Manual Kubernetes Installation on AWS EC2 Instances

## Step 1: Launch EC2 Instances

* Launch at least 2 EC2 instances (1 master, 1+ workers) using Ubuntu AMI.
* Choose instance types like t2.medium or higher.
* Open the following ports in the security group:
* TCP 22 (SSH)
* TCP 6443 (Kubernetes API server)
* TCP 2379-2380 (etcd server client API)
* TCP 10250-10252 (Kubelet, scheduler, controller-manager)
* TCP/UDP 8472 (Flannel)
* UDP 8285, 8472 (Weave)
* TCP 10255 (Read-only Kubelet)
* TCP 30000-32767 (NodePort Services)

## Step 2: Prepare All Nodes

* sudo hostnamectl set-hostname <hostname>
* sudo swapoff -a
* sudo sed -i '/ swap / s/^/#/' /etc/fstab
* cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf  
  br\_netfilter  
  EOF
* sudo modprobe br\_netfilter
* cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf  
  net.bridge.bridge-nf-call-ip6tables = 1  
  net.bridge.bridge-nf-call-iptables = 1  
  net.ipv4.ip\_forward = 1  
  EOF
* sudo sysctl --system

## Step 3: Install Container Runtime (containerd)

* sudo apt update && sudo apt install -y containerd
* sudo mkdir -p /etc/containerd
* containerd config default | sudo tee /etc/containerd/config.toml
* sudo systemctl restart containerd
* sudo systemctl enable containerd

## Step 4: Install Kubernetes Components

* sudo apt update && sudo apt install -y apt-transport-https curl
* curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
* cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list  
  deb https://apt.kubernetes.io/ kubernetes-xenial main  
  EOF
* sudo apt update
* sudo apt install -y kubelet kubeadm kubectl
* sudo apt-mark hold kubelet kubeadm kubectl

## Step 5: Initialize Master Node

* sudo kubeadm init --pod-network-cidr=192.168.0.0/16
* mkdir -p $HOME/.kube
* sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
* sudo chown $(id -u):$(id -g) $HOME/.kube/config

## Step 6: Install CNI Plugin (Calico)

* kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.0/manifests/calico.yaml

## Step 7: Join Worker Nodes

* Run the kubeadm join command from master node output on each worker node.
* If lost, regenerate with:
* kubeadm token create --print-join-command

## Step 8: Verify Cluster

* kubectl get nodes
* kubectl get pods -A