Kubernetes Installation:

Step 1: Launch EC2 Instances

- **AMI:** Ubuntu 24.04 LTS
- **Instance types:**
  - Control plane: **t3.medium** (2 vCPU, 4GB RAM)
  - Worker: **t3.small** or larger
- **Security group (open these ports):**

22	(SSH)
6443	(Kubernetes API server)
2379-2380	(etcd)
10250-10259	(kubelet & control-plane)
30000-32767	(NodePort)

Step 2: Update and Upgrade Ubuntu (all nodes)

Run on each node:

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

Step 3: Disable Swap (all nodes)

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

Step 4: Add Kernel Parameters (all nodes)

```
sudo tee /etc/modules-load.d/containerd.conf <<EOF
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
```

Configure the critical kernel parameters for Kubernetes using the following:

```
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
```

Then, reload the changes:

```
sudo sysctl --system
```

Step 5: Install Containerd Runtime (all nodes)

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

Enable the Docker 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"
```

Update the package list and install containerd:

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

Configure containerd to start using systemd as cgroup:

```
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 the containerd service:

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

Step 6: Add Apt Repository for Kubernetes (all nodes)

```
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
```

Step 7: Install Kubectl, Kubeadm, and Kubelet (all nodes)

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

Step 8: Initialize Kubernetes Cluster with Kubeadm (master node)

On the **master node**:

```
sudo kubeadm init
```

Step 9: Configure kubectl on Master

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

Step 10: Install Pod Network (Calico or Flannel)

```
kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/v3.25.
0/manifests/calico.yaml
```

Step 11: Join Worker Nodes

On **each worker node**:

```
kubeadm join 138.197.184.45:6443 --token
72ww2b.6orffywqcf5s4p2z --discovery-token-ca-cert-hash
sha256:aafb79cdd45a6e3b3fac01fb3efba0817360b01f90a4b6c3f11567
108a36ba67
```

Step 12: Verify Cluster

On the master:

```
kubectl get nodes
kubectl get pods -A
```

All nodes should be `Ready`.