Samujjwaal Dey
Posted on Dec 27, 2020 • Updated on May 6, 2021

Hadoop Installation on Windows 10 using WSL

#hadoop #wsl #linux #bigdata

This article provides a step-by-step guide to install ***Hadoop 3.3.0*** on Windows 10 via WSL (Windows Subsystem for Linux). These instructions are also applicable for installing Hadoop on Linux systems.

Hadoop 3.3.0 (released on July 14, 2020) is the most recent release in the Apache Hadoop 3.3 line as of the time of this article.

Prerequisites

Follow the steps mentioned on the page below to first enable Windows Subsystem for Linux and then install a Linux distribution of your choice.

Windows Subsystem for Linux Installation Guide for Windows 10

▶ Initial Linux Configuration (optional)

Configure passphraseless ssh

Hadoop requires SSH access to manage its nodes.

This is a critical step and please make sure you follow the steps

Make sure you can SSH to localhost in Ubuntu:

```
ssh localhost
```

If you encounter the error `ssh: connect to host localhost port 22: Connection refused`, run the following commands:

```
sudo apt remove openssh-server
sudo apt install openssh-server
sudo service ssh start
```

If you cannot ssh to localhost without a passphrase, run the following command to initialize your private and public keys:

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```

Now `ssh localhost` should work without any passphrase.

Hadoop Installation

Install Java JDK

Update the package index:

```
sudo apt update
```

Check whether Java is already installed:

```
java -version
```

If Java isn't installed, install Java 8 from OpenJDK via the following command:

```
sudo apt-get install openjdk-8-jdk
```

Check the version installed:

```
java -version
openjdk version "1.8.0_275"
OpenJDK Runtime Environment (build 1.8.0_275-8u275-b01-0ubuntu1~20.04-b01)
OpenJDK 64-Bit Server VM (build 25.275-b01, mixed mode)
```

Download Hadoop binary

Visit the Hadoop releases page to find a download URL for Hadoop 3.3.0:

Hadoop Releases

Run the following command to download the latest binary for Hadoop 3.3.0:

```
wget https://downloads.apache.org/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
```

Wait until the download is completed.

Unzip Hadoop binary

Create a `hadoop` folder under the user home folder:

```
mkdir ~/hadoop
```

Unzip the downloaded binary package:

```
tar -xvzf hadoop-3.3.0.tar.gz -C ~/hadoop
```

After the package is unzipped, change the current directory to the Hadoop folder:

```
cd ~/hadoop/hadoop-3.3.0/
```

Setup environment variables

```
nano ~/.bashrc
```

Setup Hadoop and Java environment variables at the end of the `.bashrc` file as below and then save the bash file and close it.

```
#Set Hadoop-related environment variables
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=~/hadoop/hadoop-3.3.0
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
```

For applying all these changes to the current Terminal, execute the source command.

```
source ~/.bashrc
```

Make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal by,

```
java -version
openjdk version "1.8.0_275"
OpenJDK Runtime Environment (build 1.8.0_275-8u275-b01-0ubuntu1~20.04-b01)
OpenJDK 64-Bit Server VM (build 25.275-b01, mixed mode)


hadoop version
Hadoop 3.3.0
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r aa96f1871bfd858
Compiled by brahma on 2020-07-06T18:44Z
Compiled with protoc 3.7.1
From source with checksum 5dc29b802d6ccd77b262ef9d04d19c4
This command was run using /home/daftdey/hadoop/hadoop-3.3.0/share/hadoop/common/hadoop-
```

Configure the pseudo-distributed mode (Single Node mode)

This page below is the official documentation to configure a single node cluster:

Pseudo-Distributed Operation

All the Hadoop configuration files should be located in the `~/hadoop/hadoop-3.3.0/etc/hadoop` directory.

```
cd ~/hadoop/hadoop-3.3.0/etc/hadoop
```

The configuration setup steps are as follows:

1. Edit file `hadoop-env.sh`:

   ```
   nano hadoop-env.sh
   ```

   Set Java environment variable as,

   ```
   # Java path
   export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
   ```

2. Edit file `core-site.xml`:
   `core-site.xml` informs the Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

   ```
   nano core-site.xml
   ```

   Add the following configuration:

   ```
   <configuration>
       <property>
           <name>fs.defaultFS</name>
           <value>hdfs://localhost:9000</value>
       </property>
   </configuration>
   ```

3. Edit file `hdfs-site.xml`:
   `hdfs-site.xml` contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

   ```
   nano hdfs-site.xml
   ```

   Add the following configuration:

   ```
   <configuration>
       <property>
   ```

```
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

4. Edit file `mapred-site.xml` :

   `mapred-site.xml` contains configuration settings of MapReduce application like the number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

   ```
   nano mapred-site.xml
   ```

   Add the following configuration:

   ```
   <configuration>
       <property>
           <name>mapreduce.framework.name</name>
           <value>yarn</value>
       </property>
       <property>
           <name>mapreduce.application.classpath</name>
           <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/h
       </property>
   </configuration>
   ```

5. Edit file `yarn-site.xml` :

   `yarn-site.xml` contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

   ```
   nano yarn-site.xml
   ```

   Add the following configuration:

   ```
   <configuration>
       <property>
           <name>yarn.nodemanager.aux-services</name>
           <value>mapreduce_shuffle</value>
       </property>
       <property>
           <name>yarn.nodemanager.env-whitelist</name>
           <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_
   ```

```
    </property>
</configuration>
```

Format namenode

Go to the Hadoop home directory and format the Hadoop namenode:

```
cd ~/hadoop/hadoop-3.3.0
bin/hdfs namenode -format
```

This formats the HDFS via the NameNode. Formatting the file system means initializing the directory specified by the `dfs.name.dir` variable.

This command should be executed only for the first time. Never format an up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

Run HDFS daemons

The *NameNode* is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the files stored across the cluster.

On startup, a *DataNode* connects to the *Namenode* and it responds to the requests from the *Namenode* for different operations.

Start NameNode daemon and DataNode daemon

```
sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [tars]
```

Check status via `jps` command

```
jps
8544 SecondaryNameNode
8304 DataNode
8149 NameNode
8702 Jps
```

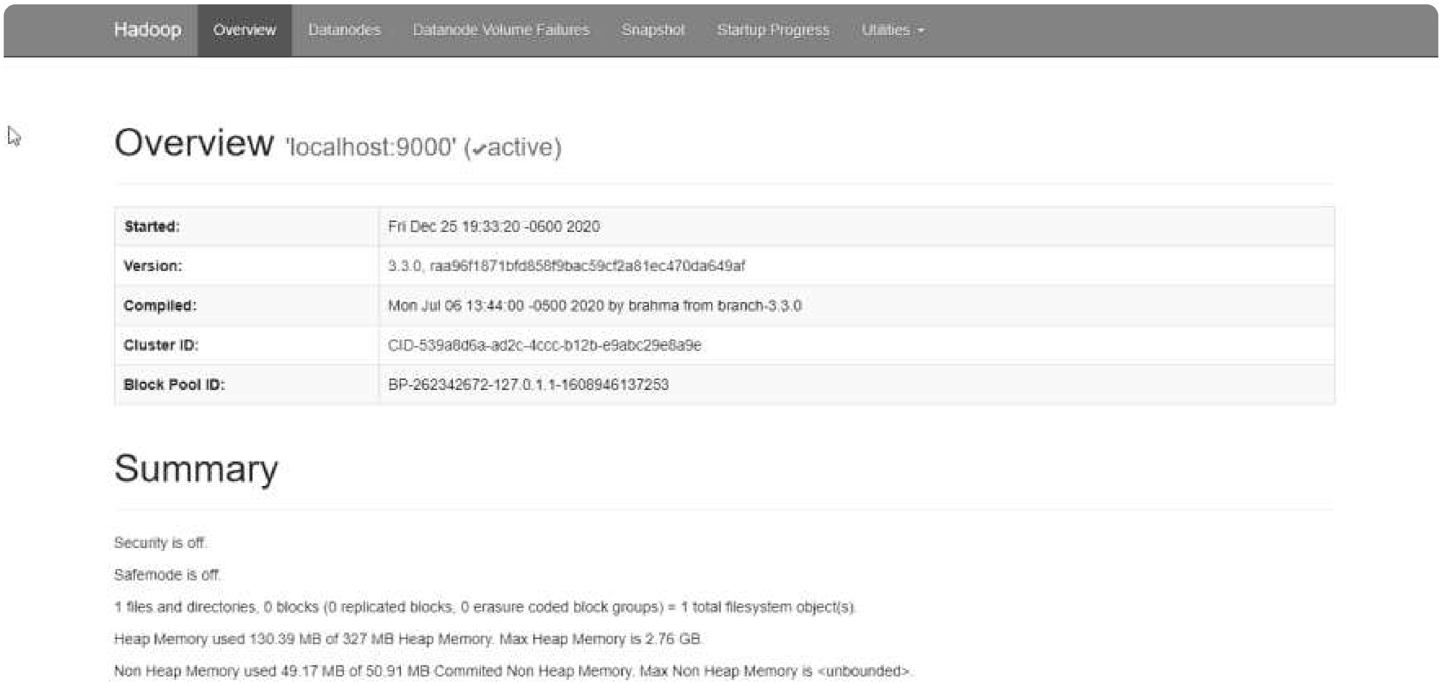
When the HDFS services are successfully initiated, you should be able to see these four processes running.

View name node portal

Now open the browser and go to the following URL to view the NameNode interface. Scroll down and see the *number of live nodes*.

http://localhost:9870/dfshealth.html#tab-overview

The web UI looks like this:



Overview 'localhost:9000' (✓active)

Started:	Fri Dec 25 19:33:20 -0600 2020
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Mon Jul 06 13:44:00 -0500 2020 by brahma from branch-3.3.0
Cluster ID:	CID-539a8d6a-ad2c-4ccc-b12b-e9abc29e8a9e
Block Pool ID:	BP-262342672-127.0.1.1-1608946137253

Summary

Security is off.

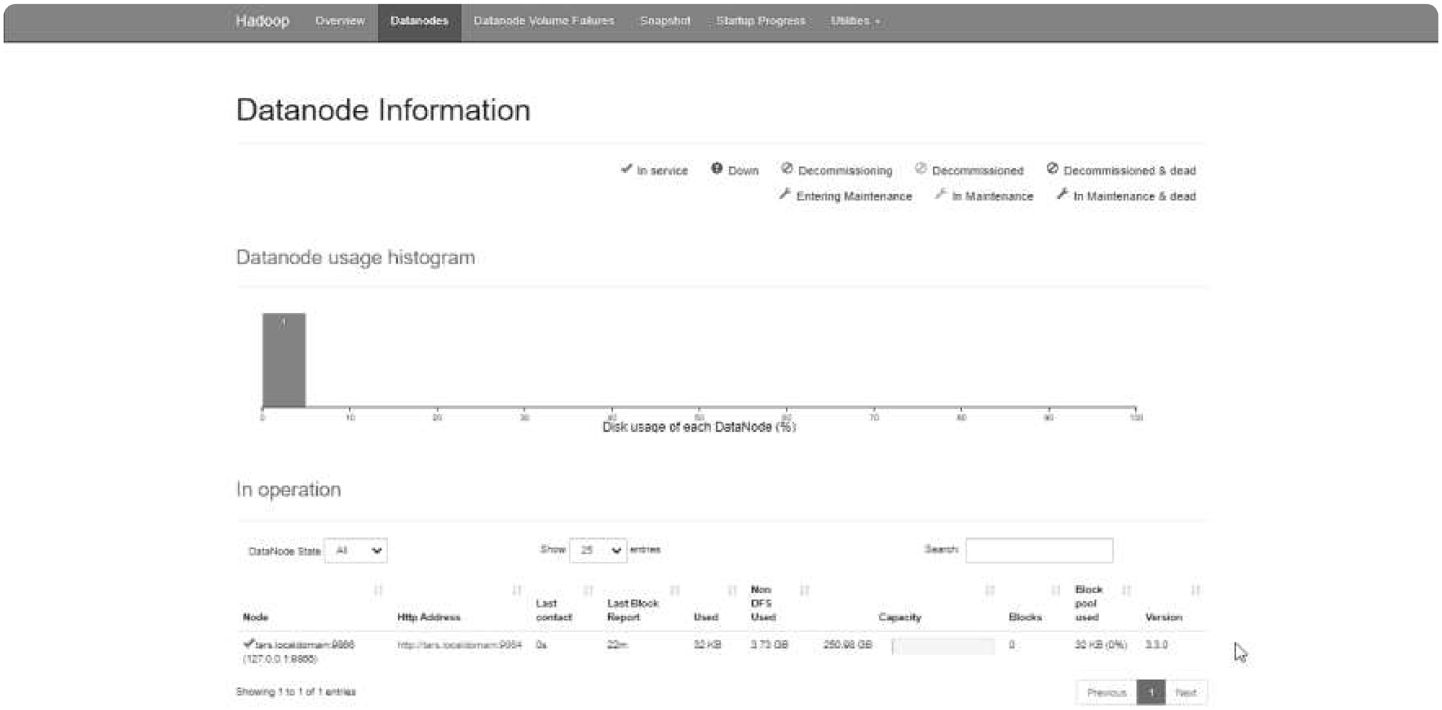
Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 130.39 MB of 327 MB Heap Memory. Max Heap Memory is 2.76 GB.

Non Heap Memory used 49.17 MB of 50.91 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Data nodes information can be also viewed through the *Datanodes* menu link:



Datanode Information

Datanode usage histogram

In operation

DataNode State: All

Show 25 entries

Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓tars.localdomain:9866 (127.0.0.1:9866)	http://tars.localdomain:9864	0s	22m	32 KB	3.73 GB	250.98 GB	0	32 KB (0%)	3.3.0

Showing 1 to 1 of 1 entries

Run YARN daemon

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each *NodeManagers* and each application's ApplicationMaster.

The *NodeManager* in each machine framework is the agent that is responsible for managing containers, monitoring their resource usage, and reporting the same to the ResourceManager.

Start ResourceManager daemon and NodeManager daemon

```
sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

Check status using the `jps` command

```
jps
11831 ResourceManager
11352 DataNode
12347 Jps
11196 NameNode
```

```
11981 NodeManager
11597 SecondaryNameNode
```

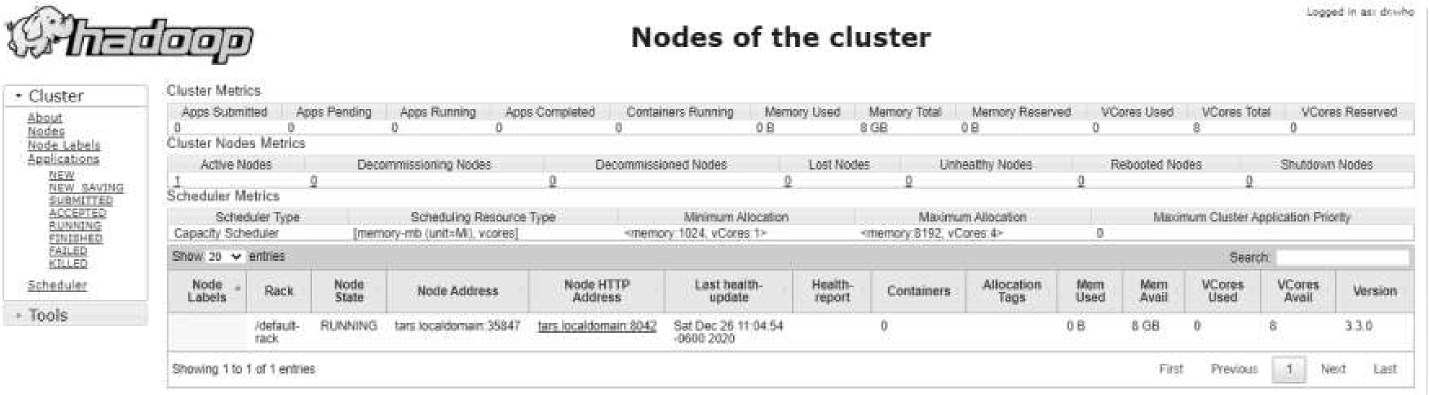
Once the YARN services are started, two more processes for NodeManager and ResourceManage can be seen running.

View YARN web portal

Now open the browser and go to the following URL to view the YARN resource manager web UI.

http://localhost:8088/cluster

The web UI looks like this:



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	tars.localdomain:35847	tars.localdomain:8042	Sat Dec 26 11:04:54 -0600 2020		0		0 B	8 GB	0	8	3.3.0

Showing 1 to 1 of 1 entries

Execution of all the Hadoop applications can be monitored through this web portal.

Shutdown Hadoop services

After completion of the execution of all Hadoop applications, the HDFS and YARN services can be stopped as follows:

```
sbin/stop-yarn.sh
sbin/stop-dfs.sh
```

Verify status via `jps` command, only the *jps* service should be running:

```
jps
13257 Jps
```

Summary

Congratulations, you have successfully installed a single-node Hadoop cluster in one go on your Linux subsystem of Windows 10. It's relatively easier compared to installing on Windows as we don't need to download or build native Hadoop HDFS libraries.

Have fun exploring with Hadoop.

Try out an example application: Hadoop MapReduce computational model to perform analyses on DBLP publication data

If you encounter any issues while following the steps, please post a comment and I will try my best to help.

Further References

Install Hadoop 3.3.0 on Windows 10 using WSL

Install Hadoop: Setting up a Single Node Hadoop Cluster

Discussion (0)

Code of Conduct • Report abuse

Samujjwaal Dey

here to learn

LOCATION
Chicago, Illinois

JOINED
Feb 21, 2020

Trending on DEV Community 👩‍💻 👨‍💻 🔥

3 effective methods of learning new programming technologies (and tips on how to use them)
#learning #beginners #productivity #discuss

Should I share my current salary with recruiters?

#anonymous #discuss #career #help

I Made website for me. Check it

#webdev #javascript #discuss #website