**Kubernetes (k8s) step-by-step installation guide on Ubuntu 20.04/22.04/24.04 from scratch**

**Step 1: Update Your System**

**Update the package list and upgrade installed packages:**

**sudo apt update && sudo apt upgrade -y**

**sudo reboot**

**Step 2: Install Dependencies**

**Install required tools and packages:**

**sudo apt install -y apt-transport-https ca-certificates curl gnupg lsb-release**

**Step 3: Disable Swap**

**Kubernetes requires swap to be disabled:**

**sudo swapoff -a**

**sudo sed -i '/ swap / s/^/#/' /etc/fstab**

**Step 4: Install Docker or containerd**

Kubernetes needs a container runtime. We'll use **containerd** (recommended).

**4.1: Install containerd**

**sudo apt install -y containerd**

**4.2: Configure containerd**

**Generate the default configuration file:**

**sudo mkdir -p /etc/containerd**

**sudo containerd config default | sudo tee /etc/containerd/config.toml**

**Restart containerd:**

**sudo systemctl restart containerd**

**sudo systemctl enable containerd**

**Step 5: Install Kubernetes Components**

Install **kubeadm**, **kubelet**, and **kubectl**.

**5.1: Add the Kubernetes APT repository**

**curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -**

**echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list**

**sudo apt update**

**5.2: Install kubeadm, kubelet, and kubectl**

**sudo apt install -y kubelet kubeadm kubectl**

**sudo apt-mark hold kubelet kubeadm kubectl**

**Step 6: Initialize the Kubernetes Cluster**

Run the following command on the **master node**:

**sudo kubeadm init --pod-network-cidr=192.168.0.0/16**

**6.1: Configure kubectl for the current user**

**mkdir -p $HOME/.kube**

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

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

**6.2: Save the Join Command**

The output of kubeadm init will include a kubeadm join command for worker nodes. Save this for later.

**Step 7: Install a Pod Network Add-On**

Choose a CNI (Container Network Interface). We'll use **Calico**.

**7.1: Apply the Calico manifest**

**kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml**

**Step 8: Add Worker Nodes**

**Run the kubeadm join command from Step 6.2 on each worker node:**

**sudo kubeadm join <master-ip>:6443 --token <token> \**

**--discovery-token-ca-cert-hash sha256:<hash>**

**Step 9: Verify the Cluster**

**9.1: Check Nodes**

On the master node, check the status of nodes:

**kubectl get nodes**

**9.2: Check Pods**

Verify that the cluster is working by checking pods in the kube-system namespace:

**kubectl get pods -n kube-system**

**Optional: Enable Autocompletion for kubectl**

**sudo apt install -y bash-completion**

**echo 'source <(kubectl completion bash)' >>~/.bashrc**

**source ~/.bashrc**

**Troubleshooting Tips**

1. **Reset the Cluster**  
   If the installation fails, reset kubeadm and try again:

**sudo kubeadm reset -f**

**sudo rm -rf $HOME/.kube**

1. **Check Logs**  
   Use the following commands to troubleshoot issues:

**journalctl -xeu kubelet**

**kubectl describe pod <pod-name> -n kube-system**

This guide provides a basic Kubernetes cluster installation. For production, consider high-availability setups, security configurations, and monitoring tools.