# **Kubernetes Master and Worker Node Installation on AWS (Ubuntu t3.medium)**

## **Master Node Installation**

### **Prerequisites**

AWS Instance Details:  
- Instance Type: t3.medium  
- Operating System: Ubuntu  
- User Access: Ensure you have SSH access to the instance.

### **Security Group Settings**

Open required ports:  
- 6443 (K8s API Server)  
- 2379-2380 (etcd)  
- 10250-10255 (Kubelet)  
- 179 (Calico Network Policy)  
- 443 (HTTPS)  
- 53 (DNS)  
- 30000-32767 (NodePort Services)

### **Step-by-Step Installation**

**1. Set Hostname:**  
 sudo hostnamectl set-hostname HirdeshMaster

**2. Disable Swap: #Kubernetes requires swap to be disabled.** sudo swapoff -a

**3. Load Kernel Modules for Networking: Loads essential kernel modules.**  
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf  
overlay  
br\_netfilter  
EOF  
sudo modprobe overlay  
sudo modprobe br\_netfilter

**4. Configure Sysctl Parameters: Ensures proper packet forwarding and iptables 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

**5. Verify Kernel Modules: Checks if the required kernel modules are loaded.**

lsmod | grep br\_netfilter

lsmod | grep overlay

**6. Install container runtime (containerd) and Kubernetes components:  
# Update Package Lists**

sudo apt-get update

**# Install Required Packages**  
sudo apt-get install -y ca-certificates curl

**# Add Docker GPG Key & Repository**sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL [https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc](https://download.docker.com/linux/ubuntu/gpg%20-o%20/etc/apt/keyrings/docker.asc)

sudo chmod a+r /etc/apt/keyrings/docker.asc

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo \"$VERSION\_CODENAME\") stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**# Install containerd**

sudo apt-get update  
sudo apt-get install -y containerd.io

**# Configure containerd**  
containerd config default | sed -e 's/SystemdCgroup = false/SystemdCgroup = true/' -e 's/sandbox\_image = "registry.k8s.io\/pause:3.6"/sandbox\_image = "registry.k8s.io\/pause:3.9"/' | sudo tee /etc/containerd/config.toml

**# Restart and Verify containerd: Ensures containerd is running.**  
sudo systemctl restart containerd

sudo systemctl status containerd

**7. Install Kubernetes Components (kubelet, kubeadm, kubectl):Add Kubernetes Repository**

sudo apt-get update

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

**#Add Kubernetes GPG Key**

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

**# Add Kubernetes Repo:**

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

**# Install Kubernetes Components:**

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

**8. Initialize Kubernetes Master Node: Initializes the Kubernetes master.**  
 sudo kubeadm init

**# Setup kubeconfig for Current User. Configures kubectl access.**

mkdir -p "$HOME"/.kube

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

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

**9. Install Calico for Networking**

kubectl apply -f <https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml>

## **Worker Node Installation**

### **Prerequisites**

AWS Instance Details:  
- Instance Type: t3.medium  
- Operating System: Ubuntu  
- User Access: Ensure SSH access to the instance.

### **Security Group Settings**

Open the required Kubernetes Ports:  
- 10250 (Kubelet API)  
- 30000-32767 (NodePort Services)  
- 443 (HTTPS)  
- 53 (DNS)

### **Step-by-Step Installation**

**1. Set Hostname:**  
 sudo hostnamectl set-hostname KumarSlave

**2. Disable Swap:**  
 sudo swapoff -a

**3. Load Kernel Modules for Networking: Loads essential kernel modules.**

**OR**

**Load Kernel Modules and Configure Networking (Same as Master Node)**  
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf  
overlay  
br\_netfilter  
EOF  
sudo modprobe overlay  
sudo modprobe br\_netfilter

**4. Configure Sysctl Parameters: Ensures proper packet forwarding and iptables 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

**5. Verify Kernel Modules: Checks if the required kernel modules are loaded.**

lsmod | grep br\_netfilter

lsmod | grep overlay

**6. Install container runtime (containerd) and Kubernetes components:**

**OR**

**Install container runtime (containerd) and Kubernetes components (Same as Master Node)  
# Update Package Lists**

sudo apt-get update

**# Install Required Packages**  
sudo apt-get install -y ca-certificates curl

**# Add Docker GPG Key & Repository**sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL [https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc](https://download.docker.com/linux/ubuntu/gpg%20-o%20/etc/apt/keyrings/docker.asc)

sudo chmod a+r /etc/apt/keyrings/docker.asc

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo \"$VERSION\_CODENAME\") stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**# Install containerd**

sudo apt-get update  
sudo apt-get install -y containerd.io

**# Configure containerd**  
containerd config default | sed -e 's/SystemdCgroup = false/SystemdCgroup = true/' -e 's/sandbox\_image = "registry.k8s.io\/pause:3.6"/sandbox\_image = "registry.k8s.io\/pause:3.9"/' | sudo tee /etc/containerd/config.toml

**# Restart and Verify containerd: Ensures containerd is running.**  
sudo systemctl restart containerd

sudo systemctl status containerd

**7. Install Kubernetes Components (kubelet, kubeadm, kubectl):Add Kubernetes Repository**

sudo apt-get update

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

**#Add Kubernetes GPG Key**

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

**# Add Kubernetes Repo:**

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

**# Install Kubernetes Components:**

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

**8. Join Worker Node to Master:**  
kubeadm join 172.31.16.239:6443 --token 3q2pux.rir12z85b797fhb6 \  
--discovery-token-ca-cert-hash sha256:b6710716169bea5710abf3c759f84e9778a1fa1d4960645575aa9038012af831

### Verification

**1. Check Nodes from Master Node:**  
kubectl get nodes

**Expected Output:  
NAME STATUS ROLES AGE VERSION  
HirdeshMaster Ready control-plane 10m v1.29  
KumarSlave Ready worker 2m v1.29**

**2. Check Cluster Info:**  
 kubectl cluster-info

### **Final Steps**

### 🎉 **Your Kubernetes Master and Worker Nodes have been successfully added to the cluster!** 🚀 You are now ready to deploy pods, services, and workloads in your Kubernetes environment.