



Compte Rendu Architecture Logicielle

Equipe :

- Marc Perret
- Clément Guihaire
- Louis Ledoux
- Julien Durand

Local

Nous avons tout d'abord créé une machine virtuelle avec un OS Centos 7. Sur celui-ci nous avons téléchargé le projet Netflix, recipes-rss. Nous avons build le projet, il fabrique 2 services sous forme d'application java (.jar). Ces services permettent de récupérer les flux RSS. Ensuite, nous avons téléchargé de la même manière le projet Eureka. Cependant, malgré de nombreux essais et reconfigurations de la VM il était impossible de build le service Eureka. Ce service est un projet open-source développé par Netflix et permet la découverte de service tel que les 2 services rss présentés précédemment.

Les versions non dockerisées n'ont jamais réussi à fonctionner sur nos machines et pourtant nous étions 4 à avoir testé sur différentes machines ce qui augmentait les choix dans les versions de java, de docker, etc.. C'est pour cela que nous avons décidé de passer aux dockers pour avoir des installations propres et partir d'images propres contenant déjà des outils installés avec des bonnes versions.

Docker

Nous avons ensuite essayé de développer notre architecture grâce à différents docker. Ces docker permettent d'avoir un environnement par service.

Eureka

On commence ce dockerfile à partir d'une image java 1.8. On installe tout ce dont on a besoin, en premier wget. On supprime tomcat si il est présent puis on le télécharge avec la version 8.5.0 sur le répertoire maven. Ensuite on télécharge directement le .war de eureka au lieu de build le projet avec gradle qui nous donnait des erreurs. On copie cette archive web dans le webapps du tomcat.

On expose le port 8080 du docker. Celui-ci est redirigé au moment du "run" avec l'option -p 8080:8080 afin qu'on y accède avec l'url "localhost:8080". On lance le script "catalalina.sh" avec l'option "run" dans le cmd afin que le tomcat se lance lors du "docker run"

Middletier

On commence ce dockerfile à partir d'une image ubuntu. On télécharge ce dont on a besoin, java, git et gradle. On télécharge le projet "recipes-rss" sur github, on le build puis on lance l'application java "rss-middletier.jar" en donnant l'adresse du service Eureka. On expose ici les ports 9191 et 9192. Lors du "run" on lie ce service au service tomcat.

Edge

Ce dockerfile fonction exactement comme celui du middletier sauf la commande qui change car on lance cette fois l'application "rss-edge.jar" en lui donnant encore l'adresse du

service Eureka. On expose ici les ports 9090 et 9192. Lors du “run” on lie ce service à tomcat mais aussi à middletier.

Docker Compose

Une fois les trois dockers créés, testés nous voulions simplifier le lancement des dockers. En effet, pour le moment il fallait les lancer à la main, dans un ordre et attendre que celui de tomcat se lance bien en lançant le server pour ensuite lancer les autres.

Pour palier à ce problème nous avons utilisé docker-compose, c’est un programme qui s’occupe de gérer, de surveiller des dockers. Ce programme lit un fichier de règles pour savoir comment lancer les docker. Ce fichier est au format .yaml. En l’occurrence, le fichier n’est pas très conséquent car il contient juste la description des trois commandes run à lancer et dans l’ordre dans lequel on veut les lancer. Chaque run va attendre que le run précédent ait fini de s’initialiser et va se lancer avec les bon links et les bon ports à écouter.