

Hadoop

[Apache Hadoop](#) est un framework open-source pour stocker et traiter les données volumineuses sur un cluster. Il est utilisé par un grand nombre de contributeurs et utilisateurs. Il a une licence Apache 2.0.



Hadoop et Docker :

Pour déployer le framework Hadoop, nous allons utiliser des conteneurs [Docker](#). L'utilisation des conteneurs va garantir la consistance entre les environnements de développement et permettra de réduire considérablement la complexité de configuration des machines (dans le cas d'un accès natif) ainsi que la lourdeur d'exécution (si on opte pour l'utilisation d'une machine virtuelle).

Installation :

Nous allons utiliser pour ce projet trois conteneurs représentant respectivement un noeud maître (Namenode) et deux noeuds esclaves (Datanodes).

Nous avons pour cela installé docker sur notre machine, et l'avoir correctement configuré.

Téléchargement de l'image docker uploadée sur dockerhub:

```
C:\Users\pc>docker pull liliasfaxi/spark-hadoop:hv-2.7.2
```

notre image est uploadée :

```
C:\Users\pc>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
redis	latest	e10bd12f0b2d	2 months ago	138MB
postgres	13	d28d5ae8775c	2 months ago	413MB
apache/airflow	2.7.3	56d2fbb018bb	2 months ago	1.54GB
alpine/git	latest	b337a04161f7	22 months ago	38.2MB
liliasfaxi/spark-hadoop	hv-2.7.2	d64a47823a96	4 years ago	1.94GB

Création du réseau nommé hadoop qui permettra de relier les trois conteneurs:

```
C:\Users\pc> docker network create --driver=bridge hadoop
```

le réseau hadoop a été bien crée :

```
C:\Users\pc>docker network ls
NETWORK ID          NAME       DRIVER      SCOPE
f8ee19ffb55c        bridge     bridge      local
fc721d77bbdc        hadoop     bridge      local
64edab956fdd        host       host        local
f8eeff81af43        none       null        local
```

Création des conteneurs :

Conteneur **hadoop-master** :

```
C:\Users\pc>docker run -itd --net=hadoop -p 50070:50070 -p 8088:8088 -p 7077:7077 -p 16010:16010 ^
--name hadoop-master --hostname hadoop-master ^
liliasfaxi/spark-hadoop:hv-2.7.2
```

Conteneur **hadoop-slave1** :

```
C:\Users\pc>docker run -itd -p 8040:8042 --net=hadoop ^
--name hadoop-slave1 --hostname hadoop-slave1 ^
liliasfaxi/spark-hadoop:hv-2.7.2
```

Conteneur **hadoop-slave2** :

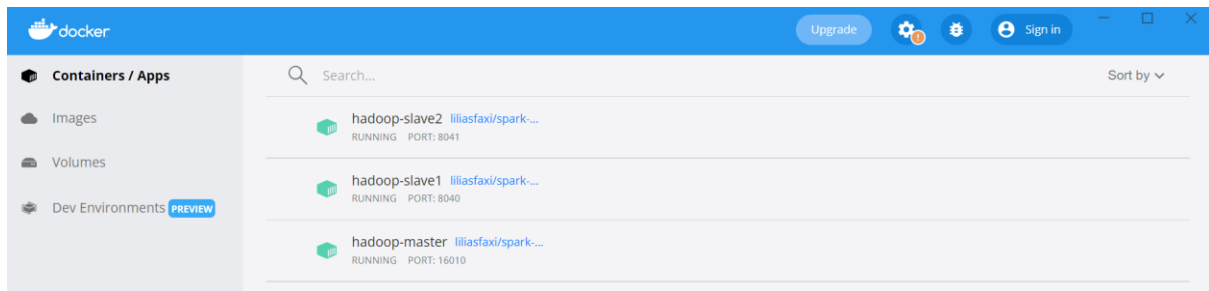
```
C:\Users\pc>docker run -itd -p 8041:8042 --net=hadoop ^
--name hadoop-slave2 --hostname hadoop-slave2 ^
liliasfaxi/spark-hadoop:hv-2.7.2
```

les instructions -p permettent de faire un mapping entre les ports de la machine hôte et ceux du conteneur.

Les 3 Conteneurs ont été bien créés :

```
C:\Users\pc>docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        NAMES
RTS
a3577cec732e   liliasfaxi/spark-hadoop:hv-2.7.2   "sh -c 'service ssh ..." 5 weeks ago   Exited       hadoop-slave2
0.0.0:8041->8042/tcp
447eeebffe8d   liliasfaxi/spark-hadoop:hv-2.7.2   "sh -c 'service ssh ..." 5 weeks ago   Exited       hadoop-slave1
0.0.0:8040->8042/tcp
e29202d6f11f   liliasfaxi/spark-hadoop:hv-2.7.2   "sh -c 'service ssh ..." 5 weeks ago   Exited       hadoop-master
0.0.0:7077->7077/tcp, 0.0.0:8088->8088/tcp, 0.0.0:16010->16010/tcp, 0.0.0:50070->50070/tcp
0737b8bbe864   liliasfaxi/spark-hadoop:hv-2.7.2   "sh -c 'service ssh ..." 6 weeks ago   Exited       condensing_wilbur
1ad89847a0ce   alpine/git                          "git clone https://g..." 21 months ago Exited       repo
```

Nos 3 conteneurs sont actives et prêtes à être utilisés :



On entre dans le conteneur master pour commencer à l'utiliser :

```
C:\Users\pc> docker exec -it hadoop-master bash
root@hadoop-master:~#
root@hadoop-master:~#
```

Nous nous retrouvons dans le shell du namenode, et nous pourrions maintenant manipuler le cluster à notre guise. La première chose à faire, une fois dans le conteneur, est de lancer hadoop et yarn :

```
root@hadoop-master:~# ./start-hadoop.sh

Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.4' (ECDSA) to the list of known hosts.
hadoop-master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-hadoop-master.out
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave1.out
hadoop-slave2: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave2.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-hadoop-master.out

starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-hadoop-master.out
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave1.out
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave2.out
```

Création du répertoire data dans HDFS :

```
root@hadoop-master:~# hadoop fs -mkdir -p data
root@hadoop-master:~# hadoop fs -ls
Found 5 items
drwxr-xr-x  - root supergroup          0 2024-01-13 19:37 data
-rw-r--r--  2 root supergroup          0 2023-12-03 22:34 file1.txt
drwxr-xr-x  - root supergroup          0 2023-12-03 15:18 input
drwxr-xr-x  - root supergroup          0 2023-12-03 16:09 inputs
drwxr-xr-x  - root supergroup          0 2023-12-03 16:06 root
root@hadoop-master:~#
```

Copier les fichiers csv scrapés de ma machine locale vers le conteneur **hadoop-master** :

```
C:\Users\pc>docker cp Downloads/commitments_cleaned.csv hadoop-master:/

C:\Users\pc>docker cp Downloads/question_ecrites.csv hadoop-master:/
CreateFile C:\Users\pc\Downloads\question_ecrites.csv: The system cannot find the file specified.

C:\Users\pc>docker cp Downloads/question_e`crites.csv hadoop-master:/

C:\Users\pc>docker cp Downloads/question_orales.csv hadoop-master:/

C:\Users\pc>
```

On revient maintenant vers hadoop-master pour vérifier si les fichiers existent bien dans le namenode :

```
root@hadoop-master:~# ls ..
bin      commitments_cleaned.csv  etc      lib      media    opt      question_ecrites.csv  root  sbin  sys  usr
boot     dev                    home     lib64    mnt      proc     question_orales.csv  run   srv   tmp  var
root@hadoop-master:~#
```

Copier les fichiers csv du **hadoop-master** vers **HDFS** dans le répertoire **data**:

```
root@hadoop-master:~# hadoop fs -copyFromLocal /commitments_cleaned.csv /data/
root@hadoop-master:~# hadoop fs -copyFromLocal /question_orales.csv /data/
root@hadoop-master:~# hadoop fs -copyFromLocal /question_ecrites.csv /data/
root@hadoop-master:~#
```

Nos fichiers existent bien dans le répertoire **data** dans HDFS:

```
root@hadoop-master:~# hadoop fs -ls /data
Found 3 items
-rw-r--r--  2 root supergroup      644196 2024-01-13 20:26 /data/commitments_cleaned.csv
-rw-r--r--  2 root supergroup    23625795 2024-01-13 20:43 /data/question_ecrites.csv
-rw-r--r--  2 root supergroup     9143517 2024-01-13 20:36 /data/question_orales.csv
root@hadoop-master:~#
```

Interfaces web pour Hadoop :

Hadoop offre plusieurs interfaces web pour pouvoir observer le comportement de ses différentes composantes. Vous pouvez afficher ces pages en local sur votre machine grâce à l'option `-p` de la commande `docker run`. En effet, cette option permet de publier un port du conteneur sur la machine hôte. Pour pouvoir publier tous les ports exposés, vous pouvez lancer votre conteneur en utilisant l'option `-P`.

En regardant le contenu du fichier `start-container.sh` fourni dans le projet, vous verrez que deux ports de la machine maître ont été exposés:

- Le port 50070: qui permet d'afficher les informations du namenode.
- Le port 8088: qui permet d'afficher les informations du resource manager de Yarn et visualiser le comportement des différents jobs.

Notre cluster est lancé et prêt à l'emploi sur : <http://localhost:50070>.

Started:	Sat Jan 13 19:10:32 UTC 2024
Version:	2.7.2, rUnknown
Compiled:	2016-05-27T18:05Z by root from Unknown
Cluster ID:	CID-b721bea8-93cb-45f0-9023-dff705808b00
Block Pool ID:	BP-195783961-172.17.0.3-1550840521902

Summary

Security is off.
Safemode is off.
14 files and directories, 4 blocks = 18 total filesystem object(s).
Heap Memory used 129.71 MB of 232.5 MB Heap Memory. Max Heap Memory is 889 MB.

Dans la section summary on obtient des informations supplémentaires telles que :

- Distributed File System(DFS) used.
- DFS remaining
- live nodes
- nombre de blocs

←

→

↺

localhost:50070/dfshealth.html#tab-overview

🔍

☆

📁

🖨

⌵

Google

1.1.1 Big Data Era O...

2001.05140.pdf

2012.09699.pdf

Summary

Security is off.

Safemode is off.

14 files and directories, 4 blocks = 18 total filesystem object(s).

Heap Memory used 129.71 MB of 232.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 53.56 MB of 54.63 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	501.96 GB
DFS Used:	81.92 MB (0.02%)
Non DFS Used:	39.35 GB
DFS Remaining:	462.54 GB (92.15%)
Block Pool Used:	81.92 MB (0.02%)
DataNodes usages% (Min/Median/Max/stdDev):	0.02% / 0.02% / 0.02% / 0.00%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0

Datanodes :

←

→

↺

localhost:50070/dfshealth.html#tab-datanode

🔍

☆

📁

🖨

⌵

⋮

Google

1.1.1 Big Data Era O...

2001.05140.pdf

2012.09699.pdf

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities -

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
hadoop-slave2.50010 (172.18.0.2.50010)	1	In Service	250.98 GB	40.96 MB	19.67 GB	231.27 GB	4	40.96 MB (0.02%)	0	2.7.2
hadoop-slave1.50010 (172.18.0.3.50010)	0	In Service	250.98 GB	40.96 MB	19.67 GB	231.27 GB	4	40.96 MB (0.02%)	0	2.7.2

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction
------	--------------	-------------------------	------------------------------	--

Sous **Utilities** on peut voir les informations de notre **DFS** :

localhost:50070/explorer.html#/

Google 1.1.1 Big Data Era O... 2001.05140.pdf 2012.09699.pdf

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse the file system
Logs

Browse Directory

/ Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	13/01/2024, 21:43:36	0	0 B	data
drwxr-xr-x	root	supergroup	0 B	03/12/2023, 16:18:06	0	0 B	user

Sous le repertoire **data** on trouve nos fichiers csv :

localhost:50070/explorer.html#/data

Google 1.1.1 Big Data Era O... 2001.05140.pdf 2012.09699.pdf

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/data Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-xr-x	root	supergroup	629.1 KB	13/01/2024, 21:26:43	2	128 MB	commitments_cleaned.csv
-rwxr-xr-x	root	supergroup	22.53 MB	13/01/2024, 21:43:36	2	128 MB	question_ecrites.csv
-rwxr-xr-x	root	supergroup	8.72 MB	13/01/2024, 21:36:04	2	128 MB	question_orales.csv

On peut également visualiser l'avancement et les résultats de nos Jobs sur :
<http://localhost:8088>

localhost:8088/cluster

Google 1.1.1 Big Data Era O... 2001.05140.pdf 2012.09699.pdf

hadoop

Logged in as: dr.who

All Applications

Cluster

- About
- Nodes
- Node Labels
- Applications
- NEW
- NEW SAVING
- SUBMITTED
- ACCEPTED
- RUNNING
- FINISHED
- FAILED
- KILLED
- Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	16 GB	0 B	0	16	0	2	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
No data available in table											

Showing 0 to 0 of 0 entries

First Previous Next Last

