作业一 环境配置

一、实验目的

1. 掌握虚拟机环境下大数据相关组件的安装与配置
2. 实现 SSH 免密登录配置
3. 完成 JDK、Scala 开发环境的搭建
4. 部署 Hadoop 分布式文件系统及相关组件
5. 配置 Zookeeper、HBase 和 Spark 并进行功能验证

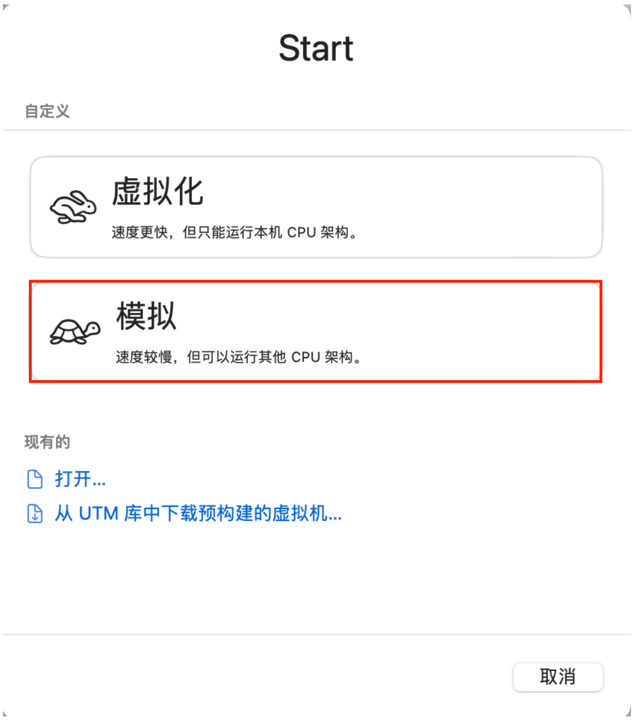
二、实验环境

- 操作系统：Ubuntu 22.04 LTS
- 虚拟机：UTM
- 集群节点：1 个 master 节点，2 个 slave 节点（slave1, slave2）
- 网络配置：静态 IP 地址

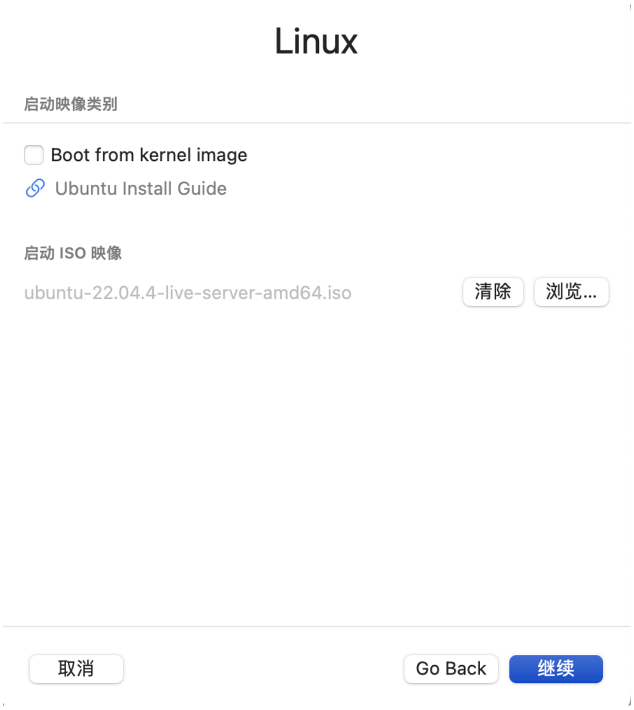
三、实验内容与步骤

3.1 虚拟机搭建

1. **创建虚拟机**：使用 UTM 创建 3 台 Ubuntu 虚拟机，分别命名为 master、slave1 和 slave2。
    1. 在UTM中创建新的虚拟机，由于我的系统是ARM架构，要运行x86_64架构的Ubuntu需要选择模拟选项


Start
自定义
虚拟化
速度更快，但只能运行本机 CPU 架构。
模拟
速度较慢，但可以运行其他 CPU 架构。
现有的
打开...
从 UTM 库中下载预构建的虚拟机...
取消

    2. 选择下载好的ISO映像



Linux

启动映像类别

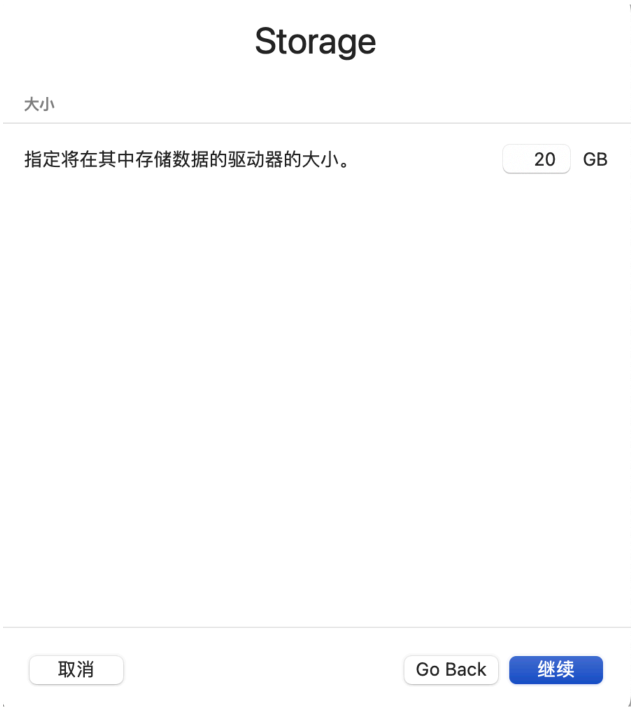
□ Boot from kernel image

Ubuntu Install Guide

启动 ISO 映像

ubuntu-22.04.4-live-server-amd64.iso

3. 使用默认选项配置，并设置20G的磁盘容量

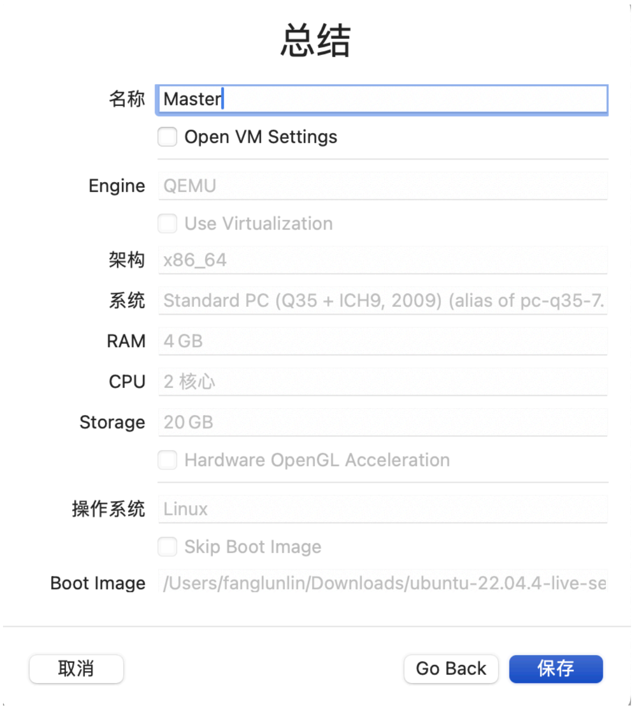


Storage

大小

指定将在其中存储数据的驱动器的大小。 20 GB

4. 最终虚拟机设置如下



总结

名称 Master

□ Open VM Settings

Engine QEMU

□ Use Virtualization

架构 x86_64

系统 Standard PC (Q35 + ICH9, 2009) (alias of pc-q35-7.

RAM 4 GB

CPU 2 核心

Storage 20 GB

□ Hardware OpenGL Acceleration

操作系统 Linux

□ Skip Boot Image

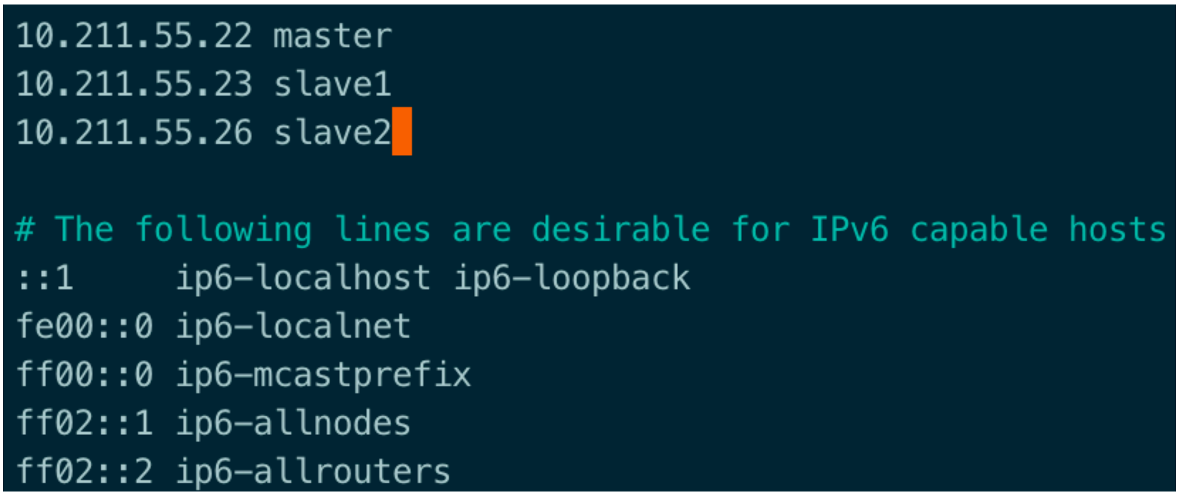
Boot Image /Users/fanglunlin/Downloads/ubuntu-22.04.4-live-se

2. **配置虚拟机**：
   - 统一设置主机名为lfl
   - 设置用户名为 master、slave1、slave2

```
$ vi /etc/hostname
```

   - 关于文件存放
     - 软件目录：~/package
     - 安装目录：~/install

3. **网络配置**：
   - 设置静态 IP 地址：
     - master：10.211.55.22
     - slave1：10.211.55.32
     - slave2：10.211.55.36
   - 修改`/etc/hosts`文件，添加节点 IP 与主机名映射（所有节点均设置）

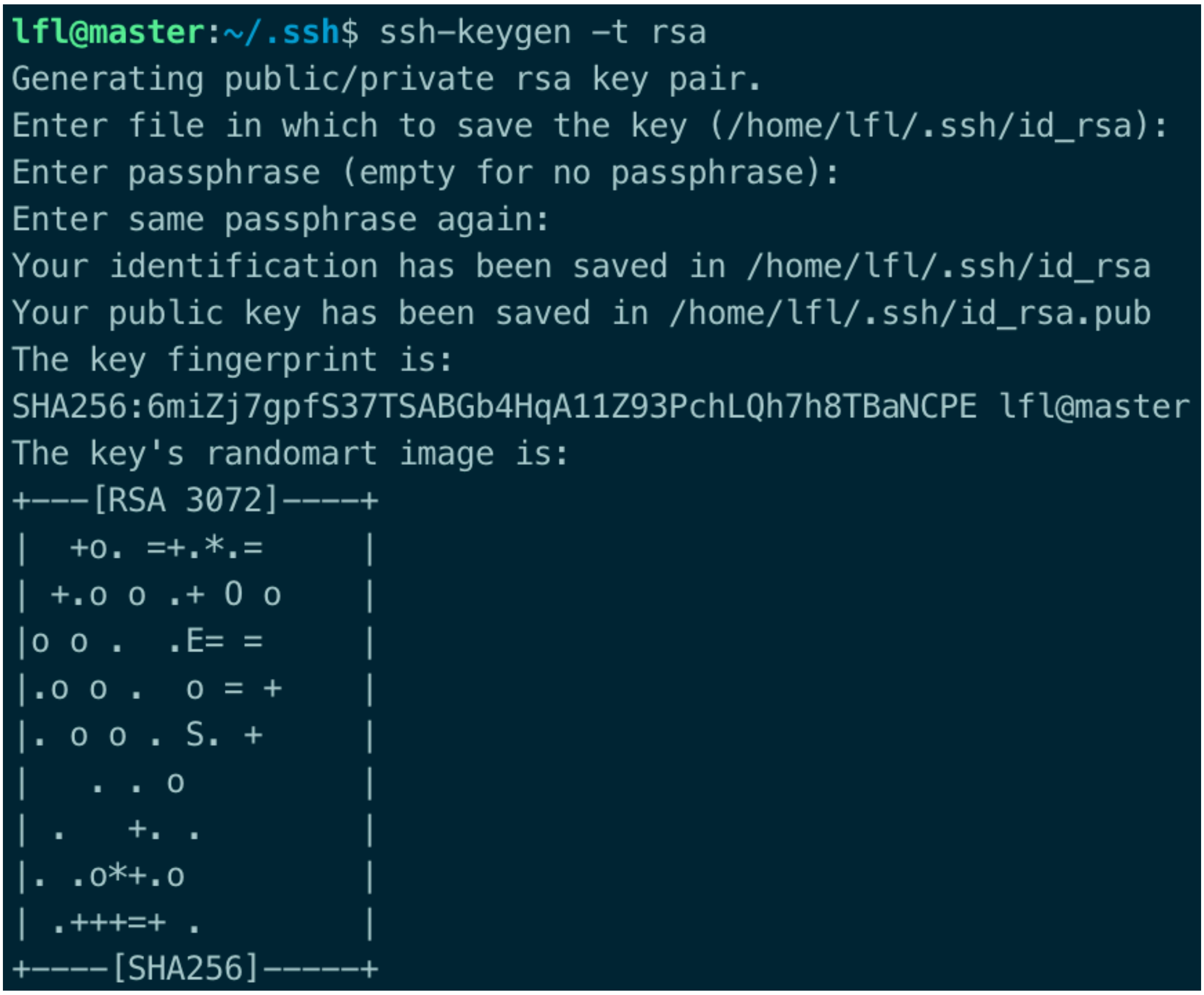


```
10.211.55.22 master
10.211.55.23 slave1
10.211.55.26 slave2

# 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
```

3.2 SSH 及免密登录配置

1. 创建 SSH 密钥：在 master 节点执行以下命令

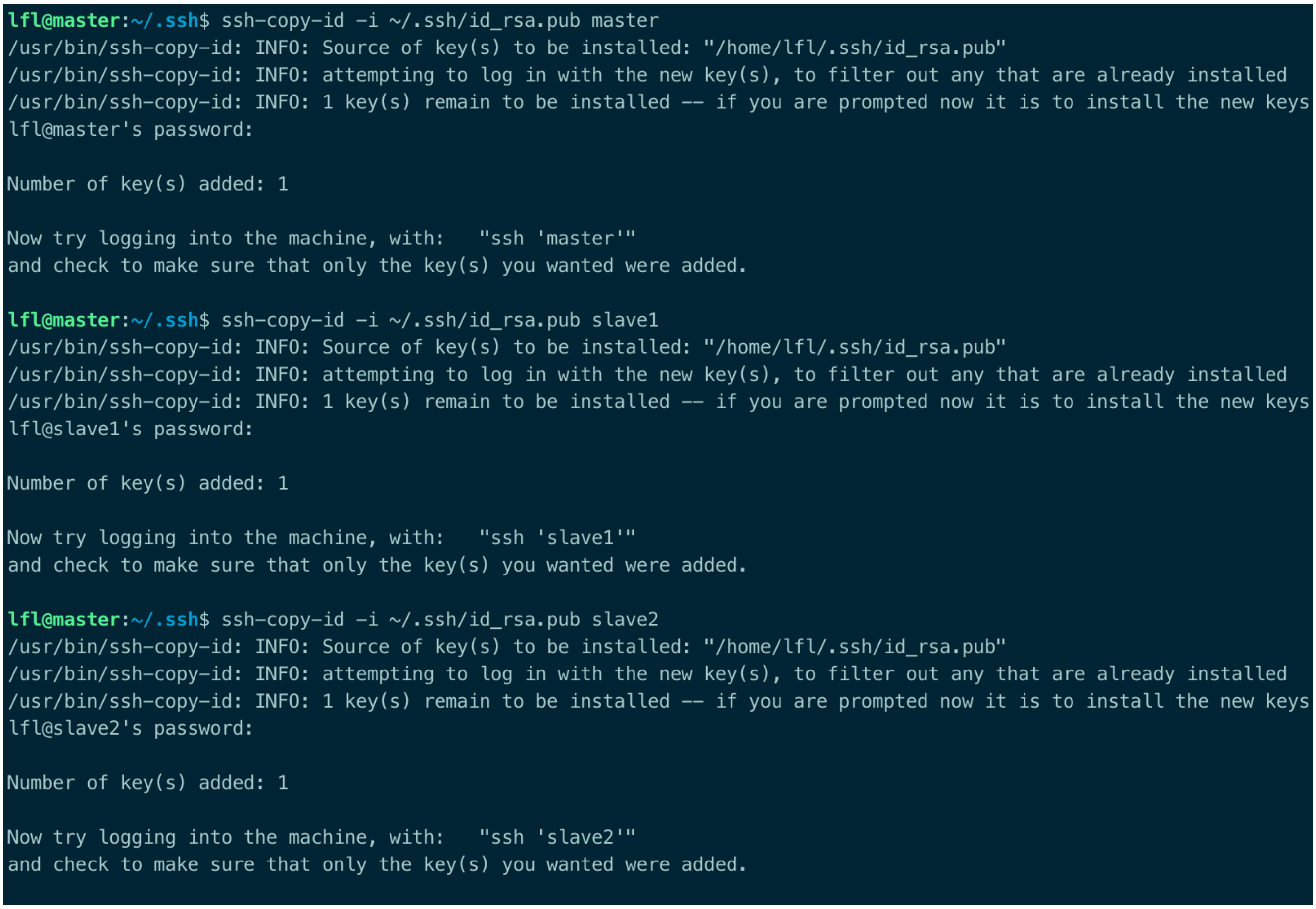
```
cd ~
mkdir .ssh
cd .ssh
ssh-keygen -t rsa
```



```
lfl@master:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lfl/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/lfl/.ssh/id_rsa
Your public key has been saved in /home/lfl/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:6miZj7gpfS37TSABGb4HqA11Z93PchLQh7h8TBaNCPE lfl@master
The key's randomart image is:
+---[RSA 3072]----+
|  +o. =+.*.=     |
| +.o o .+ O o    |
|o o .  .E= =     |
|.o o .  o = +    |
|. o o . S. +     |
|    . . o        |
| .   +. .        |
|. .o*+.o         |
| .+++=+ .        |
+----[SHA256]-----+
```

2. 配置免密登录：将 master 的公钥复制到 slave 节点

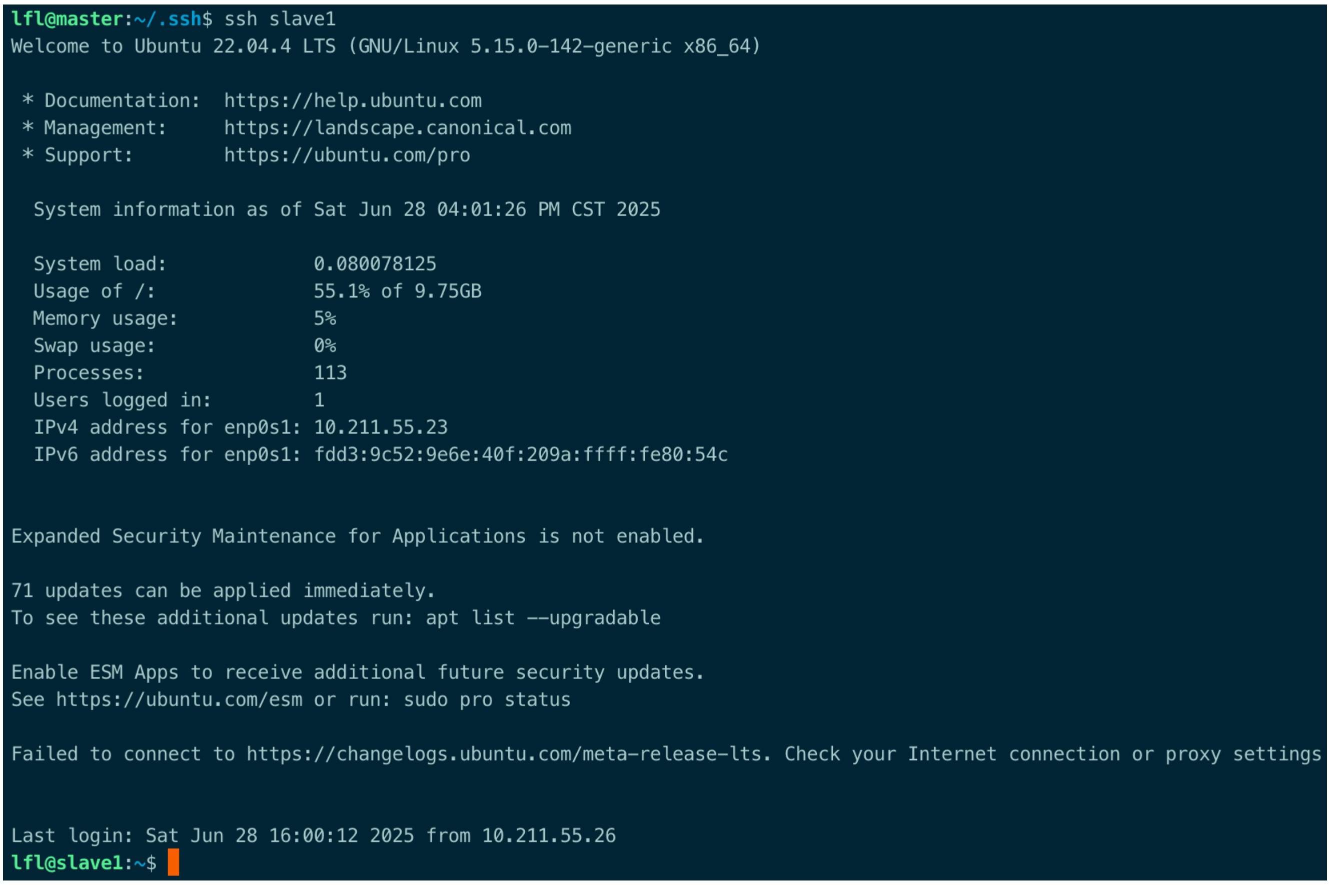
```
ssh-copy-id -i ~/.ssh/id_rsa.pub master
ssh-copy-id -i ~/.ssh/id_rsa.pub slave1
ssh-copy-id -i ~/.ssh/id_rsa.pub slave2
```



```
lfl@master:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub master
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/lfl/.ssh/id_rsa.pub"
/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 prompted now it is to install the new keys
lfl@master's password:

Number of key(s) added: 1

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

lfl@master:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub slave1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/lfl/.ssh/id_rsa.pub"
/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 prompted now it is to install the new keys
lfl@slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave1'"
and check to make sure that only the key(s) you wanted were added.

lfl@master:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub slave2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/lfl/.ssh/id_rsa.pub"
/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 prompted now it is to install the new keys
lfl@slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave2'"
and check to make sure that only the key(s) you wanted were added.
```

3. 验证免密登录：在 master 节点尝试登录 slave1

```
ssh slave1
```



```
lfl@master:~/.ssh$ ssh slave1
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-142-generic x86_64)

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

  System information as of Sat Jun 28 04:01:26 PM CST 2025

  System load:             0.080078125
  Usage of /:              55.1% of 9.75GB
  Memory usage:            5%
  Swap usage:              0%
  Processes:               113
  Users logged in:         1
  IPv4 address for enp0s1: 10.211.55.23
  IPv6 address for enp0s1: fdd3:9c52:9e6e:40f:209a:ffff:fe80:54c


Expanded Security Maintenance for Applications is not enabled.

71 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sat Jun 28 16:00:12 2025 from 10.211.55.26
lfl@slave1:~$
```

4. slave1、slave2节点重复同样操作，所有节点两两之间进行ssh免密钥配置3.3 JDK 环境搭建

3.3 JDK 环境搭建

1. 解压 JDK：将下载好的 JDK 安装包解压到指定目录

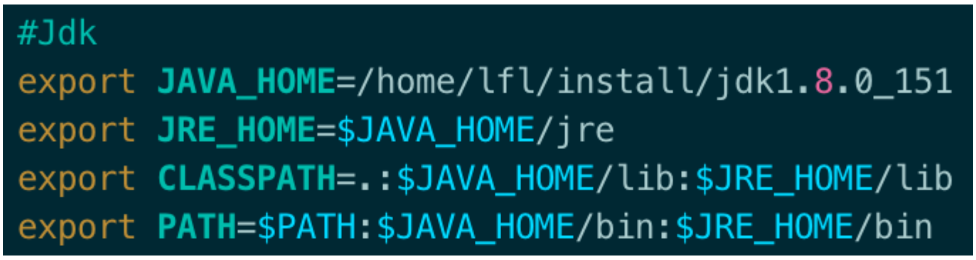
```
sudo tar -zxvf jdk-8u151-linux-x64.tar.gz -C ~/install
```

2. 配置环境变量：

   修改文件

```
/etc/profile
```

```
export JAVA_HOME=/home/hadoop/install/jdk1.8.0_151
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```



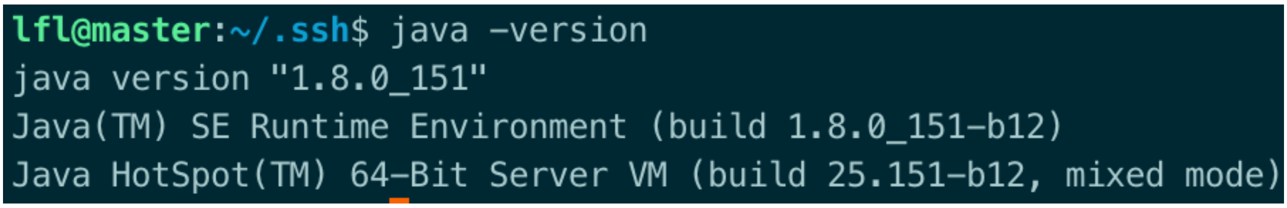
```
#Jdk
export JAVA_HOME=/home/lfl/install/jdk1.8.0_151
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

3. 使配置生效：

```
source /etc/profile
```

4. 验证安装：

```
java -version
```



```
lfl@master:~/.ssh$ java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

5. 将文件夹scp到其它节点服务器上（下示为slave1，slave2同理）

```
$ scp -r ~/install/jdk1.8.0_151 $(用户名)@slave1:~/install/jdk1.8.0_151
```

6. 将环境变量scp到其它节点服务器上（下示为slave1，slave2同理）

```
$ sudo scp /etc/profile $(用户名)@slave1:/etc
```

3.4 Scala 环境搭建

1. 解压 Scala：将下载好的 Scala 安装包解压到指定目录

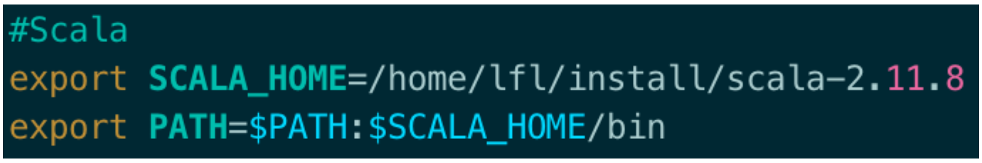
```
sudo tar -zxvf scala-2.11.8.tgz -C ~/install
```

2. 配置环境变量：

   修改文件

```
/etc/profile
```

```
export SCALA_HOME=/home/hadoop/install/scala-2.11.8
export PATH=$PATH:$SCALA_HOME/bin
```



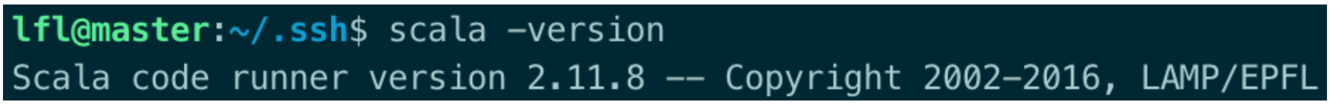
```
#Scala
export SCALA_HOME=/home/lfl/install/scala-2.11.8
export PATH=$PATH:$SCALA_HOME/bin
```

3. 使配置生效：

```
source /etc/profile
```

4. 验证安装：

```
scala -version
```



```
lfl@master:~/.ssh$ scala -version
Scala code runner version 2.11.8 -- Copyright 2002-2016, LAMP/EPFL
```

5. 将文件夹scp到其它节点服务器上（下示为slave1，slave2同理）

```
$ scp -r ~/install/jdk1.8.0_151 $(用户名)@slave1:~/install/scala-2.11.8
```

6. 将环境变量scp到其它节点服务器上（下示为slave1，slave2同理）

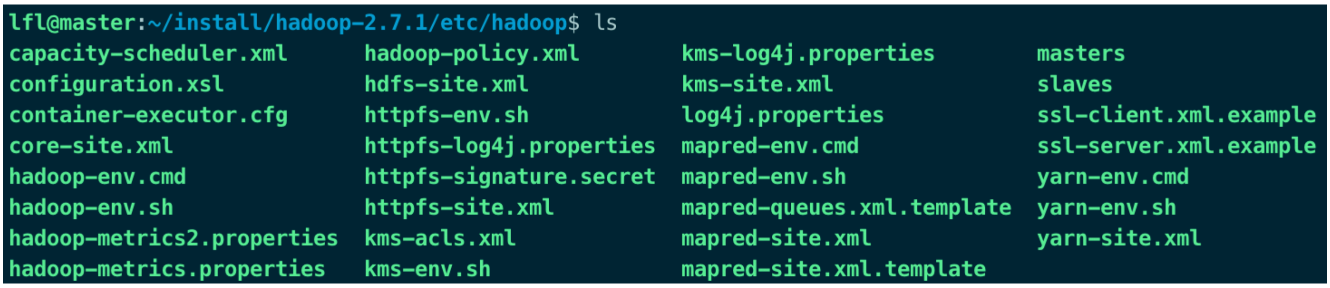
```
$ sudo scp /etc/profile $(用户名)@slave1:/etc
```

3.5 Hadoop 环境搭建

1. 解压 Hadoop：将下载好的 Hadoop 安装包解压到指定目录

```
tar -zxvf hadoop-2.7.1.tar.gz -C ~/install
```

2. 修改配置文件（**hadoop安装目录的/etc/hadoop目录下**）：



```
lfl@master:~/install/hadoop-2.7.1/etc/hadoop$ ls
capacity-scheduler.xml      hadoop-policy.xml        kms-log4j.properties        masters
configuration.xsl           hdfs-site.xml            kms-site.xml                slaves
container-executor.cfg      httpfs-env.sh            log4j.properties            ssl-client.xml.example
core-site.xml               httpfs-log4j.properties  mapred-env.cmd              ssl-server.xml.example
hadoop-env.cmd              httpfs-signature.secret  mapred-env.sh               yarn-env.cmd
hadoop-env.sh               httpfs-site.xml          mapred-queues.xml.template  yarn-env.sh
hadoop-metrics2.properties  kms-acls.xml             mapred-site.xml             yarn-site.xml
hadoop-metrics.properties   kms-env.sh               mapred-site.xml.template
```

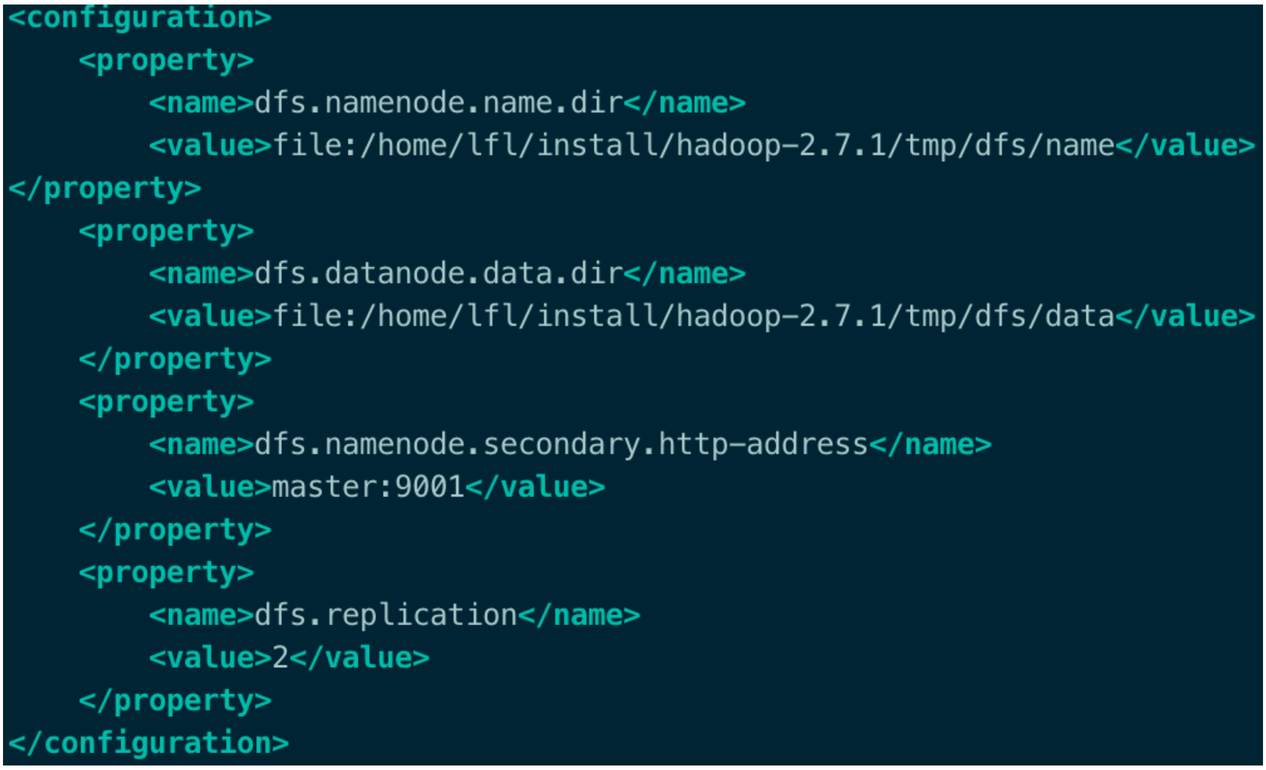
1. `core-site.xml`：设置 HDFS 默认地址和临时目录

```
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://master:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/home/lfl/hadoop/tmp</value>
    </property>
</configuration>
```

```
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://master:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/home/lfl/hadoop/tmp</value>
    </property>
</configuration>
```

2. `hdfs-site.xml`：设置 NameNode 和 DataNode 数据存储目录

```
<configuration>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/home/lfl/install/hadoop-2.7.1/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/home/lfl/install/hadoop-2.7.1/tmp/dfs/data</value>
    </property>
    <property>
        <name>dfs.namenode.secondary.http-address</name>
        <value>master:9001</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
    </property>
</configuration>
```



```
<configuration>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/home/lfl/install/hadoop-2.7.1/tmp/dfs/name</value>
</property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/home/lfl/install/hadoop-2.7.1/tmp/dfs/data</value>
    </property>
    <property>
        <name>dfs.namenode.secondary.http-address</name>
        <value>master:9001</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
    </property>
</configuration>
```

3. `mapred-site.xml`：配置 MapReduce 框架为 YARN

   （实验中使用的是旧版Hadoop，需要`cp mapred-site.xml.template mapred-site.xml`将文件复制后修改）

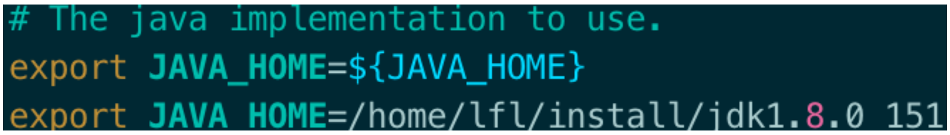
```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.address</name>
        <value>master:10020</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>master:19888</value>
     </property>
</configuration>
```

4. `yarn-site.xml`：设置 ResourceManager 主机名

```
<configuration>
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
    </property>
    <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-aggregation.retain-seconds</name>
        <value>604800</value>
    </property>
</configuration>
```

5. `hadoop-env.sh`：指定 JAVA_HOME 路径

```
export JAVA_HOME=/home/lfl/install/jdk1.8.0_151
```



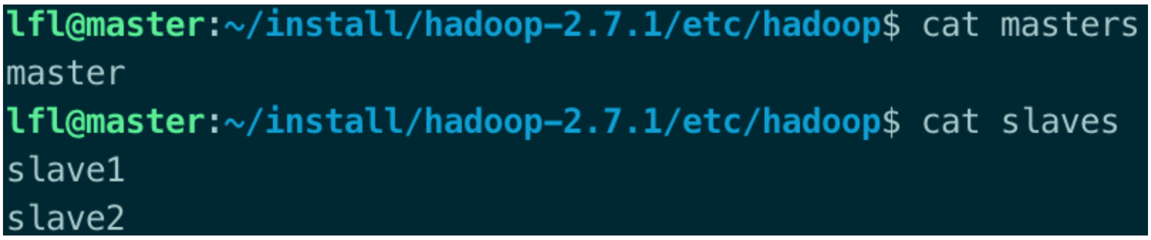
```
# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/home/lfl/install/jdk1.8.0_151
```

6. `masters`：添加 master 节点列表

```
master
```

7. `slaves`：添加 slave 节点列表

```
slave1
slave2
```



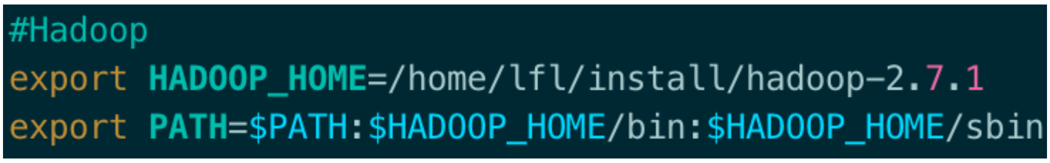
```
lfl@master:~/install/hadoop-2.7.1/etc/hadoop$ cat masters
master
lfl@master:~/install/hadoop-2.7.1/etc/hadoop$ cat slaves
slave1
slave2
```

3. 配置环境变量：

   修改文件

```
/etc/profile
```

```
export HADOOP_HOME=/home/hadoop/install/hadoop-2.7.1
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```



```
#Hadoop
export HADOOP_HOME=/home/lfl/install/hadoop-2.7.1
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

4. 将文件夹scp到其它节点服务器上（下示为slave1，slave2同理）

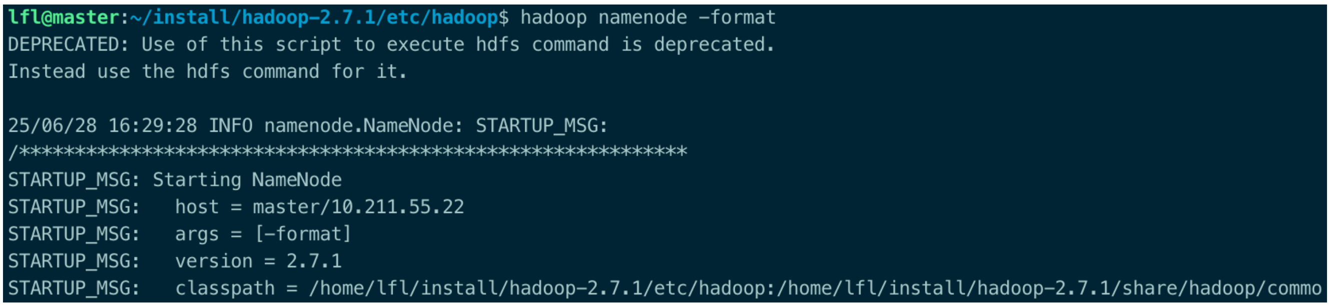
```
$ scp -r ~/install/hadoop-2.7.1 lfl@slave1:~/install/hadoop-2.7.1
```

5. 将环境变量scp到其它节点服务器上（下示为slave1，slave2同理）

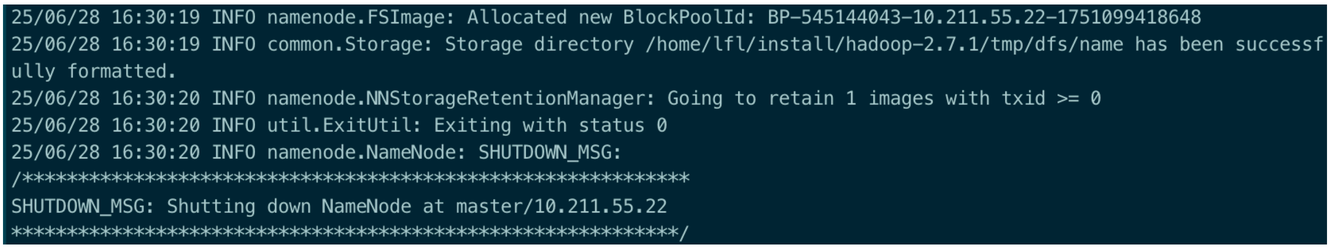
```
$ sudo scp /etc/profile $(用户名)@slave1:/etc
```

6. 格式化 NameNode：

```
hadoop namenode -format
```



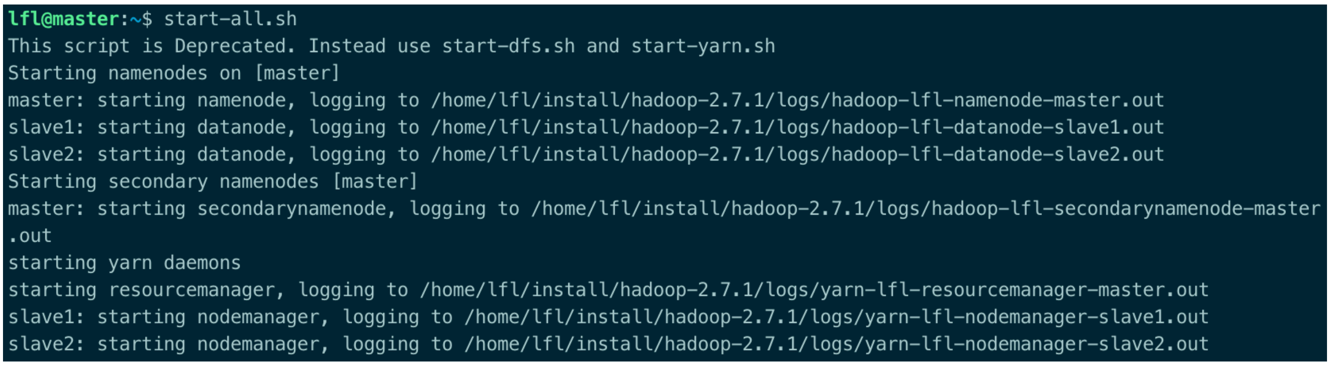
```
lfl@master:~/install/hadoop-2.7.1/etc/hadoop$ hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

25/06/28 16:29:28 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/10.211.55.22
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.1
STARTUP_MSG:   classpath = /home/lfl/install/hadoop-2.7.1/etc/hadoop:/home/lfl/install/hadoop-2.7.1/share/hadoop/commo
```



```
25/06/28 16:30:19 INFO namenode.FSImage: Allocated new BlockPoolId: BP-545144043-10.211.55.22-1751099418648
25/06/28 16:30:19 INFO common.Storage: Storage directory /home/lfl/install/hadoop-2.7.1/tmp/dfs/name has been successf
ully formatted.
25/06/28 16:30:20 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
25/06/28 16:30:20 INFO util.ExitUtil: Exiting with status 0
25/06/28 16:30:20 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master/10.211.55.22
************************************************************/
```

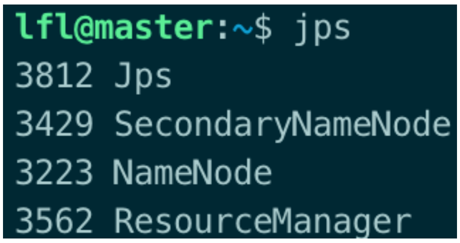
7. 启动 Hadoop 集群：

```
start-all.sh
```



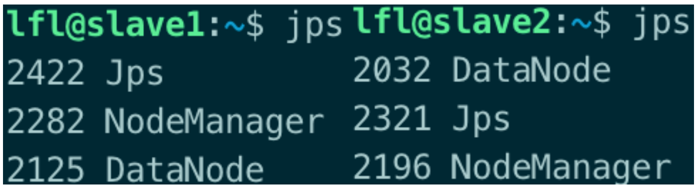
```
lfl@master:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /home/lfl/install/hadoop-2.7.1/logs/hadoop-lfl-namenode-master.out
slave1: starting datanode, logging to /home/lfl/install/hadoop-2.7.1/logs/hadoop-lfl-datanode-slave1.out
slave2: starting datanode, logging to /home/lfl/install/hadoop-2.7.1/logs/hadoop-lfl-datanode-slave2.out
Starting secondary namenodes [master]
master: starting secondarynamenode, logging to /home/lfl/install/hadoop-2.7.1/logs/hadoop-lfl-secondarynamenode-master
.out
starting yarn daemons
starting resourcemanager, logging to /home/lfl/install/hadoop-2.7.1/logs/yarn-lfl-resourcemanager-master.out
slave1: starting nodemanager, logging to /home/lfl/install/hadoop-2.7.1/logs/yarn-lfl-nodemanager-slave1.out
slave2: starting nodemanager, logging to /home/lfl/install/hadoop-2.7.1/logs/yarn-lfl-nodemanager-slave2.out
```

8. 验证集群：
    - 使用`jps`查看进程：
        - master:



```
lfl@master:~$ jps
3812 Jps
3429 SecondaryNameNode
3223 NameNode
3562 ResourceManager
```

    - slave:



```
lfl@slave1:~$ jps
2422 Jps
2282 NodeManager
2125 DataNode
```

```
lfl@slave2:~$ jps
2032 DataNode
2321 Jps
2196 NodeManager
```

    - 可看到master节点：NameNode、SecondNameNode、ResourceManager
    - slave1~4节点：DataNode、NodeManager
  - 访问 Web 界面：http://10.211.55.22:50070

3.6 Zookeeper 环境搭建

1. 解压 Zookeeper：将下载好的 Zookeeper 安装包解压到指定目录

```
tar -zxvf zookeeper-3.4.10.tar.gz -C ~/install
```

2. 创建数据和日志目录：

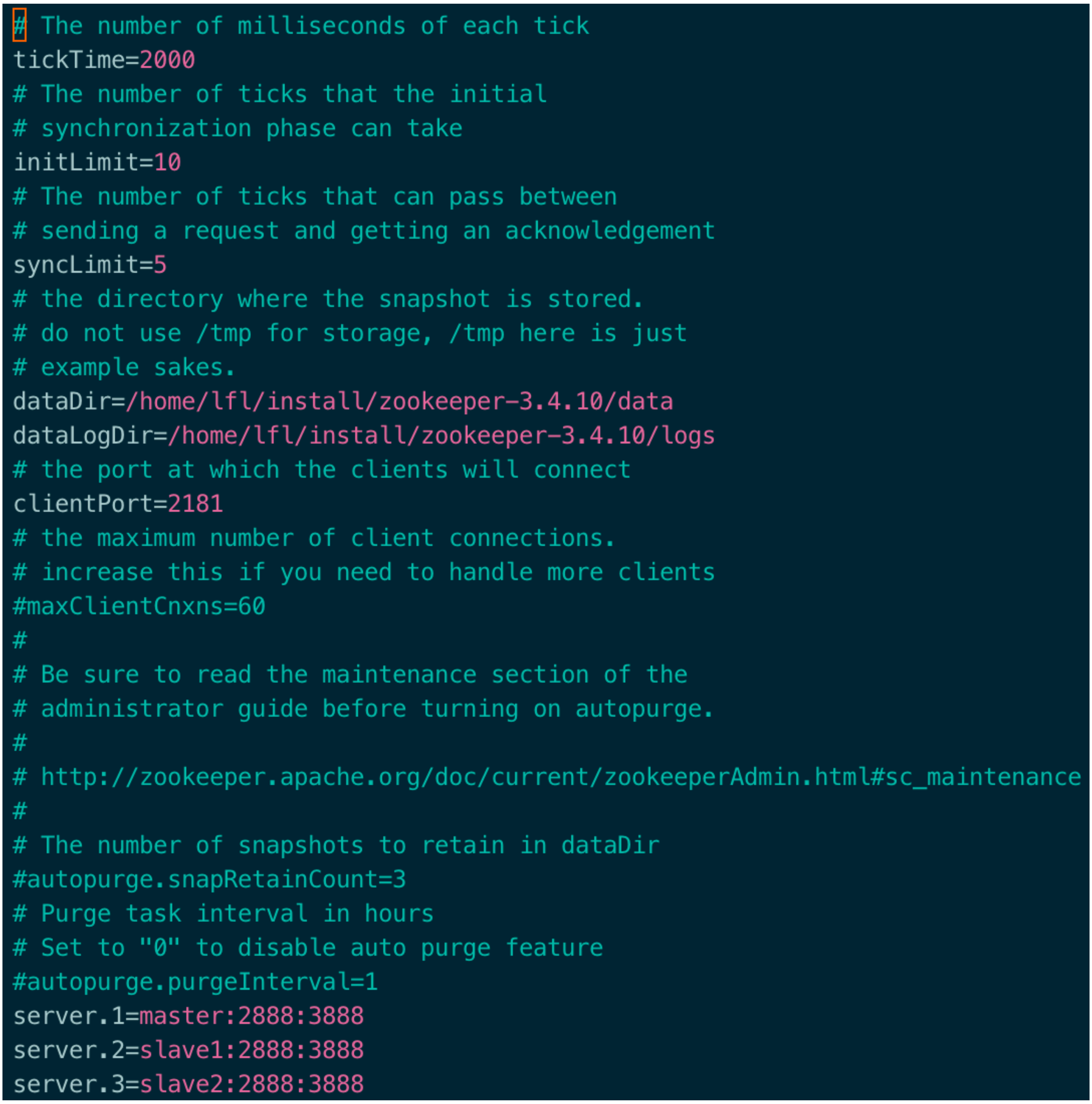
```
mkdir data
mkdir logs
```

3. 创建配置文件

```
$ cd /home/lfl/install/zookeeper-3.4.10/conf
$ cp zoo_sample.cfg zoo.cfg
```

4. 配置 zoo.cfg：设置 tickTime、dataDir、clientPort 等参数，并添加集群节点配置

```
tickTime=2000
initLimit=10
syncLimit=5
dataDir=/home/lfl/install/zookeeper-3.4.10/data
dataLogDir=/home/lfl/install/zookeeper-3.4.10/logs
clientPort=2181
server.1=master:2888:3888
server.2=slave1:2888:3888
server.3=slave2:2888:3888
server.4=slave3:2888:3888
server.5=slave4:2888:3888
```



```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/home/lfl/install/zookeeper-3.4.10/data
dataLogDir=/home/lfl/install/zookeeper-3.4.10/logs
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
server.1=master:2888:3888
server.2=slave1:2888:3888
server.3=slave2:2888:3888
```

5. 配置 myid：在 data 目录下创建 myid 文件，master 节点为 1，slave1 为 2，slave2 为 3

```
$ cd /home/lfl/install/zookeeper-3.4.10/data
$ echo '1' > myid
```

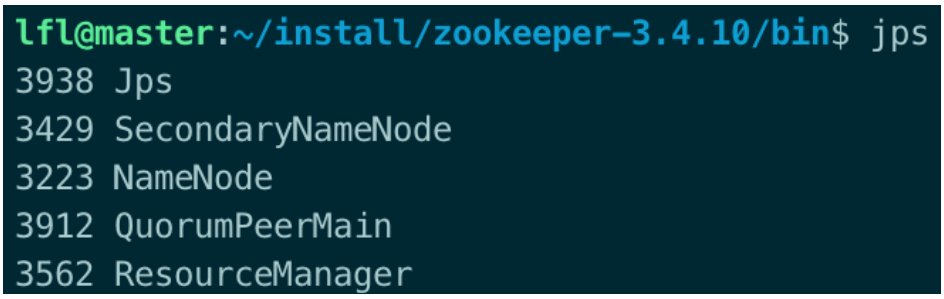
6. 启动 Zookeeper（**所有节点都需要启动**）：

```
cd /home/lfl/install/zookeeper-3.4.10/bin
./zkServer.sh start
```

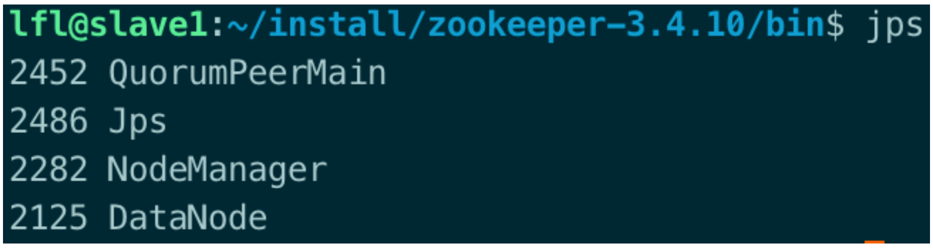
7. 查看进程：

   `jps`

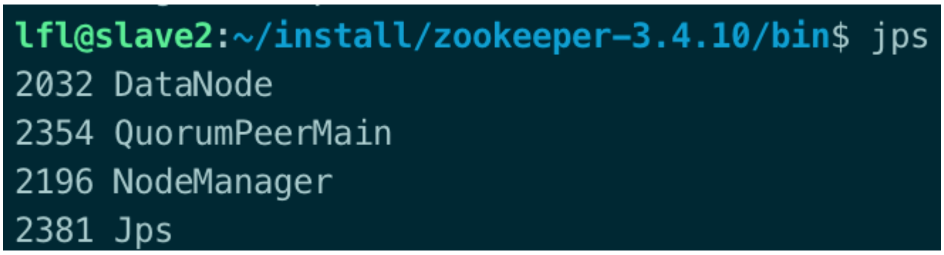
   确认所有节点都存在 QuorumPeerMain 进程



```
lfl@master:~/install/zookeeper-3.4.10/bin$ jps
3938 Jps
3429 SecondaryNameNode
3223 NameNode
3912 QuorumPeerMain
3562 ResourceManager
```



```
lfl@slave1:~/install/zookeeper-3.4.10/bin$ jps
2452 QuorumPeerMain
2486 Jps
2282 NodeManager
2125 DataNode
```



```
lfl@slave2:~/install/zookeeper-3.4.10/bin$ jps
2032 DataNode
2354 QuorumPeerMain
2196 NodeManager
2381 Jps
```

3.7 HBase 环境搭建

1. 解压 HBase：将下载好的 HBase 安装包解压到指定目录

```
tar -zxvf hbase-1.2.1-bin.tar.gz -C ~/install
```

2. 修改配置文件（**hbase的配置目录下**）：

```
cd /home/lfl/install/hbase-1.2.1/conf
```

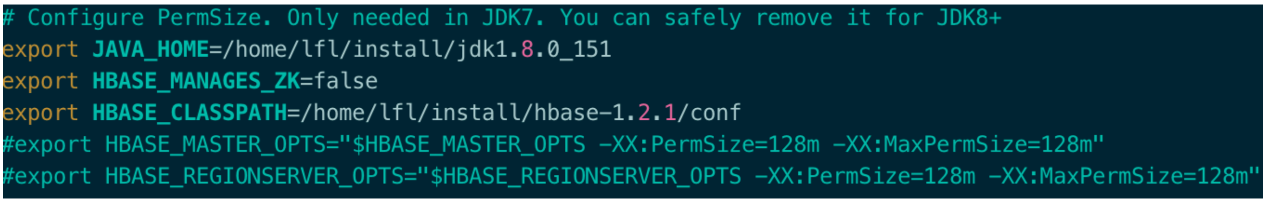
   1. `hbase-env.sh`：指定 JAVA_HOME，设置 HBASE_MANAGES_ZK 为 false

      添加以下内容

```
export JAVA_HOME=/home/lfl/install/jdk1.8.0_151
export HBASE_MANAGES_ZK=false
export HBASE_CLASSPATH=/home/lfl/install/hbase-1.2.1/conf
```

      注释掉以下行：

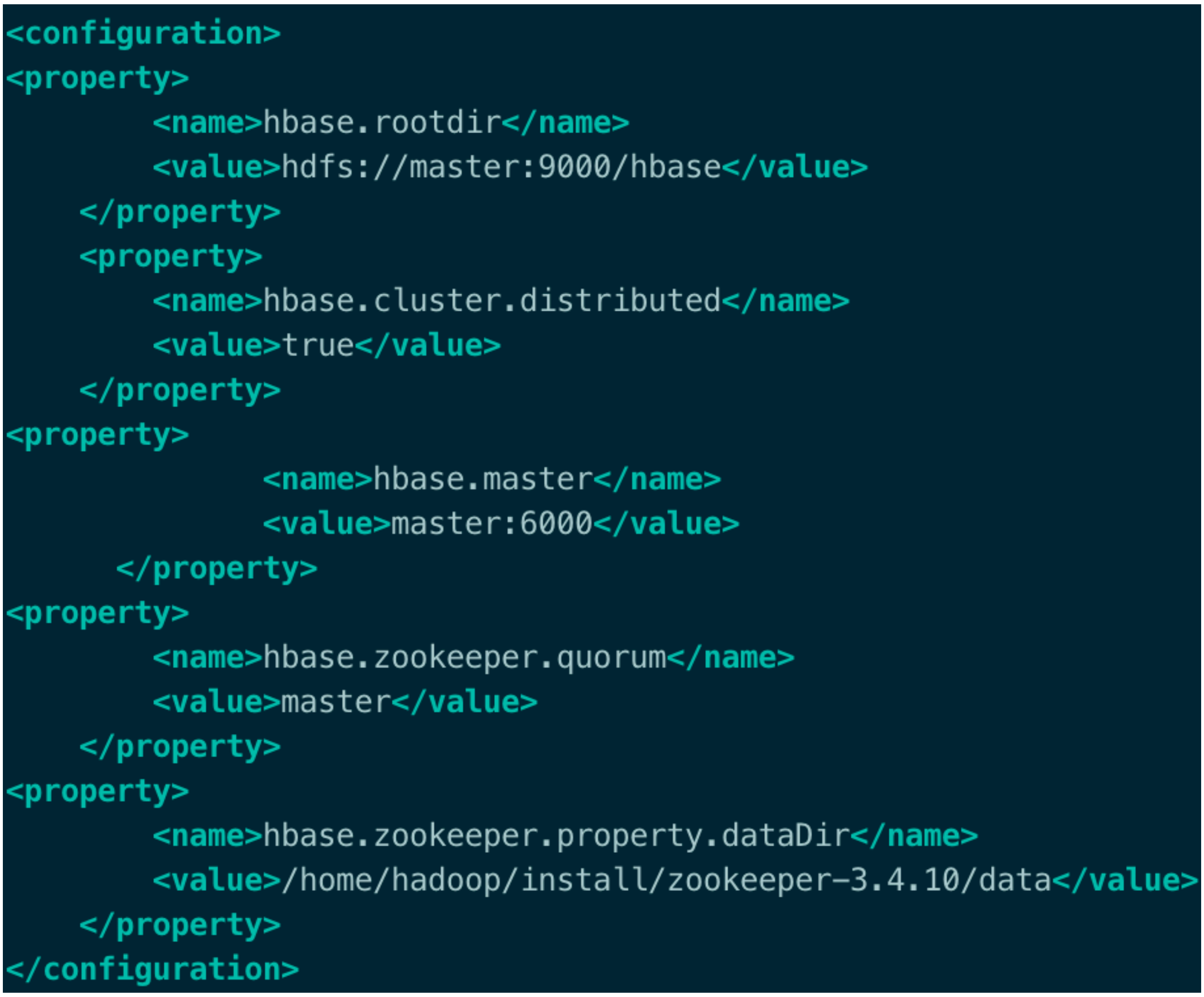
```
#export HBASE_MASTER_OPTS="$HBASE_MASTER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m"
#export HBASE_REGIONSERVER_OPTS="$HBASE_REGIONSERVER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m"
```



```
# Configure PermSize. Only needed in JDK7. You can safely remove it for JDK8+
export JAVA_HOME=/home/lfl/install/jdk1.8.0_151
export HBASE_MANAGES_ZK=false
export HBASE_CLASSPATH=/home/lfl/install/hbase-1.2.1/conf
#export HBASE_MASTER_OPTS="$HBASE_MASTER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m"
#export HBASE_REGIONSERVER_OPTS="$HBASE_REGIONSERVER_OPTS -XX:PermSize=128m -XX:MaxPermSize=128m"
```

2. `hbase-site.xml`：配置 HBase 根目录和 Zookeeper 集群

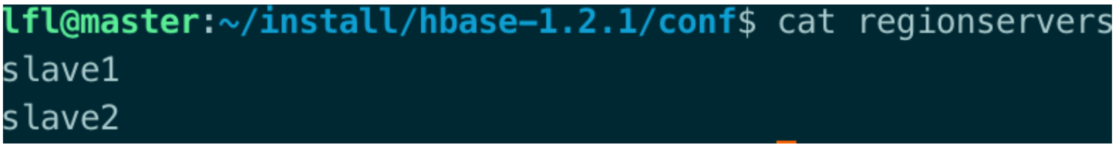
```
<configuration>
<property>
       <name>hbase.rootdir</name>
       <value>hdfs://master:9000/hbase</value>
   </property>
   <property>
       <name>hbase.cluster.distributed</name>
       <value>true</value>
   </property>
<property>
              <name>hbase.master</name>
              <value>master:6000</value>
      </property>
<property>
       <name>hbase.zookeeper.quorum</name>
       <value>master</value>
   </property>
<property>
       <name>hbase.zookeeper.property.dataDir</name>
       <value>/home/hadoop/install/zookeeper-3.4.10/data</value>
   </property>
</configuration>
```



```
<configuration>
<property>
        <name>hbase.rootdir</name>
        <value>hdfs://master:9000/hbase</value>
    </property>
    <property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
    </property>
<property>
              <name>hbase.master</name>
              <value>master:6000</value>
      </property>
<property>
        <name>hbase.zookeeper.quorum</name>
        <value>master</value>
    </property>
<property>
        <name>hbase.zookeeper.property.dataDir</name>
        <value>/home/hadoop/install/zookeeper-3.4.10/data</value>
    </property>
</configuration>
```

3. `regionservers`：添加 slave 节点列表

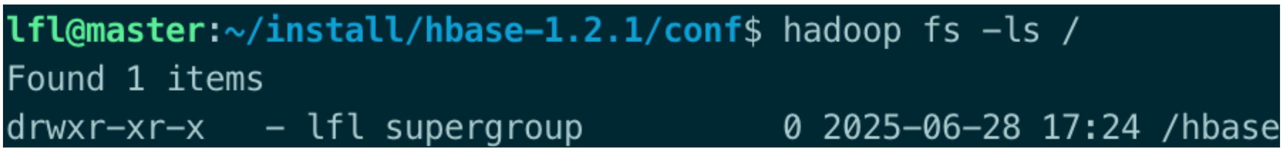
```
slave1
slave2
slave3
slave4
```



```
lfl@master:~/install/hbase-1.2.1/conf$ cat regionservers
slave1
slave2
```

3. 在hadoop 分布式文件系统 HDFS 创建 HBase 目录：

```
hadoop fs -mkdir /hbase
```



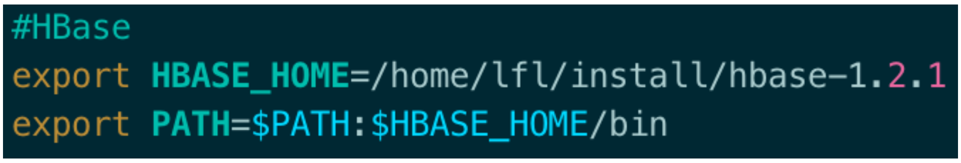
```
lfl@master:~/install/hbase-1.2.1/conf$ hadoop fs -ls /
Found 1 items
drwxr-xr-x   - lfl supergroup          0 2025-06-28 17:24 /hbase
```

4. 配置环境变量：

修改文件

```
/etc/profile
```

```
export HBASE_HOME=/home/lfl/install/hbase-1.2.1
export PATH=$PATH:$HBASE_HOME/bin
```



```
#HBase
export HBASE_HOME=/home/lfl/install/hbase-1.2.1
export PATH=$PATH:$HBASE_HOME/bin
```

5. 将文件夹scp到其它节点服务器上（下示为slave1，slave2同理）

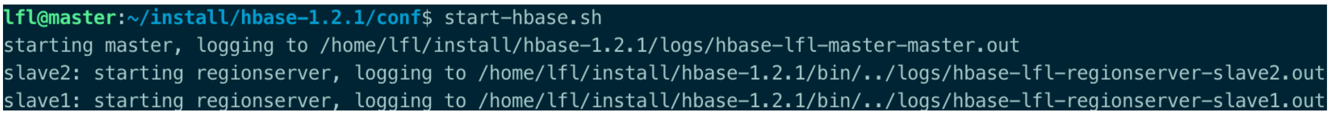
```
$ scp -r ~/install/hbase-1.2.1 lfl@slave1:~/install/hbase-1.2.1
```

6. 将环境变量scp到其它节点服务器上（下示为slave1，slave2同理）

```
$ sudo scp /etc/profile $(用户名)@slave1:/etc
```

7. 启动 HBase：

```
start-hbase.sh
```

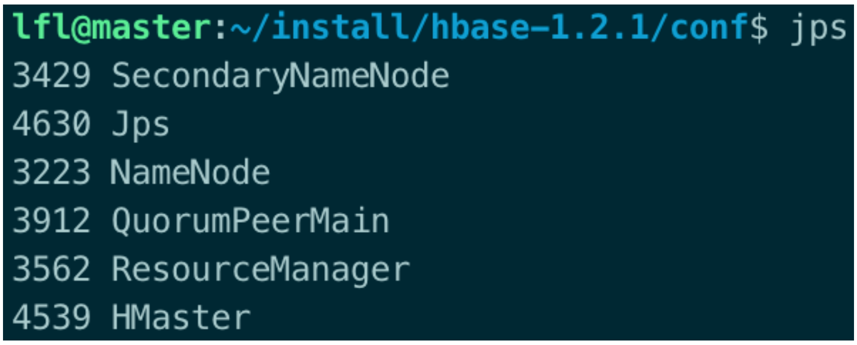


```
lfl@master:~/install/hbase-1.2.1/conf$ start-hbase.sh
starting master, logging to /home/lfl/install/hbase-1.2.1/logs/hbase-lfl-master-master.out
slave2: starting regionserver, logging to /home/lfl/install/hbase-1.2.1/bin/../logs/hbase-lfl-regionserver-slave2.out
slave1: starting regionserver, logging to /home/lfl/install/hbase-1.2.1/bin/../logs/hbase-lfl-regionserver-slave1.out
```

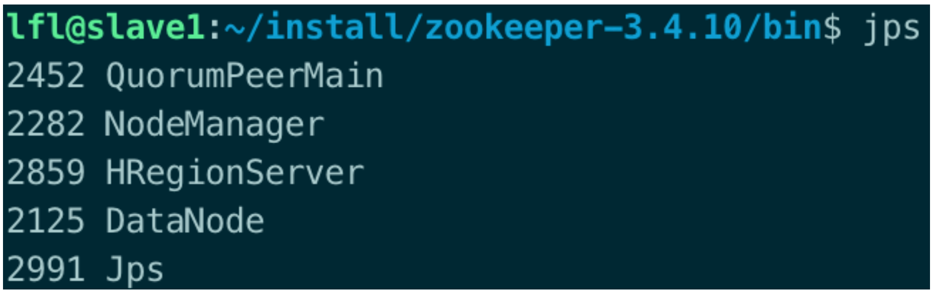
8. 查看进程：

`jps`

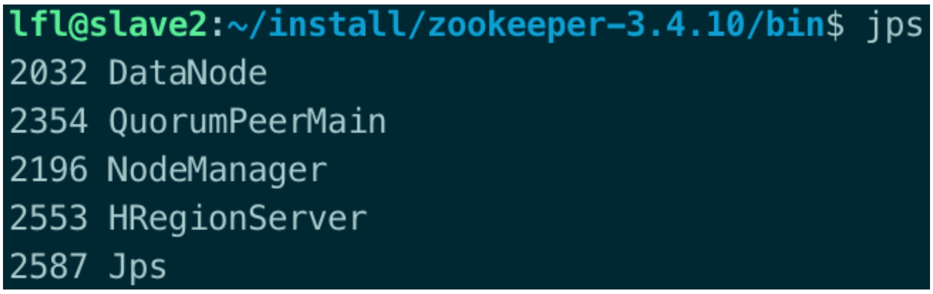
确认master节点有HMaster进程，slave1~2有HRegionServer进程



```
lfl@master:~/install/hbase-1.2.1/conf$ jps
3429 SecondaryNameNode
4630 Jps
3223 NameNode
3912 QuorumPeerMain
3562 ResourceManager
4539 HMaster
```



```
lfl@slave1:~/install/zookeeper-3.4.10/bin$ jps
2452 QuorumPeerMain
2282 NodeManager
2859 HRegionServer
2125 DataNode
2991 Jps
```



```
lfl@slave2:~/install/zookeeper-3.4.10/bin$ jps
2032 DataNode
2354 QuorumPeerMain
2196 NodeManager
2553 HRegionServer
2587 Jps
```

3.8 Spark 环境搭建

1. 解压 Spark：将下载好的 Spark 安装包解压到指定目录

```
tar -xzvf spark-2.2.0-bin-hadoop2.7.tgz -C ~/install
```

2. 修改配置文件（**spark的配置目录下**）

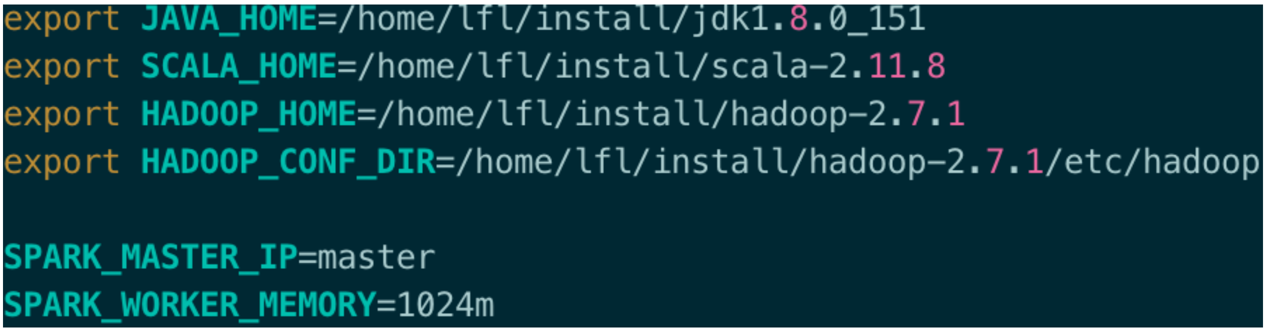
```
$ cd /home/lfl/install/spark-2.2.0-bin-hadoop2.7/conf
```

修改`workers.template`、`spark-env.sh.template`、`spark-defaults.conf.template`文件的文件名（**对于旧版spark，workers.template对应slaves.template，workers对应slaves**）

```
$ mv workers.template workers

$ mv spark-env.sh.template spark-env.sh

$ mv spark-defaults.conf.template spark-defaults.conf
```

   1. `spark-env.sh`：指定 JAVA_HOME、SCALA_HOME、HADOOP_HOME

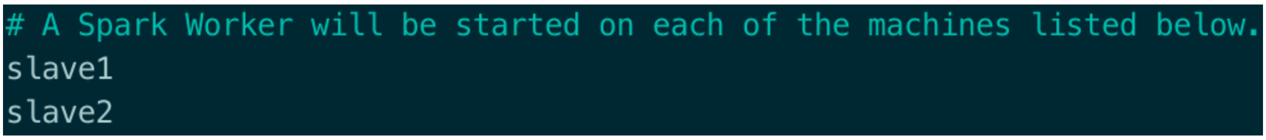
```
export JAVA_HOME=/home/$(用户名)/install/jdk1.8.0_151
export SCALA_HOME=/home/$(用户名)/install/scala-2.11.8
export HADOOP_HOME=/home/$(用户名)/install/hadoop-2.7.1
export HADOOP_CONF_DIR=/home/$(用户名)/install/hadoop-2.7.1/etc/hadoop

SPARK_MASTER_IP=master
SPARK_WORKER_MEMORY=1024m
```



```
export JAVA_HOME=/home/lfl/install/jdk1.8.0_151
export SCALA_HOME=/home/lfl/install/scala-2.11.8
export HADOOP_HOME=/home/lfl/install/hadoop-2.7.1
export HADOOP_CONF_DIR=/home/lfl/install/hadoop-2.7.1/etc/hadoop

SPARK_MASTER_IP=master
SPARK_WORKER_MEMORY=1024m
```

   2. `slaves`：添加 slave 节点列表

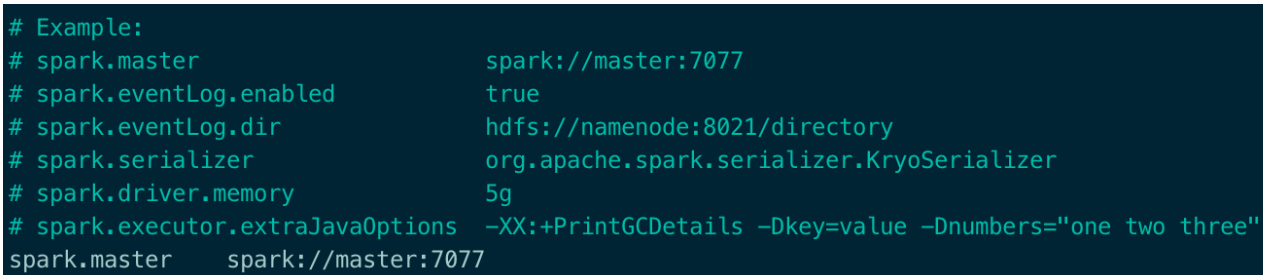
```
slave1
slave2
```



```
# A Spark Worker will be started on each of the machines listed below.
slave1
slave2
```

   3. `spark-defaults.conf`：设置 Spark master 地址

```
spark.master    spark://master:7077
```



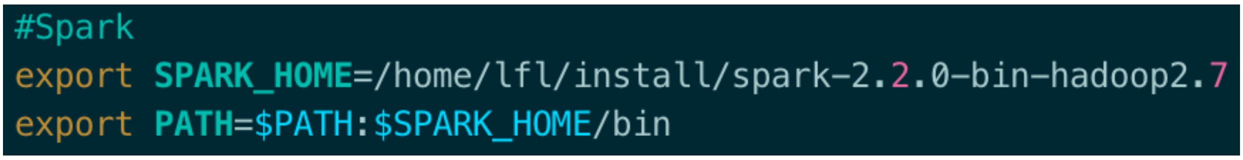
```
# Example:
# spark.master                     spark://master:7077
# spark.eventLog.enabled           true
# spark.eventLog.dir               hdfs://namenode:8021/directory
# spark.serializer                 org.apache.spark.serializer.KryoSerializer
# spark.driver.memory              5g
# spark.executor.extraJavaOptions  -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
spark.master    spark://master:7077
```

3. 配置环境变量：

   修改文件

```
/etc/profile
```

```
export SPARK_HOME=/home/lfl/install/spark-2.2.0-bin-hadoop2.7
export PATH=$PATH:$SPARK_HOME/bin
```



```
#Spark
export SPARK_HOME=/home/lfl/install/spark-2.2.0-bin-hadoop2.7
export PATH=$PATH:$SPARK_HOME/bin
```

4. 将文件夹scp到其它节点服务器上（下示为slave1，slave2同理）

```
scp -r ~/install/spark-2.2.0-bin-hadoop2.7 $(用户名)@slave1:~/install/spark-2.2.0-bin-hadoop2.7
```

5. 将环境变量scp到其它节点服务器上（下示为slave1，slave2同理）

```
$ sudo scp /etc/profile $(用户名)@slave1:/etc
```

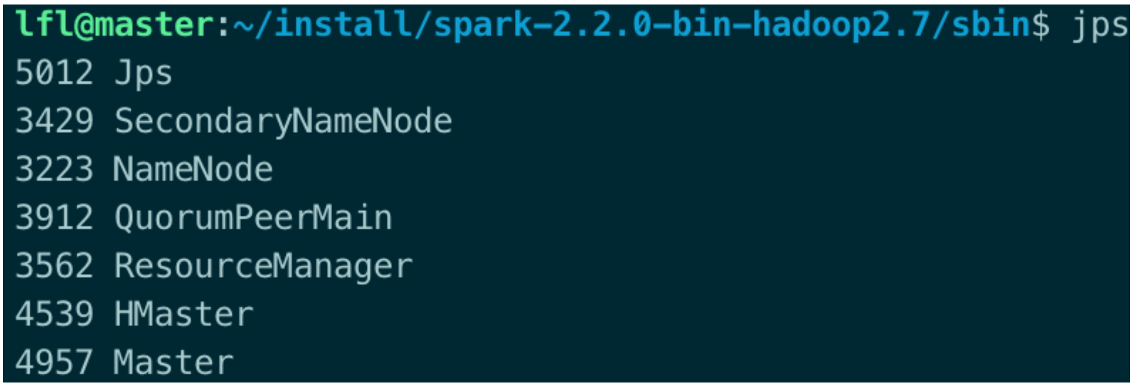
6. 启动 Spark

```
cd /home/lfl/install/spark-2.2.0-bin-hadoop2.7/sbin
./start-all.sh
```

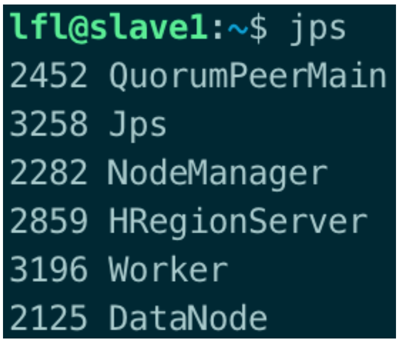
7. 查看进程：

   `jps`

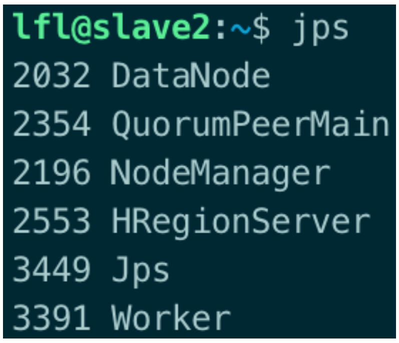
确认 master节点有master进程，slave1~2有worker进程



```
lfl@master:~/install/spark-2.2.0-bin-hadoop2.7/sbin$ jps
5012 Jps
3429 SecondaryNameNode
3223 NameNode
3912 QuorumPeerMain
3562 ResourceManager
4539 HMaster
4957 Master
```



```
lfl@slave1:~$ jps
2452 QuorumPeerMain
3258 Jps
2282 NodeManager
2859 HRegionServer
3196 Worker
2125 DataNode
```



```
lfl@slave2:~$ jps
2032 DataNode
2354 QuorumPeerMain
2196 NodeManager
2553 HRegionServer
3449 Jps
3391 Worker
```

四、实验结果与验证

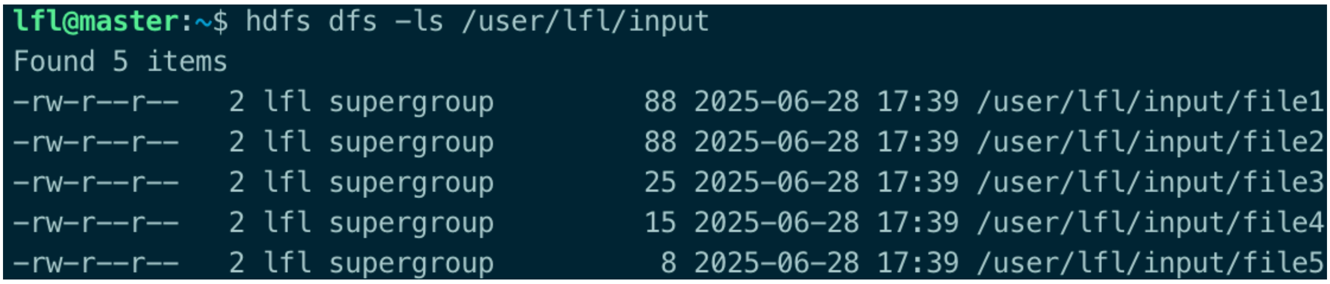
4.1 Hadoop 功能验证

1. 上传文件到 HDFS

```
hdfs dfs -mkdir -p /user/hadoop/input
hdfs dfs -put \
~/spark-data/file1 \
~/spark-data/file2 \
~/spark-data/file3 \
~/spark-data/file4 \
~/spark-data/file5 \
/user/lfl/input/
```

2. 查看文件列表

```
hdfs dfs -ls /user/lfl/input
```



```
lfl@master:~$ hdfs dfs -ls /user/lfl/input
Found 5 items
-rw-r--r--   2 lfl supergroup         88 2025-06-28 17:39 /user/lfl/input/file1
-rw-r--r--   2 lfl supergroup         88 2025-06-28 17:39 /user/lfl/input/file2
-rw-r--r--   2 lfl supergroup         25 2025-06-28 17:39 /user/lfl/input/file3
-rw-r--r--   2 lfl supergroup         15 2025-06-28 17:39 /user/lfl/input/file4
-rw-r--r--   2 lfl supergroup          8 2025-06-28 17:39 /user/lfl/input/file5
```

五、遇到的问题与解决方案

5.1 Hadoop 日志目录权限问题

- **问题描述**：启动 Hadoop 时提示无法创建日志目录

```
java.io.IOException: Cannot create directory /home/lfl/install/hadoop-2.7.1/tmp/dfs/name/current
```

- **解决方案**：手动创建日志目录并设置权限

```
sudo chmod -R 755 /home/lfl/install/hadoop-2.7.1/tmp/
sudo chown -R lfl:lfl /home/lfl/install/hadoop-2.7.1/tmp/
```

5.2 SecondaryNameNode 未启动

- **问题描述**：jps 命令未显示 SecondaryNameNode 进程
- **解决方案**：检查 hdfs-site.xml 配置，确保 SecondaryNameNode 端口正确，重新启动 Hadoop 集群

六、实验总结

本次实验我完成了大数据开发环境的搭建，包括虚拟机配置、SSH 免密登录、JDK、Scala、Hadoop、Zookeeper、HBase 和 Spark 的安装与配置。成功验证了 Hadoop 的文件存储功能和 HBase 的表操作，为后续的大数据处理实验奠定了基础，在搭建过程中我遇到了一些权限、配置和依赖问题，通过查阅文档和网上资料得以解决，加深了我对大数据组件工作原理的理解，未来可进一步学习各组件的高级配置和优化，以及如何在集群上运行实际的大数据处理任务。

启动命令：

Hadoop:

```
$ start-all.sh
```

Zookeeper:

```
$ cd /home/lfl/install/zookeeper-3.4.10/bin
$ ./zkServer.sh start
```

Hbase:

```
$ start-hbase.sh
```

Spark:

```
$ cd /home/lfl/install/spark-2.2.0-bin-hadoop2.7/sbin
$ ./start-all.sh
```