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：

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disable
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
# 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/tool

s.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

```
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two main targets:
#
# multi-user.target: analogous to runlevel 3
# graphical.target: analogous to runlevel 5
#
# To view current default target, run:
# systemctl get-default
#
# To set a default target, run:
# systemctl set-default TARGET.target
#
```

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

```
[root@z01 /]# date -R
Tue, 11 Apr 2017 09:52:28 +0800
```

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：

```
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(5).

driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict ::1

# Hosts on local network are less restricted.
restrict 192.168.122.0 mask 255.255.255.0 nomodify notrap
server 127.127.1.0
fudge 127.127.1.0 stratum 10

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.centos.pool.ntp.org iburst
#server 1.centos.pool.ntp.org iburst
#server 2.centos.pool.ntp.org iburst
#server 3.centos.pool.ntp.org iburst

#broadcast 192.168.1.255 autokey        # broadcast server
#broadcastclient                        # broadcast client
#broadcast 224.0.1.1 autokey            # multicast server
#multicastclient 224.0.1.1              # multicast client
#manycastserver 239.255.254.254         # manycast server
#manycastclient 239.255.254.254 autokey # manycast client

# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8
"/etc/ntp.conf" 60L, 2053C
```

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

```
[root@z02 ~]# ntpdate z01
11 Apr 10:28:10 ntpdate[3843]: adjust time server 192.168.122.200 offset -0.000428 sec
```

3.2.6 制定计划任务,周期性同步时间

crontab -e

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

```
[root@z02 ~]# crontab -l
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

*/10 * * * * /usr/sbin/ntpdate z01
```

3.2.7 重启定时任务

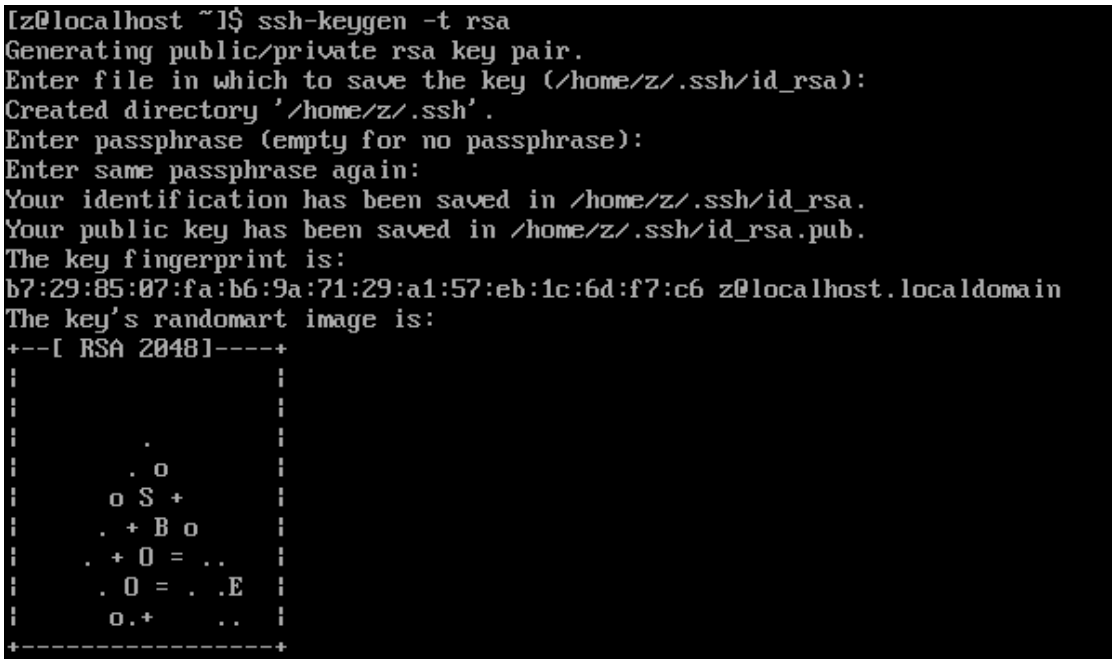
systemctl restart crond.service，centOS7 以下使用：service crond restart，其他台机器的配置同理

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.ConfiguredFailoverProxyProvi
der</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：

```
$ 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

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</va
lue>
    </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/
```