Installation of Apache Hadoop in Pseudo-Distributed Operation

> Hadoop can also be run on a single-node in a pseudo-distributed mode where each Hadoop daemon runs in a separate Java process.

We will use the official installation documentation from Apache Hadoop

SSH

SSH into your machine with `ssh student@bdlc-XX.el.eee.intern`, where `XX` is your personal virtual machine number.

Create HDFS Folders

As user `student`, create the two folders in our `/data/` directory:

```
sudo mkdir -p /data/hdfs/namenode
sudo mkdir -p /data/hdfs/datanode
```

and give the `hadoop` user the access rights to the folders:

```
sudo chown hadoop:root -R /data/hdfs/
```

Setup HDFS / YARN / MapReduce

Switch to the user `hadoop`.

```
su - hadoop
```

Edit your bash profile in `~/.bashrc` and add these three lines at the end:

```
# Hadoop Config
export PDSH_RCMD_TYPE=ssh
export HADOOP_HOME=/home/hadoop/hadoop
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

```
nano ~/.bashrc
```

Activate the changes with:

```
source ~/.bashrc
```

General Note about `*.xml` Files

We will edit several `*.xml` files for `hadoop`. Place the `<property>` attributes between the `<configuation>` tags:

```
<configuration>
<!-- properties go here -->
</configuration>
```

Core Site / HDFS Site

Replace / add the configuration part in `~/hadoop/etc/hadoop/core-site.xml` with the content:

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

```
nano ~/hadoop/etc/hadoop/core-site.xml
```

for `~/hadoop/etc/hadoop/hdfs-site` with:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.name.dir</name>
        <value>file:///data/hdfs/namenode</value>
    </property>
    <property>
        <name>dfs.data.dir</name>
        <value>file:///data/hdfs/datanode</value>
    </property>
</configuration>
```

```
nano ~/hadoop/etc/hadoop/hdfs-site.xml
```

Setup Passphraseless ssh

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

Test the setup, the `ssh` command should work *without* a password!

```
# say yes, if this question comes: Are you sure you want to continue
connecting (yes/no/[fingerprint])? yes
ssh localhost
```

Exit this shell again:

```
exit
```

Format HDFS

Format HDFS - Look out for ERRORS or WARNINGS.

```
~/hadoop/bin/hdfs namenode -format
```

Starting Daemons

Start the `hdfs` daemon with:

```
~/hadoop/sbin/start-dfs.sh
```

You should see something like:

```
hadoop/sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bdlc-test]
```

You can verify that hdfs is running by performing a `jps`:

```
20530 SecondaryNameNode
20166 NameNode
20282 DataNode
20733 Jps
```

On your personal computer - you can access the `namenode` dashboard under http://bdlc-XX.el.eee.intern:9870/

Create `hadoop`'s user folder in `hdfs`:

```
~/hadoop/bin/hdfs dfs -mkdir /user
~/hadoop/bin/hdfs dfs -mkdir /user/hadoop
```

YARN

Replace the configuration part in `~/hadoop/etc/hadoop/mapred-site.xml` with:

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

```
nano ~/hadoop/etc/hadoop/mapred-site.xml
```

Replace the configuration part in `~/hadoop/etc/hadoop/yarn-site.xml` with:

```
<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,CLASS
PATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,PATH,LANG,TZ,HADOOP_MA
PRED_HOME</value>
  </property>
</configuration>
```

```
nano ~/hadoop/etc/hadoop/yarn-site.xml
```

Start Yarn

```
~/hadoop/sbin/start-yarn.sh
```

You can verify that `yarn` is running by performing a `jps`:

```
SecondaryNameNode
NodeManager         # here it is
NameNode
ResourceManager     # here it is
DataNode
Jps
```

On your computer, you can access the `namenode` dashboard under http://bdlc-XX.el.eee.intern:8088/

HistoryServer

Start the `historyserver`:

```
mr-jobhistory-daemon.sh start historyserver
```

And again, with `jps` you can verify that the `historyserver` is up and running:

```
SecondaryNameNode
DataNode
NameNode
JobHistoryServer    # here it is
NodeManager
Jps
ResourceManager
```

Setup Python3

We will use `python3` in our module. Install the following packages as user `student`:

```
sudo apt install python-is-python3
sudo apt install python3-pip
```

Verify that we have `Python 3.8.10` activated by checking the version:

```
python --version
```

Homework

Check the Homework

References

- Hadoop - Single Cluster
- HadoopStreaming