

Open in app ↗



Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Setting up Hadoop with Docker and using MapReduce framework



Guillermo Velazquez · [Follow](#)

5 min read · Feb 11, 2022



Listen



Share

... More

Prerequisites

Have Git and Docker installed

1.Clone docker-hadoop repo from github

Go to this link <https://github.com/big-data-europe/docker-hadoop> and clone the repository, if you're using console, you can type the next command:

git clone <https://github.com/big-data-europe/docker-hadoop>

```
PS C:\Users\TE502801\Desktop\Uni\big data> git clone https://github.com/big-data-europe/docker-hadoop
Cloning into 'docker-hadoop'...
remote: Enumerating objects: 539, done.
Remote: Total 539 (delta 0), reused 0 (delta 0), pack-reused 539
Receiving objects: 100% (539/539), 111.09 KiB | 861.00 KiB/s, done.
Resolving deltas: 100% (240/240), done.
PS C:\Users\TE502801\Desktop\Uni\big data> |
```

And you will get the next folder structure

.git	2/10/2022 9:56 AM	File folder	
base	2/10/2022 9:56 AM	File folder	
datanode	2/10/2022 9:56 AM	File folder	
historyserver	2/10/2022 9:56 AM	File folder	
namenode	2/10/2022 9:56 AM	File folder	
nginx	2/10/2022 9:56 AM	File folder	
nodemanager	2/10/2022 9:56 AM	File folder	
resource manager	2/10/2022 9:56 AM	File folder	
submit	2/10/2022 9:56 AM	File folder	
.gitignore	2/10/2022 9:56 AM	Text Document	1 KB
docker-compose.yml	2/10/2022 9:56 AM	Yaml Source File	2 KB
docker-compose-v3.yml	2/10/2022 9:56 AM	Yaml Source File	3 KB
hadoop.env	2/10/2022 9:56 AM	ENV File	3 KB
Makefile	2/10/2022 9:56 AM	File	2 KB
README.md	2/10/2022 9:56 AM	Markdown Source...	3 KB

2. Start the necessary containers using docker-compose

Run the next command:

```
docker-compose up -d
```

This will start the five necessary containers (if it is the first time you're doing this, you'll have to wait until the download finish)

Docker Compose allows us to run multi-container Docker applications and use multiple commands using only a YAML file.

So calling the previous command will run all the lines in our docker-compose.yml file, and will download the necessary images, if needed.

```
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> docker-compose up -d
[+] Running 6/6
- Network docker-hadoop_default Created                                0.6s
- Container namenode Started                                           4.1s
- Container nodemanager Started                                        3.6s
- Container resource manager Started                                   2.8s
- Container historyserver Started                                     3.6s
- Container datanode Started                                           3.0s
```

You can check if all the five containers are running typing “docker ps” command.

3. Get in our master node “namenode”

```
docker exec -it namenode bash
```

We're getting in our namenode container, which is the master node of our Hadoop cluster. It is basically a mini-linux. It will allow us to manage our HDFS file system.

```
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> docker exec -it namenode bash
root@55f096a5f729:/# |
```

4. Create folder structure to allocate input files

First, you can list all the files in our HDFS system

```
hdfs dfs -l /
```

Now, we have to create a /user/root/ file, since hadoop works with this defined structure

```
hdfs dfs -mkdir -p /user/root
```

We can verify if it was created correctly

```
hdfs dfs -ls /user/
```

```
root@55f096a5f729:/# hdfs dfs -ls /user/
Found 1 items
drwxr-xr-x  - root supergroup          0 2022-02-09 05:49 /user/root
root@55f096a5f729:/# |
```

5. Download MapReduce script

We will use a .jar file containing the classes needed to execute MapReduce algorithm. You can do this manually, compiling the .java files and zipping them. But we will download the .jar file ready to go

Go to: <https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-mapreduce-examples/2.7.1/> and download the one named **hadoop-mapreduce-examples-2.7.1-sources.jar**

6. Download or create the .txt file that you want to process

Most of the time we will use Hadoop to process a fairly large file, maybe several gigabytes, but for this occasion we will use a 1MB file, the classic book “Don Quijote de La Mancha” as plain text, you can use any text you want.

You can download the text file I used here:

https://gist.github.com/jsdario/6d6c69398cb0c73111e49f1218960f79#file-el_quijote.txt

7. Move our .jar & .txt file into the container

First, move these from where you downloaded them and put them in the cloned repository folder. Then type the next command (you have to be in your normal terminal, not inside the namenode container, you can use “exit” command).

```
docker cp hadoop-mapreduce-examples-2.7.1-sources.jar namenode:/tmp
```

Do the same for the .txt file

```
docker cp el_quijote.txt namenode:/tmp
```

```
root@55f096a5f729:/# exit
exit
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> docker cp hadoop-mapreduce-examples-2.7.1-sources.jar namenode:/tmp
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> docker cp el_quijote.txt namenode:/tmp
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> |
```

8. Create the input folder inside our namenode container

Get in the namenode container again

```
docker exec -it namenode bash
```

Then

```
hdfs dfs -mkdir /user/root/input
```

```
root@55f096a5f729:/# hdfs dfs -mkdir /user/root/input
root@55f096a5f729:/# hdfs dfs -ls /user/root/
Found 1 items
drwxr-xr-x - root supergroup 0 2022-02-10 19:03 /user/root/input
root@55f096a5f729:/# |
```

9. Copy your .txt file from /tmp to the input file

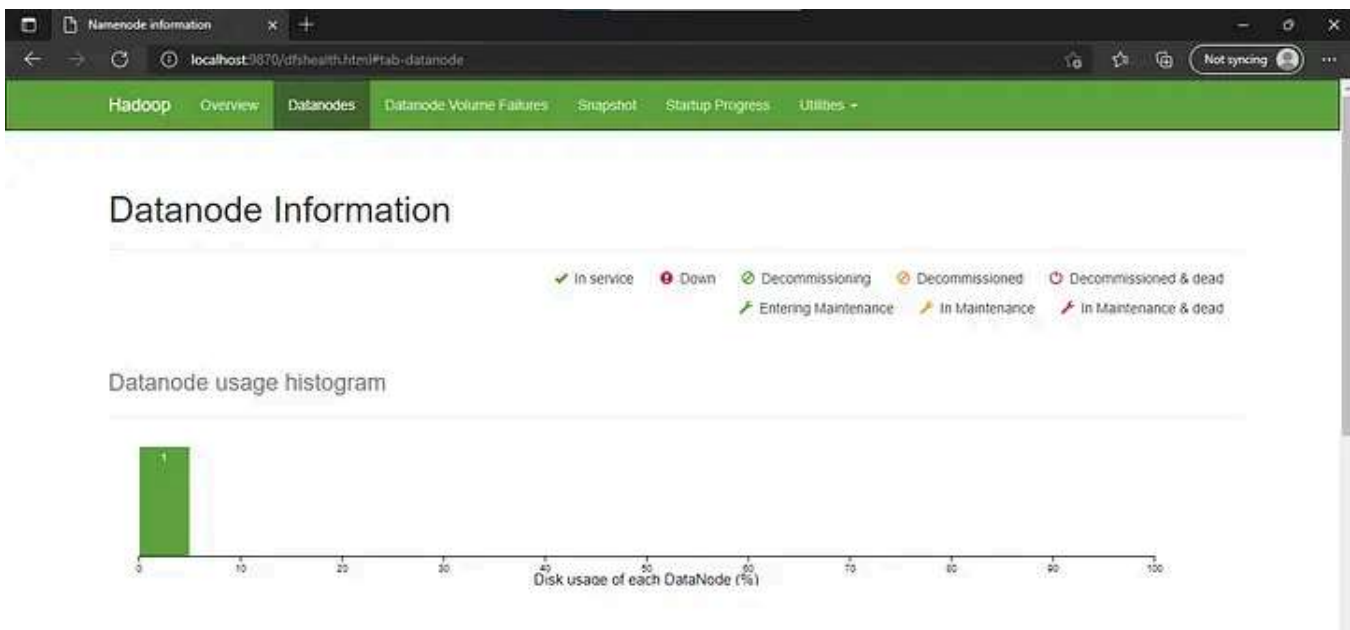
First cd to /tmp (cd /tmp)

```
hdfs dfs -put el_quijote.txt /user/root/input
```

```
root@55f096a5f729:/# cd tmp
root@55f096a5f729:/tmp# hdfs dfs -put el_quijote.txt /user/root/input
2022-02-10 19:07:15,293 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
root@55f096a5f729:/tmp# hdfs dfs -ls /user/root/input
Found 1 items
-rw-r--r-- 3 root supergroup 1060259 2022-02-10 19:07 /user/root/input/el_quijote.txt
root@55f096a5f729:/tmp# |
```

9.1 Visualize the dashboard in your localhost

Access to localhost:9870 in your browser



In this case there is just one node (our computer) in later posts we will see how to add more nodes.

10. Run MapReduce

```
hadoop jar hadoop-mapreduce-examples-2.7.1-sources.jar  
org.apache.hadoop.examples.WordCount input output
```

You'll see a large input, but if you have done the past steps right everything should be fine.

11. See the results

```
hdfs dfs -cat /user/root/output/*
```

Depending on your text file, you'll see something like this

```

Ya 2
-Yo 5
-[Pase] 1
-a ñ adi ó 1
-despu é s 1
-di jo 61
-di jo, 1
-dir á n- 1
-pregunt ó 6
-prosigui ó 1
-que 1
-replic ó 13
-respondi ó 53
-¡Bendito 1
-¡Defi é ndete, 1
-¡Donosa 1
-¡V á late 1
-¡Vos 1
-¿Carta 1
-¿C ó mo 3
-¿De 1
-¿Por 1
-¿Qu é 4
-¿Sabes 1
-¿Tan 1
-¿Y 2
‘ ‘ ¡Ah, 1
“ é ste 1
root@55f096a5f729:/tmp# |

```

You can check the results accesing to the output folder

`hdfs dfs -ls /user/root/putput`

```

root@fb3a39d66140:/# hdfs dfs -ls /user/root/output
Found 2 items
-rw-r--r--  3 root supergroup      0 2022-02-10 19:11 /user/root/output/_SUCCESS
-rw-r--r--  3 root supergroup 257233 2022-02-10 19:11 /user/root/output/part-r-000000
root@fb3a39d66140:/#

```

We're interested in the part-r-00000 file, wich has our word count.

We can export this file putting it in a txt file and moving to our base folder

`hdfs dfs -cat /user/root/output/part-r-00000 > /tmp/quijote_wc.txt`

`exit`

`docker cp namenode:/tmp/quijote_wc.txt .`

Now you will have the text file in your repository folder

12. Turn down the containers

`docker-compose down`

```
root@55f096a5f729:/tmp# exit
exit
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> docker-compose down
[+] Running 6/6
 - Container historyserver      Removed                  14.3s
 - Container namenode          Removed                  11.5s
 - Container datanode           Removed                  14.3s
 - Container nodemanager        Removed                  11.1s
 - Container resourcemanager     Removed                  14.3s
 - Network docker-hadoop_default Removed                   0.7s
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> |
```

What you can do next

Now that you have your word count file, you can do several things to improve it.

- Modify the Java class to do some text preprocessing. Things like remove punctuation marks, so the word “dijo” is not different from “dijo,”.
- Add more nodes to process larger files.
- Create a WordCloud map, to illustrate your work. I will leave an example at the end.

This is it, thank you for reading!!

