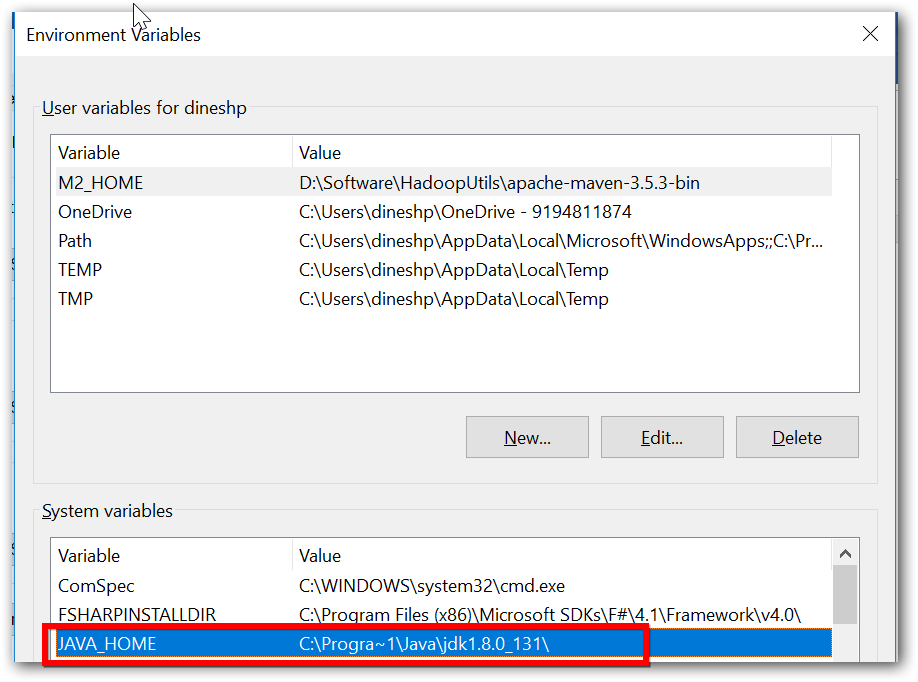
Hands on

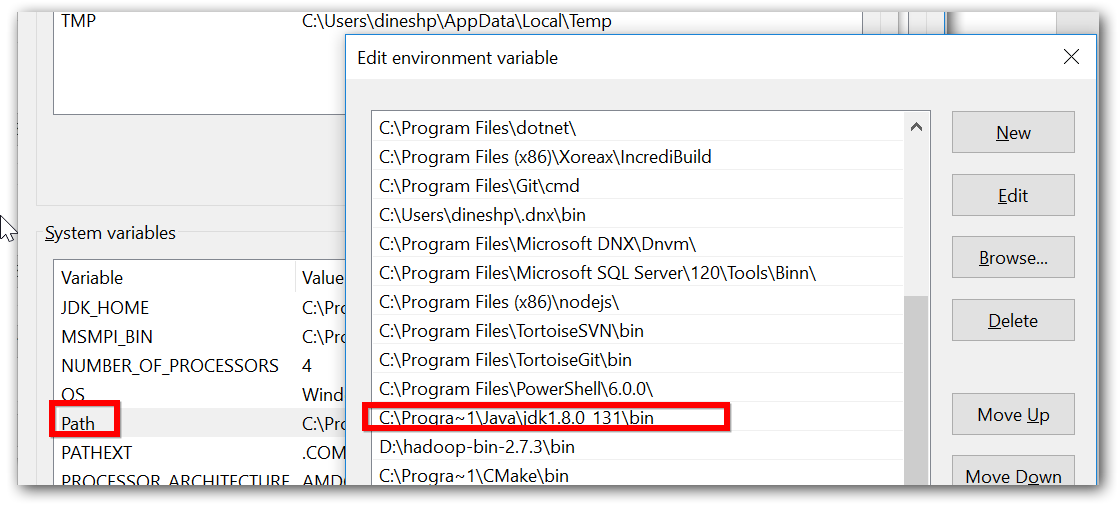
# Pre-requisites

1. Install **7z1801-x64** in your machine.
2. Untar **“hadoop-3.0.1.tar”** using 7z. Prefer any other drive apart from OS drive. For e.g. if ‘C:\’ is OS drive, have Hadoop package in ‘D:\’ or some other drive.
3. Install Java 8 in your machine.
4. Set JAVA\_HOME in environment variables – system variables

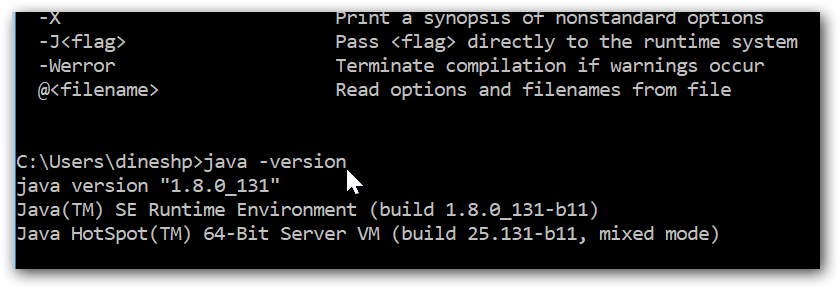
<https://confluence.atlassian.com/doc/setting-the-java_home-variable-in-windows-8895.html>



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.



# Hadoop cluster installation

Do the following changes in specified file located in directory “D:\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 location. |
| core-site.xml | <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://localhost:9000</value>  </property>  **<property>**  **<name>fs.azure.account.key.dineshazuredemo.blob.core.windows.net</name>**  **<value>xxxxxxxxxxxxxx</value>**  **<description>Storage account Access key copied from Azure portal</description>**  **</property>**  **<property>**  **<name>fs.AbstractFileSystem.wasb.impl</name>**  **<value>org.apache.hadoop.fs.azure.Wasb</value>**  **</property>**  **<property>**  **<name>fs.wasb.impl</name>**  **<value>org.apache.hadoop.fs.azure.NativeAzureFileSystem</value>**  **</property>**  **<property>**  **<name>fs.hdfs.impl</name>**  **<value>org.apache.hadoop.hdfs.DistributedFileSystem</value>**  **</property>**  **<property>**  **<name>fs.file.impl</name>**  **<value>org.apache.hadoop.fs.LocalFileSystem</value>**  **</property>**  </configuration> | Changes in **bold** are required if you need to configure blob storage in the cluster. If not specified, the cluster will be with local storage alone. |
| hdfs-site.xml | <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///d:/Data/NameNode</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:///d:/Data/DataNode</value>  </property>  </configuration> | Prefer having any other drive apart from ‘C:\’ |

**Note:** In case of blob storage copy the following jars from “D:\hadoop-3.0.1\share\hadoop\tools\lib” to “D:\hadoop-3.0.1\share\hadoop\hdfs\lib”

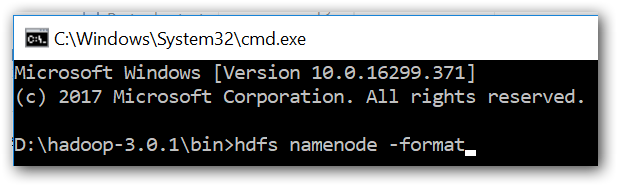
1. azure-storage-5.4.0.jar
2. hadoop-azure-3.0.1.jar

# Format your cluster and make your file system ready for use

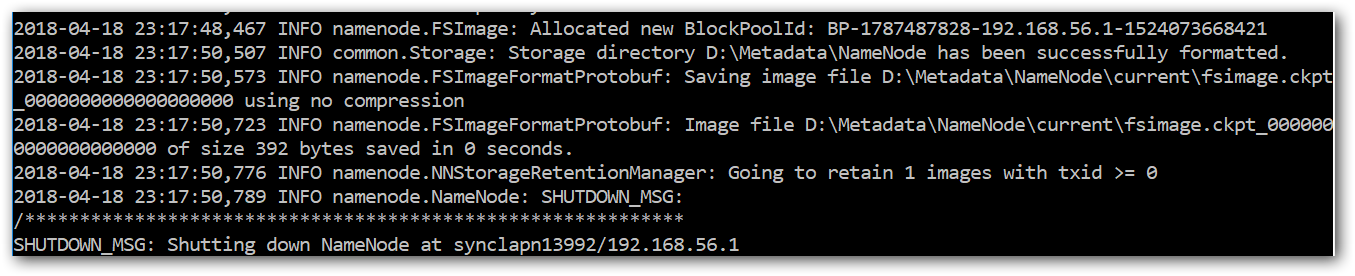
Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,

1. Format the cluster

\bin>hdfs namenode -format



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

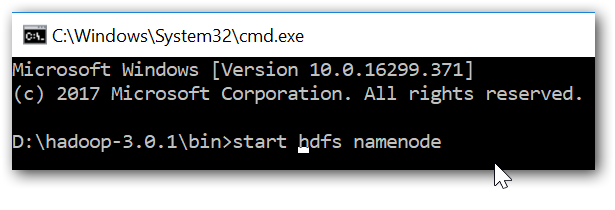


# Start Hadoop cluster services

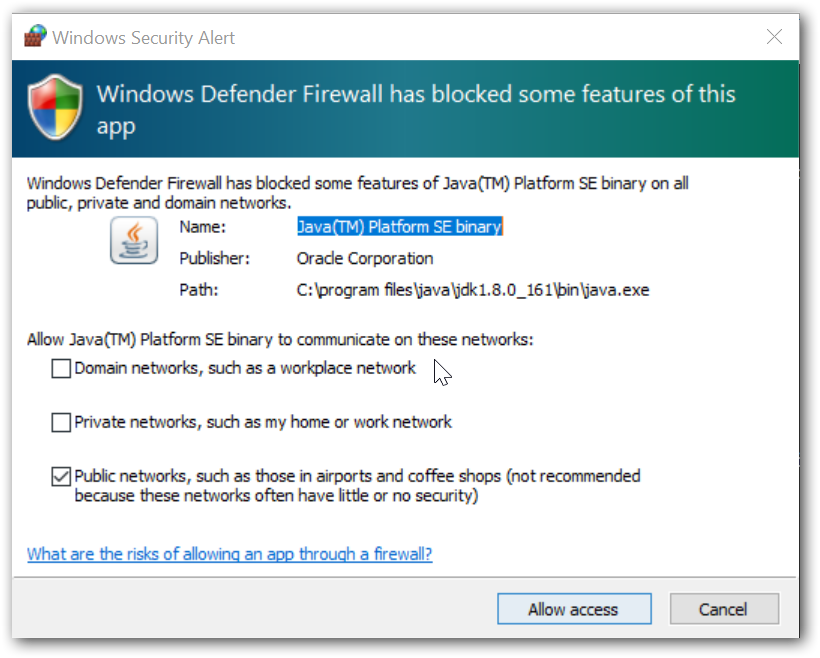
Open command prompt as ‘Administrator’ and navigate to Hadoop bin path,

1. Start NameNode

\bin>start hdfs namenode

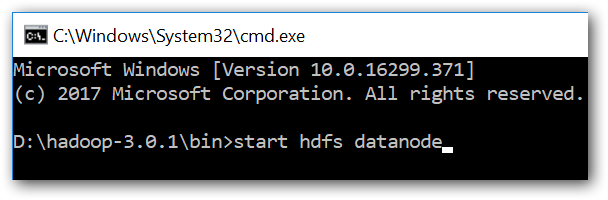


Note: If such a popup is showed while starting service, click ‘Allow access’



1. Start DataNode

\bin>start hdfs datanode



1. Web UI - HDFS - <http://localhost:9870/dfshealth.html#tab-overview>

# Work-out samples

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

1. Create a folder called ‘Data’,

**HDFS:**

**bin>** hdfs dfs -mkdir /Data

**BLOB:**

**bin>** hadoop fs -mkdir wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data

1. Upload data into HDFS from local file system,

**HDFS:**

**bin>** hdfs dfs -put "D:\Dataset\Customers.csv" /Data

**BLOB:**

**bin>** hdfs dfs -put "D:\Dataset\Customers.csv" wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data

1. Word count,

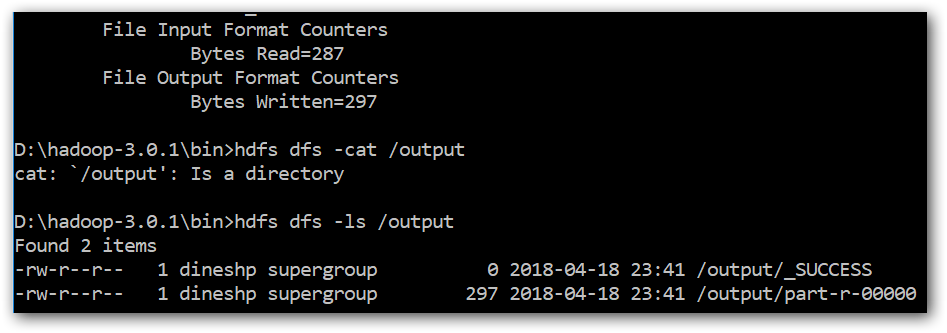
Replace the jar file location based on your Hadoop package location,

**HDFS:**

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

**BLOB:**

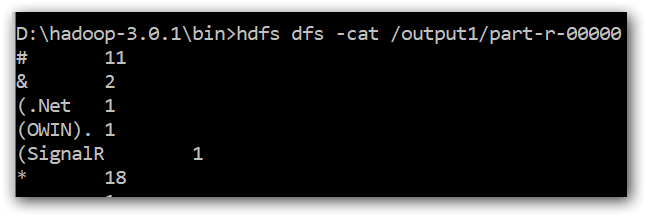
**bin>** yarn jar "D:\hadoop-3.0.1\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.0.1.jar" wordcount wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data wasb://newcontainer@dineshazuredemo.blob.core.windows.net/output1



1. The result stored in file /output1/part-r-00000. Command to list the “/output1” directory:

**bin>** hdfs dfs -ls /output1

1. Command to check the output of job: hdfs dfs -cat /output1/part-r-00000



1. Copy data between clusters,

**HDFS:** Copy from HDFS to HDFS

**bin>** hadoop distcp /Data /Data-Copy

**BLOB:** Copy from Azure Blob to Azure Blob

**bin>** hadoop distcp wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data-Copy

1. Copy from Azure Blob to Hadoop HDFS:

**bin>** hadoop distcp wasb://newcontainer@dineshazuredemo.blob.core.windows.net/Data /Data-Copy