— Setting up a Single Node Cluster — # 3.4B Pseudo-Distributed Operation

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

NOTE: Setting up a pseudo-distributed operation requires ssh and pdsh to be setup in addition to a Hadoop Installation and setting up of Hadoop's Environment Variables.

NOTE: Here we are setting up only Hadoop and HDFS. We are not running YARN just yet. Complete this and move onto 3.4C to set up YARN.

Setting up HDFS and MapReduce

Configuring HDFS

The following 2 must be edited to have the following structure.

1. **core-site.xml** - Configures IP details of the HDFS site

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

2. **hdfs-site.xml** - Configures the locations of the NameNode and the DataNodes

```
<configuration>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///opt/hadoop_tmp/hdfs/datanode</value>
  </property>

  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///opt/hadoop_tmp/hdfs/namenode</value>
  </property>

  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

NOTE: We set dfs.replication to 1 because this is a one-machine cluster - we can't replicate files any more than once here.

NOTE: The directories assigned above ***/opt/hadoop_tmp/hdfs/datanode*** and ***/opt/hadoop_tmp/hdfs/namenode*** do not exist yet. It must be created and permissions must be adjusted to 'read-able and write-able' by the current user.

```
sudo mkdir -p /opt/hadoop_tmp/hdfs/datanode
sudo mkdir -p /opt/hadoopt_tmp/hdfs/namenode
sudo chown guru:guru -R /opt/hadoop_tmp
```

NOTE: The `-p` flag (equivalent to `--parents`) of mkdir is used for making parent directories as needed.

3. **Format the HDFS with** `hdfs namenode -format (-force)` > **NOTE:** All data on hdfs will be deleted with this operation. For more information, check sources. > **NOTE:** You should get a bunch of output and then a 'SHUTDOWN_MSG'
4. Start up hadoop. > Run the following command to run HDFS `start-dfs.sh` **NOTE:** logs are $HADDOP_LOG_DIR (defaults to $HADOOP_HOME/logs)
5. Ensure that HDFS is running correctly > The following command lists all JVM's running on the current machine `jps` **NOTE:** You should see a NameNode and a DataNode, at minimum in that list.

Check that HDFS is behaving correctly

We will run a MapReduce job to ensure that Hadoop and HDFS is configured correctly.

1. Create a directory in HDFS and then listing the contents of the HDFS > `hdfs dfs -mkdir /test //creates directory hdfs dfs -ls / //lists contents of HDFS's root` You should be able to see your directory when you list the contents of hdfs.
2. Next create a user directory and the input directory > `hdfs dfs -mkdir -p /user/<username> hdfs dfs -mkdir -p /user/guru //what the above command is for me hdfs dfs -mkdir input //creates this directory in <username>` NOTE: the `input` directory of HDFS is created `/user/<username>` by default. If you want to change the home directory, refer SOURCES, point no. 4.
3. Copy the input files into the distributed filesystem > `cd $HADOOP_HOME // equivalent to 'cd opt/hadoop' hdfs dfs -put etc/hadoop/*.xml input //copies files into input`
4. Run one of the provided examples > `hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-example grep input output 'dfs[a-z.]+'`
5. Examine the output files. > Viewing the output files on HDFS: `hdfs dfs -cat output/*` > Or you could copy the output files from HDFS to your

local filesystem by: `hdfs dfs -get output output  cat output/*`

6. When you're done, stop the daemons. > `stop-dfs.sh`

   NOTE: If you set the mapreduce framework as yarn in yarn-site.xml, then the above commands will not work correctly unless yarn has also been set up properly. Setting up yarn will build upon this and will be covered next.

— Troubleshooting HDFS: 1.Datanode not being seen with jps

```
stop-all.sh
cd /tmp
rm -Rf hadoop-<username>
hadoop namenode -format
start-dfs.sh
jps
```

NOTE: If that doesn't work, stop-all.sh cd /tmp rm -Rf hadoop- rm -Rf hadoop cd /opt sudo rm -Rf hadoop_tmp sudo mkdir -p hadoop_tmp/hdfs/namenode sudo mkdir -p hadoop_tmp/hdfs/datanode sudo chown guru:guru -R /opt/hadoop_tmp hadoop namenode -format start-dfs.sh jps

NOTE: The cause hasn't been found out yet. Could be something to do with namenode, datanode, or permissions (possible explanations with amount of information gathered till now)

Sources:

Sources

1. Apache Hadoop documentation (Main)
2. Dev - Hadoop installation tutorial(Main)
3. What does 'hdfs namenode -format' do?
4. Default HDFS home directory (steps 7 and 8)
5. HDFS Operations - clarifies syntax and meanings of HDFS operations
6. Where does HDFS store data on the local filesystem
7. Read answer of and comments on AdreianKhisbe's answer - tells you tmp directory is at /tmp by default, if hadoop.tmp.dir is not configured; also check user persmissions

Other potentially useful resources

1. Running Hadoop on a multi-node cluster