Environmental configuration of Spark

## Prerequisites

Ubuntu16.04 and later

Hadoop 2.7.1 and later ([Instruction](#_Install_hadoop))

Java JDK 1.7 and later ([Instruction](#_(4)_Install_Java))

Spark 2.1.0([Instruction](#_Install_spark))

## The creation of new user in Ubuntu system

In this tutorial, we set ‘hadoop’ as user name to login the ubantu system. The user name is set as follows:

*$ sudo useradd –m hadoop –s /bin/bash # create a new user name:’hadoop’*

*$ sudo passwd hadoop # setup password as ‘hadoop’*

*$ sudo adduser hadoop sudo # modify permission*

And then select the newly created ‘hadoop’ user to login Ubuntu system.

## Install hadoop

If you have installed hadoop on your server, this step can be skipped. Otherwise you should install hadoop before install spark.

### 1. Install pre-required software

#### (1) Update apt

*$ sudo apt-get update*

#### (2) Update vim

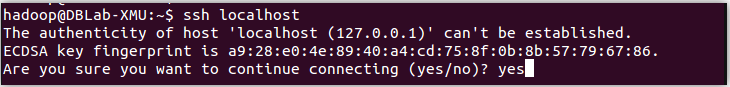
*$ sudo apt-get install vim*

#### (3) Install SSH and configure SSH to login without password

Ubuntu have installed SSH client by default, and you need to install SSH server as following command:

*$ sudo apt-get install openssh-server*

You can use *$ ssh localhost* to login the localhost, select ‘yes’ when the following prompts appear, and enter the password ‘hadoop’

**

Then exit

However, the password is required for login every time, so we can configure SSH to login without password.

*$ exit # exit ssh localhost*

*$ cd ~/.ssh/ # if can’t find this dictionary, please use $ ssh localhost first*

*$ ssh-keygen –t rsa # press ‘Endter’ when prompt messages appeared*

*$ cat ./id\_rsa.pub >> ./authorized\_keys*

Now you can use command ‘*ssh localhost* ’ without password.

#### (4) Install Java Runtime Environment

You can choose JDK of Oracle or openJDK for Java runtime environment, in this tutorial we use openJDK7. If it failed, please try openJDK8 (replace all JDK7 in following commands with JDK8)

1. Install

*$ sudo apt-get install openjdk-7-jre openjdk-7-jdk*

1. Configure environmental variables

After you have installed the OpenJDK, you need to find the corresponding installation path, which is used to configure the JAVA\_HOME environment variable.

You can find it with following command:

*$ dpkg -L openjdk-7-jdk | grep '/bin/javac'*

The command outputs a path for configuration of JAVA\_HOME environment variable which need to remove the "/ bin/javac" at the end of the path, such as the result is /usr/lib/jvm/java-7-openjdk-amd64/bin/javac, the path can be set as /usr/lib/jvm/java-7-openjdk-amd64.

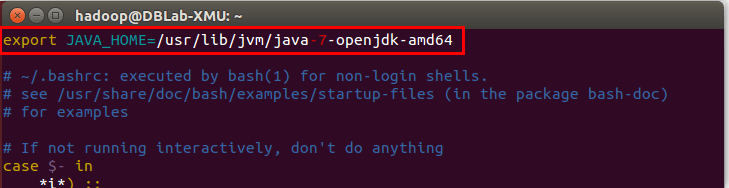
And then open ~/.bashrc with vim

*$ vim ~/.bashrc*

Add an individual line at the top of the file as following:

*$ export JAVA\_HOME=\* # \* is the path of openJDK*

For example, shown in the red square,the /usr/lib/java-7-openjdk-amd64 is the path of openJDK in this tutorial.



Save the file and use the following command to make environment variables effective.

*$ source ~/.bashrc*

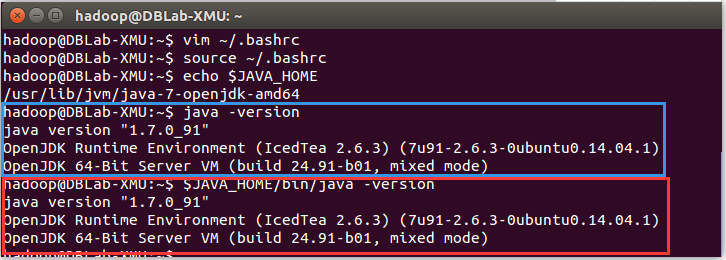
You can check it with the following commands:

*$ echo $JAVA\_HOME*

*$ java -version*

*$ JAVA\_HOME/bin/java –version*

If it works, the message shown in the blue square and red square are the same.



Till now, Java running environment required for Hadoop has been is installed.

### 2. Install hadoop 2

You can download hadoop 2 at <http://mirror.bit.edu.cn/apache/hadoop/common/> or <http://mirrors.cnnic.cn/apache/hadoop/common/> .To get a stable version, you’d better choose hadoop-2.x.y.tar.gz under the directory ‘stable’.

In this tutorial, we install Hadoop to /usr/local/

*$ sudo tar -zxf ~/ Downloads/hadoop-2.6.0.tar.gz -C /usr/local*

*$ cd /usr/local/*

*$ sudo mv ./hadoop-2.6.0/ ./hadoop #* *Rename the folder to 'hadoop'*

*$ sudo chown -R hadoop ./hadoop #* *modify permission*

The Hadoop can be used when it is decompressed. Enter the following command to check if Hadoop is available, and if success it would show the Hadoop version information:

*$ cd /usr/local/hadoop*

*$ ./bin/hadoop version*

### 3.Hadoop stand-alone configuration (non-distributed)（if you want to configuration distributed, you can click [here](http://data-flair.training/blogs/install-hadoop-2-x-ubuntu-hadoop-multi-node-cluster/)）

The Hadoop default mode is non-distributed mode (local mode) and runs without additional configuration.

Here we use a example ‘grep’ to run hadoop, We use all the files in the input folder as input, and select the words that match the regular expression ‘DFS [a-z.]+’ and count the number of occurrences, and the final results output to the output folder.

*$ cd /usr/local/hadoop*

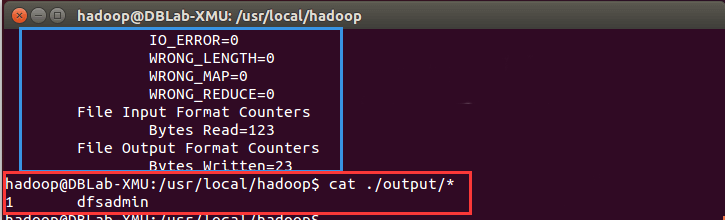
*$ mkdir ./input*

*$ cp ./etc/hadoop/\*.xml ./input*

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

*$ cat ./output/\**

If it works, it would outputs the following messages:

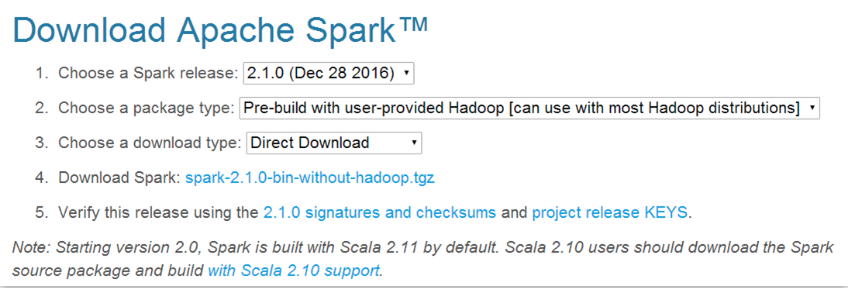


The blue square is the output message and the red square is the result.

Till now, the hadoop is ready to use.

## Install spark

Spark is available at <http://spark.apache.org/downloads.html> as following picture:



Click ‘spark-2.1.0-bin-without-hadoop.tgz’ and the package would be stored in /home/hadoop/Downloads.

Spark deployment pattern mainly has four kinds: Local mode (stand-alone mode), Standalone mode (using Spark’s own simple cluster manager), the YARN model (using the YARN as the cluster manager) and Mesos mode (using the Mesos as cluster manager)

We install spark with local model with Spark 2.1.0，and login with the username ‘hadoop’

*$ sudo tar -zxf ~/Downloads/spark-2.1.0-bin-without-hadoop.tgz -C /usr/local/*

*$ cd /usr/local*

*$ sudo mv ./spark-2.1.0-bin-without-hadoop/ ./spark*

*$ sudo chown -R hadoop:hadoop ./spark*

After installation, Spark's configuration file ‘Spark -env. Sh’ is modified as follows:

*$ cd /usr/local/spark*

*$ bin/run-example SparkPi*

Open ‘spark-env.sh’ with vim (vim ./conf/spark-env.sh), add the following messages at the top of this file

export SPARK\_DIST\_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)

We can run an example to check if it works.

*$ cd /usr/local/spark*

*$ bin/run-example SparkPi*

It will be a great amount of output information, if it is hard to find the result ‘Pi is roundly 3.14588’, you can use linux command ‘grep’ command to find the messages you need:



If you can get the following results, that means the Spark is working successfully.



## Reference

<http://dblab.xmu.edu.cn/blog/1307-2/>

<http://spark.apache.org/downloads.html>