**hadoop-2.7.3+zookeeper-3.4.8+hadoop-2.7.3分布式环境搭建整理**

# 一.准备环境

## 1.1. 安装包

1）准备4台PC

2）安装配置Linux系统：CentOS-6.5

下载地址：

<http://vault.centos.org/6.5/isos/x86_64/CentOS-6.5-x86_64-bin-DVD1.iso>

<http://vault.centos.org/6.5/isos/x86_64/CentOS-6.5-x86_64-bin-DVD2.iso>

3）安装配置Java环境：jdk-8u101-linux-x64.tar.gz

下载地址：<http://download.oracle.com/otn-pub/java/jdk/8u101-b13/jdk-8u101-linux-x64.tar.gz>

4）安装配置Hadoop：hadoop-2.7.3.tar.gz

下载地址：<http://mirrors.hust.edu.cn/apache/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

4）安装配置zookeeper：zookeeper-3.4.8.tar.gz

下载地址：http://mirrors.hust.edu.cn/apache/zookeeper/zookeeper-3.4.8/zookeeper-3.4.8.tar.gz

5）安装配置Hbase：hbase-1.2.2-bin.tar.gz

下载地址：http://www.apache.org/dist/hbase/stable/hbase-1.2.2-bin.tar.gz

## 1.2. 网络配置

|  |  |
| --- | --- |
| 主机名 | IP |
| master | 192.168.10.198 |
| slaver1 | 192.168.10.170 |
| slaver2 | 192.168.10.143 |
| slaver3 | 192.168.10.168 |

## 1.3. 常用命令

# service iptables start  #运行一个服务

# service iptables stop #停止一个服务

# service iptables restart #重启一个服务

# service iptables status  #显示一个服务（无论运行与否）的状态

# chkconfig iptables on  #在开机时启用一个服务

# chkconfig iptables off  #在开机时禁用一个服务

# reboot #重启主机

# shutdown -h now  #立即关机

# source /etc/profile  #配置文件修改立即生效

 # vi /etc/sysconfig/iptables #编辑防火墙配置文件

# yum install net-tools

# 二.安装配置CentOS

## 2.1安装CentOS

1）选择启动盘CentOS-6.5-x86\_64-bin-DVD1.iso，启动安装

2）选择Install CentOS 6.5，回车，继续安装

3）选择语言，默认是English，学习可以选择中文，正时环境选择English

4）配置网络和主机名，主机名：master，网络选择开启，配置手动的IPV4

5）选择安装位置；在分区处选择手动配置；选择标准分区，点击这里自动创建他们，点击完成，收受更改

6）修改root密码，密码：123456

7）重启，安装完毕。

## 2.2配置IP (root身份)

### 2.2.1检查IP

# ip addr

或

# ip link

### 2.2.2配置IP和网管

#cd /etc/sysconfig/network-scripts #进入网络配置文件目录

# find ifcfg-\*  #查到网卡配置文件，例如ifcfg-em1

# vi ifcfg-eth0  #编辑网卡配置文件

或

# vi /etc/sysconfig/network-scripts/ifcfg-eth0 #编辑网卡配置文件

配置内容：

DEVICE=eth0 #网卡名称

BOOTPROTO=static  #静态IP配置为static，动态配置为dhcp

ONBOOT=yes #开机启动

HWADDR=08:00:27:6E:57:12 #mac地址

IPADDR=192.168.10.198  #IP地址

NETMASK=255.255.255.0  #子网掩码

GATEWAY=192.168.10.1 #网关

DNS1=202.96.134.133 #主DNS

DNS2=202.96.134.188 #备DNS

# service network restart  #重启网络

(如果提示错误，则参考《CentOS Linux解决Device eth0 does not seem to be present.doc》)

### 2.2.3配置hosts

# vi /etc/hosts

编辑内容：

192.168.10.198 master

192.168.10.170 slaver1

192.168.10.143 slaver2

192.168.10.168 slaver3

### 2.2.3修改CentOS5.5主机名称

安装的CentOS5.5主机名称默认是 localhost.localdomain 不太好记，需要修改。

要求不重新启动机器，在终端场景下修改CentOS5.5主机名的修改步骤如下:

1. 修改CentOS主机名称配置文件 /etc/sysconfig/network

# vi /etc/sysconfig/network

修改HOSTNAME配置为想要的名称,例如:

NETWORKING=yes

NETWORKING\_IPV6=no

HOSTNAME=master

2. 修改 /etc/hosts 配置文件

用户在进行网络连接时，首先查找该文件，寻找对应主机名（或域名）对应的IP地址。

修改之后的用户名对应的IP地址为127.0.0.1，修改 /etc/hosts 文件如下:

# vi /etc/hosts

# Do not remove the following line, or various programs

# that require network functionality will fail.

127.0.0.1 WebServer localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6

3. 使用 hostname 命令修改主机名

完成以上两步，你会发现实际的主机名称并没有发生变化，因为修改的配置文件会在下次重启的时候才能加载。此时使用hostname命令来修改主机名称:

# hostname

# hostname master

# hostname

## 2.3关闭防火墙

# service iptables status  #检查防火墙状态

# service iptables stop  #关闭防火墙

# chkconfig iptables off  #禁止开机启动防火墙

## 2.4时间同步

# yum install -y ntp  #安装ntp服务

# ntpdate cn.pool.ntp.org  #同步网络时间

## 2.5安装配置jdk

### 2.5.1卸载自带jdk

安装好的CentOS会自带OpenJdk,用命令java -version ，会有下面的信息：

    java version"1.6.0"  
   OpenJDK Runtime Environment (build 1.6.0-b09)  
   OpenJDK 64-Bit Server VM (build 1.6.0-b09, mixedmode)

最好还是先卸载掉openjdk,再安装sun公司的jdk.

先查看rpm -qa | grep java

显示如下信息：

     java-1.4.2-gcj-compat-1.4.2.0-40jpp.115  
     java-1.6.0-openjdk-1.6.0.0-1.7.b09.el5

卸载：

     rpm -e –nodeps java-1.4.2-gcj-compat-1.4.2.0-40jpp.115  
     rpm -e –nodeps java-1.6.0-openjdk-1.6.0.0-1.7.b09.el5

还有一些其他的命令

    rpm -qa | grep gcj

    rpm -qa | grep jdk

如果出现找不到openjdksource的话，那么还可以这样卸载

     yum -y remove javajava-1.4.2-gcj-compat-1.4.2.0-40jpp.115  
     yum -y remove javajava-1.6.0-openjdk-1.6.0.0-1.7.b09.el5

### 2.5.2安装jdk

上传jdk-8u101-linux-x64.tar.gz 安装包到opt目录

# tar -zxvf jdk-8u101-linux-x64.tar.gz

### 2.5.3各个主机之间复制jdk (可以先处理后面的免密问题再传输，以免不停需要输入密码)

# scp -r /opt/jdk1.8.0\_101 slaver1:/opt/

# scp -r /opt/jdk1.8.0\_101 slaver2:/opt/

# scp -r /opt/jdk1.8.0\_101 slaver3:/opt/

### 2.5.4各个主机配置jdk环境变量

# vi /etc/profile

编辑内容

export JAVA\_HOME=/opt/jdk1.8.0\_101

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

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

# source /etc/profile  #使配置文件生效

# java -version  #查看java版本

## 2.6配置ssh无密钥访问

分别在各个主机上检查ssh服务状态：

# service sshd status  #检查ssh服务状态

# yum install openssh-server openssh-clients  #安装ssh服务，如果已安装，则不用执行该步骤

# service sshd start  #启动ssh服务，如果已安装，则不用执行该步骤

分别在各个主机上生成密钥

# ssh-keygen -t rsa  #生成密钥

在slaver1上

# cp ~/.ssh/id\_rsa.pub ~/.ssh/slaver1.id\_rsa.pub

# scp ~/.ssh/slaver1.id\_rsa.pub master:~/.ssh

在slaver2上

# cp ~/.ssh/id\_rsa.pub ~/.ssh/slaver2.id\_rsa.pub

# scp ~/.ssh/slaver2.id\_rsa.pub master:~/.ssh

 在slaver3上

# cp ~/.ssh/id\_rsa.pub ~/.ssh/slaver3.id\_rsa.pub

# scp ~/.ssh/slaver3.id\_rsa.pub master:~/.ssh

 在master上

# cd ~/.ssh

# cat id\_rsa.pub >> authorized\_keys

# cat slaver1.id\_rsa.pub >>authorized\_keys

# cat slaver2.id\_rsa.pub >>authorized\_keys

# cat slaver3.id\_rsa.pub >>authorized\_keys

# scp authorized\_keys slaver1:~/.ssh

# scp authorized\_keys slaver2:~/.ssh

# scp authorized\_keys slaver3:~/.ssh

# 三.安装配置hadoop

## 3.1安装hadoop

上传hadoop-2.7.3.tar.gz安装包到opt根目录

# tar -zxvf hadoop-2.7.3.tar.gz

# mkdir ./hadoop-2.7.3/tmp

# mkdir ./hadoop-2.7.3/logs

# mkdir ./hadoop-2.7.3/hdf

# mkdir ./hadoop-2.7.3/hdf/data

# mkdir ./hadoop-2.7.3/hdf/name

### 3.1.1在hadoop中配置hadoop-env.sh文件

edit the file etc/hadoop/hadoop-env.sh todefine some parameters as follows:

  # vi ./hadoop-2.7.3/etc/hadoop/hadoop-env.sh

  export JAVA\_HOME=**/opt/jdk1.8.0\_101**

### 3.1.2修改yarn-env.sh

# vi ./hadoop-2.7.3/etc/hadoop/yarn-env.sh

#export JAVA\_HOME=/home/y/libexec/jdk1.7.0/

export JAVA\_HOME=**/opt/jdk1.8.0\_101**

### 3.1.3修改slaves

# vi ./hadoop-2.7.3/etc/hadoop/slaves

配置内容：

删除：localhost

添加：

slaver1

slaver2

slaver3

### 3.1.4修改core-site.xml

# vi ./hadoop-2.7.3/etc/hadoop/core-site.xml

配置内容：

<configuration>

  <property>

    <name>fs.default.name</name>

    <value>hdfs://master:9000</value>

  </property>

  <property>

    <name>hadoop.tmp.dir</name>

    <value>file:/opt/hadoop-2.7.3/tmp</value>

  </property>

</configuration>

### 3.1.5修改hdfs-site.xml

# vi ./hadoop-2.7.3/etc/hadoop/hdfs-site.xml

配置内容：

<configuration>

  <property>

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

    <value>/opt/hadoop-2.7.3/hdf/data</value>

    <final>true</final>

  </property>

  <property>

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

    <value>/opt/hadoop-2.7.3/hdf/name</value>

    <final>true</final>

  </property>

</configuration>

### 3.1.6修改mapred-site.xml

# cp ./hadoop-2.7.3/etc/hadoop/mapred-site.xml.template ./hadoop-2.7.3/etc/hadoop/mapred-site.xml

# vi ./hadoop-2.7.3/etc/hadoop/mapred-site.xml

配置内容：

<configuration>

  <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>

</configuration>

### 3.1.7修改yarn-site.xml

# vi ./hadoop-2.7.3/etc/hadoop/yarn-site.xml

配置内容：

<configuration>

  <property>

    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

    <value>org.apache.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>

## 3.2各个主机之间复制hadoop

# scp -r ./hadoop-2.7.3 slaver1:/opt

# scp -r ./hadoop-2.7.3 slaver2:/opt

# scp -r ./hadoop-2.7.3 slaver3:/opt

## 3.3各个主机配置hadoop环境变量(每个主机都需要这样操作)

# vi /etc/profile

编辑内容：

export HADOOP\_HOME=/opt/hadoop-2.7.3

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

export HADOOP\_LOG\_DIR=/opt/hadoop-2.7.3/logs

export YARN\_LOG\_DIR=$HADOOP\_LOG\_DIR

# source /etc/profile  #使配置文件生效

## 3.4格式化namenode (只需要在master机上执行,严格来说你需要那个做为namenode就在那个上面执行)

# cd /opt/hadoop-2.7.3/sbin

# hdfs namenode -format

## 3.5启动hadoop (只需要在master机上执行)

启动hdfs：

# cd /opt/hadoop-2.7.3/sbin

# start-all.sh

检查hadoop启动情况：

http://192.168.10.198:50070

http://192.168.10.198:8088/cluster

查看状态：

# /opt/hadoop-2.7.3/bin/hadoop dfsadmin -report

[root@master sbin]# ../bin/hdfs dfsadmin -report

Configured Capacity: 56338194432 (52.47 GB)

Present Capacity: 33484337152 (31.18 GB)

DFS Remaining: 33484255232 (31.18 GB)

DFS Used: 81920 (80 KB)

DFS Used%: 0.00%

Under replicated blocks: 0

Blocks with corrupt replicas: 0

Missing blocks: 0

Missing blocks (with replication factor 1): 0

-------------------------------------------------

Live datanodes (3):

Name: 192.168.10.168:50010 (slaver3)

Hostname: slaver3

Decommission Status : Normal

Configured Capacity: 18779398144 (17.49 GB)

DFS Used: 24576 (24 KB)

Non DFS Used: 5563150336 (5.18 GB)

DFS Remaining: 13216223232 (12.31 GB)

DFS Used%: 0.00%

DFS Remaining%: 70.38%

Configured Cache Capacity: 0 (0 B)

Cache Used: 0 (0 B)

Cache Remaining: 0 (0 B)

Cache Used%: 100.00%

Cache Remaining%: 0.00%

Xceivers: 1

Last contact: Fri Sep 02 00:23:30 PDT 2016

Name: 192.168.10.170:50010 (slaver1)

Hostname: slaver1

Decommission Status : Normal

Configured Capacity: 18779398144 (17.49 GB)

DFS Used: 28672 (28 KB)

Non DFS Used: 8745037824 (8.14 GB)

DFS Remaining: 10034331648 (9.35 GB)

DFS Used%: 0.00%

DFS Remaining%: 53.43%

Configured Cache Capacity: 0 (0 B)

Cache Used: 0 (0 B)

Cache Remaining: 0 (0 B)

Cache Used%: 100.00%

Cache Remaining%: 0.00%

Xceivers: 1

Last contact: Fri Sep 02 00:23:32 PDT 2016

Name: 192.168.10.143:50010 (slaver2)

Hostname: slaver2

Decommission Status : Normal

Configured Capacity: 18779398144 (17.49 GB)

DFS Used: 28672 (28 KB)

Non DFS Used: 8545669120 (7.96 GB)

DFS Remaining: 10233700352 (9.53 GB)

DFS Used%: 0.00%

DFS Remaining%: 54.49%

Configured Cache Capacity: 0 (0 B)

Cache Used: 0 (0 B)

Cache Remaining: 0 (0 B)

Cache Used%: 100.00%

Cache Remaining%: 0.00%

Xceivers: 1

Last contact: Fri Sep 02 00:23:29 PDT 2016

看到类似如上信息则表示成功了。

检查进程：

# jps

master主机包含ResourceManager、SecondaryNameNode、NameNode等，则表示启动成功，例如

2212 ResourceManager

2484 Jps

1917 NameNode

2078 SecondaryNameNode

各个slave主机包含DataNode、NodeManager等，则表示启用成功，例如

17153 DataNode

17334 Jps

17241 NodeManager

## 3.6出现问题

1.启动以后如果活着的slaver节点少于部署的，可以看看各自logs下的日志。若出现【UnknownHostException: VM-172.16.54.11: VM-172.16.54.11: Name or service not known】，即可能为你修改hostsname 以后没有重启电脑，执行hostname master 即可

2.注意把防火墙关了

# 四.安装配置zookeeper

## 4.1配置zookeeper环境变量

# vi /etc/profile

export ZOOKEEPER\_HOME=/opt/zookeeper-3.4.8

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

# source /etc/profile

## 4.2配置zookeeper

1、到zookeeper官网下载zookeeper     http://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.4.8/

2、在slaver1,slaver2,slaver3上面搭建zookeeper

例如：

slaver1 192.168.10.170

slaver2 192.168.10.143

slaver3 192.168.10.168

3、上传zookeeper-3.4.8.tar.gz到任意一台服务器的根目录，并解压：zookeeper：

# tar -zxvf zookeeper-3.4.8.tar.gz

4、在zookeeper目录下建立data目录，同时将zookeeper目录下conf/zoo\_simple.cfg文件复制一份成zoo.cfg

# mkdir /opt/zookeeper-3.4.8/data

# cd /opt/zookeeper-3.4.8/conf/

# cp zoo\_sample.cfg zoo.cfg

5、修改zoo.cfg

# vi zoo.cfg

# Thenumber of milliseconds of each tick

 tickTime=2000

# Thenumber of ticks that the initial

#synchronization phase can take

initLimit=10

# Thenumber of ticks that can passbetween

#sending a request and getting anacknowledgement

 syncLimit=5

# thedirectory where the snapshot isstored.

# do notuse /tmp for storage, /tmp hereis just

#example sakes.

**dataDir=/opt/zookeeper-3.4.8/data**

# theport at which the clients willconnect

**clientPort=2181**

# themaximum number of clientconnections.

#increase this if you need to handle moreclients

#maxClientCnxns=60

#

# Besure to read the maintenance sectionof the

# administratorguide before turning onautopurge.

#

#http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

#

# Thenumber of snapshots to retain indataDir

#autopurge.snapRetainCount=3

# Purgetask interval in hours

# Set to"0" to disable autopurge feature

#autopurge.purgeInterval=1

**server.1=slaver1:2888:3888**

**server.2=slaver2:2888:3888**

**server.3=slaver3:2888:3888**

6、拷贝zookeeper目录到另外两台服务器：

# cd /opt

# scp -r /opt/zookeeper-3.4.8 slaver1:/opt

# scp -r /opt/zookeeper-3.4.8 slaver2:/opt

# scp -r /opt/zookeeper-3.4.8 slaver3:/opt

分别在几台服务器的data目录下建立myid其ip对应相应的server.\*

server.1的myid内容为1

# cd /opt/zookeeper-3.4.8/data/

# touch myid

# echo 1 > myid

server.2的myid内容为2

# cd /opt/zookeeper-3.4.8/data/

# touch myid

# echo 2 > myid

server.3的myid内容为3

# cd /opt/zookeeper-3.4.8/data/

# touch myid

# echo 3 > myid

7、启动ZooKeeper集群，在每个节点上分别启动ZooKeeper服务：

|  |  |
| --- | --- |
| # cd /opt/zookeeper-3.4.8/bin/ | |
| # ./zkServer.sh start |  |

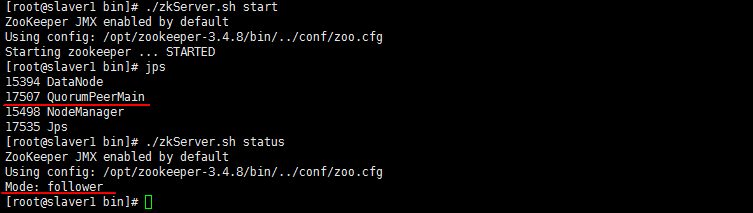
8、可以查看ZooKeeper集群的状态，保证集群启动没有问题：分别查看每台服务器的zookeeper状态zookeeper#bin/zkServer.shstatus查看那些是following那个是leader

Eg:

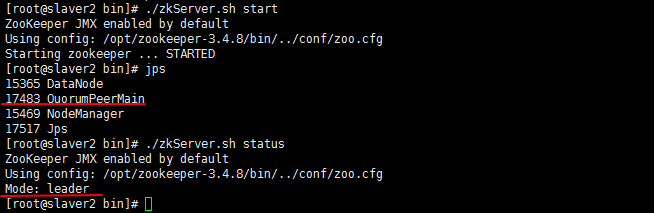
# ./jps

# ./zkServer.sh status

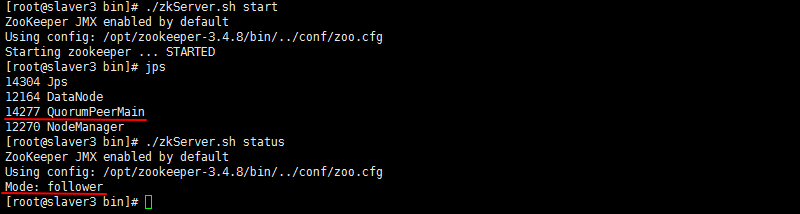
slaver1:



slaver2:



slaver3:



看到以上信息则表示安装成功。

# 五.安装配置hbase

## 5.1安装hbase

上传hbase-1.2.2-bin.tar.gz安装包到root根目录

# tar -zxvf hbase-1.2.2-bin.tar.gz

# mkdir /opt/hbase-1.2.2/logs

## 5.2配置hbase环境变量 (各主机)

# vi /etc/profile

export HBASE\_HOME=/opt/hbase-1.2.2

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

# source /etc/profile

## 5.3修改hbase-env.sh

# vi /opt/hbase-1.2.2/conf/hbase-env.sh

配置内容(直接替换所有内容或开放相应配置内容)：

export JAVA\_HOME=**/opt/jdk1.8.0\_101**

export HBASE\_LOG\_DIR=${HBASE\_HOME}/logs

export HBASE\_MANAGES\_ZK=false

## 5.4修改regionservers

# vi /opt/hbase-1.2.2/conf/regionservers

配置内容：

删除：localhost

添加：

slaver1

slaver2

slaver3

## 5.5修改hbase-site.xml

# vi /opt/hbase-1.2.2/conf/hbase-site.xml

配置内容：

<configuration>

<property>

<name>hbase.rootdir</name>

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

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>slaver1,slaver2,slaver3</value>

</property>

<property>

<name>hbase.master</name>

<value>hdfs://master:60000</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/opt/zookeeper-3.4.8/data</value>

</property>

<property>

<name>hbase.zookeeper.property.clientPort</name>

<value>2181</value>

</property>

<!--默认HMaster HTTP访问端口-->

<property>

<name>hbase.master.info.port</name>

<value>16010</value>

</property>

<!--默认HRegionServer HTTP访问端口-->

<property>

<name>hbase.regionserver.info.port</name>

<value>16030</value>

</property>

</configuration>

## 5.6各个主机之间复制hbase

# scp -r /opt/hbase-1.2.2 slaver1:/opt

# scp -r /opt/hbase-1.2.2 slaver2:/opt

# scp -r /opt/hbase-1.2.2 slaver3:/opt

## 5.7启动hbase (在master上执行，其它机器不需要执行)

启动之前先启动hadoop和zookeeper集群

启动hbase：

# cd /opt/hbase-1.2.2/bin

#./start-hbase.sh

## 5.8启动hbase 后活着的点只有本机器的，其他slaver点日志上

【INFO [regionserver/localhost/127.0.0.1:16020] regionserver.HRegionServer: reportForDuty to master=localhost,16000,1495017286384 with port=16020, startcode=1495017287226

2017-05-17 18:59:55,323 WARN [regionserver/localhost/127.0.0.1:16020] regionserver.HRegionServer: error telling master we are up

com.google.protobuf.ServiceException: java.net.ConnectException: Connection refused】

修改方法：将每个集群的主机的主机名修改为regionservers文件中对应的主机名

vi /etc/sysconfig/networks

HOSTNAME=主机名

如：在master上修改 HOSTNAME=master

在slaver1上修改 HOSTNAME= slaver1

在slaver2上修改 HOSTNAME= slaver2

在slaver3上修改 HOSTNAME= slaver3

等等

重启network服务设置生效

service network restart

## 5.9时间不同步导致hbase启动不了

解决方法：

一般情况下，将slaver们的节点同步到master就行了，执行命令：

ntpdate -bu master 【ntpdate -bu 192.168.59.128】

就行了

但是在虚拟机上，可能上述时间同步命令不好使，并且报错：

【24 Jul 21:59:25 ntpdate[2480]: no server suitable for synchronization found】

解决方法：【得能联网】，执行命令

yum install -y rdate

安装完成以后，执行命令：

rdate -s time-b.nist.gov

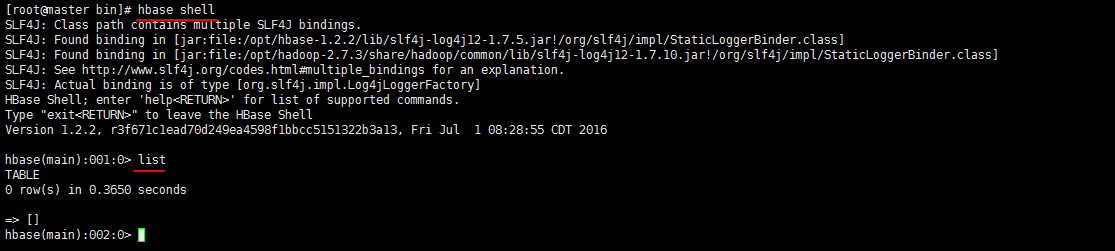
这样就行了

检查hbase启动情况：

# cd /opt/hbase-1.2.2/bin

#./hbase shell

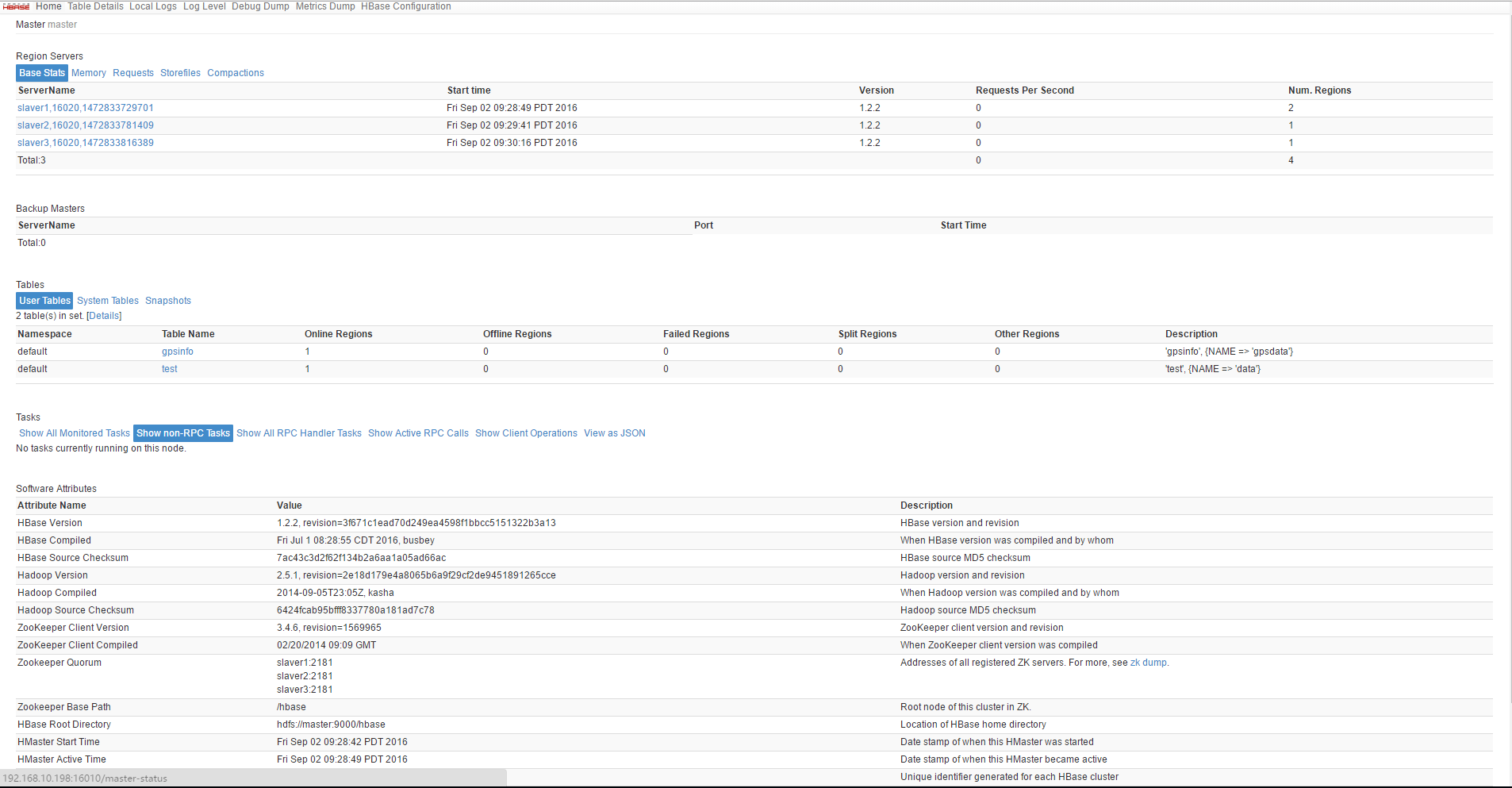
显示如下信息则为正常：



<http://192.168.10.198:16010/master-status> (注意：是master机)

<http://192.168.10.170:16030/rs-status> (注意：是slaver机)

(注：如果是在本机浏览器打开，需要注意本身的hosts配置或如果已经配置，则可以使用主机名直接访问，否则不可以)



看到上图表示正常安装启动。

检查进程：

# jps

master主机包含ResourceManager、SecondaryNameNode、NameNode、HQuorumPeer、HMaster等，则表示启动成功，例如

2212 ResourceManager

2999 Jps

2697 HQuorumPeer

1917 NameNode

2078 SecondaryNameNode

2751 HMaster

各个slave主机包含DataNode、NodeManager、HRegionServer、HQuorumPeer等，则表示启用成功，例如

17540 Jps

17142 NodeManager

17338 HRegionServer

17278 HQuorumPeer

17055 DataNode

# 六.试用hbase (hbase shell常用操作命令)

  根据下面tb1表的结构来演示hbase增删改查用法：

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **info** | | **address** |
| sex | age |
| zhangsan | 22 | man | beijing |
| lisi | 23 | woman | shanghai |

  # hbase shell  #进入字符页面

  6.1 创建表tb1，并有两个列族name、info和address，info列族下有sex和age列

|  |  |
| --- | --- |
|  | hbase(main):024:0> create 'tb1','name','info','address' |

  6.2 查看表结构

|  |  |
| --- | --- |
|  | hbase(main):025:0> describe 'tb1' |

  6.3 列出所有表

|  |  |
| --- | --- |
|  | hbase(main):025:0> list |

  6.4 插入几条记录

|  |  |
| --- | --- |
|  | hbase(main):028:0> put 'tb1','zhangsan','info:sex','22'    hbase(main):039:0> put 'tb1','zhangsan','info:age','man'    hbase(main):031:0> put 'tb1','zhangsan','address','beijing'    hbase(main):046:0> put 'tb1','lisi','info:age','woman'    hbase(main):047:0> put 'tb1','lisi','info:sex','23'    hbase(main):048:0> put 'tb1','lisi','address','shanghai' |

  6.5 查看所有记录（全表扫描）

|  |  |
| --- | --- |
|  | hbase(main):040:0> scan 'tb1'   ROW              COLUMN+CELL   zhangsan       column=address:,timestamp=1435129009088,value=beijing   zhangsan        column=info:age,timestamp=1435129054098, value=man   zhangsan        column=info:sex,timestamp=1435128714392, value=22 |

**说明：**

  ROW：行，用来检索记录的主键。

  COLUMN family：列族，是表的一部分，必须在创建表时定义，可以看到列名是以列族作为前缀，一个列族可以有多个列（column）。

  CELL：存储单位，存储实际数据，也就是所看到的value，cell中没有数据类型，全部是字节码形式存储。

  timestamp：时间戳，可以看做是数据版本号，hbase写时自动赋值，为当前系统时间，精确到毫秒。如果每个cell保存同一份数据多个版本时，可通过时间戳来索引版本。

  6.6 统计表中记录总数

|  |  |
| --- | --- |
|  | hbase(main):050:0> count 'tb1'    2 row(s) in 0.0190 seconds      => 2 |

  6.7 查看表中某条记录

|  |  |
| --- | --- |
|  | hbase(main):054:0> get 'tb1','zhangsan'    COLUMN                  CELL     address:               timestamp=1435129096397,value=beijing     info:age               timestamp=1435129054098,value=man     info:sex               timestamp=1435128714392,value=22 |

  6.8 查看表中某行某列族中的所有数据

|  |  |
| --- | --- |
|  | hbase(main):055:0> get 'tb1','zhangsan','info'    COLUMN                  CELL     info:age               timestamp=1435129054098,value=man     info:sex               timestamp=1435128714392,value=22 |

  6.9 更新一条记录（覆盖）

|  |  |
| --- | --- |
|  | hbase(main):063:0> put 'tb1','zhangsan','info:sex','23'    0 row(s) in 0.0080 seconds |

  6.10 给lisi增加一个comment字段

|  |  |
| --- | --- |
|  | hbase(main):070:0> incr 'tb1','lisi','info:comment' |

  6.11 删除某行某列族数据

|  |  |
| --- | --- |
|  | hbase(main):065:0> delete 'tb1','zhangsan','info:sex' |

  6.12 删除某行所有记录

|  |  |
| --- | --- |
|  | hbase(main):067:0> deleteall 'tb1','zhangsan' |

  6.13 删除一个表

|  |  |
| --- | --- |
|  | hbase(main):072:0> disable 'tb1'  #先禁用    hbase(main):073:0> drop 'tb1'   #再删除 |