



# Sinhgad Institutes

Name of the Student: \_\_\_\_\_ Roll no: \_\_\_\_\_

CLASS:- T.E.[IT]

Division: A

Course: - 2019

Subject: 314457: Data Science and Big Data Analytics Laboratory

PART\_ A \_Assignment No. 01

Marks: \_\_\_\_/10

Date of Performance: \_\_\_\_/\_\_\_\_/\_\_\_\_

Sign with Date: \_\_\_\_\_

**Part- A**  
**ASSIGNMENT NO: 01**

**TITLE:**

Hadoop Installation.

**AIM:**

To Hadoop Installation on a) Single Node b) Multiple Node.

**OBJECTIVE:**

- 1) To understand Big data primitives and fundamentals.
- 2) To understand the different Big data processing techniques.

**SOFTWARE USED:** Hadoop 3.2.2, jdk11.

**THEORY:****Cluster Computing:**

A collection of computers configured in such a way that they can be used to solve a problem by means of parallel processing.

**Homogeneous Cluster**

The cluster in which every single node is exactly same, from the motherboard and the memory, to the disk drives and the NIC.

**Heterogeneous Cluster**

Made from different kinds of computers, SPARC, DEC ALPHA

- Software Library: Hadoop is a software library for distributed computing.
- Distributed Storage: Hadoop platform provides distributed storage.
- Computational Capabilities: Hadoop platform provides distributed computational capabilities.

**Benefits of using Hadoop**

The architecture of Hadoop allows you to scale your hardware as and when you need to. New nodes can be added incrementally without having to worry about the change in data formats or the handling of applications that sit on the file system.

One of the most important features of Hadoop is that it allows you to save enormous amounts of money by substituting cheap commodity servers for expensive ones. This is possible because Hadoop transfers the responsibility of fault tolerance from the hardware layer to the application layer.

## Installing Hadoop

Installing and getting Hadoop up and running is quite straightforward. However, since this process requires editing multiple configuration and setup files, make sure that each step is properly followed.

### 1. Install Java

Hadoop requires Java to be installed, so let's begin by installing Java: `apt-get update`  
`apt-get install default-jdk`

These commands will update the package information on your VPS and then install Java. After executing these commands, execute the following command to verify that Java has been installed: `java -version`

If Java has been installed, this should display the version details as illustrated in the following image:

```
root@tutorials:~# java -version
java version "1.7.0_51"
OpenJDK Runtime Environment (IcedTea 2.4.4) (7u51-2.4.4-0ubuntu0.13.10.1)
OpenJDK 64-Bit Server VM (build 24.45-b08, mixed mode)
root@tutorials:~#
```

### 2. Create and Setup SSH Certificates

Hadoop uses SSH (to access its nodes) which would normally require the user to enter a password. However, this requirement can be eliminated by creating and setting up SSH certificates using the following commands:

```
ssh-keygen -t rsa -P ''
```

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

After executing the first of these two commands, you might be asked for a filename. Just leave it blank and press the enter key to continue. The second command adds the newly created key to the list of authorized keys so that Hadoop can use SSH without prompting for a password.

```

root@tutorials:~# ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
23:42:08:de:26:bc:6e:4c:3e:3e:55:f7:a6:4d:8f:bb root@tutorials
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|o...
|+.o.
|+. . .
|o o o S
|= . . . =
|* . . . = o
|o.. . o .
|.. . Eo
+-----+
root@tutorials:~# cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
root@tutorials:~# █

```

### Fetch and Install Hadoop

First let's fetch Hadoop from one of the mirrors using the following command:

**wget <http://www.motorlogy.com/apache/hadoop/common/current/hadoop-3.2.2.tar.gz>**

Note: This command uses a download a link on one of the mirrors listed on the Hadoop website. The list of mirrors can be found on this link. You can choose any other mirror if you want to. To download the latest stable version, choose the *\*hadoop-X.Y.Z.tar.gz\** file from the current or the current2 directory on your chosen mirror.\*

After downloading the Hadoop package, execute the following command to extract it:

**tar xzf hadoop-2.6.0.tar.gz**

This command will extract all the files in this package in a directory named hadoop-2.3.0. For this tutorial, the Hadoop installation will be moved to the /usr/local/hadoop directory using the following command: **mv hadoop-2.6.0 /usr/local/hadoop**

Note: The name of the extracted folder depends on the Hadoop version you have downloaded and extracted. If your version differs from the one used in this tutorial, change the above command accordingly.

### 3. Edit and Setup Configuration Files

To complete the setup of Hadoop, the following files will have to be modified:

- ~/.bashrc
- /usr/local/hadoop/etc/hadoop/hadoop-env.sh
- /usr/local/hadoop/etc/hadoop/core-site.xml
- /usr/local/hadoop/etc/hadoop/yarn-site.xml
- /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
- /usr/local/hadoop/etc/hadoop/hdfs-site.xml

### i. Editing ~/.bashrc

Before editing the .bashrc file in your home directory, we need to find the path where Java has been installed to set the JAVA\_HOME environment variable. Let's use the following command to do that:

update-alternatives --config java

This will display something like the following:

```
root@tutorials:~# update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java
Nothing to configure.
root@tutorials:~#
```

The complete path displayed by this command is:

/usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java

The value for JAVA\_HOME is everything before /jre/bin/java in the above path - in this case, /usr/lib/jvm/java-7-openjdk-amd64.

Make a note of this as we'll be using this value in this step and in one other step. Now use nano (or your favoured editor) to edit ~/.bashrc using the following command:  
nano ~/.bashrc

This will open the .bashrc file in a text editor. Go to the end of the file and paste/type the following content in it:

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

Note 1: If the value of JAVA\_HOME is different on your VPS, make sure to alter the first export statement in the above content accordingly.

Note 2: Files opened and edited using nano can be saved using Ctrl + X. Upon the prompt to save changes, type Y. If you are asked for a filename, just press the enter key.

The end of the .bashrc file should look something like this:

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi

#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

After saving and closing the .bashrc file, execute the following command so that your system recognizes the newly created environment variables: `source ~/.bashrc`

Putting the above content in the .bashrc file ensures that these variables are always available when your VPS starts up.

## ii. Editing /usr/local/hadoop/etc/hadoop/hadoop-env.sh

Open the /usr/local/hadoop/etc/hadoop/hadoop-env.sh file with nano using the following command:

```
nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

In this file, locate the line that exports the JAVA\_HOME variable. Change this line to the following:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

Note: If the value of JAVA\_HOME is different on your VPS, make sure to alter this line accordingly.



The hadoop-env.sh file should look something like this:

```
# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
# export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

Save and close this file. Adding the above statement in the hadoop-env.sh file ensures that the value of JAVA\_HOME variable will be available to Hadoop whenever it is started up.

### iii. Editing /usr/local/hadoop/etc/hadoop/core-site.xml

The /usr/local/hadoop/etc/hadoop/core-site.xml file contains configuration properties that Hadoop uses when starting up. This file can be used to override the default settings that Hadoop starts with.

Open this file with nano using the following command:

```
nano /usr/local/hadoop/etc/hadoop/core-site.xml
```

In this file, enter the following content in between the <configuration></configuration> tag:

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

The core-site.xml file should look something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

Save and close this file.

#### iv. Editing /usr/local/hadoop/etc/hadoop/yarn-site.xml

The /usr/local/hadoop/etc/hadoop/yarn-site.xml file contains configuration properties that MapReduce uses when starting up. This file can be used to override the default settings that MapReduce starts with.

Open this file with nano using the following command: nano /usr/local/hadoop/etc/hadoop/yarnsite.xml

In this file, enter the following content in between the <configuration></configuration> tag:

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

The yarn-site.xml file should look something like this:

```
<?xml version="1.0"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<configuration>

    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>

</configuration>
```

Save and close this file.



**v. Creating and Editing /usr/local/hadoop/etc/hadoop/mapred-site.xml**

By default, the /usr/local/hadoop/etc/hadoop/ folder contains the /usr/local/hadoop/etc/hadoop/mapred-site.xml.template file which has to be renamed/copied with the name mapred-site.xml. This file is used to specify which framework is being used for MapReduce.

This can be done using the following command: `cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml`

Once this is done, open the newly created file with nano using the following command: `nano /usr/local/hadoop/etc/hadoop/mapred-site.xml`

In this file, enter the following content in between the <configuration></configuration> tag:

```
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
```

The mapred-site.xml file should look something like this:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Save and close this file.

**vi. Editing /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

The /usr/local/hadoop/etc/hadoop/hdfs-site.xml has to be configured for each host in the cluster that is being used. It is used to specify the directories which will be used as the namenode and the datanode on that host.

Before editing this file, we need to create two directories which will contain the

namenode and the datanode for this Hadoop installation. This can be done using the following commands: `mkdir -p /usr/local/hadoop_store/hdfs/namenode` `mkdir -p /usr/local/hadoop_store/hdfs/datanode` Note: You can create these directories in different locations, but make sure to modify the contents of `hdfs-site.xml` accordingly.

Once this is done, open the `/usr/local/hadoop/etc/hadoop/hdfs-site.xml` file with nano using the following command:

```
nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

In this file, enter the following content in between the `<configuration>``</configuration>` tag:

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/datanode</value>
</property>
```

The `hdfs-site.xml` file should look something like this :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
  </property>
</configuration>
```

Save and close this file.

## Format the New Hadoop Filesystem

After completing all the configuration outlined in the above steps, the Hadoop filesystem needs to be formatted so that it can start being used. This is done by executing the following command: `hdfs namenode -format`

Note: This only needs to be done once before you start using Hadoop. If this command is executed again after Hadoop has been used, it'll destroy all the data on the Hadoop file system.

## Start Hadoop

All that remains to be done is starting the newly installed single node cluster:

`Start-all.sh`

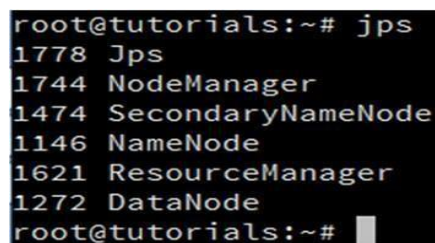
While executing this command, you'll be prompted twice with a message similar to the following: Are you sure you want to continue connecting (yes/no)?

Type in yes for both these prompts and press the enter key. Once this is done, execute the following command: `start-yarn.sh`

Executing the above two commands will get Hadoop up and running. You can verify this by typing in the following command:

`jps`

Executing this command should show you something similar to the following:



```
root@tutorials:~# jps
1778 Jps
1744 NodeManager
1474 SecondaryNameNode
1146 NameNode
1621 ResourceManager
1272 DataNode
root@tutorials:~#
```

If you can see a result similar to the depicted in the screenshot above, it means that you now have a functional instance of Hadoop running on your VPS.

## CONCLUSION:

After the study of this assignment we are familiar with the installation of Hadoop.

**Write Short Answers for Following Questions:**

1. What is Big Data?
2. Why Hadoop was introduced and who introduced it? Explain the use of Hadoop in Big Data.
3. Draw an Ecosystem of Hadoop.
4. What is SSH? How to setup SSH certificate?
5. Explain the role of each daemons of Hadoop.

**Viva Questions:**

1. How user can interface with Hadoop Framework through browser?
2. Which configuration file we need to edit while installing Hadoop? List it.
3. What is multimode cluster?