

**Hasna Daoui**

**Diawara Nana**



---

*Projet 3-p2 : Script Simple d'Installation Kubernetes*

---

## I. INTRODUCTION

Dans le contexte actuel d'augmentation des volumes de données et de la complexité des applications, l'orchestration de conteneurs et la gestion efficace des ressources de calcul et de stockage sont devenues essentielles. Kubernetes s'est imposé comme la solution de référence pour automatiser le déploiement, la mise à l'échelle et la gestion d'applications conteneurisées. Toutefois, afin de répondre à des besoins spécifiques comme le **machine learning** ou le **stockage distribué**, des extensions telles que **Kubeflow** et **Rook Ceph** viennent enrichir l'écosystème Kubernetes.

Ce rapport présente une solution complète, robuste et automatisée pour le déploiement d'un cluster Kubernetes avec intégration de **Kubeflow** pour les workflows de machine learning, et de **Rook Ceph** pour une gestion dynamique du stockage persistant. L'objectif est de permettre une installation reproductible, sécurisée et efficace, tout en réduisant les interventions manuelles.

Dans ce cadre, nous proposons :

- L'installation automatisée des dépendances critiques (Docker, kubeadm, Containerd, etc.),
- La configuration du **nœud maître** via `kubeadm init`,
- La génération d'un **token sécurisé** pour permettre aux nœuds **workers** de rejoindre le cluster,
- L'installation du **CNI (Calico)** pour le réseau inter-pods,
- Le déploiement de **Rook Ceph** pour mettre en place une solution de stockage distribuée performante,
- Et l'installation de **Kubeflow**, une plateforme MLOps complète pour la gestion des workflows IA.

Un script Bash complet et interactif a été conçu pour automatiser toutes ces étapes, incluant des mécanismes de vérification d'état, de gestion des erreurs, et de résilience. Ce rapport s'adresse aussi bien aux ingénieurs systèmes qu'aux data scientists souhaitant mettre en place un environnement de calcul évolutif et intégré.

## II. Structure des fichiers

```
└─ installmaster.sh #Sur le nœud master
└─ installworker.sh #Sur le(s) nœud(s) worker
└─ "install_$(date +%Y%m%d_%H%M%S').log" #fichier log horodatée
```

## Configuration réseau

#Le nœud master doit être en mode Bridge

#Rajouter une partition de disque avant de lancer le script

### III. Exécution

#Sur la machine master

bash

sudo nano installmaster.sh

sudo chmod +x installmaster.sh

./installmaster.sh

#Sur la machine worker

Bash

sudo nano installworker.sh

sudo chmod +x installworker.sh

./installworker.sh

### IV. Dépannage

#### Erreurs communes

#### 1. ✗ Erreur : "swap is enabled" lors de `kubeadm init`

**Cause :** Kubernetes ne fonctionne pas avec la mémoire swap activée.

**Solution :**

```
sudo swapoff -a
```

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

Puis relancer la commande `kubeadm init`.

#### 2. ✗ Erreur : "cgroup driver conflict" entre Docker et kubelet

**Symptôme :** kubelet plante, journal affiche : `cgroup driver mismatch: cgroupfs vs systemd`

**Solution :** Aligner le driver cgroup de Docker avec celui de kubelet (`systemd` recommandé) :

bash

CopierModifier

```
cat <<EOF | sudo tee /etc/docker/daemon.json
```

```
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

```
EOF
```

```
sudo systemctl restart docker
```

```
sudo systemctl restart kubelet
```

### 3. ✖ Pods bloqués en statut `pending`

**Cause possible :** CNI (réseau de pods) non installé ou mal configuré.

**Solution :** Vérifier l'installation de Calico ou autre CNI :

```
bash
CopierModifier
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
```

### 4. ✖ Erreur dans KubeFlow UI : "Unable to fetch pipeline runs"

**Cause :** Le service `ml-pipeline-ui` ne peut pas se connecter à l'API.

**Solution :**

- Vérifier les logs :

```
bash
CopierModifier
kubectl logs -n kubeflow deployment/ml-pipeline-ui
```

- Vérifier les services avec :

```
bash
CopierModifier
kubectl get svc -n kubeflow
```

- Vérifier que `istio-ingressgateway` ou `ambassador` est bien configuré pour accéder au dashboard.

### 5. ✖ Rook Ceph : `CrashLoopBackOff` des pods `ceph-mon` ou `ceph-osd`

**Cause fréquente :** Pas assez de volumes libres / Pas de permissions sur les volumes.

**Solution :**

- S'assurer qu'un ou plusieurs disques libres sont disponibles (pas utilisés par /).
- Vérifier les logs :

```
bash
CopierModifier
kubectl -n rook-ceph logs <pod-name>
```

### 6. ✖ "Connection refused" sur `kubectl port-forward`

**Cause :** Le port n'est pas exposé ou le pod ciblé est tombé.

**Solution :**

- S'assurer que le pod est `Running`
- Vérifier les ports définis dans le `Deployment` ou `Service`

- Exemple de port-forward valide :

```
bash
CopierModifier
kubectl port-forward svc/ml-pipeline-ui -n kubeflow 8080:80
```

## V. CONCLUSION

La mise en œuvre d'une infrastructure moderne basée sur **Kubernetes**, enrichie par **Kubeflow** pour l'orchestration des workflows d'apprentissage automatique, et **Rook Ceph** pour le stockage distribué, représente une avancée significative vers l'automatisation, la scalabilité et la résilience des environnements cloud-native. Ce rapport a permis de détailler non seulement les concepts clés derrière ces technologies, mais aussi de proposer un **guide de déploiement pas à pas**, facilitant leur intégration sur des clusters réels.