Setting up Hadoop with Docker and using MapReduce framework

Guillermo Velazquez · Follow

5 min read · Feb 10, 2022

Prerequisites

Have Git and Docker installed

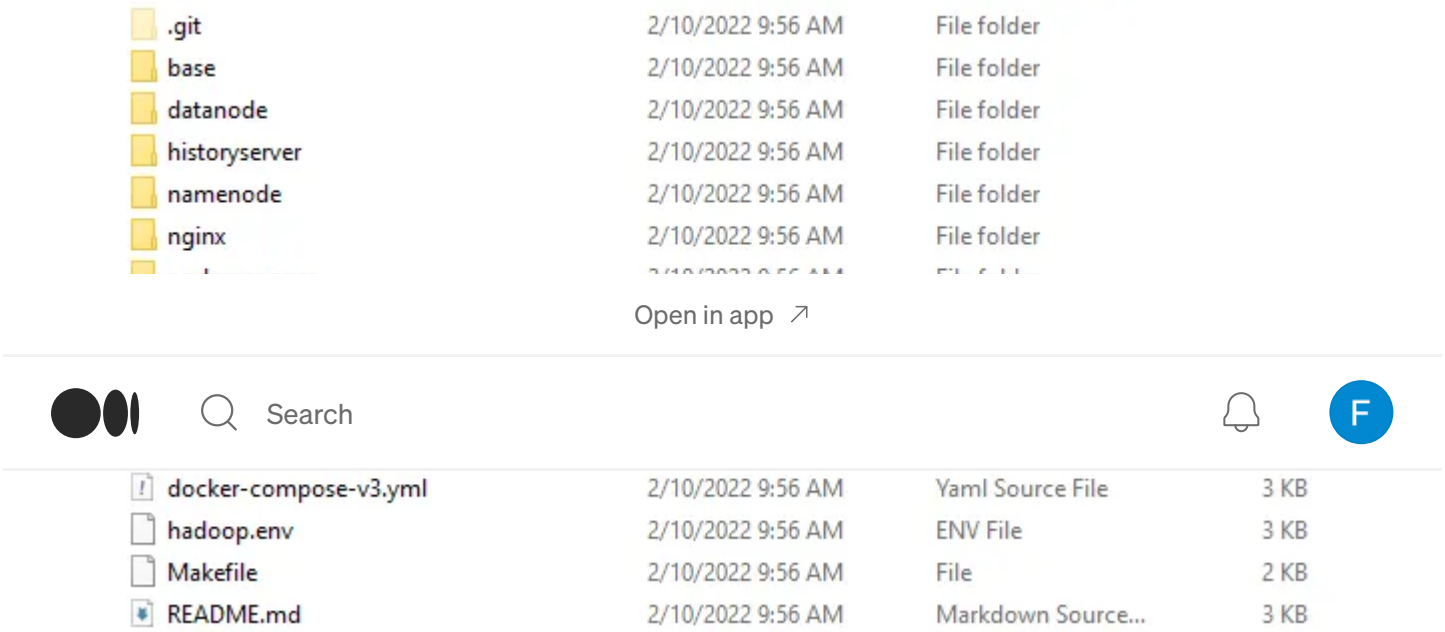
1.Clone docker-hadoop repo from github

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.
Rremote: Total 539 (delta 0), reused 0 (delta 0), pack-reused 539eceiving objects
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



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 resourcemanager      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-sour
ces.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:/#
```

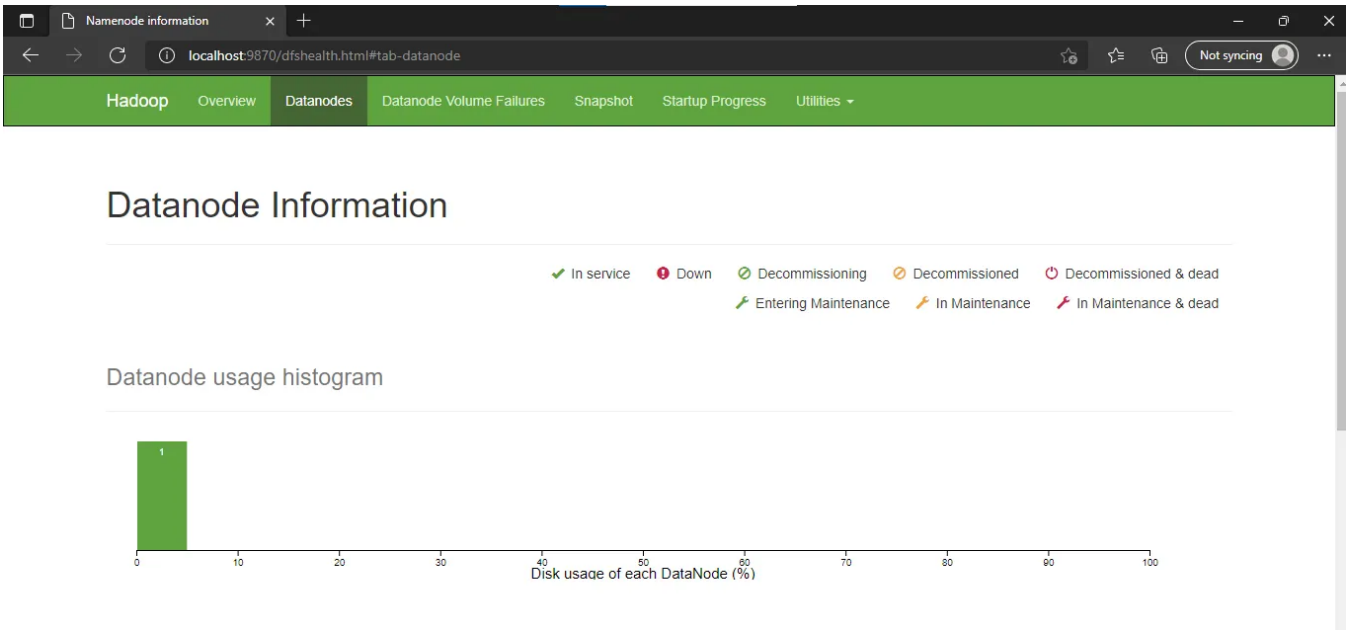
9. Copy your .txt file from /tmp to the input file

Firts 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

Acces to localhost:9870 in your browser



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

10. Run MapReduce

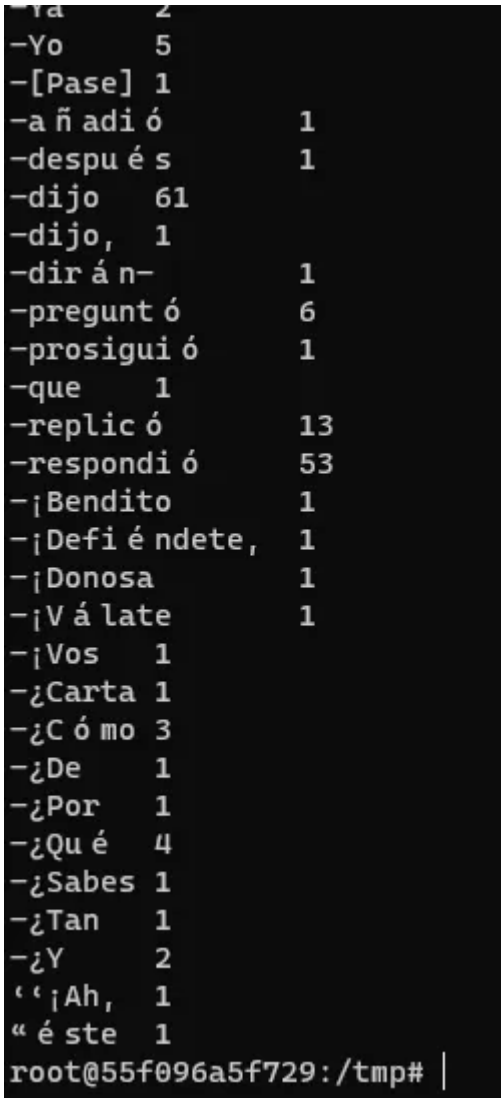
hadoop jar hadoop-mapreduce-examples-2.7.1-sources.jar org.apache.hadoop.examples.WordCount input output

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, youll see something like this



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-00000
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

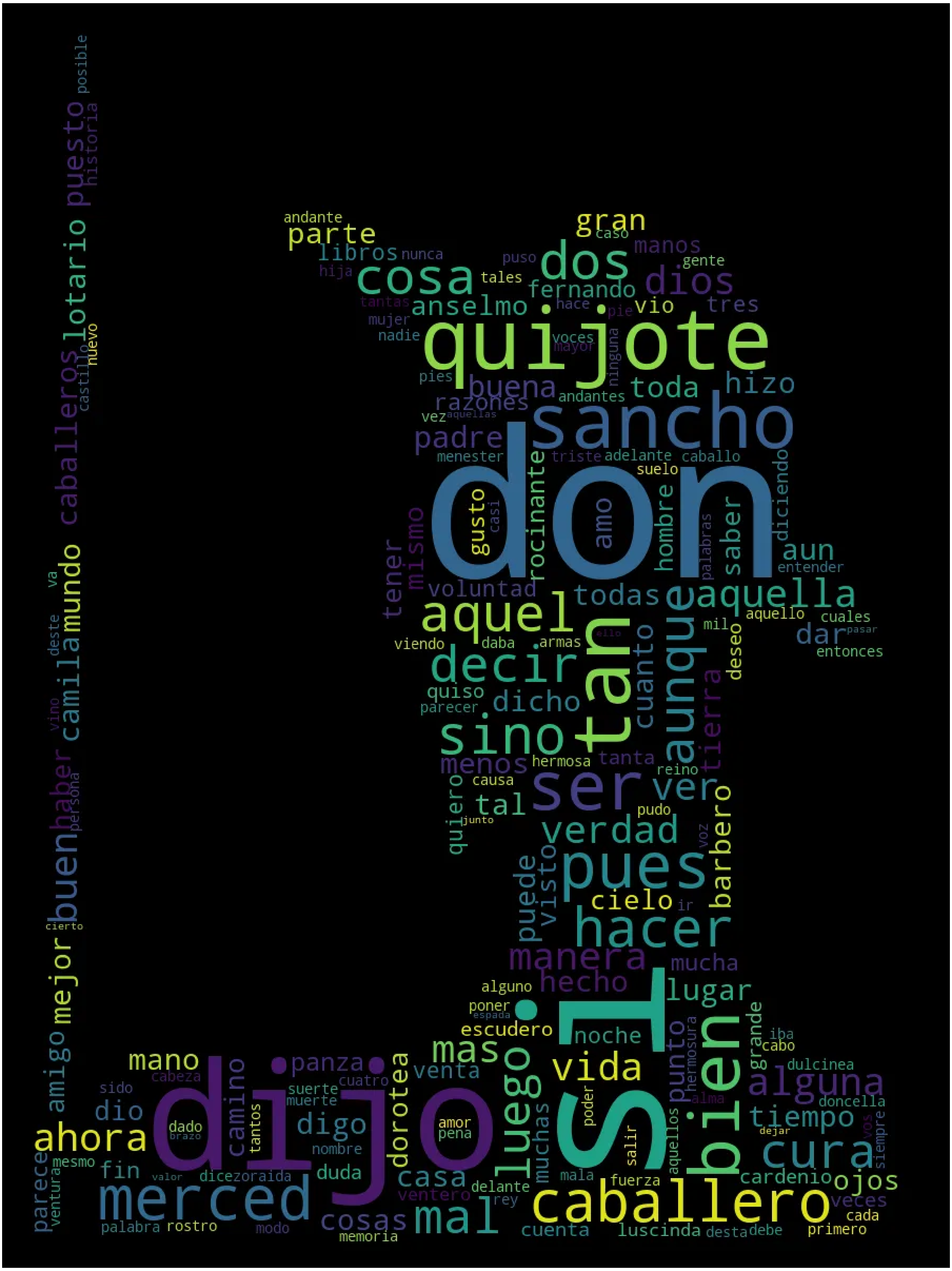
```
root@55f096a5f729:/tmp# exit
exit
PS C:\Users\TE502801\Desktop\Uni\big data\docker-hadoop> docker-compose down
[+] Running 6/6
 - Container historyserver        Removed                                14.3s
 - Container namenode             Removed                                11.5s
 - Container 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 ilustrate your work. I will leave an example at the end.

This is it, thank you for reading!!



Docker Docker Compose Hadoop Word Count Word Cloud

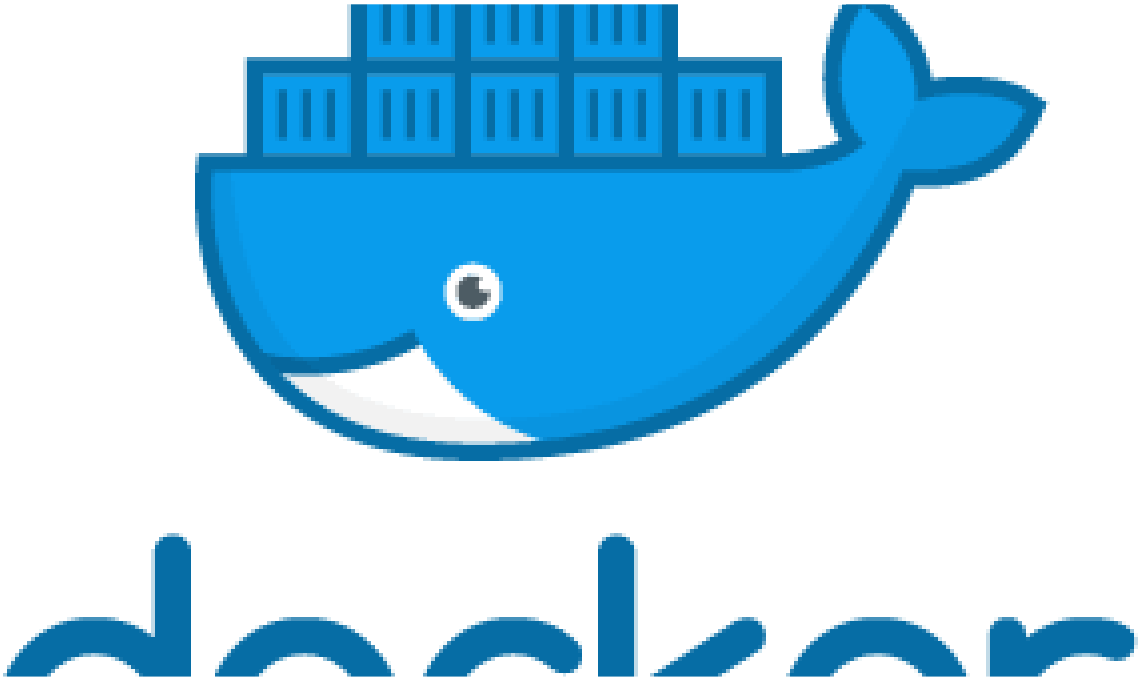
Follow

Written by Guillermo Velazquez

6 Followers

IA Software Developer

More from Guillermo Velazquez



Guillermo Velazquez

Docker para tus Jupyter Notebooks de Deep Learning y Data Science

Tabla de contenidos:

7 min read · Jul 17

Guillermo Velazquez in LCC-Unison

Calcula cuántos años durará tu matrimonio con IA | Un vistazo a los casos de divorcio mexicanos

Sin más preámbulos, “calcula” aquí cuantos años durará tu matrimonio.

5 min read · Dec 9, 2021

See all from Guillermo Velazquez

Recommended from Medium

Tirth Shah

Word Count in Apache Hadoop (MapReduce)

Due to rise and surge in the amount of data being generated nowadays is leading a major challenge to the software companies to handle and...

7 min read · Aug 3

Clément Delteil 🌱 in Towards AI

Real-Time Sentiment Analysis with Docker, Kafka, and Spark Streaming

A Step-By-Step Guide to Deploying a Pre-trained Model in an ETL Process

12 min read · May 6

 2

Lists

Coding & Development

11 stories · 226 saves

New_Reading_List

174 stories · 154 saves

Natural Language Processing

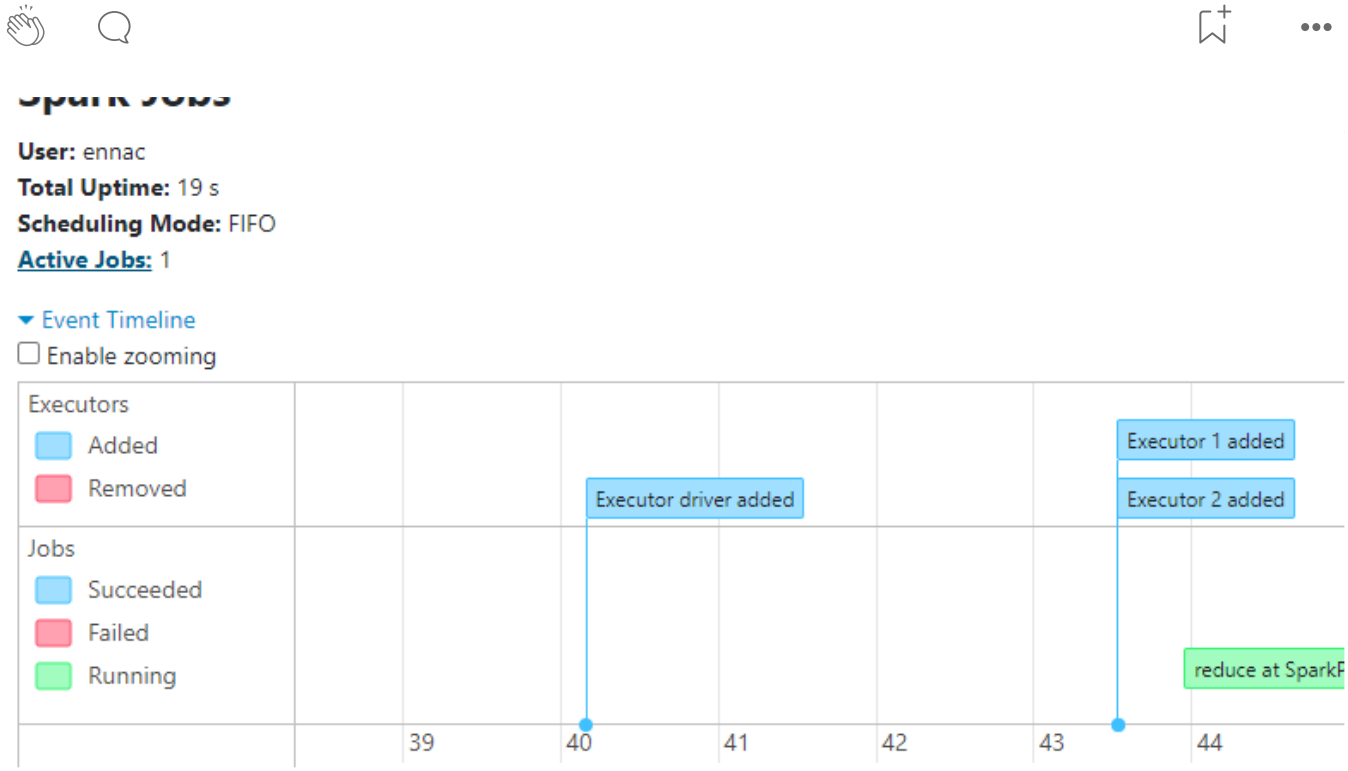
739 stories · 328 saves

 Aysel Aydin

2— Stemming & Lemmatization in NLP: Text Preprocessing Techniques

In the previous article, we explained the importance of text preprocessing and explained some of the text preprocessing techniques. Click...

4 min read · Oct 11



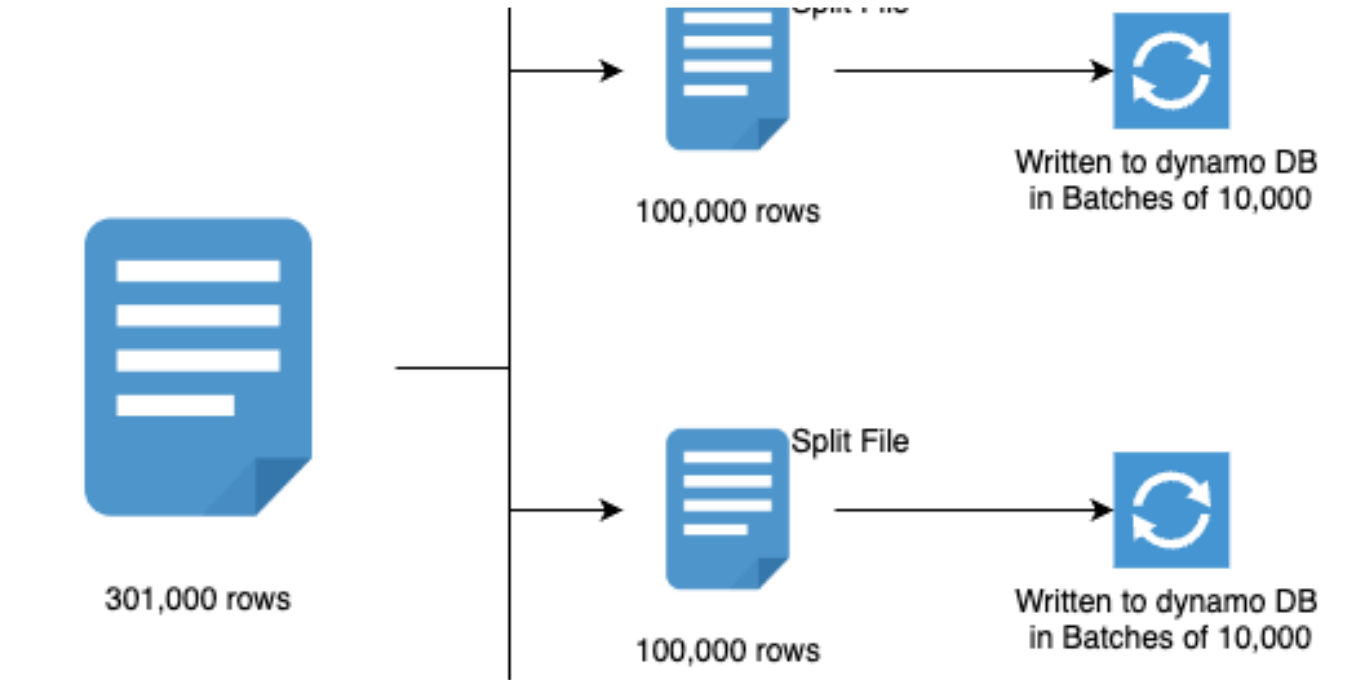
Safouane Ennasser

Deploying Apache Spark on a Local Kubernetes Cluster: A Comprehensive Guide

This is the 2/3 part of the artice series

15 min read · Jun 20

14



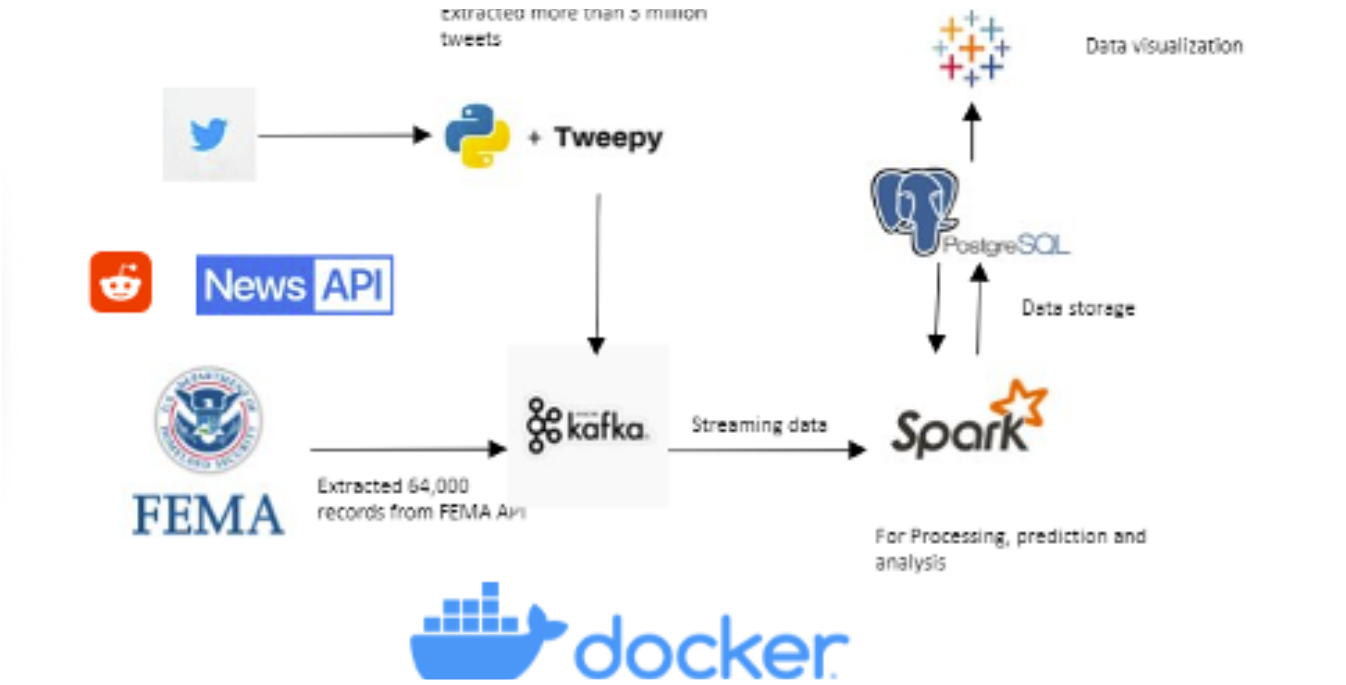
Abhisek Roy in Credit Saison (India)

Loading Large Datasets into Dynamo DB Tables Efficiently

One of the most easy-to-use databases in the AWS ecosystem is the DynamoDB. It's a fully managed solution, so you just need to create...

5 min read · Jun 27

50 1



Anitateladevalapalli

Big Data Analytics Project using Kafka, PySpark and tableau

This is the project to do the sentimental analysis on the natural disaster data obtained from various sources.

5 min read · Jun 7

55

See more recommendations