**Linux Hadoop**

**说明：Hadoop从版本2开始加入了Yarn这个资源管理器，Yarn并不需要单独安装。只要在机器上安装了JDK就可以直接安装Hadoop，单纯安装Hadoop并不依赖Zookeeper之类的其他东西。**

1下载hadoop

2安装3个虚拟机并实现ssh免密码登录

        2.1安装3个机器

        2.2检查机器名称

        2.3修改/etc/hosts文件

        2.4 给3个机器生成秘钥文件

        2.5 在HiyaHadoopServer1上创建authorized\_keys文件

        2.6将authorized\_keys文件复制到其他机器

        2.7 测试使用ssh进行无密码登录

                   2.7.1在HiyaHadoopServer1上进行测试

                   2.7.2在HiyaHadoopServer2上进行测试

                   2.7.3在HiyaHadoopServer3上进行测试

3安装jdk和hadoop

         3.1安装JDK

         3.2安装hadoop

                  3.2.1上载文件并解压缩

                  3.2.2新建几个目录

                  3.2.3修改etc/hadoop中的一系列配置文件

                              3.2.3.1修改core-site.xml

                              3.2.3.2修改hadoop-env.sh

                              3.2.3.3修改hdfs-site.xml

                              3.2.3.4新建并且修改mapred-site.xml

                              3.2.3.5修改slaves文件

                              3.2.3.6修改yarn-site.xml文件

4启动hadoop

              4.1在namenode上执行初始化

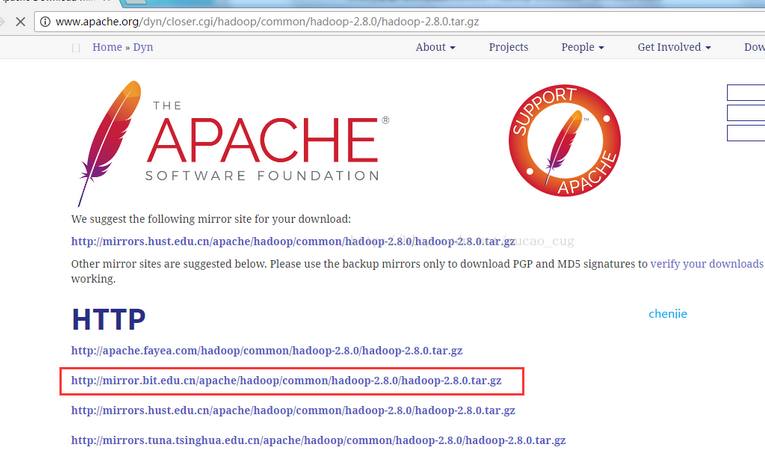
              4.2在namenode上执行启动命令

5测试hadoop

## 1下**载hadoop（2.8.0）**

打开下载地址选择页面：

[http://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.8.0/hadoop-2.8.0.tar.gz](http://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.8.0/hadoop-2.8.0.tar.gz" \t "https://blog.csdn.net/pucao_cug/article/details/_blank)



使用的地址是：

**[http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.8.4/hadoop-2.8.4.tar.gz](http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-2.8.0/hadoop-2.8.0.tar.gz" \t "https://blog.csdn.net/pucao_cug/article/details/_blank)**

## 2安装3个虚拟机并实现ssh免密码登录

### 2.1安装3个机器

这里用的Linux系统是CentOS7(其实Ubuntu也很好，但是这里用的是CentOS7演示)，安装方法就不多说了， 安装3个机器，机器名称分别叫HiyaHadoopServer1、HiyaHadoopServer2、HiyaHadoopServer3 (说明机器名不这么叫可以，待会用hostname命令修改也行)。

说明：为了免去后面一系列授权的麻烦，这里直接使用root账户登录和操作了。

使用ifconfig命令，查看这3个机器的IP。我的机器名和ip的对应关系是:

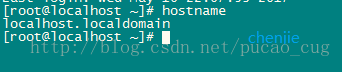
192.168.119.128   HiyaHadoopServer1

192.168.119.129   HiyaHadoopServer2

192.168.119.130   HiyaHadoopServer3

### 2.2检查机器名称

为了后续操作方便，确保机器的hostname是我们想要的。拿192.168.119.128这台机器为例，用root账户登录，然后使用hostname命令查看机器名称



发现，这个机器名称不是我们想要的。不过这个好办， 我给它改个名称，命令是：

hostname   HiyaHadoopServer1

IMG_264

执行完成后，在检查看，是否修改了，敲入hostname命令：

IMG_265

类似的，将其他两个机器，分别改名为HiyaHadoopServer2和HiyaHadoopServer3。

### 2.3 修改/etc/hosts文件

修改这3台机器的/etc/hosts文件，在文件中添加以下内容：

192.168.119.128 HiyaHadoopServer1

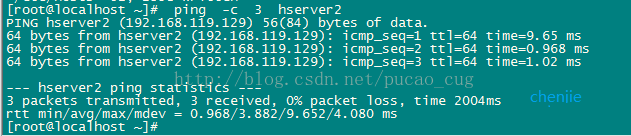
192.168.119.129 HiyaHadoopServer2

192.168.119.130 HiyaHadoopServer3



说明：IP地址没必要和我的一样，这里只是做一个映射，只要映射是对的就可以，至于修改方法，可以用vim命令，也可以在你的本地机器上把hosts文件内容写好后，拿到Linux机器上去覆盖。

配置完成后使用ping命令检查这3个机器是否相互ping得通，以HiyaHadoopServer1为例，在什么执行命令：ping  -c 3  HiyaHadoopServer2



执行命令：

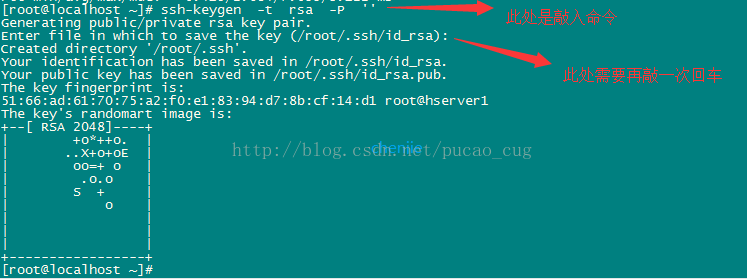
ping  -c  3  HiyaHadoopServer3

ping得通，说明机器是互联的，而且hosts配置也正确。

### 2.4给3个机器生成秘钥文件。

以HiyaHadoopServer1为例，执行命令，生成空字符串的秘钥(后面要使用公钥)，命令是：

ssh-keygen  -t   rsa   -P  ''



因为我现在用的是root账户，所以秘钥文件保存到了/root/.ssh/目录内，可以使用命令查看，命令是：

ls    /root/.ssh/

IMG_270

使用同样的方法为HiyaHadoopServer2和HiyaHadoopServer3生成秘钥(命令完全相同，不用做如何修改)。

### 2.5在HiyaHadoopServer1上创建authorized\_keys文件

### 接下来要做的事情是在3台机器的/root/.ssh/目录下都存入一个内容相同的文件，文件名称叫authorized\_keys，文件内容是我们刚才为3台机器生成的公钥。为了方便，我下面的步骤是现在HiyaHadoopServer1上生成authorized\_keys文件，然后把3台机器刚才生成的公钥加入到这个HiyaHadoopServer1的authorized\_keys文件里，然后在将这个authorized\_keys文件复制到HiyaHadoopServer2和HiyaHadoopServer3上面。

首先使用命令，在HiyaHadoopServer1的/root/.ssh/目录中生成一个名为authorized\_keys的文件，命令是：

touch  /root/.ssh/authorized\_keys

IMG_271

可以使用命令看，是否生成成功，命令是：

ls   /root/.ssh/

IMG_272

其次将HiyaHadoopServer1上的/root/.ssh/id\_rsa.pub文件内容，HiyaHadoopServer2上的/root/.ssh/id\_rsa.pub文件内容，HiyaHadoopServer3上的/root/.ssh/id\_rsa.pub文件内容复制到这个authorized\_keys文件中，复制的方法很多了，可以用cat命令和vim命令结合来弄，也可以直接把这3台机器上的/root/.ssh/id\_rsa.pub文件下载到本地，在本地将authorized\_keys文件编辑好在上载到这3台机器上。

HiyaHadoopServer1机器上我的/root/.ssh/id\_rsa.pub内容是:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD8fTIVorOxgDo81yCEgcJTstUcyfOBecL+NZ/OLXCEzaBMw5pLV0UNRX6SZnaAgu/erazkz4sw74zfRIMzEeKKCeNcZ6W78cg+ZNxDcj8+FGeYqY5+nc0YPhXFVI7AwFmfr7fH5hoIT14ClKfGklPgpEgUjDth0PeRwnUTvUy9A1x76npjAZrknQsnoLYle7cVJZ/zO3eGxS75YEdTYDMv+UMiwtcJg7UxOqR+9UT3TO+xLk0yOl8GIISXzMhdCZkmyAH+DmW56ejzsd+JWwCMm177DtOZULl7Osq+OGOtpbloj4HCfstpoiG58SM6Nba8WUXWLnbgqZuHPBag/Kqjroot@HiyaHadoopServer1

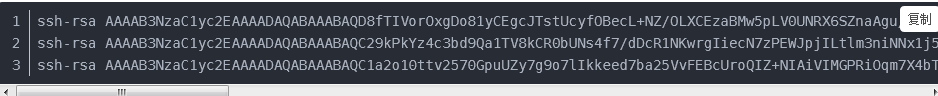
HiyaHadoopServer2机器上我的/root/.ssh/id\_rsa.pub内容是:

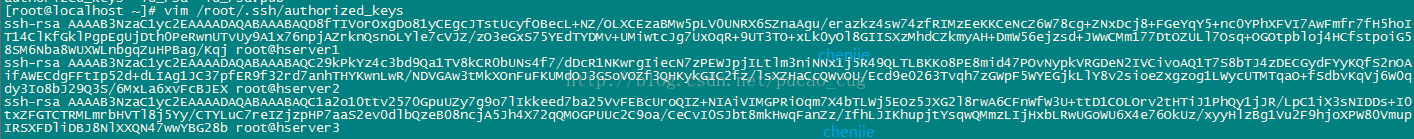
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC29kPkYz4c3bd9Qa1TV8kCR0bUNs4f7/dDcR1NKwrgIiecN7zPEWJpjILtlm3niNNx1j5R49QLTLBKKo8PE8mid47POvNypkVRGDeN2IVCivoAQ1T7S8bTJ4zDECGydFYyKQfS2nOAifAWECdgFFtIp52d+dLIAg1JC37pfER9f32rd7anhTHYKwnLwR/NDVGAw3tMkXOnFuFKUMdOJ3GSoVOZf3QHKykGIC2fz/lsXZHaCcQWvOU/Ecd9e0263Tvqh7zGWpF5WYEGjkLlY8v2sioeZxgzog1LWycUTMTqaO+fSdbvKqVj6W0qdy3Io8bJ29Q3S/6MxLa6xvFcBJEXroot@HiyaHadoopServer2

HiyaHadoopServer2机器上我的/root/.ssh/id\_rsa.pub内容是:

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC1a2o10ttv2570GpuUZy7g9o7lIkkeed7ba25VvFEBcUroQIZ+NIAiVIMGPRiOqm7X4bTLWj5EOz5JXG2l8rwA6CFnWfW3U+ttD1COLOrv2tHTiJ1PhQy1jJR/LpC1iX3sNIDDs+I0txZFGTCTRMLmrbHVTl8j5Yy/CTYLuC7reIZjzpHP7aaS2ev0dlbQzeB08ncjA5Jh4X72qQMOGPUUc2C9oa/CeCvI0SJbt8mkHwqFanZz/IfhLJIKhupjtYsqwQMmzLIjHxbLRwUGoWU6X4e76OkUz/xyyHlzBg1Vu2F9hjoXPW80VmupIRSXFDliDBJ8NlXXQN47wwYBG28broot@HiyaHadoopServer3

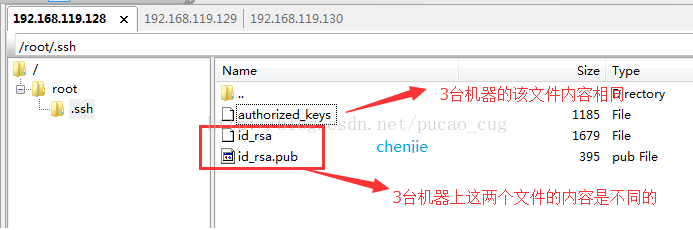
合并之后，我的HiyaHadoopServer1机器上的/root/.ssh/authorized\_keys文件内容是：





### 2.6将authorized\_keys文件复制到其他机器

HiyaHadoopServer1机器的/root/.ssh/目录下已经有authorized\_keys这个文件了，该文件的内容也已经OK了，接下来要将该文件复制到HiyaHadoopServer2的/root/.ssh/和HiyaHadoopServer3的/root/.ssh/。复制的方法有很多，最简单的就是用SecureFX可视化工具操作吧。复制完成后，可以看到三台机器的/root/.ssh目录下都有了这样的文件.下图已经说得很清楚了，三台机器的/root/.ssh都有同名的文件，但是只有authorized\_keys文件的内容是相同的。

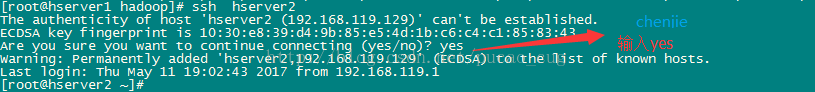


## 2.7测试使用ssh进行无密码登录

### 2.7.1在HiyaHadoopServer1上进行测试

输入命令：

ssh   HiyaHadoopServer2



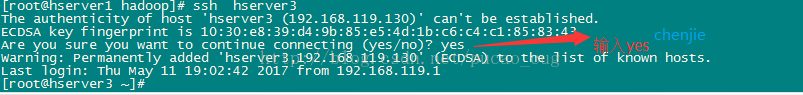
输入命令：

 exit回车

IMG_276

输入命令：

ssh   HiyaHadoopServer3



输入命令：

 exit回车

IMG_278

### 2.7.2 在HiyaHadoopServer2上进行测试

方法类似2.7.1，只不过命令变成了ssh  HiyaHadoopServer1和ssh  HiyaHadoopServer3，但是一定要注意的是，每次ssh完成后，都要执行exit，否则你的后续命令是在另外一台机器上执行的。

### 2.7.3 在HiyaHadoopServer3上进行测试

方法类似2.7.1，只不过命令变成了ssh  HiyaHadoopServer1和ssh  HiyaHadoopServer2，但是一定要注意的是，每次ssh完成后，都要执行exit，否则你的后续命令是在另外一台机器上执行的。

## 3安装jdk和hadoop

说明，为了省去一系列获取管理员权限，授权等繁琐操作，精简教程，这里都是使用root账户登录并且使用root权限进行操作。

### 3.1 安装JDK

安装jdk在这里不在细数，如果有需要可以参考该博文(虽然那篇博文用的是ubuntu，但是jdk安装在CentOS下也一样)：

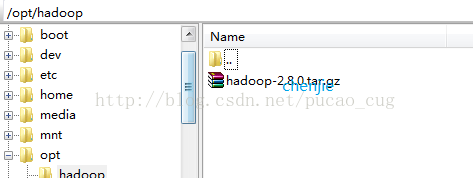
[http://blog.csdn.net/pucao\_cug/article/details/68948639](http://blog.csdn.net/pucao_cug/article/details/68948639" \t "https://blog.csdn.net/pucao_cug/article/details/_blank)

### 3.2  安装hadoop

**注意： 3台机器上都需要重复下面所讲的步骤。**

**3.2.1 上载文件并解压缩**

在opt目录下新建一个名为hadoop的目录，并将下载得到的hadoop-2.8.0.tar上载到该目录下，如图:



进入到该目录，执行命令:

cd   /opt/hadoop

 执行解压命令：

 tar  -xvf   hadoop-2.8.0.tar.gz

说明：3台机器都要进行上述操作，解压缩后得到一个名为hadoop-2.8.0的目录。

**3.2.2新建几个目录**

在/root目录下新建几个目录，复制粘贴执行下面的命令：

mkdir /root/hadoop

mkdir /root/hadoop/tmp

mkdir /root/hadoop/var

mkdir /root/hadoop/dfs

mkdir /root/hadoop/dfs/name

mkdir /root/hadoop/dfs/data

**3.2.3 修改etc/hadoop中的一系列配置文件**

修改/opt/hadoop/hadoop-2.8.0/etc/hadoop目录内的一系列文件。

**3.2.3.1 修改core-site.xml**

 修改/opt/hadoop/hadoop-2.8.0/etc/hadoop/core-site.xml文件

在<configuration>节点内加入配置:

 <property>

        <name>hadoop.tmp.dir</name>

        <value>/root/hadoop/tmp</value>

        <description>Abase for other temporary directories.</description>

   </property>

   <property>

        <name>fs.default.name</name>

        <value>hdfs://HiyaHadoopServer1:9000</value>

   </property>

</configuration>

**3.2.3.2 修改hadoop-env.sh**

修改/opt/hadoop/hadoop-2.8.0/etc/hadoop/hadoop-env.sh文件

将export   JAVA\_HOME=${JAVA\_HOME}

修改为：

export   JAVA\_HOME=/opt/java/jdk1.8.0\_121

说明：修改为自己的JDK路径

**3.2.3.3 修改hdfs-site.xml**

修改/opt/hadoop/hadoop-2.8.0/etc/hadoop/hdfs-site.xml文件

在<configuration>节点内加入配置:

<property>

   <name>dfs.name.dir</name>

   <value>/root/hadoop/dfs/name</value>

   <description>Path on the local filesystem where theNameNode stores the namespace and transactions logs persistently.</description>

</property>

<property>

   <name>dfs.data.dir</name>

   <value>/root/hadoop/dfs/data</value>

   <description>Comma separated list of paths on the localfilesystem of a DataNode where it should store its blocks.</description>

</property>

<property>

   <name>dfs.replication</name>

   <value>2</value>

</property>

<property>

      <name>dfs.permissions</name>

      <value>false</value>

      <description>need not permissions</description>

</property>

说明：dfs.permissions配置为false后，可以允许不要检查权限就生成dfs上的文件，方便倒是方便了，但是你需要防止误删除，请将它设置为true，或者直接将该property节点删除，因为默认就是true。

**3.2.3.4 新建并且修改mapred-site.xml**

在该版本中，有一个名为mapred-site.xml.template的文件，复制该文件，然后改名为mapred-site.xml，命令是：

cp /opt/hadoop/hadoop-2.8.0/etc/hadoop/mapred-site.xml.template /opt/hadoop/hadoop-2.8.0/etc/hadoop/mapred-site.xml

修改这个新建的mapred-site.xml文件，在<configuration>节点内加入配置:

 <property>

   <name>mapred.job.tracker</name>

   <value>HiyaHadoopServer1:49001</value>

</property>

<property>

      <name>mapred.local.dir</name>

       <value>/root/hadoop/var</value>

</property>

<property>

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

       <value>yarn</value>

</property>

**3.2.3.5 修改slaves文件**

修改/opt/hadoop/hadoop-2.8.0/etc/hadoop/slaves文件，将里面的localhost删除，添加如下内容：

HiyaHadoopServer2

HiyaHadoopServer3

**3.2.3.6 修改yarn-site.xml文件**

修改/opt/hadoop/hadoop-2.8.0/etc/hadoop/yarn-site.xml文件，

在<configuration>节点内加入配置(注意了，内存根据机器配置越大越好，我这里只配2个G是因为机器不行):

<property>

        <name>yarn.resourcemanager.hostname</name>

        <value>HiyaHadoopServer1</value>

   </property>

   <property>

        <description>The address of the applications manager interface in the RM.</description>

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

        <value>${yarn.resourcemanager.hostname}:8032</value>

   </property>

   <property>

        <description>The address of the scheduler interface.</description>

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

        <value>${yarn.resourcemanager.hostname}:8030</value>

   </property>

   <property>

        <description>The http address of the RM web application.</description>

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

        <value>${yarn.resourcemanager.hostname}:8088</value>

   </property>

   <property>

        <description>The https adddress of the RM web application.</description>

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

        <value>${yarn.resourcemanager.hostname}:8090</value>

   </property>

   <property>

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

        <value>${yarn.resourcemanager.hostname}:8031</value>

   </property>

   <property>

        <description>The address of the RM admin interface.</description>

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

        <value>${yarn.resourcemanager.hostname}:8033</value>

   </property>

   <property>

        <name>yarn.nodemanager.aux-services</name>

        <value>mapreduce\_shuffle</value>

   </property>

   <property>

        <name>yarn.scheduler.maximum-allocation-mb</name>

        <value>2048</value>

        <discription>每个节点可用内存,单位MB,默认8182MB</discription>

   </property>

   <property>

        <name>yarn.nodemanager.vmem-pmem-ratio</name>

        <value>2.1</value>

   </property>

   <property>

        <name>yarn.nodemanager.resource.memory-mb</name>

        <value>2048</value>

</property>

   <property>

        <name>yarn.nodemanager.vmem-check-enabled</name>

        <value>false</value>

</property>

**参数解释**

**1） yarn.resourcemanager.address**

参数解释：ResourceManager 对客户端暴露的地址。客户端通过该地址向RM提交应用程序，杀死应用程序等。

默认值：${yarn.resourcemanager.hostname}:8032

**2） yarn.resourcemanager.scheduler.address**

参数解释：ResourceManager 对ApplicationMaster暴露的访问地址。ApplicationMaster通过该地址向RM申请资源、释放资源等。

默认值：${yarn.resourcemanager.hostname}:8030

**3） yarn.resourcemanager.resource-tracker.address**

参数解释：ResourceManager 对NodeManager暴露的地址.。NodeManager通过该地址向RM汇报心跳，领取任务等。

默认值：${yarn.resourcemanager.hostname}:8031

**4） yarn.resourcemanager.admin.address**

参数解释：ResourceManager 对管理员暴露的访问地址。管理员通过该地址向RM发送管理命令等。

默认值：${yarn.resourcemanager.hostname}:8033

**5） yarn.resourcemanager.webapp.address**

参数解释：ResourceManager对外web ui地址。用户可通过该地址在浏览器中查看集群各类信息。

默认值：${yarn.resourcemanager.hostname}:8088

**6） yarn.resourcemanager.scheduler.class**

参数解释：启用的资源调度器主类。目前可用的有FIFO、Capacity Scheduler和Fair Scheduler。

默认值：

org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler

**7） yarn.resourcemanager.resource-tracker.client.thread-count**

参数解释：处理来自NodeManager的RPC请求的Handler数目。

默认值：50

**8） yarn.resourcemanager.scheduler.client.thread-count**

参数解释：处理来自ApplicationMaster的RPC请求的Handler数目。

默认值：50

**9） yarn.scheduler.minimum-allocation-mb/ yarn.scheduler.maximum-allocation-mb**

参数解释：单个可申请的最小/最大内存资源量。比如设置为1024和3072，则运行MapRedce作业时，每个Task最少可申请1024MB内存，最多可申请3072MB内存。

默认值：1024/8192

说明：yarn.nodemanager.vmem-check-enabled这个的意思是忽略虚拟内存的检查，如果你是安装在虚拟机上，这个配置很有用，配上去之后后续操作不容易出问题。如果是实体机上，并且内存够多，可以将这个配置去掉。

## 4启动hadoop

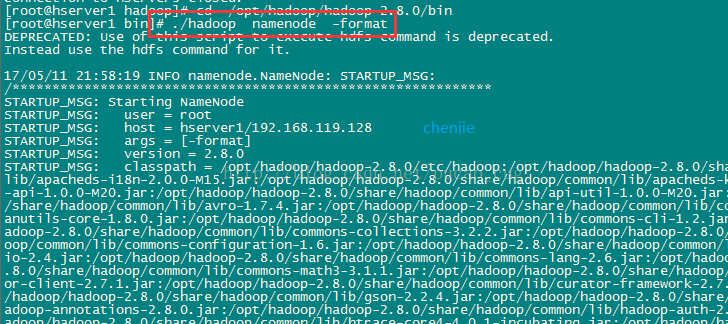
### 4.1在namenode上执行初始化

因为HiyaHadoopServer1是namenode，HiyaHadoopServer2和HiyaHadoopServer3都是datanode，所以只需要对HiyaHadoopServer1进行初始化操作，也就是对hdfs进行格式化。进入到HiyaHadoopServer1这台机器的/opt/hadoop/hadoop-2.8.0/bin目录，也就是执行命令：

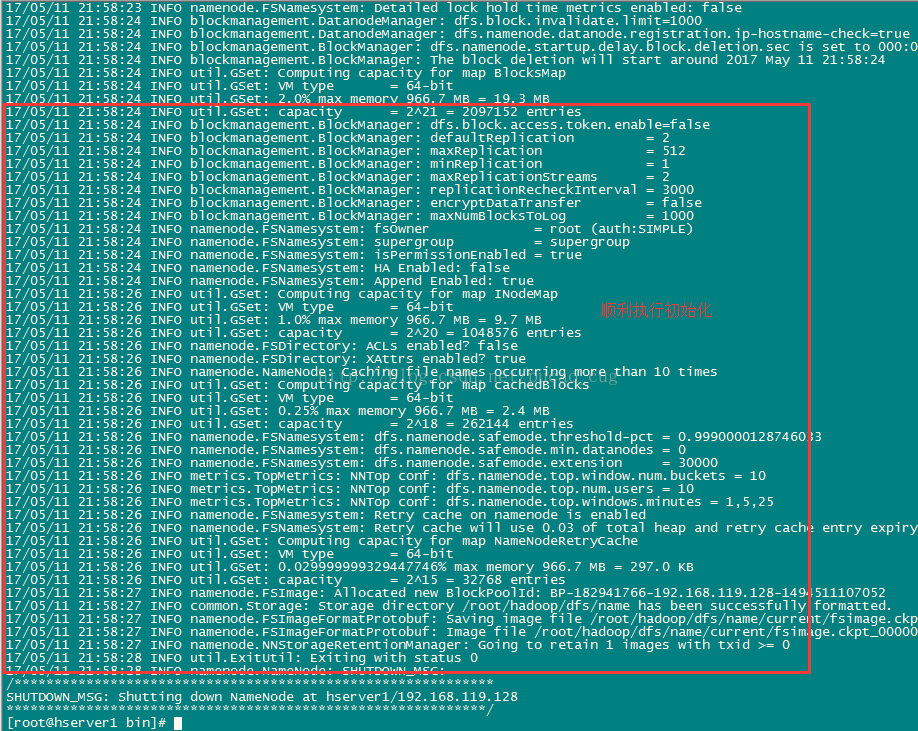
cd   /opt/hadoop/hadoop-2.8.0/bin

 执行初始化脚本，也就是执行命令：

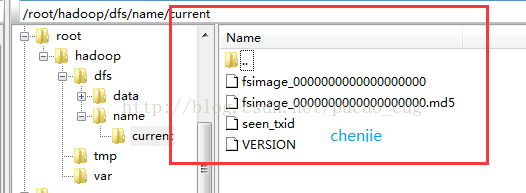
 ./hadoop  namenode  -format



稍等几秒，不报错的话，即可执行



格式化成功后，可以在看到在/root/hadoop/dfs/name/目录多了一个current目录，而且该目录内有一系列文件



### 4.2在namenode上执行启动命令

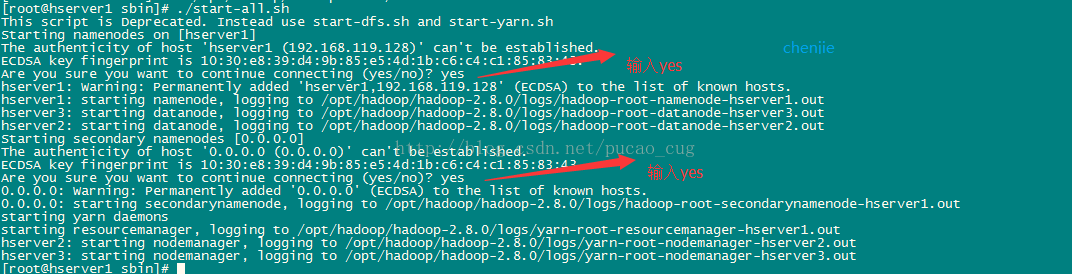
因为HiyaHadoopServer1是namenode，HiyaHadoopServer2和HiyaHadoopServer3都是datanode，所以只需要再HiyaHadoopServer1上执行启动命令即可。进入到HiyaHadoopServer1这台机器的/opt/hadoop/hadoop-2.8.0/sbin目录，也就是执行命令：

cd    /opt/hadoop/hadoop-2.8.0/sbin

执行初始化脚本，也就是执行命令：

 ./start-all.sh

第一次执行上面的启动命令，会需要我们进行交互操作，在问答界面上输入yes回车



**或者 ： 分别启动namenode和datanode节点**

[hadoop@db01 hadoop-2.8.0]$sbin/hadoop-daemon.sh start namenode  
[hadoop@db01 hadoop-2.8.0]$sbin/hadoop-daemon.sh start datanode

**启动yarn**

[hadoop@db01 hadoop-2.8.0]$ sbin/yarn-daemon.sh start resourcemanager  
[hadoop@db01 hadoop-2.8.0]$ sbin/yarn-daemon.sh start nodemanage

**[hadoop@db01 hadoop-2.5.0]$ jps**  
14573 NodeManager  
13490 DataNode  
13400 NameNode  
14685 Jps  
14315 ResourceManager

## 5测试hadoop

 haddoop启动了，需要测试一下hadoop是否正常。

执行命令，关闭防火墙，CentOS7下，命令是：

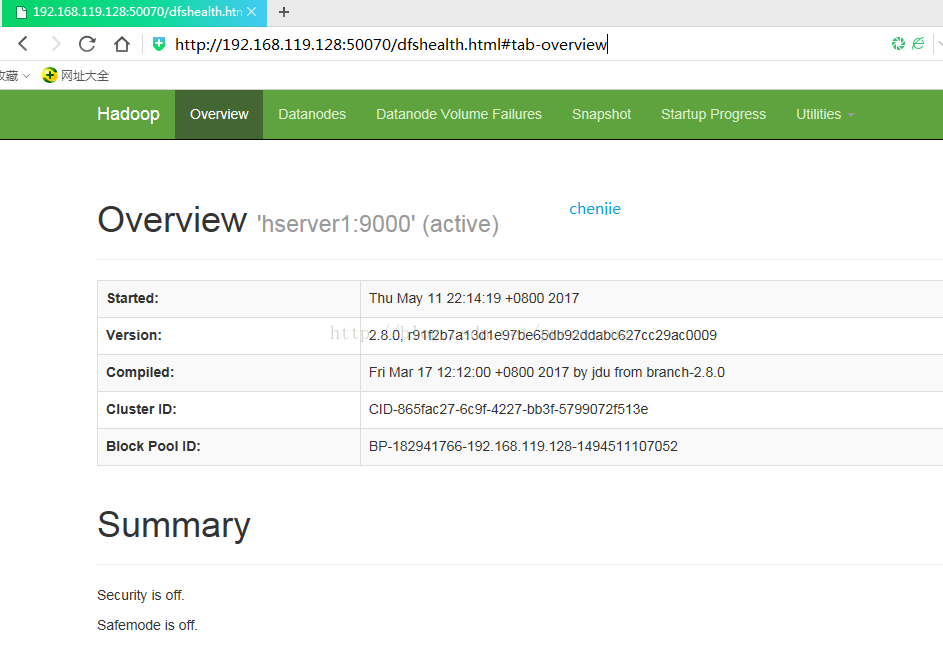
systemctl   stop   firewalld.service

IMG_284

HiyaHadoopServer1是我们的namanode，该机器的IP是192.168.119.128，在本地电脑访问如下地址:

[http://192.168.119.128:50070/](http://192.168.119.128:50070/" \t "https://blog.csdn.net/pucao_cug/article/details/_blank)

自动跳转到了hdfs页面



在本地浏览器里访问如下地址：

[http://192.168.119.128:8088/](http://192.168.119.128:8088/" \t "https://blog.csdn.net/pucao_cug/article/details/_blank)

自动跳转到了yarn监控页面

