Start instance

```
aws ec2 run-instances
--image-id ami-5189a661
--count 1
--instance-type t2.micro
--key-name aws_keypair
--security-groups hadSecGroup
```

Install Java

```
$ sudo apt-add-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

Add Hadoop group and user

```
$ sudo addgroup hadoop
$ sudo adduser --ingroup hadoop hduser
```

Download and unpack Hadoop

 # NB Find nearest mirror to your location

$ wget http://apache. (…) /hadoop-2.7.1.tar.gz

 # unpack to /usr/local

$ sudo tar -zxvf hadoop-2.7.1.tar.gz -C /usr/local

 # clean up

$ rm hadoop-2.7.1.tar.gz

Editing system wide profile - I

 # create file
$ sudo touch /etc/profile.d/custom.sh

 # add execute permission

$ sudo chmod +x /etc/profile.d/custom.sh

 # edit file
$ sudo vim /etc/profile.d/custom.sh

Editing System Wide Profile - II

 # add lines

#!/bin/sh

export PATH=$PATH:/usr/local/hadoop-2.7.1/bin/
export PATH=$PATH:/usr/local/hadoop-2.7.1/sbin/
export JAVA_HOME=/usr/lib/jvm/java-8-oracle/

 # execute (NB will run on next login)
$. /etc/profile.d/custom.sh

Checking Environment Variables

 # Show

$ printenv

PATH=/usr/local/sbin:(...):/usr/local/hadoop-2.7.1/bin/

(...)

JAVA_HOME=/usr/lib/jvm/java-8-oracle/

(...)

Hadoop commands

 # show version
$ hadoop version
Hadoop 2.7.1
(…)

 # all commands
$ hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
(…)
Most commands print help when invoked w/o parameters.

Hadoop Help

fs

$ hadoop fs

Usage: hadoop fs [generic options]

[-appendToFile <localsrc> ... <dst>]

[-cat [-ignoreCrc] <src> …]

(…)

[-copyFromLocal [-f] [-p] [-l] <localsrc> ... <dst>]

[-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]

(...)

Standalone Operation

From documentation:

“By default, Hadoop is configured to run in a non-distributed mode, as a single Java process. This is useful for debugging.”

Running a job

 # Standalone Operation (default)

$ mkdir input

$ cp /usr/local/hadoop-2.7.1/etc/hadoop/*.xml input

$ hadoop jar /usr/local/hadoop-2.7.1/share/hadoop/mapreduce/**hadoop-mapreduce-examples-2.7.1.jar** grep input **output** 'dfs[a-z.]+'

$ cat output/*

1 dfsadmin

$ hadoop jar /usr/local/hadoop-2.7.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input **output** 'security[a-z.]+'

(…)

org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory file:/home/ubuntu/output already exists

(...)

Pseudo-Distributed Operation

From documentation:

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

Configuration

Edit files

$ sudo vim /usr/local/hadoop-2.7.1/etc/hadoop/core-site.xml

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

$ sudo vim /usr/local/hadoop-2.7.1/etc/hadoop/hdfs-site.xml

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

Setup passphraseless ssh

 # check
$ ssh localhost

 # setup if required
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
$ export HADOOP_PREFIX=/usr/local/hadoop-2.7.1

Other configurations

 # edit hadoop-env.sh

$ sudo vim /usr/local/hadoop-2.7.1/etc/hadoop/hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-oracle/

 # add user to group

$ sudo usermod -a -G hadoop ubuntu

 # setup log directory

$ sudo mkdir /usr/local/hadoop-2.7.1/logs

$ sudo chown ubuntu:hadoop /usr/local/hadoop-2.7.1/logs/

Execution - I

 # Format the filesystem
$ hdfs namenode -format

 # Start NameNode daemon and DataNode daemon
$ start-dfs.sh

 # Install Apache
$ sudo apt-get install apache2
$ sudo /etc/init.d/apache2 start

 # Browse the web interface for the NameNode
http://localhost:50070/

Execution - II

 # Create HDFS directory

$ hdfs dfs -mkdir /input

 # Copy input files into HDFS

$ hdfs dfs -put <hadoop dir>/etc/hadoop/* /input

NB On error stop hdfs,clear /tmp/hadoop-user and reformat namenode

 # Run the example

$ hadoop jar /usr/local/hadoop-2.7.1/share/hadoop/mapreduce/**hadoop-mapreduce-examples-2.7.1.jar** grep /input **output** 'dfs[a-z.]+'

View output files and stop HDFS

 # List HDFS files
$ hdfs dfs -cat output/*

 # Copy to local filesystem
$ hdfs dfs -get output output

 # Stop HDFS daemons
$ stop-dfs.sh

YARN on a Single Node - I

 # Edit etc/hadoop/mapred-site.xml

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

 # Edit etc/hadoop/yarn-site.xml

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

YARN on a Single Node - II

 # Start ResourceManager and NodeManager daemons
$ start-yarn.sh

 # Browse the web interface for the ResourceManager
http://localhost:8088/

Run a MapReduce job

# Stop daemons
$ stop-yarn.sh