Installation Guide for Hadoop Cluster with Docker on Cloudlab

Shangyu Xie

October 28, 2016

1 Prepration

1.1 Background

As background, you should be familiar with Cloudlab, including image, profile, clone, and snapshot, etc..

a. think about the proper operating system for your experiment. You need consider for the future. For example, currently Ubuntu 15.04 is not supported officially, while you may choose Ubuntu 16.04 or some other versions.

b. select the base system profile for your own.

c. initiate it, and clone it as your own image. Note this is very important, because the system image cannot be snapshot(modified) by you; instead you can modify it after cloning it to your own image profile.

1.2 Note

a. When you log into an instance on Cloudlab, it will create user as your own, and you can give the user authority. And also in home directory, there exists /local directory, which could be saved into disk image when you snapshot.

b. Note that you would better do your software installation and test at the /local directory, and do not modify the system directory.

2 Configuration

When you finish the software and package install, you need to smooth the test by adding some configuration files, for example, boot config, path variable. For the test, we need to configure the hadoop home path, and another essential file for booting up environment.

2.1 Generate SSH Key

For the ssh connection. Since Cloudlab would not save your file in the user path, instead in /local file, so we would better save the all the modifications in /local file, including hadoop installation. And generate ssh-key generate command, and save the key pair in the /local file.

2.2 Pre-run

This is created for ssh key to transfer to the user authorized keys, and you shall generate the key at first and saved it in the /local file. The file is .sh file. Note that you should verify the command to see if it works via ssh localhost; if not, just type the command line in the bash. Another is in /etc/rc.local, to ensure execute the .sh file when system boots.

2.3 Environement Variable

For Java and Hadoop home path. Config it by edit a new file under /etc/profile.d /hadoop.sh, which would read it when system starts.

3 Install Hadoop

Consists of installing Java, install and configure Hadoop. Note that before installation, you should make sure all the pre-work done, ssh localhost, ensure the authentication of your own user.

3.1 Install Java

Make sure that sudo apt-get update before every installation. The operating system is Ubuntu 16.04. And install the JDK AND JRE 8 via apt-get install. Then modifiy the Java Home Path in /etc/profile.d /hadoop.sh. And verify by java -version.

3.2 Install Hadoop

Choose the stable release version, Hadoop-2.7.1. Copy the source code .tar.gz and .mds of Hadoop link, use wget to download them in /local. The md5sum command is used for checking the completeness of Hadoop file.

After configuring the completeness, extract the Hadoop file and rename it ”hadoop” for convenience. Note that all the commands above need sudo. Add the Hadoop path into /etc/profile.d/hadoop.sh. Finally, you can verity the hadoop, by hadoop version to see if environment variables work.

3.3 Configure Hadoop for Cluster

There are six configuration files to set up clusters, under /hadoop/etc/hadoop/. And Hadoop Prefix and Home are configured properly in /etc/profile.d.

a. slaves. It determines the hostname of slaves(or DataNode). For convenience, you can increase to 10 slaves, so that you can use the base image to set up the maxmum 10 slaves cluster.
slave1
slave2
...
slave 10

b. hadoop-env.sh. Do site-specific customization of Hadoop daemons' process. NOTE it is essential to set JAVA HOME in this file to make cluster up. Other daemons are optional,for example, namenode or datanode.

c. core-site.xml

fs.defaultFS	NameNode URI
io.file.buffer.size	131072

d. hdfs-site.xml Configure NameNode & DataNode. Note the dfs.namenode.name.dir and dfs.datanode.name.dir are path on the local system to store the namespace and transactions log persistently. Here put them in /dfs/name and /dfs/data.

e. yarn-site.xml Configure ResourceManager & NodeManager.

f. mapred-site.xml Configure MapReduce Applications & JobHistory Server. Note there are two important parameters.

mapreduce.jobhistory.intermediate-done-dir	/mr-history/tmp
mapreduce.jobhistory.done-dir	/mr-history/done

More details about configurations, please refer to the offical website.

4 Start Hadoop Cluster

Note that ensure the ssh nodes between workers, and /etc/hosts configure correctly.

a. Operating Command

   (a) Format HDFS. Very Important, do not forget.
   $ hdfs namenode -format

   (b) Start HDFS.
   $ start-dfs.sh, since you configure the path before.

    (c) Start YARN.

    ```
    $ start-yarn.sh.
    ```

    (d) Start MapReduce history server

    ```
    $ mr-jobhistory-daemon.sh start historyserver
    ```

b. Verification Run jps on master and slave node respectively, you will see the processes to see if cluster works.

    (a) Master
    NameNode, ResourceManager, SecondrryNameNode, JobHistoryServer

    (b) Slave
    DataNode, NodeManager.

5 Test Experiment

You can choose the exmaple to test the function of cluster, for example, the grep or tera.

6 Docker

Note that you should carefully select your system architecture, because Docker is not supported for every system architecture. Run

```
$ uname -a
```

Since the operating system is Ubuntu 16.04, amd64. So just follow the Docker website guide.
Then test it by run

```
$ sudo docker run helloworld
```