Docker - SGBD

PROHACKTIVE

Sommaire

1. Pré-requis

Une VM debian 11 servira d'hôte aux conteneurs Dockers.
Pour cette tâche, on installe au préalable les services requis:

Docker et docker-compose:
Installation des paquet nécessaire pour permettre à apt d'utiliser des dépots par https:

```
# apt install ca-certificates curl gnupg lsb-release
```

On ajoute la clé GPG de docker:

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg
--dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

On ajoute les dépôts:

```
$ echo \
  "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

On met à jour la liste des paquets et on install docker:

```
# apt update
# apt install docker-ce docker-ce-cli containerd.io
```

Installation de docker-compose:

```
# curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-compo
se-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
# chmod +x /usr/local/bin/docker-compose
# ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Reverse proxy nginx:
Afin d'accéder aux différents conteneurs, un reverse proxy est installé permettant de pointer chaque sous-domaine vers l'ip interne du conteneur.

```
# apt install nginx
```

Configurations réseau:

Ip : **172.17.0.77**

fichier /etc/hosts :

```
172.17.0.77 cms.phk.lan
```

2. Instances SGBD

Afin que les instances sgbd soient pleinement fonctionnelles, elles seront créé via l'outil docker-compose:

MariaDB:

```
# mkdir ~/mariadb
# cd ~/mariadb
# nano docker-compose.yml
```

```
version: '3.1'

networks:
  static:
    ipam:
      config:
        - subnet: 172.20.0.0/24

services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb
    restart: always
    environment:
      MARIADB_ROOT_PASSWORD: db
      MARIADB_DATABASE: db
      MARIADB_USER: db
      MARIADB_PASSWORD: db
    networks:
      static:
        ipv4_address: 172.20.0.22
```

```
  adminer_mariadb:
    image: adminer:latest
    container_name: adminer_mariadb
    depends_on:
      - dblatest
    restart: always
    networks:
      frontend:
        ipv4_address: 172.20.0.2
```

PS: Ici, la version du sgbd est définie sur latest. Les mêmes étapes seront effectuées avec différentes versions.

Exécution du Docker Compose :

```
# docker-compose up -d
```

MongoDB:

```
# mkdir ~/mongodb
# cd ~/mongodb
# nano docker-compose.yml
```

```
version: '3.1'
networks:
  static:
    ipam:
      config:
        - subnet: 172.21.0.0/24

services:
  mongo:
    image: mongo:latest
    container_name: mongo
    restart: always
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: mongo
    networks:
      static:
        ipv4_address: 172.21.0.22
```

```
  mongo-express:
    image: mongo-express:latest
    container_name: mongo-express
    depends_on:
      - mongo
    environment:
      ME_CONFIG_MONGODB_URL: mongodb://root:mongo@mongo:27017/
    restart: always
    networks:
      static:
        ipv4_address: 172.21.0.2
```

PS: Ici, la version du sgbd est définie sur latest. Les mêmes étapes seront effectuées avec différentes versions.

Exécution du Docker Compose :

```
# docker-compose up -d
```

PostgreSQL:

```
# mkdir ~/postgres
# cd ~/postgres
# nano docker-compose.yml
```

```
version: '3.1'
networks:
  static:
    ipam:
      config:
        - subnet: 172.22.0.0/24

services:
  postgresql:
    image: postgres:latest
    container_name: postgresql
    restart: always
    environment:
      POSTGRES_USER: postgresql
      POSTGRES_PASSWORD: postgresql
      POSTGRES_DB: postgresql
    networks:
      static:
        ipv4_address: 172.22.0.22
```

```
  adminer_postgresql:
    image: adminer:latest
    container_name: adminer_postgresql
    depends_on:
      - postgresql
    restart: always
    networks:
      static:
        ipv4_address: 172.22.0.2
```

PS: Ici, la version du sgbd est définie sur latest. Les mêmes étapes seront effectuées avec différentes versions.

Exécution du Docker Compose :

```
# docker-compose up -d
```

3. Reverse proxy

Une fois les conteneurs lancés, il faut configurer le reverse proxy pour y accéder.

MariaDB:

```
# nano /etc/nginx/sites-available/mariadb.conf
```

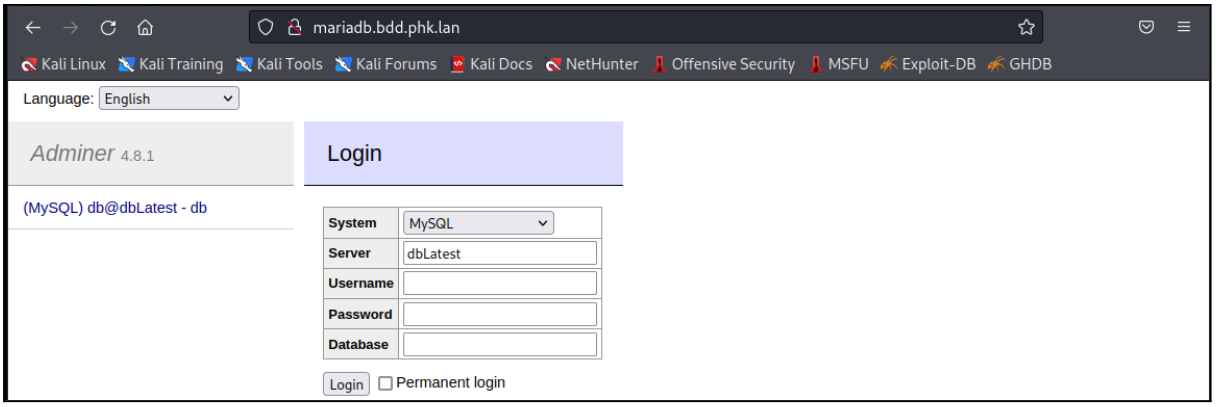
```
server {
      listen 80;
      server_name mariadb.bdd.phk.lan;
      location / {
            proxy_pass        http://172.20.0.2:8080;

            proxy_set_header  X-Real-IP         $remote_addr;
            proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header  X-Forwarded-Proto $scheme;
            proxy_set_header  Host              $host;
            proxy_set_header  X-Forwarded-Host  $host;
            proxy_set_header  X-Forwarded-Port  $server_port;
      }
}
```

*PS: Le DNS est configuré pour pointer *.bdd.phk.lan vers 172.17.0.77*

Activation du site:

```
# ln -s /etc/nginx/sites-available/mariadb.conf ../sites-enabled/
# systemctl restart nginx
```

On accède maintenant au sgbd:
http://mariadb.bdd.phk.lan



MongoDB:

```
# nano /etc/nginx/sites-available/mongodb.conf
```

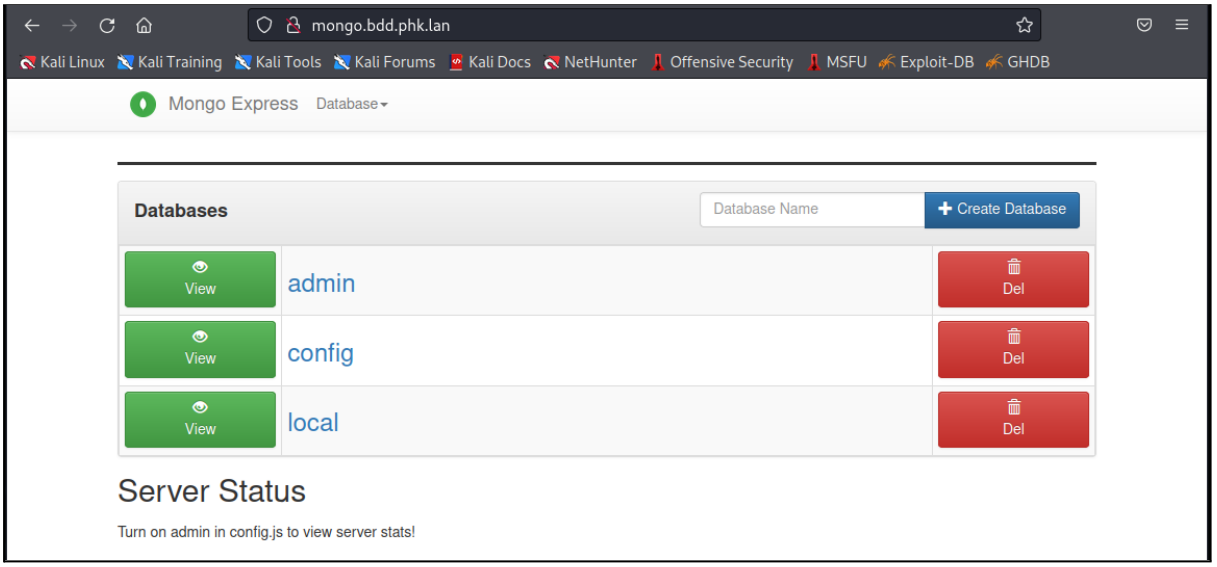
```
server {
      listen 80;
      server_name mongodb.bdd.phk.lan;
      location / {
            proxy_pass        http://172.21.0.2:8080;

            proxy_set_header  X-Real-IP         $remote_addr;
            proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header  X-Forwarded-Proto $scheme;
            proxy_set_header  Host              $host;
            proxy_set_header  X-Forwarded-Host  $host;
            proxy_set_header  X-Forwarded-Port  $server_port;
      }
}
```

*PS: Le DNS est configuré pour pointer *.bdd.phk.lan vers 172.17.0.77*

Activation du site:

```
# ln -s /etc/nginx/sites-available/mongodb.conf ../sites-enabled/
# systemctl restart nginx
```

On accède maintenant au sgbd:
http://mongodb.bdd.phk.lan



PostgreSQL:

```
# nano /etc/nginx/sites-available/postgres.conf
```

```
server {
      listen 80;
      server_name postgres.bdd.phk.lan;
      location / {
            proxy_pass        http://172.22.0.2:8080;

            proxy_set_header  X-Real-IP         $remote_addr;
            proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header  X-Forwarded-Proto $scheme;
            proxy_set_header  Host              $host;
            proxy_set_header  X-Forwarded-Host  $host;
            proxy_set_header  X-Forwarded-Port  $server_port;
      }
}
```

*PS: Le DNS est configuré pour pointer *.bdd.phk.lan vers 172.17.0.77*

Activation du site:

```
# ln -s /etc/nginx/sites-available/postgres.conf ../sites-enabled/
# systemctl restart nginx
```

On accède maintenant au sgbd:
http://postgres.bdd.phk.lan