[1] SSH with pwd

linsong@linsongdeMacBook-Air ~ % **ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa**

Generating public/private rsa key pair.

Created directory '/Users/linsong/.ssh'.

Your identification has been saved in /Users/linsong/.ssh/id\_rsa.

Your public key has been saved in /Users/linsong/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:cQzN6EvZMCSq8FU7Fal3AOoVQ76XN+3H4BwpMPiP7B0 linsong@linsongdeMacBook-Air.local

The key's randomart image is:

+---[RSA 3072]----+

| .Bo=\* |

| = B\*oo |

|. + \*o+\*o |

| o + ..+=\*o. . |

| o . .oSo+ = |

| o.+ \* + |

| o E + o |

| . . . . |

| . . |

+----[SHA256]-----+

linsong@linsongdeMacBook-Air ~ % **cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys**

linsong@linsongdeMacBook-Air ~ % **chmod 0600 ~/.ssh/authorized\_keys**

(base) linsong@linsongdeMacBook-Air /etc % **ssh localhost**

Last login: Tue Jan 26 09:48:59 2021

[2] Install Hadoop

linsong@linsongdeMacBook-Air ~ % **brew install hadoop**

linsong@linsongdeMacBook-Air ~ % **ls /usr/local/Cellar/hadoop**

3.3.0

[3] Java (Java SE Development Kit 8)

linsong@linsongdeMacBook-Air hadoop % **java -version**

java version "15.0.2" 2021-01-19

Java(TM) SE Runtime Environment (build 15.0.2+7-27)

Java HotSpot(TM) 64-Bit Server VM (build 15.0.2+7-27, mixed mode, sharing)

linsong@linsongdeMacBook-Air hadoop % **cd /Library/Java/JavaVirtualMachines**

linsong@linsongdeMacBook-Air JavaVirtualMachines % **ls -la**

total 0

drwxr-xr-x 6 root wheel 192 Jan 26 09:07 .

drwxr-xr-x 4 root wheel 128 Oct 24 2019 ..

drwxr-xr-x 3 root wheel 96 Apr 24 2020 adoptopenjdk-8.jdk

drwxr-xr-x 3 root wheel 96 Jan 25 21:15 jdk-15.0.2.jdk

drwxr-xr-x 3 root wheel 96 Jan 26 09:07 jdk1.8.0\_281.jdk

drwxr-xr-x@ 3 root wheel 96 Mar 5 2020 openjdk-14.0.1.jdk

linsong@linsongdeMacBook-Air JavaVirtualMachines % **/usr/libexec/java\_home -V**

Matching Java Virtual Machines (4):

15.0.2, x86\_64: "Java SE 15.0.2" /Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home

14.0.1, x86\_64: "OpenJDK 14.0.1" /Library/Java/JavaVirtualMachines/openjdk-14.0.1.jdk/Contents/Home

1.8.0\_281, x86\_64: "Java SE 8" /Library/Java/JavaVirtualMachines/jdk1.8.0\_281.jdk/Contents/Home

1.8.0\_252, x86\_64: "AdoptOpenJDK 8" /Library/Java/JavaVirtualMachines/adoptopenjdk-8.jdk/Contents/Home

/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home

Open **~/.bash\_profile**, and add:

export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0\_281.jdk/Contents/Home

export PATH=$JAVA\_HOME/bin:$PATH

linsong@linsongdeMacBook-Air JavaVirtualMachines % **echo $JAVA\_HOME**

linsong@linsongdeMacBook-Air JavaVirtualMachines % **source ~/.bash\_profile**

(base) linsong@linsongdeMacBook-Air JavaVirtualMachines % **echo $JAVA\_HOME**

/Library/Java/JavaVirtualMachines/jdk1.8.0\_281.jdk/Contents/Home

[4] Configuration

linsong@linsongdeMacBook-Air ~ % **cd /usr/local/Cellar/hadoop/3.3.0/libexec/etc/Hadoop**

(base) linsong@linsongdeMacBook-Air hadoop % **/usr/libexec/java\_home**

/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home

**hadoop-env.sh**

export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0\_281.jdk/Contents/Home

**core-site.xml**

<configuration>

<!--namenode address-->

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

**hdfs-site.xml**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>dfs.http.address</name>

<value>localhost:9870</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value> <!--turn off firewall-->

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>/Users/linsong/Documents/2021/BigData/hadoop/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/Users/linsong/Documents/2021/BigData/hadoop/dfs/data1/hadoop</value>

</property>

</configuration>

**mapred-site.xml**

<configuration>

<!--hadoop MR(Map/Reduce) over YAR-->

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

**yarn-site.xml**

<configuration>

<!--nomenodeManager shuffle-->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<!--NodeManagers env variables-->

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>

</property>

</configuration>

[5] Format

linsong@linsongdeMacBook-Air ~ % **cd /usr/local/Cellar/hadoop/3.3.0/**

**Mac: 选择系统偏好设置->选择共享->点击远程登录**

linsong@linsongdeMacBook-Air 3.3.0 % **bin/hdfs namenode -format**

linsong@linsongdeMacBook-Air 3.3.0 % **sbin/start-yarn.sh**

Starting resourcemanager

resourcemanager is running as process 38270. Stop it first.

Starting nodemanagers

localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.

Open: <http://localhost:8088/cluster>

[6] Install Hive

linsong@linsongdeMacBook-Air 3.3.0 % **brew install hive**

linsong@linsongdeMacBook-Air 3.3.0 % **cd /usr/local/Cellar/hadoop/3.3.0/**

linsong@linsongdeMacBook-Air 3.3.0 % **sbin/start-all.sh**

linsong@linsongdeMacBook-Air 3.3.0 % **jps**

40160 SecondaryNameNode

38611 NodeManager

40029 DataNode

40510 Jps

38270 ResourceManager

[7] MySql

Root: 12345678

Open **~/.bash\_profile**, and add:

export PATH=${PATH}:/usr/local/mysql/bin

linsong@linsongdeMacBook-Air JavaVirtualMachines % **source ~/.bash\_profile**

(base) linsong@linsongdeMacBook-Air bin % **mysql -u root -p**

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

mysql> **create database metastore;**

Query OK, 1 row affected (0.26 sec)

Add **hive\_site.xml** to /usr/local/Cellar/hive/3.1.2\_2/libexec/conf:

|  |
| --- |
| <configuration>  　　<property>  <name>hive.metastore.local</name>  <value>true</value>  </property>  <property>  <name>javax.jdo.option.ConnectionURL</name>  <value>jdbc:mysql://localhost/metastore</value>  </property>  <property>  <name>javax.jdo.option.ConnectionDriverName</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <property>  <name>javax.jdo.option.ConnectionUserName</name>  <value>root</value>  </property>  　　<property>  <name>javax.jdo.option.ConnectionPassword</name>  <value>12345678</value>  </property>  <property>  <name>hive.exec.local.scratchdir</name>  <value>/tmp/hive</value>  </property>  <property>  <name>hive.querylog.location</name>  <value>/tmp/hive</value>  </property>  <property>  <name>hive.downloaded.resources.dir</name>  <value>/tmp/hive</value>  </property>  <property>  <name>hive.server2.logging.operation.log.location</name>  <value>/tmp/hive</value>  </property>  </configuration> |

Download Connector/J 8.0.23 with platform independent, place **mysql-connector-java-8.0.23.jar** under **/usr/local/Cellar/hive/3.1.2\_2/libexec/lib**

linsong@linsongdeMacBook-Air 3.3.0 % **schematool -initSchema -dbType mysql**

mysql> show databases;

mysql> show tables;

[8] Verification

(base) linsong@linsongdeMacBook-Air local % **hadoop version**

Hadoop 3.3.0

Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r aa96f1871bfd858f9bac59cf2a81ec470da649af

Compiled by brahma on 2020-07-06T18:44Z

Compiled with protoc 3.7.1

From source with checksum 5dc29b802d6ccd77b262ef9d04d19c4

This command was run using /usr/local/Cellar/hadoop/3.3.0/libexec/share/hadoop/common/hadoop-common-3.3.0.jar

<http://localhost:9870/>

<http://localhost:8088/cluster>

(base) linsong@linsongdeMacBook-Air 3.3.0 % **hive**

hive> **show databases;**

OK

default

Time taken: 6.103 seconds, Fetched: 1 row(s)

hive> **show tables;**

OK

Time taken: 0.438 seconds

linsong@linsongdeMacBook-Air 3.3.0 % **hadoop fs -ls /**

2021-01-26 12:00:19,026 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 2 items

drwx-wx-wx - linsong supergroup 0 2021-01-26 11:47 /tmp

drwxr-xr-x - linsong supergroup 0 2021-01-26 11:47 /user

linsong@linsongdeMacBook-Air **bin % hadoop jar /usr/local/Cellar/hadoop/3.3.0/libexec/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar pi 10 10**

Job Finished in 92.717 seconds

Estimated value of Pi is 3.20000000000000000000

[9] Spark

Add below to ~/.bash\_profile:

CLASSPATH=$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/dt.jar:.

export CLASSPATH

(base) linsong@linsongdeMacBook-Air ~ % **source ~/.bash\_profile**

(base) linsong@linsongdeMacBook-Air ~ % **java -version**

java version "1.8.0\_281"

Java(TM) SE Runtime Environment (build 1.8.0\_281-b09)

Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)

(base) linsong@linsongdeMacBook-Air ~ % **python3**

Python 3.8.5 (default, Sep 4 2020, 02:22:02)

[Clang 10.0.0 ] :: Anaconda, Inc. on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>

(base) linsong@linsongdeMacBook-Air ~ % **Scala -version**

Scala code runner version 2.13.2 -- Copyright 2002-2020, LAMP/EPFL and Lightbend, Inc.

Add below to ~/.bash\_profile:

export SCALA\_HOME=/usr/local/Cellar/scala/2.13.2

export PATH=$PATH:$SCALA\_HOME/bin

(base) linsong@linsongdeMacBook-Air 2.13.2 % **source ~/.bash\_profile**

linsong@linsongdeMacBook-Air ~ % **scala**

Welcome to Scala 2.13.2 (OpenJDK 64-Bit Server VM, Java 15.0.1).

Type in expressions for evaluation. Or try :help.

scala> **util.Properties.versionString**

val **res0**: **String** = version 2.13.2

scala> **:q**

linsong@linsongdeMacBook-Air ~ %

Download spark-3.0.1-bin-hadoop3.2.tgz to /usr/local and rename to spark3.0.1, and add below to

export SPARK\_HOME=/usr/local/spark3.0.1

export PATH=$PATH:$SPARK\_HOME/bin

(base) linsong@linsongdeMacBook-Air 2.13.2 % **source ~/.bash\_profile**

(base) linsong@linsongdeMacBook-Air Cellar % **cd /usr/local/spark3.0.1/conf/**

(base) linsong@linsongdeMacBook-Air conf % **cp spark-env.sh.template spark-env.sh**

Add below to **spark-env.sh**:

export SCALA\_HOME=/usr/local/Cellar/scala

export SPARK\_MASTER\_IP=localhost

export SPARK\_WORKER\_MEMORY=4g

(base) linsong@linsongdeMacBook-Air spark3.0.1 % **spark-shell**

2021-01-26 20:45:29,540 WARN util.Utils: Your hostname, linsongdeMacBook-Air.local resolves to a loopback address: 127.0.0.1; using 192.168.18.3 instead (on interface en0)

2021-01-26 20:45:29,552 WARN util.Utils: Set SPARK\_LOCAL\_IP if you need to bind to another address

2021-01-26 20:45:32,403 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

Spark context Web UI available at http://192.168.18.3:4040

Spark context available as 'sc' (master = local[\*], app id = local-1611665154817).

Spark session available as 'spark'.

Welcome to

\_\_\_\_ \_\_

/ \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_

\_\ \/ \_ \/ \_ `/ \_\_/ '\_/

/\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 3.0.1

/\_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0\_281)

Type in expressions to have them evaluated.

Type :help for more information.

scala> import org.apache.spark.{SparkConf, SparkContext}

import org.apache.spark.{SparkConf, SparkContext}

scala> SparkContext

**res0**: **org.apache.spark.SparkContext.type** = org.apache.spark.SparkContext$@1780bc15

linsong@linsongdeMacBook-Air spark3.0.1 % **pwd**

/usr/local/spark3.0.1

linsong@linsongdeMacBook-Air spark3.0.1 % **sbin/start-all.sh**

starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark3.0.1/logs/spark-linsong-org.apache.spark.deploy.master.Master-1-linsongdeMacBook-Air.local.out

localhost: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark3.0.1/logs/spark-linsong-org.apache.spark.deploy.worker.Worker-1-linsongdeMacBook-Air.local.out

And then:

(base) linsong@linsongdeMacBook-Air spark3.0.1 % **spark-shell**

scala> **val textFile = sc.textFile("file:///usr/local/spark3.0.1/README.md")**

**textFile**: **org.apache.spark.rdd.RDD[String]** = file:///usr/local/spark3.0.1/README.md MapPartitionsRDD[1] at textFile at <console>:24

scala> **textFile.count()**

**res0**: **Long** = 108

scala> **textFile.first()**

**res1**: **String** = # Apache Spark

scala> **val linesWithSpark = textFile.filter(line => line.contains("Spark"))**

**linesWithSpark**: **org.apache.spark.rdd.RDD[String]** = MapPartitionsRDD[2] at filter at <console>:25

scala> **linesWithSpark.count()**

**res2**: **Long** = 19

scala> **textFile.map(line => line.split(" ").size).reduce((a, b) => if(a>b) a else b)**

**res3**: **Int** = 16

scala> **import java.lang.Math**

import java.lang.Math

scala> **textFile.map(line => line.split(" ").size).reduce((a, b) => Math.max(a, b))**

**res4**: **Int** = 16

scala> **val wordCounts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey((a,b) => a+ b)**

**wordCounts**: **org.apache.spark.rdd.RDD[(String, Int)]** = ShuffledRDD[7] at reduceByKey at <console>:26

scala> **wordCounts.collect()**

**res5**: **Array[(String, Int)]** = Array((package,1), (this,1), (integration,1), (Python,2), (cluster.,1), (its,1), ([run,1), (There,1), (general,2), (have,1), (pre-built,1), (Because,1), (YARN,,1), (locally,2), (changed,1), (locally.,1), (several,1), (only,1), (Configuration,1), (This,2), (basic,1), (first,1), (learning,,1), (documentation,3), (graph,1), (Hive,2), (info,1), (["Specifying,1), ("yarn",1), ([params]`.,1), ([project,1), (prefer,1), (SparkPi,2), (engine,2), (version,1), (file,1), (documentation,,1), (MASTER,1), (example,3), (are,1), (systems.,1), (params,1), (scala>,1), (DataFrames,,1), (provides,1), (refer,2), (configure,1), (Interactive,2), (R,,1), (can,6), (build,3), (when,1), (easiest,1), (Maven](https://maven.apache.org/).,1), (Apache,1), (guide](ht...

[10] Spark from Hive

Below looks good, supporting hive from local:

scala> **import org.apache.spark.sql.hive.HiveContext**

import org.apache.spark.sql.hive.HiveContext