IUT de Vélizy-Rambouillet

CAMPUS DE VÉLIZY-VILLACOUBLAY

Durand Antonin
Jougla Maxime
Parciany Benjamin
Zehren William

Compte rendu installation application

Table des matières

I- Introduction

Dans ce rapport sera présenté les différentes étapes, et installations effectuées pour mettre en place le kit Cluster hat ainsi que l'application qui sera hébergée sur ce dernier. En premier lieu, sera montré les différentes étapes pour mettre en route le kit Cluster, de la création de l'iso pour le raspberry principal, en passant par l'allumage des 4 raspberry pi zero, à l'installation de différents utilisateurs et logiciels pour manager ce dernier. Puis, dans un second temps, sera présenté comment l'application est installée, hébergée et gérée sur le kit Cluster à l'aide de docker swarm.

II- Installation du kit Cluster hat

III- Installation de l'application

A) Introduction

Une fois le kit Cluster configuré, et les 4 pi zero accessibles, nous allons installer l'application en utilisant docker swarm. Docker swarm est une fonctionnalité avancée de docker permettant de gérer un cluster de containers et un ensemble de services. Il est composé d'un manager, le RaspberryPi principal dans notre cas, qui s'occupe de gérer les différents workers, et services du docker swarm. Il peut ajouter, supprimer un worker, ajouter, modifier et supprimer un service. Un worker ne possède pas de droits et son seul rôle est d'exécuter les tâches données par le manager, soit un ou plusieurs services. L'avantage de docker swarm est qu'il permet d'assurer la haute disponiblité des services en gérant automatiquement le fail-over et le load-balancing, dans notre cas, l'application doit être disponible en permanance. Dans notre situation, chaque worker est un RaspberryPi zero.

Nous aurons un service web qui sera le front-end et le back-end du site. Il aura 3 réplicas, ce qui permet au site d'être accesssible si l'un des RaspberryPi rencontre une erreur. Ensuite, un autre service sera la base de données MySql de l'application, avec 2 réplicas. Enfin, il faut créer deux autres services identiques aux précédents, mais qui seront utilisés pour tester les changements effectués dans l'application avant son déploiement

B) Création du docker swarm

On installe docker s'il ne l'est pas encore. Ensuite, on initialise un docker swarm, et on rajoute 4 workers, où le manager est le RasperryPi principal, et chaque worker est un RaspberryPi zero. L'objectif ensuite est d'avoir 1 stack contenant l'application déployée et 1 autre stack contenant l'application en production.

C) Création de l'image pour le serveur web

Pour créer le service web de notre application, nous avons besoin d'une image docker personnalisée.

Cette dernière se base sur l'image existante **php:8.2-apache**, car elle contient un serveur apache capable de lire et d'exécuter des fichiers PHP. Elle doit contenir l'ensemble du front et back-end de l'application, soit les fichiers du site web.

Ensuite, on crée notre image personnalisée à l'aide d'un fichier **dockerfile**.

D) Création de l'image pour la base de données

Pour créer le service bd de notre application, nous avons besoin d'une image docker personnalisée.

Cette dernière se base sur l'image existante **mysql**, car elle contient déjà un système de gestion de base de données (SGBD). Elle doit contenir la base de données de notre application.

Ensuite, on crée notre image personnalisée à l'aide d'un fichier **dockerfile**.

E) Création du stack de l'application

Enfin, une fois les images pour la partie web et la partie base de données créées, on utilise docker stack pour créer un stack, collection contenant un service web et un service base de données.

Pour ce faire, nous allons utiliser un fichier de configuration du stack **docker-compose.yml**, qui crée un service en utilisant l'image du serveur web, et un autre en utilisant l'image de la base de données. Dans ce fichier de configuration, on précise aussi les ports des services ainsi que les replicas et le volume pour le service de base de données.