尚硅谷大数据技术之Hive基础

（作者：尽际）

官网：<http://www.atguigu.com>

# 一、Linux网络配置复习

## 1.1 CentOS 6

1)Linux网络配置

# vi /etc/udev/rules.d/70-persistent-net.rules

复制ATTR{address}=="00:0c:29:e1:bc:79"引号中的值到ifcfg-etho0中

# vi /etc/sysconfig/network-scripts/ifcfg-eth0

配置如下：

DEVICE=eth0

HWADDR=00:0C:29:E1:BC:79

TYPE=Ethernet

UUID=a6713a48-ce04-4bb5-9e4d-8f929976196c

ONBOOT=yes

NM\_CONTROLLED=yes

BOOTPROTO=static

IPADDR=192.168.122.10

GATEWAY=192.168.122.2

NETMASK=255.255.255.0

DNS1=114.215.126.16

DNS2=192.168.122.2

2)Linux防火墙

service iptables status （功能描述：查看防火墙状态）

chkconfig iptables –list （功能描述：查看防火墙开机启动状态）

service iptables stop （功能描述：临时关闭防火墙）

chkconfig iptables off （功能描述：关闭防火墙开机启动）

chkconfig iptables on （功能描述：开启防火墙开机启动）

## 1.2 CentOS 7

1)Linux网络配置

# vi /etc/sysconfig/network-scripts/ifcfg-eno16777736

TYPE="Ethernet"

BOOTPROTO="static"

DEFROUTE="yes"

IPV4\_FAILURE\_FATAL="no"

IPV6INIT="yes"

IPV6\_AUTOCONF="yes"

IPV6\_DEFROUTE="yes"

IPV6\_FAILURE\_FATAL="no"

NAME="eno16777736"

UUID="43d220b2-1f75-4811-8a4b-2cf798f36b46"

DEVICE="eno16777736"

ONBOOT="yes"

DNS1="192.168.122.2"

DNS2="202.102.227.68"

IPADDR=192.168.122.200

PREFIX=24

GATEWAY=192.168.122.2

IPV6\_PEERDNS=yes

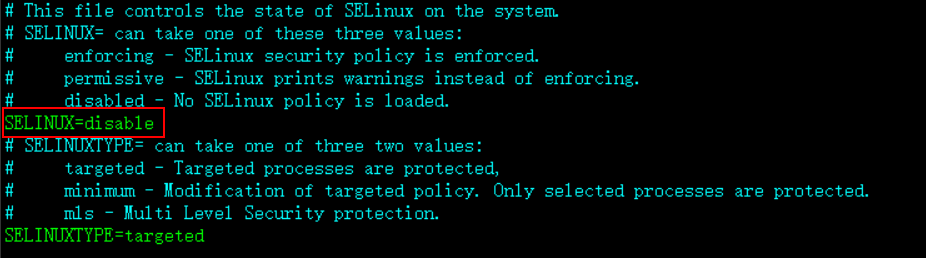
IPV6\_PEERROUTES=yes

NETMASK=255.255.255.0

2)Linux防火墙

# vi /etc/sysconfig/selinux

将设置改为禁用：SELINUX=disabled，如图：



# systemctl stop firewalld.service（关闭防火墙）

# systemctl disable firewalld.service（防火墙开机禁用）

# 二、Linux软件环境配置

## 1.1 JDK安装

$ tar -zxf /opt/softwares/jdk-8u121-linux-x64.gz -C /opt/modules/

## 1.2 JDK环境变量配置

# vi /etc/profile

#JAVA\_HOME

JAVA\_HOME=/opt/modules/jdk1.8.0\_121

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

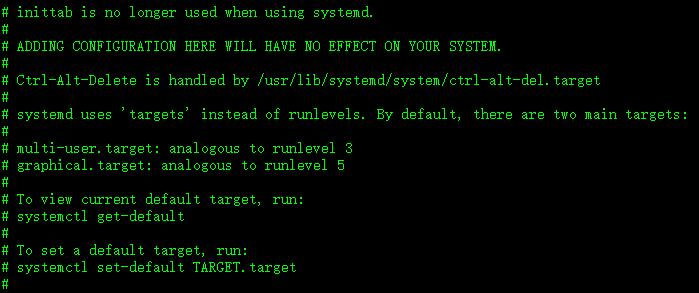
export

PATH=$PATH:$JAVA\_HOME/bin

# 三、Linux 其他准备操作

## 3.1字符模式启动

# cat /etc/inittab



# systemctl set-default multi-user.target，来设置无界面启动linux

# systemctl set-default graphical.target，来设置有界面启动linux

## 3.2 配置NTP时间服务器

### 3.2.1检查时区

对于我们当前这种案例，主要目标是把z01这台服务器设置为时间服务器，剩下的z02，z03这两台机器同步z01的时间，我们需要这样做的原因是因为，整个集群架构中的时间，要保持一致。

检查当前系统时区，使用命令：# date -R

http://upload-images.jianshu.io/upload_images/4951489-0e7365df67bee854.png?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240

注意这里，如果显示的时区不是+0800，你可以删除localtime文件夹后，再关联一个正确时区的链接过去，命令如下：

# rm -rf /etc/localtime

# ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

### 3.2.2同步时间

# ntpdate pool.ntp.org

### 3.2.3修改NTP配置文件

# vi /etc/ntp.conf

去掉下面这行前面的# ,并把网段修改成自己的网段：

restrict 192.168.122.0 mask 255.255.255.0 nomodify notrap

注释掉以下几行：

#server 0.centos.pool.ntp.org

#server 1.centos.pool.ntp.org

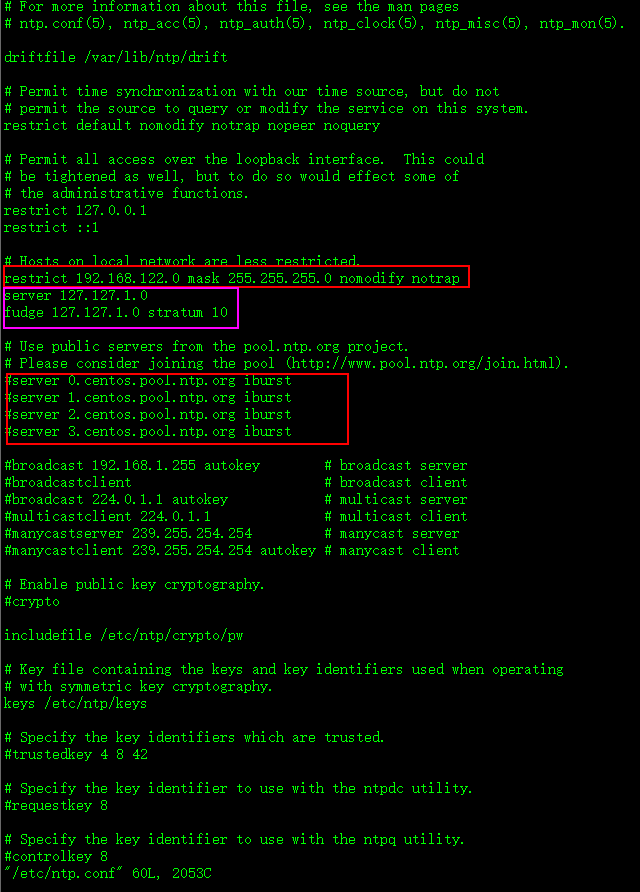
#server 2.centos.pool.ntp.org

把下面两行前面的#号去掉,如果没有这两行内容,需要手动添加

server  127.127.1.0    # local clock

fudge  127.127.1.0 stratum 10

最后，如图所示：



### 3.2.4重启ntp服务

# systemctl start ntpd.service，注意，如果是centOS7以下的版本，使用命令：service ntpd start

# systemctl enable ntpd.service，注意，如果是centOS7以下的版本，使用命令：chkconfig ntpd on

### 3.2.5集群其他节点去同步这台时间服务器时间

首先需要关闭这两台计算机的ntp服务

# systemctl stop ntpd.service，centOS7以下，则：service ntpd stop

# systemctl disable ntpd.service，centOS7以下，则：chkconfig ntpd off

# systemctl status ntpd，查看ntp服务状态

# pgrep ntpd，查看ntp服务进程id

同步第一台服务器z01的时间：

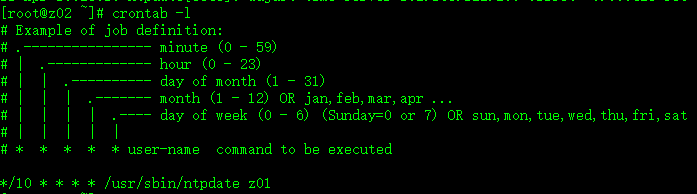
# ntpdate z01

http://upload-images.jianshu.io/upload_images/4951489-2cb91f2f52e57c48.png?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240

### 3.2.6制定计划任务,周期性同步时间

# crontab -e

\*/10 \* \* \* \* /usr/sbin/ntpdate z01



### 3.2.7重启定时任务

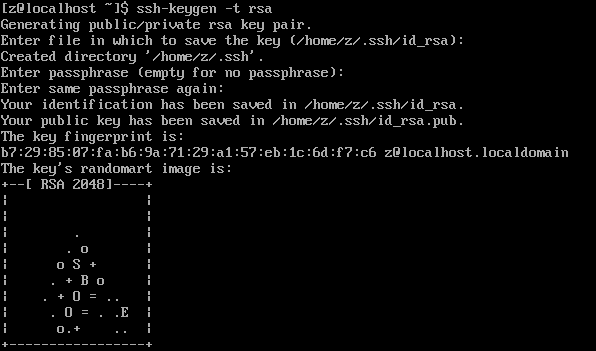
# systemctl restart crond.service，centOS7以下使用：service crond restart，其他台机器的配置同理

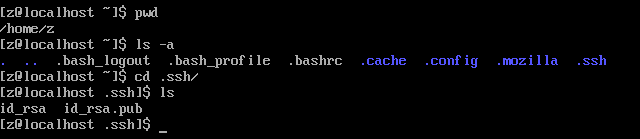
## 3.3 ssh无秘钥登录

配置hadoop集群，首先需要配置集群中的各个主机的ssh无密钥访问

在z04上，通过如下命令，生成一对公私钥对

$ ssh-keygen -t rsa，一顿回车操作，这条命令执行完毕后（注意使用普通用户执行该命令），会在/home/z/.ssh/目录下生成两个文件：id\_rsa 和 id\_rsa.pub，如图所示：





生成之后呢，把z01生成的公钥拷贝给z01,z02,z03这三台机器，对，没错，包含当前机器。

$ ssh-copy-id z01

$ ssh-copy-id z02

$ ssh-copy-id z03

完成后，其他机器同理。

# 四、CDH安装

## 4.1 CDH之zookeeper

### 4.1.1修改zoo.cfg配置文件

修改conf目录下的zoo.cfg文件，如果没有该文件，请自行重命名sample.cfg文件，修改内容为：

dataDir=/opt/modules/cdh/zookeeper-3.4.5-cdh5.3.6/zkData

server.1=z01:2888:3888

server.2=z02:2888:3888

server.3=z03:2888:3888

同时创建dataDir属性值所指定的目录

### 4.1.2在zkData目录下创建myid文件，修改值为1，如：

$ cd /opt/modules/cdh/zookeeper-3.4.5-cdh5.3.6/zkData

$ touch myid

$ echo 1 > myid

### 4.1.3将zookeeper安装目录scp到其他机器节点

$ scp -r /opt/modules/cdh/zookeeper-3.4.5-cdh5.3.6/ z05:/opt/modules/cdh/

$ scp -r /opt/modules/cdh/zookeeper-3.4.5-cdh5.3.6/ z06:/opt/modules/cdh/

### 4.1.4修改其他机器节点的myid文件为2和3

$ echo 2 > myid

$ echo 3 > myid

### 4.1.5在每个节点上启动zookeeper以及查看状态

$ bin/zkServer.sh start

$ bin/zkServer.sh status

## 4.2 CDH 之hadoop

### 4.2.1NameNode HA

#### \* hdfs-site.xml

<configuration>

<!-- 指定数据冗余份数 -->

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<!-- 完全分布式集群名称 -->

<property>

<name>dfs.nameservices</name>

<value>mycluster</value>

</property>

<!-- 集群中NameNode节点都有哪些 -->

<property>

<name>dfs.ha.namenodes.mycluster</name>

<value>nn1,nn2</value>

</property>

<!-- nn1的RPC通信地址 -->

<property>

<name>dfs.namenode.rpc-address.mycluster.nn1</name>

<value>z04:8020</value>

</property>

<!-- nn2的RPC通信地址 -->

<property>

<name>dfs.namenode.rpc-address.mycluster.nn2</name>

<value>z05:8020</value>

</property>

<!-- nn1的http通信地址 -->

<property>

<name>dfs.namenode.http-address.mycluster.nn1</name>

<value>z04:50070</value>

</property>

<!-- nn2的http通信地址 -->

<property>

<name>dfs.namenode.http-address.mycluster.nn2</name>

<value>z05:50070</value>

</property>

<!-- 指定NameNode元数据在JournalNode上的存放位置 -->

<property>

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

<value>qjournal://z04:8485;z05:8485;z06:8485/mycluster</value>

</property>

<!-- 配置隔离机制，即同一时刻只能有一台服务器对外响应 -->

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence</value>

</property>

<!-- 使用隔离机制时需要ssh无秘钥登录-->

<property>

<name>dfs.ha.fencing.ssh.private-key-files</name>

<value>/home/z/.ssh/id\_rsa</value>

</property>

<!-- 声明journalnode服务器存储目录-->

<property>

<name>dfs.journalnode.edits.dir</name>

<value>/opt/modules/cdh/hadoop-2.5.0-cdh5.3.6/data/jn</value>

</property>

<!-- 关闭权限检查-->

<property>

<name>dfs.permissions.enable</name>

<value>false</value>

</property>

<!-- 访问代理类：client，mycluster，active配置失败自动切换实现方式-->

<property>

<name>dfs.client.failover.proxy.provider.mycluster</name>

<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>

</property>

</configuration>

#### \* core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://mycluster</value>

</property>

<property>

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

<value>/opt/modules/cdh/hadoop-2.5.0-cdh5.3.6/data</value>

</property>

</configuration>

完成后远程拷贝给其他服务器

命令操作：

启动服务

在各个JournalNode节点上，输入以下命令启动journalnode服务：

注意：设置ha的时候需要（如果之前已有集群的话）重新格式化一下namenode，不然会报错

$ sbin/hadoop-daemon.sh start journalnode

在[nn1]上，对其进行格式化，并启动

$ bin/hdfs namenode -format

$ sbin/hadoop-daemon.sh start namenode

在[nn2]上，同步nn1的元数据信息，并启动

$ bin/hdfs namenode -bootstrapStandby

$ sbin/hadoop-daemon.sh start namenode

手动把nn1设置为active

$ bin/hdfs haadmin -transitionToActive nn1

查看服务状态

$ bin/hdfs haadmin -getServiceState nn1

Namenode故障自动转移：

1. 在hdfs-site.xml中需要增加

<!--namenode故障自动转移-->

<property>

<name>dfs.ha.automatic-failover.enabled</name>

<value>true</value>

</property>

1. 在core-site.xml中需要增加，确保ha能与zookeeper连接上

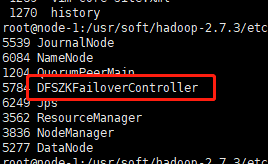
<property>

<name>ha.zookeeper.quorum</name> <value>192.168.136.111:2181,192.168.136.112:2181,192.168.136.113:2181</value>

</property>

1. 需要在namenode执行如下命令初始化ha在zookeeper中的状态

bin/hdfs zkfc -formatZK，执行完后会有如下一个进程，此进程就是用来处理故障自动转移的



### 4.2.2ResourceManager HA

\* yarn-site.xml

<configuration>

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

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.log-aggregation-enable</name>

<value>true</value>

</property>

<property>

<name>yarn.log.server.url</name>

<value>http://z01:19888/jobhistory/logs/</value>

</property>

<property>

<name>yarn.log-aggregation.retain-seconds</name>

<value>86400</value>

</property>

<!--启用resourcemanager ha-->

<property>

<name>yarn.resourcemanager.ha.enabled</name>

<value>true</value>

</property>

<!--声明两台resourcemanager的地址-->

<property>

<name>yarn.resourcemanager.cluster-id</name>

<value>cluster-yarn1</value>

</property>

<property>

<name>yarn.resourcemanager.ha.rm-ids</name>

<value>rm1,rm2</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm1</name>

<value>z02</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm2</name>

<value>z03</value>

</property>

<!--指定zookeeper集群的地址-->

<property>

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

<value>z01:2181,z02:2181,z03:2181</value>

</property>

<!--启用自动恢复-->

<property>

<name>yarn.resourcemanager.recovery.enabled</name>

<value>true</value>

</property>

<!--指定resourcemanager的状态信息存储在zookeeper集群-->

<property>

<name>yarn.resourcemanager.store.class</name> <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</value>

</property>

</configuration>

完成后远程拷贝给其他服务器

$ scp etc/hadoop/yarn-site.xml z02:/opt/modules/hadoop-2.5.0/etc/hadoop/

通过jps查看每个服务器的zookeeper服务QuorumPeerMain已经运行，没有运行则开启，方式前文已经说过，不再赘述。

在z02中执行：

$ sbin/start-yarn.sh

在z03中执行：

$ sbin/yarn-daemon.sh start resourcemanager

查看服务状态

$ bin/yarn rmadmin -getServiceState rm1

测试

$ bin/yarn jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0.jar wordcount /input/ /output/

注意：resourcemanager需要一个一个启动

Node-1，node-2需要单独启动resourcemanager