Hadoop完全分布式集群搭建（CentOS6.7+Hadoop2.6）

**小组成员**

**常明明 U201317460**

**陈天阳 U201317453**

**李辉 U201317462**

**包玉龙 U201310321**

**胡超 U201317451**

**Part 0**

笔者从一个Linux小白的角度来讲述亲身经历安装Hadoop的全过程。虽说网上教程是有许多的，但是总感觉或因版本问题或因作者互抄不负责的原因，许多需要强调的细节并没有讲清楚。笔者自己在安装的过程中，由于对Linux不熟悉的原因还是踩了许多坑的，因此把整个过程分享出来。

由于分的服务器莫名的问题我们决定将整个Hadoop环境部署在虚拟机环境中，计划开3个虚拟机，一个作为主节点(Master)，两个从节点(Slave)。

主节点名字叫：DataWorks.Master，从节点分别叫DataWorks.Node1，DataWorks.Node2

**整个安装过程大致分这几步：**

1.安装配置CentOS虚拟机

1.1 下载并安装CentOS

1.2 基本工作：配置yum源

1.3 安装java

1.4 为hadoop专门建立一个用户叫做hadoop（或者其他任何你喜欢的名字）

1.5 将一台CentOS虚拟机克隆成3台

1.6 配置ip，主机名（这里主要是网卡的设置），并设置好hosts

2.配置ssh免密码登陆

3.下载，安装并配置hadoop

4.运行hadoop检查是否一切正常

**Part 1 安装配置CentOS虚拟机**

（下文在linux下如不特别指出就是root用户，所以不特别用sudo命令了，期间可能存在各种用户切换来切换去，有时候可能需要sudo一下，这一点就不做特别说明了）

**1.1 下载并安装CentOS**

第一步就是安装CentOS咯——首先访问中科大源http://mirrors.ustc.edu.cn/，在右侧选取一个发行版（我这里是选择的是CentOS 6.7(x86\_64,LiveDVD)，下载下来是一个iso文件,Live版的一般都会自带图形界面，我安装以后发现它KDE和GNOME都自带了），然后VMware添加虚拟机按照默认的下一步（选择安装iso）然后启动以后按照默认的进行安装（分配了4G内存，20G硬盘），这里都是下一步下一步可以解决的，暂时没发现有坑可以略过……

**1.2 配置yum源**

一个好的yum源对于（像我这样的）新手来说还是很重要的，否则出现奇怪的“明明有些网上教程可以用yum安装的我却要下了源码自己编译好麻烦”这类问题就太打击人了。

基本的源的镜像可以考虑刚才的中科大景镜像源，它的说明里也讲得很清楚，亲测无坑。https://lug.ustc.edu.cn/wiki/mirrors/help/centos 但是CentOS自带的源软件很少，更新很慢（比如python的版本……）所以我们需要一些额外的第三方软件源，epel就是很不错的一个，安装也很自动

**命令如下：**

1. sudo yum install epel-release
2. sudo yum makecache

**1.3 安装java**

因为hadoop是运行在java环境上的，所以jdk一定要装。可以用yum方式轻松安装

yum list java\* #目的是查看一下有哪些java打头的yum包

这里你可以选择安装java 1.7.0版本也可以1.8.0版本，我选择了1.7，用了“我不管反正全都装上去”的任性模式

yum install java-1.7.0-openjdk\* #这样就把java-1.7.0-openjdk打头的所有包都给装上了

随后需要配置一下java环境变量

（如果不确定java安装到哪了，可以用rpm -qa | grep java查看一下有哪些包,rpm -ql java-1.7.0-openjdk-1.7.0.91-2.6.2.2.el6\_7.x86\_64查看一下这个叫java-1.7.0-openjdk-1.7.0.91-2.6.2.2.el6\_7.x86\_64的包有哪些java相关的文件安装到哪些目录了，最后确定下来JAVA\_HOME应该是/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.91.x86\_64）

1.vim /etc/profile

在尾部加上

1.export JAVA\_HOME=/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.91.x86\_64

2.export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

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

大功告成。

**1.4 为hadoop专门建立一个用户**

（命令行默认root用户）

1.groupadd hadoop #添加一个叫hadoop的用户组

2.useradd hadoop -g hadoop #添加一个hadoop用户并加入hadoop组

3.vim /etc/sudoers #编辑sudoers文件，给hadoop用户sudo权限

4.hadoop ALL=(ALL) ALL #在sudoers尾部加上这一行

**1.5 将一台CentOS虚拟机克隆成3台**

因为1.1-1.4这四步是共享的（每个节点都要配置yum源，安装java，都要新建hadoop用户），所以现在我们就把刚才那个虚拟机克隆两份。VMware菜单栏里有克隆选项，操作很傻瓜，在这里就掠过了。目前我们有三个虚拟机了，分别叫做DataWorks.Master，DataWorks.Node1，DataWorks.Node2

**1.6 配置ip，主机名（这里主要是网卡的设置），并设置好hosts**

这一步里头我不小心踩了一个屎一样的坑，请听我细细道来

首先我们需要设计一下ip啊，因为偏爱11这个数字，那么网段就设置在192.168.11.吧 VMware菜单栏“编辑->虚拟网络编辑器”，取消勾选“使用本地DHCP服务将IP地址分配给虚拟机”**

看一下NAT设置里的网关ip——

**

【这里刚开始我看到网关时192.168.11.2，很不开心，强迫症把它改成了192.168.11.1，之后发现连不上网了……其实192.168.11.1是这台物理计算机在192.168.11.网段的ip地址，网关GATEWAY还是老老实实设置成192.168.11.2吧】

由于某种强迫症，不想让Master机器有类似192.168.11.3这样的ip，于是这样设计固定ip——

1、192.168.11.10 DataWorks.Master

2、192.168.11.11 DataWorks.Node1

3、192.168.11.12 DataWorks.Node2

进入CentOS后需要设置一下网络参数，列举一下主要配置文件

* /etc/sysconfig/network-scripts/ifcfg-eth0这是网卡eth0的配置文件，如果只有一张网卡那肯定默认是eth0，之后是eth1, eth2 ... 目前我们只有一张网卡，问题被简化了许多
* /etc/sysconfig/network 这是设置主机名(hostname)的地方
* /etc/resolv.conf 这是配置dns的地方   
  （这里稍微遇到了一点小坑，网络上的教程多半是要求直接设置/etc/sysconfig/network-scripts/ifcfg-eth0的，但是不知道为何这个文件本来不存在，只有ifcfg-lo，而且ifconfig也只能看到lo，还好是有gnome图形界面的，所以我只能在右上角里添加了一个eth0，Edit Connections...   
  ）   
  ifcfg-eth0的配置参考了《鸟哥的Linux私房菜服务器篇第三版》

**DataWorks.Master**

1. DEVICE="eth0" #第二行是HWADDR，是MAC地址，如果只有一个网卡，可以省略此项目，我们目前的情况就是如此，因此跳过
2. NM\_CONTROLLED="no" #不要受到其他软件的网络管理
3. ONBOOT="yes" #是否默认启动
4. BOOTPROTO=none #取得ip的方式，关键词只有dhcp，手动可以填none或者static
5. IPADDR=192.168.11.10 #ip地址啦
6. NETMASK=255.255.255.0 #子网掩码
7. GATEWAY=192.168.11.2 #网关，上文提到了是192.168.11.2

本质上只要设置这8（包括HWADDR在内）个就够了。按esc, 然后:wq保存退出。然后设置/etc/resolv.conf

1. vim /etc/resolv.conf
2. #在尾部填上dns，比如
3. nameserver 192.168.11.2
4. nameserver 8.8.8.8
5. vim /etc/sysconfig/network #这里主要为了更改hostname
6. NETWORKING=yes
7. NETWORKING\_IPV6=no
8. HOSTNAME=DataWorks.Master #这里
9. GATEWAY=192.168.11.2 #这里
10. NTPSERVERARGS=iburst
11. vim /etc/hosts #设置hosts文件，这样计算机之间就可以用计算机名来访问了
12. 127.0.0.1 localhost.localdomain localhost
13. ::1 localhost6.localdomain6 localhost6
14. #增加了下面3行
15. 192.168.11.10 DataWorks.Master
16. 192.168.11.11 DataWorks.Node1
17. 192.168.11.12 DataWorks.Node2

/etc/init.d/network restart #重启网络服务，然后判断是否设置好了

将这个步骤在DataWorks.Node1机器与DataWorks.Node2机器上都重复一遍，设置好对应的hostname与ip地址，并保证网络畅通就行了。   
（笔者在配置过程中多次出现配置成功后一会儿网卡eth0莫名就丢失了，重复了几遍以后才正常，至今是个未解之谜）

**Part 2 配置ssh免密码登陆**

这一部分比较重要。这时候请su hadoop切换到hadoop用户。

1. $ cd ~ #进入hadoop用户目录
2. $ ssh-keygen -t rsa -P ""
3. #这是生成ssh密码的命令，-t 参数表示生成算法，有rsa和dsa两种；-P表示使用的密码，这里使用""空字符串表示无密码。
4. #回车后，会提示三次输入信息，我们直接回车即可。这样就在~/.ssh目录下生成了几个东西
5. $ cd ~/.ssh
6. $ cat id\_rsa.pub >> authorized\_keys #这个命令将id\_rsa.pub的内容追加到了authorized\_keys的内容后面

任何一台机器的~/.ssh/authorized\_keys里保存了DataWorks.Master这台机器的id\_rsa.pub，那么这台机器就可以被DataWorks.Master用免密码方式ssh登录。

接下去我们要做的事情是去另外两台机器用同样的方式生成id\_rsa.pub，

这时候我们进入从节点的两台机器都用hadoop用户进入cd ~目录，生成一下密钥——

$ ssh-keygen -t rsa -P ""

然后我们分别把它们复制到DataWorks.Master机器上，用scp命令

**在DataWorks.Node1执行**

$ scp id\_rsa.pub DataWorks.Master:/home/hadoop/.ssh/id\_rsa.pub.node1

**在DataWorks.Node2执行**

$ scp id\_rsa.pub DataWorks.Master:/home/hadoop/.ssh/id\_rsa.pub.node2

然后我们回到DataWorks.Master的~/.ssh目录，将这两个公钥都追加到authorized\_keys后面

$ cat id\_rsa.pub.node1 >> authorized\_keys

$ cat id\_rsa.pub.node2 >> authorized\_keys

最后我们把这个authorized\_keys用scp命令复制到两个从节点上，这样三台机器两两之间就能互相访问了——

scp authorized\_keys DataWorks.Node1:/home/hadoop/.ssh/

scp authorized\_keys DataWorks.Node2:/home/hadoop/.ssh/

不要以为到这里就结束了，如果你这个时候尝试从DataWorks.Master进行ssh到某个从节点，会发现它依旧要求你输入密码。这时候我不得不吐槽一下网上某些一本正经写笔记的人（你们真的不是抄的吗？）

解决办法是需要给.ssh目录和.ssh/authorized\_keys设置权限，否则如果权限太高系统会认为有被修改的嫌疑，依旧会要求输入密码验证。

一定要设置权限啊！ 一定要设置权限啊！ 一定要设置权限啊！重要的事请说三遍。

三台机器上都进行如下操作——

sudo chmod 644 ~/.ssh/authorized\_keys

sudo chmod 700 ~/.ssh

这时候从三台机器上任何一台都可以无密码访问另外两台了（都用hadoop用户ssh访问一次，第一次会需要输入一个"yes"，把你添加到认识的名单中）

**Part3 下载，安装并配置hadoop**

这是容易的部分，从http://hadoop.apache.org/releases.html 找到一个你想要的版本，下载binary包（下载到你的Master机上，配置完毕以后再复制到其他Slave机上），我这里是hadoop-2.6.2.tar.gz。把它解压在~下，并把文件夹重命名为hadoop——

[hadoop@DataWorks ~]$ tar -zxvf hadoop-2.6.2.tar.gz

[hadoop@DataWorks ~]$ mv hadoop-2.6.2 hadoop

这里要涉及到的配置文件有7个：

~/hadoop/etc/hadoop/hadoop-env.sh

~/hadoop/etc/hadoop/yarn-env.sh

~/hadoop/etc/hadoop/slaves

~/hadoop/etc/hadoop/core-site.xml

~/hadoop/etc/hadoop/hdfs-site.xml

~/hadoop/etc/hadoop/mapred-site.xml

~/hadoop/etc/hadoop/yarn-site.xml

$ vim ~/hadoop/etc/hadoop/hadoop-env.sh

# 这个文件要修改的地方就是JAVA\_HOME环境变量，刚才我们设置过JAVA\_HOME的，在我的案例里改成如下——

# The java implementation to use.

export JAVA\_HOME=/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.91.x86\_64

$ vim ~/hadoop/etc/hadoop/yarn-env.sh

# yarn的环境配置，同样只需要修改JAVA\_HOME就行，找到下面这行——

# some Java parameters

export JAVA\_HOME=/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.91.x86\_64

$ vim ~/hadoop/etc/hadoop/slaves

# 这是设置从节点hostname的地方，一行一个，我们的例子里只要在文件里写上如下两行就行了

DataWorks.Node1

DataWorks.Node2

$ vim mkdir ~/hadoop/tmp #新建一个tmp文件夹

$ vim ~/hadoop/etc/hadoop/core-site.xml3.

#fs.default.name配置了hadoop的HDFS系统的命名，位置为主机的9000端口；hadoop.tmp.dir配置了hadoop的tmp目录的根位置。这里使用了一个文件系统中没有的位置，所以要先用mkdir命令新建一下。

<configuration>

<property>

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

<value>/home/hadoop/hadoop/tmp</value>

</property>

<property>

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

<value>hdfs://DataWorks.Master:9000</value>

</property>

</configuration>

$ vim ~/hadoop/etc/hadoop/hdfs-site.xml

#这个是hdfs的配置文件，dfs.http.address配置了hdfs的http的访问位置；dfs.replication配置了文件块的副本数，一般不大于从机的个数。

<configuration>

<property>

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

<value>DataWorks.Master:50070</value>

</property>

<property>

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

<value>DataWorks.Master:50090</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

接下去是mapreduce的配置文件mapred-site.xml，在2.6.2这个版本里，并没有看到它，不过有一个mapred-site.xml.template的模板文件，把它复制一下吧

$ cd ~/hadoop/etc/hadoop/

$ cp mapred-site.xml.template mapred-site.xml

$ vim mapred-site.xml

在<configuration></configuration>中添加设置。在mapreduce.framework.name属性下配置为yarn。mapred.map.tasks和mapred.reduce.tasks分别为map和reduce的任务数——

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>DataWorks.Master:9001</value>

</property>

<property>

<name>mapred.map.tasks</name>

<value>20</value>

</property>

<property>

<name>mapred.reduce.tasks</name>

<value>4</value>

</property>

<property>

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

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>DataWorks.Master:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>DataWorks.Master:19888</value>

</property>

</configuration>

$ vim ~/hadoop/etc/hadoop/yarn-site.xml

#该文件为yarn框架的配置,主要是一些任务的启动位置

<configuration>

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

<property>

<name>yarn.resourcemanager.address</name>

<value>DataWorks.Master:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>DataWorks.Master:8030</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>DataWorks.Master:8088</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>DataWorks.Master:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>DataWorks.Master:8033</value>

</property>

<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>

</configuration>

配置完成了

1、将配置好的hadoop复制到其他节点

2、scp –r ~/hadoop hadoop@DataWorks.Node1:~/

3、scp –r ~/hadoop hadoop@DataWorks.Node2:~/

这样就安装完了。是不是心里有点小激动？别着急，我们接下去就启动它来验证一下是否正常工作。由于Master机是namenode，所以我们要先对它的hdfs格式化一下。在DataWorks.Master下：

1.$ cd ~/hadoop

2.$ ./bin/hdfs namenode -format #格式化namenode

3.$ ./sbin/start-all.sh #启动hadoop。2.6.2版本的start-all.sh是在./sbin目录下的，看到网上其他一些教程写的是./bin，估计就是版本差异

我是64位的CentOS，在启动时候会出现一个warning，

1.WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

这个问题搜索一下能找到解决办法，主要就是因为下载的bin是32位机器上编译的，所以本地库不好用，需要拿源码在本机重新编译一下。不过据说不影响使用，暂时就不管它了。

**Part 4**

1. [hadoop@DataWorks hadoop]$ jps
2. 43551 ResourceManager
3. 43206 NameNode
4. 43398 SecondaryNameNode
5. 43950 Jps

DataWorks.Master上执行jps查看发现有四个进程

DataWorks.Node1上大致有

Jps

DataNode

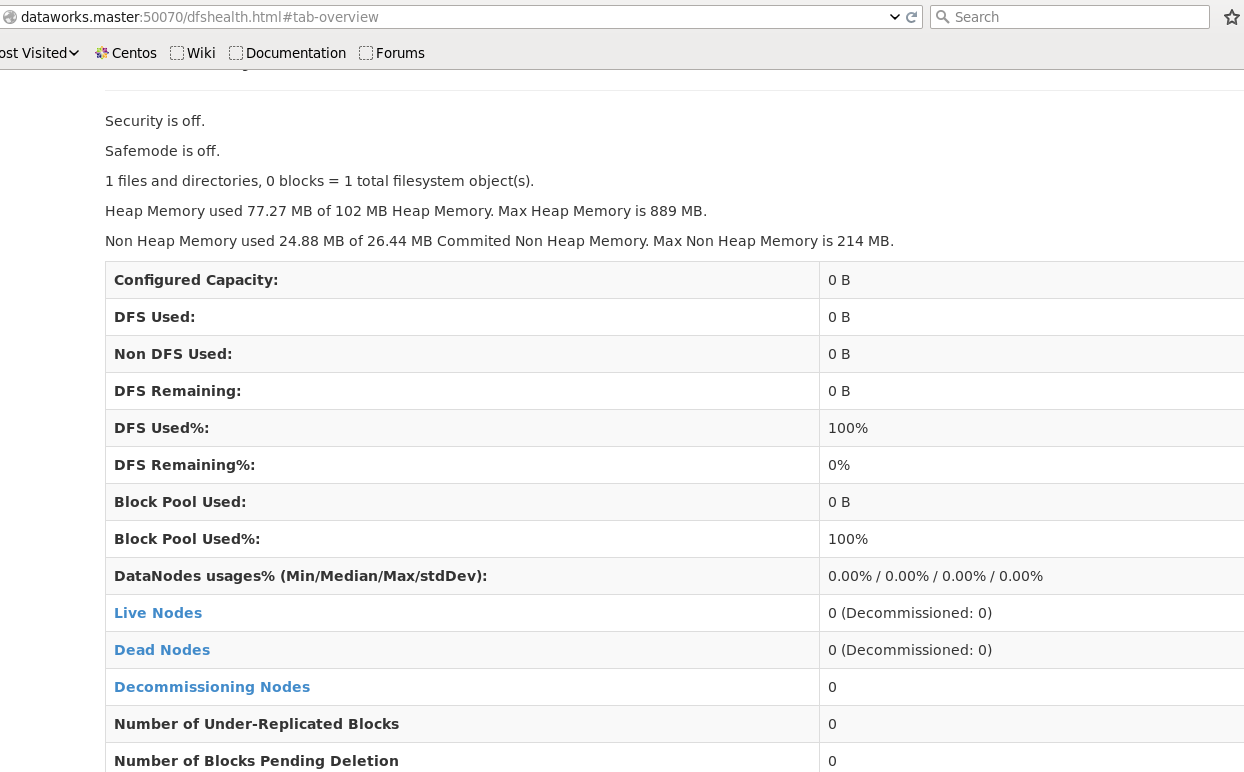
NodeManager

这三个进程。

打开浏览器访问一下http://DataWorks.Master:50070

结果——发生了一些奇怪的事情，我看到空间used 100%，live nodes: 0, dead nodes:0

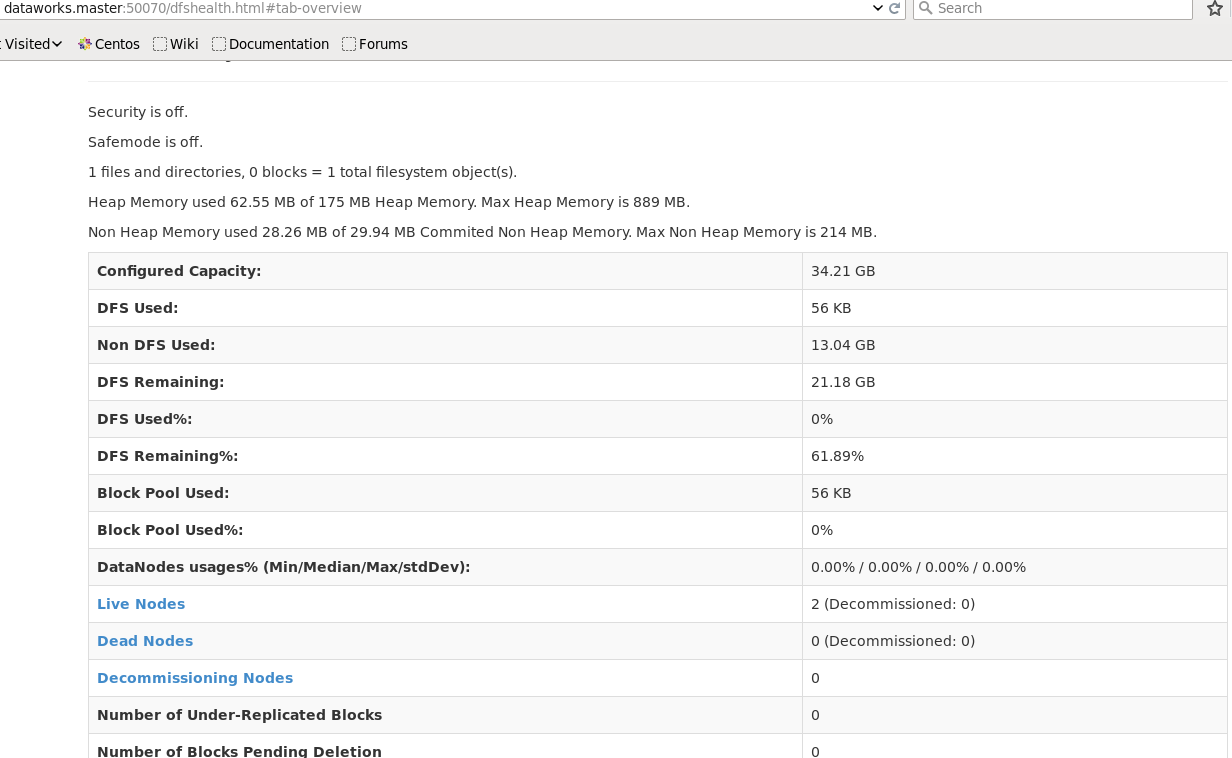
居然都是0……



于是翻看log，在DataWorks.Master的log里并没有看出异常，在DataWorks.Node1中发现了warning，说连接DataWorks.Master:9000失败。

于是我把防火墙关了：

1. sudo service iptables stop

再次访问DataWorks.Master:50070，正常显示了。   
  
但是总感觉完全关闭防火墙不是什么好事情，所以我们添加一些防火墙规则，允许一些端口tcp

注：其实下面这些端口好像并不是全部，所以建议还是直接关掉防火墙。毕竟不该太过纠缠于iptables，这是运维的事情——

1. $ sudo vim /etc/sysconfig/iptables
2. -A INPUT -p tcp -m state --state NEW -m tcp --dport 8020 -j ACCEPT
3. -A INPUT -p tcp -m state --state NEW -m tcp --dport 8021 -j ACCEPT
4. -A INPUT -p tcp -m state --state NEW -m tcp --dport 50010 -j ACCEPT
5. -A INPUT -p tcp -m state --state NEW -m tcp --dport 50020 -j ACCEPT
6. -A INPUT -p tcp -m state --state NEW -m tcp --dport 9000 -j ACCEPT
7. -A INPUT -p tcp -m state --state NEW -m tcp --dport 50030 -j ACCEPT
8. -A INPUT -p tcp -m state --state NEW -m tcp --dport 50060 -j ACCEPT
9. -A INPUT -p tcp -m state --state NEW -m tcp --dport 50075 -j ACCEPT
10. -A INPUT -p tcp -m state --state NEW -m tcp --dport 50090 -j ACCEPT
11. -A INPUT -p tcp -m state --state NEW -m tcp --dport 28680 -j ACCEPT
12. $ sudo service iptables restart #重启防火墙

安装完毕

**参考资料**

1.http://blog.csdn.net/zwx19921215/article/details/1964134，关于ssh免密码登录参考（但是此文没有提到权限的事）

2.http://jingyan.baidu.com/article/27fa73269c02fe46f9271f45.html ，配置hadoop参数时的参考

3.《鸟哥的Linux私房菜服务器篇第三版》，配置网络参数，ssh免密码登录时的参考