following github link.**

https://github.com/Musaele/Public-code.git

#apply it by using command

kubectl apply –f net.yaml

**Step 11: Join Worker Nodes to the Cluster**

Run the command you got as a result of running “sudo kubeadm init”