

Hadoop 集群安装配置文档

日期	内容	修订人
2015.6.3	文档初始化	易新

目录

1 文档概要	5
1.1 软件版本	5
1.2 机器配置	5
2 虚拟机配置	5
2.1 新建虚拟机	5
2.2 虚拟网络配置	8
3 CentOS 安装及配置	9
3.1 系统安装	9
3.2 系统配置	10
3.2.1 防火墙配置	10
3.2.2 SELinux 配置	10
3.2.3 IP 配置	11
3.2.4 安装 vim	12
3.2.5 新建用户	12
3.2.6 SSH 客户端连接	12
3.2.7 FTP 客户端连接	13
3.2.8 克隆虚拟机	13
3.2.9 设置 hostname	14
3.2.10 设置 hosts	15
3.2.11 配置 SSH 免密码登陆	15
3.2.12 安装 jdk	20
4 Hadoop 配置与优化	20
4.1 配置	20
4.1.1 最大文件打开数	20
4.1.2 上传安装包	21
4.1.3 解压安装包	21
4.1.4 删除安装包	21
4.1.5 配置 hadoop-env.sh	21
4.1.6 配置 core-site.xml	21
4.1.7 配置 hdfs-site.xml	21
4.1.8 配置 mapred-site.xml	22
4.1.9 配置 masters 文件	22
4.1.10 配置 slaves 文件	22
4.1.11 复制到其他节点	22
4.1.12 格式化 namenode	23
4.1.13 启动 hadoop	23
4.1.14 查看启动结果	23
5 Zookeeper 配置与优化	24
5.1 配置	24
5.1.1 上传安装包	24
5.1.2 解压安装包	24
5.1.3 删除安装包	24
5.1.4 配置 zoo.cfg	24
5.1.5 复制到其他节点	24

5.1.6 配置 server_id	24
5.1.7 启动 zookeeper	25
5.1.8 查看启动结果	25
6 Hbase 配置与优化	25
6.1 配置	25
6.1.1 上传安装包	25
6.1.2 解压安装包	25
6.1.3 删除安装包	25
6.1.4 配置 hbase-env.sh	25
6.1.5 配置 hbase-site.xml	26
6.1.6 配置 regionserver	27
6.1.7 修改 hdfs-site.xml	27
6.1.8 复制到其他节点	27
6.1.9 启动 hbase	27
6.1.10 查看启动结果	27
7 Kafka 配置与优化	28
7.1 配置	28
7.1.1 上传安装包	28
7.1.2 解压安装包	28
7.1.3 删除安装包	28
7.1.4 配置 server.properties	28
7.1.5 启动 kafka	29
7.1.6 查看启动结果	29
8 Hive 配置与优化	29
8.1 配置	29
8.1.1 上传安装包	29
8.1.2 解压安装包	29
8.1.3 删除安装包	29
8.1.4 配置 hive-site.xml	29
8.1.5 配置环境变量	30
8.1.6 配置 mysql 驱动包	30
8.1.7 配置 mysql	31
8.1.8 启动 hive	31
8.1.9 查看启动结果	31
9 Storm 配置与优化	31
9.1 配置	31
9.1.1 上传安装包	31
9.1.2 解压安装包	31
9.1.3 删除安装包	32
9.1.4 配置 storm.yaml	32
9.1.5 启动 storm	32
9.1.6 查看启动结果	32
10 Spark 配置与优化	33
10.1 配置	33
10.1.1 上传安装包	33
10.1.2 解压安装包	33

10.1.3 删除安装包	33
10.1.4 配置 conf/slaves	33
10.1.5 配置 conf/spark-env.sh.....	33
10.1.6 启动 spark	34
10.1.7 查看启动结果	34

1 文档概要

1.1 软件版本

本文档针对的系统版本和 Hadoop 版本，见表

名称	版本	说明
VMware Workstation	11.0.0	
Linux 发行版	CentOS-6.3	
SSH 客户端	Xshell 5	
FTP 客户端	Xftp 5	
Hadoop	hadoop-2.5.0-cdh5.2.0	
Zookeeper	zookeeper-3.4.5-cdh5.2.0	
Kafka	kafka_2.10-0.8.2.1.tgz	
Hbase	hbase-0.98.6-cdh5.2.0	
Hive	hive-0.13.1-cdh5.2.0	
Storm	apache-storm-0.9.4.tar.gz	
Spark	spark-1.3.0-bin-hadoop2.4.tgz	

1.2 机器配置

宿主机及虚拟机配置情况，见表

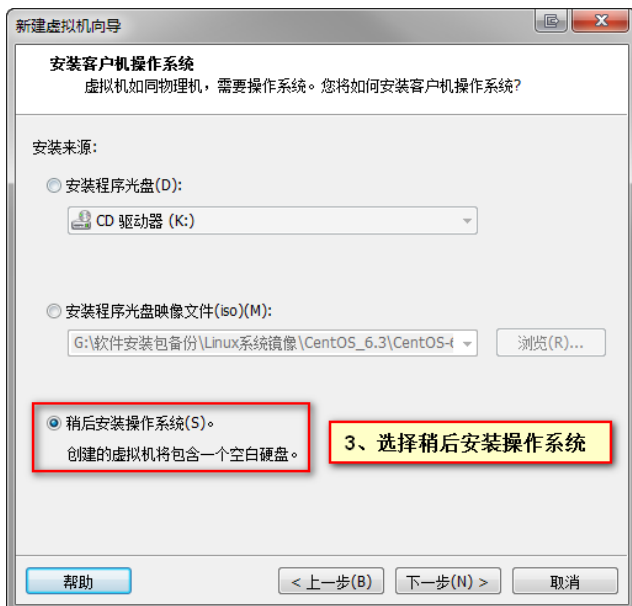
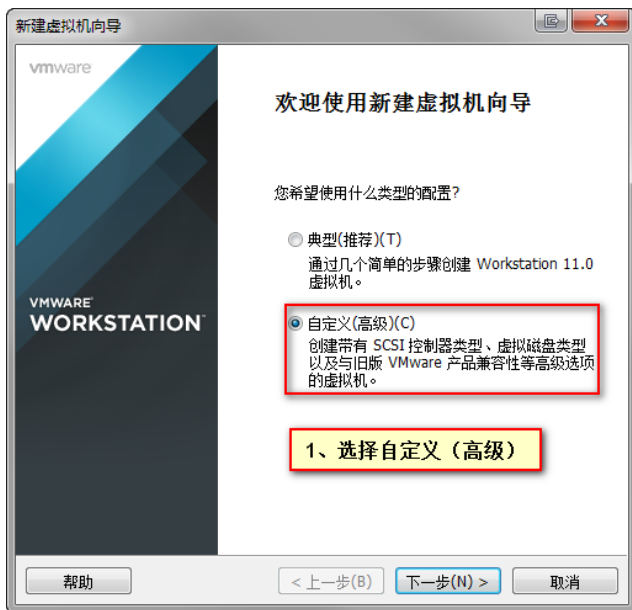
	虚拟机 master	虚拟机 slave1	虚拟机 slave2
内存	1G	1G	1G
硬盘	20G	20G	20G
网络	192.168.163.10	192.168.163.11	192.168.163.12

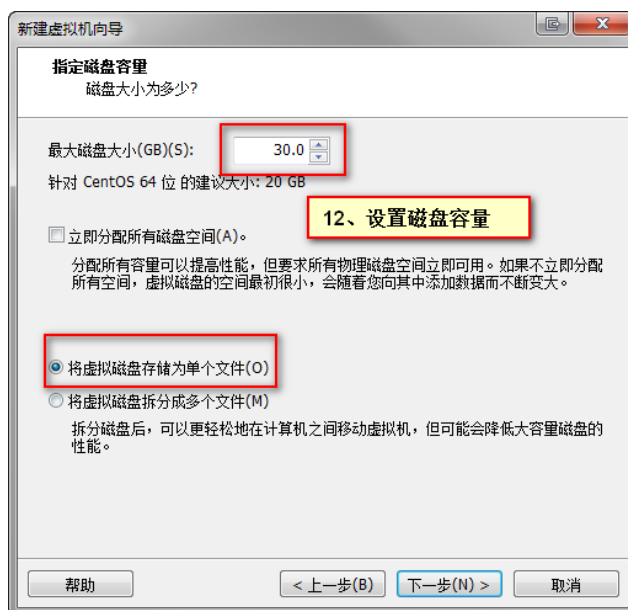
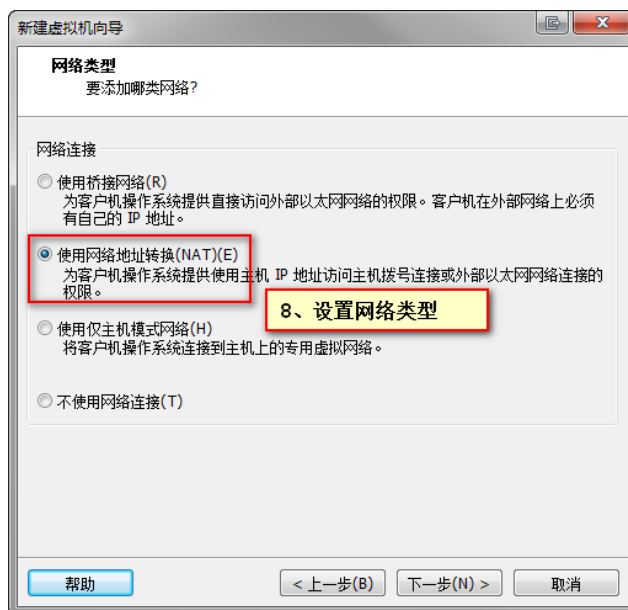
说明：虚拟机内存及硬盘的大小需要根据宿主机情况，进行合理分配。

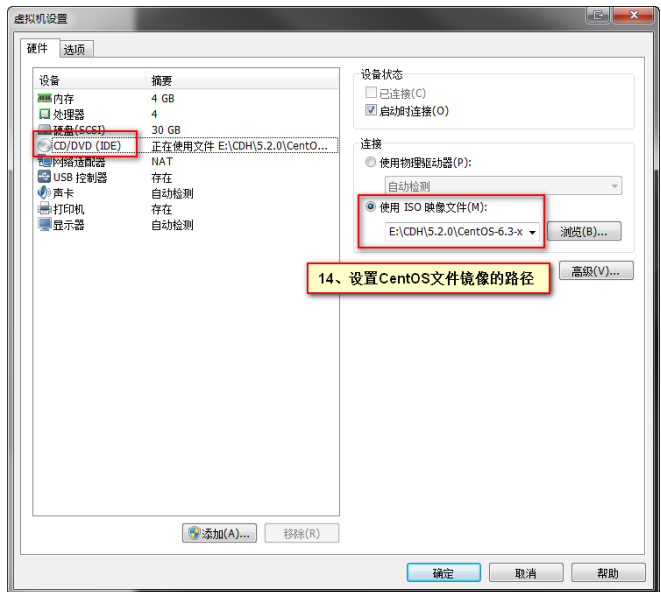
2 虚拟机配置

2.1 新建虚拟机

文件 - 新建虚拟机

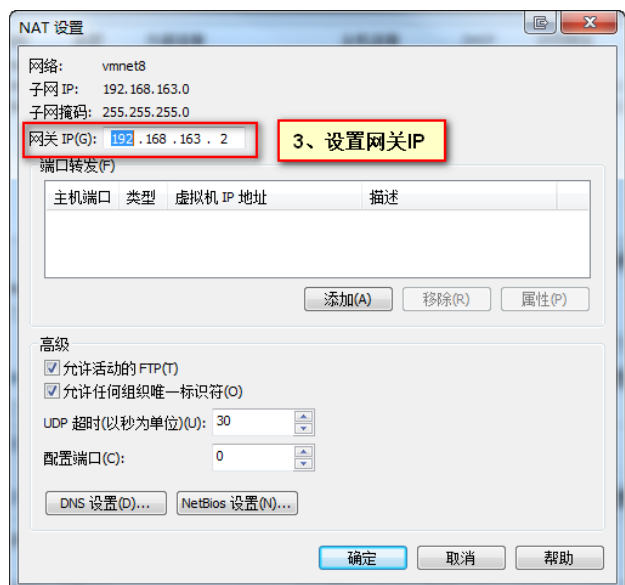
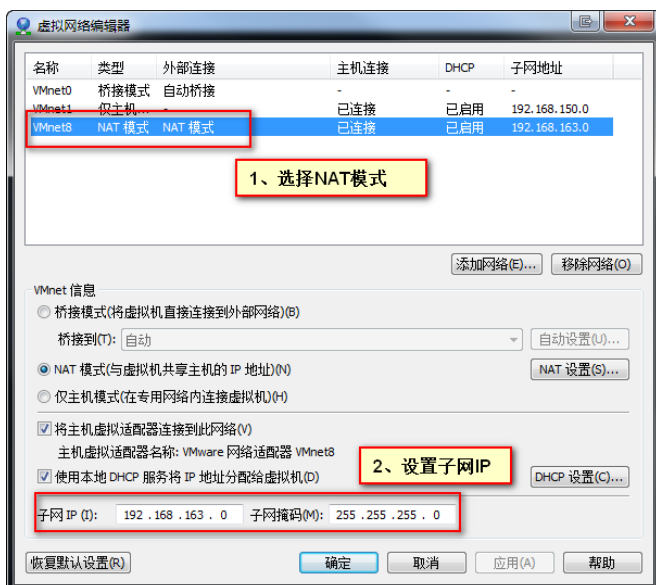






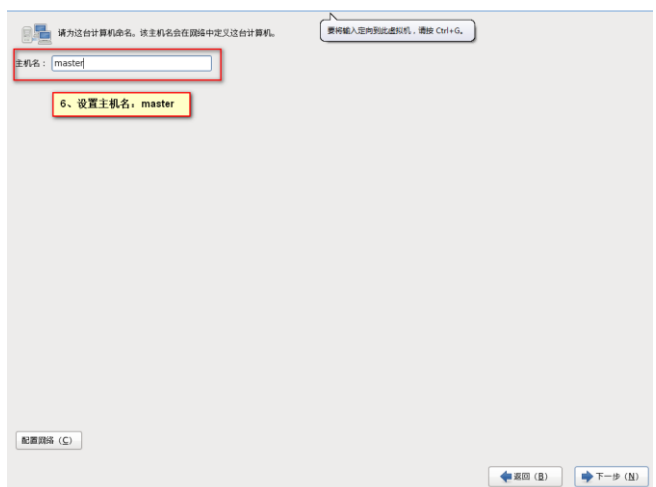
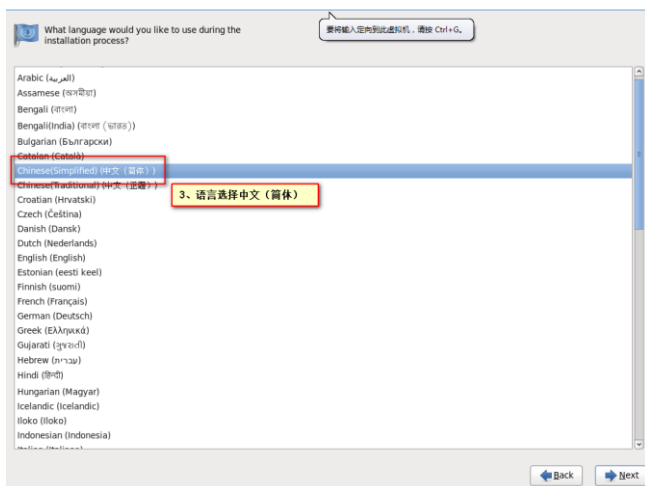
2.2 虚拟网络配置

编辑 - 虚拟网络编辑器



3 CentOS 安装及配置

3.1 系统安装





3.2 系统配置

3.2.1 防火墙配置

关闭防火墙，运行完成后查看防火墙关闭状态

```
[root@master ~]# service iptables stop
iptables: Flushing firewall rules:      [ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Unloading modules:          [ OK ]
[root@master ~]# chkconfig iptables off
[root@master ~]# service iptables status
iptables: Firewall is not running.
```

3.2.2 SELinux 配置

打开 SELinux 配置文件，将 SELINUX 的值改成 disabled，即关闭安全服务。

```
[root@master ~]# vi /etc/sysconfig/selinux

# 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=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

修改配置文件后在下次启动时才生效，所以当前正在运行的 selinux 服务需要手动关闭。

```
[root@master ~]#
[root@master ~]# setenforce 0
[root@master ~]# getenforce
Permissive
```

3.2.3 IP 配置

配置 IP，打开配置文件

```
[root@master ~]#
[root@master ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

按下图进行配置，IP 地址段和网关，参见虚拟机设置

```
DEVICE="eth0"
BOOTPROTO=""static""
HWADDR="00:0C:29:7D:3A:EA"
NM_CONTROLLED="yes"
ONBOOT=""yes""
TYPE="Ethernet"
UUID="0000b0c6-7632-4146-b11c-b10c49e2b848"
IPADDR=192.168.163.10
NETMASK=255.255.255.0
GATEWAY=192.168.163.2
DNS1=8.8.8.8
DNS2=8.8.4.4
IPV6INIT=no
```

重启网络服务，以便生效

```
[root@master ~]#
[root@master ~]# service network restart
Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: [ OK ]
```

查看网络信息

```
[root@master ~]#
[root@master ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:7D:3A:EA
inet addr:192.168.163.10  Bcast:192.168.163.255  Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe7d:3aea/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 b)  TX bytes:720 (720.0 b)

lo        Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING  MTU:16436  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

ping 百度，看是否已经联网成功

```
[root@master ~]#  
[root@master ~]# ping www.baidu.com  
PING www.a.shifen.com (180.97.33.108) 56(84) bytes of data.  
64 bytes from 180.97.33.108: icmp_seq=1 ttl=128 time=6.07 ms  
64 bytes from 180.97.33.108: icmp_seq=2 ttl=128 time=9.24 ms  
64 bytes from 180.97.33.108: icmp_seq=3 ttl=128 time=9.62 ms
```

3.2.4 安装 vim

通过 yum install 安装 vim

```
[root@master ~]#  
[root@master ~]# yum -y install vim
```

出现 complete 后，安装成功

```
Installed:  
  vim-enhanced.x86_64 2:7.2.411-1.8.el6  
  
Dependency Installed:  
  gpm-libs.x86_64 0:1.20.6-12.el6  
  perl.x86_64 4:5.10.1-136.el6_6.1  
  perl-Module-Pluggable.x86_64 1:3.90-136.el6_6.1  
  perl-Pod-Escapes.x86_64 1:1.04-136.el6_6.1  
  perl-Pod-Simple.x86_64 1:3.13-136.el6_6.1  
  perl-libs.x86_64 4:5.10.1-136.el6_6.1  
  perl-version.x86_64 3:0.77-136.el6_6.1  
  vim-common.x86_64 2:7.2.411-1.8.el6  
  
Complete!
```

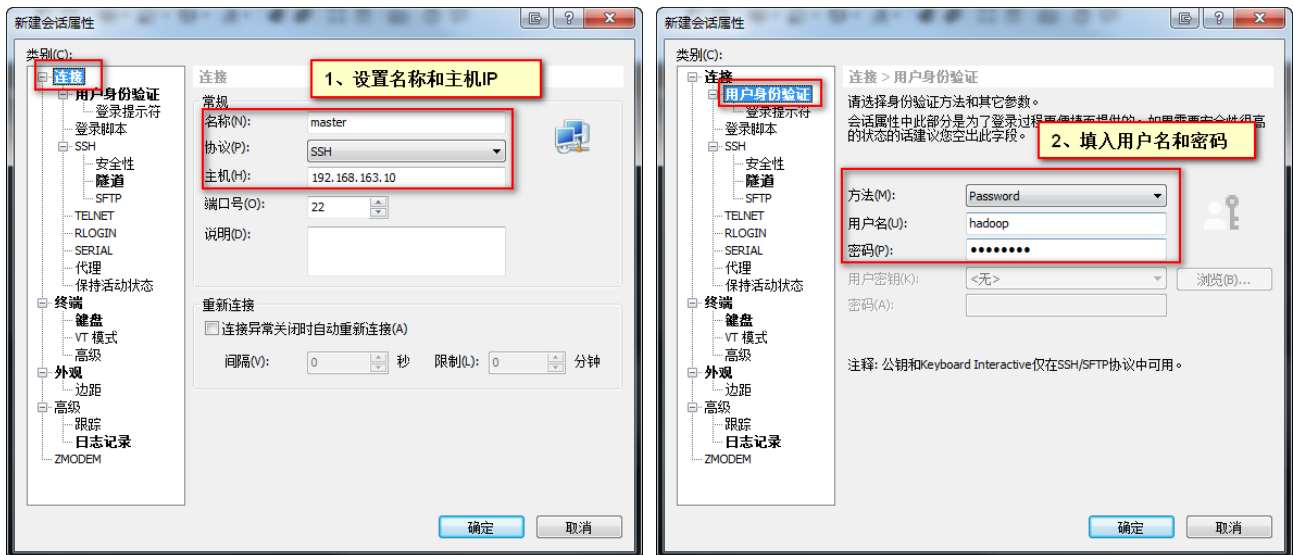
3.2.5 新建用户

新建系统用户，用户名为 hadoop，密码为 password

```
[root@master ~]#  
[root@master ~]# useradd hadoop  
[root@master ~]# passwd hadoop  
Changing password for user hadoop.  
New password:  
BAD PASSWORD: it is based on a dictionary word  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

3.2.6 SSH 客户端连接

打开 Xshell，新建会话



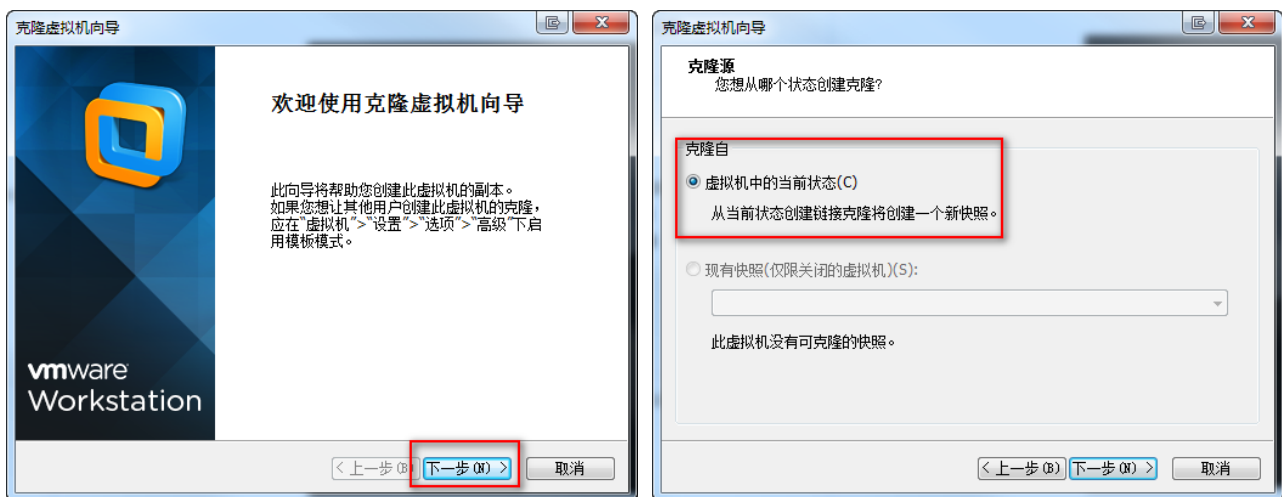
3.2.7 FTP 客户端连接

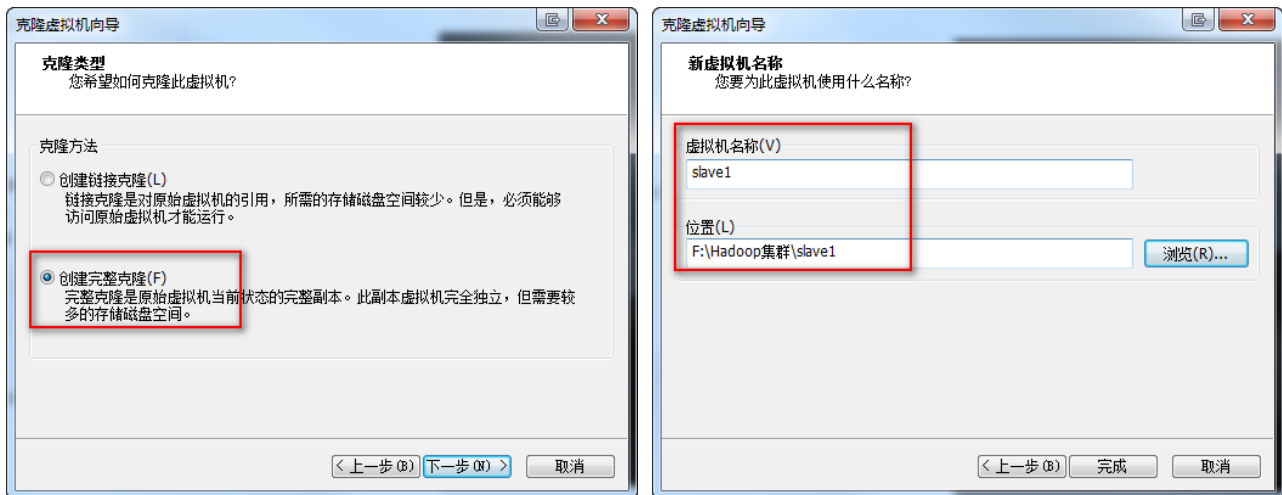
按照 3.2.6 登陆会话后，可直接通过菜单栏的文件传输按钮，建立 FTP 连接



3.2.8 克隆虚拟机

至此，系统的基本配置已设置完成。由于虚拟机一共有三台，每台都按照上面的过程进行设置是较为费时的，所以采用克隆虚拟机的办法。





克隆完成后，启动 slave1 和 slave2 虚拟机。由于克隆后的系统 MAC 地址都是一样的，造成 MAC 地址冲突，无法连接网络，按下面步骤进行设置。以 slave1 为例：

打开配置文件

```
[root@master ~]#  
[root@master ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0 _
```

删除 Mac 地址行 HWADDR，设置 slave1 节点的 IP 为 192.168.163.11

```
DEVICE="eth0"  
BOOTPROTO="static"  
HWADDR="88:0C:29:7D:3A:EA"  
NM_CONTROLLED="yes"  
ONBOOT="yes"  
TYPE="Ethernet"  
UUID="0000b0c6-7632-4146-b11c-b10c49e2b848"  
IPADDR=192.168.163.11  
NETMASK=255.255.255.0  
GATEWAY=192.168.163.2  
DNS1=8.8.8.8  
DNS2=8.8.4.4  
IPV6INIT=no
```

删除/etc/udev/rules.d/70-persistent-net.rules

```
[root@master ~]#  
[root@master ~]# rm -f /etc/udev/rules.d/70-persistent-net.rules _
```

slave2 节点也按照上面步骤进行操作，IP 设置为 192.168.163.12

重新启动 slave1 和 slave2 两台虚拟机，通过 ifconfig 命令查看 IP 是否正确，网络是否畅通。

3.2.9 设置 hostname

打开 hostname 配置文件

```
[root@master ~]#  
[root@master ~]# vim /etc/sysconfig/network _
```

将 HOSTNAME 设置为 master (slave1 虚拟机设置为 slave1，slave2 虚拟机设置为 slave2)

```
NETWORKING=yes  
HOSTNAME=master
```

重启虚拟机，通过 hostname 命令查看是否修改成功

```
[root@master ~]#  
[root@master ~]# hostname  
master
```

3.2.10 设置 hosts

打开 hosts 配置文件

```
[root@master ~]#  
[root@master ~]# vim /etc/hosts_
```

将 master, slave1, slave2 加到 hosts 文件。三台虚拟机的 hosts 文件都要按下图进行设置

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
192.168.163.10 master  
192.168.163.11 slave1  
192.168.163.12 slave2
```

配置完成后，尝试 ping 各个节点的 hostname，看是否配置成功。

```
[root@master ~]#  
[root@master ~]# ping slave1  
PING slave1 (192.168.163.11) 56(84) bytes of data.  
64 bytes from slave1 (192.168.163.11): icmp_seq=1 ttl=64 time=1.83 ms  
64 bytes from slave1 (192.168.163.11): icmp_seq=2 ttl=64 time=0.180 ms  
64 bytes from slave1 (192.168.163.11): icmp_seq=3 ttl=64 time=0.197 ms  
^C
```

3.2.11 配置 SSH 免密码登陆

3.2.11.1 配置 master 节点到 slave 节点的免密码登陆

查看是否安装有 ssh 和 rsync，可以看到 ssh 已经安装，但 rsync 还没有安装

```
[root@master ~]#  
[root@master ~]# rpm -qa | grep openssh  
openssh-server-5.3p1-81.el6.x86_64  
openssh-5.3p1-81.el6.x86_64  
openssh-clients-5.3p1-81.el6.x86_64  
[root@master ~]# rpm -qa | grep rsync  
[root@master ~]# _
```

通过 yum 安装 rsync

```
[root@slave1 ~]#  
[root@slave1 ~]# yum -y install rsync_
```

安装完毕后，重启 ssh 服务

```
[root@master ~]#  
[root@master ~]# service sshd restart_
```

在 master 节点上用 hadoop 用户登陆，并进入该用户的家目录。执行命令，生成公钥和私钥。

```
[hadoop@master ~]$  
[hadoop@master ~]$ ssh-keygen -t rsa -P ''  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):  
Created directory '/home/hadoop/.ssh'.  
Your identification has been saved in /home/hadoop/.ssh/id_rsa.  
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.  
The key fingerprint is:  
fa:91:be:20:ca:71:35:7b:1e:ef:02:7b:9e:48:b5:11 hadoop@master  
The key's randomart image is:  
+--[ RSA 2048 ]-----+  
|  
|          E          |  
|      .              |  
|     o S             |  
|    .. = +           |  
|   . o =o*          |  
|  . + o .Bo=         |  
| o   .oB+o          |  
+-----+  
|
```

查看用户目录下，是否生成.ssh 文件夹，公钥和私钥文件是否已经生成

```
[hadoop@master ~]$ ll -a .ssh/  
total 16  
drwx----- 2 hadoop hadoop 4096 Jun  2 01:06 .  
drwx----- 3 hadoop hadoop 4096 Jun  2 01:06 ..  
-rw----- 1 hadoop hadoop 1675 Jun  2 01:06 id_rsa  
-rw-r--r-- 1 hadoop hadoop  395 Jun  2 01:06 id_rsa.pub  
[hadoop@master ~]$ _
```

把 id_rsa.pub 公钥追加到授权文件中

```
[hadoop@master ~]$  
[hadoop@master ~]$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
[hadoop@master ~]$ _
```

修改授权文件的读写权限，如果不进行设置，在验证时仍提示要输入密码

```
[hadoop@master ~]$  
[hadoop@master ~]$ chmod 600 ~/.ssh/authorized_keys  
[hadoop@master ~]$ ll ~/.ssh  
total 12  
-rw----- 1 hadoop hadoop  395 Jun  2 01:09 authorized_keys  
-rw----- 1 hadoop hadoop 1675 Jun  2 01:06 id_rsa  
-rw-r--r-- 1 hadoop hadoop  395 Jun  2 01:06 id_rsa.pub
```

切换到 root 用户，修改 ssh 配置文件


```
[hadoop@master ~]$ su
Password:
[root@master hadoop]# vim /etc/ssh/sshd_config
```

删除这三项前面的注释符号

```

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
#AuthorizedKeysCommand none

```

重启 SSH 服务

```
[root@master ~]#
[root@master ~]# service sshd restart
```

SSH 登陆自己，看是否能成功，没问题的话，继续往下走

```
[root@master hadoop]#
[root@master hadoop]# ssh localhost
The authenticity of host 'localhost (:::1)' can't be established.
RSA key fingerprint is 3a:98:a5:63:24:79:d9:e4:a0:5a:98:77:2a:58:c8:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
root@localhost's password:
Last login: Tue Jun  2 00:46:20 2015
```

切换到 hadoop 用户，将 master 上生成的公钥，复制到 slave 节点上，例如传到 slave1。

询问是否连接时，输入 yes。询问密码时，输入 slave1 上 hadoop 用户的登陆密码。

```
[hadoop@master ~]$
[hadoop@master ~]$ scp ~/.ssh/id_rsa.pub hadoop@192.168.163.11:~/
The authenticity of host '192.168.163.11 (192.168.163.11)' can't be established.
RSA key fingerprint is 3a:98:a5:63:24:79:d9:e4:a0:5a:98:77:2a:58:c8:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.163.11' (RSA) to the list of known hosts.
hadoop@192.168.163.11's password:
id_rsa.pub                                100% 395      0.4KB/s   00:00
```

在 slave1 上用 hadoop 用户登录，查看是否已经有 id_rsa.pub 这个文件

```
[hadoop@slave1 ~]$
[hadoop@slave1 ~]$ ls
id_rsa.pub
[hadoop@slave1 ~]$ _
```

查看/home/hadoop 下是否有.ssh 文件夹

```
[hadoop@slave1 ~]$
[hadoop@slave1 ~]$ ls -a
. . . .bash_logout .bash_profile .bashrc id_rsa.pub
[hadoop@slave1 ~]$
```

如果没有此文件夹，需要手动创建。并把权限修改为 700

```
[hadoop@slave1 ~]$
[hadoop@slave1 ~]$ mkdir ~/.ssh
[hadoop@slave1 ~]$ chmod 700 ~/.ssh
[hadoop@slave1 ~]$ ll -a | grep .ssh
drwx----- 2 hadoop hadoop 4096 Jun  2 02:06 .ssh
```

把 master 的公钥追加到 slave2 的授权文件中，并修改 slave2 授权文件的权限。

```
[hadoop@slave1 ~]$  
[hadoop@slave1 ~]$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys  
[hadoop@slave1 ~]$ chmod 600 ~/.ssh/authorized_keys  
[hadoop@slave1 ~]$ ll ~/.ssh | grep authorized_keys  
-rw----- 1 hadoop hadoop 395 Jun  2 02:09 authorized_keys
```

切换到 root 用户，修改 ssh 配置文件

```
[hadoop@master ~]$ su  
Password:  
[root@master hadoop]# vim /etc/ssh/sshd_config
```

删除这三项前面的注释符号

```
RSAAuthentication yes  
PubkeyAuthentication yes  
AuthorizedKeysFile      .ssh/authorized_keys  
#AuthorizedKeysCommand none
```

重启 SSH 服务

```
[root@master ~]#  
[root@master ~]# service sshd restart
```

在 master 节点上，测试是否能够免密码登录到 slave1 节点

```
[hadoop@master ~]$  
[hadoop@master ~]$ ssh slave1  
Last login: Tue Jun  2 02:12:06 2015 from master
```

在 slave1 节点上，删除刚才 master 节点发过来的公钥文件

```
[root@slave1 hadoop]#  
[root@slave1 hadoop]# rm -f id_rsa.pub  
[root@slave1 hadoop]#
```

按照上面步骤，完成 master 到 slave2 节点的免密码配置。

3.2.11.2 slave 节点到 master 节点的免密码登陆

以 slave1 为例，在 slave1 上生成自己的公钥和私钥

```

[hadoop@slave1 ~]$ ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
f9:6d:af:ca:c0:ab:38:00:8d:b6:03:fd:1e:16:93:cd hadoop@slave1
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|
| + +
|+.o + E .
|o... o S
| o. + . . .
| .+ . o . o
| o. + . .
| ..... o....
+-----+

```

查看是否生成

```

[hadoop@slave1 ~]$ ll ~/.ssh
total 16
-rw----- 1 hadoop hadoop 395 Jun  2 02:09 authorized_keys
-rw----- 1 hadoop hadoop 1675 Jun  2 17:45 id_rsa
-rw-r--r-- 1 hadoop hadoop 395 Jun  2 17:45 id_rsa.pub
-rw-r--r-- 1 hadoop hadoop 403 Jun  2 02:15 known_hosts

```

追加到授权文件中

```

[hadoop@slave1 ~]$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

```

把公钥复制到 master 节点

```

[hadoop@slave1 ~]$ scp ~/.ssh/id_rsa.pub hadoop@192.168.163.10:~/
The authenticity of host '192.168.163.10 (192.168.163.10)' can't be established.
RSA key fingerprint is 3a:98:a5:63:24:79:d9:e4:a0:5a:98:77:2a:58:c8:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.163.10' (RSA) to the list of known hosts.
hadoop@192.168.163.10's password:
id_rsa.pub                                100% 395      0.4KB/s   00:00

```

切换到 master 节点，查看公钥文件是否已经过来，然后把公钥文件追加到授权文件，最后删除公钥文件。

```

[hadoop@master ~]$ ll
total 4
-rw-r--r-- 1 hadoop hadoop 395 Jun  2 17:51 id_rsa.pub
[hadoop@master ~]$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
[hadoop@master ~]$ rm -f ~/id_rsa.pub

```

在 slave1 节点测试是否可以免密码登录到 master 节点。

至此 slave1 到 master 的免密码登陆配置完成，slave2 到 master 也照此步骤进行即可。

```
[hadoop@slave1 ~]$  
[hadoop@slave1 ~]$ ssh master  
Last login: Tue Jun 2 17:55:11 2015 from slave1  
[hadoop@master ~]$ _
```

3.2.12 安装 jdk

通过 3.2.7 中的 FTP，将 jdk 安装包放入 /home/hadoop/soft 中，并解压，解压后的目录为 jdk1.7.0_75。

```
[hadoop@master soft]$  
[hadoop@master soft]$ tar -zxvf jdk-7u75-linux-x64.gz _
```

切换到 root 用户，打开环境变量配置文件

```
[root@master hadoop]#  
[root@master hadoop]# vim /etc/profile_
```

在文件末尾增加以下内容，并用 wq 命令进行保存并退出 vim

```
#set java  
export JAVA_HOME=/home/hadoop/soft/jdk1.7.0_75  
export PATH=$JAVA_HOME/bin:$PATH  
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

使配置文件生效

```
[root@master soft]#  
[root@master soft]# source /etc/profile  
[root@master soft]# _
```

检验 jdk 是否配置成功

```
[root@master soft]#  
[root@master soft]# java -version  
java version "1.7.0_75"
```

4 Hadoop 配置与优化

4.1 配置

4.1.1 最大文件打开数

```
[root@master soft]#  
[root@master soft]# ulimit -n 65535
```

4.1.2 上传安装包

上传 `hadoop-2.5.0-cdh5.2.0.tar.gz` 到 `/home/hadoop/soft`。

4.1.3 解压安装包

使用 `tar -zxvf hadoop-2.5.0-cdh5.2.0.tar.gz` 命令，进行解压。

4.1.4 删除安装包

使用 `rm -f hadoop-2.5.0-cdh5.2.0.tar.gz` 删除 hadoop 安装包。

4.1.5 配置 `hadoop-env.sh`

进入 `/home/hadoop/soft/hadoop-2.5.0-cdh5.2.0/etc/hadoop`，打开 `hadoop-env.sh`，修改 `JAVA_HOME`。
`export JAVA_HOME=/home/hadoop/soft/jdk1.7.0_75`

4.1.6 配置 `core-site.xml`

```
<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/hdfs/tmp</value>
    <description>A base for other temporary directories.</description>
  </property>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
  </property>
</configuration>
```

4.1.7 配置 `hdfs-site.xml`

```
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/home/hadoop/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/hadoop/hdfs/data</value>
  </property>
</configuration>
```

```
</property>
<property>
    <name>dfs.datanode.max.xcievers</name>
    <value>4096</value>
</property>
<property>
    <name>dfs.replication</name>
    <value>3</value>
</property>
<property>
    <name>dfs.support.append</name>
    <value>true</value>
</property>
</configuration>
```

4.1.8 配置 mapred-site.xml

```
<configuration>
    <property>
        <name>mapred.job.tracker</name>
        <value>http://master:9001</value>
    </property>
</configuration>
```

4.1.9 配置 masters 文件

master

4.1.10 配置 slaves 文件

slave1

slave2

4.1.11 复制到其他节点

将 master 上配置好的 hadoop，复制到其他节点。为了提高复制速度，可以采用先打包，然后复制，然后再解压的办法。

打包命令：tar -zcvf hadoop-2.5.0-cdh5.2.0.tar.gz hadoop-2.5.0-cdh5.2.0/

复制命令：scp hadoop-2.5.0-cdh5.2.0.tar.gz hadoop@slave1:~/

4.1.12 格式化 namenode

进入 `/home/hadoop/soft/hadoop-2.5.0-cdh5.2.0/bin`

执行 `./hadoop namenode -format`

4.1.13 启动 hadoop

进入 `/home/hadoop/soft/hadoop-2.5.0-cdh5.2.0/sbin`

执行 `./start-all.sh`

4.1.14 查看启动结果

在 master 中通过 `jps` 命令查看，包含以下进程：

NameNode

SecondaryNameNode

ResourceManager

在 slave 中通过 `jps` 命令查看，包含以下进程

DataNode

NodeManager

在浏览器中打开以下地址查看

Namenode information: <http://192.168.163.101:50070/>

Overview 'master:9000' (active)

Started:	Tue Jun 02 21:37:06 CST 2015
Version:	2.5.0-cdh5.2.0, re1f20a08bde76a33b79df026d00a0c91b2298387
Compiled:	2014-10-11T21:00Z by jenkins from Unknown
Cluster ID:	CID-891735a5-fc2c-40b5-8b8c-6efaa6b52cb7
Block Pool ID:	BP-218923006-192.168.163.10-1433249532523

All applications: <http://192.168.163.10:8088>



Cluster	Cluster Metrics				
	Apps	Apps	Apps	Apps	Containers
	Submitted	Pending	Running	Completed	Running
	0	0	0	0	0

5 Zookeeper 配置与优化

5.1 配置

5.1.1 上传安装包

上传 zookeeper-3.4.5-cdh5.2.0.tar.gz 到/home/hadoop/soft 目录下。

5.1.2 解压安装包

使用 `tar -zxvf zookeeper-3.4.5-cdh5.2.0.tar.gz` 命令，进行解压。

5.1.3 删除安装包

使用 `rm -f zookeeper-3.4.5-cdh5.2.0.tar.gz`，删除安装包。

5.1.4 配置 zoo.cfg

进入/home/hadoop/soft/zookeeper-3.4.5-cdh5.2.0/conf，复制 zoo_sample.cfg，并改名为 zoo.cfg

修改 dataDir=/home/hadoop/zookeeperdata

新增以下参数

server.1=master:2888:3888

server.2=slave1:2888:3888

server.3=slave2:2888:3888

5.1.5 复制到其他节点

将 master 上配置好的 zookeeper，复制到其他节点。为了提高复制速度，可以采用先打包，然后复制，然后再解压的办法。

打包命令：`tar -zcvf zookeeper-3.4.5-cdh5.2.0.tar.gz zookeeper-3.4.5-cdh5.2.0/`

复制命令：`scp zookeeper-3.4.5-cdh5.2.0.tar.gz hadoop@slave2:~/`

5.1.6 配置 server_id

手动创建好 5.1.4 中的目录：`mkdir /home/hadoop/zookeeperdata`

在目录下创建文件，文件名为“myid”：`touch myid`

编辑该文件：`vim myid`

根据 5.1.4 中设置的 id 号，填入相应的 id，如果是 master 节点，myid 的内容就是 1。

5.1.7 启动 zookeeper

进入 `soft/zookeeper-3.4.5-cdh5.2.0/bin/`

执行 `./zkServer.sh start`

每个节点都要手动启动。

5.1.8 查看启动结果

启动后，执行 `./zkServer.sh status`，可以查看该节点是 `follower` 还是 `leader`。

在 `master` 或者 `slave` 节点上，通过 `jps` 查看，进程名为：

`QuorumPeerMain`

6 Hbase 配置与优化

6.1 配置

6.1.1 上传安装包

上传 `hbase-0.98.6-cdh5.2.0.tar.gz` 到 `/home/hadoop/soft` 目录下。

6.1.2 解压安装包

使用 `tar -zxvf hbase-0.98.6-cdh5.2.0.tar.gz` 命令，进行解压。

6.1.3 删除安装包

使用 `rm -f hbase-0.98.6-cdh5.2.0.tar.gz`，删除安装包。

6.1.4 配置 `hbase-env.sh`

```
export JAVA_HOME=/home/hadoop/soft/jdk1.7.0_75/
export HBASE_HOME=/home/hadoop/soft/hbase-0.98.6-cdh5.2.0
export HBASE_CLASSPATH=/home/hadoop/soft/hadoop-2.5.0-cdh5.2.0/etc/hadoop
export HBASE_HEAPSIZE=200
export HBASE_OPTS="-ea -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode"
export HBASE_LOG_DIR=/home/hadoop/soft/hbase-0.98.6-cdh5.2.0/logs
export HBASE_PID_DIR=/home/hadoop/soft/hadoop-2.5.0-cdh5.2.0/pids
export HBASE_MANAGES_ZK=false
```

6.1.5 配置 hbase-site.xml

```
<configuration>
  <property>
    <name>hbase.tmp.dir</name>
    <value>file:/home/hadoop/hbasetmp</value>
  </property>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://master:9000/hbase</value>
  </property>
  <property>
    <name>dfs.datanode.max.xcievers</name>
    <value>4096</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
    <description>The mode the cluster will be in. Possible values are
      false: standalone and pseudo-distributed setups with managed Zookeeper
      true: fully-distributed with unmanaged Zookeeper Quorum (see hbase-env.sh)
    </description>
  </property>
  <property>
    <name>hbase.zookeeper.property.clientPort</name>
    <value>2181</value>
    <description>Property from ZooKeeper's config zoo.cfg.
      The port at which the clients will connect.
    </description>
  </property>
  <property>
    <name>hbase.zookeeper.quorum</name>
    <value>master,slave1,slave2</value>
    <description>Comma separated list of servers in the ZooKeeper Quorum.
      For example, "host1.mydomain.com,host2.mydomain.com,host3.mydomain.com".
      By default this is set to localhost for local and pseudo-distributed modes
      of operation. For a fully-distributed setup, this should be set to a full
      list of ZooKeeper quorum servers. If HBASE_MANAGES_ZK is set in hbase-env.sh
      this is the list of servers which we will start/stop ZooKeeper on.
    </description>
  </property>
  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>file:/home/hadoop/zookeeperdata</value>
    <description>Property from ZooKeeper's config zoo.cfg.
      The directory where the snapshot is stored.
  </property>
```

```
        </description>
    </property>
    <property>
        <name>hbase.master</name>
        <value>hdfs://master:60000</value>
    </property>
</configuration>
```

6.1.6 配置 regionserver

slave1

slave2

6.1.7 修改 hdfs-site.xml

修改 hadoop 集群配置文件 hdfs-site.xml，如果没有下面这一项，则在此文件中增加

```
<property>
    <name>dfs.support.append</name>
    <value>true</value>
</property>
```

每台节点的配置文件都要修改。

6.1.8 复制到其他节点

将 master 上配置好的 hbase，复制到其他节点。为了提高复制速度，可以采用先打包，然后复制，然后再解压的办法。

打包命令：tar -zcvf hbase-0.98.6-cdh5.2.0.tar.gz hbase-0.98.6-cdh5.2.0/

复制命令：scp hbase-0.98.6-cdh5.2.0.tar.gz hadoop@slave2:~/

6.1.9 启动 hbase

在 master 节点，进入 hbase-0.98.6-cdh5.2.0/bin/

执行 ./start-hbase.sh

6.1.10 查看启动结果

在 master 节点，通过 jps 命令查看，进程名为

HMaster

在 slave 节点，通过 jps 命令查看，进程名为

HRegionServer

在浏览器中打开以下地址查看

Master: <http://192.168.163.10:60010>

RegionServer: <http://192.168.163.11:60030>

7 Kafka 配置与优化

7.1 配置

7.1.1 上传安装包

上传 kafka_2.10-0.8.2.1.tgz 到/home/hadoop/soft 目录下。

7.1.2 解压安装包

使用 tar -zxvf kafka_2.10-0.8.2.1.tgz 命令，进行解压。

7.1.3 删除安装包

使用 rm -f kafka_2.10-0.8.2.1.tgz，删除安装包。

7.1.4 配置 server.properties

master 节点的 server.properties 配置如下：

broker.id=0

host.name=master

log.dirs=/home/hadoop/kafka/kafka-logs

zookeeper.connect=master:2181,slave1:2181,slave2:2181

slave1 节点的 server.properties 配置如下：

broker.id=1

host.name=slave1

log.dirs=/home/hadoop/kafka/kafka-logs

zookeeper.connect=master:2181,slave1:2181,slave2:2181

slave2 节点的 server.properties 配置如下：

broker.id=2

host.name=slave2

log.dirs=/home/hadoop/kafka/kafka-logs

zookeeper.connect=master:2181,slave1:2181,slave2:2181

7.1.5 启动 kafka

每台节点都要启动

nohup ./kafka_2.10-0.8.2.1/bin/kafka-server-start.sh kafka_2.10-0.8.2.1/config/server.properties &

7.1.6 查看启动结果

在 master 和 slave 节点上，通过 jps 命令查看，进程名为 kafka

创建一个名称为 test 的 topic

./bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test

创建 producer

./bin/kafka-console-producer.sh --broker-list 192.168.163.10:9092 --topic test

创建 consumer

./bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test

8 Hive 配置与优化

8.1 配置

8.1.1 上传安装包

上传 hive-0.13.1-cdh5.2.0.tar.gz 到/home/hadoop/soft 目录下。

需要注意的是，hive 只需安装在一个节点，不用每个节点都安装。

8.1.2 解压安装包

使用 tar -zxvf hive-0.13.1-cdh5.2.0.tar.gz 命令，进行解压。

8.1.3 删除安装包

使用 rm -f hive-0.13.1-cdh5.2.0.tar.gz，删除安装包。

8.1.4 配置 hive-site.xml

进入 hive 的/conf 目录中，将 hive-default.xml.template 复制一份，重命名为 hive-site.xml

hive 需要数据库配合，例如 mysql 或者 sqlserver，这里以 mysql 为例。

配置 hive-site.xml

将 mysql-ip 需要替换成你的 mysql 地址

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://mysql-ip:3306/dbhive?createDatabaseIfNotExist=true</value>
  <description>JDBC connect string for a JDBC metastore</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.jdbc.Driver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>dbhive </value>
  <description>username to use against metastore database</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>dbhive </value>
  <description>password to use against metastore database</description>
</property>
```

8.1.5 配置环境变量

切换到 root 用户，打开环境变量配置文件 vim /etc/profile

增加以下内容

```
#hadoop
export HADOOP_HOME=/home/hadoop/soft/hadoop-2.5.0-cdh5.2.0
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

#hive
export HIVE_HOME=/home/hadoop/soft/hive-0.13.1-cdh5.2.0
export PATH=$PATH:$HIVE_HOME/bin
```

执行 source /etc/profile，使环境变量生效

8.1.6 配置 mysql 驱动包

将 mysql 的连接驱动包 mysql-connector-java-5.1.35-bin.jar 放入 /hive-0.13.1-cdh5.2.0/lib 目录下

8.1.7 配置 mysql

登陆 mysql:

```
mysql -u root -p
```

创建数据库:

```
create database dbhive;
```

新建用户:

```
create user dbhive identified by 'dbhive';
```

设置权限

```
grant all privileges on dbhive.* to dbhive@'%' identified by 'dbhive';
```

```
grant all privileges on dbhive.* to dbhive @localhost identified by ' dbhive ';
```

8.1.8 启动 hive

进入 hive 的 bin 目录，执行./hive

8.1.9 查看启动结果

通过 jps 命令查看，进程名为 RunJar

执行./hive 启动 hive 后，可以进行查询。

例如执行 show tables;

```
hive> show tables;  
OK  
Time taken: 0.543 seconds
```

9 Storm 配置与优化

9.1 配置

9.1.1 上传安装包

上传 apache-storm-0.9.4.tar.gz 到/home/hadoop/soft 目录下。

9.1.2 解压安装包

使用 tar -zxvf apache-storm-0.9.4.tar.gz 命令，进行解压。

9.1.3 删除安装包

使用 `rm -f apache-storm-0.9.4.tar.gz`，删除安装包。

9.1.4 配置 storm.yaml

注意，yaml 配置文件对格式要求较高，编写时需要仔细。

```
storm.zookeeper.servers:
```

- "master"
- "slave1"
- "slave2"

```
nimbus.host: "master"
```

```
ui.port: 12321
```

```
supervisor.slots.ports:
```

- 6700
- 6701

9.1.5 启动 storm

在 master 上启动 ui 和 nimbus

```
nohup ./apache-storm-0.9.4/bin/storm ui &
```

```
nohup ./apache-storm-0.9.4/bin/storm nimbus &
```

在 slave 上启动 supervisor

```
nohup ./apache-storm-0.9.4/bin/storm supervisor &
```

9.1.6 查看启动结果

在 master 节点上，通过 `jps` 命令查看，进程名为

core

nimbus

在 slave 节点上，通过 `jps` 命令查看，进程名为

supervisor

在浏览器中打开以下地址查看

Storm UI: <http://192.168.163.10:12321>

Storm UI

Cluster Summary

Version	Nimbus uptime	Supervisors	Used slots
0.9.4	1m 18s	2	0

Topology summary

Name	Id	Status	Uptime	Num workers
------	----	--------	--------	-------------

10 Spark 配置与优化

10.1 配置

10.1.1 上传安装包

上传 spark-1.3.0-bin-hadoop2.4.tgz 到/home/hadoop/soft 目录下。

10.1.2 解压安装包

使用 tar -zxvf spark-1.3.0-bin-hadoop2.4.tgz 命令，进行解压。

10.1.3 删除安装包

使用 rm -f spark-1.3.0-bin-hadoop2.4.tgz，删除安装包。

10.1.4 配置 conf/slaves

```
master
slave1
slave2
```

10.1.5 配置 conf/spark-env.sh

```
export JAVA_HOME=/home/hadoop/soft/jdk1.7.0_75
export SPARK_MASTER_IP=master
export SPARK_MASTER_WEBUI_PORT=8011
export SPARK_MASTER_PORT=7071
```

```
export SPARK_WORKER_CORES=2
export SPARK_WORKER_MEMORY=500m
export SPARK_WORKER_INSTANCES=2
```

10.1.6 启动 spark

将以上两个配置文件复制到其他节点后，在 master 节点执行
./spark-1.3.0-bin-hadoop2.4/sbin/start-all.sh

10.1.7 查看启动结果

在 master 和 slave 节点上，通过 jps 命令查看，进程名为
Master
Worker

在浏览器中打开以下地址查看
Spark UI: <http://192.168.163.10:8011/>



URL: spark://master:7071
REST URL: spark://master:6066 (*cluster mode*)
Workers: 5
Cores: 10 Total, 0 Used
Memory: 2.4 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE