

# 4KUBE



## Installation Cluster : Fleetman

**BOULAY** Claude

**ROSSIGNOL** Hugo

**COUBLANT** Steven



# Sommaire

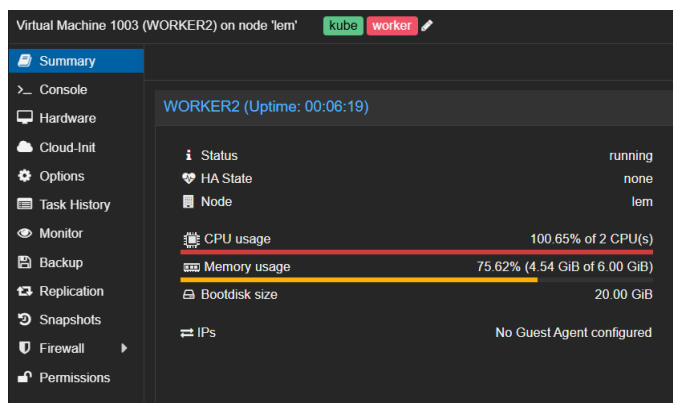
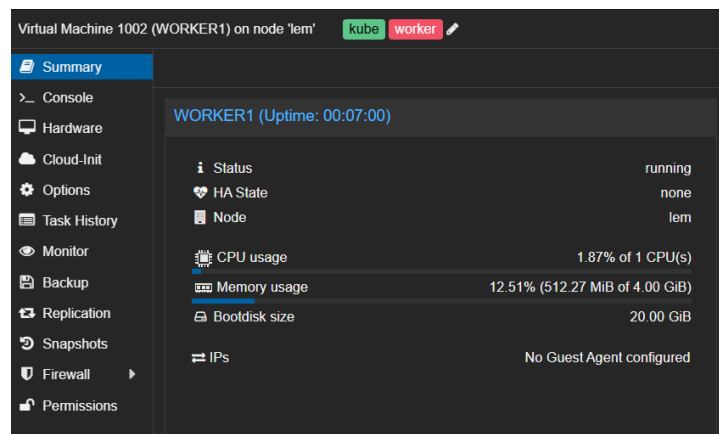
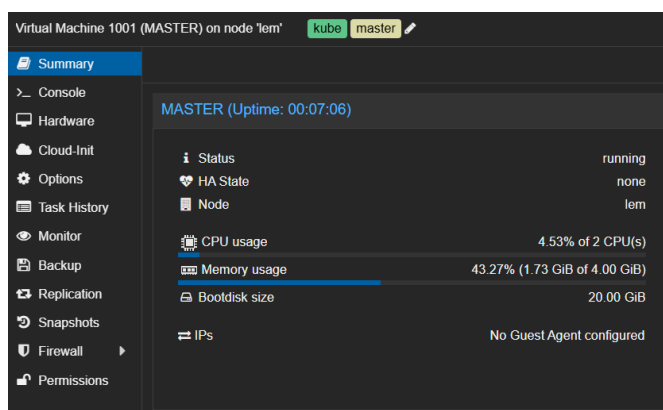
<b>1 – Introduction .....</b>	<b>3</b>
<b>1.1 – Architecture .....</b>	<b>3</b>
<b>2 – Installation du Master.....</b>	<b>4</b>
<b>2.1 – Prérequis .....</b>	<b>4</b>
<b>2.2 – Charger les modules kernel .....</b>	<b>4</b>
<b>2.3 – Installation de containerd .....</b>	<b>5</b>
<b>2.4 – Installation de QEMU.....</b>	<b>5</b>
<b>2.5 – Installation de Kubernetes .....</b>	<b>5</b>
<b>2.6 – Initialisation du cluster .....</b>	<b>6</b>
<b>2.7 – Initialisation de Flannel .....</b>	<b>6</b>
<b>2.8 – Générer la commande pour rejoindre .....</b>	<b>6</b>
<b>3 – Installation du Worker .....</b>	<b>6</b>
<b>3.1 – Étapes d’installations (identiques au master) .....</b>	<b>6</b>
<b>3.2 – Rejoindre le cluster .....</b>	<b>6</b>
<b>3.3 Vérification.....</b>	<b>7</b>
<b>4 – Déploiement FleetMan .....</b>	<b>7</b>
<b>5 – Sources et références.....</b>	<b>8</b>

# 1 – Introduction

Cette documentation décrit l'installation d'un cluster Kubernetes sur trois nodes Debian 11 hébergées sur Proxmox.

## 1.1 – Architecture

- Master node : (2 CPU, 4GB RAM, 20GB disque)
- 2 Worker nodes: (2 CPU, 4GB RAM, 20GB disque)
- CNI: Flannel
- Runtime: Containerd



## 2 – Installation du Master

Nous partons du principe que la machine virtuelle est déjà créée.

### 2.1 – Prérequis

Désactiver le swap :

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

# Vérifier que le swap est bien désactivé

free -h
```

### 2.2 – Charger les modules kernel

```
sudo modprobe br_netfilter
sudo modprobe overlay

# Rendre persistant au redémarrage

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# Appliquer

sudo sysctl -system
```

## 2.3 – Installation de containerd

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

curl -fsSL https://download.docker.com/linux/debian/gpg | 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/debian $(lsb_release -cs)
stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

apt update
apt install -y containerd.io

mkdir -p /etc/containerd
containerd config default | tee /etc/containerd/config.toml

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

systemctl restart containerd
systemctl enable containerd
```

## 2.4 – Installation de QEMU

Pour des questions de compatibilité, il sera nécessaire d'installer QEMU pour supporter les images arm64 du projet.

```
sudo apt install -y qemu-user-static binfmt-support
```

## 2.5 – Installation de Kubernetes

```
curl -fsSL
https://pkgs.k8s.io/core:/stable:/v1.33/deb/Release.key | sudo
gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-
keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.33/deb/ /' |
sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list

sudo apt update
sudo apt install -y kubelet kubeadm kubectl
sudo systemctl enable kubelet
```

## 2.6 – Initialisation du cluster

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16

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

kubectl get nodes
```

## 2.7 – Initialisation de Flannel

```
kubectl apply -f \ https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml

kubectl get pods -n kube-flannel -w
```

## 2.8 – Générer la commande pour rejoindre

```
kubeadm token create --print-join-command
```

Veillez à bien conserver l’output de cette commande, elle servira plus tard à faire rejoindre nos Workers au cluster.

# 3 – Installation du Worker

Cette étape est à répéter selon le nombre de workers souhaités

## 3.1 – Étapes d’installations (identiques au master)

Toujours en partant du principe que votre machine est déjà créée et fonctionnel, répétez les sections **2.1** à **2.5**

## 3.2 – Rejoindre le cluster

Entrez la commande générée dans la section **2.7** :

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

### 3.3 Vérification

Pour vérifier que nos deux nodes Worker ont bien intégrés notre cluster :

```
kubectl get nodes
```

```
kube@MASTER:~$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
master      Ready    control-plane   11d   v1.33.5
worker1     Ready    <none>        11d   v1.33.5
worker2     Ready    <none>        10d   v1.33.5
```

## 4 – Déploiement FleetMan

```
kubectl apply -f k8s-manifests/
kubectl get pods -n fleetman
kubectl get service -n fleetman
```

# Voir la repartition sur les nodes

```
Kubectl get pods -n fleetman -o wide
```

```
kube@MASTER:~$ kubectl get pods -n fleetman -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
fleetman-api-gateway-6684564f8f-44l4g    1/1     Running   0          44s   10.244.1.228   worker1
fleetman-api-gateway-6684564f8f-4wq5h    1/1     Running   0          44s   10.244.2.186   worker2
fleetman-mongodb-5dc8bb7f58-z7vsz        1/1     Running   0          44s   10.244.2.187   worker2
fleetman-position-simulator-55c9c4d49c-dlbt6 1/1     Running   0          44s   10.244.2.188   worker2
fleetman-position-tracker-6f7d78d99d-24mj4 1/1     Running   0          44s   10.244.2.189   worker2
fleetman-position-tracker-6f7d78d99d-zndx8 1/1     Running   0          44s   10.244.1.229   worker1
fleetman-queue-6cbc7df5cb-6hlh6         1/1     Running   0          44s   10.244.2.190   worker2
fleetman-web-app-79dfc474bf-9wvd9        1/1     Running   0          44s   10.244.1.230   worker1
fleetman-web-app-79dfc474bf-zjglj        1/1     Running   0          44s   10.244.2.191   worker2
```

Une fois l'application déployée sur le cluster, vous devriez être en mesure d'y accéder via : **http://<IP-NODE>:30080**

Name	Last seen	Speed mph
City Truck	14:49:15	9
Dronfield Round	14:49:07	22
Dronfield Round B	14:49:12	10
Electronics Pcb Rush Deliveries	14:49:14	7
Express Delivery1	14:49:12	0
Factory Run A	14:49:13	7
Factory Run B	14:49:13	5
Factory Run C	14:49:16	12

## 5 – Sources et références

- **Installation Kubernetes** : <https://discuss.kubernetes.io/t/building-a-kubernetes-1-27-cluster-with-kubeadm/25585>
- **Installation d'un cluster** :
  - <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/>
  - <https://discuss.kubernetes.io/t/building-a-kubernetes-1-27-cluster-with-kubeadm/25585>
- **Flannel** : <https://github.com/flannel-io/flannel/>
- **Containerd** :
  - <https://www.nocentino.com/posts/2021-12-27-installing-and-configuring-containerd-as-a-kubernetes-container-runtime/>
  - <https://docs.docker.com/engine/install/debian/>