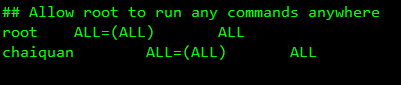
# Hadoop伪分布式安装

**1.给用户提高权限**

**Vi /etc/sudoers 将用户添加上去**



**2.在根目录下创建文件夹存放数据**

Mkdir opt 用来所需要的所有数据

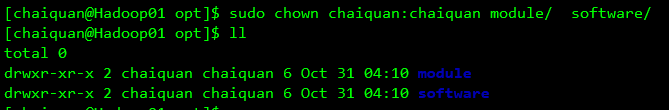
Mkdir module 用来存放减压文件之后的数据

Mkdir software 用来存放安装包文件



**3.修改文件权限**

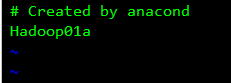
Sudo chown 用户名：用户名 文件名



**4.修改主机名**

Vi /etc/hostname 修改主机名

Vi /etc/sysconfig/network

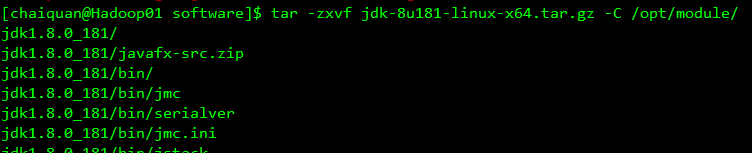


**5.将需要的java文件移动到software文件目录**



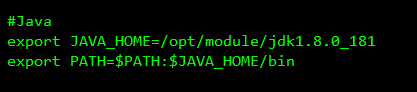
**6.将java文件解压到module目录下**

**Tar -zxvf 压缩包 -C 指定目录**



1. **配置java环境变量**

Vi /etc/profile



#JAVA

export JAVA\_HOME=/opt/module/jdk1.8.0\_181

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

#HADOOP

export HADOOP\_HOME=/opt/module/hadoop-2.7.2

export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

刷配置文件：source /etc/profile

1. **Hadoop配置文件**
2. hadoop-env.sh

cd /opt/module/hadoop/etc/hadoop

Vi hadoop-env.sh



1. core-site.xml

Vi core-site.xml

 <!-- 指定HDFS中 NameNode的地址 -->

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.99.200:9000</value>

</property>

<!-- 指定hadoop运行时产生文件的存储目录 -->

<property>

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

<value>/opt/module/hadoop-2.7.2/data/tmp</value>

</property>

1. hdfs-site.xml

Vi hdfs-site.xml



<!-- 指定HDFS副本的数量 -->

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>/opt/module/hadoop-2.7.2/hdpdata/name</value>

</property>

<property>

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

<value>/opt/module/hadoop-2.7.2/hdpdata/data</value>

</property>

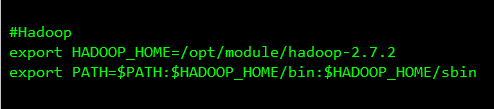
4.slaves

vi slaves 直接追加主机名字



1. **配置Hadoop环境变量**

Vi /etc/profile



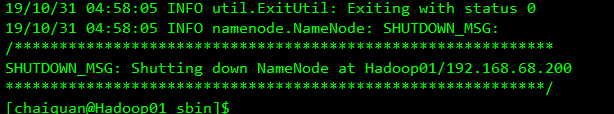
#HADOOP

export HADOOP\_HOME=/opt/module/hadoop-2.7.2

export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

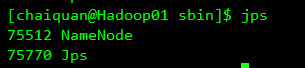
刷配置文件：source /etc/profile

**格式化namenode**



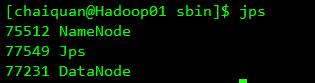
**启动namenode进程**

hadoop-daemon.sh start namenode



**启动datanode进程**

hadoop-daemon.sh start datanode



**打开进程之后，可以通过本地浏览器访问**  主机IP地址:50070

**10.防火墙**

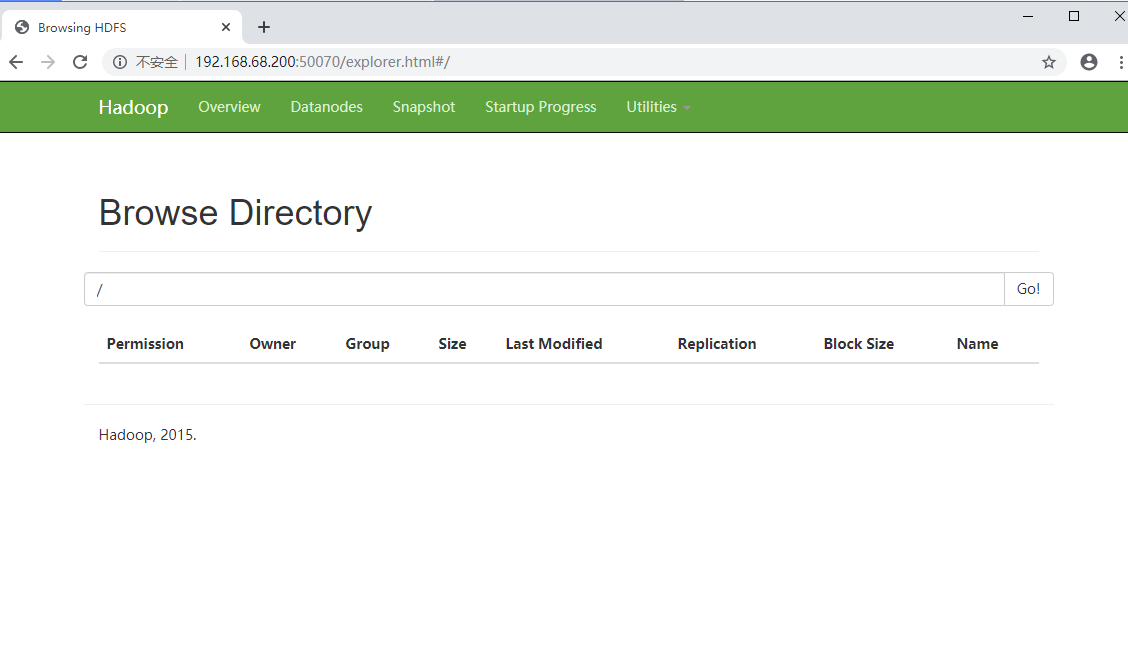
启动： systemctl start firewalld

关闭： systemctl stop firewalld

查看状态： systemctl status firewalld

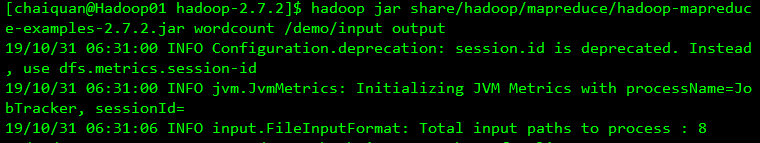
开机禁用 ： systemctl disable firewalld

开机启用 ： systemctl enable firewalld

**11.测试程序**

hadoop fs -mkdir /demo

hadoop fs -put input/ /demo  
 hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7. jar wordcount /demo/input output



# Hadoop完全分布式安装

#### **1.搭建Hadoop集群**

#### **1.配置服务器**

#### **1个主节点：Mage1(192.168.105.10)，2个（从）子节点， Mage2(192.168.105.11)，Mage3(192.168.105.12)**

#### **配置主节点名**

****vi /etc/sysconfig/network****

**添加内容：**

NETWORKING=yes  
 HOSTNAME=Mage1

**另外两台子节点分别将Mage1改为Mage2和Mage3**

#### **配置hosts**

#### ****vi /etc/hosts****

****添加内容****

192.168.105.10 Mage1

192.168.105.11 Mage2

192.168.105.12 Mage3

****将保存到文件拷贝到另外两台子节点上****

****scp /etc/hosts**[root@192.168.1](mailto:root@192.168.102.136)**05.11:/etc/****

****scp /etc/hosts**[root@192.168.1](mailto:root@192.168.102.136)**05.12:/etc/****

****使这个主机名生效****

****/bin/hostname Mage1****

#### ****4.**关闭防火墙**

**1.查看防火墙状态**  
 ****firewall-cmd --state****

**2.停止防火墙**  
 ****systemctl stop firewalld.service****

**3.禁止防火墙开机启动**  
 ****systemctl disable firewalld.service****

#### ****5.**时间同步**

**命令行输入：**yum install ntp** 下载ntp插件** ，  
 **下载完成后 命令行输入：**ntpdate -u**[ntp1.aliyun.com](http://ntp1.aliyun.com/)**  
 **然后命令行输入**：****date****

#### **配置ssh无密码访问=====可不设置**

#### ****生成公钥密钥对****

****在每个节点上分别执行：****

****命令行输入：ssh-keygen -t rsa****

****一直按回车直到生成结束****

****执行结束之后每个节点上的/root/.ssh/目录下生成了两个文件 id\_rsa 和 id\_rsa.pub****

****其中前者为私钥，后者为公钥****

****在主节点上执行：****

****命令行输入：cp id\_rsa.pub authorized\_keys****

****将子节点的公钥拷贝到主节点并添加进authorized\_keys****

****将两个子节点的公钥拷贝到主节点上****

****分别在两个子节点上执行：****

****Scp /root/.ssh/id\_rsa.pub root@Mage1:/root/.ssh/id\_rsa\_Mage2.pub****

****scp /root/.ssh/id\_rsa.pub root@Mage1:/root/.ssh/id\_rsa\_Mage3.pub****

****然后在主节点上，将拷贝过来的两个公钥合并到authorized\_keys文件中去****

****主节点上执行：****

****cat id\_rsa\_Mage2.pub>> authorized\_keys****

****cat id\_rsa\_Mage3.pub>> authorized\_keys****

****将主节点的authorized\_keys文件分别替换子节点的authorized\_keys文件****

****主节点上用scp命令将authorized\_keys文件拷贝到子节点的相应位置****

****scp authorized\_keys rootMage2:/root/.ssh/****

****scp authorized\_keys rootMage3:/root/.ssh/****

1. **安装jdk**

**详细配置：**<https://blog.csdn.net/superman404/article/details/83591324>

**精简配置：**<https://www.cnblogs.com/dummyly/p/10005731.html>

**1.将jdk解压到制定个文件夹和目录下然后配置环境变量，这里直接把Hadoop的环境变量一起添加上**

**Vi /etc/profile**

|  |
| --- |
| ****export JAVA\_HOME=/opt/module/jdk1.8.0\_181****  ****export JRE\_HOME=$JAVA\_HOME/jre****  ****export PATH=$JAVA\_HOME/bin:$PATH****  ****export CLASSPATH=$JAVA\_HOME/lib:$JRE\_HOME/lib****  ****#hadoop****  ****export HADOOP\_HOME=/opt/module/hadoop-2.7.2****  ****export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin**** |

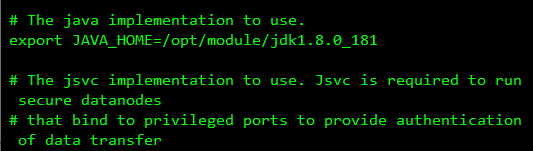
****然后刷新刚刚配置的环境变量 source /etc/profile****

1. ****将Hadoop安装到指定目录下，因为刚刚配置了环境变量，所以这里就不用配置环境变量，直接修改配置文件就行，**需要配置的文件的位置为/opt/module/hadoop-2.7.2/etc/hadoop，需要修改的有以下几个文件,**

**[hadoop-env.sh](http://hadoop-env.sh/)**  
 **[yarn-env.sh](http://yarn-env.sh/) **core-site.xml  
 hdfs-site.xml  
 mapred-site.xml****  
 ****yarn-site.xml****  
 ****slaves****

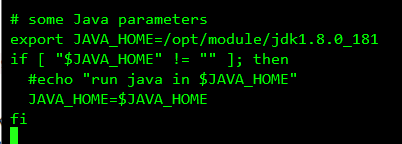
**其中hadoop-env.sh和yarn-env.sh里面都要添加jdk的环境变量**

##### **hadoop-env.sh中把这个添加进去export JAVA\_HOME=/opt/module/jdk1.8.0\_181**



##### **yarn-env.sh**

**添加这样的代码进去export JAVA\_HOME=/opt/module/jdk1.8.0\_181**



##### **core-site.xml**

|  |
| --- |
| <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://Mage1:9000</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/opt/module/hadoop-2.7.2/data/tmp</value>  </property>  </configuration> |

##### **hdfs-site.xml**

|  |
| --- |
| <configuration>  <property>  <name>dfs.namenode.name.dir</name>  <value>/opt/module/hadoop-2.7.2/dpdata/name</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>/opt/module/hadoop-2.7.2/dpdata/data</value>  </property>  <property>  <name>dfs.replication</name>  <value>2</value>  </property>  </configuration> |

##### **mapred-site.xml**

**（注意要将mapred-site.xml.template重命名为 .xml的文件 Mv mapred-site.xml.template mapred-site.xml）**

|  |
| --- |
| <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration> |

##### **yarn-site.xml**

|  |
| --- |
| <configuration>  <!-- Site specific YARN configuration properties -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <property>  <name>yarn.resourcemanager.hostname</name>  <value>mage1</value>  </property>  <property>  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>  <value>org.apache.hadoop.mapred.ShuffleHandler</value>  </property>  </configuration> |

##### **Slaves**

****Mage2  
 Mage3****

****拷贝hadoop安装文件到子节点****

****主节点上执行：****

****scp -r /usr/hadoop-2.6.4 root@Mage2:/usr****

****scp -r /usr/hadoop-2.6.4 root@Mage3:/usr****

****拷贝profile到子节点****

****主节点上执行：****

****scp /etc/profile root@Mage2:/etc/****

****scp /etc/profile root@Mage3:/etc/****

****在两个子节点上分别使新的profile生效：****

****source /etc/profile****

****格式化主节点的namenode****

****主节点上进入hadoop目录****

****然后执行：****

****./bin/hadoop namenode -format****

****新版本用下面的语句不用hadoop命令了****

****./bin/hdfs namenode -format****

****提示：successfully formatted表示格式化成功****

****启动hadoop****

****主节点上在hadoop目录下执行：****

****./sbin/start-all.sh****

****主节点上jps进程如下：****

****NameNode****

****SecondaryNameNode****

****ResourceManager****

****每个子节点上的jps进程如下：****

****DataNode****

****NodeManager****

****如果这样表示hadoop集群配置成功****

# 在线安装mysql

## 安装mysql

1.下载yum库

wget http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm

1. 安装yum库

yum localinstall -y mysql57-community-release-el7-7.noarch.rpm

1. 安装数

yum install -y mysql-community-server据库

1. 启动MySQL服务

systemctl start mysqld.service

1. 查看MySQL的启动状态

systemctl status mysqld

1. 设置“开机启动”

systemctl enable mysqld

systemctl daemon-reload

## 配置mysql

修改root本地配置密码，mysql安装完成之后，在/var/log/mysqld.log文件中给root生成了一个默认密码

1. 找到默认密码

grep 'temporary password' /var/log/mysqld.log

1. 登录MySQL

mysql -uroot -p

1. 修改默认密码

ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码';

## 配置默认编码为utf-8

****修改/etc/my.cnf配置文件，在[mysqld]下添加编码配置****

[mysqld]

character\_set\_server=utf8

init\_connect='SET NAMES utf8'

# 安装hive

## 解压文件到指定目录

1.解压

Tar -zxvf hive文件 -C /opt/module/

2.将hive名字更改一下方便使用

Mv hive名字 hive

3.配置环境变量

vi /etc/profile

在文件中加入以下内容：

#hive

export HIVE\_HOME=/opt/module/hive

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$HIVE\_HOME/bin

4.刷新环境变量

source /etc/profile

5.将jdbc包放到hive的lib目录下

Cpmysql-connector-java-5.1.40-bin.jar /usr/local/src/hive-1.1.0-cdh5.10.0/lib

## 配置hive

进入hive/conf目录

**1. 根据模板文件创建 hive-env.sh 文件**

cp hive-env.sh.template hive-env.sh

1. **修改 hive-env.sh 文件**

#hive

export HADOOP\_HOME=/opt/module/hadoop-2.6.0

export HIVE\_CONF\_DIR=/opt/module/hive/conf

1. **创建hive-site.xml 文件**

touch hive-site.xml

4.编辑**hive-site.xml**文件

vim hive-site.xml

添加配置

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>javax.jdo.option.ConnectionURL</name>  <value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</value>  </property>  <property>  <name>javax.jdo.option.ConnectionDriverName</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <property>  <name>javax.jdo.option.ConnectionUserName</name>  <value>root</value><!-- 数据库连接用户名 -->  </property>  <property>  <name>javax.jdo.option.ConnectionPassword</name>  <value>Pengtao@520</value><!-- 数据库连接密码 -->  </property>  </configuration> |

1. 注释
   1. 如果配置文件没有错误的话可能是hadoop/share/hadoop/yarn目录的jline-2.12.jar版本老化了，需要将hive的这个jar包拷贝过来

# 安装zookeeper

### 1.将zookeeper基本配置布置好

1. 将安装包解压到指定文件夹

tar -zxvf zookeeper-3.4.7.tar.gz -C /usr/local/src/

1. 将目录下的zookeeper改名成zookeeper
2. 进去zookeeper主目录创建一个data存放数据的文件

Cd /usr/local/src/zookeeper

Mkdir data

1. 分别进去每台虚拟机zookeeper的data目录创建一个zookeeper的id文件

Echo 1 > myid

Echo 2 > myid

Echo 3 > myid

### 集群配置

1. 进入zookeeper的conf文件下将zoo.sample.cfg拷贝并且编辑

cp zoo-sample.cfg zoo.cfg

vi zoo.cfg

将这个配置进行修改

|  |
| --- |
| dataDir=/usr/local/src/zookeeper/data  这个是数据保存的路径  server.1=mage1:2888:3888  server.2=mage2:2888:3888  server.3=mage3:2888:3888  这些节点都是固定的 |

1. 配置文件

Vi /etc/profile

添加zookeeper的启动目录

|  |
| --- |
| #ZOOKEEPER  export ZOOKEEPER\_HOME=/usr/local/src/zookeeper  export PATH=$ZOOKEEPER\_HOME/bin:$PATH |

1. 刷新配置文件

Source /etc/profile

### 拷贝文件夹到其他虚拟机

1. 拷贝zookeeper文件夹到其他虚拟机上

scp -r /usr/local/src/zookeeper/ root@mage2:/usr/local/src

scp -r /usr/local/src/zookeeper/ root@mage3:/usr/local/src

1. 拷贝配置文件到其他虚拟机

/etc/profile root@mage2:/etc/

/etc/profile root@mage3:/etc/

1. 然后到每个子节点刷新配置文件

Source /etc/profile

### ****5.****启动zookeeper

    启动zookeeper的各种命令操作如下，可以使用绝对路径操作这些命令，也可使用相对路径操作这些命令，相对路径需要进到zookeeper服务的bin目录进行操作。

#启动ZK服务:

bin/zkServer.sh start#停止ZK服务:

bin/zkServer.sh stop#重启ZK服务:

bin/zkServer.sh restart#查看ZK服务状态:

bin/zkServer.sh status

# 安装kafka

### 将安装包解压到指定目录

1. 进入安装包文件夹将kafka解压

**tar -xzvf kafka\_2.11-0.10.2.1.tgz -C /usr/local/src**

1. 进入安装目录将kafka改名成kafka

**mv kafka\_2.11-0.10.2.1 kafka2.11**

### 2. 配置环境变量

1. 编辑环境变量

vi /etc/profile

1. 添加配置文件

export KAFKA\_HOME=/opt/kafka2.11

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

1. 刷新配置文件

Source /etc/profile

### 3. 修改kafka的**server.properties配置**

1. 进入kafka 的config目录

Cd /usr/local/src/kafka/config

1. 编辑 server.properties  
    broker.id=1 分别将其他虚拟机改成2,3

这下面两个都是在文件最后了

|  |
| --- |
| message.max.byte=5242880 #在log.retention.hours=168属性下加上如下三行配置  default.replication.factor=3  replica.fetch.max.bytes=5242880  将这个ip改成自己虚拟机的三个IP  zookeeper.connect=192.168.123.50:2181,192.168.123.51:2181,192.168.123.52:2181 #修改zookeeper引用外部的zookeeper |

### 4.将kafka文件拷贝到其他虚拟机

1.将kafka文件拷贝到其他节点

Scp -r /usr/local/src/kafka root@mage2:/usr/local/src

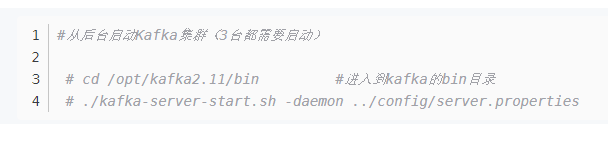
Scp -r /usr/local/src/kafka root@mage3:/usr/local/src

2. 将环境变量拷贝到其他节点

Scp /etc/profile root@mage2:/etc/

Scp /etc/profile root@mage3:/etc/

注意修改broker.id为123



# 离线安装mysql

### 准备工作

1. 先将自带的数据库删掉

rpm -qa | grep mariadb  
 rpm -e --nodeps 文件名

### 将mysql安装包解压

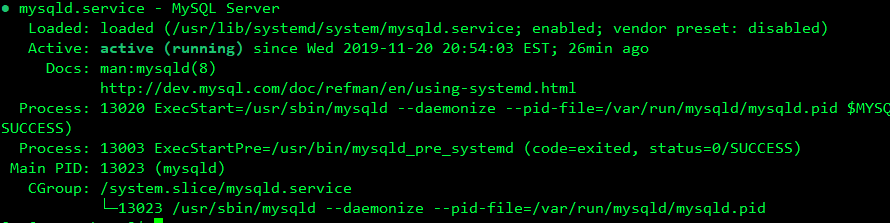
1. 在指定目录下创建一个mysql 文件夹
   * 1. Mkdir /usr/local/src/mysql
2. 将mysql解压到mysql文件夹，这里不是用-zxvf而是-xvf
   * 1. tar -xvf mysql-5.7.23-1.el7.x86\_64.rpm-bundle.tar.gz -C /usr/local/src/mysql
3. 进入mysql文件夹
   * 1. Cd /usr/local/src/mysql
4. 将一些rmp包安装
   * 1. rpm -ivh mysql-community-common-5.7.27-1.el7.x86\_64.rpm
     2. rpm -ivh mysql-community-libs-5.7.27-1.el7.x86\_64.rpm
     3. rpm -ivh mysql-community-devel-5.7.27-1.el7.x86\_64.rpm
     4. rpm -ivh mysql-community-libs-compat-5.7.27-1.el7.x86\_64.rpm
     5. rpm -ivh mysql-community-client-5.7.27-1.el7.x86\_64.rpm
     6. rpm -ivh mysql-community-server-5.7.27-1.el7.x86\_64.rpm

### 将mysql服务启动

1. 查看服务是否启动
   * 1. Service msyqld status
2. 启动服务
   * 1. Systemctl start mysqld

### 如果这个时候报错“Job for mysqld.service failed because the control process exited with error code‘’

* 1. 提示是不能初始化，这时候将var/lib/mysql删除
     1. Rm -rf /var/lib/mysql

1. 重新启动服务
   * 1. Systemctl start mysqld
2. 查看服务是否启动
   * 1. Service msyqld status
     2. 
     3. 表示服务成功启动

### 启动mysql

1. 查看mysql的原始密码
   * 1. grep 'temporary password' /var/log/mysqld.log
2. 启动mysql
   * 1. Mysql -u root -p
3. 将获取到的原始密码粘贴
4. 进去mysql之后将mysql密码更改
   * 1. SET PASSWORD FOR 'root'@'localhost'= "Pengtao@520";
5. 退出mysql
   * 1. Exit

### 设置mysql自启动

* 1. 7.至此，mysql已安装完成，并设好root密码，这里设置开机自启动。  
        法一   、vi /etc/rc.local       添加service mysqld start启动

# 安装flume

网址：<https://blog.csdn.net/weixin_34021089/article/details/89652712>

### 将flume的安装包解压到指定目录下

1. 将flume安装包解压
   * 1. tar -zxvf apache-flume-1.6.0-bin.tar.gz -C /usr/local/src/
2. 将flume 更改名字
   * 1. Mv flume的名字 flume

### 2. 配置环境变量

1. 配置flume的环境变量

#FLUME

export FLUME\_HOME=/usr/local/src/flume

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

export FLUME\_CONF\_DIR=$FLUME\_HOME/conf

1. 刷新环境变量
   * 1. Source /etc/profile

### 3. 添加flume的配置文件

1. 进入flume的conf目录下

Cd /usr/local/src/flume/conf/

* 1. 将flume-conf.properties.template拷贝并且改名成 flume-conf.properties
     1. cp flume-conf.properties.template flume-conf.properties
  2. 编辑这个文件
     1. Vi flume-conf.properties
     2. 添加一下这些配置

|  |
| --- |
| * + 1. a1.sources = r1     2. a1.sinks = k1     3. a1.channels = c1     4. # Describe/configure the source     5. a1.sources.r1.type = spooldir     6. a1.sources.r1.spoolDir = /usr/local/src/flume/temp #创建一个监控目录     7. a1.sources.r1.fileHeader = true     8. a1.sources.r1.deserializer.outputCharset=UTF-8     9. # Describe the sink     10. a1.sinks.k1.type = hdfs     11. a1.sinks.k1.hdfs.path = hdfs://mage1:9000/log #改成主节点的路径     12. a1.sinks.k1.hdfs.fileType = DataStream     13. a1.sinks.k1.hdfs.writeFormat=Text     14. a1.sinks.k1.hdfs.maxOpenFiles = 1     15. a1.sinks.k1.hdfs.rollCount = 0     16. a1.sinks.k1.hdfs.rollInterval = 0     17. a1.sinks.k1.hdfs.rollSize = 1000000     18. a1.sinks.k1.hdfs.batchSize = 100000     19. # Use a channel which buffers events in memory     20. a1.channels.c1.type = memory     21. a1.channels.c1.capacity = 1000000     22. a1.channels.c1.transactionCapacity = 100000     23. # Bind the source and sink to the channel     24. a1.sources.r1.channels = c1     25. a1.sinks.k1.channel = c1 |

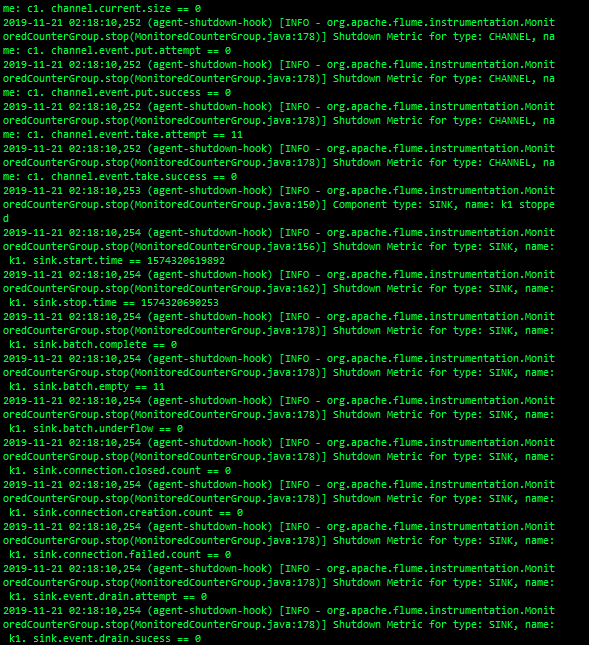
* 1. 创建监控的文件夹
     1. Mkdir /usr/local/src/temp

### 4. 运行

1. 进去flume主目录

Cd /usr/local/src/flume

* 1. 启动flume
     1. *bin/flume-ng agent --conf conf --conf-file conf/flume-conf.properties --name a1 -Dflume.root.logger=INFO,console*

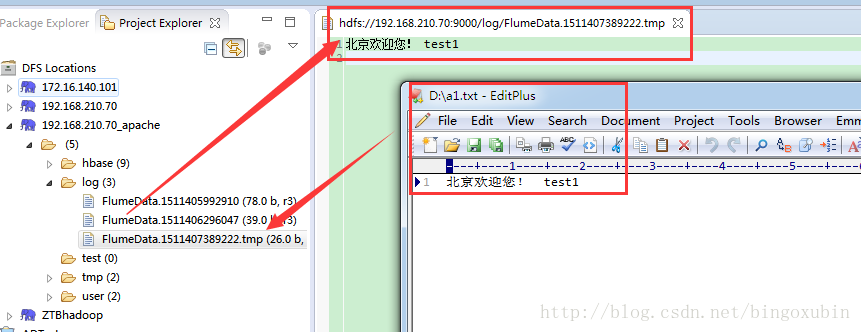


### 5. 运行效果

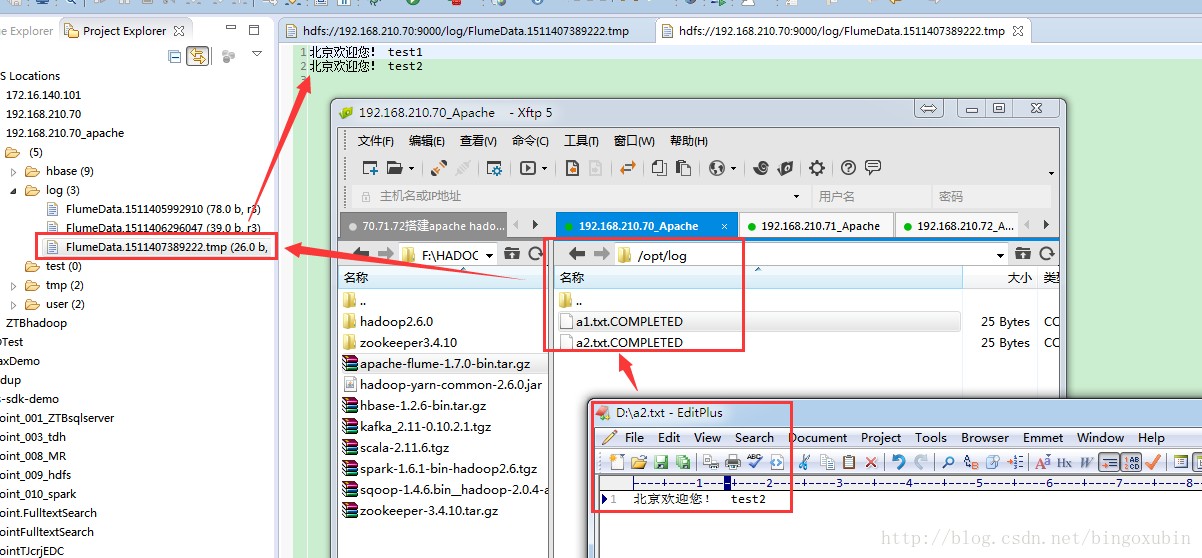
****命令执行后，它会存在一个进程，一直在监听目录，所以会一直开着！****

****效果就是：当你在/opt/log目录下添加了txt文档后，HDFS中的log目录下会生成一个文件，将txt中的内容拷贝到HDFS的log中，当再传入一个文件时，Flume会将txt中的内容添加到已创建的文件后面！如果关闭这个flume agent，重新开启一个，那么HDFS的log中，会重新生成一个文件进行收集！****

在/opt/log目录下，上传一个a1.txt文档，里面写着“北京欢迎您，test1”，在/opt/log/目录下刷新，看到a1.txt文件，立马改变成为a1.txt.COMPLETED，查看HDFS的log目录，发现增加了一个文件，查看其中内容。



当在/opt/log目录下，再上传一个a2.txt文档，里面写着“北京欢迎您，test2”，查看效果。  
****注：再次上传的文件名，不能和之前的文件名相同，否则报错了！****



# 安装sqoop

### 将安装包解压到指定目录下

1. 进去安装包目录下
   * 1. cd /h3cu
2. 将安装包解压到指定目录下安装
   * 1. tar -zxvf sqoop-1.4.5-cdh5.4.5.tar.gz -C /usr/local/src/
3. 将sqoop改名字为sqoop
   * 1. MV sqoop的初始名字 sqoop

### 配置环境变量

1. 配置sqoop的环境变量
   * 1. Vi /etc/proflie
2. 添加这些变量

|  |
| --- |
| * + 1. #SQOOP     2. export SQOOP\_HOME=/usr/local/src/sqoop     3. export PATH=$SQOOP\_HOME/bin:$PATH |

1. 刷新环境变量
   * 1. Source /etc/profile

### 3. 修改sqoop-env.sh文件

1. 进入sqoop的conf目录

Cd /usr/local/src/sqoop/conf

* 1. 将sqoop的文件拷贝
     1. *cp sqoop-env-template.sh sqoop-env.sh*
  2. 编辑这个文件
     1. *vim sqoop-env.sh*
  3. *在这个文件修改这些配置*

|  |
| --- |
| * + 1. #Set path to where bin/hadoop is available     2. export HADOOP\_COMMON\_HOME=/usr/local/src/hadoop     3. #Set path to where hadoop-\*-core.jar is available     4. export HADOOP\_MAPRED\_HOME=/usr/local/src/hadoop     5. #set the path to where bin/hbase is available     6. #export HBASE\_HOME=     7. #Set the path to where bin/hive is available     8. export HIVE\_HOME=/usr/local/src/hive |

### 4.将msyql的驱动jar包拷贝到sqoop的lib目录下

1. 找到mysql的jar包

Cd /h3cu

2. 拷贝这个mysql的jar包到sqoop的lib目录下

cp mysql-connector-java-5.1.47.jar /usr/local/src/sqoop/lib/

### 5.启动sqoop服务

1. 检验sqoop是否安装成功

Sqoop help

1. 查看hive下面所有的表
   1. sqoop list-tables --username root --password Pengtao@520 --connect jdbc:mysql://192.168.123.50:3306/hive?characterEncoding=UTF-8
   2. 