



الجامعة الأورومتوسطية بفاس

EUROMED UNIVERSITY OF FES

UNIVERSITÉ EUROMED DE FÈS



École d'Ingénierie Digitale
et d'Intelligence Artificielle

3ème année en Cybersécurité

2024

Configuration et déploiement de JUPYTERHUB



Préparé par :

Wissal BOUTAYEB

Fatima BOUYAR MANE

Aya LOTFI

Hajar TAIFI BERNOUSSI

Encadré par :

M. AMAMOU AHMED

Sommaire

| | |
|--|--|
| I. Introduction..... | |
| a. Contexte et objectifs du projet..... | |
| b. Présentation générale des technologies utilisées..... | |
| II. Infrastructure et Environnement..... | |
| a. Description des machines virtuelles (master et workers) | |
| b. Configuration réseau et sécurité..... | |
| c. Les partitions de stockage créer | |
| III. Installation et Configuration de Docker 20..... | |
| a. Processus d'installation de Docker..... | |
| b. Configuration des environnements Docker sur chaque machine..... | |
| IV. Déploiement de Kubernetes 1.25..... | |
| a. Installation de Kubernetes sur le cluster..... | |
| b. Configuration du cluster Kubernetes..... | |
| c. Gestion des nœuds (master et workers). | |
| V. Mise en Place de Rook Ceph (dernier version) | |
| a. Installation de Rook Ceph..... | |
| b. Configuration de Ceph pour le stockage persistant..... | |
| VI Installation et Configuration de Kubeflow 1.7..... | |
| a. Processus d'installation de Kubeflow et du container sur le cluster Kubernetes..... | |
| VII. Déploiement de JupyterHub..... | |
| a. Installation de JupyterHub sur Kubeflow..... | |
| b. Configuration des utilisateurs et des espaces de travail..... | |
| c. Gestion des ressources et de la scalabilité..... | |
| VIII. Conclusion et Perspectives..... | |

I. Introduction

Contexte et Objectifs du Projet :

Le projet consiste à déployer **JupyterHub** sur un cluster **Kubernetes** pour offrir un environnement de calcul interactif et collaboratif aux utilisateurs. Les objectifs principaux sont de garantir la scalabilité, la haute disponibilité et une gestion efficace des ressources. En intégrant des technologies comme **Docker**, **Kubernetes**, **Rook Ceph** et **Kubeflow**, le projet vise à créer une infrastructure robuste et performante pour la recherche, l'enseignement et les projets d'apprentissage automatique.

Présentation Générale des Technologies Utilisées :

Docker : Plateforme de conteneurisation qui permet d'empaqueter les applications et leurs dépendances dans des conteneurs portables et isolés.

Kubernetes : Système de gestion de conteneurs qui orchestre l'exécution et la mise à l'échelle des applications conteneurisées sur un cluster de machines.

Rook Ceph : Orchestrateur de stockage persistant utilisant Ceph pour fournir un stockage distribué, résilient et hautement disponible.

Kubeflow : Plateforme (**Framework Open Source**) pour l'automatisation des workflows d'apprentissage automatique sur Kubernetes, facilitant le déploiement et la gestion des modèles ML.

JupyterHub : Service multi-utilisateurs permettant d'exécuter des notebooks Jupyter dans un environnement partagé, idéal pour la collaboration et l'enseignement.

II. Infrastructure et Environnement

1- Description des Machines Virtuelles (Master et Workers)

1.1 Machine Virtuelle Master :

Rôle : Cette VM contrôle l'ensemble du cluster Kubernetes, gérant la planification des pods, la maintenance de l'état du cluster et la coordination des nœuds worker.

Configuration matérielle :

CPU : 4 vCPU

Mémoire : 16 Go RAM

Stockage : 200 Go

Système d'exploitation : Ubuntu 20.04 LTS

1.2 Machines Virtuelles Workers :

Rôle : Ces VMs exécutent les pods et les conteneurs, gérant les charges de travail assignées par le master.

Configuration matérielle:

CPU : 4 vCPU (chaque)

Mémoire : 8 Go RAM (chaque)

Stockage : 150 Go (chaque)
Système d'exploitation : Ubuntu 20.04 LTS

2- Configuration Réseau et Sécurité :

1. Réseau :

- **Architecture réseau** Un réseau privé virtuel (VPN) connecte toutes les machines virtuelles pour assurer la communication sécurisée.

Adresses IP:

- **Master** : 192.168.244.156
- **Worker 1** : 192.168.244.157
- **Worker 2** : 192.168.244.158

DNS : Configuration de DNS interne pour la résolution de noms des services Kubernetes .

Le fichier **/etc/hosts** est configurer comme la suite dans les machines **master et workers**

```
192.168.244.156 master_node
192.168.244.157 worker1_node
192.168.244.158 worker2_node
```

Sécurité :

- **Pare-feu** : Mise en place de règles de pare-feu pour restreindre l'accès aux ports nécessaires uniquement (par exemple, port 6443 pour l'API Kubernetes, ports 10250-10255 pour la communication entre les nœuds).
- **SSL/TLS** : Utilisation de certificats SSL/TLS pour sécuriser les communications entre les composants du cluster.

Les Partitions de Stockage Crées:

1-Stockage Local:

- **Master :**
 - **/**: Partition principale pour le système d'exploitation et les applications (50 Go).
 - **/var/lib/docker** : Partition dédiée pour Docker (50 Go).

- **/var/lib/kubelet** : Partition dédiée pour Kubernetes (100 Go).
- **Workers**:
 - **/**: Partition principale pour le système d'exploitation et les applications (50 Go).
 - **/var/lib/docker** : Partition dédiée pour Docker (50 Go).
 - **/var/lib/kubelet** : Partition dédiée pour Kubernetes (50 Go).

2- Stockage Persistant (Rook Ceph):

- **Pools de stockage** : Création de pools de stockage Ceph pour différents types de données (par exemple, stockage de blocs, objets et fichiers).
- **Volumes persistants** : Définition de **PersistentVolumeClaims (PVC)** pour les applications nécessitant un stockage persistant.
- **Répartition des données** : Utilisation de la réplication et de l'érasure coding pour garantir la résilience des données.
- **Monitoring et Gestion** : Mise en place d'outils de monitoring (par exemple, Ceph Dashboard) pour surveiller l'état du stockage et effectuer des ajustements si nécessaire.

III .Installation et Configuration de Docker 20 dans Ubuntu 20.04 (Dans chacun des Machines Master et workers):

Cette commande installe des dépendances pour s'assurer que la commande apt transfère les fichiers de manière sécurisée via https

```
master@ubuntu:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Il s'agit d'une fonctionnalité de sécurité pour garantir l'authenticité des fichiers d'installation.

```
master@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
master@ubuntu:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:4 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [43.5 kB]
```

Installer les dépôts Docker Ajouter le dépôt Docker aux sources apt en utilisant la commande

```
master@ubuntu:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
```

Mettre à jour les dépôts Utilisons la commande ci-dessous pour mettre à jour les dépôts ajoutés dans la commande précédente

```
master@ubuntu:~$ sudo apt install docker-ce=5:20.10.8~3-0~ubuntu-focal docker-ce-cli=5:20.10.8~3-0~ubuntu-focal cont
atnerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
Suggested packages:
```

La commande installe la version 20 de Docker

```
master@ubuntu:~$ sudo systemctl start docker
```

Pour Démarrer le service Docker sur notre machine. Cette commande permet de lancer le démon Docker, qui est responsable de la gestion et de l'exécution des conteneurs Docker.

```
master@ubuntu:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
```

Configurer le service Docker pour qu'il démarre automatiquement à chaque démarrage de la machine.

```
master@ubuntu:~$ systemctl restart docker
master@ubuntu:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-05-23 07:33:35 PDT; 7s ago
    TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 19162 (dockerd)
        Tasks: 10
       Memory: 28.9M
      CGroup: /system.slice/docker.service
              └─19162 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

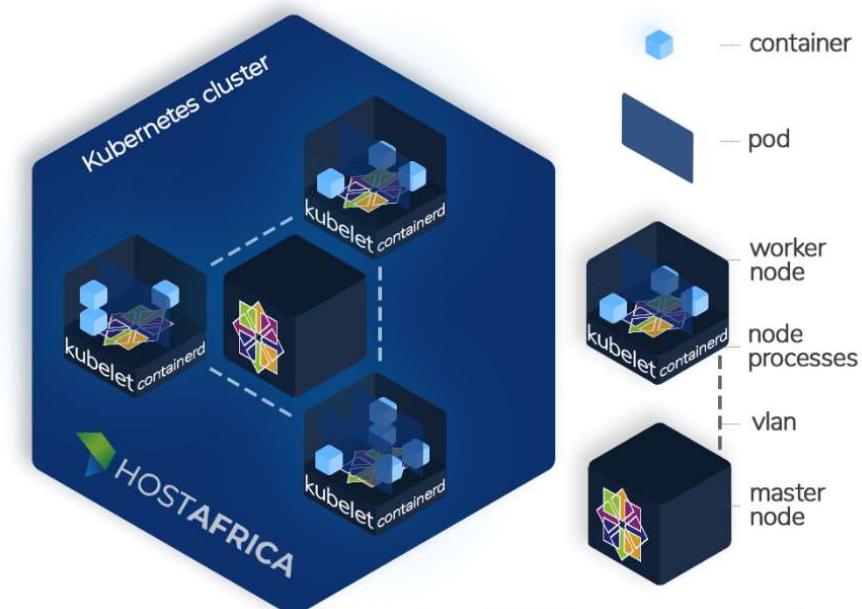
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.436293542-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.437660099-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.549475712-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.549782910-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.699242233-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.766961557-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.793169713-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.793292366-07:00" level=info
May 23 07:33:35 ubuntu dockerd[19162]: time="2024-05-23T07:33:35.823905926-07:00" level=info
May 23 07:33:35 ubuntu systemd[1]: Started Docker Application Container Engine.
```

La sortie indique que le service Docker est chargé et actif, ce qui indique une installation réussie.

```
master@ubuntu:~$ docker --version
Docker version 20.10.8, build 3967b7d
master@ubuntu:~$
```

Cette commande nous affiche la version du Docker

IV. Déploiement de Kubernetes 1.25:



1-Configuration du Master :

```
master@ubuntu:~$ wget https://dl.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz
--2024-05-22 08:34:24--  https://dl.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz
Resolving dl.k8s.io (dl.k8s.io)... 34.107.204.206, 2600:1901:0:26f3:::
Connecting to dl.k8s.io (dl.k8s.io)|34.107.204.206|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.dl.k8s.io/release/v1.25.0/kubernetes-server-linux-amd64.tar.gz [following]
--2024-05-22 08:34:24--  https://cdn.dl.k8s.io/release/v1.25.0/kubernetes-server-linux-amd64.tar.gz
Resolving cdn.dl.k8s.io (cdn.dl.k8s.io)... 146.75.53.55, 2a04:4e42:7d::311
Connecting to cdn.dl.k8s.io (cdn.dl.k8s.io)|146.75.53.55|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 334115819 (319M) [application/x-tar]
Saving to: 'kubernetes-server-linux-amd64.tar.gz'

kubernetes-server-linux-amd64 100%[=====] 318.64M  819KB/s   in 4m 29s

2024-05-22 08:38:53 (1.18 MB/s) - 'kubernetes-server-linux-amd64.tar.gz' saved [334115819/334115819]
```

télécharge le fichier kubernetes-server-linux-amd64.tar.gz depuis l'URL <https://dl.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz>. Dans la machine master

```
master@ubuntu:~$ tar -xvf kubernetes-server-linux-amd64.tar.gz
kubernetes/
kubernetes/kubernetes-src.tar.gz
kubernetes/addons/
kubernetes/server/
kubernetes/server/bin/
kubernetes/server/bin/kube-scheduler.tar
kubernetes/server/bin/kube-controller-manager.docker_tag
kubernetes/server/bin/kube-proxy
kubernetes/server/bin/kube-proxy.docker_tag
kubernetes/server/bin/kube-apiserver.docker_tag
kubernetes/server/bin/kubeadm
kubernetes/server/bin/kube-aggregator
kubernetes/server/bin/kube-controller-manager.tar
kubernetes/server/bin/kube-proxy.tar
kubernetes/server/bin/mounter
kubernetes/server/bin/apiextensions-apiserver
```

Utilisée pour extraire le fichier téléchargé.

```
master@ubuntu:~$ sudo cp $(find ./kubernetes/server/bin -maxdepth 1 -type f | sed 's/^\.\/\///' | grep -v "\.") /usr/b
in/
[sudo] password for master:
```

copier les fichiers binaires de Kubernetes dans un emplacement accessible par le système

```
master@ubuntu:~$ sudo mkdir /etc/systemd/system/kubelet.service.d
master@ubuntu:~$ cd /etc/systemd/system/kubelet.service.d
master@ubuntu:/etc/systemd/system/kubelet.service.d$ sudo nano /etc/systemd/sys
f
```

Ce fichier de configuration peut être utilisé pour personnaliser le comportement du service kubelet , par exemple en spécifiant des paramètres spécifiques au déploiement Kubernetes.

```

master@ubuntu:~$ sudo nano /etc/systemd/system/kubelet.service
[Service]
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/config --config=/var/lib/kubelet/config.yaml"
EnvironmentFile=/var/lib/kubelet/kubeadm-flags.env
EnvironmentFile=/etc/default/kubelet
ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS $KUBELET_CONFIG_ARGS $KUBELET_KUBEADM_ARGS $KUBELET_EXTRA_ARGS

master@ubuntu:~$ sudo nano /usr/lib/systemd/system/kubelet.service
[Unit]
Description=kubelet: The Kubernetes Node Agent
Documentation=https://kubernetes.io/docs/home/
Wants=network-online.target
After=network-online.target
[Service]
ExecStart=/usr/bin/kubelet
Restart=always
StartLimitInterval=0
RestartSec=10
[Install]
WantedBy=multi-user.target

```

Cette fonction **WantedBy=multi-user.target** spécifie quand le service kubelet doit être activé. Plus précisément, cela indique à systemd d'activer le service kubelet lorsqu'il entre dans le niveau d'exécution multi-utilisateurs. Le niveau d'exécution multi-utilisateurs est un niveau d'exécution qui fournit un environnement multi-utilisateurs complet avec toutes les ressources nécessaires, y compris le réseau. Cela garantit que le **kubelet** est lancé lorsque le système est prêt pour un fonctionnement multi-utilisateurs complet, ce qui est souvent nécessaire pour les clusters **Kubernetes** fonctionnant dans des environnements de production.

active le service **kubelet** de manière persistante. Cela signifie que le service sera automatiquement démarré au démarrage du système. Cette commande crée des liens **symboliques** appropriés pour activer le service kubelet,

```

master@ubuntu:~$ sudo systemctl daemon-reload
master@ubuntu:~$ sudo systemctl start kubelet.service
master@ubuntu:~$ sudo systemctl enable kubelet.service
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
master@ubuntu:~$ wget https://github.com/containerNetworking/plugins/releases/download/v1.0.0/cni-plugins-linux-amd64-v1.0.0.tgz
--2024-05-22 08:49:29-- https://github.com/containerNetworking/plugins/releases/download/v1.0.0/cni-plugins-linux-amd64-v1.0.0.tgz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/84575398/4ec3d785-a8dc-43d9-9568-2240054a337d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240522%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240522T154929Z&X-Amz-Expires=300&X-Amz-Signature=4e2a413d4d04b3864b44846449bcebe2a9702e66bcd276b3602ed857a7183426&X-Amz-SignedHeaders=host&actor_id=0&key_id=84575398&response-content-disposition=attachment%20filename%3Dcni-plugins-linux-amd64-v1.0.0.tgz&response-content-type=application%2Foctet-stream [f
ollowing]

```

Télécharger l'archive

```

master@ubuntu:~$ sudo mkdir -p /opt/cni/bin/
master@ubuntu:~$ sudo tar -xvf ./cni-plugins-linux-amd64-v1.0.0.tgz --directory /opt/cni/bin/
./
./macvlan
./static
./vlan
./portmap
./host-local
./vrf
./bridge
./tuning
./firewall
./host-device
./sbr
./loopback
./dhcp
./ptp
./ipvlan

```

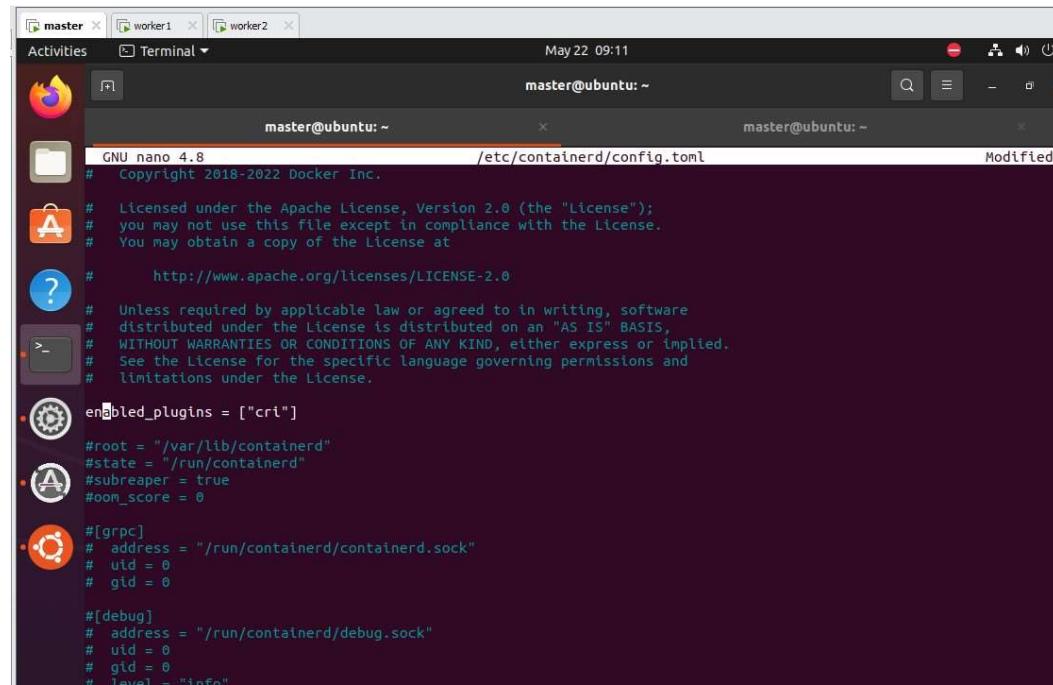
Extraire le fichier téléchargé dans le dossier `/opt/cni/bin/`

```

master@ubuntu:~$ sudo kubeadm init --kubernetes-version=v1.25.0 --pod-network-cidr=10.244.0.0/16
[kubelet] Using Kubernetes version: v1.25.0
[preflight] Running pre-flight checks
    [WARNING FileExisting-socat]: socat not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
    [ERROR CRI]: container runtime is not running: output: time="2024-05-22T09:05:24-07:00" level=fatal msg="getting status of runtime: rpc error: code = Unimplemented desc = unknown service runtime.v1alpha2.RuntimeService"
, error: exit status 1
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher

```

Initialisation de notre Cluster



```

master@ubuntu:~$ nano /etc/containerd/config.toml
# Copyright 2018-2022 Docker Inc.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at

#     http://www.apache.org/licenses/LICENSE-2.0

# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

enabled_plugins = ["cri"]

#root = "/var/lib/containerd"
#state = "/run/containerd"
#subreaper = true
#oom_score = 0

#[grpc]
# address = "/run/containerd/containerd.sock"
# uid = 0
# gid = 0

#[debug]
# address = "/run/containerd/debug.sock"
# uid = 0
# gid = 0
# level = "info"

```

```

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

```

Créer le dossier `$HOME/.kube` et copier le fichier `admin.conf` dans le `$HOME/.kube/config`

```
y
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!
```

To start using your cluster, you need to run the following as a regular user:

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

Alternatively, if you are the root user, you can run:

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 192.168.238.165:6443 --token mxycwk.qv5yix3cdgpxil3a \
--discovery-token-ca-cert-hash sha256:52d38836c0fabbff2883f0f216bf6811b90bbd7a42f441a6cf5b8466259d7e94
master@ubuntu:~$ [preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
^C
master@ubuntu:~$ sudo hostnamectl set-hostname "master-node"
master@ubuntu:~$ hostname
master-node
master@ubuntu:~$ exec bash
```

```
[preflight] This might take a minute or two, depending on the
[preflight] You can also perform this action in beforehand usi
^C
master@ubuntu:~$ sudo hostnamectl set-hostname "master-node"
master@ubuntu:~$ hostname
master-node
master@ubuntu:~$ exec bash
```

Cette ligne de commande est utilisée pour joindre les deux machines workers dans le même cluster

2- Configuration des workers (en suivants les mêmes Étapes de configuration du Master):

```
w1@ubuntu:~$ tar -xvf kubernetes-client-linux-amd64.tar.gz
kubernetes/
kubernetes/client/
kubernetes/client/bin/
kubernetes/client/bin/kubectl-convert
kubernetes/client/bin/kubectl
w1@ubuntu:~$ sudo cp `find ./kubernetes/client/bin -maxdepth 1 -type f | sed 's/^\.\/\///' | grep -v "\."` /usr/bi
n/
[sudo] password for w1:
w1@ubuntu:~$
```

```

worker2 - VMware Workstation
File Edit View VM Tabs Help || Library Activities Terminal ~ May 22 09:28
w2@ubuntu: ~
curl -s https://dl.k8s.io/v1.25.0/kubeadm.sh | sh
curl -s https://kubernetes.io/.../v1.25.0/kubelet.tar.gz | tar -xvf
curl -s https://kubernetes.io/.../v1.25.0/kubelet.tar.gz | tar -xvf
curl -s https://kubernetes.io/.../v1.25.0/kubelet.tar.gz | tar -xvf

```

```

w1@ubuntu:~$ wget https://dl.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz
--2024-05-22 09:39:03-- https://dl.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz
Resolving dl.k8s.io (dl.k8s.io)... 34.107.204.206, 2009:1901:0:26f3::1
Connecting to dl.k8s.io (dl.k8s.io)|34.107.204.206|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz
--2024-05-22 09:39:04-- https://cdn.k8s.io/v1.25.0/kubernetes-server-linux-amd64.tar.gz
Resolving cdn.dl.k8s.io (cdn.dl.k8s.io)... 151.101.93.55, 151.101.129.55, 151.101.05.55, ...
Connecting to cdn.dl.k8s.io (cdn.dl.k8s.io)|151.101.193.55|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 334115819 (319MB) [application/x-tar]
Saving to: "kubernetes-server-linux-amd64.tar.gz"

kubernetes-server-linux-amd64.tar.gz          100%[=====] 318.64M  2.58MB/s   in 2m 51s

2024-05-22 09:41:59 (1.86 MB/s) - "kubernetes-server-linux-amd64.tar.gz" saved [334115819/334115819]

w1@ubuntu:~$ tar -xvf kubernetes-server-linux-amd64.tar.gz
kubernetes/
kubernetes/kubernetes-.src.tar.gz
kubernetes/addons/
kubernetes/server/
kubernetes/server/btn/
kubernetes/server/btn/kube-scheduler.tar
kubernetes/server/btn/kube-controller-manager.docker_tag

```

```

w1@ubuntu:~$ 
w1@ubuntu:~$ sudo systemctl start kubelet.service
w1@ubuntu:~$ sudo systemctl enable kubelet.service

```

```

master@master-node:~ $ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
master-node  NotReady control-plane  4m25s  v1.25.0
worker1-node  NotReady <none>    83s    v1.25.0
master@master-node:~ $ kubectl apply -f https://github.com/coreos/flannel/raw/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
master@master-node:~ $ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
master-node  NotReady control-plane  5m18s  v1.25.0
worker1-node  NotReady <none>    2m16s  v1.25.0
master@master-node:~ $ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
master-node  Ready    control-plane  6m41s  v1.25.0
worker1-node  Ready    <none>    3m39s  v1.25.0
master@master-node:~ $ kubectl label node worker-node node-role.kubernetes.io/worker=worker
Error from server (NotFound): nodes "worker-node" not found
master@master-node:~ $ kubectl label node worker1-node node-role.kubernetes.io/worker=worker
node/worker1-node labeled
master@master-node:~ $ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
master-node  Ready    control-plane  7m26s  v1.25.0
worker1-node  Ready    worker     4m24s  v1.25.0
master@master-node:~ $

```

Dans la machine master on voit clairement que les deux workers ont rejoint notre cluster avec le statut de prêt.

```

master@ubuntu:~$ sudo vt /etc/contalnerd/config.toml
[sudo] password for master:
master@ubuntu:~$ sudo nano /etc/contalnerd/config.toml
[sudo] password for master:
master@ubuntu:~$ sudo systemctl restart contalnerd
master@ubuntu:~$ sudo systemctl status contalnerd
● contalnerd.service - contalnerd container runtime
   Loaded: loaded (/lib/systemd/system/contalnerd.service; enabled; vendor pr...
   Active: active (running) since Thu 2024-05-23 07:17:13 PDT; 7s ago
     Docs: https://contalnerd.io
   Process: 7877 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/S...
 Main PID: 7878 (contalnerd)
   Tasks: 10
    Memory: 13.5M
      CGroup: /system.slice/contalnerd.service
           └─7878 /usr/bin/contalnerd

May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660194152-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660255628-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660293814-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660329184-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660379205-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660390261-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660397748-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660404368-07...
May 23 07:17:13 ubuntu contalnerd[7878]: time="2024-05-23T07:17:13.660461676-07...
May 23 07:17:13 ubuntu systemd[1]: Started contalnerd container runtime.

master@ubuntu:~$ ps -ef | grep contalnerd
root      7878      1  0 07:17 ?        00:00:00 /usr/bin/contalnerd
master     7895    7844  0 07:17 pts/0    00:00:00 grep --color=auto contalnerd
master@ubuntu:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

```

On voit clairement ici le Statut de notre container qui est activer

```

master@ubuntu:~$ sudo sed -ri '/sswap\s/s/^#?/#/' /etc/fstab
master@ubuntu:~$ sudo swapoff -a
master@ubuntu:~$ 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
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
master@ubuntu:~$ export VERSION = "1.25.0-00"
bash: export: `=': not a valid identifier
bash: export: `1.25.0-00': not a valid identifier

```

Cette commande est utilisée pour désactiver le Swappe et aussi pour configurer notre fichier **kubernetes.conf**

```

master@ubuntu:~$ systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/kubelet.service.d
     └─10-kubeadm.conf
     Active: activating (auto-restart) (Result: exit-code) since Thu 2024-05-23 07:38:09 P
       Docs: https://kubernetes.io/docs/home/
     Main PID: 19381 (code=exited, status=1/FAILURE)
       Tasks: 0 (limit: 19050)
      Memory: 0B
      CGroup: /system.slice/kubelet.service

swap[...]. 0 swap[...]. failed, no such file or directory
master@master-node:~$ sudo sed -i '/ swap / s/^(\.*$)/#\1/g' /etc/fstab
master@master-node:~$ sudo kubeadm config images pull

I0523 07:51:36.398982    20851 version.go:256] remote version is much newer: v1.30.1; falling ba
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.25.16
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.25.16
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.25.16
[config/images] Pulled registry.k8s.io/kube-proxy:v1.25.16
[config/images] Pulled registry.k8s.io/pause:3.8
[config/images] Pulled registry.k8s.io/etcd:3.5.4-0
[config/images] Pulled registry.k8s.io/coredns/coredns:v1.9.3
master@master-node:~$ 
master@master-node:~$ 

```

V. Mise en Place de Rook Ceph (dernier version):

```

master@master-node:~$ git clone --single-branch --branch master https://github.com/rook/rook.git
Cloning into 'rook'...
remote: Enumerating objects: 98766, done.
remote: Counting objects: 100% (5342/5342), done.
remote: Compressing objects: 100% (455/455), done.
remote: Total 98766 (delta 5017), reused 4987 (delta 4883), pack-reused 93424
Receiving objects: 100% (98766/98766), 52.40 MiB | 1.79 MiB/s, done.
Resolving deltas: 100% (69335/69335), done.
master@master-node:~$ cd rook/cluster/examples/kubernetes/ceph
bash: cd: rook/cluster/examples/kubernetes/ceph: No such file or directory
master@master-node:~$ sudo mkdir rook/cluster/examples/kubernetes/ceph
[sudo] password for master:
mkdir: cannot create directory 'rook/cluster/examples/kubernetes/ceph': No such file or directory
master@master-node:~$ sudo mkdir rook/cluster/examples/kubernetes
mkdir: cannot create directory 'rook/cluster/examples/kubernetes': No such file or directory
master@master-node:~$ sudo mkdir rook
mkdir: cannot create directory 'rook': File exists
master@master-node:~$ sudo mkdir rook/
mkdir: cannot create directory 'rook/': File exists
master@master-node:~$ cd rook/cluster/
bash: cd: rook/cluster/: No such file or directory
master@master-node:~$ cd rook/cluster
bash: cd: rook/cluster: No such file or directory
master@master-node:~$ cd rook
master@master-node:~/rook$ ls
ADOPTERS.md          CONTRIBUTING.md  go.mod        LICENSE           pkg
build                DCO                 go.sum        Makefile          README.md
cmd                  deploy              GOVERNANCE.md  mkdocs.yml      ROADMAP.md
CODE_OF_CONDUCT.md   design              images        OWNERS.md       SECURITY.md
CODE-OWNERS          Documentation      INSTALL.md    PendingReleaseNotes.md tests
master@master-node:~/rook$ cd cluster/examples/kubernetes/ceph
bash: cd: cluster/examples/kubernetes/ceph: No such file or directory
master@master-node:~/rook$ cd deploy/examples/kubernetes/ceph
bash: cd: deploy/examples/kubernetes/ceph: No such file or directory
master@master-node:~/rook$ cd deploy
master@master-node:~/rook/deploy$ ls
charts  examples  olm

```

Télécharger depuis ce lien de GitHub la dernière version de **Rook**

```

bash: cd: deploy/examples/kubernetes/ceph: No such file or directory
master@master-node:~/rook$ cd deploy
master@master-node:~/rook/deploy$ ls
charts examples olm
master@master-node:~/rook/deploy$ cd examples
master@master-node:~/rook/deploy/examples$ ls
bucket-notification-endpoint.yaml          object-bucket-claim-a.yaml
bucket-notification.yaml                   object-bucket-claim-delete.yaml
bucket-topic.yaml                         object-bucket-claim-notification.yaml
ceph-client.yaml                          object-bucket-claim-retain.yaml
cluster-external-management.yaml          object-b.yaml
cluster-external.yaml                     object-ec.yaml
cluster-multus-test.yaml                 object-external.yaml
cluster-on-local-pvc.yaml                object-multisite-pull-realm-test.yaml
cluster-on-pvc-minikube.yaml             object-multisite-pull-realm.yaml
cluster-on-pvc.yaml                      object-multisite-test.yaml
cluster-stretched-aws.yaml               object-multisite.yaml
cluster-stretched.yaml                  object-openshift.yaml
cluster-test.yaml                        object-shared-pools-test.yaml
cluster.yaml                            object-shared-pools.yaml
common-external.yaml                    object-test.yaml
common-second-cluster.yaml              object-user.yaml
common.yaml                            object.yaml
cosi                                    operator-openshift.yaml
crds.yaml                             operator.yaml
create-external-cluster-resources.py    osd-env-override.yaml
create-external-cluster-resources-tests.py osd-purge.yaml
csi                                    pool-built-in-mgr.yaml
csi-ceph-conf-override.yaml            pool-ec.yaml
dashboard-external-https.yaml          pool-mirrored.yaml
dashboard-external-http.yaml           pool-test.yaml
dashboard-ingress-https.yaml          pool.yaml
dashboard-loadbalancer.yaml           psp.yaml
direct-mount.yaml                     radosnamespace.yaml
external                                rbdmirror.yaml
filesystem-ec.yaml                     README.md
filesystem-mirror.yaml

```

L'accès au dossier /rook/deploy/examples

```

master@master-node:~      master@master-node:~      master@master-node:~/rook/depl...
...-volume-balancer.yaml   volume-replication-class.yaml
nfs-test.yaml              volume-replication.yaml
nfs.yaml                  wordpress.yaml

master@master-node:~/rook/deploy/examples$ kubectl apply -f crds.yaml
customresourcedefinition.apiextensions.k8s.io/cephblockpoolradosnamespaces.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephblockpools.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephbucketnotifications.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephbuckettopics.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephclients.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephclusters.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephcosiddrivers.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephfilesystemmirrors.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephfilesystems.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephfilesystemsubvolumegroups.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephfileses.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephobjectrealms.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephobjectstores.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephobjectstoreusers.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephobjectzonegroups.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephobjectzones.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/cephrbdmirrors.ceph.rook.io created
customresourcedefinition.apiextensions.k8s.io/objectbucketclaims.objectbucket.io created
customresourcedefinition.apiextensions.k8s.io/objectbuckets.objectbucket.io created
master@master-node:~/rook/deploy/examples$ kubectl apply -f common.yaml
namespace/rook-ceph created
clusterrole.rbac.authorization.k8s.io/cephfs-csi-nodeplugin created
clusterrole.rbac.authorization.k8s.io/cephfs-external-provisioner-runner created
clusterrole.rbac.authorization.k8s.io/objectstorage-provisioner-role created
clusterrole.rbac.authorization.k8s.io/rbd-csi-nodeplugin created
clusterrole.rbac.authorization.k8s.io/rbd-external-provisioner-runner created
clusterrole.rbac.authorization.k8s.io/rook-ceph-cluster-mgmt created
clusterrole.rbac.authorization.k8s.io/rook-ceph-global created
clusterrole.rbac.authorization.k8s.io/rook-ceph-mgr-cluster created
clusterrole.rbac.authorization.k8s.io/rook-ceph-mgr-system created
clusterrole.rbac.authorization.k8s.io/rook-ceph-object-bucket created
clusterrole.rbac.authorization.k8s.io/rook-ceph-osd created
clusterrole.rbac.authorization.k8s.io/rook-ceph-system created

```

kubectl apply -f crds.yaml, Kubernetes ajoute ces définitions personnalisées à son API, permettant ainsi aux utilisateurs de créer des instances de ces ressources personnalisées.

kubectl apply -f common.yaml: Cette commande est utilisée pour créer ou mettre à jour des ressources Kubernetes à partir d'un fichier YAML nommé common.yaml. Ce fichier contient généralement des définitions de ressources Kubernetes

```

serviceaccount/rook-csi-cephfs-provisioner-sa created
serviceaccount/rook-csi-rbd-plugin-sa created
serviceaccount/rook-csi-rbd-provisioner-sa created
master@master-node:~/rook/deploy/examples$ kubectl apply -f operator.yaml
configmap/rook-ceph-operator-config created
deployment.apps/rook-ceph-operator created
master@master-node:~/rook/deploy/examples$ kubectl apply -f crds.yaml
customresourcedefinition.apirextensions.k8s.io/cephblockpoolradosnamespaces.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephblockpools.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephbucketnotifications.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephbuckettopics.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephclients.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephclusters.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephcosldrivers.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephfilesystemmirrors.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephfilesystems.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephfilesystemsubvolumegroups.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephnfses.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephobjectrealms.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephobjectstores.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephobjectstorers.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephobjectzonegroups.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephobjectzones.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/cephrbdmirrors.ceph.rook.io unchanged
customresourcedefinition.apirextensions.k8s.io/objectbucketclaims.objectbucket.io unchanged
customresourcedefinition.apirextensions.k8s.io/objectbuckets.objectbucket.io unchanged
master@master-node:~/rook/deploy/examples$ kubectl create -f cluster.yaml
cephcluster.ceph.rook.io/rook-ceph created
master@master-node:~/rook/deploy/examples$ cd csi
master@master-node:~/rook/deploy/examples/csi$ cd rbd/
master@master-node:~/rook/deploy/examples/csi/rbd$ ls
pod-ephemeral.yaml pvc-restore.yaml snapshot.yaml storageclass.yaml
pod.yaml pvc.yaml storageclass-ec.yaml
pvc-clone.yaml snapshotclass.yaml storageclass-test.yaml
master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl create -f storageclass-test.yaml
cephblockpool.ceph.rook.io/replicapool created
storageclass.storage.k8s.io/rook-ceph-block created

```

On crée le fichier **storageclass-test.yaml** qui est utile pour configurer le stockage persistant dans le cluster

```

master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl create -f storageclass-test.yaml
cephblockpool.ceph.rook.io/replicapool created
storageclass.storage.k8s.io/rook-ceph-block created
master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl -n rook-ceph get pod -l app=rook-ceph-operator
NAME                               READY   STATUS    RESTARTS   AGE
rook-ceph-operator-6f67847bd-8qknk   1/1    Running   0          2m6s
master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl -n rook-ceph get pods -A
NAMESPACE      NAME                                         READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-4s88c                         1/1    Running   0          16m
kube-flannel   kube-flannel-ds-6k7fr                         1/1    Running   0          16m
kube-system    coredns-565d847f94-hxh7z                      1/1    Running   0          21m
kube-system    coredns-565d847f94-p4qc4                      1/1    Running   0          21m
kube-system    etcd-master-node                            1/1    Running   0          21m
kube-system    kube-apiserver-master-node                   1/1    Running   0          21m
kube-system    kube-controller-manager-master-node        1/1    Running   0          21m
kube-system    kube-proxy-vltwm                          1/1    Running   0          21m
kube-system    kube-proxy-xwddn                          1/1    Running   0          18m
kube-system    kube-scheduler-master-node                  1/1    Running   0          21m
rook-ceph      csi-cephfsplugin-p8nl2                      2/2    Running   0          2m6s
rook-ceph      csi-cephfsplugin-provisioner-64d7866486-ds225 0/5    Pending   0          2m6s
rook-ceph      csi-cephfsplugin-provisioner-64d7866486-jfq7p 5/5    Running   1 (28s ago) 2m6s
rook-ceph      csi-rbdplugin-gq8p4                        2/2    Running   0          2m6s
rook-ceph      csi-rbdplugin-provisioner-644468dd8c-flvs9  5/5    Running   1 (51s ago) 2m6s
rook-ceph      csi-rbdplugin-provisioner-644468dd8c-mp5ld  0/5    Pending   0          2m6s
rook-ceph      rook-ceph-operator-6f67847bd-8qknk         1/1    Running   0          3m36s
master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl taint nodes $(hostname) node-role.kubernetes.io/master:NoSchedule-
error: taint "node-role.kubernetes.io/master:NoSchedule" not found
master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl taint nodes master-node has-gpu:NoSchedule-
error: taint "has-gpu:NoSchedule" not found
master@master-node:~/rook/deploy/examples/csi/rbd$ 
master@master-node:~/rook/deploy/examples/csi/rbd$ kubectl get nodes
NAME      STATUS   ROLES   AGE   VERSION
master-node  Ready   control-plane  23m   v1.25.0
worker1-node Ready   worker     20m   v1.25.0
master@master-node:~/rook/deploy/examples/csi/rbd$ 
master@master-node:~/rook/deploy/examples/csi/rbd$ 
```

```

master@ubuntu: ~          master@master-node: ~          master@master-node: ~/rook/deploy/examples
serviceaccount/rook-csi-rbd-plugin-sa created
serviceaccount/rook-csi-rbd-provisioner-sa created
master@master-node:~/rook/deploy/examples$ kubectl apply -f operator.yaml
configmap/rook-ceph-operator-config created
deployment.apps/rook-ceph-operator created
master@master-node:~/rook/deploy/examples$ kubectl apply -f cluster.yaml
cephcluster.ceph.rook.io/rook-ceph created
master@master-node:~/rook/deploy/examples$ kubectl -n rook-ceph get pods
NAME           READY   STATUS    RESTARTS   AGE
csi-cephfsplugin-provisioner-75875b5887-djjwt  0/5    ContainerCreating   0      30s
csi-cephfsplugin-provisioner-75875b5887-jw9xx  0/5    Pending    0      30s
csi-cephfsplugin-rkhj9   0/2    ContainerCreating   0      30s
csi-rbdplugin-4cpbh  2/2    Running   0      30s
csi-rbdplugin-provisioner-56d69f5d8-8j4c2  0/5    ContainerCreating   0      30s
csi-rbdplugin-provisioner-56d69f5d8-hnr8t  0/5    Pending    0      30s
rook-ceph-operator-54774afbf-dct9b  1/1    Running   0      101s
rook-ceph-tools-5679b7d8f-cm5hg   0/1    ContainerCreating   0      46s
master@master-node:~/rook/deploy/examples$ kubectl apply -f csi/rbd/storageclass.yaml
cephblockpool.ceph.rook.io/replicapool created
storageclass.storage.k8s.io/rook-ceph-block created
master@master-node:~/rook/deploy/examples$ kubectl apply -f csi/cephfs/storageclass.yaml
storageclass.storage.k8s.io/rook-cephfs created
master@master-node:~/rook/deploy/examples$ kubectl create -f toolbox.yaml
deployment.apps/rook-ceph-tools created
master@master-node:~/rook/deploy/examples$ kubectl get pods
No resources found in default namespace.
master@master-node:~/rook/deploy/examples$ kubectl -n rook-ceph get pods
NAME           READY   STATUS    RESTARTS   AGE
csi-cephfsplugin-provisioner-75875b5887-djjwt  5/5    Running   0      2m36s
csi-cephfsplugin-provisioner-75875b5887-jw9xx  0/5    Pending    0      2m36s
csi-cephfsplugin-rkhj9   2/2    Running   0      2m36s
csi-rbdplugin-4cpbh  2/2    Running   0      2m36s
csi-rbdplugin-provisioner-56d69f5d8-8j4c2  5/5    Running   0      2m36s
csi-rbdplugin-provisioner-56d69f5d8-hnr8t  0/5    Pending    0      2m36s
rook-ceph-operator-54774afbf-dct9b  1/1    Running   0      3m47s
rook-ceph-tools-5679b7d8f-cm5hg   0/1    ContainerCreating   0      46s
master@master-node:~/rook/deploy/examples$ kubectl get storageclass
NAME        PROVISIONER   RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
rook-ceph-block   rook-ceph.rbd.csi.ceph.com   Delete   Immediate   true   114s
rook-cephfs   rook-ceph.cephfs.csi.ceph.com   Delete   Immediate   true   109s
master@master-node:~/rook/deploy/examples$ ls
bucket-notification-endpoint.yaml   direct-mount.yaml   operator-openshift.yaml
bucket-notification.yaml   filesystem-ec.yaml   operator.yaml
bucket-topic.yaml   filesystem-mirror.yaml   osd-env-override.yaml
ceph-client.yaml   filesystem-test.yaml   osd-purge.yaml
cluster-external-management.yaml   filesystem.yaml   pool-builtin-mgr.yaml
cluster-external.yaml   images.txt   pool-ec.yaml
cluster-multus-test.yaml   import-external-cluster.sh   pool-mirrored.yaml
cluster-on-local-pvc.yaml   monitoring   pool-test.yaml
cluster-on-pvc.yaml   mysql.yaml   pool.yaml
cluster-stretched-aws.yaml   nfs-load-balancer.yaml   psp.yaml
cluster-stretched.yaml   nfs-test.yaml   radosnamespace.yaml
cluster-test.yaml   nfs.yaml   rbdmirror.yaml
cluster.yaml   object-bucket-claim-delete.yaml   README.md
common-external.yaml   object-bucket-claim-notification.yaml   rgw-external.yaml
common-second-cluster.yaml   object-bucket-claim-retain.yaml   sqitevfs-client.yaml
common.yaml   object-ec.yaml   storageclass-bucket-delete.yaml
crds.yaml   object-external.yaml   storageclass-bucket-retain.yaml
create-external-cluster-resources.py   object-multisite-pull-realm-test.yaml   subvolumegroup.yaml
create-external-cluster-resources-tests.py   object-multisite-pull-realm.yaml   toolbox-job.yaml
csi   object-multisite-test.yaml   toolbox.yaml
csi-ceph-conf-override.yaml   object-multisite.yaml   volume-replication-class.yaml
dashboard-external-https.yaml   object-openshift.yaml   volume-replication.yaml
dashboard-external-https.yaml   object-test.yaml   wordpress.yaml
dashboard-ingress-https.yaml   object-user.yaml
dashboard-loadbalancer.yaml   object.yaml
master@master-node:~/rook/deploy/examples$ sudo nano pvc.yaml
master@master-node:~/rook/deploy/examples$ sudo nano pvc.yaml
master@master-node:~/rook/deploy/examples$ kubectl apply -f pvc.yaml
persistentvolumeclaim/test-pvc created
master@master-node:~/rook/deploy/examples$ sudo nano pod.yaml
master@master-node:~/rook/deploy/examples$ kubectl apply -f pod.yaml
pod/test-pod created
master@master-node:~/rook/deploy/examples$ kubectl get pod test-pod
NAME     READY   STATUS    RESTARTS   AGE
test-pod  0/1    Pending   0      8s
master@master-node:~/rook/deploy/examples$ kubectl get pvc test-pvc
NAME     STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
test-pvc  Pending   rook-ceph-block   113s
master@master-node:~/rook/deploy/examples$ kubectl apply -f toolbox.yaml
Warning: resource deployments/rook-ceph-tools is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
deployment.apps/rook-ceph-tools configured
master@master-node:~/rook/deploy/examples$ kubectl -n rook-ceph get pod -l "app=rook-ceph-tools"
NAME     READY   STATUS    RESTARTS   AGE
rook-ceph-tools-5679b7d8f-cm5hg  0/1    ContainerCreating   0      7m42s
master@master-node:~/rook/deploy/examples$ kubectl -n rook-ceph exec -it <rook-ceph-tools-pod> -- bash
bash: rook-ceph-tools-pod: No such file or directory
master@master-node:~/rook/deploy/examples$ 

```

Dans cette étape on test le fonctionnement de notre pods

```

rook-cephfs   rook-ceph.cephfs.csi.ceph.com   Delete   Immediate   true   109s
master@master-node:~/rook/deploy/examples$ ls
bucket-notification-endpoint.yaml   direct-mount.yaml   operator-openshift.yaml
bucket-notification.yaml   filesystem-ec.yaml   operator.yaml
bucket-topic.yaml   filesystem-mirror.yaml   osd-env-override.yaml
ceph-client.yaml   filesystem-test.yaml   osd-purge.yaml
cluster-external-management.yaml   filesystem.yaml   pool-builtin-mgr.yaml
cluster-external.yaml   images.txt   pool-ec.yaml
cluster-multus-test.yaml   import-external-cluster.sh   pool-mirrored.yaml
cluster-on-local-pvc.yaml   monitoring   pool-test.yaml
cluster-on-pvc.yaml   mysql.yaml   pool.yaml
cluster-stretched-aws.yaml   nfs-load-balancer.yaml   psp.yaml
cluster-stretched.yaml   nfs-test.yaml   radosnamespace.yaml
cluster-test.yaml   nfs.yaml   rbdmirror.yaml
cluster.yaml   object-bucket-claim-delete.yaml   README.md
common-external.yaml   object-bucket-claim-notification.yaml   rgw-external.yaml
common-second-cluster.yaml   object-bucket-claim-retain.yaml   sqitevfs-client.yaml
common.yaml   object-ec.yaml   storageclass-bucket-delete.yaml
crds.yaml   object-external.yaml   storageclass-bucket-retain.yaml
create-external-cluster-resources.py   object-multisite-pull-realm-test.yaml   subvolumegroup.yaml
create-external-cluster-resources-tests.py   object-multisite-pull-realm.yaml   toolbox-job.yaml
csi   object-multisite-test.yaml   toolbox.yaml
csi-ceph-conf-override.yaml   object-multisite.yaml   volume-replication-class.yaml
dashboard-external-https.yaml   object-openshift.yaml   volume-replication.yaml
dashboard-external-https.yaml   object-test.yaml   wordpress.yaml
dashboard-ingress-https.yaml   object-user.yaml
dashboard-loadbalancer.yaml   object.yaml
master@master-node:~/rook/deploy/examples$ sudo nano pvc.yaml
master@master-node:~/rook/deploy/examples$ sudo nano pvc.yaml
master@master-node:~/rook/deploy/examples$ kubectl apply -f pvc.yaml
persistentvolumeclaim/test-pvc created
master@master-node:~/rook/deploy/examples$ sudo nano pod.yaml
master@master-node:~/rook/deploy/examples$ kubectl apply -f pod.yaml
pod/test-pod created
master@master-node:~/rook/deploy/examples$ kubectl get pod test-pod
NAME     READY   STATUS    RESTARTS   AGE
test-pod  0/1    Pending   0      8s
master@master-node:~/rook/deploy/examples$ kubectl get pvc test-pvc
NAME     STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
test-pvc  Pending   rook-ceph-block   113s
master@master-node:~/rook/deploy/examples$ kubectl apply -f toolbox.yaml
Warning: resource deployments/rook-ceph-tools is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
deployment.apps/rook-ceph-tools configured
master@master-node:~/rook/deploy/examples$ kubectl -n rook-ceph get pod -l "app=rook-ceph-tools"
NAME     READY   STATUS    RESTARTS   AGE
rook-ceph-tools-5679b7d8f-cm5hg  0/1    ContainerCreating   0      7m42s
master@master-node:~/rook/deploy/examples$ kubectl -n rook-ceph exec -it <rook-ceph-tools-pod> -- bash
bash: rook-ceph-tools-pod: No such file or directory
master@master-node:~/rook/deploy/examples$ 

```

récupérer des informations sur un **PersistentVolumeClaim (PVC)** spécifique dans un cluster

Kubernetes

```

master@master-node: ~/rook/deploy/examples/csi/rbd
  NAME          READY   STATUS    RESTARTS   AGE
rook-ceph      0/5     Pending   0          5m59s
rook-ceph      0/1     Terminating   0          2s
rook-ceph      1/1     Running   0          7m29s
master@master-node: ~/rook/deploy/examples/csi/rbd$ kubectl describe pod csi-rbdplugin-provisioner-644468dd8c-mp5ld -n rook-ceph
Name:           csi-rbdplugin-provisioner-644468dd8c-mp5ld
Namespace:      rook-ceph
Priority:       2000000000
Priority Class Name: system-cluster-critical
Service Account: rook-csi-rbd-provisioner-sa
Node:           <none>
Labels:         app=csi-rbdplugin-provisioner
                contains=csi-rbdplugin-metrics
                pod-template-hash=644468dd8c
Annotations:    <none>
Status:         Pending
IP:             <none>
Controlled By: ReplicaSet/csi-rbdplugin-provisioner-644468dd8c
Containers:
  csi-provisioner:
    Image:      registry.k8s.io/sig-storage/csi-provisioner:v4.0.1
    Port:       <none>
    Host Port: <none>
    Args:
      --csi-address=$(ADDRESS)
      --v=0
      --timeout=2m30s
      --retry-interval-start=500ms
      --leader-election=true
      --leader-election-namespace=rook-ceph
      --leader-election-lease-duration=2m17s
      --leader-election-renew-deadline=1m47s
      --leader-election-retry-period=26s
      --default-fstype=ext4
      --extra-create-metadata=true

```

Cette commande **kubectl describe** permet de donner des informations sur le pod ainsi que son Statut (.....)

```

master@master-node: ~/rook/deploy/examples/csi/rbd
  NAME          READY   STATUS    RESTARTS   AGE
kube-system   kube-proxy-xwddn          1/1     Running   0          26m
kube-system   kube-scheduler-master-node 1/1     Running   0          29m
rook-ceph     csi-cephfsplugin-nqlb7      0/2     ContainerCreating   0          5s
rook-ceph     csi-cephfsplugin-p8nl2      2/2     Running   0          10m
rook-ceph     csi-cephfsplugin-provisioner-64d7866486-ds225 0/5     Pending   0          10m
rook-ceph     csi-cephfsplugin-provisioner-64d7866486-jfq7p  5/5     Running   1 (8m41s ago) 10m
rook-ceph     csi-rbdplugin-429lh        0/2     ContainerCreating   0          5s
rook-ceph     csi-rbdplugin-gq8p4        2/2     Running   0          10m
rook-ceph     csi-rbdplugin-provisioner-644468dd8c-flvs9      5/5     Running   1 (9m4s ago) 10m
rook-ceph     csi-rbdplugin-provisioner-644468dd8c-mp5ld      0/5     Pending   0          10m
rook-ceph     rook-ceph-csi-detect-version-fgcnf            0/1     Init:0/1   0          3s
rook-ceph     rook-ceph-operator-6f67847bd-8qknk            1/1     Running   0          11m
master@master-node: ~/rook/deploy/examples/csi/rbd$ kubectl delete pod csi-rbdplugin-provisioner-644468dd8c-mp5ld -n rook-ceph
pod "csi-rbdplugin-provisioner-644468dd8c-mp5ld" deleted
master@master-node: ~/rook/deploy/examples/csi/rbd$ kubectl -n rook-ceph get pods -A
NAMESPACE      NAME          READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-4s88c      1/1     Running   0          28m
kube-flannel   kube-flannel-ds-6k7fr      1/1     Running   0          28m
kube-system    coredns-565d847f94-hxh7z    1/1     Running   0          33m
kube-system    coredns-565d847f94-p4qc4    1/1     Running   0          33m
kube-system    etcd-master-node          1/1     Running   0          33m
kube-system    kube-apiserver-master-node 1/1     Running   0          33m
kube-system    kube-controller-manager-master-node 1/1     Running   0          33m
kube-system    kube-proxy-vltwm         1/1     Running   0          33m
kube-system    kube-proxy-xwddn         1/1     Running   0          30m
kube-system    kube-scheduler-master-node 1/1     Running   0          33m
rook-ceph      csi-cephfsplugin-nqlb7      2/2     Running   1 (3m33s ago) 4m15s
rook-ceph      csi-cephfsplugin-p8nl2      2/2     Running   0          14m
rook-ceph      csi-cephfsplugin-provisioner-64d7866486-ds225 5/5     Running   2 (24s ago) 14m
rook-ceph      csi-cephfsplugin-provisioner-64d7866486-jfq7p  5/5     Running   1 (12m ago) 14m
rook-ceph      csi-rbdplugin-429lh        2/2     Running   1 (3m32s ago) 4m15s
rook-ceph      csi-rbdplugin-gq8p4        2/2     Running   0          14m
rook-ceph      csi-rbdplugin-provisioner-644468dd8c-flqgm     5/5     Running   0          3m57s
rook-ceph      csi-rbdplugin-provisioner-644468dd8c-flvs9      5/5     Running   1 (13m ago) 14m
rook-ceph      rook-ceph-operator-6f67847bd-8qknk            1/1     Running   0          15m

```

```

Did you mean this?
    top
master@master-node:~/rook/deploy/examples$ kubectl get nodes -o json | jq '.items[] | {name: .metadata.name, taints: .spec.taints}'
Command 'jq' not found, but can be installed with:
sudo snap install jq # version 1.5+dfsg-1, or
sudo apt install jq # version 1.6~ubuntu0.20.04.1
See 'snap info jq' for additional versions.

master@master-node:~/rook/deploy/examples$ sudo apt install jq
[sudo] password for master:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libjq1 libonig5
The following NEW packages will be installed:
  jq libjq1 libonig5
0 upgraded, 3 newly installed, 0 to remove and 252 not upgraded.
Need to get 313 kB of archives.
After this operation, 1,062 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libonig5 amd64 6.9.4-1 [142 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libjq1 amd64 1.6~ubuntu0.20.04.1 [121 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 jq amd64 1.6~ubuntu0.20.04.1 [50.2 kB]
Fetched 313 kB in 0s (324 kB/s)
Selecting previously unselected package libonig5:amd64.
(Reading database ... 158511 files and directories currently installed.)
Preparing to unpack .../libonig5_6.9.4-1_amd64.deb ...
Unpacking libonig5:amd64 (6.9.4-1) ...
Selecting previously unselected package libjq1:amd64.
Preparing to unpack .../libjq1_1.6~ubuntu0.20.04.1_amd64.deb ...
Unpacking libjq1:amd64 (1.6~ubuntu0.20.04.1) ...
Selecting previously unselected package jq.
Preparing to unpack .../jq_1.6~ubuntu0.20.04.1_amd64.deb ...
Unpacking jq (1.6~ubuntu0.20.04.1) ...
Setting up libonig5:amd64 (6.9.4-1) ...
Setting up libjq1:amd64 (1.6~ubuntu0.20.04.1) ...
Setting up jq (1.6~ubuntu0.20.04.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Setting up jq (1.6~ubuntu0.20.04.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
master@master-node:~/rook/deploy/examples$ kubectl get nodes -o json | jq '.items[] | {name: .metadata.name, taints: .spec.taints}'
{
  "name": "ubuntu",
  "taints": [
    {
      "effect": "NoSchedule",
      "key": "node-role.kubernetes.io/control-plane"
    }
  ]
}
{
  "name": "worker2-node",
  "taints": null
}
master@master-node:~/rook/deploy/examples$ kubectl taint nodes <node-name> <taint-key>
bash: syntax error near unexpected token `<''
master@master-node:~/rook/deploy/examples$ kubectl taint nodes ubuntu node-role.kubernetes.io/control-plane
error: at least one taint update is required
master@master-node:~/rook/deploy/examples$ 
master@master-node:~/rook/deploy/examples$ kubectl get nodes -o json | jq '.items[] | {name: .metadata.name, taints: .spec.taints}'
{
  "name": "ubuntu",
  "taints": [
    {
      "effect": "NoSchedule",
      "key": "node-role.kubernetes.io/control-plane"
    }
  ]
}
{
  "name": "worker2-node",
  "taints": null
}
master@master-node:~/rook/deploy/examples$ kubectl taint nodes <node-name> <taint-key>
bash: syntax error near unexpected token `<''
master@master-node:~/rook/deploy/examples$ kubectl taint nodes ubuntu node-role.kubernetes.io/control-plane
error: at least one taint update is required
master@master-node:~/rook/deploy/examples$ kubectl get nodes
NAME        STATUS   ROLES      AGE     VERSION
ubuntu      Ready    control-plane   100m   v1.25.0
worker2-node Ready    <none>      91m    v1.25.0
master@Master-node:~/rook/deploy/examples$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
test-pod    0/1     Pending   0          21m
master@master-node:~/rook/deploy/examples$ kubectl get pods rook-ceph -n pods

Command 'kubectl' not found, did you mean:
  command 'kubectl' from snap kubectl (1.29.5)
See 'snap info <snapname>' for additional versions.

master@master-node:~/rook/deploy/examples$ kubectl pod rook-ceph -n pods
error: unknown command "pod" for "kubectl"

Did you mean this?
    top
master@Master-node:~/rook/deploy/examples$ kubects

Command 'kubects' not found, did you mean:
  command 'kubectx' from snap kubectx (0.9.5)
  command 'kubectl' from snap kubectl (1.29.5)
See 'snap info <snapname>' for additional versions.

```

```

master@master-node:~/rook/deploy/examples$ kubectl delete pod csi-rbdplugin-provisioner-56d69f5d8-hnr8t -n rook-ceph
pod "csi-rbdplugin-provisioner-56d69f5d8-hnr8t" deleted
master@master-node:~/rook/deploy/examples$ kubectl get pods -n rook-ceph
NAME                               READY   STATUS    RESTARTS   AGE
csi-cephfsplugin-fg9gk            2/2     Running   0          4m44s
csi-cephfsplugin-provisioner-75875b5887-djjwt  5/5     Running   0          39m
csi-cephfsplugin-provisioner-75875b5887-jw9xx  5/5     Running   0          39m
csi-cephfsplugin-rkhj9            2/2     Running   0          39m
csi-rbdplugin-4cpfh              2/2     Running   0          39m
csi-rbdplugin-bd6mk              2/2     Running   0          4m44s
csi-rbdplugin-provisioner-56d69f5d8-8j4c2      5/5     Running   0          39m
csi-rbdplugin-provisioner-56d69f5d8-xwr7x      5/5     Running   0          4m29s
rook-ceph-operator-547744ffbf-dct9b        1/1     Running   0          40m
rook-ceph-tools-5679b7d8f-cm5hg       0/1     ContainerCreating   0          37m
master@master-node:~/rook/deploy/examples$ 

```

```

master@master-node: ~/rook/deploy/e
master@ubuntu: ~
pod.yaml
GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
  - name: test-container
    image: busybox
    command: ["/bin/sh", "-c", "sleep 3600"]
    volumeMounts:
    - mountPath: "/mnt/test"
      name: test-volume
  volumes:
  - name: test-volume
    persistentVolumeClaim:
      claimName: test-pvc

```

Le fichier `pod.yaml` est un fichier YAML utilisé pour définir la configuration d'un pod dans Kubernetes. Un pod est l'unité de base de déploiement dans Kubernetes, représentant un ou plusieurs conteneurs qui partagent des ressources, telles que l'espace de stockage et l'adresse IP du réseau, et qui sont toujours planifiés ensemble sur le même nœud dans le cluster Kubernetes.

Dans un fichier `pod.yaml`, nous configurons les propriétés suivantes pour définir le comportement et les caractéristiques du pod :

metadata: Cette section contient des métadonnées telles que le nom, l'espace de noms (namespace), les annotations, les labels, etc., qui aident à identifier et à organiser le pod dans le cluster.

spec: Cette section définit les spécifications du pod, notamment :

containers: Liste des conteneurs à exécuter dans le pod, avec des détails tels que le nom du conteneur, l'image Docker à utiliser, les variables d'environnement, les ports à exposer, les ressources (CPU, mémoire) à allouer, etc.

volumes: Définition des volumes de stockage à monter dans le pod pour partager des données entre les conteneurs ou pour stocker des données persistantes.

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: rook-ceph-block

```

VI Installation et Configuration de Kubeflow version 1.7

```

master@master-node:~ $ cd ~/Downloads
master@master-node:~ $ cd Downloads
master@master-node:~ $ ls
Kustomize_V5_0_0_Linux_amd64.tar.gz  manifests-1.7.0.zip
master@master-node:~ $ tar -xvf Downloads/kustomize_v5.0.0_linux_amd64.tar.gz
kustomize
master@master-node:~ $ ls
cert-plugins-linux-amd64-v1.0.0.tgz  Downloads          kustomize      Public
cni-crtct-v1.22.0-ltinx-and64.tar.gz  kubelet.service  manifests.glt  rook
Desktop                         kubernetes       Music          Templates
Documents                        kubernetes-server-linux-amd64.tar.gz  Pictures      Videos
master@master-node:~ $ cd kustomize
bash: cd: kustomize: Not a directory
master@master-node:~ $ sudo nano kustomize
[sudo] password for master:
master@master-node:~ $ sudo nano kustomize
master@master-node:~ $ sudo mv kustomize /usr/local/bin/
master@master-node:~ $ kustomize version
v5.0.0
master@master-node:~ $ cd Downloads
master@master-node:~/Downloads $ ls
Kustomize_V5_0_0_Linux_amd64.tar.gz  manifests-1.7.0.zip
master@master-node:~/Downloads $ tar -xvf manifests-1.7.0.zip
tar: This does not look like a tar archive
tar: Skipping to next header
tar: Exiting with failure status due to previous errors
master@master-node:~/Downloads $ cd Manifests-1.7.0/
master@master-node:~/Downloads/Manifests-1.7.0 $ sudo su
root@master-node:/home/master/Downloads/Manifests-1.7.0# while ! kustomize build example | awk '!/well-defined/' | kubectl apply -f -; do echo "Retrying to apply resources"; sleep 10; done
The connection to the server localhost:8080 was refused - did you specify the right host or port?
# Warning: 'patchesStrategicMerge' is deprecated. Please use 'patches' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'vars' is deprecated. Please use 'replacements' instead. [EXPERIMENTAL] Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'vars' is deprecated. Please use 'replacements' instead. [EXPERIMENTAL] Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'patchesStrategicMerge' is deprecated. Please use 'patches' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'patchesStrategicMerge' is deprecated. Please use 'patches' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'bases' is deprecated. Please use 'resources' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'bases' is deprecated. Please use 'resources' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'bases' is deprecated. Please use 'resources' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'bases' is deprecated. Please use 'resources' instead. Run 'kustomize edit fix' to update your Kustomization automatically.
# Warning: 'vars' is deprecated. Please use 'replacements' instead. [EXPERIMENTAL] Run 'kustomize edit fix' to update your Kustomization automatically.

```

Télécharger et extraire kustomize et manifestes version 1.7.0

Nous pouvons Installer avec une seule commande tous les composants officiels de Kubeflow
while ! Kustomize build example |awk '!/well-defined/' | kubectl apply -f -; do echo

```

root@master-node:/home/master/Downloads/Manifests-1.7.0# while ! kustomize build example | awk '!/well-defined/' | kubectl apply -f -; do echo "Retrying to apply resources"; sleep 10; done

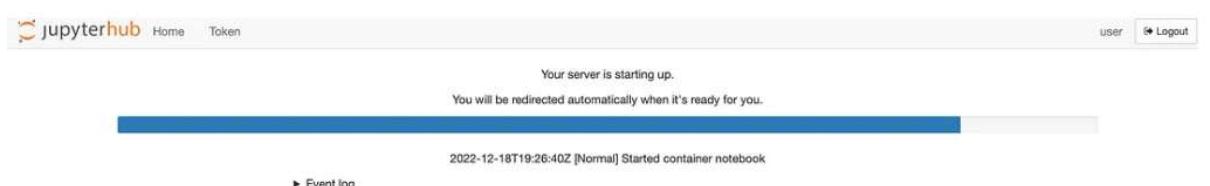
```

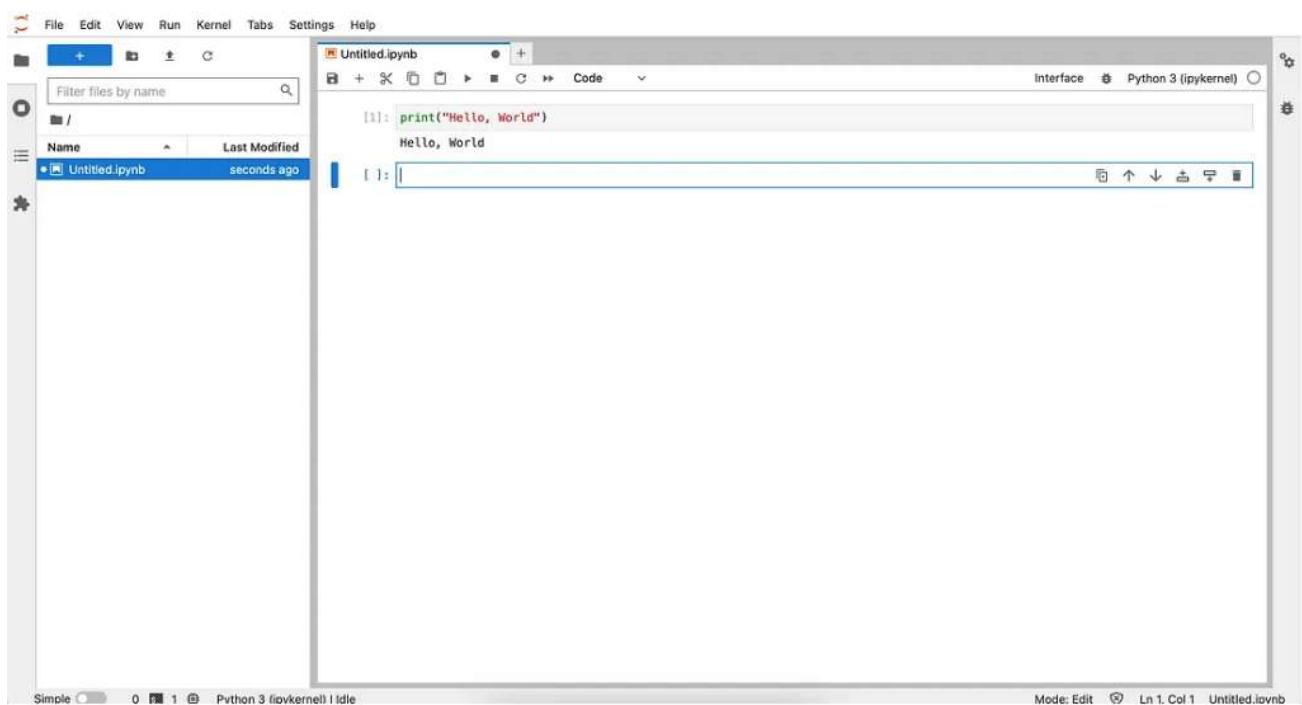
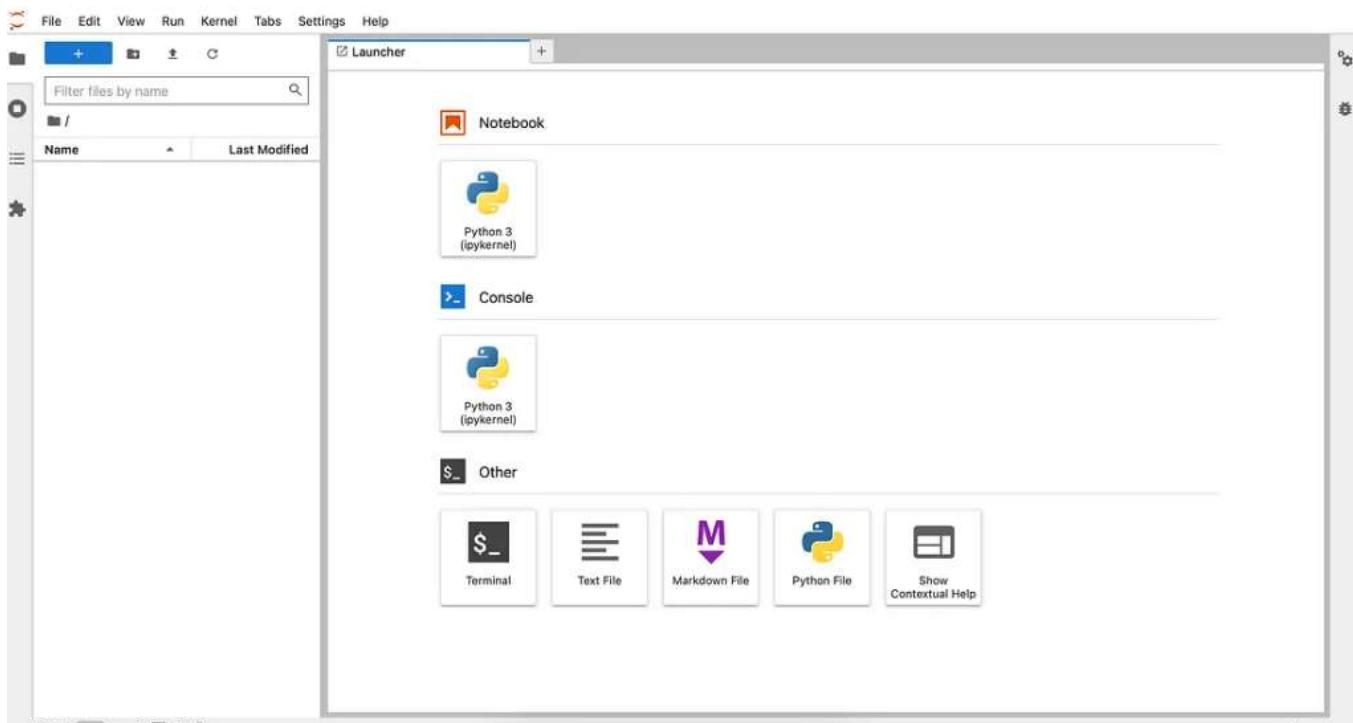
```
namespaces/kubeflow unchanged
customresourcedefinition.aplxtenstions.k8s.io/apiserversources.sources.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/authcodes.dex.coreos.com unchanged
customresourcedefinition.aplxtenstions.k8s.io/authorizationpolicies.security.istio.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/bokers.eventing.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/certificaterequests.cert-manager.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/certificates.cert-manager.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/certifications.networking.internal.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/challenges.acme.cert-manager.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/channels.messaging.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/clusterdomainclaims.networking.internal.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/clusterissuers.cert-manager.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/clustersevnguruntimes.serving.kserve.io configured
customresourcedefinition.aplxtenstions.k8s.io/clusterworkflowtemplates.argoproj.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/compositecontrollers.metacontroller.k8s.io configured
customresourcedefinition.aplxtenstions.k8s.io/configuration.serving.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/containerresources.sources.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/controllerrevisions.metacontroller.k8s.io configured
customresourcedefinition.aplxtenstions.k8s.io/cronworkflows.argoproj.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/decoratorcontrollers.metacontroller.k8s.io configured
customresourcedefinition.aplxtenstions.k8s.io/destinationrules.networking.istio.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/domainmappings.serving.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/envoyfilters.networking.istio.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/eventtypes.eventing.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/experiments.kubeflow.org unchanged
customresourcedefinition.aplxtenstions.k8s.io/gateways.networking.istio.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/images.caching.internal.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/inferencegraphs.serving.kserve.io configured
customresourcedefinition.aplxtenstions.k8s.io/inferenceservices.serving.kserve.io configured
customresourcedefinition.aplxtenstions.k8s.io/ingresses.networking.internal.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/issuers.cert-manager.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/istiooperators.install.istio.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/metrics.autoscaling.internal.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/pjpbjobs.kubeflow.org unchanged
customresourcedefinition.aplxtenstions.k8s.io/xjbs.kubeflow.org unchanged
customresourcedefinition.aplxtenstions.k8s.io/notebooks.kubeflow.org configured
customresourcedefinition.aplxtenstions.k8s.io/orders.acme.cert-manager.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/paddlejobs.kubeflow.org unchanged
customresourcedefinition.aplxtenstions.k8s.io/parallels.flows.knative.dev unchanged
customresourcedefinition.aplxtenstions.k8s.io/peerauthentications.security.istio.io unchanged
customresourcedefinition.aplxtenstions.k8s.io/plingsources.sources.knative.dev unchanged
```

| POD | NAME | READY | STATUS | RESTARTS | AGE |
|--------------|---|-------|-----------|---------------|------|
| rook-ceph | auth | 1/1 | Running | 0 | 10m |
| istio-system | istio-ingressgateway-67f7b5f88d-rkbbg | 0/1 | Running | 0 | 10m |
| istio-system | istiod-56f7cf9bd6-hkdw7 | 1/1 | Running | 0 | 10m |
| kube-system | etcd-master-node | 1/1 | Running | 0 | 28m |
| kube-system | kube-apiserver-master-node | 1/1 | Running | 0 | 28m |
| kube-system | kube-controller-manager-master-node | 1/1 | Running | 0 | 28m |
| kube-system | kube-scheduler-master-node | 1/1 | Running | 0 | 28m |
| kubeflow | centraldashboard-655c7d894c-dg5vh | 2/2 | Running | 0 | 67m |
| kubeflow | jupyter-web-app-deployment-c995cdd8c-h94pb | 2/2 | Running | 0 | 67m |
| kubeflow | katib-ui-7859bc4c67-w7s84 | 0/2 | Completed | 8 (51m ago) | 67m |
| kubeflow | kserve-models-web-app-99849d9f7-t85jx | 2/2 | Running | 0 | 67m |
| kubeflow | kubeflow-pipelines-profile-controller-59ccbd47b9-lpf9r | 1/1 | Running | 0 | 67m |
| kubeflow | metacontroller-0 | 1/1 | Running | 0 | 67m |
| kubeflow | metadata-envoy-deployment-5b6c575b98-7q2nc | 1/1 | Running | 0 | 67m |
| kubeflow | ml-pipeline-persistenceagent-798dbf66f2-2scn8 | 2/2 | Running | 0 | 63m |
| kubeflow | ml-pipeline-scheduledworkflow-859ff9c7b-2fxjx | 2/2 | Running | 0 | 67m |
| kubeflow | ml-pipeline-ul-6d69549787-vlcsj | 1/2 | Running | 0 | 64m |
| kubeflow | ml-pipeline-viewer-crdf-56f7cf7d9-84kkq | 2/2 | Running | 1 (66m ago) | 67m |
| kubeflow | ml-pipeline-visualizationserver-64447ffcc76-7snsm | 2/2 | Running | 0 | 67m |
| kubeflow | tensorboards-web-app-deployment-5cb4666798-qcsz7 | 1/2 | Running | 0 | 65m |
| kubeflow | tensorboards-web-app-deployment-5cb4666798-xrgwl | 0/2 | Completed | 1 | 67m |
| kubeflow | training-operator-7589458f95-kfv4p | 0/1 | Completed | 0 | 67m |
| kubeflow | training-operator-7589458f95-s7p4j | 1/1 | Running | 0 | 47m |
| kubeflow | volumes-web-app-deployment-59cf57d887-7vr8b | 1/2 | Running | 0 | 66m |
| rook-ceph | csi-cephfsplugin-b2bdb | 2/2 | Running | 1 (22h ago) | 22h |
| rook-ceph | csi-cephfsplugin-nqlb7 | 2/2 | Running | 1 (24h ago) | 24h |
| rook-ceph | csi-cephfsplugin-p8nl2 | 2/2 | Running | 0 | 24h |
| rook-ceph | csi-cephfsplugin-provisioner-64d7866486-ds225 | 5/5 | Running | 6 (96m ago) | 24h |
| rook-ceph | csi-cephfsplugin-provisioner-64d7866486-jfq7p | 5/5 | Running | 5 (7h44m ago) | 24h |
| rook-ceph | csi-rbdplugin-429lh | 2/2 | Running | 1 (24h ago) | 24h |
| rook-ceph | csi-rbdplugin-b8vwh | 2/2 | Running | 1 (22h ago) | 22h |
| rook-ceph | csi-rbdplugin-gq8p4 | 2/2 | Running | 0 | 24h |
| rook-ceph | csi-rbdplugin-provisioner-644468dd8c-flqgm | 5/5 | Running | 4 (96m ago) | 24h |
| rook-ceph | csi-rbdplugin-provisioner-644468dd8c-flvs9 | 5/5 | Running | 5 (7h44m ago) | 24h |
| rook-ceph | rook-ceph-crashcollector-worker1-node-b88568fffc-tvkvn7 | 1/1 | Running | 0 | 22h |
| rook-ceph | rook-ceph-crashcollector-worker1-node-57b5bf644-gjjw8 | 1/1 | Running | 0 | 2m2s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-4jcfx | 0/1 | Evicted | 0 | 56s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-4jwpt | 0/1 | Completed | 0 | 2m3s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-4wnrh | 0/1 | Evicted | 0 | 55s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-6ff78 | 0/1 | Evicted | 0 | 54s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-7qrk9 | 0/1 | Evicted | 0 | 57s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-cbqlh | 0/1 | Evicted | 0 | 56s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-d7mbs | 0/1 | Evicted | 0 | 55s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-f55sc5 | 0/1 | Evicted | 0 | 55s |
| rook-ceph | rook-ceph-crashcollector-worker2-node-6c4d558c8f-f6f12 | 0/1 | Evicted | 0 | 57s |

```
[root@ceph-01 ~]# kubectl port-forward svc/istio-ingressgateway -n istio-system 8080:80
[root@ceph-01 ~]# curl -v http://127.0.0.1:8080
* Rebuilt URL to: http://127.0.0.1:8080/
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
* HTTP request sent, awaiting response... 200 OK
* Content-Type: application/json
* Content-Length: 103
* 
{"status": "success", "data": "Hello from Istio Ingress Gateway!"}
```

VII. Déploiement de JupyterHub:





Les erreurs et les problèmes rencontrer lors de la Réalisation du projet :

```
./ptp
./ipvlan
./bandwidth
master@ubuntu:~$ sudo kubeadm init --kubernetes-version=v1.25.0 --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.25.0
[preflight] Running pre-flight checks
W0522 08:59:23.503463   18929 checks.go:1019] [preflight] WARNING: Couldn't create the interface used for talking to
the container runtime: crictl is required by the container runtime: executable file not found in $PATH
[WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature
gate of the kubelet
[WARNING FileExisting-ethtool]: ethtool not found in system path
[WARNING FileExisting-socat]: socat not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
[ERROR FileExisting-crictl]: crictl not found in system path
[ERROR FileExisting-conctrack]: contrack not found in system path
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with `--v=5` or higher
master@ubuntu:~$ sudo apt install -y ethtool cri-tools contrack
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package cri-tools
master@ubuntu:~$ apt install -y ethtool cri-tools contrack
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
master@ubuntu:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease
Err:2 https://download.docker.com/linux/ubuntu focal InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 7EA0A9C3F273FC08
Ign:3 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:5 https://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Err:6 https://packages.cloud.google.com/apt kubernetes-xenial Release
  404  Not Found [IP: 142.250.201.174 443]
Hit:7 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
W: An error occurred during the signature verification. The repository is not updated and the previous index files will not be used. GPG error: https://download.docker.com/linux/ubuntu focal InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 7EA0A9C3F273FC08
E: The repository 'https://apt.kubernetes.io kubernetes-xenial' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
W: Target Packages (stable/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:58 and /etc/
apt/sources.list.d/docker.list:1
W: Target Packages (stable/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:58 and /etc/ap
t/sources.list.d/docker.list:1
W: Target Translations (stable/i18n/Translation-en US) is configured multiple times in /etc/apt/sources.list:58 and
```

Lors de l'initialisation de Cluster le problème c'était au niveau de swapp qui était activer le probleme est resolu lors de la désactivation de Swapp

```
Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.238.165:6443 --token mxyckw.qv5yix3cdgpxil3a \
  --discovery-token-ca-cert-hash sha256:52d3883c0fabff2883f0f216bf6811b90bbd7a42f441a6fcfc5b8466259d7e94
master@ubuntu:~$ sudo systemctl start kubelet.service
[sudo] password for master:
master@ubuntu:~$ sudo systemctl enable kubelet.service
master@ubuntu:~$ sudo nano /etc/hosts
[sudo] password for master:
master@ubuntu:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ sudo nano /etc/hosts
[sudo] password for master:
master@ubuntu:~$ sudo systemctl restart containerd
master@ubuntu:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ sudo systemctl restart containerd
master@ubuntu:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ ^C
master@ubuntu:~$ sudo systemctl restart containerd
master@ubuntu:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ kubectl cluster-info

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ kubectl cluster-info dump
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@ubuntu:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
  Loaded: loaded (/lib/systemd/system/kubelet.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/kubelet.service.d
            └─10-kubeadm.conf
    Active: active (running) since Wed 2024-05-22 09:17:03 PDT; 15h ago
      Docs: https://kubernetes.io/docs/home/
    Main PID: 21452 (kubelet)
       Tasks: 16 (limit: 9387)
      Memory: 38.2M
     CGroup: /system.slice/kubelet.service
             └─21452 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/e
```

Ici lors de l'utilisation de kubectl get nodes on a tombe dans le problème de connexion au serveur qui est refuse dans ce cas on a vérifier le fichier /etc/hosts si il est bien configurer on a désactiver le firewall car nous a causer des problèmes de connexion

```
master@ubuntu:~$ sudo systemctl restart containerd
master@ubuntu:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0523 01:27:26.929501 34417 version.go:256] remote version is much newer: v1.30.1; falling back to: stable-1.25
[init] Using Kubernetes version: v1.25.16
[preflight] Running pre-flight checks
    [WARNING FileExisting-socat]: socat not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
    [ERROR Port-6443]: Port 6443 is in use
    [ERROR Port-10259]: Port 10259 is in use
    [ERROR Port-10257]: Port 10257 is in use
    [ERROR FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserve
r.yaml already exists
    [ERROR FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kubernetes/manifests/kube
-controller-manager.yaml already exists
    [ERROR FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/kube-schedule
r.yaml already exists
    [ERROR FileAvailable--etc-kubernetes-manifests-etcd.yaml]: /etc/kubernetes/manifests/etc.yaml already exist
s
    [ERROR Port-10250]: Port 10250 is in use
    [ERROR Port-2379]: Port 2379 is in use
    [ERROR Port-2380]: Port 2380 is in use
    [ERROR DirAvailable--var-lib-etcd]: /var/lib/etcd is not empty
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...
To see the stack trace of this error execute with --v=5 or higher
master@ubuntu:~$ sudo kubeadm token create --print-join-command
kubeadm join 192.168.238.165:6443 --token f0hfir.fb72km3sbiupf3ci --discovery-token-ca-cert-hash sha256:52d38836c0fa
bbf2883f0f216bf6811b90bbd7a42f441a6fc5b8466259d7e94
master@ubuntu:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
ubuntu    Ready    control-plane   16h    v1.25.0
master@ubuntu:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
ubuntu    Ready    control-plane   16h    v1.25.0
master@ubuntu:~$ kubectl get nodes
NAME      STATUS   ROLES      AGE     VERSION
ubuntu    NotReady <none>    9s     v1.25.0
master@ubuntu:~$ sudo kubeadm init --kubernetes-version=v1.25.0 --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.25.0
[preflight] Running pre-flight checks
    [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kuberne
tes.default.svc.cluster.local ubuntu] and IPs [10.96.0.1 192.168.238.165]
```

VIII. Conclusion et Perspectives :

Le projet de déploiement de JupyterHub sur un cluster Kubernetes avec l'intégration de Docker, Rook Ceph et Kubeflow a permis de mettre en place une infrastructure robuste, scalable et performante pour la gestion de notebooks interactifs. Les principaux résultats obtenus sont les suivants :

3- Scalabilité et Performance:

- La configuration de Kubernetes assure une distribution efficace des charges de travail et permet d'ajuster facilement les ressources en fonction des besoins.
- L'utilisation de Docker garantit l'isolation des environnements et facilite la gestion des dépendances.

4- Haute Disponibilité:

- La mise en place de Rook Ceph pour le stockage persistant offre une solution résiliente et hautement disponible, essentielle pour la durabilité des données utilisateurs.
- Les mécanismes de reprise automatique de Kubernetes augmentent la fiabilité du système.

5- Automatisation et Flexibilité:

- Kubeflow simplifie l'automatisation des workflows d'apprentissage automatique, facilitant ainsi le travail des data scientists et des ingénieurs.
- JupyterHub offre une interface conviviale pour les utilisateurs finaux, permettant une collaboration efficace et une gestion centralisée des ressources.

6- Sécurité et Gestion des Ressources:

- Les configurations réseau et de sécurité assurent une protection adéquate des données et des communications au sein du cluster.
- Les politiques de sécurité des conteneurs et le contrôle des accès basé sur les rôles (RBAC) renforcent la sécurité globale de l'infrastructure.

Perspectives

7- Amélioration de la Scalabilité:

- Envisager l'ajout de davantage de nœuds worker pour répondre à une augmentation significative de la demande.
- Explorer des solutions de scalabilité automatique pour ajuster dynamiquement les ressources en fonction des charges de travail.

8- Optimisation des Performances:

- Analyser les performances actuelles et identifier les goulots d'étranglement potentiels.
- Mettre en place des optimisations au niveau du réseau et du stockage pour améliorer encore plus les temps de réponse et la gestion des données.

9- Renforcement de la Sécurité:

- Implémenter des audits réguliers de sécurité pour identifier et corriger les vulnérabilités potentielles.
- Améliorer la gestion des identités et des accès en intégrant des solutions comme LDAP ou des fournisseurs d'identité tiers (par exemple, OAuth).

10- Intégration et Extension:

- Intégrer des outils supplémentaires pour la surveillance et la gestion (par exemple, Prometheus pour le monitoring, Grafana pour la visualisation des métriques).

- Explorer des possibilités d'extension avec d'autres services et plateformes (par exemple, services cloud, bases de données distribuées).

11- Formation et Documentation :

- Offrir des sessions de formation continue aux utilisateurs pour maximiser l'adoption et l'efficacité de la plateforme.
- Maintenir une documentation à jour pour les administrateurs et les utilisateurs, couvrant les meilleures pratiques et les procédures de dépannage.

En conclusion, ce projet établit une base solide pour une infrastructure de calcul interactif et collaboratif. Les perspectives envisagées visent à améliorer continuellement les performances, la sécurité et la flexibilité de la solution, tout en répondant aux besoins évolutifs des utilisateurs.