

LAB 8 : Mise en place d'un Cluster SWARM

1. Introduction

Docker Swarm est un outil de clustering natif pour les conteneurs Docker qui peut être utilisé pour gérer un cluster de nœuds Docker en tant que système virtuel unique. Docker Swarm se compose de deux composants principaux: le nœud Manager et le nœud Worker.

- Nœud de gestionnaire (manager) utilisé pour gérer les tâches de gestion de cluster telles que le maintien de l'état du cluster, la planification des services et la gestion des points de terminaison de l'API HTTP en mode Swarm.
- Le nœud Worker est une instance du moteur Docker qui peut être utilisée pour exécuter un conteneur.

Le gestionnaire Swarm vous permet de créer une instance de gestionnaire principale et plusieurs instances de réplication en cas de défaillance de l'instance principale.

Vous pouvez déployer des nœuds de gestionnaire et de travail au moment de l'exécution en mode Swarm du moteur Docker.

Dans ce tutorial, nous allons parcourir les instructions étape par étape sur la configuration du cluster Docker Swarm à trois nœuds sur CentOS 7. Prérequis

2. Préparation de l'environnement du cluster

Le gestionnaire Swarm vous permet de créer une instance de gestionnaire principale et plusieurs instances de réplication en cas de défaillance de l'instance principale.

- Installez Docker Engine sur 3 nœuds (1 manager + 2 workers).
- Editez sur les machines le fichier **/etc/hosts**, pour pouvoir identifier les machines à travers leurs nom respectifs.

```
[root@localhost ~]# vi /etc/hosts
```

```
192.168.208.128 manager
192.168.208.130 worker1
192.168.208.133 worker2
```

- Ensuite, vous devrez configurer le nom d'hôte sur chaque nœud selon le fichier **/etc/hosts**.
 - Sur le manager

```
[root@localhost ~]# hostnamectl set-hostname manager
```

- Sur le worker1

```
[root@localhost ~]# hostnamectl set-hostname worker1
```

- Sur le worker 2

```
[root@localhost ~]# hostnamectl set-hostname worker2
```

- Configurer le pare-feu
 - Ensuite, vous devrez ouvrir les ports 7946, 4789, 2376, 2377 et 80 sur le pare-feu pour qu'un cluster swarm fonctionne correctement. Exécutez la commande suivante sur tous les nœuds:

```
firewall-cmd --permanent --add-port=2376/tcp
firewall-cmd --permanent --add-port=2377/tcp
firewall-cmd --permanent --add-port=7946/tcp
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --permanent --add-port=7946/udp
firewall-cmd --permanent --add-port=4789/udp
```

- Enfin, rechargez le pare-feu et le service Docker pour appliquer toutes les modifications:

```
firewall-cmd --reload
systemctl restart docker
```

3. Créer un cluster Swarm

Initialisation du cluster

- Initialiser le mode Swarm sur le nœud Manager. Vous pouvez le faire en exécutant la commande `docker swarm init`. Cette commande fera de votre nœud un nœud de gestionnaire

```
[root@manager ~]# docker swarm init
Swarm initialized: current node (sh0spr4l1lc65wqwppnegz4ss) is now a manager.
```

To add a worker to this swarm, run the following command:

```
docker swarm join --token SWMTKN-1-
0gv419k957umluv2gi4mkpml09cp6ozoswwk3rd3g9227vipko-1ykngfpgkq80r8gd9g8egpp634
192.168.208.128:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

- Vérifiez l'état du cluster avec la commande

```
[root@manager ~]# docker info
```

- Affichez la liste des nœuds de votre cluster avec la commande suivante:

```
[root@manager ~]# docker node ls
```

Joindre les nœuds Worker au nœud Manager

Le nœud Manager est maintenant prêt. Ensuite, vous devrez ajouter le nœud Worker au nœud Manager.

Vous pouvez le faire en exécutant la commande `docker swarm join` sur les deux nœuds Worker

- Sur le worker 1

```
[root@worker1 ~]# docker swarm join --token SWMTKN-1-0gv419k957umluv2gi4mkpml09cp6ozoswwk3rd3g9227vipko-1ykngfpgkq80r8gd9g8egpp634192.168.208.128:2377
This node joined a swarm as a worker.
```

- Sur le Worker 2

```
docker swarm join --token SWMTKN-1-0gv419k957umluv2gi4mkpml09cp6ozoswwk3rd3g9227vipko-1ykngfpgkq80r8gd9g8egpp634192.168.208.128:2377
This node joined a swarm as a worker.
```

- Affichez la liste des nœuds de votre cluster avec la commande suivante:

```
[root@manager ~]# docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE
VERSION					
sh0spr4111c65wqwppnegz4ss *	manager	Ready	Active	Leader	20.10.2
faflsa9qrxphaimczrp4t88dh	worker1	Ready	Active		20.10.2
xfobck95cnu7dq1xbzplt6lk1	worker2	Ready	Active		20.10.2

- Inspectez le manager

```
[root@manager ~]# docker inspect manager
```

- Inspectez le worker1

```
[root@manager ~]# docker inspect worker1
```