# **Rapport de TP – Installation d’un Cluster Kubernetes:**

## **Préparation commune des nœuds (Master & Workers)**:

bash

CopyEdit

sudo apt update && sudo apt upgrade -y  
sudo apt install curl apt-transport-https ca-certificates software-properties-common

**Ajout du dépôt Docker et installation** :

bash

CopyEdit

curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg  
  
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] <https://download.docker.com/linux/ubuntu> $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
  
sudo apt update  
sudo apt install -y docker-ce docker-ce-cli containerd.io

**Désactivation du swap** (obligatoire pour kubelet) :

bash

CopyEdit

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

**Configuration des modules noyau requis** :

bash

CopyEdit

sudo tee /etc/modules-load.d/containerd.conf <<EOF  
overlay  
br\_netfilter  
EOF  
  
sudo modprobe overlay  
sudo modprobe br\_netfilter

**Paramètres sysctl pour le routage réseau** :

bash

CopyEdit

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

**Configuration de containerd** :

bash

CopyEdit

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  
sudo systemctl restart containerd  
sudo systemctl enable containerd

**Installation des outils Kubernetes** :

bash

CopyEdit

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

## **Initialisation du Nœud Maître**

**Initialisation du cluster** :

bash

CopyEdit

sudo kubeadm init

**Configuration de l’environnement utilisateur (accès kubectl)** :

bash

CopyEdit

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

**Déploiement du réseau CNI** :

Deux options ont été testées :

### **Option 1 : Calico**

bash

CopyEdit

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

### **Option 2 : Weave**

bash

CopyEdit

kubectl apply -f <https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s-1.11.yaml>

## **3️⃣ Ajout des Nœuds Workers**

Sur chaque worker, répéter **toutes les étapes de préparation** (1️⃣).

Ensuite, **depuis le master**, récupérer la commande d'adhésion :

bash

CopyEdit

kubeadm token create --print-join-command

Puis l’exécuter sur chaque **worker** :

bash

CopyEdit

sudo kubeadm join <IP\_MASTER>:6443 --token <TOKEN> --discovery-token-ca-cert-hash sha256:<HASH>