Hadoop

本文档使用hadoop2.7.2版本

## hadoop介绍

hadoop是一个可靠,可伸缩,分布式计算框架,开源软件

### 包含四大模块

common

核心jar:hadoop-common-2.7.2.jar

hdfs

mapreduce

yarn

### hadoop完全分布式进程

hdfs:namenode,datanode,secondarynamenode

启动脚本:start-all.sh(start-dfs.sh)

yarn:resourcemanager,nodemanager

启动脚本: start-all.sh(start-yarn.sh)

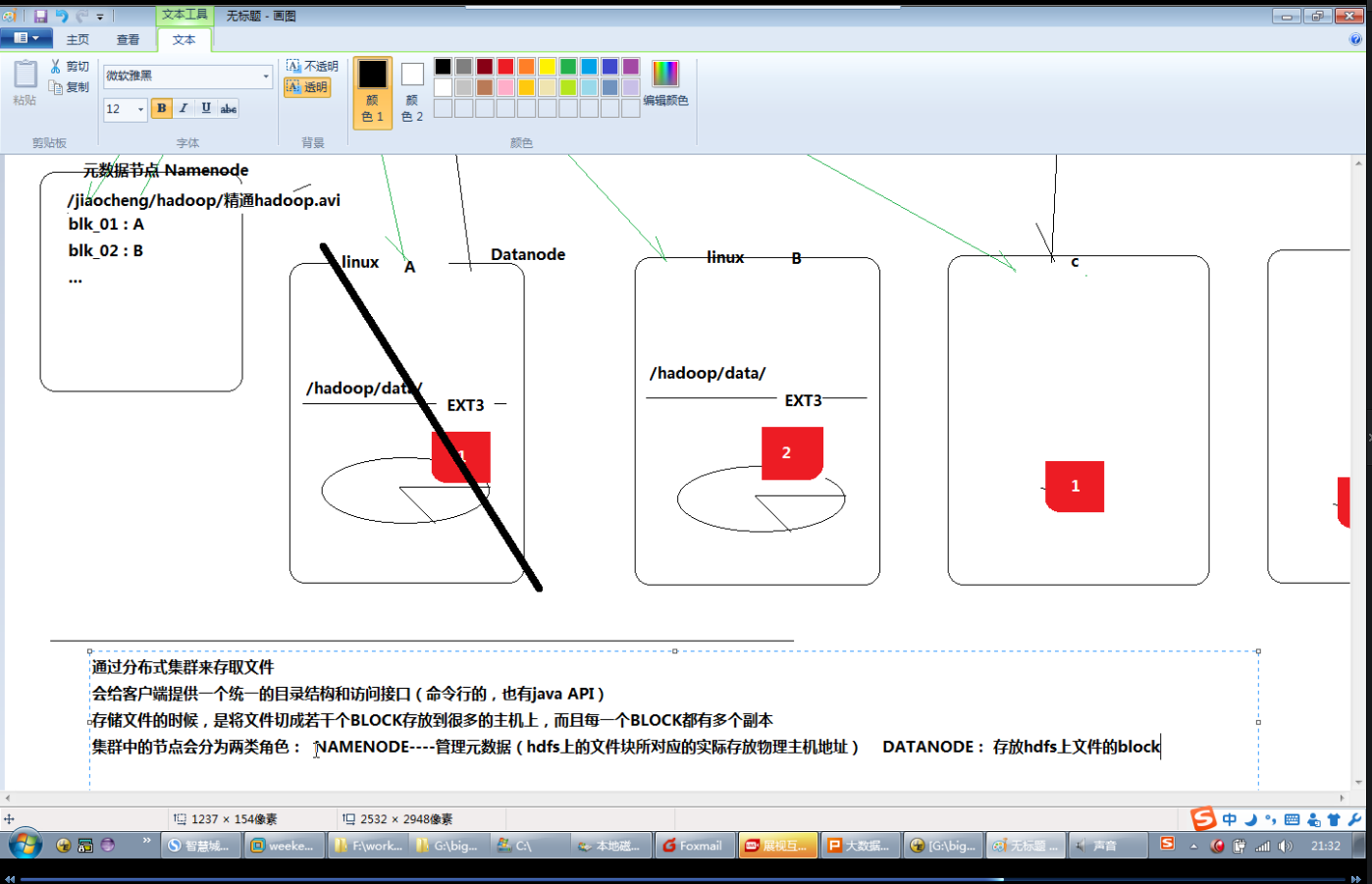
公共脚本(start-dfs,start-yarn均有使用): hadoop-deamons.sh(hdfs.sh,

yarn.sh)

## hdfs介绍

hdfs存入文件时,会将文件拆分成若干块,然后将每个块存放在不同的服务器磁盘上

hdfs加载数据时,hdfs会先从元数据节点(NameNode)中获取该数据存放的信息(数据块都在哪个磁盘上),然后根据获取的信息从不同的磁盘上读取数据



## 安装hadoop

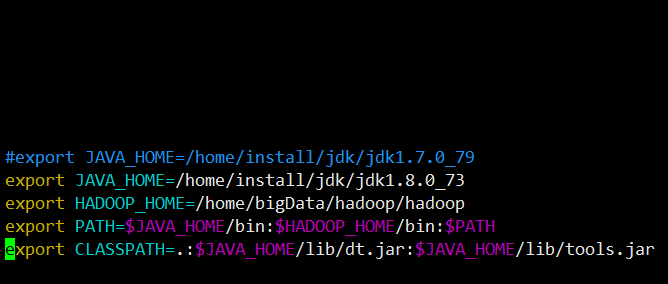
### 解压hadoop-2.7.2.tar.gz

tar -zxvf hadoop-2.7.2.tar.gz

### 配置环境变量

和jdk环境变量差不多

vim /etc/profile



刷新profile

source /etc/profile

## 配置hadoop

### standalone|local 本地模式

#### 没有守护进程,所有程序运行在同一JVM中,利于test和debug

#### 本地模式使用的是当前操作系统的文件夹无需启动(./start-all.sh)Hadoop,即可执行相关命令,系统中也没有hadoop的进程

例如hadoop fs -ls / 查看当前根目录下的文件夹

### pseudo distributed mode 伪分布模式

[配置文件位置${HADOOP\_HOME}/etc/Hadoop/\*-site.xml]

#### [core-site.xml]

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost</value>

</property>

</configuration>

#### [hdfs-site.xml]

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

#### [mapred-site.xml]该文件不存在可将mapred-site.xml.template文件复制一份

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

#### [yarn-site.xml]

<configuration>

<property>

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

<value>localhost</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

### 配置ssh免密登录

#### 安装ssh

centos7窗口化默认已经安装了ssh(客户端)和sshd(服务端)

#### 生成秘钥对

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa ##-t rsa代表加密类型使用rsa类型,-P ''代表无密登录,-f ~/.ssh/id\_rsa 代表密钥文件存放位置

#### 将生成的公钥添加到对方的认证密钥库中

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

#### centos配置免密登录失败情况总结

在centos中当配置完以上步骤后,还是不能免密登录时,需要修改一下内容

1:vim /etc/ssh/sshd\_config

将下面内容的注释打开

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

2: vim /etc/selinux/config

将SELINUX的值改为disabled

SELINUX=disabled

3:重启计算机

### 格式化文件系统(类似于新硬盘的格式化,格式化之后使用,如果之前有数据将之前数据清空)

hdfs namenode -format

### 启动关闭守护进程

start-dfs.sh ##启动hdfs

start-yarn.sh ##启动yarn

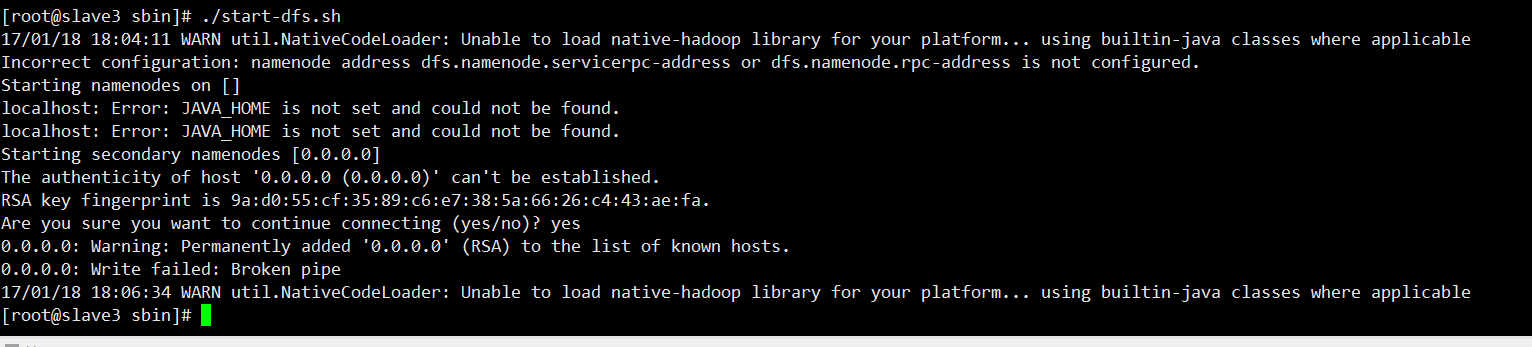
或者用start-all.sh(会启动以上两个脚本)

stop-dfs.sh ##关闭hdfs

stop -yarn.sh ##关闭yarn

或者用stop -all.sh(会启动以上两个脚本)

#### 第一次启动常见错误



出现该种情况时可能是新版本的hadoop无法识别${JAVA\_HOME}

需要在${HADOOP\_HOME}/etc/hadoop/hadoop-env.sh中设置JAVA-HOME为绝对路径,默认为相对路径

### 配置hadoop启动目录(可以根据不同目录中配置文件的内容决定启动哪种模式)

#### 默认方式

${HADOOP\_HOME}/etc/Hadoop/\*.xml

#### 通过启动参数指定启动配置目录

通过使用:启动命令 --config 启动模式的绝对路径(注意:此种方式启动后,以后在执行命令时也需要配合--config使用)

./start-all.sh --config /home/rkg/hadoop/hadoop-2.7.2/etc/hadoop

#### 通过环境变量指定启动模式

在/etc/profile中添加

export HADOOP\_CONF\_DIR=/home/rkg/hadoop/hadoop-2.7.2/etc/Hadoop

或者

export HADOOP\_CONF\_DIR=/home/rkg/hadoop/hadoop-2.7.2/etc/Hadoop-pseudo(伪分布式)

#### 通过软连接指向需要启动的模式,Hadoop默认使用${HADOOP\_HOME}/etc/hadoop/文件夹下的配置文件

ln -s hadoop\_pseudo hadoop

### 操作用户目录(该目录是namenode的目录用于存放文件路径),在伪分布式模式下必须先启动hadoop,才能执行相关命令

#### 创建目录

hadoop fs -mkdir -p /user/rkg

该命令会在hadoop的namenode根节点创建一个user文件夹,并在user文件夹下创建一个rkg文件夹

hadoop fs -mkdir /user/rkg

该命令会在hadoop的namenode根节点的user文件夹下创建一个rkg文件夹(user文件夹必须先存在,负责报错)

#### 查看目录

hadoop fs -ls / 查看当前根目录下的文件夹等同于 hadoop fs -ls -R hdfs://localhost/

hadoop fs -ls file://

hadoop fs -ls -R / 查看当前根目录下的文件夹及子文件夹

hadoop fs -lsr / 查看当前根目录下的文件夹及子文件夹(推荐使用上面那种方式查看)

hadoop fs -ls 默认查看当前用户下的目录(/user/${user},例如/user/root)

#### 删除目录

hadoop fs -rm -r /user/

#### 修改目录名称

hadoop fs -mv /user/rkg/123 /user/rkg/456

### 将本地磁盘上的文件拷贝到hadoop数据节点上

touch rkg.txt ##创建rkg.txt文件

echo “hello Hadoop ” >> rkg.txt ##将hello Hadoop写到rkg.txt文件中

和vi rkg.txt操作一样

hadoop fs -put rkg.txt . ##将创建的rkg.txt拷贝到.(点代表当前用户目录/user/root)目录下等同于hadoop fs -put rkg.txt /user/root

可以进入

/tmp/hadoop-root/dfs/data/current/BP-1459891713-10.0.6.37-1484733388520/current/finalized/subdir0/subdir0

目录下查看里面的内容,直接cat blk\_1073741825,可以发现里面的内容和刚才创建的rkg.txt内容一致

### 搭建完全分布式hadoop集群

搭建完全分布式hadoop集群最好有五台或者五台以上的机器,一台namenode,一台secondarynamenode,三台datanode

#### 查看主机名称

hostname

#### 修改主机名称(hadoop主机名称不能包含特殊字符下划线点灯)

##### centos7修改方法

hostnamectl set-hostname 主机名

##### centos6等版本修改方法

vim /etc/sysconfig/network

修改下面的hostname属性值即可

NETWORKING=yes

HOSTNAME=elasticsearch1

#### 将修改后的hostname与主机进行关联

vim /etc/hosts

添加一下内容

10.0.6.34 主机名称1

10.0.6.36 主机名称2

10.0.6.37 主机名称3

10.0.6.38 主机名称4

10.0.6.39 主机名称5

然后重启机器即可

重启后测试ping 主机名称1 发现可以ping通

#### 修改配置文件

下面内容可以直接点击打开



hadoop各节点介绍

NameNode:存放目录,整个文件系统的源数据信息

secondaryNameNode:NameNode的备份

DataNode:存放真实数据,以block形式存在,每个block大小为128MB(可以修改,默认128)

rpc:remote procedure call远程过程调用

##### 修改core-site.xml

将fs.defaultFS=hdfs://localhost

改为fs.defaultFS=hdfs://hadoop\_1:8020(rpc端口,默认是8020,默认可以不写)

<property>

<name>fs.defaultFS</name>

<value>hdfs:// hdfs://hadoop\_1</value>

</property>

##### 修改hdfs-site.xml

将最小副本数dfs.replication改为4(我有四台datanode节点,有几台datanode节点就改为几)

dfs.namenode.secondary.http-address= hadoop2:50090##配置secondarynamenode

<property>

<name>dfs.replication</name>

<value>4</value>

</property>

<property>

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

<value>hadoop2:50090</value>

</property>

##### 修改yarn-site.xml

修改yarn.resourcemanager.hostname= localhost为

yarn.resourcemanager.hostname=namenode节点服务器名称(hadoop\_1)

<configuration>

<property>

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

<value>localhost</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

##### 修改slaves

在其中添加datanode节点服务器名称

每个服务器为一个行

hadoop3

hadoop4

hadoop5

hadoop6

通过scp将修改后的文件发送到其他节点上

scp –r hadoop\_cluster root@hadoop\_2:/home/bigData/hadoop/hadoop/etc

#### 格式化文件系统

hdfs --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster namenode -format

#### 启动hdfs

start-dfs.sh --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster

#### 启动yarn

start-yarn.sh --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster

start-all.sh --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster

stop-all.sh --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster

## 常见文件配置

### core-site.xml

#### 设置hadoop临时文件目录

hadoop.tmp.dir= file:/usr/local/hadoop/tmp

### hdfs-site.xml

dfs.blocksize=64m(默认不写单位是字节)

### yarn-site.xml

## hdfs常用操作

hdfs设计目的

适用于:超大文件,streaming(流)访问,商业硬件

不适用:低延迟访问,大量小文件,多用户写入,多次修改

### 创建目录

hadoop --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster fs -mkdir -p /user/rkg

该命令会在hadoop的namenode根节点创建一个user文件夹,并在user文件夹下创建一个rkg文件夹

hadoop fs -mkdir /user/rkg

该命令会在hadoop的namenode根节点的user文件夹下创建一个rkg文件夹(user文件夹必须先存在,负责报错)

### 查看目录

hadoop --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster …

hadoop fs -ls / 查看当前根目录下的文件夹等同于 hadoop fs -ls -R hdfs://localhost/

hadoop fs -ls file://

hadoop fs -ls -R / 查看当前根目录下的文件夹及子文件夹

hadoop fs -lsr / 查看当前根目录下的文件夹及子文件夹(推荐使用上面那种方式查看)

hadoop fs -ls 默认查看当前用户下的目录(/user/${user},例如/user/root)

### 删除目录

hadoop --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster fs -rm -r /user/

### 修改目录名称

hadoop --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster fs -mv /user/rkg/123 /user/rkg/456

### 将本地文件添加到hdfs中

hadoop --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster fs -put rkg.txt . ##将创建的rkg.txt拷贝到.(点代表当前用户目录/user/root)目录下等同于hadoop fs -put rkg.txt /user/root

可以进入

/tmp/hadoop-root/dfs/data/current/BP-1459891713-10.0.6.37-1484733388520/current/finalized/subdir0/subdir0

目录下查看里面的内容,直接cat blk\_1073741825,可以发现里面的内容和刚才创建的rkg.txt内容一致

### 获取hdfs中的文件

hadoop --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster fs –get /user/rkg/data/rkg.txt 123.txt

将hdfs中/user/rkg/data目录里的rkg.txt下载到当前目录,并该名称为123.txt

### 检查文件状态

hdfs --config /home/bigData/hadoop/hadoop-2.7.2/etc/hadoop\_cluster fsck /user/rkg/data/mysql-5.7.12-1.el7.x86\_64.rpm-bundle.tar

## 常用脚本

### 远程复制脚本

#!/bin/bash

if [ $# -lt 1 ]; then

echo '对不起请输入需要scp的目录或文件'

exit

fi

##获取第一个参数

arg1=$1;

##获取当前用户名称

cuser=`whoami`

##获取输入的文件或者文件夹名称

fname=`basename $arg1`

##获取输入的文件或者文件夹所在的目录

dir=`dirname $arg1`

if [ '$dir'='.' ]; then

dir=`pwd`

fi

for ((i=3;i<=5;i=i+1))

do

echo 开始scp $fname到hadoop$i服务器

if [ -d $arg1 ]; then

scp -r $arg1 $cuser@hadoop$i:$dir

## echo -r $arg1 $cuser@hadoop$i:$dir

else

scp $arg1 $cuser@hadoop$i:$dir

## echo $arg1 $cuser@hadoop$i:$dir

fi

done

### 远程删除脚本

#!/bin/bash

if [ $# -lt 1 ]; then

echo '对不起请输入需要删除的目录或文件'

exit

fi

##获取第一个参数

arg1=$1;

##获取当前用户名称

cuser=`whoami`

##获取输入的文件或者文件夹名称

fname=`basename $arg1`

##获取输入的文件或者文件夹所在的目录

dir=`dirname $arg1`

if [ '$dir'='.' ]; then

dir=`pwd`

fi

for ((i=3;i<=5;i=i+1))

do

echo 开始从hadoop$i服务器删除$fname

ssh hadoop$i rm -rf $dir/$fname

echo 从hadoop$i服务器删除$fname成功

done

### 远程查看脚本

#!/bin/bash

if [ $# -lt 1 ]; then

echo '对不起请输入需要查看的目录或文件'

exit

fi

##获取第一个参数

arg1=$1;

##获取当前用户名称

cuser=`whoami`

##获取输入的文件或者文件夹名称

fname=`basename $arg1`

##获取输入的文件或者文件夹所在的目录

dir=`dirname $arg1`

if [ '$dir'='.' ]; then

dir=`pwd`

fi

for ((i=3;i<=5;i=i+1))

do

echo hadoop$i的内容为:

ssh hadoop$i ls -lh $dir/$fname

done

## 查看帮助

Hadoop ##直接回车查看hadoop的帮助

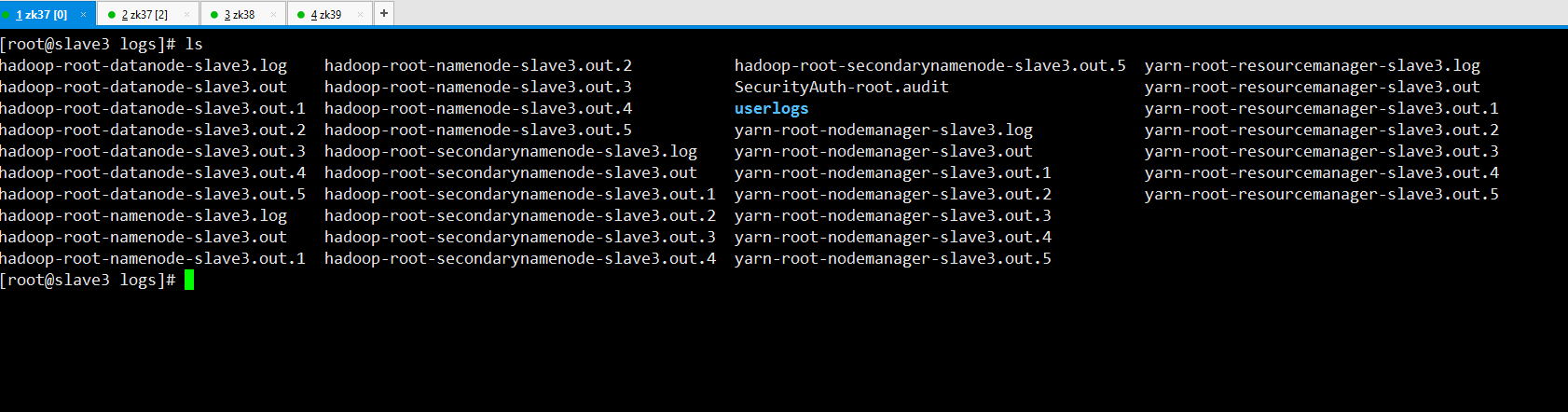
hadoop fs  ##直接回车查看fs的帮助

Hadoop fs –help put ##查看put的帮助信息

## 查看日志文件

### 通过log文件查看

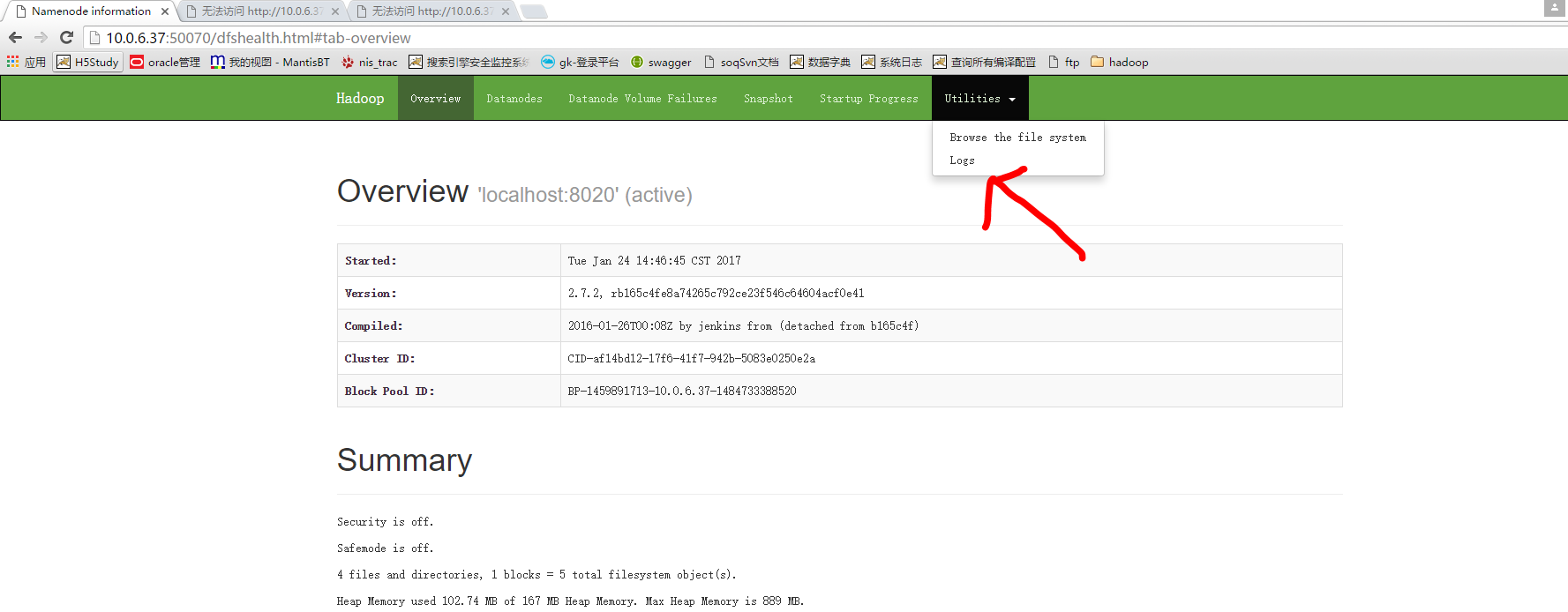
可进入${JAVA\_HOME}/logs文件夹下查看相关log



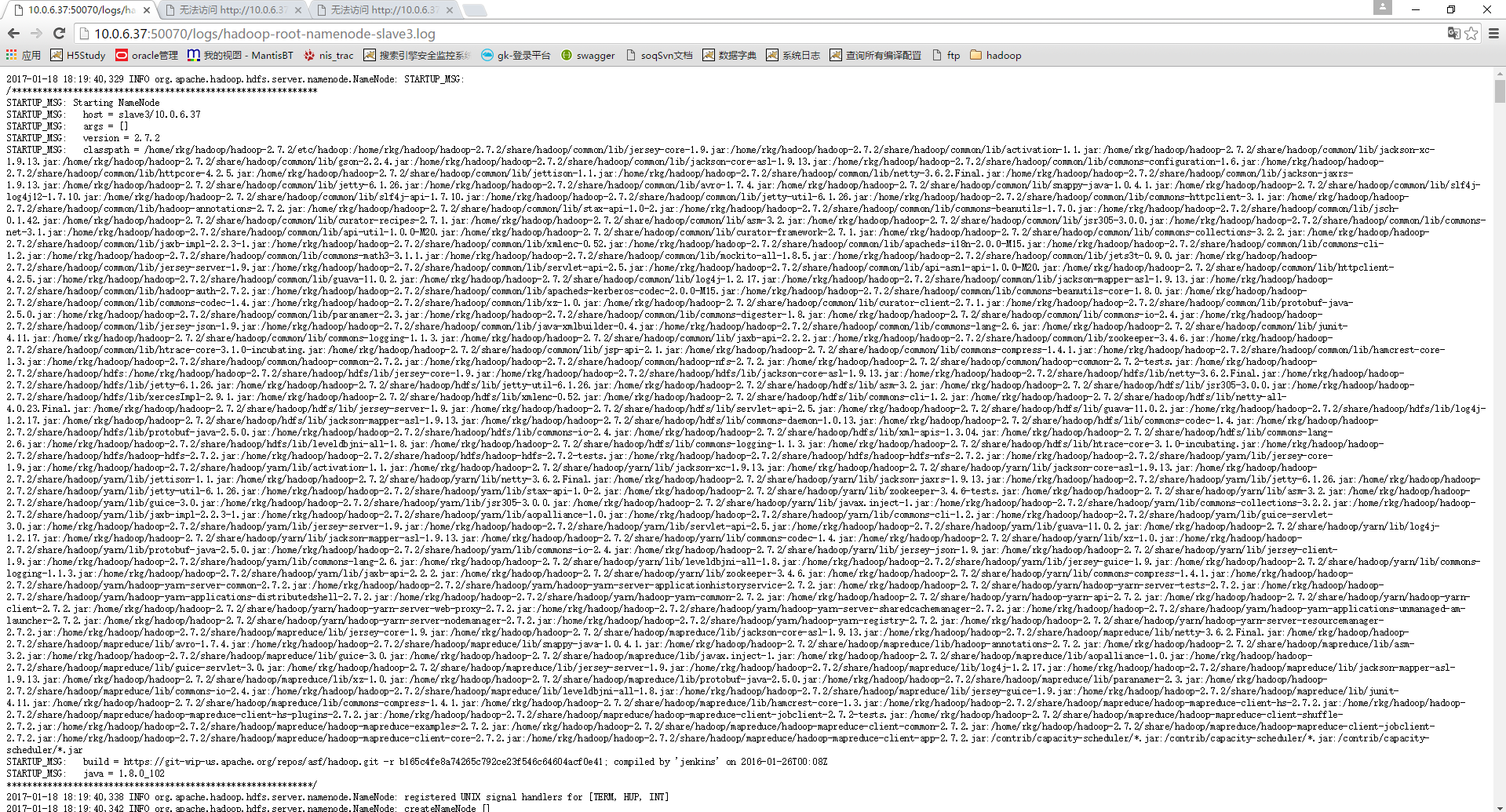
### 通过ui界面查看日志或者集群情况

<http://10.0.6.37:50070/> 查看namenode

<http://127.0.0.1:8088/> resourceManager查看集群(cluster,该url默认只能在本地访问,其他机器默认无法访问,可以通过netstat –ntlp命令查看local address列发现其内容不是0.0.0.0:8088,而是127.0.0.1:8088)

[http://127.0.0.1: /](http://127.0.0.1:8088/) 





## webui界面

namenode节点地址

## 查看端口占用

lsof -i tcp:2181

[root@slave4 bin]# lsof -i tcp:2181

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

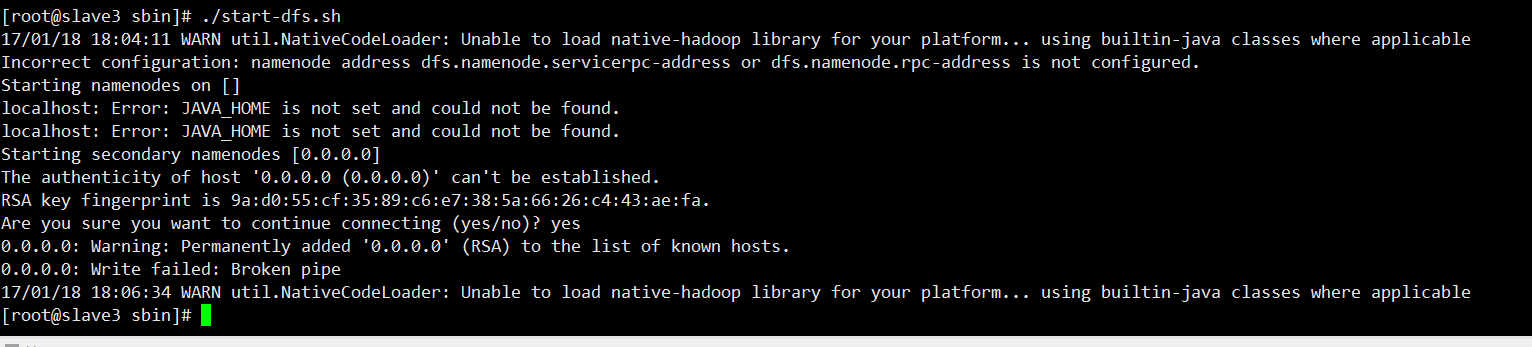
java 1708 root 23u IPv6 13505 0t0 TCP \*:eforward (LISTEN)

[root@slave4 bin]

netstat –ntlp列出所有端口

## Hadoop常见错误

### 第一次启动常见错误(./start-all.sh)

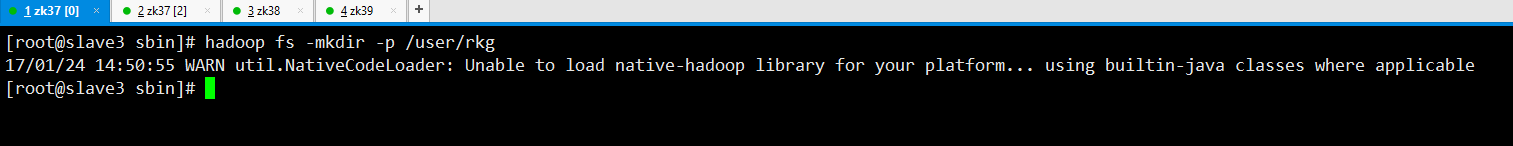


出现该种情况时可能是新版本的hadoop无法识别${JAVA\_HOME}

需要在${HADOOP\_HOME}/etc/Hadoop/Hadoop-env.sh中设置JAVA-HOME为绝对路径,默认为相对路径

### 创建用户目录出现如下警告时

Hadoop默认使用32位操作系统编译,当前系统是64位版本,如果运行hadoop需要将${JAVA\_HOME}/lib/目录下的文件替换为64位的版本



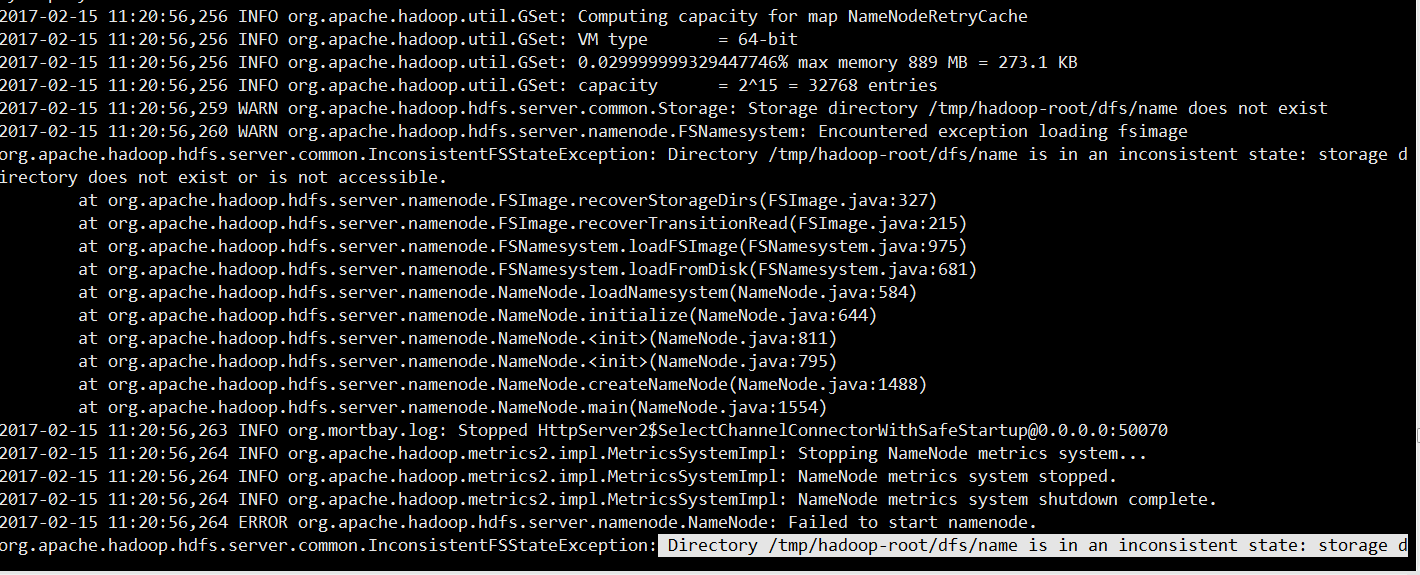
### 启动hadoop出现错误-->未格式化文件系统

伪分布式模式下启动hadoop发现namenode节点没有启动成功,查看日志发现如下错误,未创建dfs文件夹

Directory /tmp/hadoop-root/dfs/name is in an inconsistent state: storage directory does not exist or is not accessible.

解决办法:格式化文件系统即可

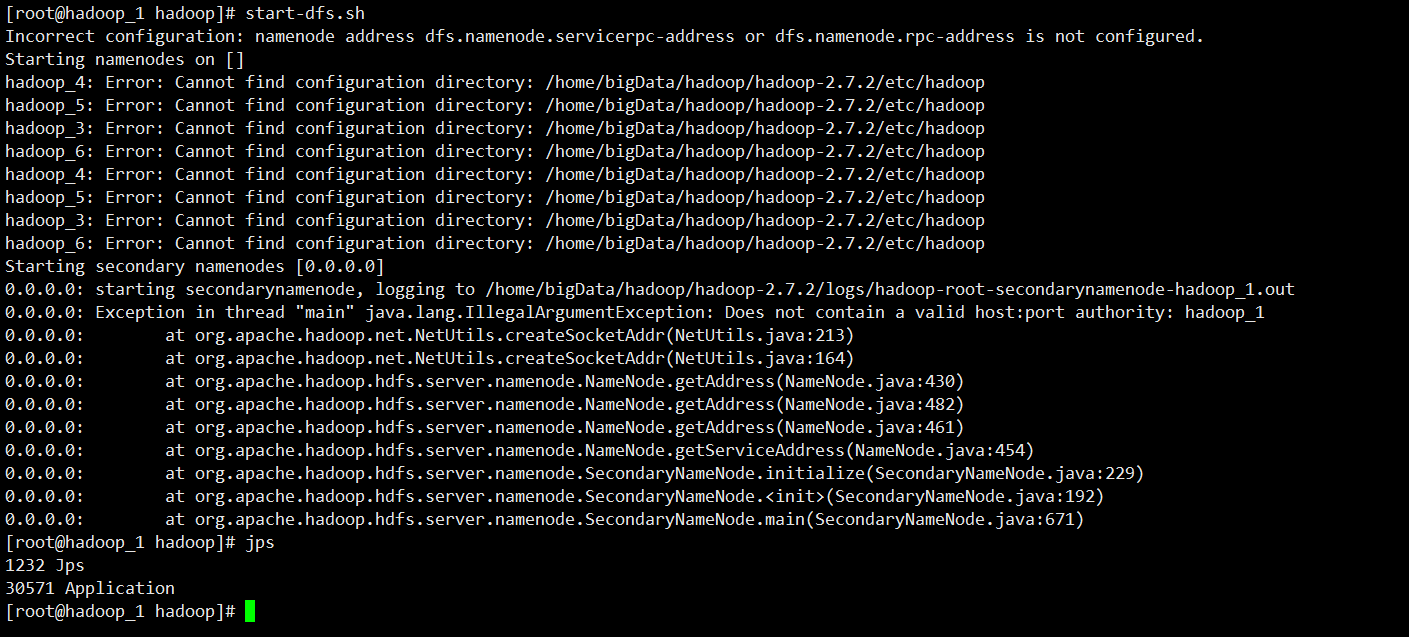
hdfs namenode -format



### 完全分布式启动hadoop报错

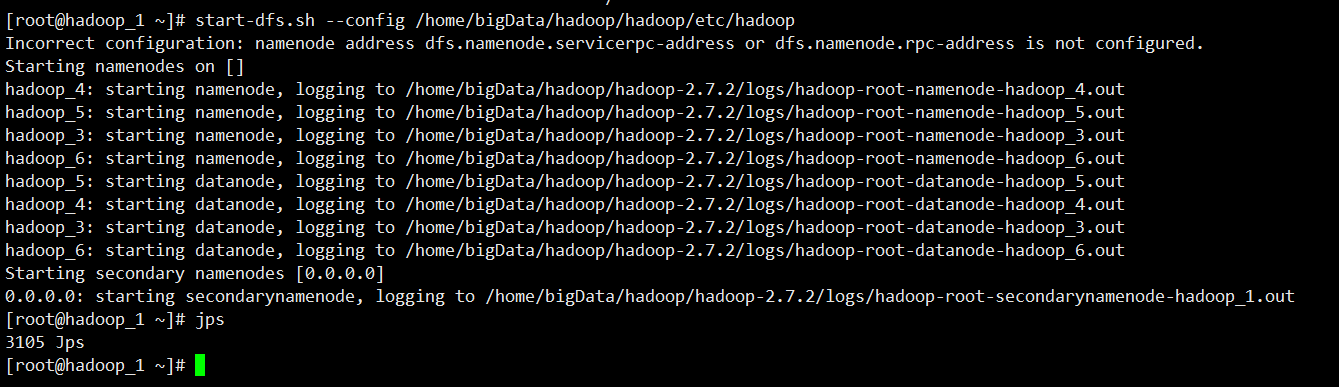
可能是已经启动过hadoop,shutdown重新启动就好了或者是重启后没有格式化文件系统

修改主机名称将hadoop\_1修改为hadoop即可

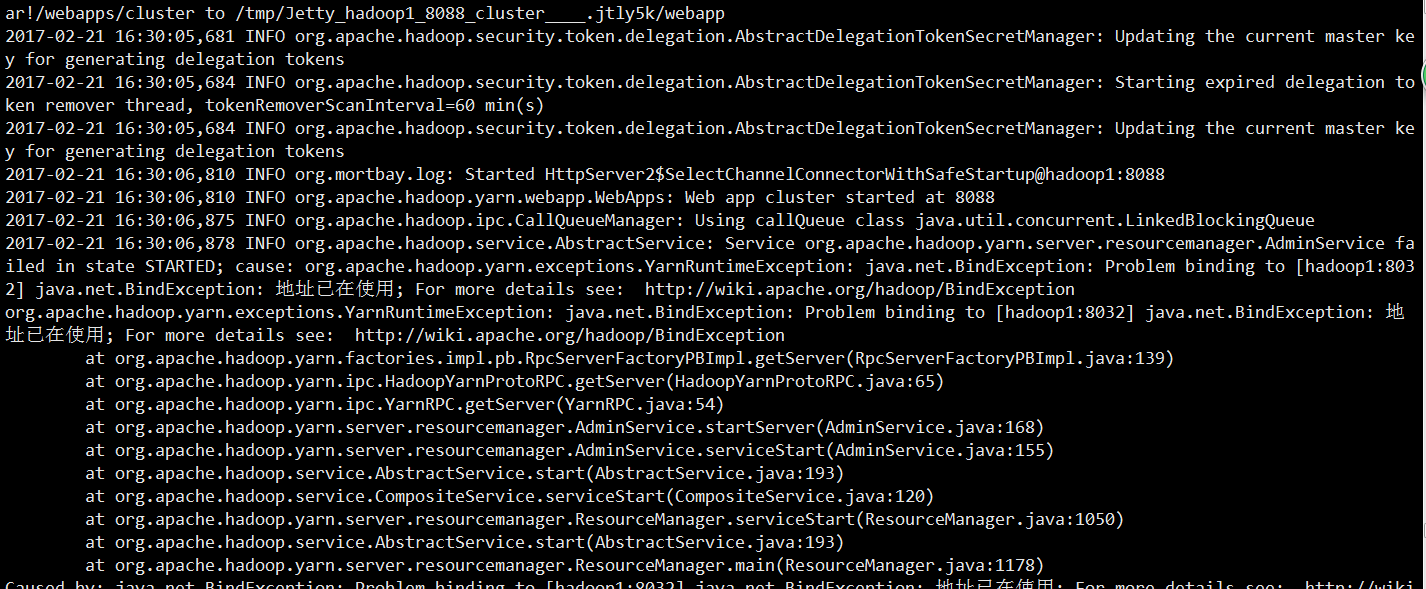


### 完全分布式启动hadoop不报错但是jps查询没有进程

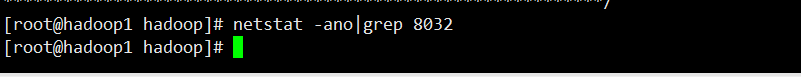
修改主机名称将hadoop\_1修改为hadoop即可



### 启动yarn报错,端口占用



查看端口占用情况,未发现被占用

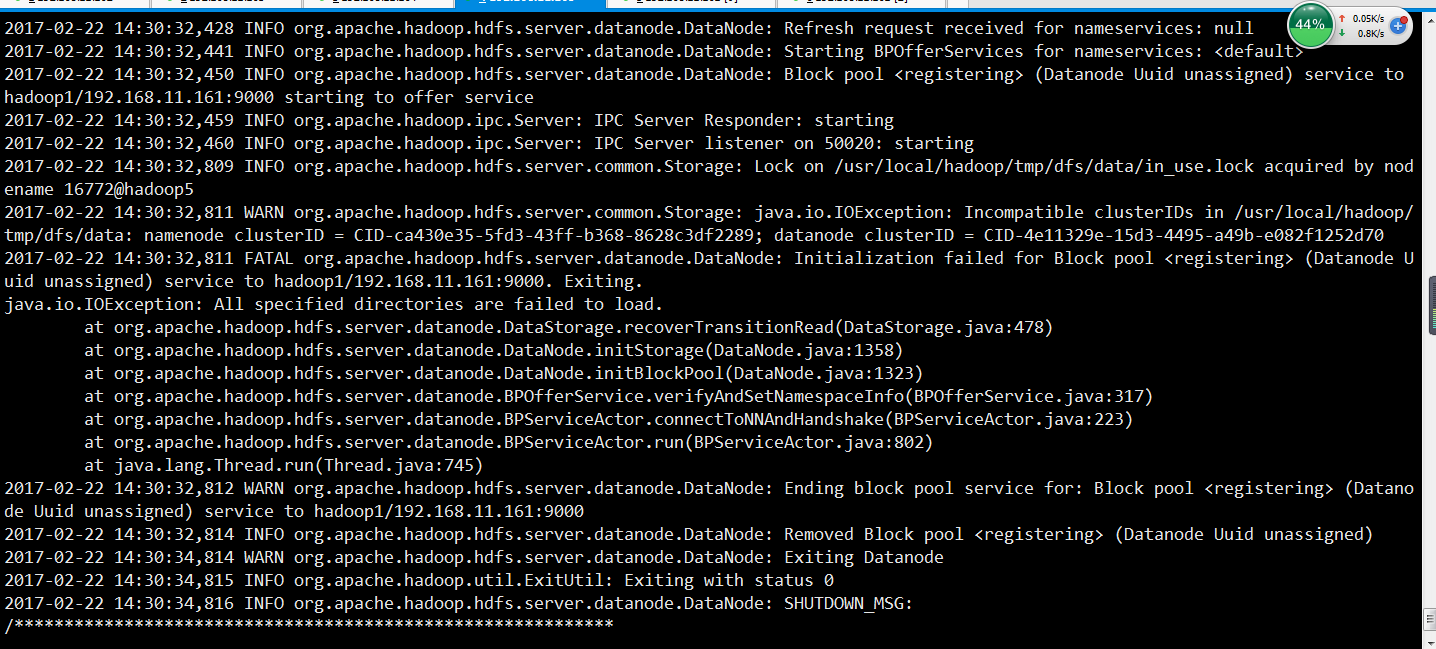


经过排查发现yarn-site.xml中

yarn.resourcemanager.address= hadoop1:8032

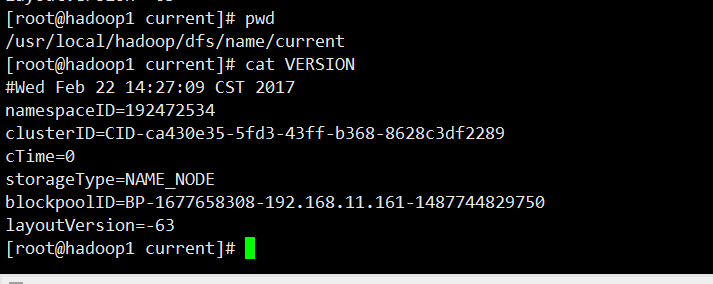
yarn.resourcemanager.admin.address= hadoop1:8032

### 启动datanode错误All specified directories are failed to load

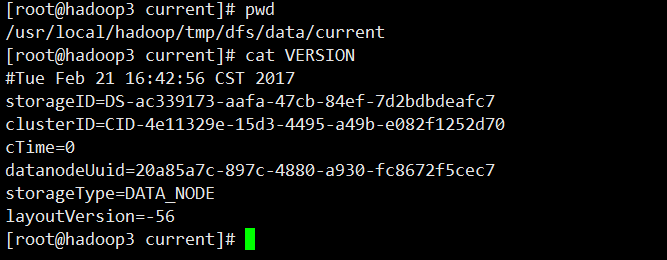


该错误是由于datanode的VERSION文件中的clusterID与namenode的VERSION文件中的clusterID不一致,将datanode中的clusterID改为与namenode一致即可

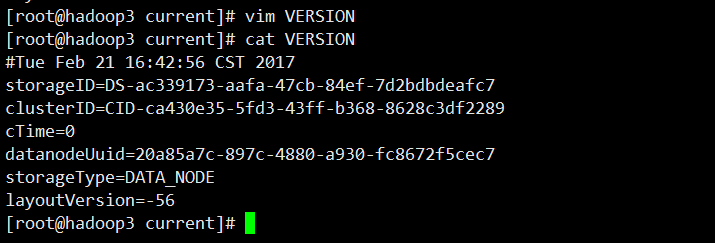
namenode



datanode



修改后



## 注意

### hadoop各集群机器名称不能包含特殊字符,例如下划线(\_)点(.)等,否则无法执行某些命令,例如完全分布式启动start-dfs.sh,会报如下错误Does not contain a valid host:port authority:hadoop\_1

### 在名称节点启动所有数据节点

hadoop-daemons.sh start datanode

### 在名称节点启动名称节点

hadoop-daemon.sh start namenode

### 启动辅助名称节点

hadoop-daemon.sh start secondarynamenode

### 在某一数据节点启动该数据节点(需要ssh进入需要启动的数据节点中)

hadoop-daemon.sh start datanode

## 查看hadoop源码

### 从源码中提取所有配置文件

core-site.xml -->hadoop-common-2.7.2.jar-->core-default.xml

hdfs-site.xml -->hadoop-hdfs-2.7.2.jar-->hdfs-default.xml

mapred-site.xml-->hadoop-mapreduce-client-core-2.7.2.jar-->mapred-default.xml

yarn-site.xml -->hadoop-yarn-common-2.7.2.jar-->yarn-default.xml

### 脚本分析

#### start-all.sh

bin=`dirname "${BASH\_SOURCE-$0}"`##获取输入的脚本名称所在的文件夹

bin=`cd "$bin"; pwd`##进入文件夹并显示命令所在的目录

DEFAULT\_LIBEXEC\_DIR="$bin"/../libexec##定义变量,值为libexec目录路径

HADOOP\_LIBEXEC\_DIR=${HADOOP\_LIBEXEC\_DIR:-$DEFAULT\_LIBEXEC\_DIR}

. $HADOOP\_LIBEXEC\_DIR/hadoop-config.sh##执行libexec里面的hadoop-config.sh

##先声明以上变量然后执行libexec/hadoop-config.sh

# start hdfs daemons if hdfs is present

if [ -f "${HADOOP\_HDFS\_HOME}"/sbin/start-dfs.sh ]; then##判断start-dfs.sh是否存在

"${HADOOP\_HDFS\_HOME}"/sbin/start-dfs.sh --config $HADOOP\_CONF\_DIR

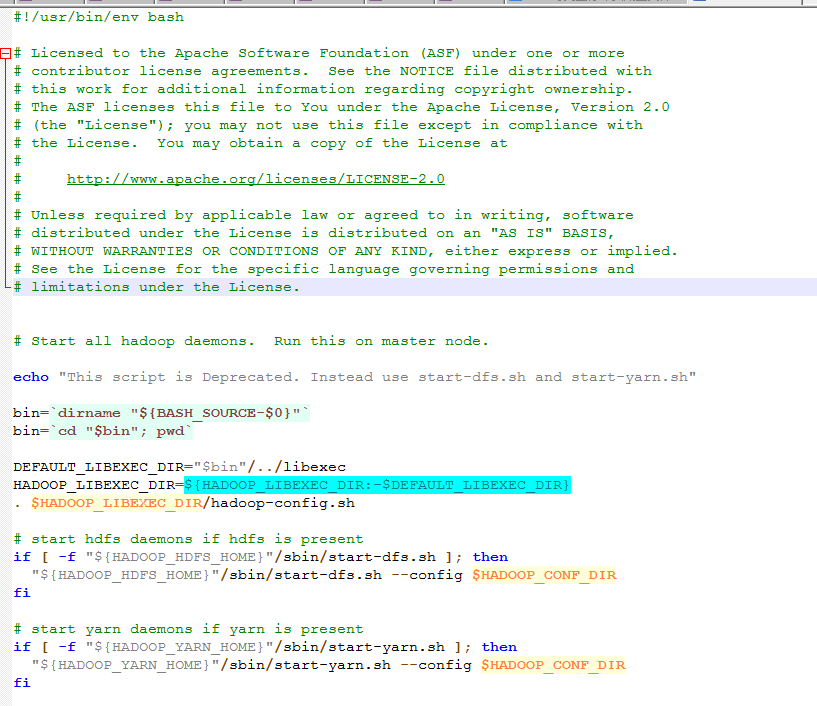
fi

# start yarn daemons if yarn is present

if [ -f "${HADOOP\_YARN\_HOME}"/sbin/start-yarn.sh ]; then

"${HADOOP\_YARN\_HOME}"/sbin/start-yarn.sh --config $HADOOP\_CONF\_DIR

fi



#### start-dfs.sh

定义变量

执行hdfs-config.sh,HADOOP\_CONF\_DIR=…##--config参数

NAMENODE={HADOOP\_HOME}/bin/hdfs getconf -namenodes//提取名称节点主机名

"$HADOOP\_PREFIX/sbin/hadoop-daemons.sh" \

--config "$HADOOP\_CONF\_DIR" \

--hostnames "$NAMENODES" \

--script "$bin/hdfs" start namenode $nameStartOpt

"$HADOOP\_PREFIX/sbin/hadoop-daemons.sh" \

--config "$HADOOP\_CONF\_DIR" \

--script "$bin/hdfs" start datanode $dataStartOpt

"$HADOOP\_PREFIX/sbin/hadoop-daemons.sh" \

--config "$HADOOP\_CONF\_DIR" \

--hostnames "$SECONDARY\_NAMENODES" \

--script "$bin/hdfs" start secondarynamenode

#### start-yarn.sh