Hadoop Ecosystem
Installation High Level Steps
Linux/MacOS

JDK

Extract jdk to destination (i.e JAVA_HOME)

Hadoop User

If possible, create a separate hadoop named user to manage hadoop operations/services.

Common Steps:

Extract downloads to destination. Make user who will be starting hadoop services able to read/write the folder/owns.

sudo tar xvzf <x.tar.gz> --directory=<destination>
sudo chown –R <user>:<group> <destination>
sudo ln –s <destination> <short_name_destination>

Configure environment variables to be able to run Hadoop and related commands available system wide.

Domain/HostName based configuration OR IP based configuration. May need edit hostname and /etc/hosts file.

Hadoop Framework:

http://hadoop.apache.org

User Equivalence Establishment: Each nodes shall be able to communicate/ssh to each other passwordless.

Create required folder in local file system about namenode, datanode, tmp and own them by Hadoop as required.

Configure environment variables to be able to run Hadoop and related commands available system wide.

Configure hadoop internal __env.sh with JAVA_HOME.

The hostname/ip setup and validation for hostname and canonical name match.

Configure Hadoop configuration properties. (Set up standalone or pseudo or cluster)

(In the case of cluster extra configuration of master and slaves information and hdfs access url:port extra)

Format HDFS

Start-dfs.sh start-yarn.sh and start job history server.

Verify processes using jps command. Pid may vary.

```
Jigar-Pandyas-MacBook:~ JigarPandya$ jps
1299 Jps
803 NameNode
1108 ResourceManager
884 DataNode
982 SecondaryNameNode
1274 JobHistoryServer
1195 NodeManager
Jigar-Pandyas-MacBook:~ JigarPandya$
```

Monitor through browser urls.

http://localhost:50070/dfshealth.html#tab-overview

http://localhost:50075/datanode.html

http://localhost:8088/cluster/nodes

Learn hdfs access commands; Browse hdfs from gui.

May create /user/<username> folder in hdfs to be able to refer relative addressing.

Compile and run wordcount program of map-reduce example, monitor and check output within hdfs.

Stop job history server, Stop-yarn.sh stop-dfs.sh

MongoDB

<It is not part of Hadoop Ecosystem as of now.>

https://www.mongodb.com

mkdir /data/db directory and own by the user

start mongod and keep the window open or run in background. It's a server process.

start mongo and test help() or version(); It's a client.

quit();

```
Jigar-Pandyas-MacBook:~ JigarPandya$ ps -ef | grep mongo
  501  1349   371   0  1:47AM ttys001    0:02.05 mongod
  501  1360   381   0  1:49AM ttys002    0:00.25 mongo
  501  1364   384   0  1:50AM ttys003    0:00.00 grep mongo
Jigar-Pandyas-MacBook:~ JigarPandya$
```

Hive

https://cwiki.apache.org/confluence/display/Hive/GettingStarted#GettingStarted-InstallingHivefromaStableRelease

Start Hadoop framework as hive writes the relations in Hadoop warehouse.

Create below required folder for hive within hdfs and make group writable:

```
hadoop fs -mkdir  -p      /tmp
hadoop fs -mkdir  -p      /user/hive/warehouse
hadoop fs -chmod g+w   /tmp
hadoop fs -chmod g+w   /user/hive/warehouse
```

```
add HIVE_HOME=<hive-install-dir> into bash profile ~/.bash_profile
```

Pig internally bundled with derby db for metadata. If you want to configure mysql or other create and configure conf/hive-site.xml.

Set Hadoop and java home within conf/hive-env.sh

i.e.

HADOOP_HOME=/opt/hadoop
JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home

Run below command to instatiate derby db default:

```
schematool -dbType derby -initSchema
```

Run hive command, type in show tables; If it show OK. Quit; Successful.

Pig

<pig can use local as well as hdfs >

http://pig.apache.org/docs/r0.17.0/index.html

Pig uses Hadoop for creating tables and loading files data within.

Run pig command, at the bottom few steps ensure that it shows correct Hadoop instance
i.e. “Connecting to hadoop file system at: hdfs://localhost:9000/”

help;
shall show you relevant help;

quit;

pig –x local; runs local.

Jigar-Pandyas-MacBook:~ JigarPandya$ jps
803 NameNode
1108 ResourceManager
1380 RunJar
884 DataNode
982 SecondaryNameNode
1274 JobHistoryServer
1530 Jps
1195 NodeManager
Jigar-Pandyas-MacBook:~ JigarPandya$

Jigar-Pandyas-MacBook:~ JigarPandya$ ps -ef |grep hive
  501 1380   382   0 1:52AM ttys004   0:28.36
/Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home/bin/java -Xmx256m -
Djava.net.preferIPv4Stack=true -Dhadoop.log.dir=/opt/hadoop/logs -Dhadoop.log.file=hadoop.log -
Dhadoop.home.dir=/opt/hadoop -Dhadoop.id.str=JigarPandya -Dhadoop.root.logger=INFO,console -
Dhadoop.policy.file=hadoop-policy.xml -Djava.net.preferIPv4Stack=true -Dproc_hivecli -
Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/opt/hive/conf/parquet-
logging.properties -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.util.RunJar /opt/hive/lib/hive-cli-
2.3.2.jar org.apache.hadoop.hive.cli.CliDriver
  501 1532   389   0 1:54AM ttys005   0:00.00 grep hive
Jigar-Pandyas-MacBook:~ JigarPandya$

HBase

http://hbase.apache.org

http://hbase.apache.org/book.html#quickstart

Make data storage folder on local file system for HBase

mkdir -p /opt/hbase/hbase_storage/
chown -R <user>:<group> /opt/hbase/hbase_storage/

Create hbase hdfs directory and configure accordingly. Your Hadoop url.

hadoop dfs -mkdir hdfs://localhost:9000/hbase

In hbase-env.sh update JAVA_HOME as required:
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home

Configure required properties for root dir, quorum data dir, backup masters, regionservers.

Start-hbase.sh

http://localhost:16010/master-status

jps shall show

Jigar-Pandyas-MacBook:~ JigarPandya$ jps
803 NameNode
1108 ResourceManager
1380 RunJar
884 DataNode
982 SecondaryNameNode
1274 JobHistoryServer
1195 NodeManager
1629 HMaster
1677 Jps
Jigar-Pandyas-MacBook:~ JigarPandya$

Stop-hbase.sh to stop the process.

Important links:

http://hadoop.apache.org/docs/current/

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml

http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-common/yarn-default.xml

https://www.mongodb.com

https://cwiki.apache.org/confluence/display/Hive/GettingStarted#GettingStarted-InstallingHivefromaStableRelease

http://pig.apache.org/docs/r0.17.0/index.html

http://hbase.apache.org

http://hbase.apache.org/book.html#quickstart

Document By:

Prof. Jigar M. Pandya
jigarpandya.ce@ddu.ac.in