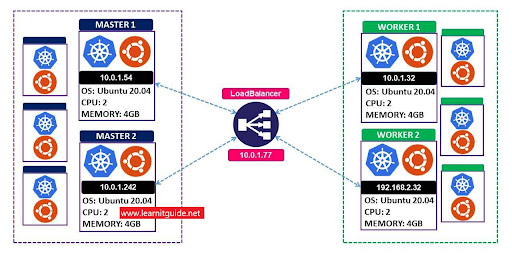
# Kubernetes Multi Master Setup with LoadBalancer on Ubuntu



# Link of the documentation

# https://www.learnitguide.net/2021/10/kubernetes-multi-master-setup-with.html

# HAProxy (Load Balancer) Installation and Configuration

## Step 1: Install HAProxy

sudo apt update

sudo apt install -y haproxy

## Step 2: Configure HAProxy

Edit the HAProxy configuration file **/etc/haproxy/haproxy.cfg** using a text editor. In this example, we'll use Vim:

sudo vim /etc/haproxy/haproxy.cfg

Paste the following content into the **haproxy.cfg** file:

**# haproxy.cfg**

global

log /dev/log local0

log /dev/log local1 notice

chroot /var/lib/haproxy

stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners

stats timeout 30s

user haproxy

group haproxy

daemon

# Default SSL material locations

ca-base /etc/ssl/certs

crt-base /etc/ssl/private

# SSL configurations (optional, adjust as needed)

defaults

log global

mode http

option httplog

option dontlognull

timeout connect 5000

timeout client 50000

timeout server 50000

errorfile 400 /etc/haproxy/errors/400.http

errorfile 403 /etc/haproxy/errors/403.http

errorfile 408 /etc/haproxy/errors/408.http

errorfile 500 /etc/haproxy/errors/500.http

errorfile 502 /etc/haproxy/errors/502.http

errorfile 503 /etc/haproxy/errors/503.http

errorfile 504 /etc/haproxy/errors/504.http

frontend kubernetes

bind 88.216.198.13:6443

option tcplog

mode tcp

default\_backend kubernetes-master-nodes

backend kubernetes-master-nodes

mode tcp

balance roundrobin

option tcp-check

server master-1 88.216.198.10:6443 check fall 3 rise 2

server master-2 88.216.198.11:6443 check fall 3 rise 2

## Step 3: Save and Exit

## Step 4: Start and Enable HAProxy

sudo systemctl restart haproxy

sudo systemctl enable haproxy

# Kubernetes Master Node Setup

## Step 1: Update System Packages

sudo apt-get update

## Step 2: Configure Kernel Modules

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

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

## Step 3: Configure sysctl Settings

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

## Step 4: Disable Swap

sudo swapoff –a

## Step 5: Install Dependencies and Containerd

sudo apt-get update

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

sudo apt-get install -y containerd

sudo containerd config default

sudo systemctl restart containerd

## Step 6: Install Kubectl

sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

sudo curl -LO "https://dl.k8s.io/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"

echo "$(cat kubectl.sha256) kubectl" | sha256sum --check

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

sudo chmod +x kubectl

sudo mkdir -p ~/.local/bin

sudo mv ./kubectl ~/.local/bin/kubectl

kubectl version –client

## Step 7: Configure Kubernetes APT Repository

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

sudo chmod 755 /etc/apt/keyrings

sudo curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg

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

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

## Step 8: Install Kubernetes Components

sudo apt-get update

sudo apt-get install -y kubelet kubeadm

sudo apt-mark hold kubelet kubeadm

## Step 9: Initialize Kubernetes Cluster

sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=ip of master node

## Step 10: Configure kubectl for the Current User

sudo mkdir -p $HOME/.kube

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

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

## Step 11: Deploy Network Plugin (Flannel)

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

## Step 12: Verify Cluster Status

kubectl get pods --all-namespaces

kubectl get nodes

**Install Worker Node**

## Step 1: Update and Load Kernel Modules

sudo apt-get update

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

## Step 2: Disable Swap

sudo swapoff –a

## Step 3: Install Dependencies and Containerd

sudo apt-get update

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

sudo apt-get install -y containerd

sudo containerd config default

sudo systemctl restart containerd

## Step 4: Install Kubectl

sudo curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

sudo curl -LO "https://dl.k8s.io/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"

echo "$(cat kubectl.sha256) kubectl" | sha256sum --check

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

sudo chmod +x kubectl

sudo mkdir -p ~/.local/bin

sudo mv ./kubectl ~/.local/bin/kubectl

kubectl version –client

## Step 5: Configure Kubernetes APT Repository

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

sudo chmod 755 /etc/apt/keyrings

sudo curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg

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

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

## Step 6: Install Kubernetes Components

sudo apt-get update

sudo apt-get install -y kubelet kubeadm

sudo apt-mark hold kubelet kubeadm

## Step 7: Verify Installation

kubectl get pods --all-namespaces

kubectl get nodes