

Exam - Cloud Computing

14 février 2020

1 Exercice

Vous devez *dockeriser* votre application qui est formée d'un serveur nodeJS et d'une base de donnée neo4J (voir la documentation à la suite). Il faut donc créer 2 dockers et un reseau commun entre eux

Le code de votre application nodeJS se trouve dans le répertoire `./Code/`, elle utilise 4 variables d'environnement :

- `PORT` : le port utilisé par l'application ;
- `NEO4JSERVER` : le serveur de base de donnée de l'application ;
- `USERDB` : l'utilisateur de la base neo4J ;
- `PASSDB` : le mot de passe de cet utilisateur.

Ces 4 variables doivent avoir une valeur sinon, l'application ne démarre pas.

Pour dockeriser une telle application, il faut créer un fichier `Dockerfile` qui lance l'installation de l'application, il est déjà fait et vous n'avez pas à vous en préoccuper. Il est disponible avec le chemin `./Node/Dockerfile`. Mais vous devez faire en sorte qu'il soit utilisé.

- Le réseau doit s'appeler `resApp` et utiliser les adresses `10.10.42.0/24`.
- Le serveur de base de donnée `neo4J` doit :
 - partager les port 7474 et 7687 avec la machine hôte ;
 - utiliser comme volume de données le répertoire `./Data` ;
 - utiliser la version 3.5 de la base ;
 - utiliser un login et mot de passe non évident que vous choisirez.
- Le docker de votre application doit :
 - s'appeler `nodeApp` ;
 - utiliser la base neo4j ;
 - écouter sur le port 80 de la machine hôte ;
 - trouver le code dans le répertoire `/home/node/app` du docker ;
 - lancer la commande `node app.js` pour démarrer.

Q.1) - Donnez le fichier `docker-compose` qui permet de lancer votre application.

1.1 Passage à Kubernetes

Vous devez maintenant déployer votre service sur kubernetes, dans ce cas, il n'est pas possible d'utiliser les partages de volumes ou le dockerfile pour que les conteneurs accèdent aux codes ou aux données. Il faut donc changer la mise en place de ces applications pour initialiser la base de donnée et l'application nodeJS. Pour cela, vous disposez de deux scripts shell qui sont disponibles en ligne :

- http://forge.univ-lyon1.fr/nodeApp/init_base.sh pour initialiser la base
- http://forge.univ-lyon1.fr/nodeApp/init_app.sh pour initialiser l'application

Dans les objets kubernetes que vous définissez, vous avez le droit de choisir les tags, mais vous devez obligatoirement ajouter au moins le tag `dev:moi`

Q.2) - Créez un secret pour stocker le login et mot de passe de la base de donnée.

Q.3) - Créez un déploiement pour lancer une instance de la base de donnée.

Q.4) - Créez un déploiement pour lancer 3 instances de votre application nodeJS.

Q.5) - Créez tous les services nécessaires pour que cela fonctionne et que votre application soit disponible sous le nom `http://monapp.univ-lyon1.fr`

A Documentation de l'image Neo4J

A.1 What is neo4j ?

Neo4j is the world's leading graph database, with native graph storage and processing.

A.2 How to use this image

You can start a Neo4j container like this :

```
docker run \  
  --publish=7474:7474 --publish=7687:7687 \  
  --volume=$HOME/neo4j/data:/data \  
  neo4j
```

which allows you to access neo4j through your browser at <http://localhost:7474>.

This binds two ports (7474 and 7687) for HTTP and Bolt access to the Neo4j API. A volume is bound to /data to allow the database to be persisted outside the container.

By default, this requires you to login with neo4j/neo4j and change the password. You can, for development purposes, disable authentication by passing `--env=NEO4J_AUTH=none` to docker run.