

**DEVOPS SHACK**

**KUBERNETES Local Setup**

**DEVOPS SHACK**

**Kubernetes Local Setup**

**Detailed Kubernetes Installation on Ubuntu (Master and Worker Nodes)**

This guide outlines the steps for setting up a Kubernetes cluster on Ubuntu machines. It includes instructions for both master and worker nodes and covers configurations for networking and container runtimes. The cluster is initialized using kubeadm with Calico as the network plugin.

**Prerequisites**

* All machines (both master and worker nodes) should run a compatible version of Ubuntu (e.g., Ubuntu 20.04 or 22.04).
* Ensure root or sudo privileges on all machines.
* At least two nodes: one for the master and one or more for the worker nodes.
* **Minimum resources**:
  + Master node: 2 CPUs and 2GB RAM
  + Worker nodes: 1 CPU and 1GB RAM each

**Step 1: Configuration on All Nodes (Master and Worker Nodes)**

**Update and Upgrade Packages**

sudo apt update && sudo apt upgrade -y

**Disable Swap**

sudo swapoff -a

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

**Load Required Kernel Modules**

sudo tee /etc/modules-load.d/containerd.conf <<EOF

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

**Network Configuration for Kubernetes**

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

**Install Required Dependencies**

sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates

**Step 2: Install and Configure containerd**

**Add Docker GPG Key and Repository**

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/docker.gpg

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

**Install containerd**

sudo apt update

sudo apt install -y containerd.io

**Configure containerd**

containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1

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

**Restart and Enable containerd**

sudo systemctl restart containerd

sudo systemctl enable containerd

**Step 3: Install Kubernetes Components**

**Add Kubernetes GPG Key and Repository**

echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /" | sudo tee /etc/apt/sources.list.d/kubernetes.list

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

**Install Kubernetes Components**

sudo apt update

sudo apt install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

**Step 4: Initialize the Master Node**

**Run kubeadm init**

sudo kubeadm init

**Set Up kubeconfig for the Master Node**

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: Set Up Networking (Calico)**

**Install Calico Networking**

kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/calico.yaml

**Step 6: Join Worker Nodes to the Cluster**

**Use the Join Command**

Run the join command from the output of the kubeadm init process on each worker node. It will look like this:

sudo kubeadm join <master-node-ip>:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>

**Verify Worker Nodes Have Joined**

On the master node, check the nodes:

kubectl get nodes

**Conclusion**

You have successfully set up a Kubernetes cluster with one master node and one or more worker nodes. The cluster uses containerd as the container runtime and Calico for networking. You can now deploy and manage containerized applications using Kubernetes' orchestration capabilities.