HE-Arc

Projet d'automne No 212

Docker Hub Taxonomy - Cahier des charges

Auteurs :
Pedretti Maël

Responsables :
Pasin Marcelo
Schiavoni Valerio

26.09.2017

Le projet Docker Hub Taxonomy est réalisé dans le cadre du Travail d'Automne, un module de 3ème année de Bachelor au sein de la Haute-École Arc — Ingénierie, section Développement Logiciel et Multimédia. Le présent document décrit le cahier des charges du projet et présente les objectifs.

haute école
neuchâtel berne jura
arc ingénierie
saint-imier le locle delémont

Table des matières

1. Introduction

Le déploiement d'applications sur les nuages de calcul a évolué sur un modèle de virtualisation au niveau du système d'exploitation. On parle alors de conteneurs, qui s'exécutent en tant que processus d'utilisateur de façon complètement isolée. Docker est une technologie qui simplifie le déploiement de conteneurs. Docker propose un dépôt d'images, un système de fichiers et un ensemble de programmes qui automatisent le téléchargement, le lancement et la gestion de conteneurs. Un très grand nombre d'acteurs importants offre la possibilité d'intégrer des conteneurs Docker dans leurs solutions de cloud. Selon des statistiques récentes, Docker offre presque un million d'images différentes, qui ont été installées sur près de 10 milliards d'ordinateurs.

Selon le modèle proposé par docker, une image peut être basée sur une autre. Par exemple, l'image "tomcat"de Docker (serveur populaire d'applications) dépend de l'image "openjdk"(environnement Java) qui dépend ensuite de l'image "alpine" (version populaire de Linux). L'installation est invisible à l'utilisateur et se fait en couches. Le logiciel de gestion de Docker télécharge et installe d'abord l'image "alpine", puis ensuite l'image "openjdk", puis "tomcat". L'image "alpine" n'est pas réinstallée si un autre conteneur est lancé avec une autre image qui en dépend, comme par exemple le serveur web "nginx". Grâce à son modèle en couches, et au fait qu'un conteneur n'est pas plus qu'un processus d'utilisateur, la surcharge imposée par ce modèle de virtualisation est bien inférieure à celle des machines virtuelles, ce qui l'a rendu très populaire.

2. Description

L'objectif de ce projet est de réaliser un crawler capable de parcourir Docker Hub et de"cartographier"le site. Il s'agit en fait de parcourir les images hébergées et de trouver les relations entre ces dernières.

Pour ce faire, il sera nécessaire de faire des recherches sur différentes technologies pour parcourir le site web, recuillir les informations nécessaires puis les afficher. Ces processus sont définis dans les objectifs au chapitre suivant.

3. Objectifs

Les objectifs sont divisés en deux parties. D'abord viennent les objectifs primaires qui sont essentiels pour une application considérée comme terminée. Ensuite les objectifs secondaires sont des fonctionnalités supplémentaires facultatives qui peuvent se greffer au projet.

3.1 Primaires

- Choisir une technologie pour le crawling de pages Web (Python, JavaScript, etc)
- Choisir une technologie pour la visualisation de la forêt (graphviz, arborjs, etc)
- Effectuer une recherche sur les possibilités qu'offre Docker Hub et éviter les blocages
- Effectuer des recherches sur la manière de récupérer les informations des images
- Effectuer des recherches sur la manière de garder les informations sur les images et pages visitées afin de me pas visiter plusieurs fois les mêmes strutures.
- Concrétiser une application qui visite le site Web Docker Hub et qui collecte les informations sur les images

3.2 Secondaires

- Choisir une technologie pour améliorer les performances du crawler
- Choisir une technologie pour le stockage des données (relationnel, noSQL, etc)
- Concrétiser une application qui affiche les forêts et ses informations.
- Choisir une technologie pour effectuer un crawl distribué

3.3 Contraintes

- Blocage éventuel du nombre de requête de la part de Docker Hub
- Stockage des données (Si le site web contient plusieurs To ou Po de données)
- Performances de la machine réalisant le crawl