

Docker Swarm

Exercice 1

L'objectif de cet exercice est de créer un cluster Docker Swarm composé de trois nœuds managers et de trois nœuds workers. Les machines seront des machines virtuelles (VM) créées avec LXD, et l'installation sera automatisée à l'aide d'Ansible.

1. Créez un fichier d'inventaire `hosts` Ansible qui contient les informations des machines. Ce fichier doit inclure `localhost`, les nœuds managers (`manager{1,2,3}`) et les nœuds workers (`worker{1,2,3}`). Assurez-vous d'utiliser la bonne méthode de connection à `localhost` et aux machines virtuelles LXD.
2. Écrivez un playbook Ansible pour créer des machines virtuelles LXD. Vous devez créer trois nœuds managers et trois nœuds workers, en vous assurant que les VM sont correctement configurées avec des ressources appropriées.
 - Utilisez l'image `ubuntu/20.04`.
 - Chaque VM disposera de 2 CPU et 2 Go de RAM.
3. Pour cette question vous pouvez suivre les étapes fournies par le site <https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>. Pour ce faire, créez un playbook Ansible pour :
 - Supprimer les anciennes installations de Docker.
 - Mettre à jour le cache des paquets APT et installer les dépendances nécessaires.
 - Ajouter la clé GPG de Docker et le dépôt APT Docker.
 - Installer Docker et démarrer le service Docker.
4. Créez un playbook qui initialise le Docker Swarm sur le premier nœud manager (`manager1`). Ensuite, ajoutez les nœuds workers et les autres nœuds managers au cluster. Voici les tâches à réaliser par votre playbook (L'adresse IP du `manager1` ainsi que les tokens d'adhésion peuvent être ajoutés au faits en utilisant le module `set_fact` ou l'option `register`). :
 - Initialiser Docker Swarm sur le premier nœud manager `manager1` en exécutant la commande `docker swarm init --advertise-addr <adr-ip-manager1>`
 - Récupérer les tokens d'adhésion pour les nœuds workers et managers.
 - Sur les nœuds workers, utilisez le token pour rejoindre le cluster Docker Swarm.
 - Sur les autres nœuds managers, utilisez le token pour rejoindre le cluster Docker Swarm.
5. Une fois que les nœuds ont rejoint le cluster, vérifiez que tous les nœuds sont présents dans le cluster Docker Swarm.
6. Créez un playbook pour faire quitter le cluster à tous les nœuds et supprimer les machines virtuelles. Pour ce faire
 - Créez un playbook qui effectue les actions suivantes :
 - Faire quitter tous les nœuds du Swarm en utilisant `docker swarm leave -force`.
 - Supprime toutes les machines virtuelles.

Exercice 2

1. Quelle commande permet de créer un service Docker avec Nginx et de le rendre accessible via le port 8080 de l'hôte ?

2. Quelles options doivent être spécifiées pour créer 3 répliques du service ?
3. Quelle commande permet de lister les services déployés sur le Swarm ?
4. Quelle commande permet de voir les détails des répliques (tâches) d'un service ?
5. Quelle commande permet d'augmenter le nombre de répliques d'un service de 3 à 5 ?
6. Observez le déploiement des nouveaux conteneurs.
7. Arrêtez un des noeuds du cluster et observez comment Swarm gère la résilience.
8. Quelle commande permet de réduire le nombre de répliques d'un service de 3 à 2 ?
9. Quelle commande permet de mettre à jour l'image du service vers une nouvelle version de Nginx ?
10. Quelle commande permet de forcer le redémarrage d'un service sans modifier son image ?
11. Quelle commande permet de supprimer un service d'un Swarm ?
12. Quelle commande permet de vérifier que le service a bien été supprimé ?
13. Arrêtez le leader du cluster.
14. Observez comment le cluster réagit et élit un nouveau leader.
15. Restaurez le noeud arrêté et vérifiez l'état du cluster.
16. Vérifiez que le trafic entre les noeuds est chiffré par défaut.
17. Que fait la commande

```
docker swarm update --autolock
```

- 18.
19. Notez la clé générée et testez le redémarrage du leader.

Exercice 3

1. Qu'est ce qu'un service de type global ?
2. Faire quitter `worker3` du cluster.
- 3.
4. Déployez un service global (`--mode global`) qui exécute un conteneur Alpine sur chaque noeud du cluster.
5. Vérifiez que le service est déployé sur tous les noeuds du cluster.
6. Réintégrer le noeud `worker3` au cluster, puis confirmez que le service global a été déployé sur le nouveau noeud.
7. Supprimez un noeud du cluster et vérifiez que le service est supprimé de ce noeud.
8. Expliquez en quoi un service global diffère d'un service avec réplicas dans le contexte d'un cluster.
9. Quelle commande permet de mettre un nœud en mode *maintenance* (drain) afin qu'il ne soit plus affecté de nouvelles tâches ?
10. Quelle commande permet de remettre un nœud en ligne une fois la maintenance terminée ?
11. Qu'est-ce qu'un *stack* Docker dans le contexte de Docker Swarm et pourquoi est-il utile pour déployer des applications multi-services ?

12. Donnez un exemple d'un fichier `docker-compose.yml` qui définit un stack composé de deux services : un serveur web *nginx* et une base de données *mysql*. Spécifiez les détails comme les variables d'environnement pour MySQL, et exposez le port 8080 pour *nginx*.
13. Quelle commande permet de déployer un stack Docker à partir d'un fichier `docker-compose.yml` sur un cluster Docker Swarm ?
14. Une fois un stack déployé, quelle commande permet de vérifier l'état des services dans le stack ?
15. Si vous souhaitez mettre à jour un service dans un stack Docker (par exemple, pour changer de version d'image), quelle commande utiliseriez-vous ?
16. Quelle commande permet de supprimer un stack Docker du Swarm ?

Déploiement d'un stack swarm

Le but de ce TP est de fournir les scripts ansible permettant :

- la création d'un cluster swarm,
- le déploiement de l'application développée dans le TP4 en utilisant un stack swarm,
- la suppression du cluster swarm.