

Projet Hadoop

Installation

Hadoop Docker

LAM Daphné, BUI Mai Anh

22 Décembre 2019

Table des matières

Sujet	2
Github	2
Environnement	2
Docker	3
Réalisation	3
Installation Docker	3
Création du Dockerfile et des fichiers de configuration	3
Lancement du cluster	4
Test Wordcount	5
Conclusion	5
Sources	6

Sujet

Nous avons choisi de réaliser un projet d'infrastructure : automatisation d'installation d'une technologie. Nous nous sommes intéressées à **l'installation automatique d'un cluster Hadoop par Docker**.

Notre objectif était de créer un cluster Hadoop simple, composé d'un **noeud master** et de **3 noeuds slaves**. Nous avons installé les composants YARN et HDFS. Enfin, nous avons testé notre cluster avec l'exemple du wordcount.

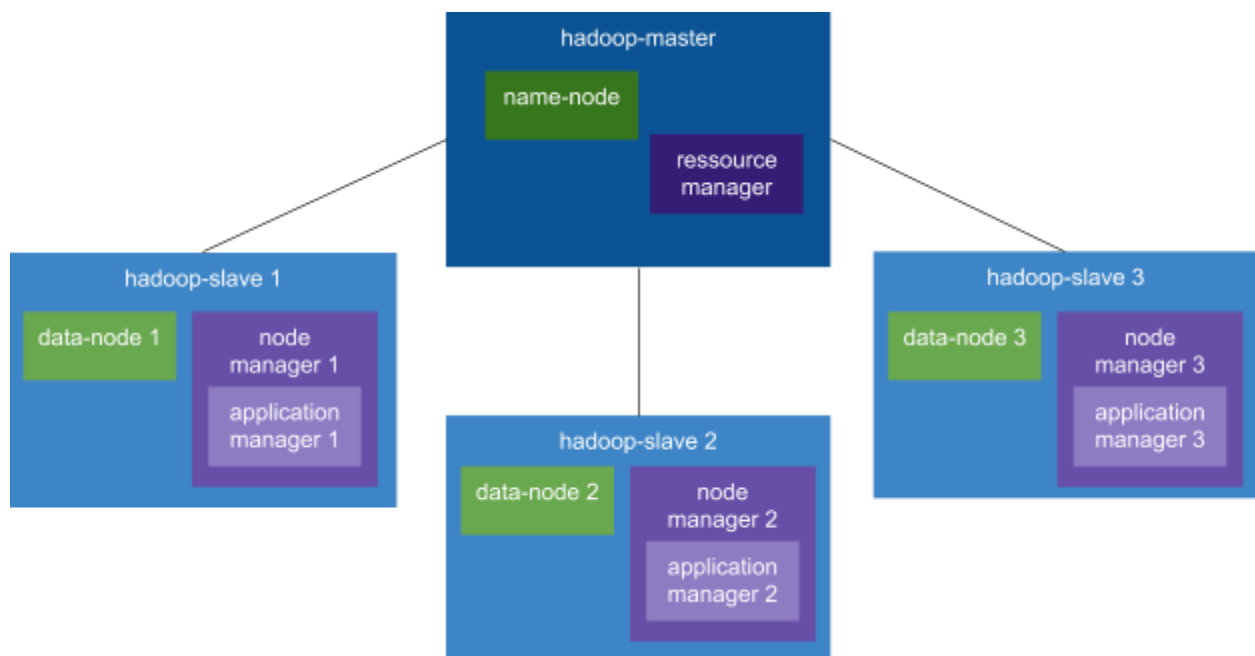


Schéma d'architecture du cluster Hadoop

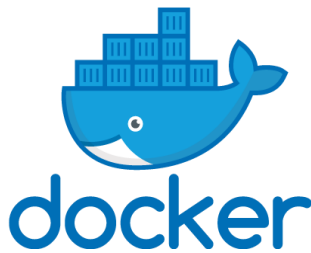
Github

L'ensemble des **fichiers d'infrastructure** utilisés sont disponibles dans le répertoire suivant :
<https://github.com/MaiAnhB/HadoopInstallDocker>

Environnement

Pour ce projet, nous avons utilisé une **machine virtuelle** (distribution Linux : Ubuntu 18.04.3).

Docker



Docker est un système permettant de créer et gérer des environnements (appelés containers) de manière à isoler les applications. Docker repose sur le kernel Linux et sur les containers dans le but de lancer du code dans un environnement isolé. Docker repose aussi sur un le composant control groups (cgroups). Cgroups a pour objectif de gérer les ressources (utilisations de la RAM, CPU).

Docker ne se comporte pas comme une VM qui isole tout son OS et dispose de ses propres ressources. Au contraire, le kernel Docker partage les ressources du système hôte et interagit avec les containers. Pour ainsi dire, Docker permet de lancer un environnement et isoler les composants de ce container avec ceux de l'hôte.

Réalisation

1. Installation Docker

Pour installer Docker, nous avons suivi la procédure officielle de **Docker Community Edition** pour Ubuntu : <https://docs.docker.com/v17.09/engine/installation/linux/docker-ce/ubuntu/>

```
root@maianh-VirtualBox: /home/maianh
File Edit View Search Terminal Help
root@maianh-VirtualBox:/home/maianh# docker --version
Docker version 18.06.0-ce, build 0ffa825
root@maianh-VirtualBox:/home/maianh# docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Capture d'écran de la bonne installation de Docker

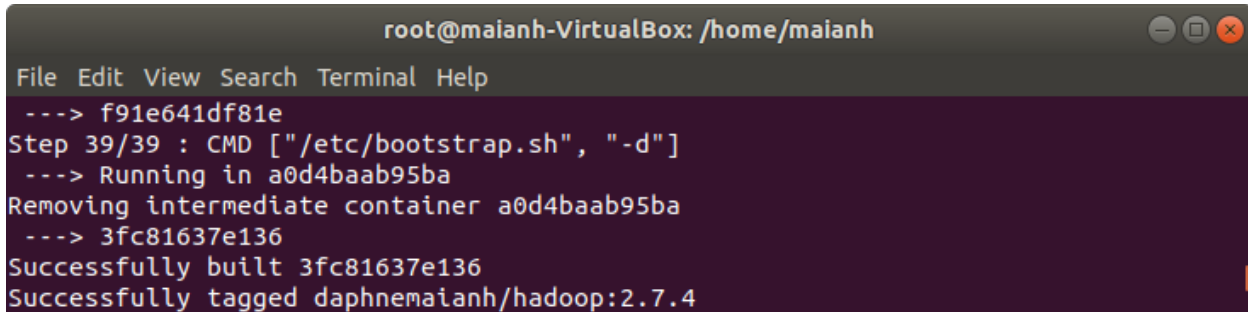
2. Création du Dockerfile et des fichiers de configuration

Le fichier **dockerfile** contient l'ensemble des commandes à exécuter pour installer un cluster Hadoop. Un build de ce fichier permet d'exécuter automatiquement les instructions sur l'OS hôte et produit ce qu'on appelle une image.

Dans notre cas, le Dockerfile permet de :

- Télécharger des packages
- Configurer des variables d'environnement

- Générer des clés SSH
- Installer Hadoop
- Ajouter des fichiers de configuration pour les composants YARN et HDFS



```
root@maianh-VirtualBox: /home/maianh
File Edit View Search Terminal Help
---> f91e641df81e
Step 39/39 : CMD ["/etc/bootstrap.sh", "-d"]
---> Running in a0d4baab95ba
Removing intermediate container a0d4baab95ba
---> 3fc81637e136
Successfully built 3fc81637e136
Successfully tagged daphnemaianh/hadoop:2.7.4
```

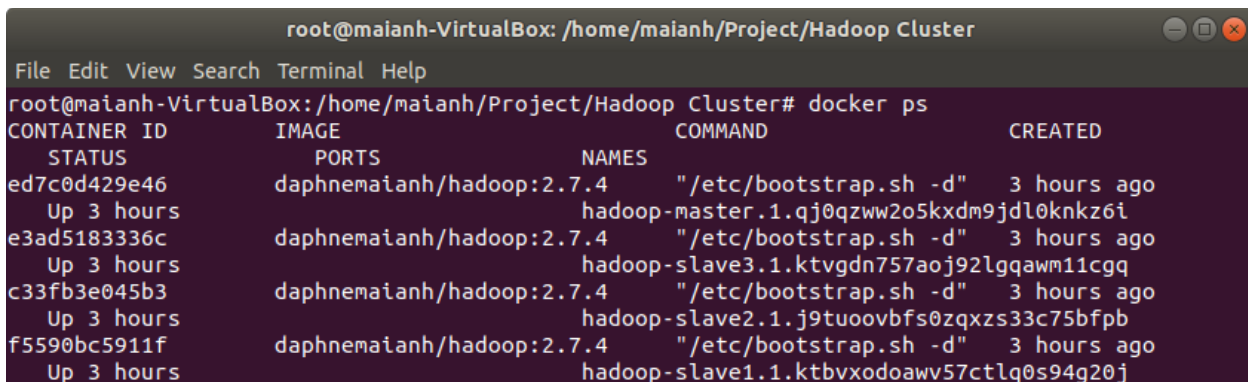
Capture d'écran du succès du build du dockerfile

Le fichier dockerfile, les fichiers de configuration ainsi que le script build_dockerfile.sh sont disponibles dans notre répertoire Github (répertoires *HadoopInstallDocker/Install/Files* et *HadoopInstallDocker/Install/Scripts*).

3. Lancement du cluster

Pour lancer le cluster, il faut créer le **docker swarm cluster** ainsi qu'un **overlay network**. Le docker swarm est l'outil permettant d'assurer la gestion des clusters Docker, le routage, la scalabilité, un déploiement rapide. Le docker swarm gère nativement le load balancing et n'est pas soumis au Single Point Of Failure (SPOF). Un overlay network permet de gérer les communications parmi les docker daemons participant au swarm. Les overlay networks sont les réseaux de docker utilisant un overlay network driver.

Par la suite, nous lançons les docker services des noeuds hadoop-master, hadoop-slave1, hadoop-slave2 et hadoop-slave3.



```
root@maianh-VirtualBox: /home/maianh/Project/Hadoop Cluster
File Edit View Search Terminal Help
root@maianh-VirtualBox:/home/maianh/Project/Hadoop Cluster# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
ed7c0d429e46       daphnemaianh/hadoop:2.7.4   "/etc/bootstrap.sh -d"   3 hours ago
Up 3 hours                hadoop-master.1.qj0qzww2o5kxdm9jdl0knkz6l
e3ad5183336c       daphnemaianh/hadoop:2.7.4   "/etc/bootstrap.sh -d"   3 hours ago
Up 3 hours                hadoop-slave3.1.ktvgn757aoj92lgqawm11cgq
c33fb3e045b3       daphnemaianh/hadoop:2.7.4   "/etc/bootstrap.sh -d"   3 hours ago
Up 3 hours                hadoop-slave2.1.j9tuovbfs0zqxzs33c75bfpb
f5590bc5911f       daphnemaianh/hadoop:2.7.4   "/etc/bootstrap.sh -d"   3 hours ago
Up 3 hours                hadoop-slave1.1.ktbvxodoawv57ctlq0s94g20j
```

Capture d'écran de la liste des containers

Les scripts `start_master_node.sh` et `start_slaves_nodes.sh` sont disponibles dans notre répertoire github (répertoire *HadoopInstallDocker/Install/Scripts*).

4. Test Wordcount

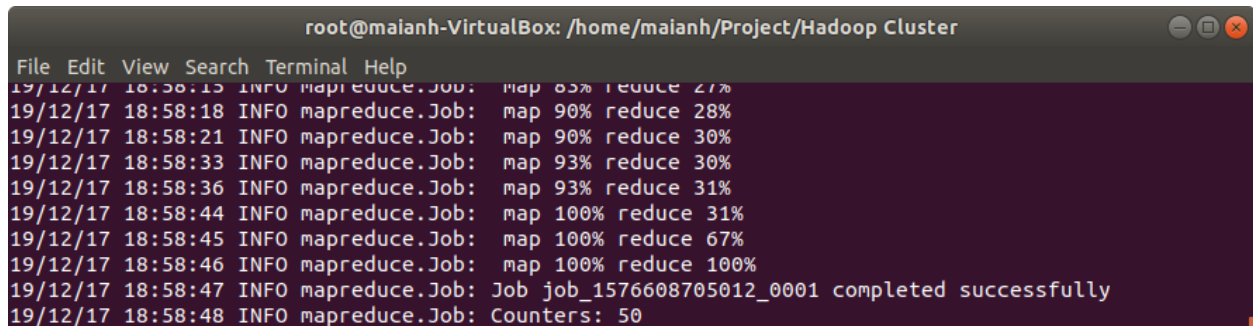
Pour tester le bon fonctionnement de notre cluster Hadoop, nous créons un répertoire HDFS et importons l'ensemble des fichiers du répertoire `etc/hadoop/`.

```
root@maianh-VirtualBox: /home/maianh/Project/Hadoop Cluster
File Edit View Search Terminal Help
bash-4.3# bin/hdfs dfs -mkdir -p /user/root/test
bash-4.3# bin/hdfs dfs -put etc/hadoop/* /user/root/test
```

```
root@maianh-VirtualBox: /home/maianh/Project/Hadoop Cluster
File Edit View Search Terminal Help
bash-4.3# hdfs dfs -ls /user/root/test/
Found 30 items
-rw-r--r--  3 root supergroup      4436 2019-12-17 17:27 /user/root/test/capacity-scheduler.xml
-rw-r--r--  3 root supergroup     1335 2019-12-17 17:27 /user/root/test/configuration.xml
-rw-r--r--  3 root supergroup      318 2019-12-17 17:27 /user/root/test/container-executor.cfg
-rw-r--r--  3 root supergroup      402 2019-12-17 17:27 /user/root/test/core-site.xml
-rw-r--r--  3 root supergroup     3670 2019-12-17 17:27 /user/root/test/hadoop-p-env.cmd
-rw-r--r--  3 root supergroup     4310 2019-12-17 17:27 /user/root/test/hadoop-p-env.sh
-rw-r--r--  3 root supergroup     2490 2019-12-17 17:27 /user/root/test/hadoop-p-metrics.properties
-rw-r--r--  3 root supergroup     2598 2019-12-17 17:27 /user/root/test/hadoop
```

Captures d'écran de l'import des fichiers test dans HDFS

Nous utilisons ensuite un programme de **map reduce** sur ces fichiers : wordcount. Le map reduce est un framework dans laquelle les données sont traitées en parallèle et de façon distribuée. La fonction wordcount exploite le principe de map reduce pour compter le nombre de mots des fichiers répartis dans HDFS.



```
root@maianh-VirtualBox: /home/maianh/Project/Hadoop Cluster
File Edit View Search Terminal Help
19/12/17 18:58:15 INFO mapreduce.Job: map 85% reduce 27%
19/12/17 18:58:18 INFO mapreduce.Job: map 90% reduce 28%
19/12/17 18:58:21 INFO mapreduce.Job: map 90% reduce 30%
19/12/17 18:58:33 INFO mapreduce.Job: map 93% reduce 30%
19/12/17 18:58:36 INFO mapreduce.Job: map 93% reduce 31%
19/12/17 18:58:44 INFO mapreduce.Job: map 100% reduce 31%
19/12/17 18:58:45 INFO mapreduce.Job: map 100% reduce 67%
19/12/17 18:58:46 INFO mapreduce.Job: map 100% reduce 100%
19/12/17 18:58:47 INFO mapreduce.Job: Job job_1576608705012_0001 completed successfully
19/12/17 18:58:48 INFO mapreduce.Job: Counters: 50
```

Capture d'écran de l'exécution du wordcount

Les commandes sont disponibles dans le fichier test_wordcount de notre répertoire github (répertoire *HadoopInstallDocker/Test*).

L'installation d'un cluster Hadoop par Docker a été réussie !

Conclusion

Pour conclure, nous avons pratiqué une installation automatique d'un cluster Hadoop simple. Ce travail nous a permis de comprendre le fonctionnement d'un docker. Ces connaissances nous seront utiles en entreprise. En effet, les outils d'installation automatiques sont et seront de plus en plus indispensables dans les contextes industriels.

Sources

Pour réaliser ce projet, nous avons utilisé les ressources suivantes :

- Article "How to set up a Hadoop Cluster in Docker",
<https://clubhouse.io/developer-how-to/how-to-set-up-a-hadoop-cluster-in-docker/>
- Article "How to quickly setup a Hadoop cluster in Docker",
<https://blog.newnius.com/how-to-quickly-setup-a-hadoop-cluster-in-docker.html>
- Documentation Docker, <https://docs.docker.com/>