Course: MMDS

*Tutorial 01*

**Set up Hadoop**

**Prerequisites**

* Operation system: Ubuntu 18.04 LTS (highly recommended)
* Chipset: x86 (optional to the expected version of Hadoop)
* If you do not have Ubuntu, consider virtual machines
  + VMWare Fusion (MacOS)
  + Oracle Virtual Box (Windows)
  + Docker (Windows)

**Installation**

**Install Hadoop**

1. Install Java 8 (highly recommended)

sudo apt install openjdk-8-jre-headless

sudo apt install openjdk-8-jdk-headless

1. Install ssh and pdsh

sudo apt install ssh

sudo apt install pdsh

1. Setup passphrase for ssh

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 0600 ~/.ssh/authorized\_keys

# ensure the file ~/.ssh/authorized\_keys exists

# check whether you can ssh to localhost

ssh localhost

1. Configure rcmd to ssh as default

sudo nano /etc/pdsh/rcmd\_default

# add “ssh” to the file

# save & quit

# HOẶC có thể dùng lệnh sau

# sudo echo “ssh” > /etc/pdsh/rmcd\_default

1. Download [Hadoop](http://www.apache.org/dyn/closer.cgi/hadoop/common/) 3.2.1 (highly recommended)

<https://hadoop.apache.org/release/3.2.1.html>

cd Desktop

wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

tar -xvf hadoop-3.2.1.tar.gz

1. Declare JAVA\_HOME for Hadoop

# cd to the extracted folder of Hadoop

nano etc/hadoop/hadoop-env.sh

# add this line to the end of the file

# check your own Java path if different

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

export PATH=${JAVA\_HOME}/bin:${PATH}

export HADOOP\_CLASSPATH=${JAVA\_HOME}/lib/tools.jar

# save & quit

1. Verify installation

# cd to the extracted folder of Hadoop

bin/hadoop

**Set up Pseudo-Distributed Mode**

**Configuration**

Edit these following files

**etc/hadoop/core-site.xml**

<configuration>

    <property>

        <name>fs.defaultFS</name>

        <value>hdfs://localhost:9000</value>

    </property>

</configuration>

**etc/hadoop/hdfs-site.xml**

<configuration>

    <property>

        <name>dfs.replication</name>

        <value>1</value>

    </property>

</configuration>

**Run a MapReduce job locally**

1. Format the filesystem

bin/hdfs namenode -format

1. Start NameNode daemon and DataNode daemon

sbin/start-dfs.sh

# check log output in .../logs as needed

# if you fail to start a daemon then,

sbin/stop-all.sh

sudo rm -rf /tmp/\*

sudo reboot

bin/hdfs namenode -format -force

# then try to start again

1. Browse the web interface for the NameNode; by default it is available at

<http://localhost:9870/>

A screenshot of a computer

Description automatically generated

1. Make the HDFS directories required to execute MapReduce jobs:

bin/hdfs dfs -mkdir /user

bin/hdfs dfs -mkdir /user/<username>

1. Copy the input files into the distributed file system:

 bin/hdfs dfs -mkdir input

 bin/hdfs dfs -put etc/hadoop/\*.xml /user/abc/input

A screen shot of a computer

Description automatically generated

1. Run some of the examples provided:

bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar grep input output 'dfs[a-z.]+'

#  ensure the corresponding jar file exists in the folder mapreduce/

A computer screen shot of a computer code

Description automatically generated

***Note: after setting up YARN, this step requires [Connecting to ResourceManager at /0.0.0.0:8032]***

1. Examine the output files: Copy the output files from the distributed file system to the local filesystem and examine them:

bin/hdfs dfs -get output output

cat output/\*

or

View the output files on the distributed file system:

bin/hdfs dfs -cat output/\*

1. When you’re done, stop the daemons with:

sbin/stop-dfs.sh

**Execute job on YARN**

The following instructions assume that 1. ~ 4. steps of the above instructions are already executed.

1. Configure parameters as follows:

**etc/hadoop/mapred-site.xml**

<configuration>

    <property>

        <name>mapreduce.framework.name</name>

        <value>yarn</value>

    </property>

    <property>

        <name>mapreduce.application.classpath</name>

        <value>$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/\*:$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/lib/\*</value>

    </property>

</configuration>

**etc/hadoop/yarn-site.xml**

<configuration>

    <property>

        <name>yarn.nodemanager.aux-services</name>

        <value>mapreduce\_shuffle</value>

    </property>

    <property>

        <name>yarn.nodemanager.env-whitelist</name>

        <value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>

    </property>

</configuration>

1. Start ResourceManager daemon and NodeManager daemon:

sbin/start-yarn.sh

1. Browse the web interface for the ResourceManager; by default it is available at:

<http://localhost:8088/>

A screenshot of a computer

Description automatically generated

1. Run a MapReduce job.
2. When you’re done, stop the daemons with:

sbin/stop-yarn.sh

# Read more about modes of Hadoop [here](https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html)

* Local (Standalone) Mode
* Pseudo-Distributed Mode
* Fully-Distributed Mode

**References**

* <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>
* <https://www.programmersought.com/article/93394144266/>
* clean up Hadoop: <https://stackoverflow.com/questions/26545524/there-are-0-datanodes-running-and-no-nodes-are-excluded-in-this-operation>
* Turn off safemode: <https://stackoverflow.com/questions/15803266/name-node-is-in-safe-mode-not-able-to-leave>