Kubernetes Production Cluster Setup Guide

Cluster Information:

- Master Node IP: 190.168.8.122

- Worker Node IP: 190.168.8.123

This guide outlines commands to be run on:

* Both Nodes
* Master Node only
* Worker Node only

# STEP 1: System Preparation (Run on BOTH NODES)

# Set Hostnames # On Master:

sudo hostnamectl set-hostname master # On Worker:

sudo hostnamectl set-hostname worker1

# Update /etc/hosts (run on both) sudo nano /etc/hosts

# Add the following:

190.168.8.122 master

190.168.8.123 worker1

# Disable Swap sudo swapoff -a

sudo sed -i '/ swap / s/^/#/' /etc/fstab

# Load Kernel Modules

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

br\_netfilter EOF

sudo modprobe overlay sudo modprobe br\_netfilter

# Configure sysctl

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf net.bridge.bridge-nf-call-ip6tables = 1

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

net.ipv4.ip\_forward = 1 EOF

sudo sysctl --system

# Install dependencies sudo apt-get update -y

sudo apt-get install -y apt-transport-https ca-certificates curl gpg lsb-release software-properties-common

# STEP 2: Install containerd, runc, and CNI (BOTH NODES)

# Install containerd wget

https://github.com/containerd/containerd/releases/download/v1.7.13/containerd-1.7.13-linux-amd64.t ar.gz

sudo tar -C /usr/local -xzvf containerd-1.7.13-linux-amd64.tar.gz

# Setup containerd service

sudo mkdir -p /usr/local/lib/systemd/system/

sudo curl -o /usr/local/lib/systemd/system/containerd.service https://raw.githubusercontent.com/containerd/containerd/main/containerd.service

sudo systemctl daemon-reexec sudo systemctl daemon-reload

sudo systemctl enable --now containerd

# Install runc

wget https://github.com/opencontainers/runc/releases/download/v1.1.12/runc.amd64 sudo install -m 755 runc.amd64 /usr/local/sbin/runc

# Install CNI plugins wget

https://github.com/containernetworking/plugins/releases/download/v1.3.0/cni-plugins-linux-amd64-v1

.3.0.tgz

sudo mkdir -p /opt/cni/bin

sudo tar -C /opt/cni/bin -xzvf cni-plugins-linux-amd64-v1.3.0.tgz

# Configure containerd

sudo mkdir -p /etc/containerd

containerd config default | sudo tee /etc/containerd/config.toml

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

sudo systemctl restart containerd

# STEP 3: Install Kubernetes Components (BOTH NODES)

# Add Kubernetes repository sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.32/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:/v1.32/deb/ /" | sudo tee /etc/apt/sources.list.d/kubernetes.list

# Install kubelet, kubeadm, kubectl sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl sudo apt-mark hold kubelet kubeadm kubectl sudo systemctl enable --now kubelet

# STEP 4: Initialize Kubernetes (MASTER NODE ONLY)

# Create kubeadm config file

cat <<EOF | sudo tee masterconfig.yaml apiVersion: kubeadm.k8s.io/v1beta3 kind: InitConfiguration

nodeRegistration:

name: master

criSocket: unix:///run/containerd/containerd.sock kubeletExtraArgs:

cgroup-driver: systemd

---

apiVersion: kubeadm.k8s.io/v1beta3 kind: ClusterConfiguration kubernetesVersion: v1.32.0

controlPlaneEndpoint: "190.168.8.122:6443" networking:

podSubnet: "192.168.0.0/16" EOF

# Initialize master

sudo kubeadm init --config=masterconfig.yaml --upload-certs

# Set up kubectl for the admin user mkdir -p $HOME/.kube

sudo cp /etc/kubernetes/admin.conf $HOME/.kube/config sudo chown $(id -u):$(id -g) $HOME/.kube/config

# Install Calico CNI plugin

kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.30.1/manifests/calico.yaml

# STEP 5: Join Worker Node (WORKER NODE ONLY)

# Run the join command (copied from master output)

# Example format (replace token/hash with actual values):

sudo kubeadm join 190.168.8.122:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>

# (Optional) To regenerate join command on master: kubeadm token create --print-join-command

# STEP 6: Post-Setup Verification (MASTER NODE ONLY)

# Check node status kubectl get nodes

# Label worker node

kubectl label node worker1 node-role.kubernetes.io/worker=

# Check all pods kubectl get pods -A