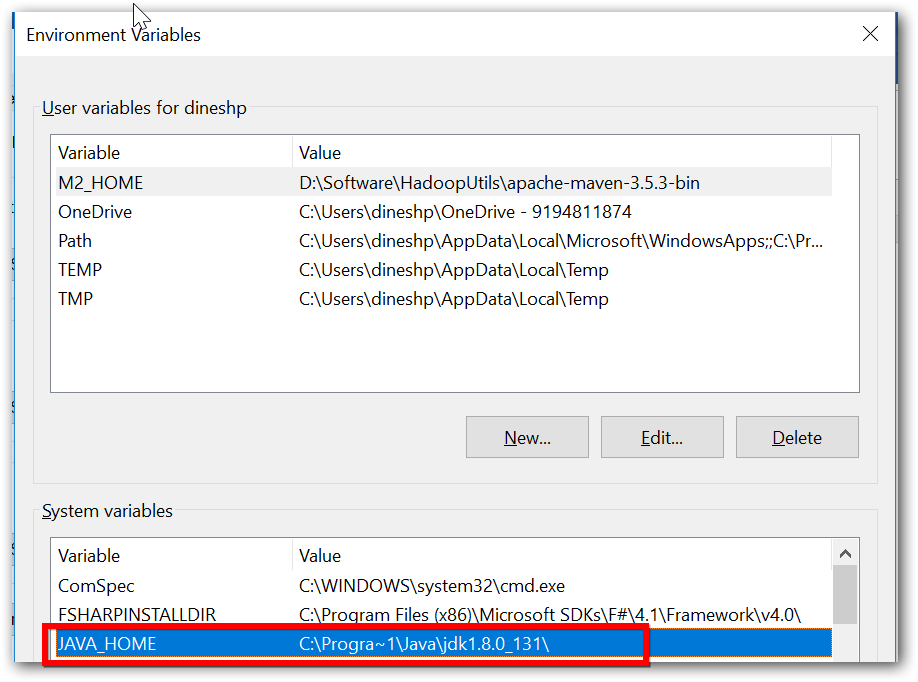
Hands on

# Pre-requisites

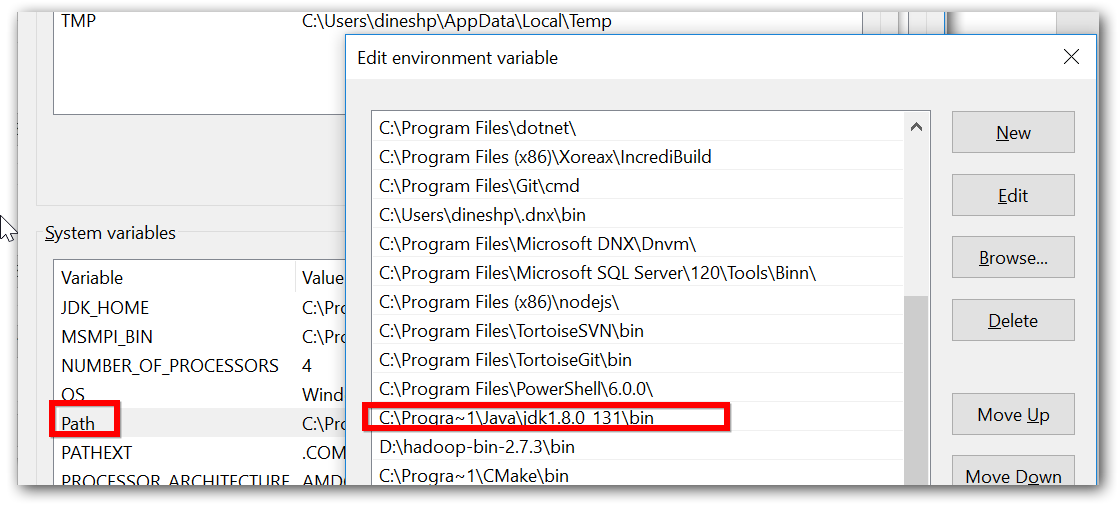
1. Install **7z1801-x64** in your machine.
2. Untar **“hadoop-3.0.1.tar”** using 7zip software installed above.
   1. Try to have the un-compressed folder in any other drive apart from main drive. Like if ‘C:\’ is your main drive, have Hadoop package in ‘D:\’ or some other drive.
   2. E.g. Have it simple like ‘D:\hadoop-3.0.1\bin’
3. Install Java 8 in your machine.
4. Set JAVA\_HOME in environment variables – system variables

<https://confluence.atlassian.com/doc/setting-the-java_home-variable-in-windows-8895.html>

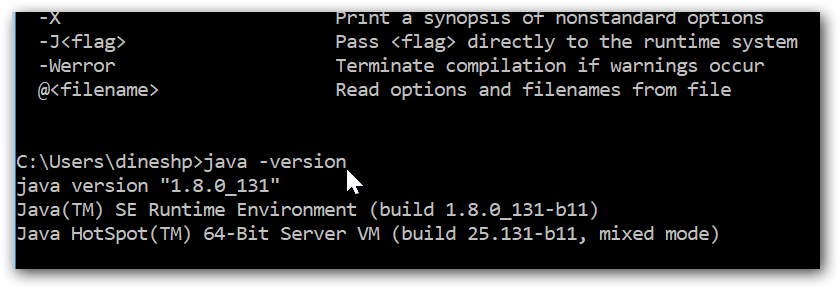
Note: Consider having your path has no spaces. You should have for 64-bit Java like, “C:\Progra~1\Java\jdk1.8.0\_161”



1. Add Java’s bin location PATH



1. Ensure java home and path is set properly.
   1. Open command prompt
   2. Execute ‘javac’ or ‘java -version’ command.



# Installation

Do the following changes in file specified in directory “..\..\hadoop-3.0.1\etc\hadoop”

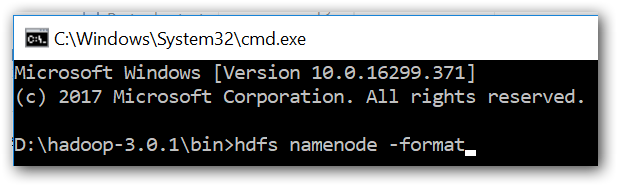
|  |  |  |
| --- | --- | --- |
| File name | Properties | Comments |
| hadoop-env.cmd | set HADOOP\_PREFIX=D:\hadoop-3.0.1  set HADOOP\_CONF\_DIR=%HADOOP\_PREFIX%\etc\hadoop  set YARN\_CONF\_DIR=%HADOOP\_CONF\_DIR%  set PATH=%PATH%;%HADOOP\_PREFIX%\bin | HADOOP\_PREFIX is the Hadoop package unzipped. |
| core-site.xml | <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://localhost:9000</value>  </property>  </configuration> |  |
| hdfs-site.xml | <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///d:/Metadata/NameNode</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:///d:/Metadata/DataNode</value>  </property>  </configuration> | Specify the directory where there won’t be any permission issues. |
| mapred-site.xml | <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>local</value>  </property>  <property>  <name>yarn.app.mapreduce.am.env</name>  <value>HADOOP\_MAPRED\_HOME=/hadoop-3.0.1</value>  </property>  <property>  <name>mapreduce.map.env</name>  <value>HADOOP\_MAPRED\_HOME=/hadoop-3.0.1</value>  </property>  <property>  <name>mapreduce.reduce.env</name>  <value>HADOOP\_MAPRED\_HOME=/hadoop-3.0.1</value>  </property> </configuration> | Specify the directory like in Linux format itself. |
| yarn-site.xml | <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>  <property>  <description>NM Webapp address.</description>  <name>yarn.nodemanager.webapp.address</name>  <value>{yourhostname}:8042</value>  </property>  </configuration> | Replace with your machine’s hostname in the property value  {yourhostname} |

# Format your cluster’s HDFS file system to make it use

1. Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,

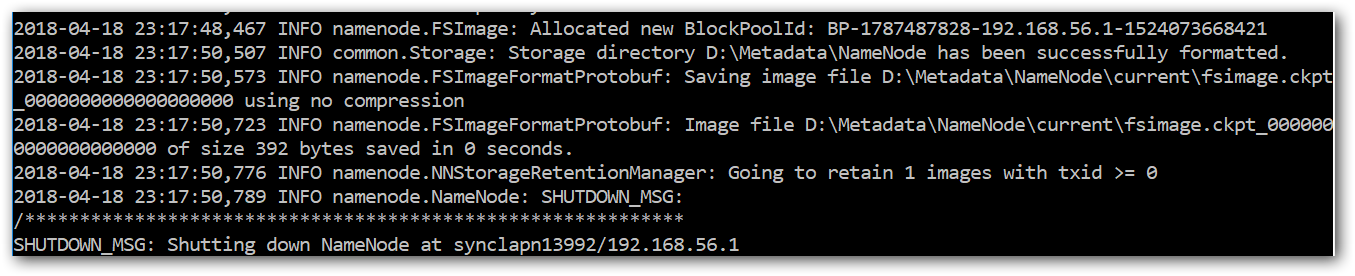
**Command:**

\bin>hdfs namenode -format



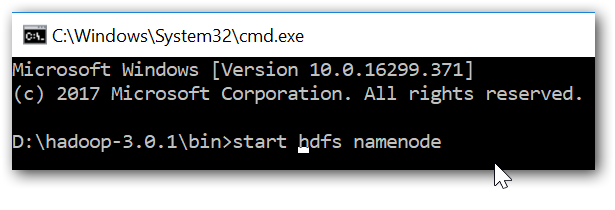
**Result:**

Ensure that format got successfully completed with similar message in below image.

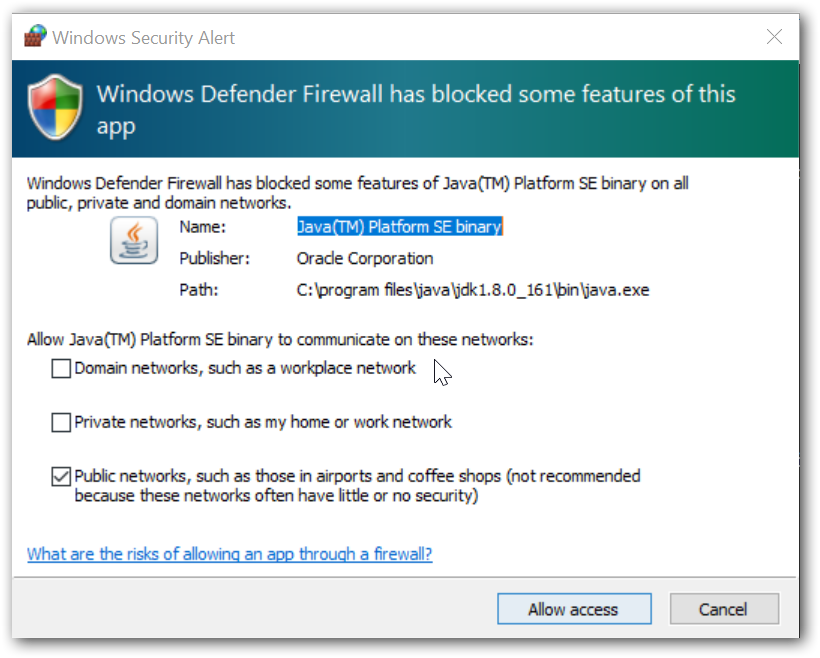


# Start Hadoop services

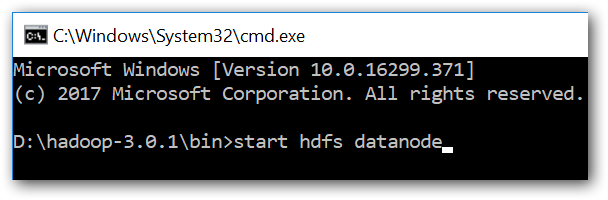
1. Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,
2. Start NameNode



Note: If such a popup is showed at any of below service, enter ‘Yes’



1. Start DataNode



# Work-out

1. Open HDFS bin location in command prompt,
2. Create a folder called ‘Data’,
   1. Command: hdfs dfs -mkdir /Data
   2. Result:
3. Upload any file into HDFS from local file system,

Note: Replace the file location properly

* 1. Command: hdfs dfs -put D:\Trip.csv /Data
  2. Result:

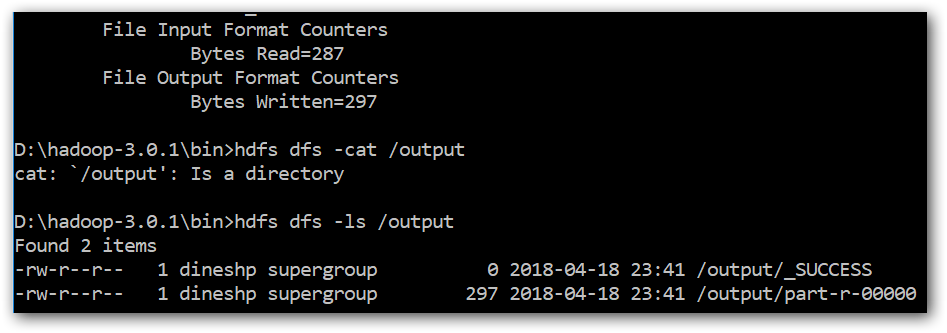
1. Perform Word count operation,

Note: Replace the jar file location properly with your Hadoop installed path

* 1. Execute the in-built sample,

bin> yarn jar "D:\hadoop-3.0.1\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.0.1.jar" wordcount /Data /output1

* 1. Result:



* 1. Check the result stored in file /output1/part-r-00000
  2. Command to list the “/output1” directory: hdfs dfs -ls /output1
  3. Open Web HDFS - <http://localhost:9870/dfshealth.html#tab-overview> and check the result.

1. Copy data between clusters,
   1. Command: hadoop distcp /Data /Data-Copy
2. HDFS health check,
   1. Command: hdfs fsck /