Kubernetes Cluster Setup (1 Master + 2 Workers)

🔒 Fully Hardened with Firewall, containerd, and Calico CNI

📌 Cluster Details

Role	Hostname	IP Address
Master	`master-node`	`192.168.1.41`
Worker	`node1`	`192.168.1.59`
Worker	`node2`	`192.168.1.60`

✅ Prerequisites (Run on ALL Nodes)

1. Set Hostnames

```
sudo hostnamectl set-hostname master-node    # Change accordingly per node
```

2. Disable Swap & Enable Bridged Networking

```
sudo swapoff -a
sudo sed -i '/swap/d' /etc/fstab

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.ipv4.ip_forward=1
net.bridge.bridge-nf-call-ip6tables=1
net.bridge.bridge-nf-call-iptables=1
EOF

sudo sysctl --system
```

🔥 Firewall Configuration

On All Nodes (Firewalld)

```
sudo systemctl enable --now firewalld

# Common ports
sudo firewall-cmd --permanent --add-port=10250/tcp          # kubelet
sudo firewall-cmd --permanent --add-port=30000-32767/tcp     # NodePort
```

```
# Calico
sudo firewall-cmd --permanent --add-port=179/tcp             # BGP
sudo firewall-cmd --permanent --add-masquerade

# Master node only
sudo firewall-cmd --permanent --add-port=6443/tcp            # API Server
sudo firewall-cmd --permanent --add-port=2379-2380/tcp       # etcd
sudo firewall-cmd --permanent --add-port=10251/tcp           # scheduler
sudo firewall-cmd --permanent --add-port=10252/tcp           # controller-manager

sudo firewall-cmd --reload
```

📆 Install containerd (All Nodes)

```
sudo dnf install -y containerd

sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml

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

sudo systemctl enable --now containerd
```

🧱 Install Kubernetes Components (All Nodes)

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core/stable/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core/stable/rpm/repodata/repomd.xml.key
EOF

sudo dnf install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
```

🧠 Initialize Master Node (Only on 192.168.1.41)

```
sudo kubeadm init \
  --pod-network-cidr=192.168.0.0/16 \
  --apiserver-advertise-address=192.168.1.41
```

Then configure kubectl:

```
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 (On Master)

```
kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/calico.yaml
```

Optional: Enforce default deny policy

```
kubectl apply -f https://docs.projectcalico.org/manifests/default-deny.yaml
```

🧱 Join Worker Nodes

On the master, generate the join command:

```
kubeadm token create --print-join-command
```

Run the printed command on **each worker node** (`192.168.1.59` and `192.168.1.60`).

✅ Verify the Cluster

```
kubectl get nodes -o wide
kubectl get pods -n kube-system
```

You should see:

- All 3 nodes in `Ready` state
- Calico pods running properly

🔐 Final Security Checks

Item	Command
Container runtime works	`crictl info`
All firewall rules active	`sudo firewall-cmd --list-all`
Kubelet running	`systemctl status kubelet`
containerd healthy	`systemctl status containerd`

Item	Command
Pods can communicate	Deploy 2 pods in same/different nodes & test ping