

## 项目概述

---

1. 项目类别：Hadoop环境搭建

2. 学时要求：4

3. 组织形式：单人完成

本项目分为5个部分，分为5个子项目。本项目为项目1，内容为：每个学生搭建一个包括3个节点的分布式集群。

## 项目目的

---

1. 掌握Linux基本命令。

2. 了解SSH免密登录的原理以及进行SSH免密登录的原因。

3. 运用SSH工具进行集群中各个主机的免密登录配置。

4. 运用Hadoop集群安装软件进行集群的安装与配置。

## 项目要求

---

1. 创建Hadoop用户。

2. 机器名及域名配置。

3. 安装JDK

4. SSH安装与配置

5. Hadoop集群安装与配置

6. 测设Hadoop环境

## 项目环境

---

1. 硬件环境：每台计算机内存8G，2个四核CPU，硬盘容量T级。每台计算机配置静态IP。

2. 软件：每台计算机安装ubuntu-16.04.2操作系统

# 相关知识

---

## 1 Linux基本命令

1.1 ls: 显示文件或目录

1.2 mkdir: 创建目录

1.3 cd: 切换目录

1.4 touch: 创建空文件

1.5 echo: 创建带有内容的文件。

1.6 cat: 查看文件内容

1.7 cp: 复制文件或目录

1.8 mv: 移动文件或修改文件名

1.9 rm: 删除文件

1.10 grep: 在文本文件中查找某个字符串

1.11 rmdir: 删除空目录

1.12 pwd: 显示当前目录

1.13 ln: 创建链接文件

1.14 more、less: 分页显示文本文件内容

1.15 head、tail: 显示文件头、尾内容

1.16 ifconfig: 查看网络情况

1.17 ping: 测试网络连通

1.18 ps: 显示瞬间进程状态

1.19 kill: 杀死进程, 可以先用ps 或 top命令查看进程的id, 然后再用kill命令杀死进程。

1.20 useradd: 创建用户

1.21 adduser: 将用户添加到组 (权限管理)

1.22 tar: 打包解压文件

1.23 sudo chown [-R] owner[:group] {File|Directory}

1.24 sudo chmod [u g o a] [+增加权限 -减少权限] [r w x] 目录名

## 2 SSH工作原理

2.1 SSH是什么?

SSH 为 Secure Shell 的缩写, 是建立在应用层和传输层基础上的安全协议。SSH 是目前较可靠、专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH最初是 UNIX系统上的一个程序, 后来又迅速扩展到其他操作平台。SSH是由[客户端](#)和[服务端](#)的软件组成, 服务端是一个守护进程(daemon), 它在后台运行并响应来自客户端的连接请求, 客户端包含ssh程序以及像scp (远程拷贝) 、slogin (远程登陆) 、sftp (安全文件传输) 等其他的应用程序

2.2 配置SSH的原因:

Hadoop名称节点 (NameNode) 需要启动集群中所有机器的Hadoop守护进程, 这个过程需要通过SSH登录来实现。Hadoop并没有提供SSH输入密码登录的形式, 因此, 为了能够顺利登录每台机器, 需要将所有机器配置为名称节点可以无密码登录它们。

## 3 Hadoop架构

一个基本的Hadoop集群中的节点主要有:

- NameNode: 负责协调集群中的数据存储
- DataNode: 存储被拆分的数据块
- SecondaryNameNode: 帮助NameNode收集文件系统运行的状态信息  
SecondaryNameNode在小型集群中可以和NameNode共用一台机器, 较大的群集可以采用与NameNode相同的硬件。

# 项目内容

---

## 1 创建Hadoop用户, 并为hadoop用户授权

【注意】以下标记[A,B]是指在A和B机器上执行的步骤，同理[A],[A,B,C]的写法

### 1.1 [A]进入终端

在远程桌面上点击右键，选择“打开终端”

### 1.2 [A]切换到root用户

进入终端之后，默认是ub用户，切换命令为：

```
su
```

如果出现认证失败的错误，说明root用户没有设置初始密码，可以执行

```
sudo passwd root
```

然后输入当前用户的密码

再两次输入root的密码

然后再次执行

```
su
```

然后输入刚刚设置的root密码

### 1.3 [A]执行下列名列查看hadoop用户是否存在。

```
cat /etc/passwd |grep hadoop
```

### 1.4 [A]创建用户hadoop，并设置密码为hadoop。-m建立用户登录的目录，-s定义用户登录后的shell

```
useradd -m hadoop -s /bin/bash
```

```
passwd hadoop
```

设置hadoop用户密码为 hadoop

1.5 [A]为hadoop用户授予sudo权限。

```
adduser hadoop sudo
```

1.6 [A]修改机器名字, 【注意】本步骤由于桌面云平台控制, 本步骤跳过  
使用自己的姓名全拼加上数字, 例如zhangsan01、zhangsan02、  
zhangsan03, 命令如下:

```
sudo gedit /etc/hostname
```

修改名字之后重启一下主机, 使用:

```
reboot
```

1.7 在另外两台主机上完成相同的操作

- (1) [B]第二台机器上重复上面的操作 (步骤1.1~1.6): 创建用户、设置密码、设置权限、修改机器名(姓名02)、重启
- (2) [C]第三台机器上重复上面的操作 (步骤1.1~1.6): 创建用户、设置密码、设置权限、修改机器名(姓名03)、重启

## 2 添加域名映射

2.1 [A,B,C]执行ifconfig命令, 查看当前主机IP地址。

```
root@lixucheng01: /home/ub
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@lixucheng01:/home/ub# sudo gedit /etc/hostname

** (gedit:3548): WARNING **: 16:56:27.728: Set document metadata failed: 不支持
设置属性 metadata::gedit-position
root@lixucheng01:/home/ub# ifconfig
enp2s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.22.252.3 netmask 255.255.252.0 broadcast 172.22.255.255
        inet6 fe80::f816:3eff:fed3:98d1 prefixlen 64 scopeid 0x20<link>
            ether fa:16:3e:d3:98:d1 txqueuelen 1000 (以太网)
            RX packets 5457 bytes 1296203 (1.2 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 5263 bytes 421618 (421.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

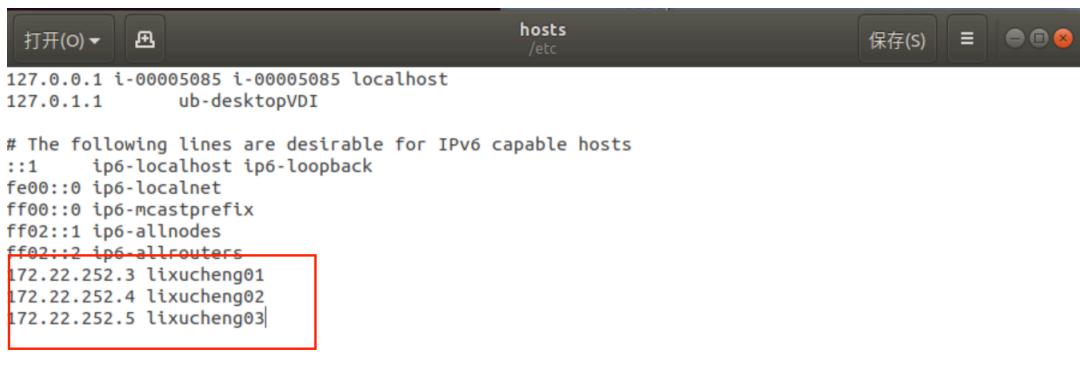
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (本地环回)
            RX packets 146 bytes 11584 (11.5 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 146 bytes 11584 (11.5 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@lixucheng01:/home/ub#
```

2.2 [A]将集群中所有节点的IP 地址与主机名写入所有主机的/etc/hosts中，完成域名映射的添加。

```
gedit /etc/hosts
```

例如：



```
hosts
/etc
保存(S) 退出(X)

127.0.0.1 i-00005085 i-00005085 localhost
127.0.1.1 ub-desktopVDI

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.22.252.3 lixucheng01
172.22.252.4 lixucheng02
172.22.252.5 lixucheng03
```

【注意】hosts文件的第一行 仅保留 127.0.0.1 localhost 如果中间有其他内容，全部删除

2.3 在两外两台主机上也完成相同的操作

(1) [B]在第二台主机上，编辑文件，把3行复制过去，保存

(2) [C]在第三台主机上, 编辑文件, 把3行复制过去, 保存

### 3 SSH登录权限设置

3.1 [A]切换到用户hadoop

```
$ su hadoop
```

3.2 在集群所有节点上生成公钥和私钥。

(1) [A]在第一个节点上操作

```
$ ssh-keygen -t rsa
```

将在~/目录下自动创建目录.ssh, 内部创建id\_rsa (私钥) 和id\_rsa.pub (公钥) 。

例如:

(2) [B]在第二个节点上完成上面的操作

(3) [C]在第三个节点上完成上面的操作

3.3 在集群的各个节点上执行下列命令:

(1) [A,B,C]安装ssh服务: 分别在3台主机上完成, 注意此时当前用户应为hadoop

更新软件

```
sudo apt-get update
```

更新客户端版本:

```
sudo apt-get install openssh-client=1:7.6p1-4
```

安装服务器:

```
sudo apt-get install openssh-server
```

(2) [A,B,C]复制公钥到3个主机：分别在3台主机上操作，一共复制9次

```
$cd ~/.ssh
```

```
$ssh-copy-id -i id_rsa.pub hadoop@lixucheng01  
$ssh-copy-id -i id_rsa.pub hadoop@lixucheng02  
$ssh-copy-id -i id_rsa.pub hadoop@lixucheng03
```

例如：

```
hadoop@lixucheng01:~/ssh$ ssh-copy-id -i id_rsa.pub hadoop@lixucheng01  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "id_rsa.pub"  
The authenticity of host 'lixucheng01 (172.22.252.3)' can't be established.  
ECDSA key fingerprint is SHA256:7gTMslMKEviD242bE/7hFLEGFebmKt0xrAj9DwtCL3c.  
Are you sure you want to continue connecting (yes/no)? yes  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter  
out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt  
ed now it is to install the new keys  
hadoop@lixucheng01's password: ←  
Number of key(s) added: 1  
输入口令，不显示  
Now try logging into the machine, with: "ssh 'hadoop@lixucheng01'"  
and check to make sure that only the key(s) you wanted were added.
```

```
hadoop@lixucheng01:~/ssh$ ssh-copy-id -i id_rsa.pub hadoop@lixucheng02  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "id_rsa.pub"  
The authenticity of host 'lixucheng02 (172.22.252.4)' can't be established.  
ECDSA key fingerprint is SHA256:vxAYwse1Av+hCfpLm+c3xsH+talW327VTrNpwkkBq98.  
Are you sure you want to continue connecting (yes/no)? yes  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter  
out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt  
ed now it is to install the new keys  
hadoop@lixucheng02's password:  
Number of key(s) added: 1  
Now try logging into the machine, with: "ssh 'hadoop@lixucheng02'"  
and check to make sure that only the key(s) you wanted were added.
```

```
hadoop@lixucheng01:~/ssh$ ssh-copy-id -i id_rsa.pub hadoop@lixucheng03
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "id_rsa.pub"
The authenticity of host 'lixucheng03 (172.22.252.5)' can't be established.
ECDSA key fingerprint is SHA256:QQ8ZjxFJ3Fek42BUxWYELBmrQLvGMjnIfSZcBHGlTpU.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
hadoop@lixucheng03's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'hadoop@lixucheng03'"
and check to make sure that only the key(s) you wanted were added.
```

3.4 [A,B,C]测试SSH免密码登录（在每台主机上都要测试能否连接到3台主机，需要执行9次验证）

```
$ssh hadoop@lixucheng01
$ssh hadoop@lixucheng02
$ssh hadoop@lixucheng03
```

测试成功后，执行exit命令结束远程登录：

```
$ssh exit
```

例如：

```
hadoop@lixucheng01:~/ssh$ ssh hadoop@lixucheng02
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 个可升级软件包。
0 个安全更新。

Your Hardware Enablement Stack (HWE) is supported until April 2023.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

hadoop@lixucheng02:~$ exit
注销
Connection to lixucheng02 closed.
hadoop@lixucheng01:~/ssh$ ssh hadoop@lixucheng03
```

## 4 安装Java环境

### 4.1 [A]进入安装位置所在的文件夹

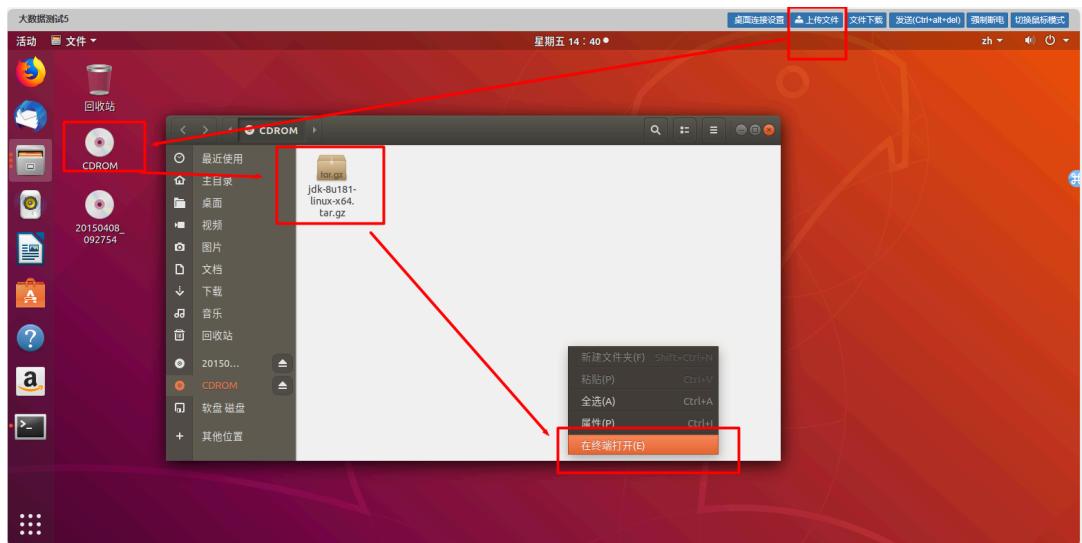
- 【实验室云桌面环境下】安装文件通过云桌面的上传文件上传，或者使用mobaxteam软件通过ssh上传,或者访问老师的ftp下载

使用老师的ftp下载方法如下

虚拟机的火狐浏览器输入：<ftp://172.22.252.7/software>，用户名ub，密码123

名称	大小	修改时间
文件: apache-maven-3.6.3-bin.zip	9378 KB	2024/3/3 GMT+8 下午2:28:00
文件: eclipse-4.7.0-linux.gtk.x86_64.tar.gz	274675 KB	2024/3/3 GMT+8 下午2:28:00
文件: hadoop-3.1.3.tar.gz	330153 KB	2024/3/1 GMT+8 下午2:37:00
文件: introduction.zip	4256 KB	2024/3/1 GMT+8 下午3:23:00
文件: jdk-8u181-linux-x64.tar.gz	181296 KB	2024/3/1 GMT+8 下午2:36:00

如果通过云桌面上传：



- 【VMware虚拟机环境下】安装文件可以直接从windows复制，粘贴到虚拟机中，，或者使用mobaxteam软件ssh上传

进入到安装文件所在的目录

```
$ cd /path/to/文件所在目录
```

注意，以下在hadoop用户下执行

#### 4.2 [A]解压缩

使用tar命令解压安装jdk-8u181-linux-x64.tar.gz文件到目录/usr/local。 (使用root用户)

```
$sudo tar -zxvf jdk-8u181-linux-x64.tar.gz -C /usr/local
```

#### 4.3 [A]为用户赋权限

将加压缩后的Java目录所有者修改为hadoop用户，此步骤需要在root用户下

```
$sudo chown -R hadoop:hadoop /usr/local/jdk1.8.0_181
```

#### 4.3 [A]配置JDK环境变量，使其生效。

切换hadoop用户使用vi命令打开用户的配置文件 ~/.bashrc。如果出现无法连接，就在前面加sudo

```
$gedit ~/.bashrc
```

在文件中加入下列内容：

```
export JAVA_HOME=/usr/local/jdk1.8.0_181
export JRE_HOME=$JAVA_HOME/jre
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
export CLASSPATH=$CLASSPATH:.:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

使环境变量生效，并验证JDK是否安装成功。

```
$ source ~/.bashrc
$ java -version
```

#### 4.4 将Java复制到另外两台机器

(1) [B,C]在另外两台机器上 (B、C) 为hadoop用户设置/usr/local的权限

```
$ sudo chown -R hadoop:hadoop /usr/local
```

(2) [A]在A上复制Java文件

```
sudo scp -r /usr/local/jdk1.8.0_181/ hadoop@lixucheng02:/usr/local/
```



```
sudo scp -r /usr/local/jdk1.8.0_181/ hadoop@lixucheng03:/usr/local/
```



(3) [B,C]在B、C上配置环境变量，使环境生效，验证是否成功

将下面的内容复制到~/.bashrc中

```
export JAVA_HOME=/usr/local/jdk1.8.0_181
export JRE_HOME=$JAVA_HOME/jre
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
export CLASSPATH=$CLASSPATH:.:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

```
hadoop@lixucheng02:/usr/local$ sudo gedit ~/.bashrc
** (gedit:4557): WARNING **: 09:07:03.041: Set document metadata failed: 不支持
设置属性 metadata::gedit-spell-language

** (gedit:4557): WARNING **: 09:07:03.041: Set document metadata failed: 不支持
设置属性 metadata::gedit-encoding

** (gedit:4557): WARNING **: 09:07:04.498: Set document metadata failed: 不支持
设置属性 metadata::gedit-position
hadoop@lixucheng02:/usr/local$ source ~/.bashrc
hadoop@lixucheng02:/usr/local$ java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

## 5 分布式集群安装配置

5.1 [A]按照4.1中进入终端的方式进入桌面对应的终端

5.2 [A]把hadoop解压缩到/usr/local下面

使用tar命令解压安装到目录/usr/local，并重命名为hadoop。

```
$ cd /hadoop安装包所在路径
```

注意上面这个步骤，是指进入安装文件所在的目录，以你自己实际目录为准，不要照抄

```
$ ls
$ sudo tar -zvxf hadoop-3.1.3.tar.gz -C /usr/local
$ cd /usr/local
```

```
$ ls  
$ sudo mv ./hadoop-3.1.3 ./hadoop
```

5. 3 [A]在节点1上，将目录/usr/local/hadoop的所有者修改为hadoop用户。

```
$ sudo chown -R hadoop:hadoop /usr/local/hadoop
```

5. 4 [A,B,C]修改环境变量，并使其生效：在集群的3个节点中都需要操作。

切换hadoop用户修改环境变量

```
$su hadoop  
$gedit ~/.bashrc
```

在配置文件.bashrc中写入下列hadoop配置信息。

```
export HADOOP_HOME=/usr/local/hadoop  
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop  
export YARN_HOME=/usr/local/hadoop  
export YARN_CONF_DIR=${YARN_HOME}/etc/hadoop  
export PATH=$HADOOP_HOME/sbin/:$HADOOP_HOME/bin/:$PATH
```

生效环境变量

```
$source ~/.bashrc
```

5. 5 [A]在节点配置Hadoop文件

进入Hadoop配置目录，使用vi命令对7个文件分别进行配置。

```
$cd /usr/local/hadoop/etc/hadoop
```

(1) [A]配置hadoop-env.sh

主要配置JAVA\_HOME

```
$gedit hadoop-env.sh
```

写入下列内容：

```
export JAVA_HOME=/usr/local/jdk1.8.0_181
```

(2) [A]配置yarn-env.sh

主要配置JAVA\_HOME

```
$gedit yarn-env.sh
```

写入下列内容：

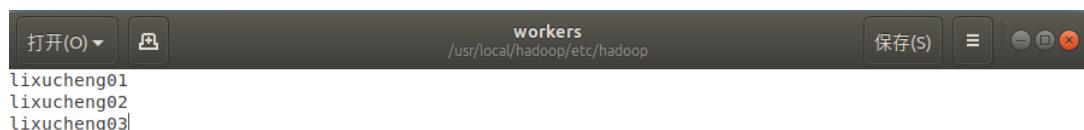
```
export JAVA_HOME=/usr/local/jdk1.8.0_181
```

(3) [A]配置workers

将所有节点的主机名写入该文件中。

```
$gedit workers
```

例如：



```
workers
/usr/local/hadoop/etc/hadoop
保存(S) 三
lixucheng01
lixucheng02
lixucheng03|
```

(4) [A]配置core-site.xml

三个节点中，选取第一节点为NameNode节点（名称节点），3个节点都作为DataNode节点（数据节点）。

写入下列内容（注意修改为自己的主机名，把第一个节点作为namenode了）：

```
<configuration>
<property>
```

```
<name>fs.defaultFS</name>
<value>hdfs://lixucheng01:8020</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>file:/usr/local/hadoop/tmp</value>
</property>

</configuration>
```

## (5) [A]配置hdfs-site.xml

修改hdfs-site.xml文件。 (使用NameNode节点名替换)

```
$gedit hdfs-site.xml
```

写入下列内容 (注意configuration) : 把第二个节点作为secondary namenode

```
<configuration>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>lixucheng02:9868</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///usr/local/hadoop/dfs/name</value>
</property>

<property>
<name>dfs.datanode.data.dir</name>
<value>file:///usr/local/hadoop/dfs/data</value>
</property>

<property>
<name>dfs.replication</name>
<value>3</value>
</property>

</configuration>
```

## (6) [A]配置mapred-site.xml

进入编辑

```
$gedit mapred-site.xml
```

写入下列内容：

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

(7) [A]配置yarn-site.xml

修改yarn-site.xml文件。

```
$gedit yarn-site.xml
```

写入下列内容：

```
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

<property>
  <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

<property>
  <name>yarn.resourcemanager.address</name>
  <value>lixucheng01:8032</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,
HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,
HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME</value>
```

```
</property>
</configuration>
```

5.6 将NameNode节点上配置好的hadoop文件夹分发给所有的DataNode 节点。使用scp远程发送文件夹的基本格式为：

(1) [B,C]在另外两个节点上的/usr/local下面创建hadoop文件夹

```
$sudo mkdir hadoop
```

(2) [B,C]为hadoop用户赋访问hadoop文件夹的权限

```
sudo chown -R hadoop:hadoop hadoop
```

(3) [A]将hadoop文件夹发送到另外两个节点上

例如：

```
# 复制到其他节点    hadoop用户
sudo scp -r /usr/local/hadoop hadoop@lixucheng02:/usr/local/
sudo scp -r /usr/local/hadoop hadoop@lixucheng03:/usr/local/
```

5.7 [A]切换到hadoop用户格式化NameNode节点。

只在NameNode节点上执行以下操作：

```
$cd /usr/local/hadoop
$bin/hdfs namenode -format
```

5.8 [A]在主节点启动Hadoop服务

```
$cd /usr/local/hadoop
$bin/start-dfs.sh
$bin/start-yarn.sh
```

5.9 [A,B,C]验证是否安装成功。

执行jps命令查看服务

节点1：

```
hadoop@lixucheng01:/usr/local/hadoop$ jps
21297 NameNode
24130 ResourceManager
21507 DataNode
24317 NodeManager
26223 Jps
```

节点2：

```
hadoop@lixucheng02:/usr/local/hadoop$ jps
8560 SecondaryNameNode
8390 DataNode
10073 NodeManager
11388 Jps
```

节点3：

```
hadoop@lixucheng03:/usr/local/hadoop$ jps
30720 DataNode
32258 NodeManager
1373 Jps
```

[A]停止服务方法

```
$ stop-dfs.sh
$ stop-yarn.sh
```

停止服务后，你可以做其他配置修改，排错等工作

[A]排错完成后，可以再次启动，但启动前需要删除缓存文件

```
$ cd /usr/local/hadoop
$ ls
$ rm -r dfs/ logs/ tmp/
```

然后重新格式化namenode

```
$ hdfs namenode -format  
$ start-dfs.sh  
$ start-yarn.sh
```

## 5.10 [A]运行MapReduce程序

```
$cd /usr/local/hadoop  
$bin/hadoop jar share/hadoop/mapreduce/  
hadoop-mapreduce-examples-3.1.3.jar pi 2 1000
```

上面第二上和第三行之间不折行。

如下图所示：

## 5.11 [A]访问管理页面

yarn管理页面

<http://mashuai01:8088/cluster>

HDFS管理页面

<http://mashuai01:9870/>

Namenode information - Mozilla Firefox

Namenode information | <ftp://172.22.252.7/softw> | +

① [mashuai01:9870/dfshealth.html#tab-datanode](http://mashuai01:9870/dfshealth.html#tab-datanode)

In operation

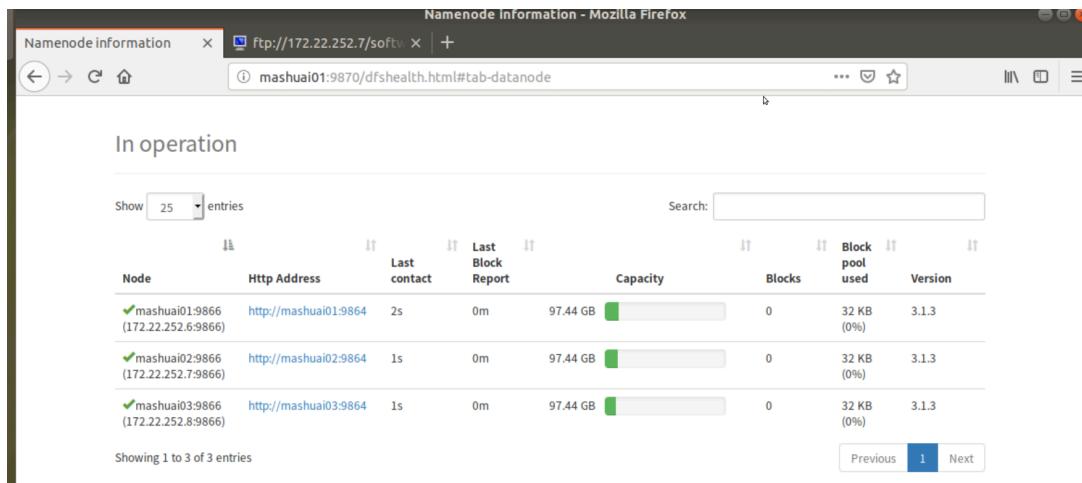
Show 25 entries

Search:

Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
✓ mashuai01:9866 (172.22.252.6:9866)	<a href="http://mashuai01:9864">http://mashuai01:9864</a>	2s	0m	97.44 GB	0	32 KB (0%)	3.1.3
✓ mashuai02:9866 (172.22.252.7:9866)	<a href="http://mashuai02:9864">http://mashuai02:9864</a>	1s	0m	97.44 GB	0	32 KB (0%)	3.1.3
✓ mashuai03:9866 (172.22.252.8:9866)	<a href="http://mashuai03:9864">http://mashuai03:9864</a>	1s	0m	97.44 GB	0	32 KB (0%)	3.1.3

Showing 1 to 3 of 3 entries

Previous 1 Next



Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
✓ mashuai01:9866 (172.22.252.6:9866)	<a href="http://mashuai01:9864">http://mashuai01:9864</a>	2s	0m	97.44 GB	0	32 KB (0%)	3.1.3
✓ mashuai02:9866 (172.22.252.7:9866)	<a href="http://mashuai02:9864">http://mashuai02:9864</a>	1s	0m	97.44 GB	0	32 KB (0%)	3.1.3
✓ mashuai03:9866 (172.22.252.8:9866)	<a href="http://mashuai03:9864">http://mashuai03:9864</a>	1s	0m	97.44 GB	0	32 KB (0%)	3.1.3