General Prerequisites

2 Ubuntu 20.04 Server VMs

You can download the `Ubuntu 20.04 server image` from here: https://releases.ubuntu.com/focal/

Note: Be careful to download the Server install image.

VM Resources

- 5-6 GB RAM
- at least 2 Virtual CPUs
- Storage disk of at least ~50-60GB (as well as 2 other additional disks, each having 10-20 GB storage)

Kubernetes installed on both machines

NOTE: These steps should be done both machines. You can easily copy the code and paste it inside your terminal.

Use `ssh` to login into both nodes and the `apt-transport-https` package, which enables working with `http` and `https` in Ubuntu's repositories. Execute the following command:

```
sudo apt install apt-transport-https curl
```

Then, add the `Kubernetes signing key` and the `Kubernetes repository` as a package source to both nodes by executing these commands:

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" >> ~/kubernetes.list
sudo mv ~/kubernetes.list /etc/apt/sources.list.d
```

After that, update the nodes and install `kubelet`, `kubeadm`, `kubectl`, `kubernetes-cni` and `containerd`:

```
sudo apt update
sudo apt-get install -y kubelet kubeadm kubectl kubernetes-cni containerd
```

Kubernetes fails to function in a system that is using `swap memory`. Hence, it must be disabled in all Kubernetes nodes. Execute the following command to disable swap memory:

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

NOTE: These next 2 steps are optional, but will be easier to follow the lab if you do it!

Optional

Your nodes must have unique hostnames for easier identification. If you are deploying a cluster with many nodes, you can set them to identify names for your worker nodes such as node-1, node-2, etc. During the lab, it would be better for your nodes to be named as `kubernetes-master` and `kubernetes-worker`. You have already set them at the time of creating the server. You can update the hostname by running the following command:

```
# Run this on the node you want as master
sudo hostnamectl set-hostname kubernetes-master
```

```
# Run this on the node you want as worker
sudo hostnamectl set-hostname kubernetes-worker
```

Letting Iptables See Bridged Traffic

For the master and worker nodes to correctly see bridged traffic, you should ensure `net.bridge.bridge-nf-call-iptables` is set to 1 in your config. First, ensure the `br_netfilter` module is loaded. You can confirm this by issuing the command:

```
# Configure persistent loading of modules
sudo tee /etc/modules-load.d/k8s.conf <<EOF
overlay
br_netfilter
EOF

# Enable kernel modules
sudo modprobe overlay
sudo modprobe br_netfilter
```

Now, you can run this command to set the value to 1:

```
# Add some settings to sysctl
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 sysctl
sudo sysctl --system
```

Test if it works

```
sudo kubeadm init

# When the init is ready, you can run these commands:
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

# Then to check if everything worked properly:
sudo kubectl get nodes
```

Now you should see something like this:

```
$ kubectl get nodes
NAME                STATUS     ROLES           AGE   VERSION
kubernetes-master   NotReady   control-plane   16s   v1.27.2
```

If you see this, it worked properly! Good job!

Now just reset the cluster:

```
sudo kubeadm reset
```