

# Docker Swarm

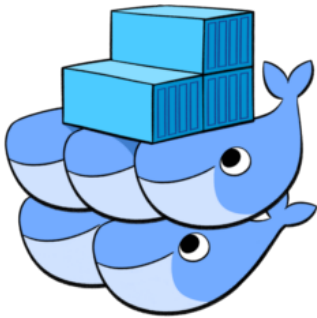
## Introduction

**Docker Swarm** est une surcouche native de Docker pour **orchestrer** un cluster de machines exécutant des conteneurs Docker.

Contrairement à une **exécution locale** de conteneurs, Swarm permet de déployer des conteneurs avec quelques avantages :

- Distribution automatique
- Haute disponibilité
- Scaling automatique
- Redondances
- Etc.

Docker Swarm transforme un ensemble de nœuds Docker en un **cluster** unique.



Chaque nœud peut être un **manager** (coordonne le cluster) ou un **worker** (exécute les tâches).

Les services sont définis avec des règles de réplication, de placement et de mise à jour.

## Exemple

- Nous partons d'une application web hébergée sur **Nginx** avec une exigence de **haute disponibilité** et de **répartition de charge**.

- Nous disposons de plusieurs **instances Docker** (machines hôtes) et nous souhaitons que l'application soit toujours accessible, même si une machine tombe en panne.

```
# Sur la machine principale (noeud principal)
docker swarm init

# Sur chaque noeud
docker swarm join IP_NOEUD_PRINCIPAL:2377

# Création du service avec 3 instances
docker service create --name web --replicas 3 -p 80:80 nginx

# Vérifier le déploiement
docker service ls
docker service ps web
```

Le service Nginx est maintenant réparti sur plusieurs nœuds, avec équilibrage de charge.

## Différence avec Docker Compose

**Docker Compose** est principalement utilisé pour le développement local ou le déploiement d'application sur une machine unique.

L'ensemble des services, réseaux et volumes nécessaires sont détaillés dans le fichier `docker-compose.yml`.

**Docker Swarm** est conçu pour le déploiement en production sur plusieurs machines. Il offre des fonctionnalités d'orchestration qui ne sont pas disponibles avec **Docker Compose**.

Fonctionnalité	Docker Compose	Docker Swarm
Scope	Machine locale	Cluster
Orchestration	Basique	Avancée
Scaling	Non	Oui

Fonctionnalité	Docker Compose	Docker Swarm
Haute dispo	Non	Oui
Load balancing	Non	Oui

# Docker Secret

**Docker Swarm** propose une fonctionnalité **Docker Secrets** pour stocker et distribuer les données sensibles de manière sécurisée aux services qui en ont besoin.

## Exemple :

```
# Création d'un secret
printf "my super secret password" | docker secret create my_secret -

# Utilisation avec un service
docker service create --name app --secret my_secret alpine:latest sh -c "cat
/run/secrets/my_secret"
```

Dans cet exemple, le conteneur `alpine` accède au mot de passe via le fichier `/run/secrets/my_secret`.

Ce fichier est monté automatiquement par **Swarm** et n'est pas visible en dehors du conteneur.

La fonctionnalité Docker Secret n'est disponible qu'avec Docker Swarm.  
Les volumes sont le meilleur moyen de gérer des informations sensible en dehors d'un cluster Docker Swarm.

Revision #7

Created 18 September 2025 18:39:54 by Thibaud FRICHET

Updated 18 September 2025 19:47:56 by Thibaud FRICHET