**Setup Hadoop on Windows 10 Machines**

<https://gist.github.com/vorpal56/5e2b67b6be3a827b85ac82a63a5b3b2e>

## Required tools

1. Java JDK - used to run the Hadoop since it's built using Java
2. 7Zip or WinRAR - unzip Hadoop binary package; anything that unzips tar.gz
3. CMD or Powershell - used to test environment variables and run Hadoop

## Step 1 - Download and extract Hadoop

Download Hadoop from their official website and unzip the file. We'll be using Hadoop 3.2.1.

<https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz>

Hadoop is portable so you can store it on an external hard drive. For the purpose of documentation, You need to extract it to somewhere on the hard drive like C:/Users/Anthony/Documents/cp-master.

If there are permission errors, run your unzipping program as administrator and unzip again.

## Step 2 - Install Hadoop native IO binary

Clone or download the winutils repository (<https://github.com/cdarlint/winutils>) and copy the contents of hadoop-3.2.1/bin into the extracted location of the Hadoop binary package. In our example, it will be C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1\bin

## Step 3 - Install Java JDK

Java JDK is required to run Hadoop, so if you haven't installed it, install it.

Oracle requires you sign up and login to download it. Google 'jdk version download.

Run the installation file and the default installation directory will be C:\Program Files\Java\jdk1.8.0\_261.

After installation, open up CMD or Powershell and confirm Java is intalled:

$ java -version

java version "1.8.0\_261"

Java(TM) SE Runtime Environment (build 1.8.0\_261-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

## Step 4 - Configure environment variables

Open the Start Menu and type in 'environment' and press enter. A new window with System Properties should open up. Click the Environment Variables button near the bottom right.

### JAVA\_HOME environment variable

1. From step 3, find the location of where you installed Java. In this example, the default directory is C:\Program Files\Java\jdk1.8.0\_261
2. Create a new **User variable** with the variable name as JAVA\_HOME and the value as C:\Program Files\Java\jdk1.8.0\_261

### HADOOP\_HOME environment variable

1. From step 1, copy the directory you extracted the Hadoop binaries to. In this example, the directory is C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1
2. Create a new **User variable** with the variable name as HADOOP\_HOME and the value as C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1

### PATH environment variable

We'll now need to add the bin folders to the PATH environment variable.

1. Click Path then Edit
2. Click New on the top right
3. Add C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1\bin
4. Add C:\Program Files\Java\jdk1.8.0\_261\bin

### Hadoop environment

Hadoop complains about the directory if the JAVA\_HOME directory has spaces. In the default installation directory, Program Files has a space which is problematic. To fix this, open the %HADOOP\_HOME%\etc\hadoop\hadoop-env.cmd and change the JAVA\_HOME line to the following:

set JAVA\_HOME=C:\PROGRA~1\Java\jdk1.8.0\_261

After setting those environment variables, you reopen CMD or Powershell and verify that the hadoop command is available:

$ hadoop -version

java version "1.8.0\_261"

Java(TM) SE Runtime Environment (build 1.8.0\_261-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

## Step 5 - Configure Hadoop

Now we are ready to configure the **most important part** - Hadoop configurations which involves Core, YARN, MapReduce, and HDFS configurations.

Each of the files are in %HADOOP\_HOME%\etc\hadoop. The full path for this example is C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1\etc\hadoop

### **Configure core site**

Edit core-site.xml and replace the configuration element with the following:

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

### **Configure HDFS**

Create two folders, one for the namenode directory and another for the data directory. The following are the two created folders in this example:

1. C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1\data\dfs\namende
2. C:\Users\Anthony\Documents\cp-master\hadoop-3.2.1\data\dfs\datanode

**Edit hdfs-site.xml and replace the configuration element with the following:**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///c:/hadoop-3.2.1/data/dfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///c:/hadoop-3.2.1/data/dfs/datanode</value>

</property>

</configuration>

### **Configure MapReduce and YARN site**

Edit mapred-site.xml and replace the configuration element with the following:

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

**Edit yarn-site.xml and replace the configuration element with the following:**

## <configuration>

## <!-- Site specific YARN configuration properties -->

## <property>

## <name>yarn.nodemanager.aux-services</name>

## <value>mapreduce\_shuffle</value>

## </property>

## <property>

## <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

## <value>org.apache.hadoop.mapred.ShuffleHandler</value>

## </property>

## <property>

## <name>yarn.nodemanager.local-dirs</name>

## <value>file:///C:/tmp</value>

## </property>

## </configuration>

## **Step 6 - Initialize HDFS and bugfix**

Run the following command and you should find the following error:

$ hdfs namenode -format

...

ERROR namenode.NameNode: Failed to start namenode.

...

To fix this, you'll need to download a JAR file with the fix (link is given below).

<https://github.com/FahaoTang/big-data/blob/master/hadoop-hdfs-3.2.1.jar>

Overwrite the existing hadoop-hdfs-3.2.1.jar in %HADOOP\_HOME%\share\hadoop\hdfs with this new JAR file (you can make a backup of the current one before overwriting if you wish).

## **Step 7 - Start HDFS daemons**

Run the following command to start HDFS daemons. When you do so, there should be two new windows that open: one for datanode and the other for namenode:

$ %HADOOP\_HOME%\sbin\start-dfs.cmd

## **Step 8 - Start YARN daemons**

You might encounter permission issues as a normal user, so open a command line with elevated permissions. **If you have the**[**yarn**](https://yarnpkg.com/)**package manager, you will NOT be able to run YARN daemons since both use the yarn command.** To fix this, you must uninstall yarn package manager.

Run the following command (with elevated permissions) to start YARN daemons. When you do so, there should be two new windows that open: one for resource manager and the other for node manager:

$ %HADOOP\_HOME%\sbin\start-yarn.cmd

## **Step 9 - Useful Web portals**

The daemons also host websites that provide useful information about the cluster

### HDFS Namenode UI info

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

### HDFS Datanode UI info

<http://localhost:9864/datanode.html>

### YARN resource manager UI

[http://localhost:8088](http://localhost:8088/)

## **Step 10 - Shutdown YARN and HDFS daemons**

You can stop the daemons by running the following commands:

$ %HADOOP\_HOME%\sbin\stop-dfs.cmd

$ %HADOOP\_HOME%\sbin\stop-yarn.cmd