

实验一 RatHat6/Cent OS 6（RatHat7/Cent OS 7）下 hadoop 集群模式安装

（3 个节点，master,slave1,slave2） 4 课时

一、学习目标：

使用红帽子 RatHat6/Cent OS 6 在虚拟机上搭建 hadoop 集群，包含 3 个节点，体验集群分布式。

二、软件需求：

三台虚拟机(这里使用 VMware Workstation)、RatHat6 版本的安装包

三、实验前提：

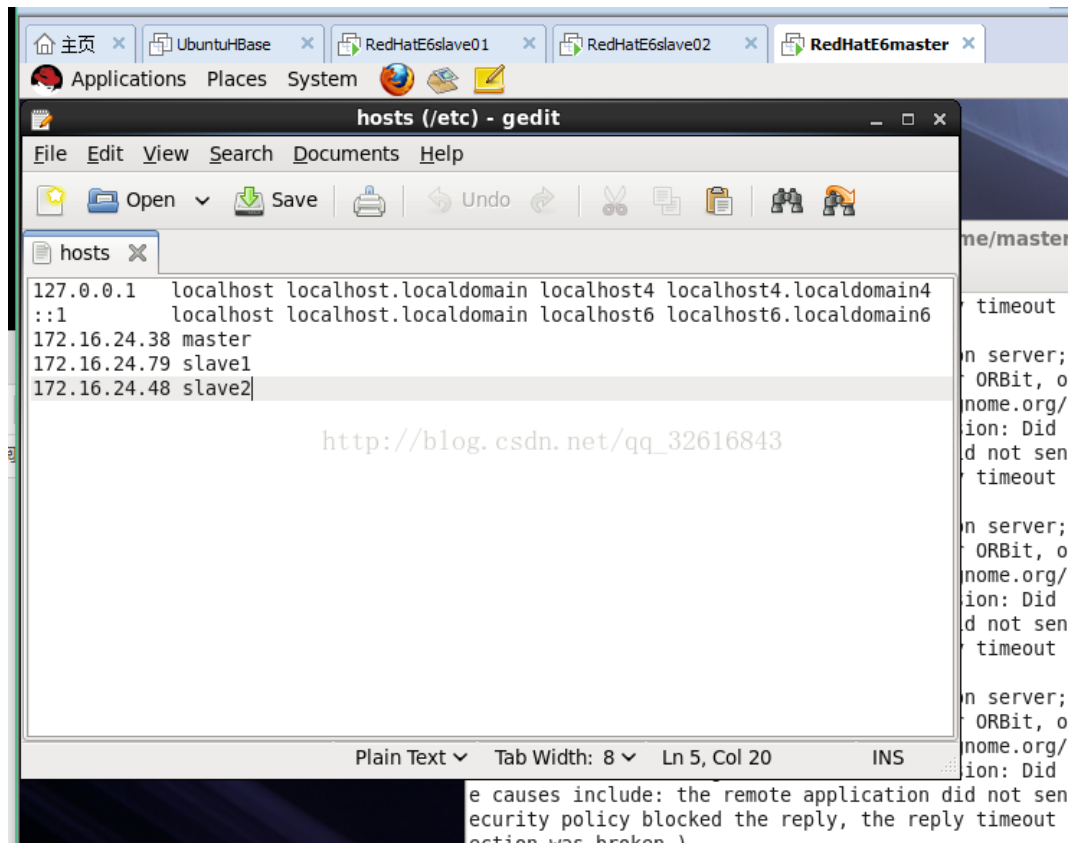
每台 PC 机已装有 RatHat6 系统，一台机命名为 master 作为 namenode，另两台为 slave1 和 slave2，作为 datanode。具体配置如下。

用户名	Host-IP	网关	HostName（互ping 时的机器名）	内存	硬盘	HDFS	YARN	备注
Master	172.16.24.38	172.16.24.254	master	2	40GB	NameNode	ResourcesManager	集群主节点
Slave01	172.16.24.79	172.16.24.254	slave1	1	20GB	DataNode	NodeManager	计算调度
Slave02	172.16.24.48	172.16.24.254	slave2	1	20GB	DataNode	NodeManager	数据计算节点

四、Hadoop 集群模式安装过程：

1.用 root 用户设置主机名和 IP 地址分配（/etc/hosts）

先在 master 机器上操作后复制到另两台机上



```

e causes include: the remote application did not send a reply, the message bus s
ecurity policy blocked the reply, the reply timeout expired, or the network conn
ection was broken.)
[root@master master]# scp /etc/hosts root@slave1:/etc/
root@slave1's password:
hosts                                100% 218      0.2KB/s   00:00
[root@master master]# scp /etc/hosts root@slave2:/etc/
The authenticity of host 'slave2 (172.16.24.48)' can't be established.
RSA key fingerprint is d6:bf:3a:11:e0:dd:8f:fa:e6:a9:d4:47:cd:6b:e9:9d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'slave2,172.16.24.48' (RSA) to the list of known host
s.
root@slave2's password:
hosts                                100% 218      0.2KB/s   00:00
[root@master master]#

```

笔记:修改系统文件时需要以 root 用户登陆

2.继续使用 root 用户创建一个 hadoop 账户并为其设置密码,专门负责操作与 hadoop 相关的业务。(图片以 master 机为例,另外两台机上做同样操作)

[命令] useradd hadoop

```

[root@master master]# useradd hadoop
[root@master master]# cd ~
[root@master ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@master ~]# cd /home
[root@master home]# ls
hadoop  master
[root@master home]#

```

笔记：用 root 创建用户会自动创建宿主目录

该步骤情况：三台机器都用 root 创建了 hadoop 用户（并且已默认创建了 hadoop 账户的宿主目录）

[命令] passwd hadoop

```
[hadoop@master home]$ su root
Password:
[root@master home]# passwd hadoop
Changing password for user hadoop.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@master home]#
```

tips：刚创建的用户需要用 root 用户为其设置密码

该步骤情况：现在三台机器都用 root 设置了 hadoop 的密码

3. 设置三台机器之间的免密码登陆：（使用 hadoop 用户）

3.1 在 master 机器上使用 hadoop 用户生成 master 机器节点的 hadoop 账户密钥对。

[命令] ssh-keygen -t rsa

可在 ~/.ssh 下查看生成的密钥对 id_rsa 和 id_rsa.pub

```
[hadoop@master ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
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:
db:4b:4e:55:7a:f3:3e:80:eb:6c:a8:76:19:ff hadoop@master
The key's randomart image is:
+---[ RSA 2048]-----+
|          .            |
|         o            |
|      S   o o         |
|      + ... o         |
|     . 0. ... .       |
|    . B.+ . . =       |
|   ..0 =+.Eoo+       |
+-----+
[hadoop@master ~]$
```

```
+-----+
[hadoop@master ~]$ ls /home/hadoop/.ssh
id_rsa id_rsa.pub known_hosts
[hadoop@master ~]$
```

3.2 继续在 master 机器上为 slave1 和 slave2 生成各自的密钥对。

```
ssh slave1 ssh-keygen -t rsa
ssh slave2 ssh-keygen -t rsa
```

```
scp hadoop@slave1:~/.ssh/id\_rsa.pub ~/.ssh/slave1.pub
```

```
scp hadoop@slave2:~/.ssh/id_rsa.pub ~/.ssh/slave2.pub
```

```
cat ~/.ssh/*.pub >> ~/.ssh/authorized_keys
```

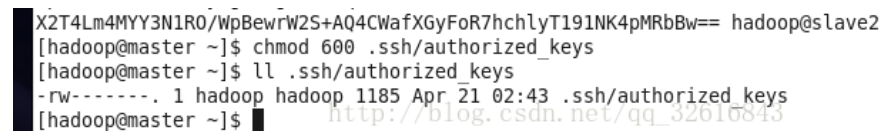
(查看文件命令)cat ~/.ssh/authorized_keys



```
[hadoop@master ~]$ scp hadoop@slave1:~/.ssh/id_rsa.pub ~/.ssh/slave1.pub
hadoop@slave1's password:
id_rsa.pub                                100% 395      0.4KB/s   00:00
[hadoop@master ~]$ scp hadoop@slave2:~/.ssh/id_rsa.pub ~/.ssh/slave2.pub
hadoop@slave2's password:
id_rsa.pub                                100% 395      0.4KB/s   00:00
[hadoop@master ~]$ cat ~/.ssh/*.pub > ~/.ssh/authorized_keys
[hadoop@master ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAuPdfKcPPk0D0NrxZgKnwxjign/hrGGx4GNMMIGGc0x5x
ytCZdQ2dc101sX228ES2Bx9fGraVU3c9rKFfng5cR4nNw5kbMwNxe6KvwRBMPc5Mxcye0KTAa1qybS9o
DXuR05NzdeQ2e9XAKG+WkR0sZlqGV9KxJ0mwPSzR6dEBINvc4kUFdUn6ZWwG8KM/b7ecOU2nFaYVsXRr
i6GSR/paUtt/vApu2wM2czJ+aHuirivD9lN14XWLJlcFUgo9lizFSzr+l9Tg/QmXauFk5JbZym2vNcuu
uoa+D7CcwL6h8UIUGSu8XThnG0illbgaQEDMDijCRZplca5TdCmYrkV+w== hadoop@master
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAAPQjvEZwVBY1nlsSZhGcVID7Gr+rNHqdMFPX0b10b8E
RSAqynk/gPsb0CSM80JGhktNXeYEIE78R0lw0oBXgb1aPaoI0p1QJmH0LeLP+bzXQFNPMwuFRerrNzG
9G5DsEzco299fJLU99p3lQjITeQ0MpFraLlvqNSI+6Hgc9Dpl0DqzmH2wLPZ6G5vyjq20Qr+EI504q1f
nwhQs1zCwsJfilGotT7MDC9JTZdqbrDRkHepYk4rDQxhxWR0Ts14QfEHtRmfF6P8uRJwi+H3/mqE6S/B
eLp8iIAR13bY3V6vPDADxxd0XKk1Lhv7kZkQ108ZwYwLBp7pJyeelyZmQ== hadoop@slave1
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAU/UEcZqzKKG18eHL6lgVlTw3W2G50wm06X0G5rgTQHQI
FvxghDALf11T6dr4ENvAwzh8AR+nX8VrX0bYf0Vvv1etBPAPgVKTsLaFENLb1RwbBh1XD06EvL4C1iSq
pNNZHnqeItd+IgRMlDzjyUfpLTNP+8ahr3Lp7pIvwF2GF7f0ZopCzu/6P5QmqywZftB+fJ47MgVYgI5S
KpZ0UchJIVPufeEyg5xJgYGEaxqCH9U3EBN6G5M60Ywids1Z3C35CKWoCfMzf9kXIFzwVb/on30bCs
X2T4Lm4MY3N1R0/WpBewrW2S+AQ4CWafXGyFor7hchlyT191NK4pMRbBw== hadoop@slave2
[hadoop@master ~]$
```

3.4 出于安全性考虑, 将这个授权 key 文件 authorized_keys 赋予 600 权限

[命令]chmod 600 .ssh/authorized_keys



```
X2T4Lm4MY3N1R0/WpBewrW2S+AQ4CWafXGyFor7hchlyT191NK4pMRbBw== hadoop@slave2
[hadoop@master ~]$ chmod 600 .ssh/authorized_keys
[hadoop@master ~]$ ll .ssh/authorized_keys
-rw-----. 1 hadoop hadoop 1185 Apr 21 02:43 .ssh/authorized_keys
[hadoop@master ~]$
```

3.5 将这个包含了所有互信机器认证 authorized_keys 认证文件复制到所有节点主机的 ~/.ssh/ 目录中, 并进行验证互信。

[命令]

```
scp ~/.ssh/authorized_keys hadoop@slave1:~/.ssh
```

```
scp ~/.ssh/authorized_keys hadoop@slave2:~/.ssh
```

(测试免密码链接)

```
ssh slave1
```

```
ssh slave2
```

```

-rw-----. 1 hadoop hadoop 1185 Apr 21 02:43 .ssh/authorized_keys
[hadoop@master ~]$ scp ~/.ssh/authorized_keys hadoop@slave1:~/.ssh/
hadoop@slave1's password:
authorized_keys                                100% 1185      1.2KB/s   00:00
[hadoop@master ~]$ scp ~/.ssh/authorized_keys hadoop@slave2:~/.ssh/
hadoop@slave2's password:
authorized_keys                                100% 1185      1.2KB/s   00:00
[hadoop@master ~]$ ssh slave1
[hadoop@slave1 ~]$ exit
logout
Connection to slave1 closed.
[hadoop@master ~]$ ssh slave2
[hadoop@slave2 ~]$ exit
logout
Connection to slave2 closed.
[hadoop@master ~]$

```

该步骤情况：现在三台机器已经可以免密码互相登陆

4.Java 的安装与配置

（这里使用版本为 jdk1.8）

4.1 创建统一管理 java 和 hadoop 的父级目录，位于 hadoop 用户主目录下/home/hadoop,如网上教程命名为 chadoop

[命令]mkdir ~/chadoop

4.2 解压 jdk 安装包（之前已使用 VMware Tools 拉近 master 机器内）

[命令]

tar xzfjdk 安装包名

（在 chadoop 中创建文件夹 java 放置 jdk 解压后的文件）

mkdir ~/chadoop/java

mv jdk 解压后的文件名 ~/chadoop/java

```

-rwxrwx-rw-. 1 master master 175271626 Oct 5 2017 jdk-8u45-linux-x64.tar.gz
[hadoop@master softpage]$ cp jdk-8u45-linux-x64.tar.gz /home/hadoop
[hadoop@master softpage]$ ls /home/hadoop
chadoop  jdk-8u45-linux-x64.tar.gz
[hadoop@master softpage]$ cd /home/hadoop
[hadoop@master ~]$ ls
chadoop  jdk-8u45-linux-x64.tar.gz
[hadoop@master ~]$ tar xzf jdk-8u45-linux-x64.tar.gz
[hadoop@master ~]$ ls
chadoop  jdk1.8.0_45  jdk-8u45-linux-x64.tar.gz
[hadoop@master chadoop]$ ls
dfs  java  tmp
[hadoop@master chadoop]$

```

4.3 修改环境变量（~/.bash_profile）加入 JAVA_HOME,CLASSPATH 和 PATH

使用 vi 编辑器进行修改，配置后使用 . ~/.bash_profile 立即生效。并使用 java -version 命令进行检测 java 是否安装成功。

[命令] vi ~/.bash_profile

[增加配置信息]（由于参考教程同时也配置了 hadoop_home 的位置，这里也先配置也可安装 hadoop 后再做对该步骤的 hadoop 操作）

JAVA&HADOOP-CONFIGURATION#

export JAVA_HOME=~/chadoop/java/jdk1.8.0_45


```
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export HADOOP_HOME=~/.chadoop/hadoop/hadoop-2.7.3
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

[命令]

```
java -version
```

该步骤情况：现在 master 机器上已安装好 jdk1.8 版本。

(这里使用版本为 `hadoop2.7.3`)

5.2 在 chadoop 文件夹内新建一个 hadoop 目录用于放置 hadoop 安装包解压后文件

[命令]

tar xzf hadoop 安装包

mkdir ~/chadoop/hadoop

mv hadoop 解压后文件 ~/chadoop/Hadoop

5.3 创建 hadoop 相关的 tmp 目录和 dfs 目录（以及其下的 name 和 data 目录）

[命令]

mkdir ~/chadoop/tmp

mkdir -p ~/chadoop/dfs/name ~/chadoop/dfs/data

```
connection to slave2 closed.
[hadoop@master ~]$ mkdir ~/chadoop
[hadoop@master ~]$ mkdir ~/chadoop/tmp
[hadoop@master ~]$ mkdir -p ~/chadoop/dfs/name ~/chadoop/dfs/data
[hadoop@master ~]$ ll
total 4
drwxrwxr-x. 4 hadoop hadoop 4096 Apr 21 02:47 chadoop
[hadoop@master ~]$ cd chadoop
[hadoop@master chadoop]$ ls
dfs tmp
[hadoop@master chadoop]$ cd dfs
[hadoop@master dfs]$ ls
data name
bash: cd: Command not found
[hadoop@master chadoop]$ ls
dfs hadoop java tmp
[hadoop@master chadoop]$ cd hadoop
[hadoop@master hadoop]$ ls
hadoop-2.7.3
[hadoop@master hadoop]$
```

5.4 为 hadoop 配置环境变量（~/bash_profile）

这一步已在 java 配置环境变量时操作，具体见 4.3.使用 hadoop version 进行验证 hadoop 是否安装成功

[命令] hadoop version

```
[hadoop@master ~]$ . ~/.bash_profile
[hadoop@master chadoop]$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
[hadoop@master chadoop]$
```

该步骤情况：master 机器上已成功安装了 hadoop2.7.3 版本

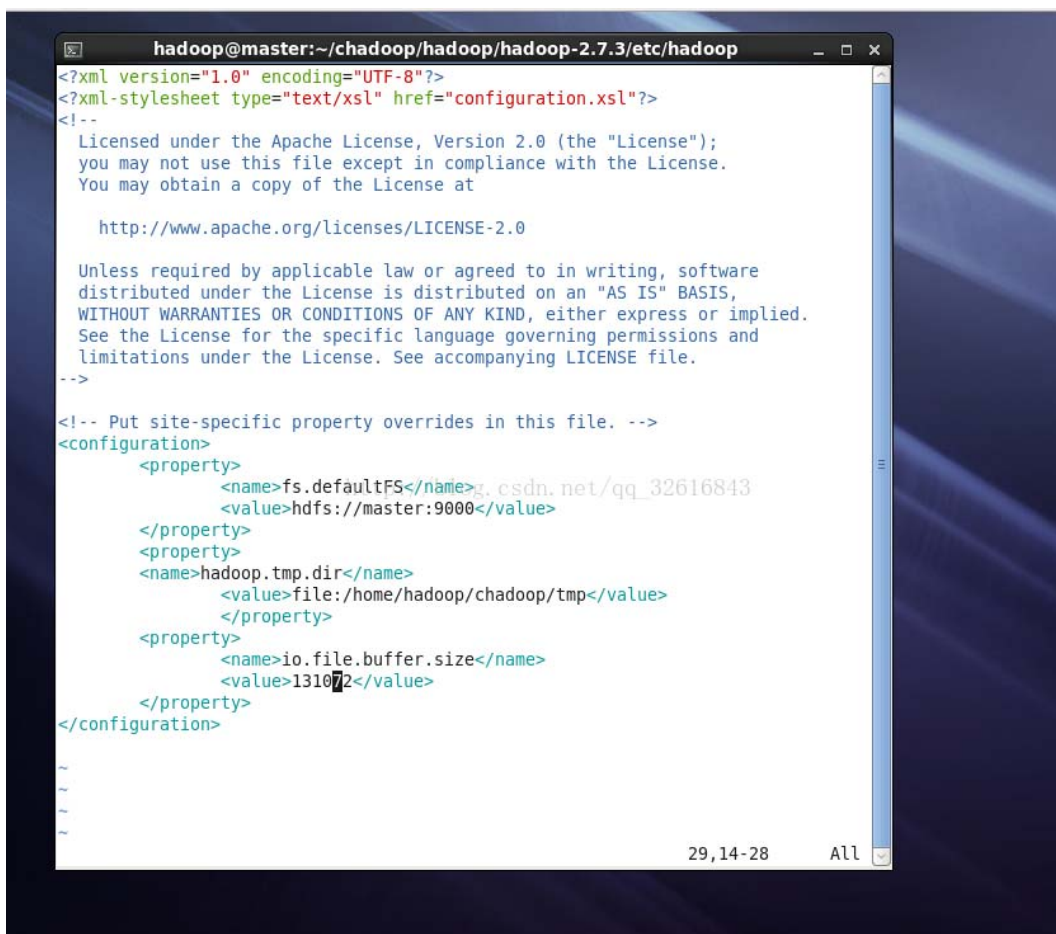
6.修改 hadoop 内置文件，配置集群模式

涉及修改文件：core-site.xml,hdfs-site.xml, mapred-site.xml,yarn-site.xml,hadoop-env.sh, mapred-env.sh,yarn-env.sh 和 slaves（均位于\$HADOOP_HOME 下的/etc/hadoop 文件夹内）

各文件配置如下：

(1) [core-site.xml 配置]

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/hadoop/chadoop/tmp</value>
</property>
<property>
<name>io.file.buffer.size</name>
<value>131072</value> //如下配置是读写 sequence file 的 buffer size,可减少 I/O 次数。在
大型的 Hadoop cluster, 建议可设定为 65536 到 131072, 默认值 4096.按照教程配置了
131702
</property>
</configuration>
```



```
hadoop@master:~/chadoop/hadoop/hadoop-2.7.3/etc/hadoop
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>fs.defaultFS</name>g.csdn.net/qq_32616843
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/home/hadoop/chadoop/tmp</value>
  </property>
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
</configuration>
```

(2) [hdfs-site.xml 配置]

```
<configuration>
```

```
<property>
<name>dfs.namenode.name.dir</name>
<value>/home/hadoop/chadoop/dfs/name</value>
<description>namenode 的目录位置</description>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/hadoop/chadoop/dfs/data</value>
<description>datanode 的目录位置</description>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
<description>hdfs 系统的副本数量</description>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>master:9001</value>
<description>备份 namenode 的 http 地址</description>
</property>
<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
<description>hdfs 文件系统的 webhdfs 使能标致</description>
</property>
</configuration>
```

```
hadoop@master:~/chadoop/hadoop/hadoop-2.7.3/etc/hadoop
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/home/hadoop/chadoop/dfs/name</value>
    <description>namenode的目录位置</description>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/hadoop/chadoop/dfs/data</value>
    <description>datanode的目录位置</description>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
    <description>hdfs系统的副本数量</description>
  </property>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>master:9001</value>
    <description>备份 namenode的 http地址</description>
  </property>
  <property>
    <name>dfs.webhdfs.enabled</name>
    <value>true</value>
    <description>hdfs文件系统的 webhdfs使能标致</description>
  </property>
</configuration>
```

(3) [mapred-site.xml 配置]

注意: mapred-site.xml 需要先复制模板生成配置文件后修改内容

[命令]cp mapred-site.xml.template mapred-site.xml

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
<description>指明 MapReduce 的调度框架为 yarn</description>
</property>
<property>
<name>mapreduce.jobhistory.address</name>
<value>master:10020</value>
<description>知名 MapReduce 的作业历史地址</description>
</property>
<property>
<name>mapreduce.jobhistory.webapp.address</name>
<value>master:19888</value>
<description>指明 MapReduce 的作业历史 web 地址</description>
</property>
</configuration>
```

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
    <description>指明 MapReduce 的调度 框架为 yarn</description>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value> master:10020</value>
    <description>知名 MapReduce 的作业 历史 地址</description>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>master:19888</value>
    <description>指明 MapReduce 的作业 历史 web 地址</description>
  </property>
</configuration>

:wq!
```

(4) [yarn-site.xml 配置]

```
<configuration>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:18040</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:18030</value>
</property>
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>master:18088</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:18025</value>
</property>
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>master:18141</value>
</property>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

```
<configuration>
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>master:18040</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:18030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>master:18088</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>master:18025</value>
  </property>
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>master:18141</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
:wq!
```

(5) [hadoop-env.sh 配置]

[操作]加入 JAVA_HOME 位置

```
hadoop@master:~/chadoop/hadoop/hadoop-2.7.3/etc/hadoop
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.
# http://blog.csdn.net/qq_32616843
# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
#export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/home/hadoop/chadoop/java/jdk1.8.0_45

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

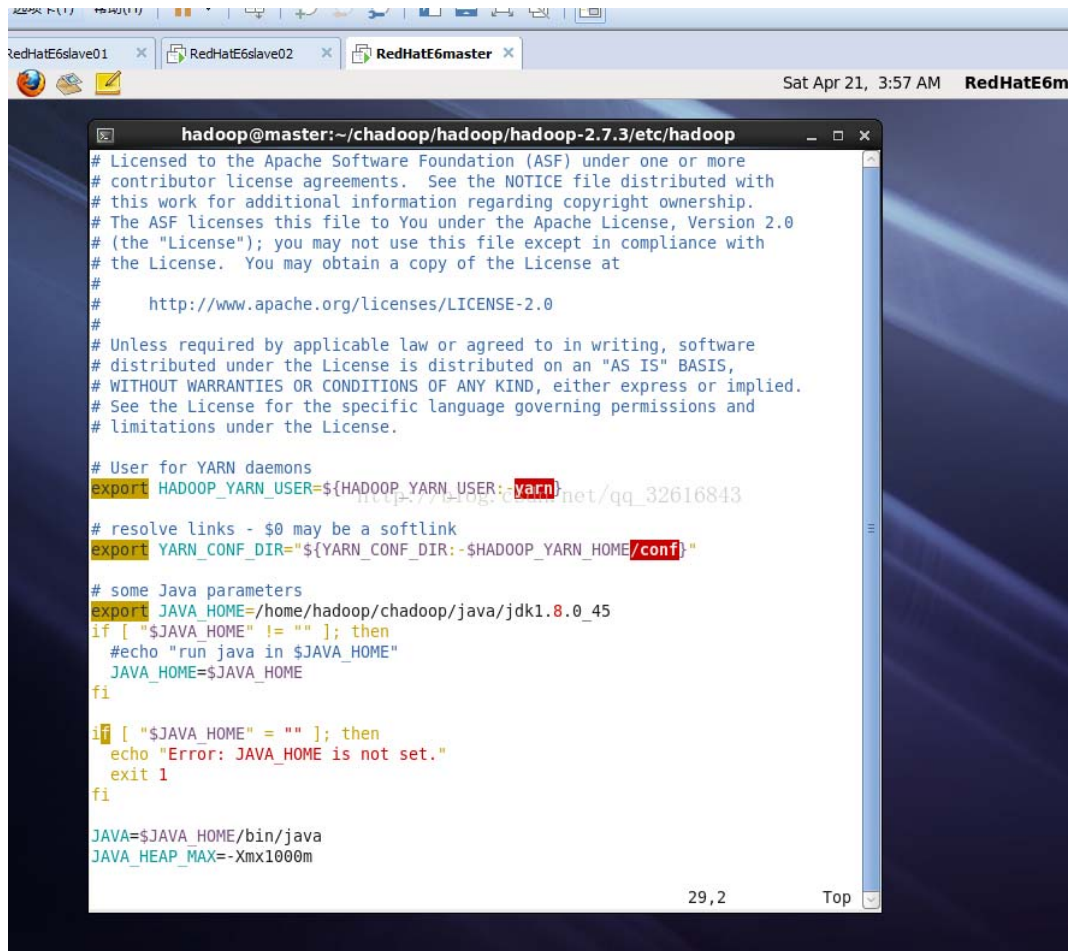
# Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
27,0-1 Top
```

(6) [mapred-env.sh 配置]
[操作]指明 JAVA_HOME 位置


```
hadoop@master:~/chadoop/hadoop/hadoop-2.7.3/etc/hadoop
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# 在这里指明 Java SDK的home目录
export JAVA_HOME=/home/hadoop/chadoop/java/jdk1.8.0_45
#
export HADOOP_JOB_HISTORYSERVER_HEAPSIZE=1000
export HADOOP_MAPRED_ROOT_LOGGER=INFO,RFA
#export HADOOP_JOB_HISTORYSERVER_OPTS=
#export HADOOP_MAPRED_LOG_DIR="" # Where log files are stored. $HADOOP_MAPRED_H
OME/logs by default.
#export HADOOP_JHS_LOGGER=INFO,RFA # Hadoop JobSummary logger.
#export HADOOP_MAPRED_PID_DIR= # The pid files are stored. /tmp by default.
#export HADOOP_MAPRED_IDENT_STRING= #A string representing this instance of hadop. $USER by default
#export HADOOP_MAPRED_NICENESS= #The scheduling priority for daemons. Defaults t
o 0.
~
~
~
~
:wq!
```

(7) [yarn-env.sh 配置]

[操作]加入 JAVA_HOME 位置



```
hadoop@master:~/chadoop/hadoop/hadoop-2.7.3/etc/hadoop
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# User for YARN daemons
export HADOOP_YARN_USER=${HADOOP_YARN_USER:yarn}

# resolve links - $0 may be a softlink
export YARN_CONF_DIR="${YARN_CONF_DIR:-$HADOOP_YARN_HOME/conf}"

# some Java parameters
export JAVA_HOME=/home/hadoop/chadoop/java/jdk1.8.0_45
if [ "$JAVA_HOME" != "" ]; then
    #echo "run java in $JAVA_HOME"
    JAVA_HOME=$JAVA_HOME
fi

if [ "$JAVA_HOME" = "" ]; then
    echo "Error: JAVA_HOME is not set."
    exit 1
fi

JAVA=$JAVA_HOME/bin/java
JAVA_HEAP_MAX=-Xmx1000m
```

(8) [slaves 配置]

[操作]加入两个节点的名称



```
hadoop@master:~/chadoop/hadoop/hadoop-2.7.3/etc/hadoop
slave1
slave2
```

7.将 master 主节点以上的配置复制到 slave1、slave2 节点

7.1 复制环境变量文件并使用各节点对象进行环境变量生效

[命令]

```
scp ~/.bash_profile hadoop@slave1:~/
```

```
scp ~/.bash_profile hadoop@slave2:~/
```

```
ssh hadoop@slave1. ~/.bash_profile
```

```
ssh hadoop@slave2. ~/.bash_profile
```

```
hadoop-metrics2.properties kms-log4j.properties ssl-server.xml.example
hadoop-metrics.properties kms-site.xml yarn-env.cmd
hadoop-policy.xml log4j.properties yarn-env.sh
hdfs-site.xml mapred-env.cmd yarn-site.xml
[hadoop@master hadoop]$ vi mapred-site.xml
[hadoop@master hadoop]$ vi yarn-site.xml
[hadoop@master hadoop]$ vi yarn-site.xml
[hadoop@master hadoop]$ vi hadoop-env.sh
[hadoop@master hadoop]$ vi hadoop-env.sh
[hadoop@master hadoop]$ vi mapred-env.sh
[hadoop@master hadoop]$ vi mapred-env.sh
[hadoop@master hadoop]$ vi yarn-env.sh
[hadoop@master hadoop]$ vi yarn-env.sh
[hadoop@master hadoop]$ vi slaves
[hadoop@master hadoop]$ cd
[hadoop@master ~]$ scp .bash_profile hadoop@slave1:~/
.bash_profile 100% 504 0.5KB/s 00:00
[hadoop@master ~]$ scp .bash_profile hadoop@slave2:~/
.bash_profile 100% 504 0.5KB/s 00:00
[hadoop@master ~]$
```

```
[hadoop@master ~]$ ssh hadoop@slave1 . ~/.bash_profile
[hadoop@master ~]$ ssh hadoop@slave2 . ~/.bash_profile
[hadoop@master ~]$
```

7.2 复制 chadoop 目录到 slave1 和 slave2 机器上

[命令]

scp -r chadoop/ hadoop@slave1:~

scp -r chadoop/ hadoop@slave2:~

```
SerialUtils.hh 100% 4514 4.4KB/s 00:00
hdfs.h 100% 33KB 32.6KB/s 00:00
StringUtils.hh 100% 2441 2.4KB/s 00:00
TemplateFactory.hh 100% 3319 3.2KB/s 00:00
[hadoop@master ~]$ scp -r chadoop/ hadoop@slave2:~
```

8.在 master 主节点上格式化 hdfs 文件系统

[命令]hdfs namenode -format

```
StringUtils.hh 100% 2441 2.4KB/s 00:00
TemplateFactory.hh 100% 3319 3.2KB/s 00:00
[hadoop@master ~]$ hdfs namenode -format
18/04/21 04:06:03 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = master/172.16.24.38
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.7.3
STARTUP_MSG: classpath = /home/hadoop/chadoop/hadoop/hadoop-2.7.3/etc/hadoop:/
/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/lib/jetty-6.1.26.jar
:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/lib/api-asn1-api-1
.0.0-M20.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common/lib/ne
tty-3.6.2.Final.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/common
/lib/zookeeper-3.4.6.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/c
ommon/lib/gson-2.2.4.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/share/hadoop/c
ommon/lib/slf4j-log4j12-1.7.10.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/shar
e/hadoop/common/lib/commons-io-2.4.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3/
share/hadoop/common/lib/jsch-0.1.42.jar:/home/hadoop/chadoop/hadoop/hadoop-2.7.3
/share/hadoop/common/lib/jersey-json-1.9.jar:/home/hadoop/chadoop/hadoop/hadoop-
2.7.3/share/hadoop/common/lib/slf4j-api-1.7.10.jar:/home/hadoop/chadoop/hadoop/h
adoop-2.7.3/share/hadoop/common/lib/servlet-api-2.5.jar:/home/hadoop/chadoop/had
```

```

18/04/21 04:06:04 INFO util.GSet: computing capacity for map namenode.retrycache
18/04/21 04:06:04 INFO util.GSet: VM type = 64-bit
18/04/21 04:06:04 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 29
7.0 KB
18/04/21 04:06:04 INFO util.GSet: capacity = 2^15 = 32768 entries
18/04/21 04:06:04 INFO namenode.FSImage: Allocated new BlockPoolId: BP-304055169
-172.16.24.38-1524308764036
18/04/21 04:06:04 INFO common.Storage: Storage directory /home/hadoop/chadoop/dfs/
name has been successfully formatted.
18/04/21 04:06:04 INFO namenode.FSImageFormatProtobuf: Saving image file /home/h
adoop/chadoop/dfs/name/current/fsimage.ckpt_000000000000000000 using no compres
sion
18/04/21 04:06:04 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/c
hadoop/dfs/name/current/fsimage.ckpt_000000000000000000 of size 353 bytes saved
in 0 seconds.
18/04/21 04:06:04 INFO namenode.NNStorageRetentionManager: Going to retain 1 ima
ges with txid >= 0
18/04/21 04:06:04 INFO util.ExitUtil: Exiting with status 0
18/04/21 04:06:04 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/172.16.24.38
*****/
[hadoop@master ~]$

```

9.关闭防火墙（现在主节点上使用 root 用户操作，关闭后再使用 ssh 命令进入另外 2 节点中关闭其余节点防火墙）

[命令]

service iptables stop

chkconfig iptables off

ssh slave1 进入后操作与 master 机器一样

ssh slave2 进入后操作与 master 机器一样

```

[hadoop@master ~]$ su
Password:
[root@master hadoop]# service iptables stop
iptables: Flushing firewall rules: [ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Unloading modules: [ OK ]
[root@master hadoop]# chkconfig iptables off
[root@master hadoop]#
[root@master hadoop]# ssh slave1
root@slave1's password:
[root@slave1 ~]# service iptables stop
iptables: Flushing firewall rules: [ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Unloading modules: [ OK ]
[root@slave1 ~]# chkconfig iptables off
[root@slave1 ~]# ssh slave2
root@slave2's password:
Last login: Fri Apr 6 09:55:05 2018 from localhost
[root@slave2 ~]# service iptables stop
iptables: Flushing firewall rules: [ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Unloading modules: [ OK ]
[root@slave2 ~]# chkconfig iptables off
[root@slave2 ~]#

```

10.在 master 机器上启动 hadoop，并用 jps 检验 hadoop 进程（此时 master 主节点有 4 个 ResourceManager,Jps, NameNode, SecondaryNamenode，slave1 节点与 slave2 节点有 3 个 NodeManager,DataNode, Jps）

[命令]

start-all.sh

jps

```
[hadoop@master ~]$ jps
9136 SecondaryNameNode
8976 NameNode
9287 ResourceManager
9548 Jps
[hadoop@master ~]$ ssh slave1 jps
bash: jps: command not found
[hadoop@master ~]$ ssh slave1
Last login: Sat Apr 21 10:21:54 2018 from master
[hadoop@slave1 ~]$ jps
6355 Jps
6135 DataNode
6218 NodeManager
[hadoop@slave1 ~]$ ssh slave2
Last login: Sat Apr 21 08:58:30 2018 from slave1
[hadoop@slave2 ~]$ jps
5994 DataNode
6203 Jps
6077 NodeManager
```

11.hadoop 的停止

[命令]stop-all.sh

```
Last login: Sat Apr 21 08:13:30 2018 from slave1
[hadoop@master ~]$ stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
Stopping namenodes on [master]
master: stopping namenode
slave1: stopping datanode
slave2: stopping datanode
Stopping secondary namenodes [master]
master: stopping secondarynamenode
stopping yarn daemons
stopping resourcemanager
slave1: stopping nodemanager
slave2: stopping nodemanager
no proxyserver to stop
[hadoop@master ~]$
```

```
9
no proxyserver to stop
[hadoop@master ~]$ jps
8496 Jps
[hadoop@master ~]$ ssh slave1
Last login: Sat Apr 21 08:20:47 2018 from master
[hadoop@slave1 ~]$ jps
5840 Jps
[hadoop@slave1 ~]$ ssh slave2
Last login: Sat Apr 21 08:21:18 2018 from slave1
[hadoop@slave2 ~]$ jps
5798 Jps
[hadoop@slave2 ~]$
```

该步骤情况：到这里 hadoop 的基本安装与配置结束。[hadoop 集群已基本搭建完成](http://blog.csdn.net/qq_32616843)。

12.增加内容

再次登陆时发现环境变量没有生效，于是将`~/.bash_profile`中的增加内容同时对`~/.bashrc`和`/etc/profile`都做了更改，复制到另外两台机器上。这里请注意修正。