Centos7安装spark-2.3.1

安装过程

1. 安装centos 64位 需要内存空间4G 其他的配置就根据个人的需求来配置了
2. 需要先下载spark的压缩包，不过在
3. 选择spark-2.3.1-bin-hadoop2.7,该版本spark需要Scala2.11，和hadoop2.7+
4. 需要先安装wget(下载文件的命令） #yum -y install wget（非必需内容，纯属为了下载软件包方便）

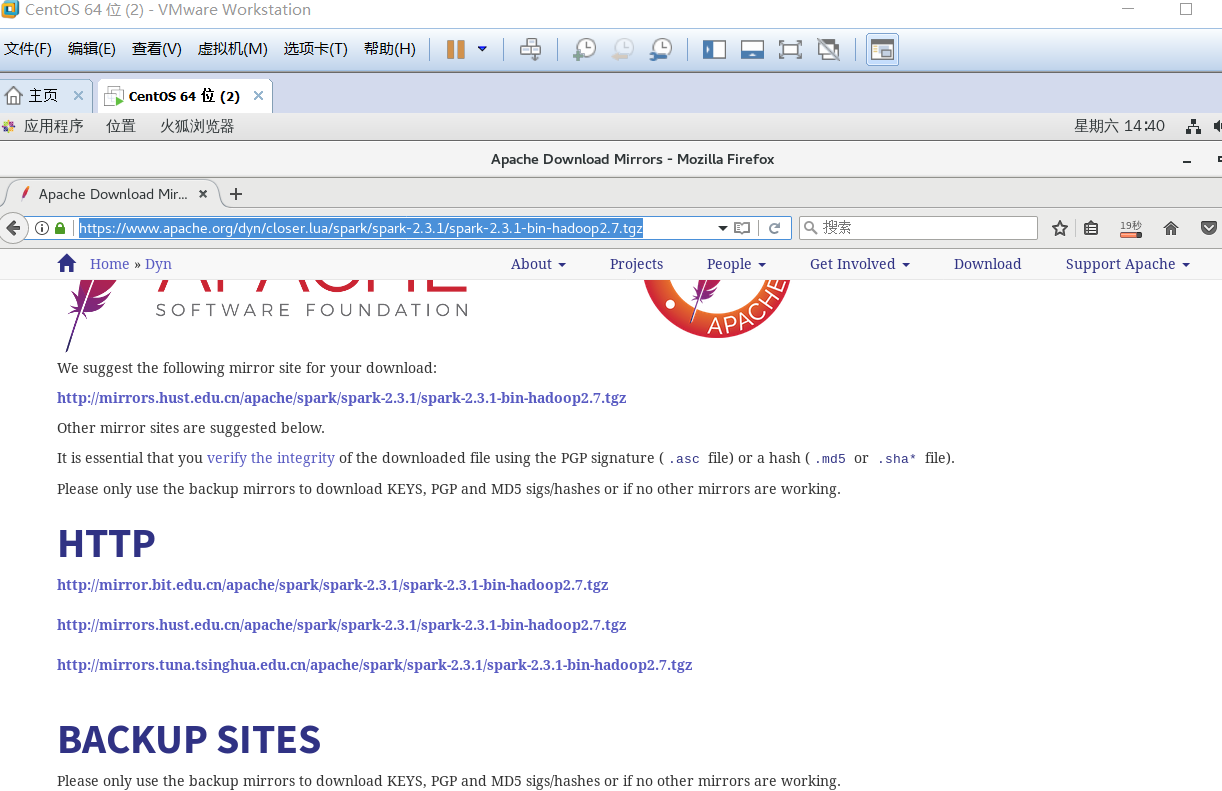


1. 下载spark-2.3.1.tgz

打开centos中自带的浏览器，打开

<https://www.apache.org/dyn/closer.lua/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>

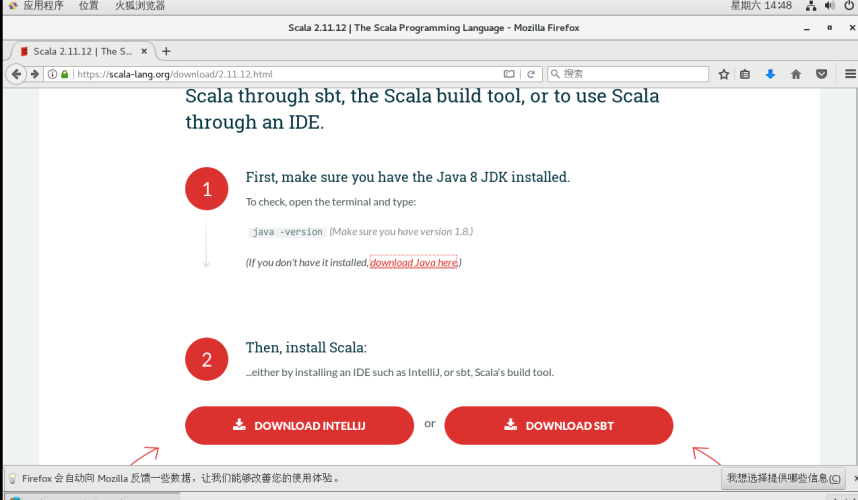
选择第一个压缩包，进行下载。



1. 选择spark-2.3.1-bin-hadoop2.7,该版本spark需要Scala2.11，和hadoop2.7+，所以我们还需要去[下载Scala2.11.12]

<https://scala-lang.org/download/2.11.12.html>

打开后，需要本机装有jdk8



下载jdk8\_64\_linux

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

还需要hadoop2.7+：

<http://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/hadoop-2.7.6/>

把安装包都下好，准备解压。

7 . 安装jdk8（centos7中会有openjdk,此时需要先卸载openjdk，安装sun的jdk)

具体方法可以参见:

https://blog.csdn.net/u010533843/article/details/54989154(转载）

将压缩包解压

1 $ cd /home/cmfchina

2 $ tar -zxvf jdk-8u131-linux-x64.tar.gz

再建立一个java文件夹，把解压后的文件移动到那里面

1 mkdir /root/java

2 mv /root/cmfchina/jdk1.8.0\_171 /root/java

修改环境变量

Vim /etc/profile

在文件的最后加上

export JAVA\_HOME=/root/java/jdk1.8.0\_171

export JRE\_HOME=JRE\_HOME=${JAVA\_HOME}/jre

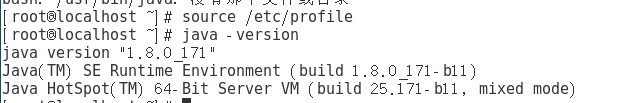
export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib:$CLASSPATH

export JAVA\_PATH=${JAVA\_HOME}/bin:${JRE\_HOME}/bin

export PATH=$PATH:${JAVA\_PATH}

然后source /etc/profile/ 通过命令source /etc/profile让profile文件立即生效

输入java -version来查看jdk版本



安装成功。

1. 解压下载好的3个安装包

mkdir /usr/spark

tar -zxvf scala-2.11.12.tgz -C /usr/spark/

tar -zxvf spark-2.3.1-bin-hadoop2.7\_3.tgz -C /usr/spark/

tar -zxvf hadoop-2.7.6.tar.gz -C /usr/spark/



解压成功，接下来设置环境变量。

gedit /etc/profile

除了刚才的jdk环境还需要加上

#scala env

export SCALA\_HOME=/usr/spark/scala-2.11.12

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

#spark env

export SPARK\_HOME=/usr/spark/spark-2.3.1-bin-hadoop2.7

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

#hadoop env

export HADOOP\_HOME=/usr/spark/hadoop-2.7.6

export PATH=$PATH:HADOOP\_HOME/bin

注意：使用source /etc/profile 命令使其生效

1. 设置hadoop配置文件

（1）新建以下目录

cd /usr/spark

mkdir hadoop

mkdir hadoop/hdfs

mkdir hadoop/hdfs/data

mkdir hadoop/hdfs/name

mkdir hadoop/temp

1. 修改/usr/spark/hadoop-2.7.6/etc/hadoop/文件夹下的多个配置文件

a.修改core-site.xml

Cd /usr/spark/hadoop-2.7.6/etc/hadoop/

然后vim core-site.xml

<configuration>

<property>

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

<value>hdfs://master:9000</value>

<description>HDFS的URI，文件系统://namenode标识:端口号</description>

</property>

<property>

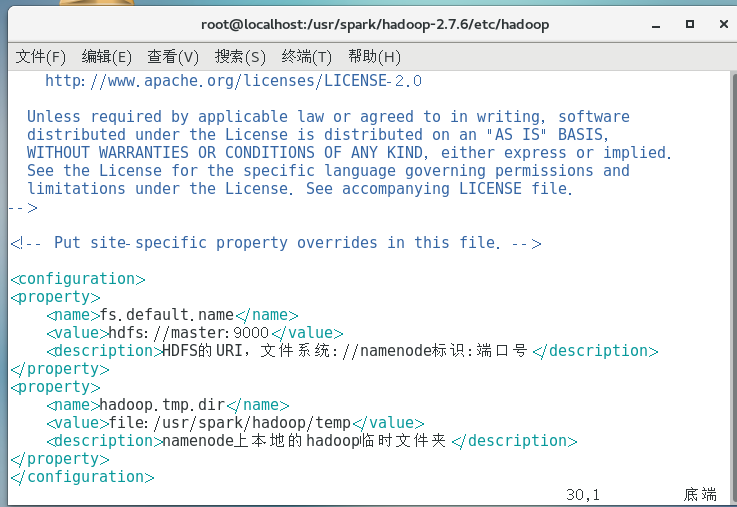
<name>hadoop.tmp.dir</name>

<value>file:/usr/spark/hadoop/temp</value>

<description>namenode上本地的hadoop临时文件夹</description>

</property>

</configuration>



b.修改hdfs-site.xml

```

<configuration>

<property>

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

<value>file:/usr/spark/hadoop/hdfs/name</value>

<description>namenode上存储hdfs名字空间元数据 </description>

</property>

<property>

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

<value>file:/usr/spark/hadoop/hdfs/data</value>

<description>datanode上数据块的物理存储位置</description>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

<description>副本个数，配置默认是3,应小于datanode机器数量</description>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>master:9001</value>

</property>

</configuration>

```

c.修改mapred-site.xml

```

<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>

```

d.修改yarn-site.xml

```

<configuration>

<!-- Site specific YARN configuration properties -->

<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>

e.接着将hadoop-env.sh,mapred-env.sh,yarn-env.sh中的JAVA\_HOME都修改为我们配置好的java路径（根据个人jdk设置而定）

比如，我的是`export JAVA\_HOME=/root/java/jdk1.8.0\_171

` 

####4、配置hadoop节点信息

修改/usr/spark/hadoop-2.7.6/etc/hadoop/下的slaves文件，添加节点，本次预计使用三个slave节点,内容如下：

```

localhost

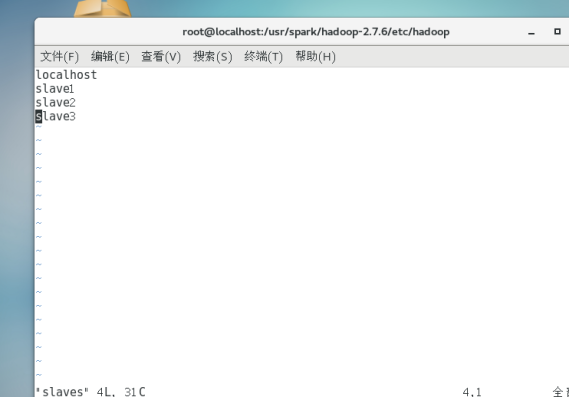
slave1

slave2

slave3

```

其中2和3是我的节点，10.133.43.83 控制和1是10.133.4.223



修改主机hosts文件，内容如下：

![](http://chuantu.biz/t6/336/1530256065x-1404729680.bmp)

修改主机名称gedit /etc/hostname

```

gedit /etc/hostname

在做的过程中，发现校网的ip不适用此次实验，因为在校网这个局域网中，两台主机是无法互相访问的，此时我们的解决办法是采用实验室的网络，不过由于DHCP自动分配，每次链接的ip地址都不同，应及时修改其配置文件。

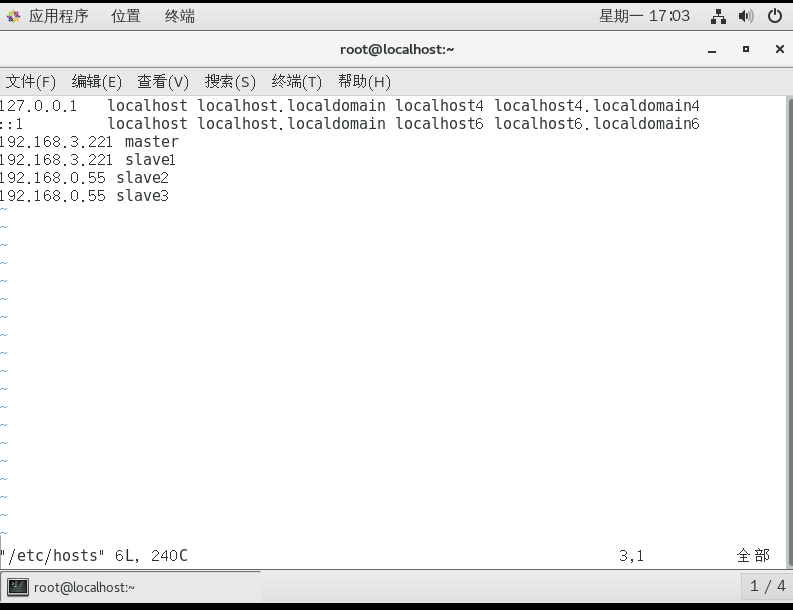
中心节点192.168.3.221 我的节点192.168.0.55

此时修改的部分为

Hostname文件中删除一些东西

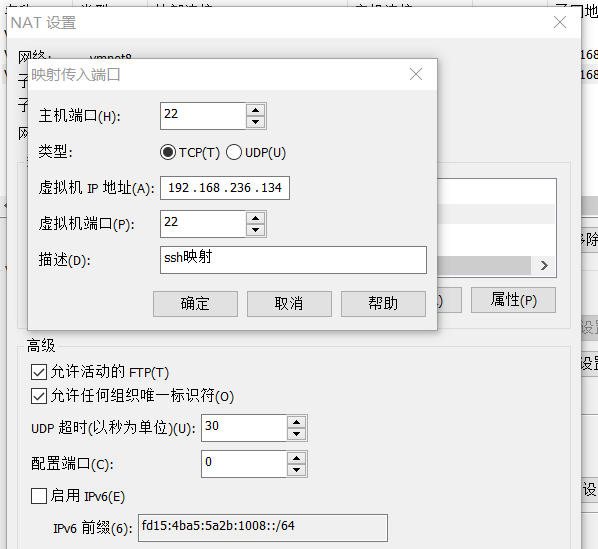


只需要保留自己节点的名称。

在hosts文件中，更改ip地址

此时需要对虚拟机进行与window端口映射

可以参考https://blog.csdn.net/u010947098/article/details/54614756



此时虚拟机与window共享一个IP地址。

需要在终端启动一下ssh,命令为

第一步

#查看本机是否安装SSH软件包

[root@localhost ~]# rpm -qa | grep ssh

openssh-server-6.6.1p1-12.el7\_1.x86\_64

openssh-clients-6.6.1p1-12.el7\_1.x86\_64

libssh2-1.4.3-8.el7.x86\_64

openssh-6.6.1p1-12.el7\_1.x86\_64

#如果没有，则需要安装

[root@localhost /]# yum install openssh-server

第二步

#开启 SSH 服务

[root@localhost ~]# service sshd start

Redirecting to /bin/systemctl start sshd.service

#查看TCP 22端口是否打开

[root@localhost ~]# netstat -ntpl | grep 22

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 17816/sshd

tcp6 0 0 :::22 :::\* LISTEN 17816/sshd

第三步

#接下来便可使用终端仿真程序（例如putty）去登陆远程主机

如果你在客户端不能连接SSH服务的话，那可能是防火墙的原因，终端命令行中输入 iptables -nL 来看是否开放了ssh tcp 22 端口：

[root@localhost ~]# iptables -nL

你可以将防火墙中的规则条目清除掉：

[root@localhost ~]# iptables -F

5、配置ssh无密码登录

(1)ubuntu自带ssh-client，我们还需要ssh-server

apt-get install openssh-server

(2)启动ssh服务

/etc/init.d/ssh start

(3)查看sshd是否启动

ps -e | grep ssh

(4)root账户默认不允许登录ssh，修改权限

`gedit /etc/ssh/sshd\_config`

找到Authentication，修改PermitRootLogin yes，保存

```

/etc/init.d/ssh restart

```

(5)ssh登录本地

`ssh localhost`

SSH首次登录会有提示，直接输入yes即可，这时是需要密码的

(6)生成秘钥

`ssh-keygen -t rsa`

之后一直按Enter键，默认将秘钥保存在.shh/id\_rsa文件中

(7)RSA公钥加入授权文件

cd .ssh

cp id\_rsa.pub authorized\_keys

(8)重新登录，实现免密码登录localhost

6、配置master免密码登录slave1

这里使用scp命令，可自行搜索相关信息，配置确保slave1节点已安装ssh-server

(1)将master根目录下密码复制到slave1的根目录下

root@master:~# scp ~/.ssh/id\_rsa.pub root@slave1:~/.ssh

(2)在slave1节点将RSA公钥加入授权文件

root@slave:~# cp .ssh/id\_rsa.pub authorized\_keys

如有多台计算机，重复以上操作即可

(3)在master节点上ssh登录slave1,实现免密码登录

ssh slave1

7、配置slave1节点的hadoop，同master节点，复制过去即可

8、Hadoop运行

(1)格式化分布式文件系统，在master节点下

cd /usr/local/hadoop/hadoop-2.7.3

bin/hadoop namenode -format

(2)启动hadoop守护进程

sbin/start-all.sh

(3)检测启动情况

jps

这时在master节点可以看到NameNode，SecondaryNameNode，ResourceManager

在slave1节点可以看到DataNode，NodeManager，因为我也将master节地点添加为slaves，所以也能看到类似信息

(4)停止hadoop进程

sbin/stop-all.sh

1. 修改spark配置文件

1)$SPARK\_HOME/conf/spark-env.sh

cp spark-env.sh.template spark-env.sh

添加以下内容:

export SCALA\_HOME=/usr/local/scala/scala-2.12.1

export JAVA\_HOME=/usr/local/java/jdk1.8.0\_111

export SPARK\_MASTER\_IP=master

export SPARK\_WORKER\_MEMORY=512m

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

2)$SPARK\_HOME/conf/slaves

这是在那个路径下再进行配置

cp slaves.template slaves

添加以下内容:

master

slave1