Kubernetes 1 Master 1 Worker Setup (Ubuntu 24.04 LTS)

✅ Prerequisites (Both Nodes)

	Master Node	**Worker Node**
Hostname	master	worker
CPU / RAM	2 CPU / 2GB RAM	1 CPU / 2GB RAM
OS	Ubuntu 24.04 LTS	Ubuntu 24.04 LTS
K8s version	v1.29.x	v1.29.x

✅ Step 1: Basic System Preparation (Both Nodes)

1.1 Update and Upgrade

```
sudo apt update && sudo apt upgrade -y
```

1.2 Set Hostnames

```
sudo hostnamectl set-hostname master    # On master
sudo hostnamectl set-hostname worker    # On worker
```

1.3 Edit /etc/hosts

```
sudo nano /etc/hosts

<Master-IP> master
<Worker-IP> worker
```

1.4 Disable Swap (Required)

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

1.5 Load Kernel Modules

```
sudo modprobe overlay
sudo modprobe br_netfilter
```

1.6 Apply sysctl params

```
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

sudo sysctl --system
```

✅ Step 2: Install containerd (Both Nodes)

2.1 Install containerd

```
sudo apt install -y containerd
```

2.2 Configure containerd defaults

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

2.3 Enable SystemdCgroup in the config

```
sudo nano /etc/containerd/config.toml

[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
SystemdCgroup = true
```

2.4 Restart and enable containerd

```
sudo systemctl restart containerd
sudo systemctl enable containerd
```

✅ Step 3: Install Kubernetes Components (Both Nodes)

3.1 Add Kubernetes repo key

Add version K8s version to v1.29, you can install other versions

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

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo
tee /etc/apt/trusted.gpg.d/kubernetes-apt-keyring.asc > /dev/null
```

3.2 Add the Kubernetes apt repository

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

3.3 Install kubelet, kubeadm, kubectl

```
sudo apt update
sudo apt install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

✅ Step 4: Initialize the Master Node

4.1 Initialize the cluster (Master node only)

<Master-IP> => IP of Master

```
sudo kubeadm init --apiserver-advertise-address=<Master-IP>
--pod-network-cidr=172.29.0.0/16
```

4.2 Setup kubeconfig for kubectl

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

✅ Step 5: Install Pod Network (Master Node)

5.1 Apply Calico network plugin

```
kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/ca
lico.yaml
```

✅ Step 6: Join Worker Node to Cluster (Worker Node)

6.1 Run the kubeadm join command (from Step 4 output)

```
sudo kubeadm join <Master-IP>:6443 --token <token>
--discovery-token-ca-cert-hash sha256:<hash>
```

✅ Step 7: Verify Cluster (Master Node)

7.1 Get nodes status

```
kubectl get nodes
```

Expected output:

```
NAME          STATUS   ROLES           AGE    VERSION
master-node   Ready    control-plane   10m    v1.29.x
worker-node   Ready    <none>          2m     v1.29.x
```

✅ Step 8: Deploy a Test NGINX App (Optional)

```
kubectl create deployment nginx --image=nginx
kubectl expose deployment nginx --port=80 --type=NodePort
kubectl get svc nginx
```

Test access by visiting NodeIP:NodePort.

✅ Summary Table

Component	Installed On
containerd	Both Nodes
kubelet	Both Nodes
kubeadm	Both Nodes
kubectl	Master (optional on Worker)

✅ What's Next?

- Install MetalLB (Load Balancer for bare metal)
- Deploy Ingress Controller (NGINX)
- Set up Helm for package management

✅ Install ArgoCD

1. Ref: https://www.youtube.com/watch?v=MeU5_k9ssrs
2. Install ArgoCD in K8s cluster: https://argo-cd.readthedocs.io/en/stable/getting_started/
   kubectl port-forward -n argocd svc/argocd-server 8080:443
   Username: admin
   Pass:
   - kubectl get secret argocd-initial-admin-secret -n argocd -o yaml
   - echo "<hash>" | base64 --decode
3. Configure ArgoCD with “Application” CRD
4. Test our setup by updating Deployment.yaml file