**EX.NO.7 FIND PROCEDURE TO SET UP THE ONE NODE HADOOP CLUSTER**

**DATE:**

**OBJECTIVE:**

To set up the one node Hadoop cluster.

**PROCEDURE:**

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware. Hadoop needs Java 6 or above to work.

Hadoop is a Java-based framework which manages the large data sets among the group of cluster machines.

**1) Installing Java**

**Step 1:** Update the package index

**$ sudo apt-get update**

**Step 2:** Install java on your Ubuntu

**$ sudo apt-get install java**

**Step 3:** Check the version.

**$ java -version**

**Output:**

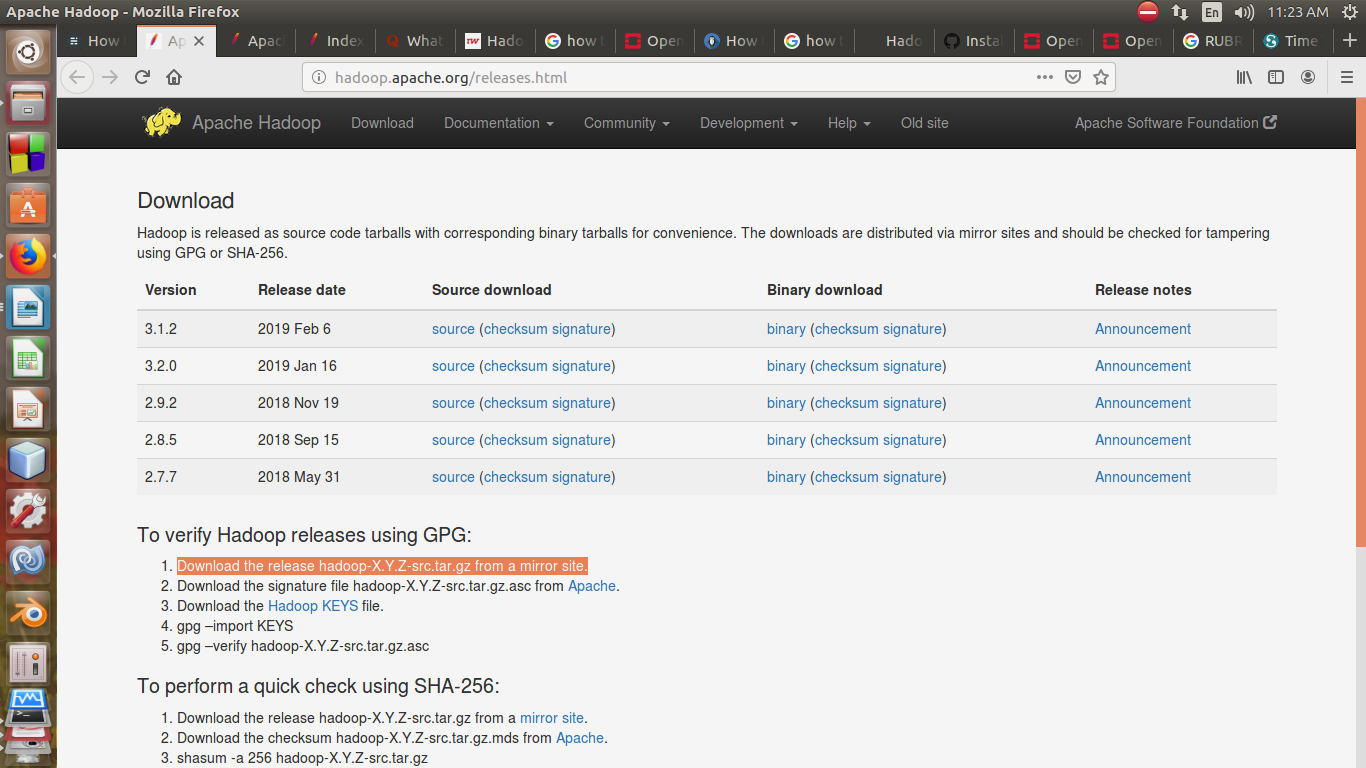
**java version "1.8.0\_171"**

**Java(TM) SE Runtime Environment (build 1.8.0\_171-b11)**

**Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)**

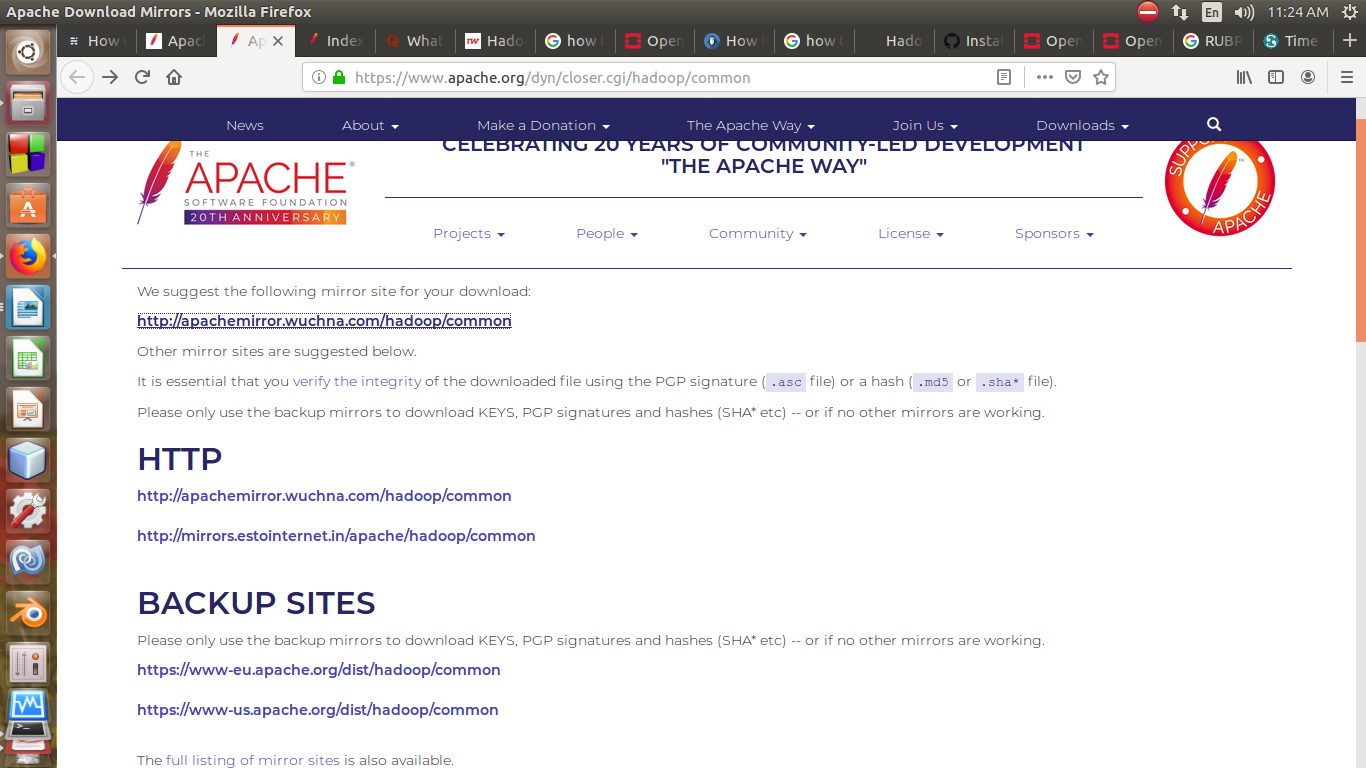
**2)****Installing Hadoop**

**Step 1:** To install the Hadoop. Go to the Apache Hadoop release page to find the [latest version of Apache](http://hadoop.apache.org/releases.html) in the given site. (**http://hadoop.apache.org/releases.html**)



You have to find the latest stable version to install hadoop 2.7 on ubuntu. Once you find the latest stable version and then copy the link by doing the right click.

Now install hadoop 2.7.7 on ubuntu.



**Step 2:** Use the below command to download the file.

**$ wget http://apache.mirrors.tds.net/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz**

**Step 3:** Check whether the file has been altered while downloading.

**a.** For that, we will be doing an SHA - 256 checks.

**b.** Find the .mds file for the version which you have downloaded.

**$ wget** [**https://dist.apache.org/repos/dist/release/hadoop/common/hadoop-2.7.7/hadoop-**](https://dist.apache.org/repos/dist/release/hadoop/common/hadoop-2.7.7/hadoop-) **2.7.7.tar.gz.mds**

**c.** After that, do the verification using the below command.

**$ shasum -a 256 hadoop-2.7.7.tar.gz**

**Output.**

**d489df3808244b906eb38f4d081ba49e50c4603db03efd5e594a1e98b09259c2 hadoop- 2.7.7.tar.gz**

**d.** Now, check the SHA-256 value

$ cat hadoop-2.7.7.tar.gz.mds

**Both output should match.**

**~/hadoop-2.7.7.tar.gz.mds**

...

**hadoop-2.7.7.tar.gz: SHA256 = D489DF38 08244B90 6EB38F4D 081BA49E 50C4603D B03EFD5E 594A1E98 B09259C2**

...

**Step 4:** Use the tar command to extract the file.

**$ tar -xzvf hadoop-2.7.7.tar.gz**

**Here:**

-x is for extracting flag

-z is for uncompressing the file.

-v for verbose output

-f specifies the extraction from the file.

**Step 5:** Move the extracted file to the /usr/local location

**$ sudo mv hadoop-2.7.3 /usr/local/hadoop**

**3)****Configuring the Hadoop to use java**

* Configure the Hadoop to use the java either in Hadoop's configuration file or using the environmental variable.
* Here /usr/bin/java and /etc/alternatives/java both are symlink to each other.
* Here, we have to use the -f flag to follow the symlink in the every part of the path that is mentioned.
* The sed will be used here to trim the path to get the bin/java. We have to do this to get the correct value of java Home from the output.

**Step 1:** Get the default java path.

**$ readlink -f /usr/bin/java | sed "s:bin/java::"**

**Output**

**/usr/lib/jvm/java-8-openjdk-amd64/jre/**

**Step 2:** Set the version of Java to the Home path of Hadoop.

The readlink command can be used to dynamically set the path .

* First, Open the hadoop-env.sh:

**$ sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

### Second,Use the redlink directly

Change the following JAVA\_HOME in env.sh

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

. . .

#export JAVA\_HOME=${JAVA\_HOME}

**export JAVA\_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")**

. . .

**4)****Running Hadoop**

**$ /usr/local/hadoop/bin/hadoop**

**Output:**

**Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]**

**CLASSNAME run the class named CLASSNAME**

**or**

**where COMMAND is one of:**

**fs run a generic filesystem user client**

**version print the version**

**jar <jar> run a jar file**

**note: please use "yarn jar" to launch**

**YARN applications, not this command.**

**checknative [-a|-h] check native hadoop and compression libraries availability**

**distcp <srcurl> <desturl> copy file or directories recursively**

**archive -archiveName NAME -p <parent path> <src>\* <dest> create a hadoop archive**

**classpath prints the class path needed to get the**

**credential interact with credential providers**

**Hadoop jar and the required libraries**

**daemonlog get/set the log level for each daemon**

If you get the above output, then it means that the Hadoop is up and running in the stand alone mode.

**5) Testing**

We will test whether it is configured properly or not.

We will use the example mapreduce program to test the Hadoop.

**Step 1:** The first step is to create a directory called input in the home screen.

**$ mkdir ~/input**

**Step 2: C**opy the Hadoop configuration to that file.

**$ cp /usr/local/hadoop/etc/hadoop/\*.xml ~/input**

**Step 3: R**un the Mapreduce hadoop-mapreduce-examples. It is a java archieve with a lot of programs inside.

Use the grep program from the Mapdreduce program.

We are going to find the occurrence of 'principal' at the end or before the declarative.

Since the expression is case sensitive, we could not find if it is capitalized.

**$/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar grep ~/input ~/grep\_example 'principal[.]\*'**

**Output:**

**File System Counters**

**FILE: Number of bytes read=1247674**

**FILE: Number of bytes written=2324248**

**FILE: Number of read operations=0**

**FILE: Number of large read operations=0**

**FILE: Number of write operations=0**

**Map-Reduce Framework**

**Map input records=2**

**Map output records=2**

**Map output bytes=37**

**Map output materialized bytes=47**

**Input split bytes=114**

**Combine input records=0**

**Combine output records=0**

**Reduce input groups=2**

**Reduce shuffle bytes=47**

**Reduce input records=2**

**Reduce output records=2**

**Spilled Records=4**

**Shuffled Maps =1**

**Failed Shuffles=0**

**Merged Map outputs=1**

**GC time elapsed (ms)=61**

**Total committed heap usage (bytes)=263520256**

**Shuffle Errors**

**BAD\_ID=0**

**CONNECTION=0**

**IO\_ERROR=0**

**WRONG\_LENGTH=0**

**WRONG\_MAP=0**

**WRONG\_REDUCE=0**

**File Input Format Counters**

**Bytes Read=151**

**File Output Format Counters**

**Bytes Written=37**

**Step 4:**

The results will be stored in the output directory and you can check it using cat.

**$** **cat ~/grep\_example/\***

**Output**

**6 principal**

**1 principal.**

The output indicates that the word principal was found after six non-occurrence.

### **RESULT:**

### Thus setting up of one node Hadoop cluster is successfully executed.