Niveau: Master professionnel

Cours : Migration cloud

Docker Swarm

Membres du groupe

AFOMALE Komi David Franck

AKAKPO Remi Eli Kokou

BOTRE LARE Aboudou

LIMDEYOU Médar Essossinam

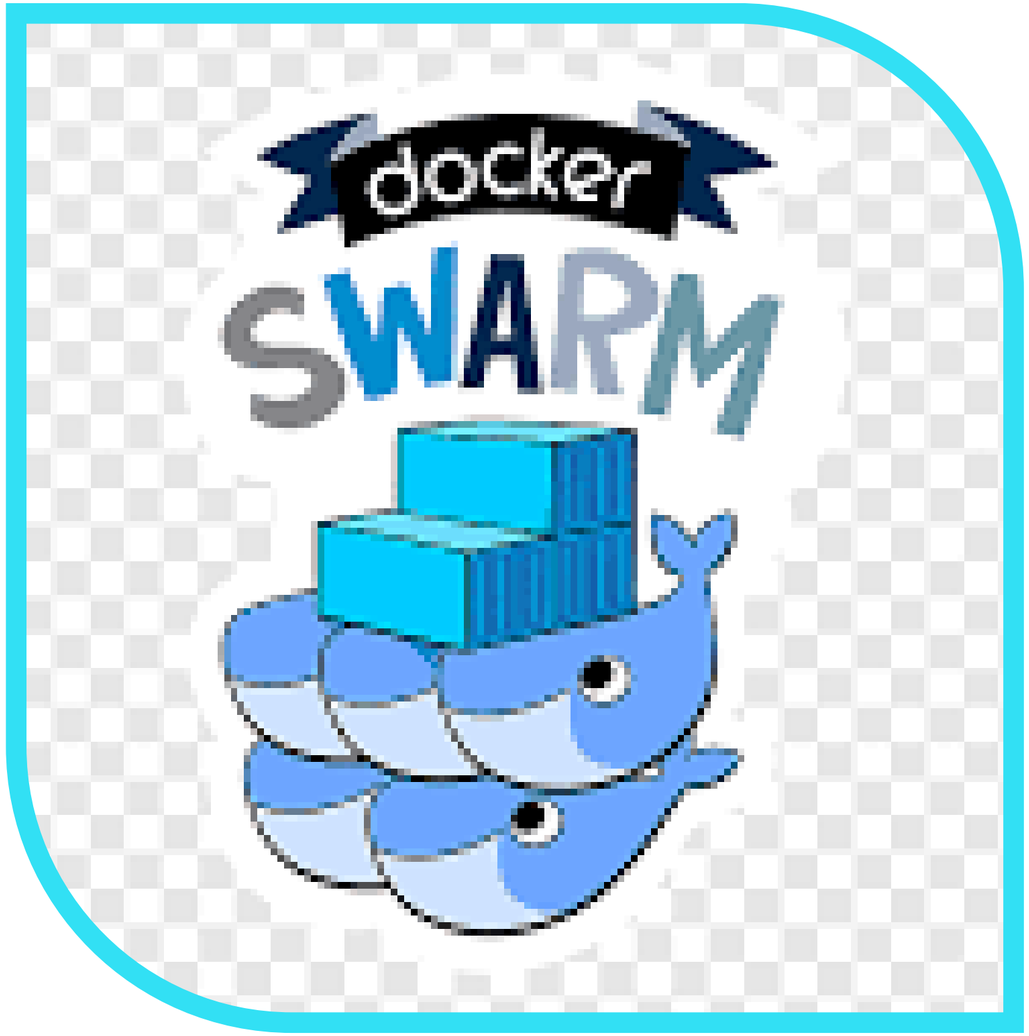
NAPO Kossi M’PAKA

NOYOULIWA Essowaza Victoire:

Enseignant

M. WOAGOU Daniel

Sommaire



Introduction

Docker Swarm est l'outil d'orchestration natif de Docker. Il permet de transformer un groupe de machines (physiques ou virtuelles) en un seul cluster virtuel afin de gérer, déployer et mettre à l'échelle des conteneurs de manière centralisée et simplifiée.

1. Pourquoi utiliser Docker Swarm ?

Haute disponibilité

Si un nœud tombe en panne, Swarm redistribue automatiquement les conteneurs sur les autres nœuds.

Scalabilité

On peut augmenter ou diminuer le nombre d'instances d'un service (réplicas) avec une seule commande.

Equilibrage de charge

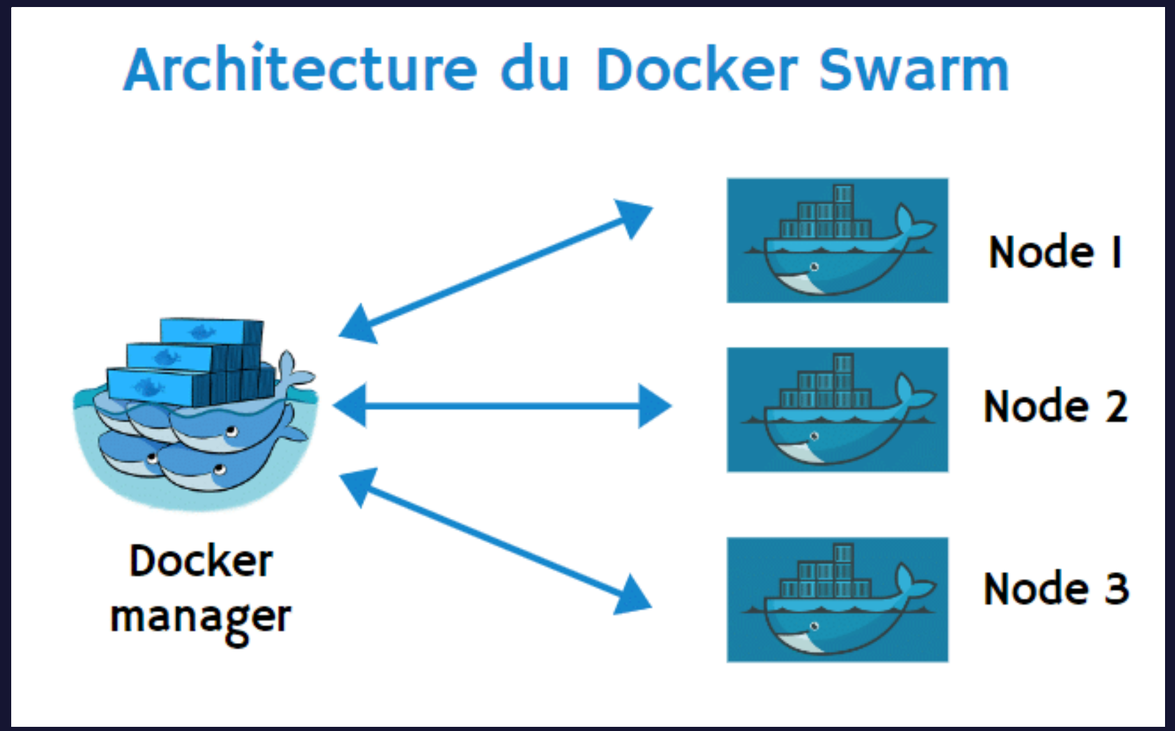
Swarm intègre un "Ingress Mesh" qui répartit les requêtes entrantes entre tous les conteneurs disponibles du service.

Modèle déclaratif

Vous définissez l'état souhaité (ex: "je veux 5 instances de Nginx") et Docker s'assure que cet état est maintenu en permanence.

2. Architecture de Docker Swarm

La fonction docker swarm reconnaît trois types de nœuds différents à savoir le Manager node, Leader node et le Worker node



Manager Node

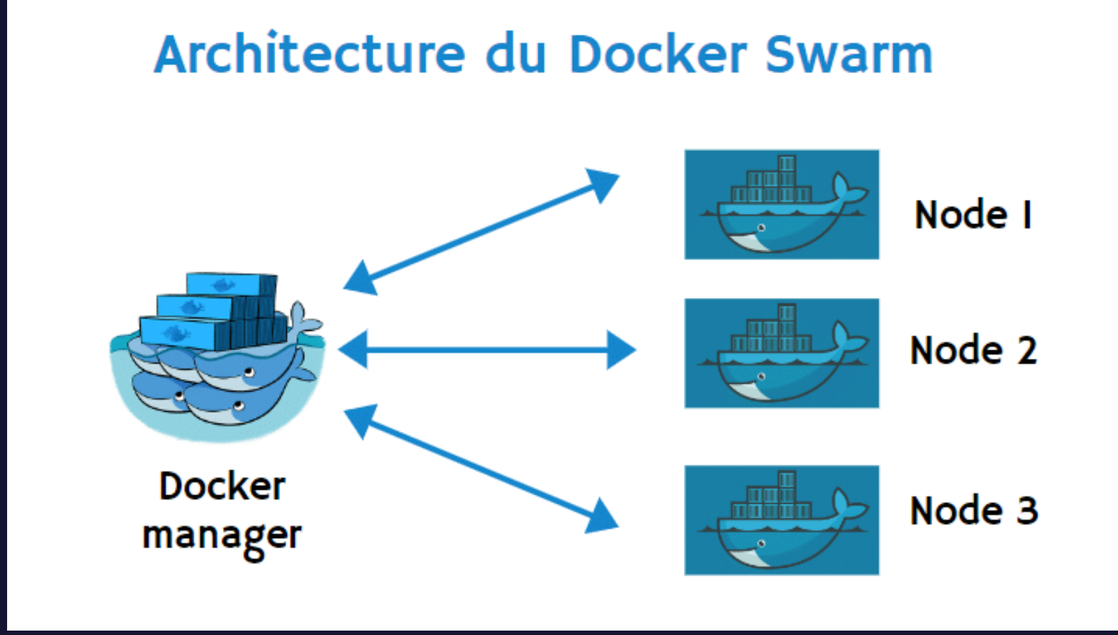
- attribue des tâches aux nœuds de travail dans le swarm
- effectuer certaines des tâches de gestion nécessaires au fonctionnement du swarm.

2. Architecture de Docker swarm (2)

Leader node

Le nœud leader prend toutes les décisions de gestion du swarm et d'orchestration des tâches pour le swarm. Si le nœud leader devient indisponible en raison d'une panne, un nouveau nœud leader peut être élu à l'aide de l'algorithme de consensus Raft.

L'algorithme de consensus Raft toutes les modifications apportées à l'état du cluster (comme la création ou la mise à jour de services) sont effectuées de manière sûre et cohérente, même en cas de défaillance de certains nœuds managers.



Worker node

chaque nœud de travail fonctionne en recevant et en exécutant les tâches qui lui sont allouées par les nœuds de gestion. Par défaut, tous les modes de gestion sont également des nœuds de travail et sont capables d'exécuter des tâches lorsqu'ils ont les ressources disponibles pour le faire.

3. Types de services en mode Docker Swarm

Les services répliqués

Les services répliqués sont conçus pour garantir qu'un nombre spécifique de répliques d'un conteneur fonctionnent simultanément sur différents nœuds du cluster. Lorsque vous créez un service répliqué, vous spécifiez le nombre exact de répliques que vous souhaitez exécuter. Le gestionnaire de Swarm s'assure alors que le nombre requis de répliques est toujours en cours d'exécution, réparties sur les nœuds disponibles du cluster.

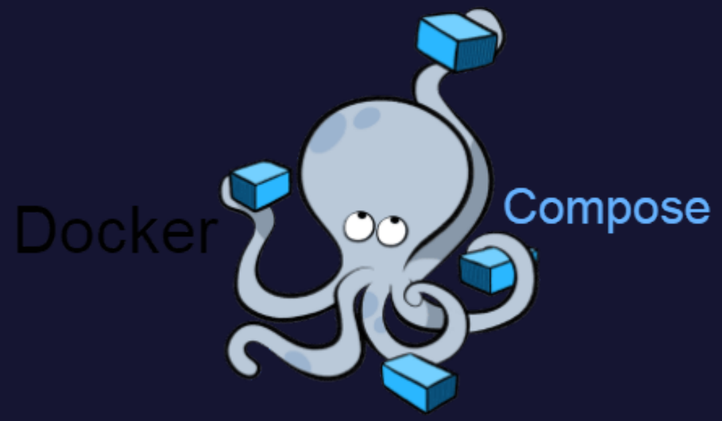
Les services globaux

Contrairement aux services répliqués qui spécifient un nombre fixe de répliques, les services globaux sont conçus pour garantir qu'une instance du service fonctionne sur chaque nœud disponible du cluster Docker Swarm. Lorsque vous créez un service global, le gestionnaire de nœuds planifie automatiquement une instance du service sur chaque nœud qui répond aux contraintes et aux besoins en ressources définis pour ce service.

4. Commandes principales pour manipuler le Swarm

Action	Commande
Initialiser le cluster (sur le manager)	docker swarm init --advertise-addr <IP>
Rejoindre en tant que worker	docker swarm join --token <TOKEN> <IP>:2377
Lister les nœuds du cluster	docker node ls
Créer un service	docker service create --name web --replicas 3 -p 80:80 nginx
Scaler un service	docker service scale web=5
Déployer une application complète	docker stack deploy -c docker-compose.yml ma_stack

5. Difference entre Docker Swarm et Docker Compose



VS

- Conçu pour gérer plusieurs conteneurs sur une seule machine (développement local).

- Conçu pour gérer des conteneurs sur un cluster de plusieurs machines (production). Il introduit la section deploy: dans le fichier YAML (réplicas, politiques de redémarrage, etc.).

6. Fonctionnalités avancées

Rolling Updates (Mises à jour progressives) : Permet de mettre à jour une application version par version sans interruption de service. Si une erreur survient, Swarm peut effectuer un Rollback automatique vers la version précédente.

Gestion des Secrets et Configurations : Permet de stocker de manière sécurisée (chiffrée) des mots de passe ou des certificats sans les inclure directement dans les images Docker.

Services Globaux vs Répliqués :
Répliqué : Vous fixez le nombre d'instances (exp: 3).
Global : Une instance tourne sur chaque nœud du cluster (idéal pour la surveillance ou les logs).

7. Conclusion

Docker Swarm est la solution idéale pour les équipes qui souhaitent une orchestration simple, rapide à mettre en place et parfaitement intégrée à l'écosystème Docker.

Bien que Kubernetes soit plus puissant pour des infrastructures massives, Swarm reste imbattable pour sa facilité d'utilisation et sa légèreté.

MERCI DE VOTRE ATTENTION