

Projet Docker 1 - Stack Applicative

WordPress et MariaDB

Objectif

Mettre en place une stack applicative composée d'une application frontend WordPress et d'une base de données MariaDB, utilisant Docker et Docker Compose.

Application WordPress

Choix de l'image

- Image de Base : wordpress:latest
 - Justification : L'utilisation de l'image officielle WordPress basée sur une distribution Linux optimisée pour WordPress.

Dockerfile

```
FROM wordpress:latest

# Ajoutez les variables d'environnement pour la base de données
ENV WORDPRESS_DB_HOST=db \
    WORDPRESS_DB_USER=root \
    WORDPRESS_DB_PASSWORD=root \
    WORDPRESS_DB_NAME=wordpress

# Exposez le port par défaut de WordPress
EXPOSE 80

# Restez en tant qu'utilisateur www-data
```

USER www-data

Base de Données MariaDB

Choix de l'image

- Image de Base : mariadb:latest
 - Justification : L'utilisation de l'image officielle MariaDB maintenue par les développeurs de MariaDB.

Fichier de Configuration my.cnf

```
[mysqld]
character-set-server=utf8mb4
collation-server=utf8mb4_unicode_ci
```

Docker Compose

Fichier docker-compose.yml

```
version: '3'

networks:
  frontend_network:
  backend_network:

services:
  wordpress:
    build: .
    image: youssef324/docker-wordpress:latest
    restart: always
    depends_on:
      - db
```

```
environment:
  WORDPRESS_DB_HOST: db
  WORDPRESS_DB_USER: root
  WORDPRESS_DB_PASSWORD: root
  WORDPRESS_DB_NAME: wordpress
ports:
  - "8000:80"
networks:
  - frontend_network
  - backend_network
```

```
db:
  image: mariadb:latest
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: root
    MYSQL_DATABASE: wordpress
  volumes:
    - ./my.cnf:/etc/mysql/my.cnf:ro

  networks:
    - backend_network
```

Réseaux Virtuels Distincts

- Deux réseaux virtuels distincts sont créés :
 - frontend_network utilisé par le frontend uniquement.
 - backend_network utilisé par les deux, la base de données et le frontend.

Projet Docker 2 - Cluster Elasticsearch avec Kibana

Objectif

Mettre en place un cluster Elasticsearch composé de trois nœuds et un conteneur Kibana qui a accès au cluster.

Structure du Projet

```
youssef/  
|  
├─ docker-compose.yml  
├─ .env  
├─ es-cluster/  
    └─ elasticsearch.yml  
    └─ kibana.yml
```

Fichier .env

```
ES_VERSION=8.11.3
```

Configuration du Cluster Elasticsearch

Dans le répertoire 'es-cluster', j'ai créé un fichier elasticsearch.yml avec la configuration Elasticsearch, et un fichier kibana.yml avec la configuration Kibana.

Fichier elasticsearch.yml

```
# Fichier elasticsearch.yml
cluster.name: "projet-cluster"
node.name: ${HOSTNAME}
discovery.seed_hosts: ["es01", "es02", "es03"]
cluster.initial_master_nodes: ["es01", "es02", "es03"]
network.host: 0.0.0.0
```

Fichier kibana.yml

```
# Fichier kibana.yml
server.name: kibana
elasticsearch.hosts: ["http://es01:9200", "http://es02:9200",
"http://es03:9200"]
```

Fichier docker-compose.yml

J'ai créé un fichier docker-compose.yml à la racine du projet avec la configuration suivante :

- Trois services Elasticsearch (es01, es02, es03) qui forment un cluster.
- Un service Kibana (kibana) connecté au cluster Elasticsearch.
- La garantie que les nœuds Elasticsearch utilisent les fichiers de configuration appropriés.
- Configuration des variables d'environnement nécessaires.

docker-compose.yml

```
version: '3'

services:
  # Configuration pour les instances Elasticsearch

  es01:
    image: docker.elastic.co/elasticsearch/elasticsearch:${ES_VERSION}
    container_name: es01
    environment:
      - node.name=es01    # Nom du nœud Elasticsearch
    volumes:
      - ./es-cluster/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml
      # Fichier de configuration Elasticsearch
      - es01-data:/usr/share/elasticsearch/data    # Répertoire de données partagé
    ports:
      - "9200:9200"    # Port d'accès à Elasticsearch

  es02:
    image: docker.elastic.co/elasticsearch/elasticsearch:${ES_VERSION}
    container_name: es02
    environment:
      - node.name=es02
    volumes:
      - ./es-cluster/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml
      - es02-data:/usr/share/elasticsearch/data
    ports:
      - "9201:9200"

  es03:
    image: docker.elastic.co/elasticsearch/elasticsearch:${ES_VERSION}
    container_name: es03
    environment:
      - node.name=es03
```

```
volumes:
  - ./es-cluster/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml
  - es03-data:/usr/share/elasticsearch/data
ports:
  - "9202:9200"

# Configuration pour Kibana

kibana:
  image: docker.elastic.co/kibana/kibana:${ES_VERSION}
  container_name: kibana
  depends_on:
    - es01      # Dépendance sur es01 (Elasticsearch doit être en cours
d'exécution avant Kibana)
  environment:
    - ELASTICSEARCH_URL=http://es01:9200      # URL d'accès à Elasticsearch
pour Kibana
  ports:
    - "5601:5601"      # Port d'accès à Kibana
```

Problèmes rencontrés

- J'ai rencontré des problèmes hier soir avec Docker Desktop et le terminal Ubuntu. Ces outils ne veulent plus démarrer correctement. Je ne pouvais plus rien faire, donc j'ai dû éteindre mon ordinateur et attendre jusqu'à aujourd'hui pour pouvoir reprendre le travail. J'ai eu le même problème avant-hier soir. Ce matin, lorsque je voulais reprendre, je n'ai pas pu même ouvrir Docker Desktop. Le raccourci me dit qu'il a été désinstallé. Je pense que c'étaient des problèmes dans les fichiers parce que hier soir, j'étais en train de supprimer des fichiers et des applications pour libérer de l'espace, mais je n'ai pas touché à des dossiers Docker. Je ne sais pas pourquoi. J'ai dû réinstaller Docker Desktop, même si cela m'a pris beaucoup de temps car je n'ai pas pu le désinstaller, ils me disent qu'ils ne trouvent pas le fichier.
- En ce qui concerne le rendu du projet sur GitHub, j'ai également rencontré des difficultés avec le terminal Ubuntu. Le push ne fonctionne pas, et j'ai rencontré des problèmes liés aux autorisations. J'ai essayé plusieurs approches pour résoudre ces problèmes, mais je n'ai pas trouvé de solution. Je voulais vous envoyer une petite vidéo de démonstration avec une documentation, mais comme je vous ai dit, Docker Desktop et Ubuntu ne veulent plus marcher.

- En ce qui concerne le projet Docker 2, j'ai également été confronté à des problèmes de ressources insuffisantes sur mon ordinateur. Les conteneurs se terminent de manière inattendue avec le code de sortie 137 après un certain temps, ce qui rend le maintien du cluster Elasticsearch et du service Kibana opérationnels difficile. Tout fonctionne correctement au début, mais les conteneurs s'arrêtent inexplicablement après un certain temps.