

来自 :<https://www.aboutyun.com/forum.php?mod=viewthread&tid=20620>

一、Linux环境准备

1. 设置静态ip

2. 关闭SELINUX

3. 关闭防火墙

4. 开启ssh

5. 修改hosts

6. 修改主机名

7. 配置ntp服务

8. 克隆节点

9. ssh免密码登录

二、安装Java

三、安装scala

四、安装hadoop

1. 解压安装包

2. 配置hadoop

接上篇: [about云日志分析项目准备5: hadoop,spark,kafka,flume,hive等工具的版本选择](#)

这节开始讲解集群搭建:

这儿选用的linux环境是CentOS-7.0-1406-x86_64-GnomeLive.iso GNOME桌面版。安装虚拟机的过程就不说了, 这儿使用的网络模式是NAT模式。目前已aboutyun用户登录master机器。本次我们要搭建的是一个三节点的Hadoop、Spark集群。

一、Linux环境准备

1. 设置静态ip

2. 关闭SELINUX

修改 /etc/sysconfig/selinux文件

`vim /etc/sysconfig/selinux`

[复制代码](#)

3. 关闭防火墙

`sudo systemctl stop firewalld.service` #停止firewall

`sudo systemctl disable firewalld.service` #禁止firewall开机启动

4. 开启ssh

`sudo systemctl start sshd.service` #开启ssh

`sudo systemctl enable sshd.service` #开机启动ssh

5. 修改hosts

`sudo vim /etc/hosts`

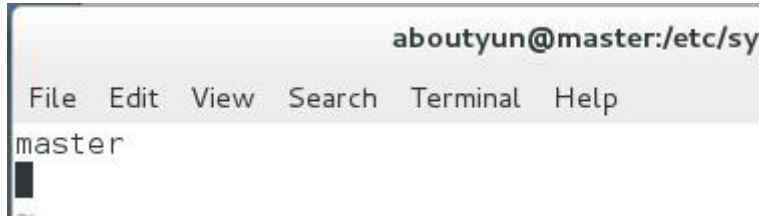
以下内容加入到hosts文件中：

```
File Edit View Search Terminal Help
127.0.0.1          localhost.localdomain localhost
::1               localhost6.localdomain6 localhost6
192.168.1.10      master
192.168.1.20      slave1
192.168.1.30      slave2
```

6. 修改主机名

```
sudo vim /etc/hostname
```

将文件内容改为master



7. 配置ntp服务

用于同步时间

```
sudo vim /etc/ntp.conf
```

设置服务器为以下几个（默认为以下服务器的不用修改）：

```
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

保存后执行：

```
sudo systemctl start ntpd.service #开启ntp服务
sudo systemctl enable ntpd.service # 开机运行ntp服务
```

8. 克隆节点

9. ssh免密码登录

需要实现在master ssh无密码登录本机、slave1和slave2。在master机器上，执行

```
ssh-keygen -t rsa
```

然后一直回车，这样就生成了aboutyun用户在master上的公钥和秘钥。

```
[aboutyun@master ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aboutyun/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aboutyun/.ssh/id_rsa.
Your public key has been saved in /home/aboutyun/.ssh/id_rsa.pub.
The key fingerprint is:
30:3a:63:31:cc:b1:2e:ce:e1:48:6a:88:88:54:3b:3c aboutyun@master
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .                |
|     o o               |
|    . * o              |
|   ..o+ o             |
|  ooo*E S             |
|0= +.o.               |
|B.+                   |
| .                    |
+-----+
[aboutyun@master ~]$
```

公钥位置

执行 以下命令,将公钥提供给master

`ssh-copy-id -i ~/.ssh/id_rsa.pub aboutyun@master`

```
[aboutyun@master ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub aboutyun@master
The authenticity of host 'master (192.168.1.10)' can't be established.
ECDSA key fingerprint is 34:fe:4e:35:9f:c6:cb:46:f8:db:35:2a:ae:dc:c2:92.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed - if you are prompt
ed now it is to install the new keys
aboutyun@master's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'aboutyun@master'"
and check to make sure that only the key(s) you wanted were added.
[aboutyun@master ~]$
```

输入yes

输入master机器上aboutyun用户的登陆密码

这样就实现了master使用ssh无密码登录本机。在依次执行以下命令：

`ssh-copy-id -i ~/.ssh/id_rsa.pub aboutyun@slave1`

`ssh-copy-id -i ~/.ssh/id_rsa.pub aboutyun@slave2`

这样就实现了在master上ssh无密码登录slave1和slave2。

二、安装Java

1. 解压安装包

```
sudo mkdir /data  
sudo chmod -R 777 /data  
tar -zxvf ~/jar/jdk-8u111-linux-x64.tar.gz -C /data
```

2. 设置环境变量

将以下内容加入到~/.bashrc文件中,

```
任意哪个配置文件都可以  
export JAVA_HOME=/data/jdk1.8.0_111  
export PATH=$JAVA_HOME/bin:$PATH  
export CLASS_PATH=$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar.
```

然后执行下面的命令:

```
source ~/.bashrc
```

三、安装scala

1. 解压安装包

```
tar -zxvf ~/jar/scala-2.11.8.tgz -C /data
```

2. 设置环境变量

将以下内容加入到~/.bashrc文件中,

```
export SCALA_HOME=/data/scala-2.11.8  
export PATH=$SCALA_HOME/bin:$PATH
```

然后执行下面的命令:

```
source ~/.bashrc
```

四、安装hadoop

1. 解压安装包

```
tar -zxvf ~/jar/hadoop-2.6.5.tar.gz -C /data
```

2. 配置hadoop

涉及到的配置文件为以下几个：

```
${HADOOP_HOME}/etc/hadoop/hadoop-env.sh  
${HADOOP_HOME}/etc/hadoop/yarn-env.sh  
${HADOOP_HOME}/etc/hadoop/slaves  
${HADOOP_HOME}/etc/hadoop/core-site.xml  
${HADOOP_HOME}/etc/hadoop/hdfs-site.xml  
${HADOOP_HOME}/etc/hadoop/mapred-site.xml  
${HADOOP_HOME}/etc/hadoop/yarn-site.xml
```

如果有的文件不存在，可以复制相应的template文件获得，例如，mapred-site.xml文件不存在，则可以从mapred-site.xml.template复制一份过来。

配置文件1：hadoop-env.sh

指定JAVA_HOME,修改如下

```
export JAVA_HOME=/data/jdk1.8.0_111
```

配置文件2：yarn-env.sh

指定JAVA_HOME,增加如下

```
export JAVA_HOME=/data/jdk1.8.0_111
```

配置文件3：slaves

将所有的从节点加入

```
aboutyun@master:/data/hadoop-2.6.5/etc/hadoop  
slave1  
slave2
```

配置文件4: core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:8020</value>
  </property>

  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:///home/aboutyun/hadoop/tmp</value>
    <description>Abase for other temporary directories.</description>
  </property>

  <property>
    <name>hadoop.proxyuser.aboutyun.hosts</name>
    <value>*</value>
    <description>abouyun用户可以代理任意机器上的用户</description>
  </property>

  <property>
    <name>hadoop.proxyuser.aboutyun.groups</name>
    <value>*</value>
    <description>abouyun用户代理任何组下的用户</description>
  </property>

  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
</configuration>
```

注意: 需要在本地创建/home/aboutyun/hadoop/tmp目录

配置文件5: hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>master:9001</value>
  </property>

  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/aboutyun/hadoop/namenode</value>
  </property>
```

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///home/aboutyun/hadoop/datanode</value>
</property>

<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>

<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>
</configuration>
<property>
```

注意: 需要在本地创建/home/aboutyun/hadoop/namenode和/home/aboutyun/hadoop/datanode目录

配置文件6: mapred-site.xml

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

  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:10020</value>
  </property>

  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>master:19888</value>
  </property>
</configuration>
```

配置文件7: yarn-site.xml

```
<configuration>
  <property>
```



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

<property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

<property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8032</value>
</property>

<property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:8030</value>
</property>

<property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>master:8031</value>
</property>

<property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>master:8033</value>
</property>

<property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>master:8088</value>
</property>
</configuration>
```

3. 设置环境变量

将以下内容加入到~/.bashrc文件中

```
export HADOOP_HOME=/data/hadoop-2.6.5
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

然后执行以下命令：

```
source ~/.bashrc
```

4. 复制到其他节点

1) 复制安装目录

在master机器上：

```
scp -r /data/hadoop-2.6.5/ /data/scala-2.11.8/ /data/jdk1.8.0_111/  
aboutyun@slave1:~/
```

在slave1和slave2机器上：

```
sudo mkdir /data  
sudo chmod 777 /data  
mv hadoop-2.6.5/ scala-2.11.8/ jdk1.8.0_111/ /data
```

2) 复制hadoop日志目录

在master机器上：

```
scp -r ~/hadoop aboutyun@slave1:~/  
scp -r ~/hadoop aboutyun@slave2:~/
```

3) 复制环境变量

在master机器上：

```
scp -r ~/.bashrc aboutyun@slave1:~/  
scp -r ~/.bashrc aboutyun@slave2:~/
```

在slave1和slave2机器上：

5. 登录验证

```
source ~/.bashrc
```

在master机器上进行如下操作：

1) 格式化hdfs

hdfs namenode -format

// hdfs

2) 启动hdfs

start-dfs.sh

在master上使用jps命令

```
[aboutyun@master Desktop]$ start-dfs.sh
Starting namenodes on [master]
master: starting namenode, logging to /data/hadoop-2.6.5/logs/hadoop-aboutyun-na
menode-master.out
slave1: starting datanode, logging to /data/hadoop-2.6.5/logs/hadoop-aboutyun-da
tanode-slave1.out
slave2: starting datanode, logging to /data/hadoop-2.6.5/logs/hadoop-aboutyun-da
tanode-slave2.out
Starting secondary namenodes [master]
master: starting secondarynamenode, logging to /data/hadoop-2.6.5/logs/hadoop-ab
outyun-secondarynamenode-master.out
[aboutyun@master Desktop]$ jps
3367 NameNode
3757 Jps
3598 SecondaryNameNode
[aboutyun@master Desktop]$
```

在slave1和slave2上使用jps命令:

```
[aboutyun@slave1 Desktop]$ jps
3336 DataNode
3695 Jps
[aboutyun@slave1 Desktop]$ a
```

上面两张图片说明了在master节点上成功启动了NameNode和SecondaryNameNode，在slave节点上成功启动了DataNode，也就说明HDFS启动成功。

3) 启动yarn

start-yarn.sh

在master使用jps命令

```
[aboutyun@master Desktop]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /data/hadoop-2.6.5/logs/yarn-aboutyun-resou
rcemanager-master.out
slave2: starting nodemanager, logging to /data/hadoop-2.6.5/logs/yarn-aboutyun-n
odemanager-slave2.out
slave1: starting nodemanager, logging to /data/hadoop-2.6.5/logs/yarn-aboutyun-n
odemanager-slave1.out
[aboutyun@master Desktop]$ jps
3367 NameNode
3850 ResourceManager
3918 Jps
3598 SecondaryNameNode
[aboutyun@master Desktop]$
```

在slave1和slave2上使用jps命令

```
[aboutyun@slave1 Desktop]$ jps
4433 Jps
4183 NodeManager
3336 DataNode
[aboutyun@slave1 Desktop]$
```

上面两张图片说明成功启动了ResourceManager和NodeManager，也就是说yarn启动成功。

4) 访问WebUI

在master、slave1和slave2任意一台机器上打开firefox，然后输入http://master:8088/，如果看到如下的图片，就说明我们的hadoop集群搭建成功了。

The screenshot shows the Hadoop WebUI interface in a Mozilla Firefox browser window. The address bar displays 'master:8088/cluster'. The page features the Hadoop logo and a sidebar with navigation options: Cluster, About, Nodes, Applications, Scheduler, and Tools. The main content area is titled 'All Applications' and includes a 'Cluster Metrics' section with a table showing 0 apps submitted, pending, running, or completed, and 0 containers running. Below this, there is a table for applications, which is currently empty, displaying 'Showing 0 to 0 of 0 entries'.

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
0	0	0	0	0	0 B	16 GB	0 B	0	16

