Hadoop

Reference: http://codingxiaxw.cn/2016/12/06/59-mac-hadoop/

1.Setup Homebrew and Cask ($ brew install brew-cask-completion)

https://www.jianshu.com/p/7d055bebab46

2. Setup JAVA (/usr/libexec/java_home -V or java -version)

https://blog.csdn.net/vvv_110/article/details/72897142

Setup environment variable

echo $JAVA_HOME to check

/Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home

Setup environment variable by $vim ~/.bash_profile, and effect it by $source .bash_profile

3. Setup ssh

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/alumi5566/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/alumi5566/.ssh/id_rsa.
Your public key has been saved in /Users/alumi5566/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:f3G/tp2TyvIR+VAAovP+8PYSYi738KVIhfBjKMZXN9o
alumi5566@ucrwpa-1-7-10-25-26-210.wnet.ucr.edu
The key's randomart image is:
+---[RSA 2048]----+
|          . ...   |
|          . .   . |
|          + . o   . |
|          .   B = . o  |
|          + oSB E.+.  |
|          . o o+o. o+. |
|          o=o oo .o|
|          ..oB=+ .++|
|          o.o*==o++|
+----[SHA256]-----+
```

If setup successful:

```
$ ssh localhost
Enter passphrase for key '/Users/alumi5566/.ssh/id_rsa':
Last login: Mon Jun  4 13:01:03 2018 from ::1
```

4. Setup Hadoop

```
$ brew install hadoop
```

Hadoop is located under /usr/local/Cellar/hadoop/3.1.0/ (may have different version number)

(a) under /usr/local/Cellar/hadoop/3.1.0/libexec/etc/hadoop/hadoop-env.sh, change

export HADOOP_OPTS="-Djava.net.preferIPv4Stack=true"

to

export HADOOP_OPTS="$HADOOP_OPTS -Djava.net.preferIPv4Stack=true -Djava.security.krb5.realm= -Djava.security.krb5.kdc="

export JAVA_HOME="/Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home"

(b) open /usr/local/Cellar/hadoop/3.1.0/libexec/etc/hadoop/core-site.xml

Add following property in <configuration> :

<property>

```
    <name>hadoop.tmp.dir</name>
    <value>/usr/local/Cellar/hadoop/hdfs/tmp</value>
    <description>A base for other temporary directories.</description>
</property>
<property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:8020</value>
</property>
```

(c) open /usr/local/Cellar/hadoop/3.1.0/libexec/etc/hadoop/mapred-site.xml

Add following property in <configuration> :

```
<property>
    <name>mapred.job.tracker</name>
    <value>localhost:8021</value>
</property>
```

(d) Setup the replicas number of hdfs, since we use the pseudo distributed mode, use 1

open /usr/local/Cellar/hadoop/3.1.0/libexec/etc/hadoop/hdfs-site.xml

Add following property in <configuration> :

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

(e) format and setup new HDFS and create new directory and initialize data

Located under /usr/local/Cellar/hadoop/3.1.0/libexec/etc/hadoop/

$hdfs namenode -format

5. Start Hadoop (scripts are under sbin/)

$./start-dfs.sh // start HDFS

$./stop-dfs.sh // stop HDFS

Error message when sudo ./start-dfs.sh

```
ucrwpa-1-7-10-25-26-210:sbin alumi5566$ sudo ./start-dfs.sh
Password:
Starting namenodes on [localhost]
ERROR: Attempting to operate on hdfs namenode as root
ERROR: but there is no HDFS_NAMENODE_USER defined. Aborting operation.
Starting datanodes
ERROR: Attempting to operate on hdfs datanode as root
ERROR: but there is no HDFS_DATANODE_USER defined. Aborting operation.
Starting secondary namenodes [ucrwpa-1-7-10-25-26-210.wnet.ucr.edu]
ERROR: Attempting to operate on hdfs secondarynamenode as root
ERROR: but there is no HDFS_SECONDARYNAMENODE_USER defined. Aborting oper
ation.
2018-06-04 15:34:04,148 WARN util.NativeCodeLoader: Unable to load native
-hadoop library for your platform... using builtin-java classes where app
licable
ucrwpa-1-7-10-25-26-210:sbin alumi5566$
```

Error message when ./start-dfs.sh

Starting namenodes on [localhost]

localhost: U@localhost: Permission denied (publickey,password,keyboard-interactive).

Starting datanodes

Solution :

Generate new keygen.

$ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

Register key gen:

$cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Use browser to chesk if we start successful

http://localhost:9870

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'localhost:8020' (active)

Started:	Mon Jun 04 15:54:29 -0700 2018
Version:	3.1.0, r16b70619a24cdcf5d3b0fcf4b58ca77238ccbe6d
Compiled:	Thu Mar 29 17:00:00 -0700 2018 by centos from branch-3.1.0
Cluster ID:	CID-bf28b6dc-9d46-4446-af21-8b414b702d10
Block Pool ID:	BP-1352234239-10.25.26.210-1528152397334

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 66.48 MB of 156 MB Heap Memory. Max Heap Memory is 2 GB.

Non Heap Memory used 56.2 MB of 60.14 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	112.8 GB
Configured Remote Capacity:	0 B

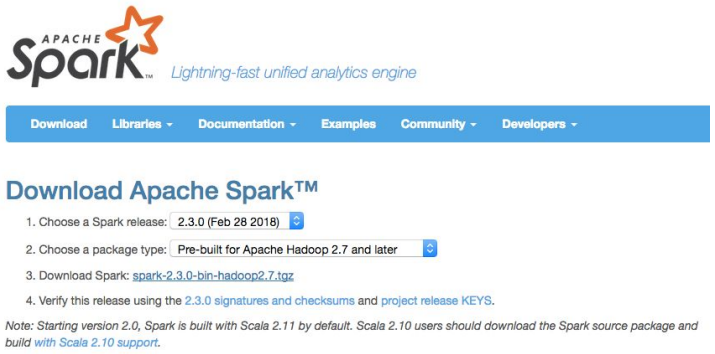
6. Start yarn (mapreduce framework) (script are under sbin/)
   $./start-yarn.sh // start yarn, a MapReduce framework
   $./stop-yarn.sh // stop yarn
7. Can also start all
   $./start-all.sh // start Hadoop
   $./stop-all.sh // stop Hadoop

Use mahout to implement k-means
$brew install mahout
The file is located under /usr/local/Cellar/mahout/0.13.0
Setup environment variable $vim ~/.bash_profile
export MAHOUT_HOME=//usr/local/Cellar/mahout/0.13.0/libexec
MAHOUT_CONF_DIR=$MAHOUT_HOME/
export PATH=$MAHOUT_HOME/bin:$PATH
$source ~/.bash_profile
And execute k-means
$time bin/hadoop jar /usr/local/Cellar/mahout/0.13.0/libexec/mahout-examples-0.13.0-job.jar org.apache.mahout.clustering.syntheticcontrol.kmeans.Job

Spark

1. Setup scala, the folder is under /usr/local/Cellar/scala/2.12.6
   $brew install scala
   Setup environment variable
   $sudo vim /etc/profile

export SCALA_HOME=/usr/local/Cellar/scala/2.12.6
export PATH=$PATH:$SCALA_HOME/bin
Effect it by $source /etc/profile
(test it by $scala)
2. Download apache-spark (choose the corresponding version)



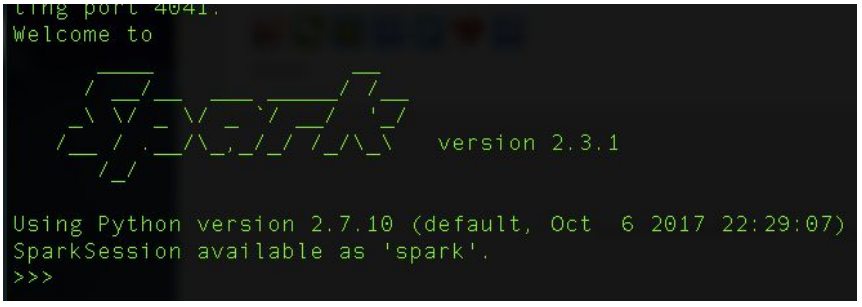
Extract and put it under /usr/local (change name to /spark)
Setup environment variable $sudo vim /etc/profile
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
3. Copy /usr/local/spark/conf/spark-env.sh.template into spark-env.sh (in the same folder)
Add following context in /usr/local/spark/conf/spark-env.sh
export SCALA_HOME=/usr/local/Cellar/scala/2.12.6
export SPARK_MASTER_IP=localhost
export SPARK_WORKER_MEMORY=4g
4. $spark-shell , lots of error message
Change to scala-2.11.12 (manually download to /usr/local/Cellar/scala)
Still lots of error message, change jdk
5. Change to jdk1.7
(http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html)
$vim ~/.bash_profile
JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.7.0_80.jdk/Contents/Home
6. Change Scala to scala-2.11.8
Download, extract, and locate to/usr/local/scala
$sudo vim /etc/profile
export SCALA_HOME=/usr/local/scala
Still error
7. Use this in the end
https://stackoverflow.com/questions/46436879/spark-shell-failed-to-initialize-compiler-error-on-a-mac
$ brew cask install java
$ brew install scala
$ brew install apache-spark
and $sudo spark-shell

(If you try to setup manually, Scala 2.11 + jdk 1.7 or 1.8 works, according to blog)

Too much trouble to use Scala, use pyspark instead

1. change SPARK_HOME=/usr/local/spark in /etc/profile to the brew version
(/usr/local/Cellar/apache-spark/2.3.1/bin)
Or just comment it, default using brew version

$sudo pyspark



Storm

Dependency: zookeeper and python

1. Download release version of apache-storm (http://storm.apache.org/downloads.html)
We download version 1.22 and locate under /usr/local/storm
Setup environment variable
$sudo vim /etc/profile
export STORM_HOME=/usr/local/storm
export PATH=$STORM_HOME/bin:$PATH
, and effect it by $source /etc/profile
2. Setup zookeeper (https://zookeeper.apache.org/releases.html#download)
We download version 3.4.10 and locate under /usr/local/zookeeper
Copy /usr/local/zookeeper/conf/zoo_sample.cfg to /usr/local/zookeeper/conf/zoo.cfg
Setup environment variable
$sudo vim /etc/profile
export ZOOKEEPER_HOME=/usr/local/zookeeper
export PATH=$PATH:$ZOOKEEPER_HOME/bin
, and effect it by $source /etc/profile
3. OSX has another dependency: Zeromq
$brew install zeromq
4. Start zookeeper,
$bin/zkServer.sh start
(use $bin/zkServer.sh status to check success or not)

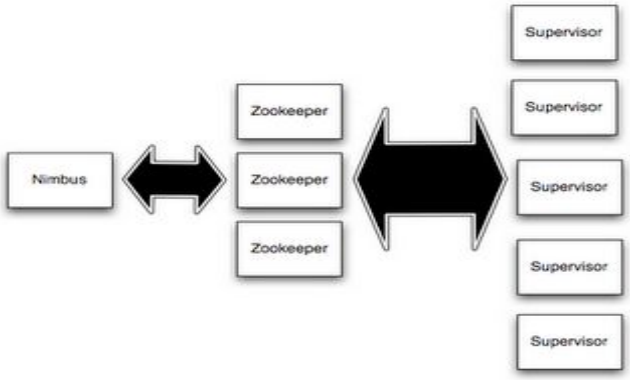
```
ucrwpa-1-7-10-25-27-11:storm-starter alumi5566$ zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Mode: standalone
```

5. Start supervisor of storm

$bin/storm nimbus & (master node)

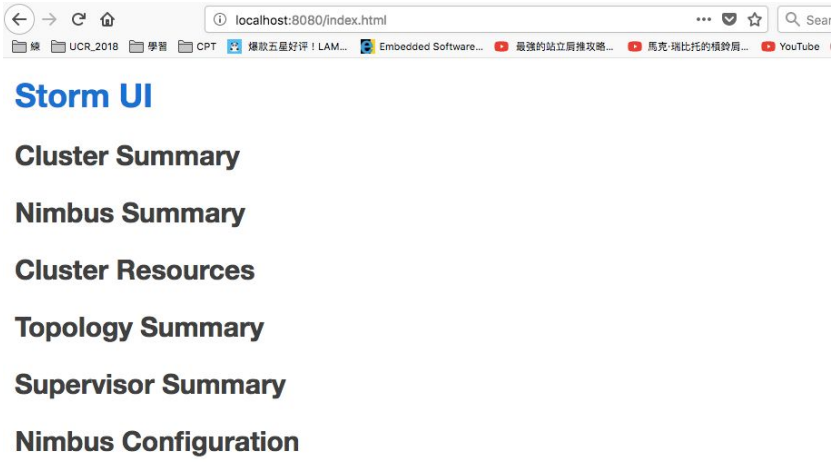
$bin/storm supervisor & (slave node)

$bin/storm ui &



Above is the relation between these three (spout and bolt are component of data flow)

6. Use http://localhost:8080/index.html to check



1. Use maven to build k-means project

$brew install maven

Use $mvn -version to check if setup success

```
ucrwpa-1-7-10-25-27-11:storm-starter alumi5566$ mvn -version
Apache Maven 3.5.3 (3383c37e1f9e9b3bc3df5050c29c8aff9f295297; 2018-02-24T1
Maven home: /usr/local/Cellar/maven/3.5.3/libexec
Java version: 10.0.1, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home
Default locale: zh_TW_#Hant, platform encoding: Big5_Solaris
OS name: "mac os x", version: "10.13.4", arch: "x86_64", family: "mac"
ucrwpa-1-7-10-25-27-11:storm-starter alumi5566$
```

2. Under /usr/local/storm/examples/storm-starter

$ mvn clean install -DskipTest

Fail when build:

Could not find artifact jdk.tools:jdk.tools:jar:1.7 at specified path mac

Somehow maven specify tools.jar of jdk.10, but this tools.jar is no longer existed after jdk.9

Manually assign in pom.xml

<dependency>
<groupId>jdk.tools</groupId>
<artifactId>jdk.tools</artifactId>
<version>1.7</version>
<scope>system</scope>
<systemPath>/Library/Java/JavaVirtualMachines/jdk1.7.0_80.jdk/Contents/Home/lib/tools.jar</systemPath>
</dependency>

```
[INFO] Installing /usr/local/Cellar/storm/1.2.2/libexec/examples/storm-starter/depe
n.xml to /var/root/.m2/repository/org/apache/storm/storm-starter/1.2.2/storm-starte
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 01:44 min
[INFO] Finished at: 2018-06-10T23:23:34-07:00
[INFO] ------------------------------------------------------------------------
ucrwpa-1-7-10-25-27-11:storm-starter alumi5566$
```

3. $sudo mvn compile exec:java -Dstorm.topology=storm.starter.WordCountTopology
Still fail, change jdk to 8 is ok

Design of Storm: data flow is composed of spout and bolt
Take word count as example, the core component is: one spout, two bolt, and one topology. The spout read in text file, and transfer to bolt. The first bolt receive and split tuple by tuple and generate word. Transfer this word to next bolt. The second bolt recerive the word and accumulate the count (in HashMap)

Useful Link

[1] Hadoop cmd
http://hadoopspark.blogspot.com/2015/09/6-hadoop-hdfs.html
[2] Hadoop IO performance
https://blog.csdn.net/bhq2010/article/details/8740154
[3] Spark Introduce
https://www.slideshare.net/imac-cloud/spark-61970801?next_slideshow=1
[4] Zookeeper
https://blog.csdn.net/liuxinghao/article/details/42747625
[5] Storm wordcount
https://blog.csdn.net/wuliusir/article/details/49910873
[6] word count and other code
https://github.com/storm-book
[7] IO performance
https://wr.informatik.uni-hamburg.de/_media/research/labs/2009/2009-12-tien_duc_dinh-evaluierung_von_hadoop-report.pdf