



# Setting up a Hadoop cluster on Windows using Docker and WSL2

🕒 2 minute read

I wanted to setup a Hadoop cluster as a playground on my Windows 10 laptop. I thought that using Docker with the new WSL2 (Windows Sub-system Linux version 2) included in Windows 10 version 0420 could be a solution. Indeed Docker can use WSL2 to run natively Linux on Windows. I basically followed the tutorial [How to set up a Hadoop cluster in Docker](https://clubhouse.io/developer-how-to/how-to-set-up-a-hadoop-cluster-in-docker/) (<https://clubhouse.io/developer-how-to/how-to-set-up-a-hadoop-cluster-in-docker/>), that is normally designed for a Linux host machine running docker (and not Windows).

## 1. Install Docker on Windows

I'm currently using docker desktop version 2.3.0.3 from the stable channel. But any version that supports WSL2 should work. The corresponding engine version is 19.03.8 and docker-compose version is 1.25.5:

```
cjlise@ /mnt/c/Users/jose$ docker --version
Docker version 19.03.8, build afacb8b7f0
cjlise@ /mnt/c/Users/jose$ docker-compose --version
docker-compose version 1.25.5, build 8a1c60f6
cjlise@ /mnt/c/Users/jose$ |
```

You can confirm that docker is running properly by launching a web server:

```
docker run -d -p 80:80 --name myserver nginx
```

## 2. Setting up Hadoop cluster using Docker

Use git to download the the Hadoop Docker files from the [Big Data Europe repository](https://github.com/big-data-europe/docker-hadoop).  
(<https://github.com/big-data-europe/docker-hadoop>):

```
git clone git@github.com:big-data-europe/docker-hadoop.git
```

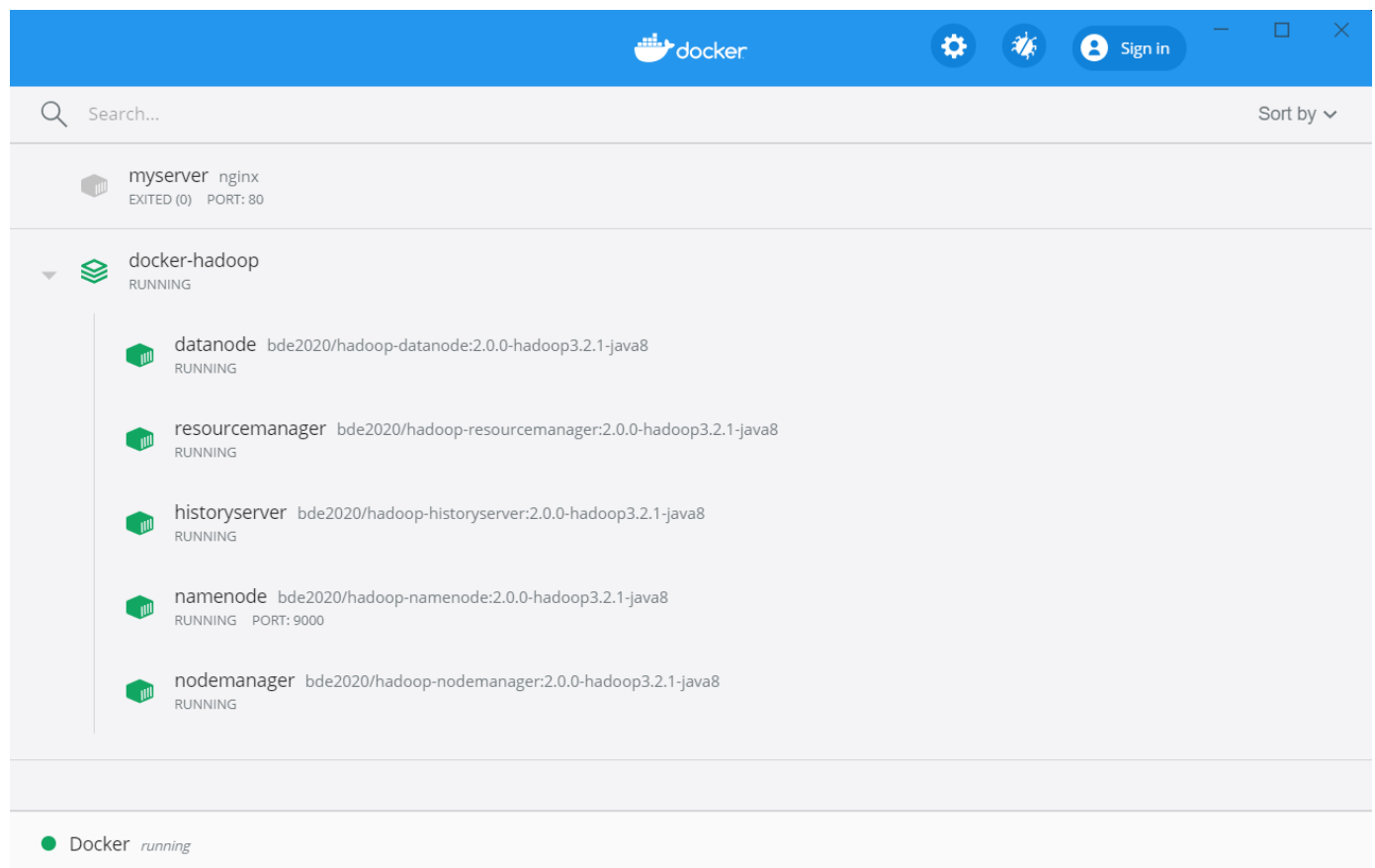
Deploy the docker cluster using the command:

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

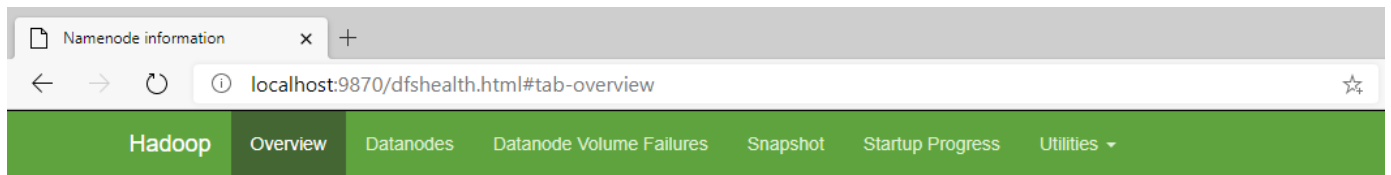
You can check that the containers are running using:

```
docker ps
```

You can also double check with the Docker dashboard:



And the current status can also be checked using the web page <http://localhost:9870>  
(<http://localhost:9870>):



## Overview 'namenode:9000' (active)

<b>Started:</b>	Sat Sep 26 11:44:48 +0200 2020
<b>Version:</b>	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
<b>Compiled:</b>	Tue Sep 10 17:56:00 +0200 2019 by rohithsharmaks from branch-3.2.1
<b>Cluster ID:</b>	CID-8a0289f6-7fe4-4478-b676-aa0938ae5323
<b>Block Pool ID:</b>	BP-1082917754-172.18.0.6-1600889809275

## Summary

Security is off.

Safemode is off.

51 files and directories, 24 blocks (24 replicated blocks, 0 erasure coded block groups) = 75 total filesystem object(s).

Heap Memory used 157.04 MB of 351 MB Heap Memory. Max Heap Memory is 2.75 GB.

Non Heap Memory used 49.27 MB of 50.5 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

<b>Configured Capacity:</b>	250.98 GB
<b>Configured Remote Capacity:</b>	0 B
<b>DFS Used:</b>	936.05 KB (0%)
<b>Non DFS Used:</b>	2.38 GB
<b>DFS Remaining:</b>	235.79 GB (93.95%)
<b>Block Pool Used:</b>	936.05 KB (0%)
<b>DataNodes usages% (Min/Median/Max/stdDev):</b>	0.00% / 0.00% / 0.00% / 0.00%
<b>Live Nodes</b>	1 (Decommissioned: 0, In Maintenance: 0)
<b>Dead Nodes</b>	0 (Decommissioned: 0, In Maintenance: 0)

## 3. Testing the Hadoop cluster

We will test the Hadoop cluster running the Word Count example.

- Open a terminal session on the namenode

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

This will open a session on the namenode for the root user.

- Create some simple text files to be used by the wordcount program

```
cd /tmp
mkdir input
echo "Hello World" >input/f1.txt
echo "Hello Docker" >input/f2.txt
```

- Create a hdfs directory named input

```
hadoop fs -mkdir -p input
```

- Put the input files in all the datanodes on HDFS

```
hdfs dfs -put ./input/* input
```

- Download on the host pc (e.g in the directory on top of the hadoop cluster directory) the word count program from [this link](https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-mapreduce-examples/2.7.1/hadoop-mapreduce-examples-2.7.1-sources.jar) (<https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-mapreduce-examples/2.7.1/hadoop-mapreduce-examples-2.7.1-sources.jar>).
- Run the command below in a terminal on the Windows host to identify the namenode container id:

```
docker container ls
```

```
:/mnt/c/Users/jose$ docker container ls
```

CONTAINER ID	IMAGE	PORTS	COMMAND NAMES	CREATED
afb235f8629c	bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8		"/entrypoint.sh /run..."	24 minutes ago
Up 24 minutes (healthy)		0.0.0.0:9000->9000/tcp, 0.0.0.0:9870->9870/tcp	namenode	
290c04575350	bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8		"/entrypoint.sh /run..."	24 minutes ago
Up 23 minutes (healthy)		8088/tcp	resourcemanager	
310ec0a5852c	bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8		"/entrypoint.sh /run..."	24 minutes ago
Up 24 minutes (healthy)		8042/tcp	nodemanager	
98510c6fd96f	bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8		"/entrypoint.sh /run..."	24 minutes ago
Up 24 minutes (healthy)		8188/tcp	historyserver	
393e3460a611	bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8		"/entrypoint.sh /run..."	24 minutes ago
Up 24 minutes (healthy)		9864/tcp	datanode	

- Use the command below on the Windows host to copy the word count program in the namenode container:

```
docker cp ../hadoop-mapreduce-examples-2.7.1-sources.jar afb235f8629c:/tmp
```

- Run the word count program in the namenode:

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

The program should display something like:

```
Deleted output  
root@afb235f8629c:/tmp#  
root@afb235f8629c:/tmp# hadoop jar hadoop-mapreduce-examples-2.7.1-sources.jar org.apache.hadoop.examples.WordCount input output  
2020-09-26 10:19:27,594 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/172.19.0.5:8032  
2020-09-26 10:19:27,722 INFO client.AHSPProxy: Connecting to Application History server at historyserver/172.19.0.4:10200  
2020-09-26 10:19:27,885 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1601113524798_0001  
2020-09-26 10:19:27,969 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false  
2020-09-26 10:19:28,087 INFO input.FileInputFormat: Total input files to process : 2  
2020-09-26 10:19:28,118 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false  
2020-09-26 10:19:28,561 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false  
2020-09-26 10:19:28,977 INFO mapreduce.JobSubmitter: number of splits:2  
2020-09-26 10:19:29,088 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false  
2020-09-26 10:19:29,098 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1601113524798_0001  
2020-09-26 10:19:29,098 INFO mapreduce.JobSubmitter: Executing with tokens: []  
2020-09-26 10:19:29,240 INFO conf.Configuration: resource-types.xml not found  
2020-09-26 10:19:29,240 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.  
2020-09-26 10:19:29,533 INFO impl.YarnClientImpl: Submitted application application_1601113524798_0001  
2020-09-26 10:19:29,563 INFO mapreduce.Job: The url to track the job: http://resourcemanager:8088/proxy/application_1601113524798_0001/  
2020-09-26 10:19:29,563 INFO mapreduce.Job: Running job: job_1601113524798_0001  
2020-09-26 10:19:35,661 INFO mapreduce.Job: Job job_1601113524798_0001 running in uber mode : false  
2020-09-26 10:19:35,665 INFO mapreduce.Job: map 0% reduce 0%  
2020-09-26 10:19:40,723 INFO mapreduce.Job: map 50% reduce 0%  
2020-09-26 10:19:41,729 INFO mapreduce.Job: map 100% reduce 0%  
2020-09-26 10:19:44,750 INFO mapreduce.Job: map 100% reduce 100%  
2020-09-26 10:19:44,775 INFO mapreduce.Job: Job job_1601113524798_0001 completed successfully
```

- Print the output of the word count program


```
hdfs dfs -cat output/part-r-000000
```

```
root@afb235f8629c:/tmp# hdfs dfs -cat output/part-r-000000  
2020-09-26 11:59:24,922 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false  
Docker 1  
Hello 2  
World 1  
root@afb235f8629c:/tmp#
```

- Shutdown the Hadoop cluster by running on the Windows host

```
docker-compose down
```

That's all !

 **Tags:** Big Data Distributed Processing Docker Hadoop Linux

 **Categories:** hadoop-spark

 **Updated:** September 26, 2020

