# Install Hadoop: Setting up a Single Node Hadoop Cluster

### **Step 1:**

### **Download the Java 8 Package. Save this file in your home directory.**

apt-get update

apt-get install default-jdk

java -version

### **Step 3:**

### **Download the Hadoop 2.7.3 Package.**

**Command:** **wget** [**https://archive.apache.org/dist/hadoop/hadoop-2.8.0.tar.gz**](https://archive.apache.org/dist/hadoop/hadoop-2.8.0.tar.gz)

**Note- if Link is not working go to** [**https://archive.apache.org**](https://archive.apache.org) **and select Hadoop and right click on hadoop-x.x.x.tar.gz file and copy its URL**

### **Step 4:**

### **Extract the Hadoop tar File.**

**Command**: tar -xvf hadoop-2.8.0.tar.gz (check your version and tar with it)

### **Step 5: Add the Hadoop and Java paths in the bash file (.bashrc).**

Open **.** **bashrc** file. Now, add Hadoop and Java Path as shown below.

**Command:**  vi .bashrc (check your path when you’re doing this if it is empty then press cd and check)

At the end of file past below lines

Java Path- **echo $JAVA\_HOME**

**export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64**

**echo $JAVA\_HOME**

**export PATH=$PATH: $JAVA\_HOME/bin**

**Note-**

1. **1It will not work for all, check your version and path of java and give it.**
2. **google it and try**

**3.if you not get anything just give enter below commends**

**export JAVA\_HOME=/usr**

**export PATH=$PATH: $JAVA\_HOME/bin**

**and check java -version is it showing version**

### **Step 6:**

### **Edit the**[**Hadoop Configuration files**](https://www.edureka.co/blog/explaining-hadoop-configuration/)**.**

***Command:*** cd your hadoop downloaded path / hadoop-2.8.0/etc/hadoop/ (check where you downloaded )

***Command:***  ls

### **Step 7:**

### **Open core-site.xml and edit the property mentioned below inside configuration tag:**

core-site.xml informs Hadoop daemon where Name Node runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

**Command:** vi core-site.xm



Paste below commends between <configuration> </configuration>

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>fs.default.name</name>

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

</property>

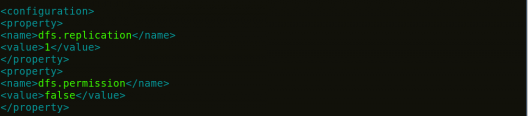
</configuration>

### **Step 8:**

### **Edit hdfs-site.xml and edit the property mentioned below inside configuration tag:**

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.

**Command:** vi hdfs-site.xml



Paste below commends between <configuration> </configuration>

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.permission</name>

<value>false</value>

</property>

</configuration>

### **Step 9:**

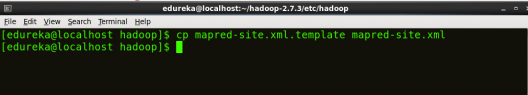
### **Edit the mapred-site.xml file and edit the property mentioned below inside configuration tag:**

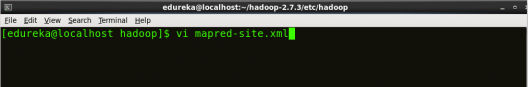
mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

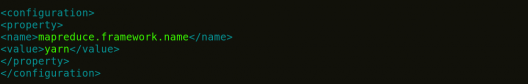
In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

**Command:** cp mapred-site.xml.template mapred-site.xml

**Command:**vi mapred-site.xml.







Paste below commends between <configuration> </configuration>

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

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

<value>yarn</value>

</property>

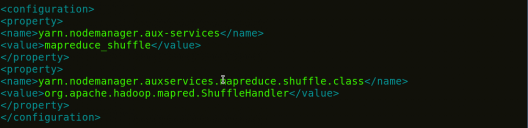
</configuration>

### **Step 10:**

### **Edit yarn-site.xml and edit the property mentioned below inside configuration tag:**

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

**Command:** vi yarn-site.xml



**<?xml version="1.0">**

**<configuration>**

**<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>**

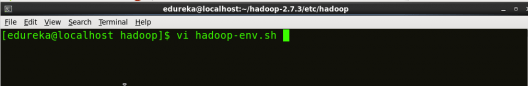
**</configuration>**

### **Step 11:**

### **Edit hadoop-env.sh and add the Java Path as mentioned below:**

hadoop-env.sh contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

**Command:** vi hadoop–env.sh





**At the end of file give java path which your given in step-5 check image to know which type of path**

### **From here I have some doubt please follow as per below steps any doubts follow below link**

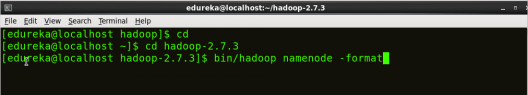
### <https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster>

### **Step 12: Go to Hadoop home directory and format the NameNode.**

**Command:** cd

**Command:** cd hadoop-2.7.3

**Command:** bin/hadoop namenode -format



This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

### **Step 13:**

### **Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.**

**Command:**cd hadoop-2.8.0/sbin

Either you can start all daemons with a single command or do it individually.

***Command:*** ./start-all.sh

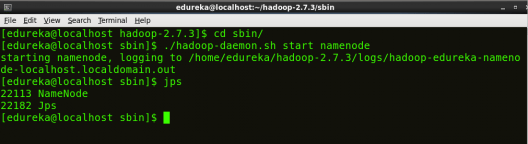
The above command is a combination of**start-dfs.sh, start-yarn.sh** & **mr-jobhistory-daemon.sh**

Or you can run all the services individually as below:

### **Start NameNode:**

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 file stored across the cluster.

**Command:**./hadoop-daemon.sh start namenode



### **Start DataNode:**

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

**Command:**./hadoop-daemon.sh start datanode

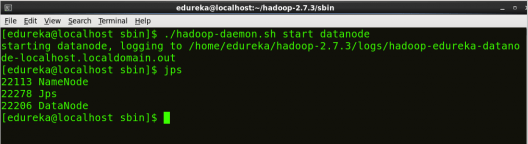


Fig: Hadoop Installation – Starting DataNode

### **Start ResourceManager:**

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 the each application’s ApplicationMaster.

**Command:**./yarn-daemon.sh start resourcemanager

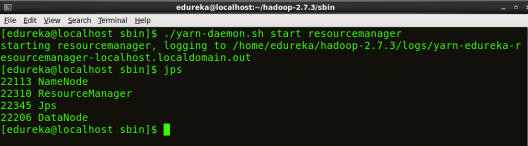


Fig: Hadoop Installation – Starting ResourceManager

### **Start NodeManager:**

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

**Command:**./yarn-daemon.sh start nodemanager

### **Start JobHistoryServer:**

JobHistoryServer is responsible for servicing all job history related requests from client.

**Command:** ./mr-jobhistory-daemon.sh start historyserver

### **Step 14:**

### **To check that all the Hadoop services are up and running, run the below command.**

**Command:**jps

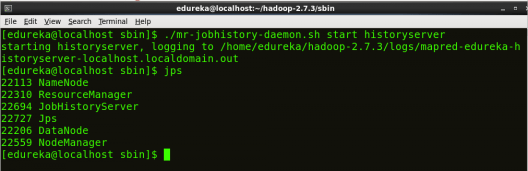


Fig: Hadoop Installation – Checking Daemons

### **Step 15: Now open the Mozilla browser and go to localhost:50070/dfshealth.html to check the NameNode interface.**

