

Hands-on 1 (HW) - Hadoop Installation Guide and HDFS

Note: You need to setup Virtual Machine by following **Assignment 0 - VM Setup** before setting up Hadoop.



If you have any issues while installing Hadoop, put up an issue on this google sheet

https://docs.google.com/spreadsheets/d/1ACrpf3xkHTdG0CrIaBCqcQVY3ljXsJI-8Dz8GK5t_S8/edit?usp=sharing

Do make sure to check if the issue you are facing has been resolved on the google sheets before adding the issue.

Every step is to be executed on the home directory. Use `cd` to move to home directory.

The commands in the guide use `pes2ug21cs532` as the notation for your username. If you have executed A0 correctly, then this should be your SRN in lowercase. This is important since the auto-evaluation depends on it. Verify your username by running `whoami` on the terminal.

Change any `/home/pes2ug21cs532/` to `/home/<your SRN>/`

Execute the following commands to move to the home directory and updating the package list and the system. This guide assumes that you are working with Ubuntu or a Debian based distribution.



Made with Super `update -y`

```
sudo apt upgrade -y
```

Downloads

Step 1 - Installing Java

Since Hadoop 3.3.3 may not support newer versions of Java, we install Java 8 using the following command.

```
sudo apt install openjdk-8-jdk -y
```

Check if Java is successfully installed and the version with the following commands.

```
java -version  
javac -version
```

Step 2 - Downloading Hadoop

Use the link given below to download and extract hadoop using the following commands.

```
cd  
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz  
tar xzf /home/$USER/hadoop-3.3.6.tar.gz
```

Installation

Step 1 - Setup passwordless SSH for Hadoop

We install the following packages to allow us to setup an ssh server on the system as well as a client to remote into it with the following commands.

```
sudo apt install openssh-server openssh-client -y
```



Made with Super

Setup passwordless SSH

Generate an SSH key pair and define the location it is to be stored in `id_rsa`. Then use the `cat` command to store the public key as `authorized_keys` in the `ssh` directory. Follow these exact commands with change in permissions.

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```

Verify passwordless SSH is setup and working with

```
ssh localhost
```

If the above command does not ask you for a password, you have successfully setup passwordless SSH.

Type `exit` or press `Ctrl+d` to quit the SSH session.

Step 2 - Single Node Deployment

The current setup is called pseudo-distributed mode, allows each Hadoop daemon to run as a single Java process. A Hadoop environment is configured by editing the following list of configuration files:

- `.bashrc`
- `hadoop-env.sh`
- `core-site.xml`
- `hdfs-site.xml`
- `mapred-site.xml`
- `yarn-site.xml`

Before editing the above mentioned files, we need to make a few directories for our namenodes and datanodes along with the required permissions.

```
cd
mkdir dfsdata
mkdir tmpdata
```



```
mkdir dfsdata/datanode
mkdir dfsdata/namenode
```

Change permissions for the directories using the following commands. Remember to replace `pes2ug21cs532` with your username.

Change `pes2ug21cs532` to your SRN at 9 locations

```
sudo chown -R pes2ug21cs532:pes2ug21cs532 /home/pes2ug21cs532/dfs
sudo chown -R pes2ug21cs532:pes2ug21cs532 /home/pes2ug21cs532/dfs
sudo chown -R pes2ug21cs532:pes2ug21cs532 /home/pes2ug21cs532/dfs
```

Editing and Setting up the ~/.bashrc config file

Open `.bashrc` with any text editor of your choice. This guide recommends using `nano`.


```
sudo nano ~/.bashrc
```

Scroll to the bottom of the file. Copy and paste the below mentioned statements to the end of the file.

Change `pes2ug21cs532` to your SRN at 1 location

```
#Hadoop Path Configs
export HADOOP_HOME=/home/pes2ug21cs532/hadoop-3.3.6
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS=-Djava.library.path=$HADOOP_HOME/lib/native
```

Press `Ctrl+o` to save and `Ctrl+x` to exit nano. Apply changes to `bash` with the following command.

 Made with Super `bashrc`

You can verify if the changes have been made by using the `echo` command and checking if the corresponding path gets printed in the terminal.

```
echo $HADOOP_HOME  
echo $PATH
```

Setup `hadoop-env.sh`

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Scroll down until you find the commented line `# export JAVA_HOME= .`. Uncomment the line and replace the path with your Java path. The final line should look like this

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```



Note: If your VM is running on a mac then change the above `JAVA_HOME` path to “`java-8-openjdk-arm64`” instead of “`java-8-openjdk-amd64`”

Save and exit the file as shown previously.

Setup `core-site.xml`

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Replace the existing configuration tags with the following

Change `pes2ug21cs532` to your SRN at 1 location

```
<configuration>  
<property>  
  <name>hadoop.tmp.dir</name>  
  <value>home/pes2ug21cs532/tmpdata</value>  
</property>
```



Made with Super

```
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

Save and exit the file.

Setup hdfs-site.xml

Open the file using

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

Replace the existing configuration tags with the following

Change pes2ug21cs532 to your SRN at 2 locations

```
<configuration>
<property>
  <name>dfs.name.dir</name>
  <value>/home/pes2ug21cs532/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/pes2ug21cs532/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Save and exit the file after making all the changes.

Setup mapred-site.xml

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```



Made with Super

existing configuration tags with the following

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

Save and exit the file.

Setup yarn-site.xml

Open the file with

```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Replace the existing configuration tags with the following

```
<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>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_COM
</property>
</configuration>
```



Save and exit the file.

Execute the following commands to move to the home directory and updating the package list and the system. This guide assumes that you are working with Ubuntu or a Debian based distribution.

```
cd
sudo apt update -y
sudo apt upgrade -y
```

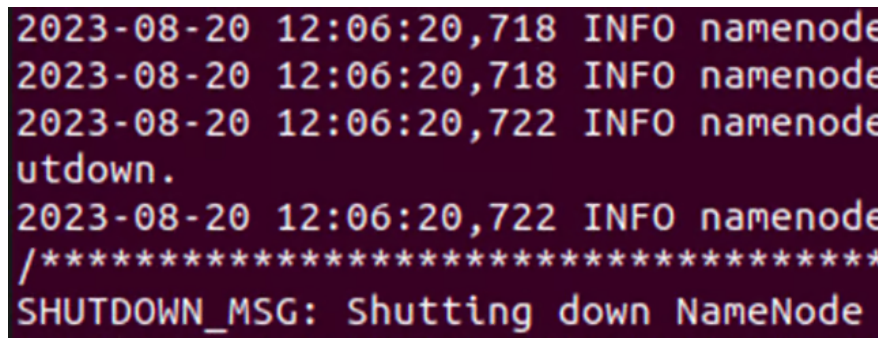
Step 3 - Format HDFS NameNode

Before starting Hadoop for the first time, the namenode must be formatted. Use the following command.

```
hdfs namenode -format
```

A **SHUTDOWN** message will signify the end of the formatting process.

If you have reached this stage, it signifies that you have successfully installed hadoop.

A terminal window with a dark background and light-colored text. It shows several log entries from the HDFS NameNode. The last entry is a 'SHUTDOWN_MSG: Shutting down NameNode' message, indicating the successful completion of the formatting process.

```
2023-08-20 12:06:20,718 INFO namenode
2023-08-20 12:06:20,718 INFO namenode
2023-08-20 12:06:20,722 INFO namenode
shutdown.
2023-08-20 12:06:20,722 INFO namenode
/*****
SHUTDOWN_MSG: Shutting down NameNode
```

Step 4 - Starting Hadoop

Navigate to the **hadoop** folder and execute the following commands.

start-all.sh is a shell script that is used to start all the processes that hadoop requires.

```
cd
cd hadoop-3.3.6/sbin/
./start-all.sh
```

Type **ps -ef** to find all the Java Processes started by the shell script.



Made with Super

see a total of 6 processes, including the **jps** process.

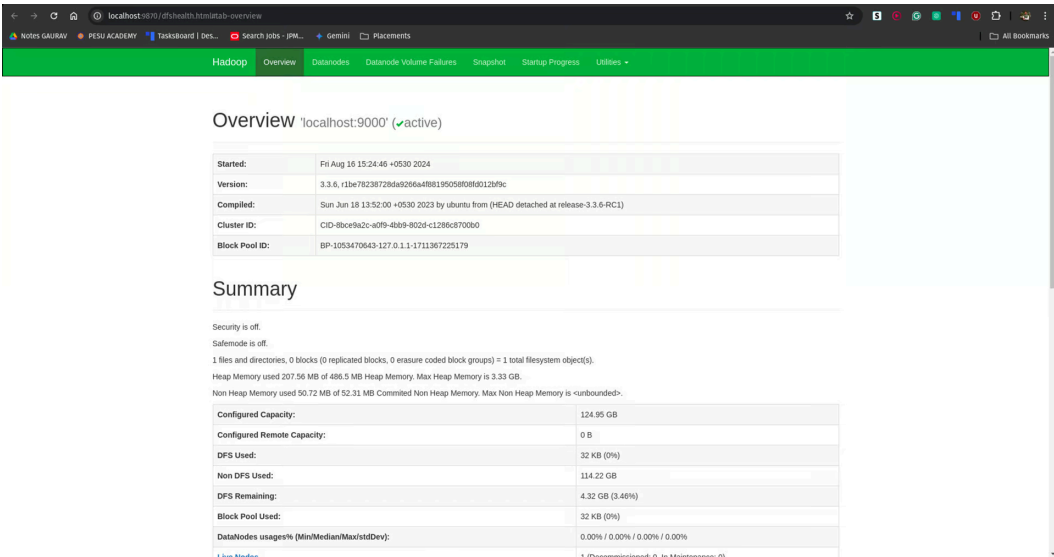
Note that the order of the items and the process IDs will be different

- 2994 DataNode
- 3219 SecondaryNameNode
- 3927 Jps
- 3431 ResourceManager
- 2856 NameNode
- 3566 NodeManager

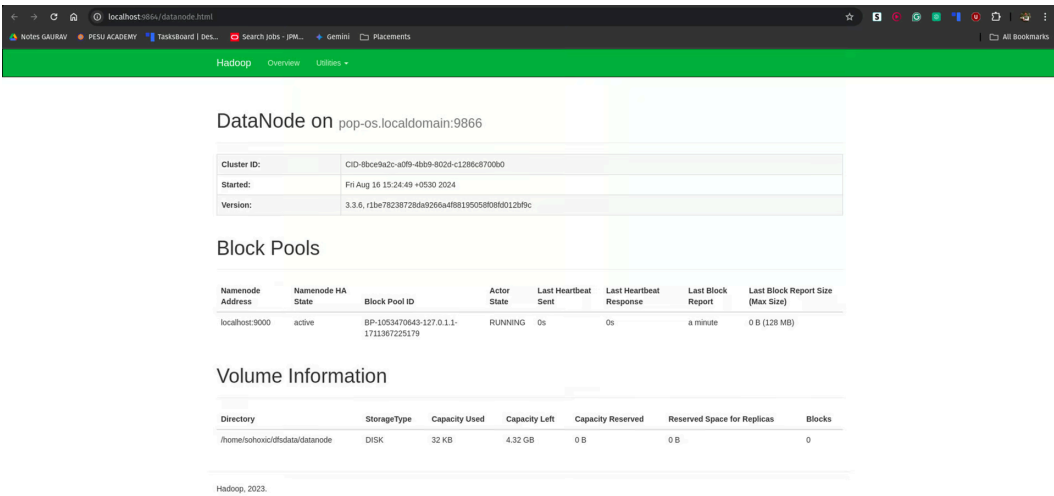
Step 5 - Accessing Hadoop from the Browser

You can access Hadoop on localhost on the following ports

- NameNode - http://localhost:9870



- DataNode - http://localhost:9864



- YARN Manager - http://localhost:8088

We will be using the Wordcount example to demonstrate the usage of Hadoop. Create a text file named `input.txt` with any content you want. Next, we will put this to the HDFS folder `/example` with the following command.

```
cd
hdfs dfs -mkdir /example
hdfs dfs -put input.txt /example
```

Run the following command for the wordcount example.

```
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-e
```



You can check the output with the following command.

```
hdfs dfs -cat /example/output/part-r-00000
```