

# 实验二 HBase的安装与使用

组号：24

组员： 2020212185

思维导图：



大纲（可点击快速跳转）：

## 实验二 HBase的安装与使用

### 伪分布式(Pseudo Distributed)下的HBase安装与配置

- 1-安装HBase
- 2-配置HBase

### 完全分布式(Fully Distributed)下的HBase安装配置与操作

- 1-安装HBase
- 2-配置HBase
- 出现的问题
- HBase Shell的操作

### 使用Java操作HBase

- 1-安装IDEA
- 2-安装maven
- 3-在IDEA中配置maven
- 4-配置log4j
- 5-在java中调用HBase

Build Success

# 伪分布式(Pseudo Distributed)下的HBase 安装与配置

实验环境如下：

Ubuntu 22.04 虚拟机

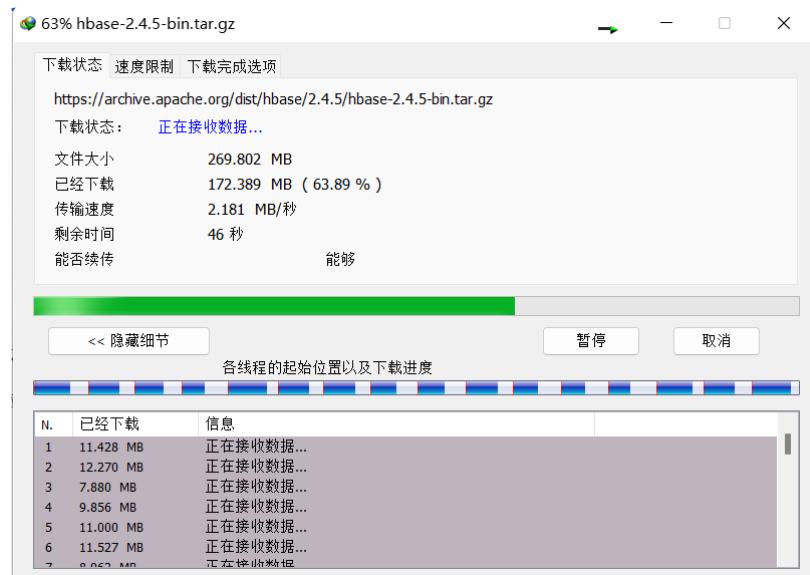
JDK 1.8

Hadoop 2.10.1

HBase 2.4.5

## 1-安装HBase

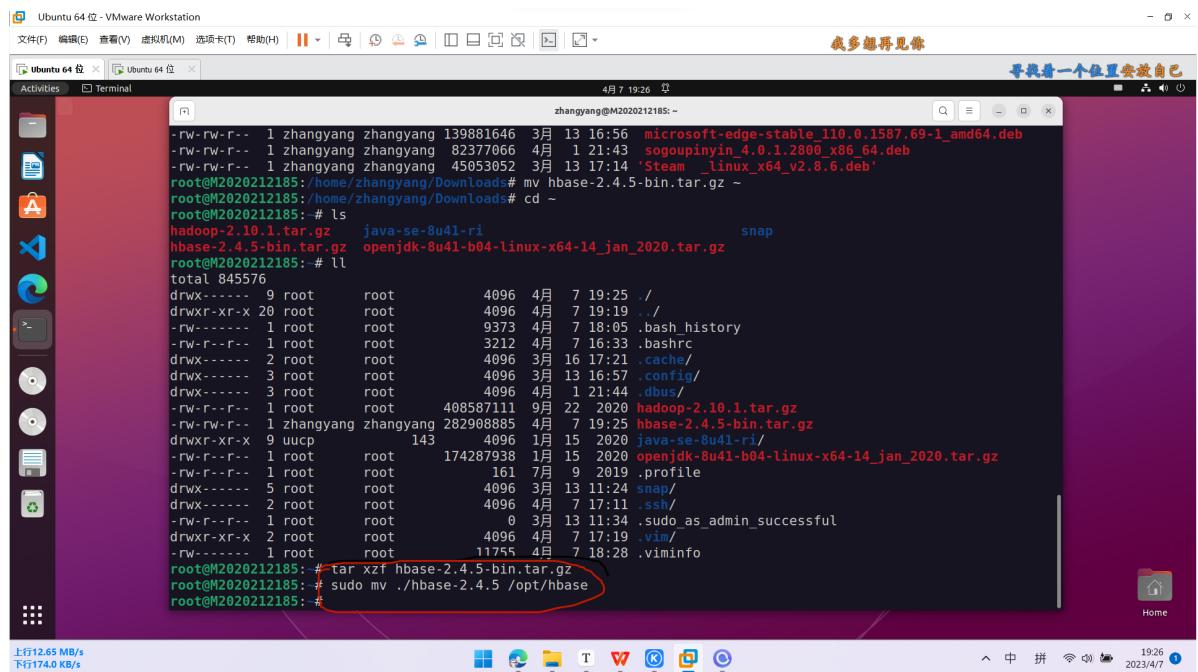
与hadoop2.10.1对应的HBase版本之一是2.4.5。从<https://archive.apache.org/dist/hbase/2.4.5/hbase-2.4.5-bin.tar.gz>下载对应的版本



下载相关文件，并将其放在想安装的位置。为了方便，选择与上一份实验手册相同的安装位置：`/opt/hbase`。

```
1 | tar xzf hbase-2.4.5-bin.tar.gz
2 | sudo mv ./hbase-2.4.5 /opt/hbase
```

张扬2020212185的截图：



## 2-配置HBase

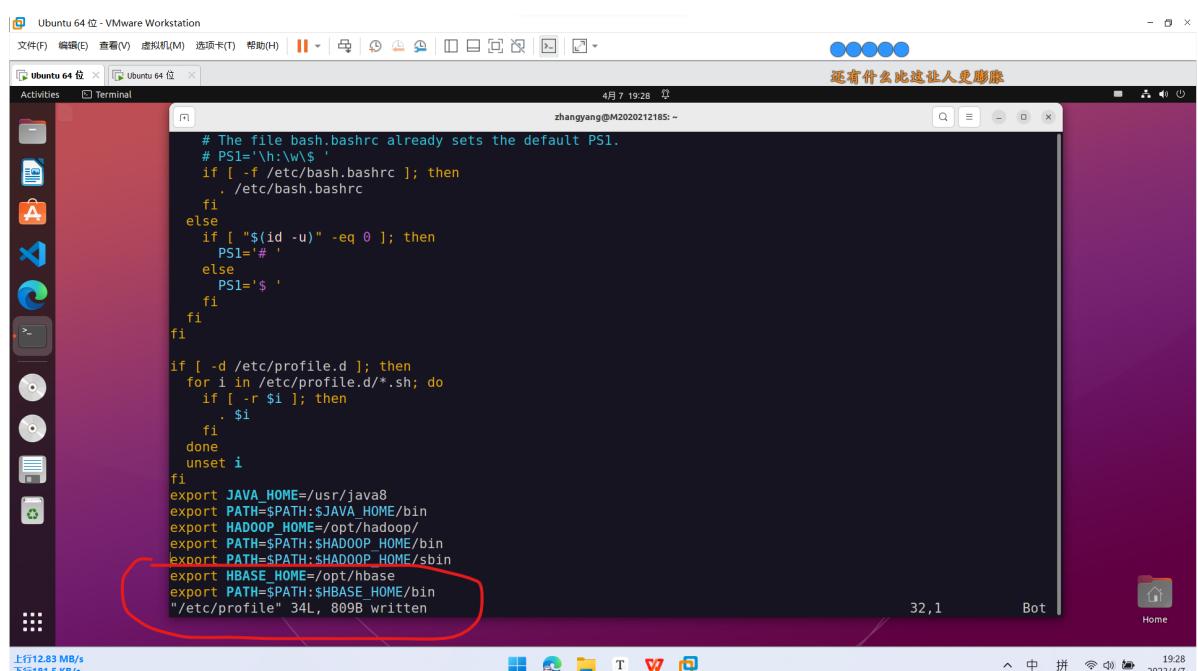
为了方便操作，为HBase配置环境变量：

```
1 | vim /etc/profile
```

在profile末尾添加以下两行配置：

```
1 | export HBASE_HOME=/opt/hbase
2 | export PATH=$PATH:$HBASE_HOME/bin
```

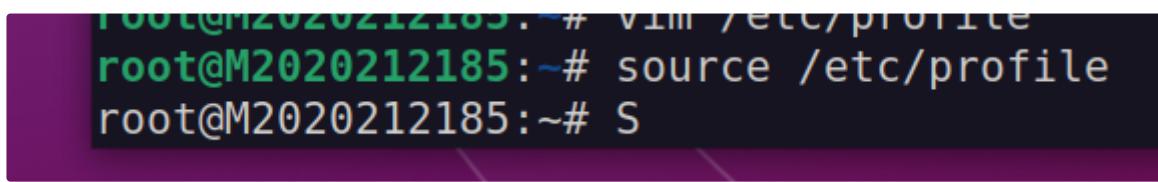
### 张扬2020212185的截图：



更新环境变量：

```
1 | source /etc/profile
```

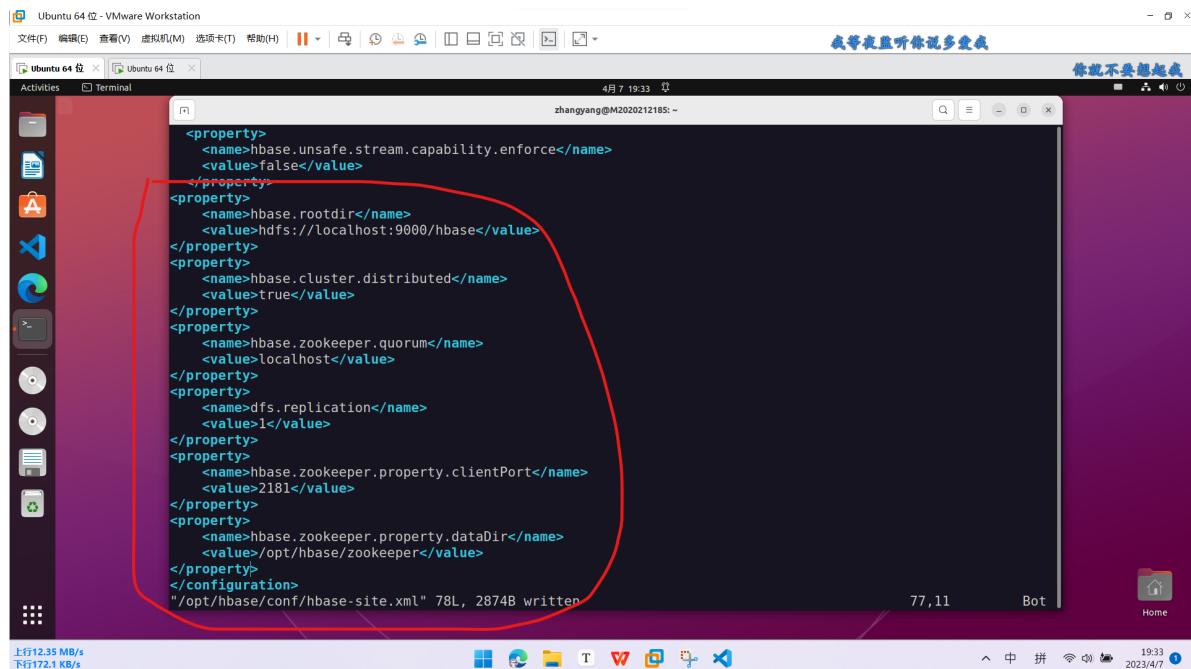
张扬2020212185的截图：



```
root@m2020212185:~# vim /etc/profile
root@m2020212185:~# source /etc/profile
root@m2020212185:~# S
```

修改配置文件 `/opt/hbase/conf/hbase-site.xml` :

张扬2020212185的截图：



注意，`hbase.rootdir` 属性的值应为 `hdfs://localhost:9000/hbase` ;  
`dfs.replication` 可以设置为1( 非必须)；因为是伪分布式，所以  
`hbase.cluster.distributed` 可以为 `true` 。

编辑 `hbase-env.sh` 文件，添加以下配置，来配置区域服务器的路径，并启动自动管理 `zookeeper`：

张扬2020212185的截图：

```
# Seconds to sleep between slave commands. Unset by default. This
# can be useful in large clusters, where, e.g., slave rsyncs can
# otherwise arrive faster than the master can service them.
# export HBASE_SLAVE_SLEEP=0.1

# Tell HBase whether it should manage its own instance of ZooKeeper or not.
# export HBASE_MANAGES_ZK=true

# The default log rolling policy is RFA, where the log file is rolled as per the size defined for the
# RFA appender. Please refer to the log4j.properties file to see more details on this appender.
# In case one needs to do log rolling on a date change, one should set the environment property
# HBASE_ROOT_LOGGER to "<DESIRED_LOG_LEVEL>,DRFA".
# For example:
# HBASE_ROOT_LOGGER=INFO,DRFA
# The reason for changing default to RFA is to avoid the boundary case of filling out disk space as
# DRFA doesn't put any cap on the log size. Please refer to HBase-5655 for more context.

# Tell HBase whether it should include Hadoop's lib when start up,
# the default value is false, means that includes Hadoop's lib.
# export HBASE_DISABLE_HADOOP_CLASSPATH_LOOKUP="true"

# Override text processing tools for use by these launch scripts.
# export GREP="${GREP-grep}"
# export SED="${SED-sed}"

export JAVA_HOME=/usr/java8
export HBASE_REGIONSERVERS=$HBASE_HOME/conf/regionservers
export HBASE_MANAGES_ZK=true
"hbbase-env.sh" 147L, 7790B written
```

启动hbase进行测试

- 1 start-all.sh
- 2 start-hbase.sh
- 3 hbase shell

张扬2020212185的截图：

```
root@M2020212185:~# start-dfs.sh
Starting namenodes on [M2020212185]
M2020212185: namenode running as process 4680. Stop it first.
M2020212185: starting datanode, logging to /opt/hadoop/logs/hadoop-root-datanode
-M2020212185.out
S2020212194: starting datanode, logging to /opt/hadoop/logs/hadoop-root-datanode
-S2020212194.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 5012. Stop it first.
root@M2020212185:~# start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop/logs/yarn-root-resourcemanager-
M2020212185.out
M2020212185: starting nodemanager, logging to /opt/hadoop/logs/yarn-root-nodemanager-M2020212185.out
S2020212194: starting nodemanager, logging to /opt/hadoop/logs/yarn-root-nodemanager-S2020212194.out
root@M2020212185:~# start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
```

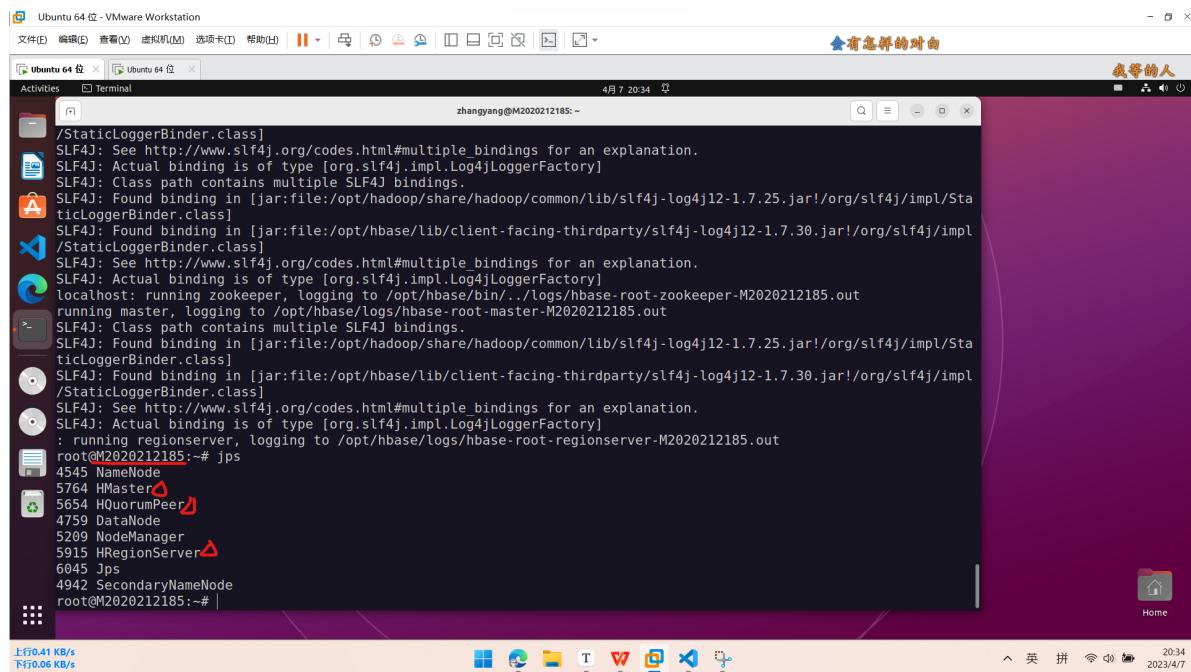
```
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:GBRqkO5Xs7f+eownuRK2c+x0bp9CJCxxUvJBA0ultU.
This host key is known by the following other names/addresses:
  -./ssh/known_hosts:1: [hashed name]
  -./ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
localhost: Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
localhost: running zookeeper, logging to /opt/hbase/bin/../logs/hbase-root-zookeper-M2020212185.out
running master, logging to /opt/hbase/logs/hbase-root-master-M2020212185.out
: running regionserver, logging to /opt/hbase/logs/hbase-root-regionserver-M2020212185.out
root@M2020212185: # hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.5_r03b8c0cf426cbae3284225b73040ec574d5bac34, Tue Jul 27 09:44:16 PDT 2021
Took 0.0023 seconds
hbase:001:>
```

运行 `jps` 指令，如果出现 `HQuorumPeer`，`HRegionServer` 和 `HMMaster`，则证明启动成功：

## 张扬2020212185的截图：

```
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:GBRqkO5Xs7f+eownuRK2c+x0bp9CJCxxUvJBA0ultU.
This host key is known by the following other names/addresses:
  -./ssh/known_hosts:1: [hashed name]
  -./ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
localhost: Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
localhost: running zookeeper, logging to /opt/hbase/bin/../logs/hbase-root-zookeper-M2020212185.out
running master, logging to /opt/hbase/logs/hbase-root-master-M2020212185.out
: running regionserver, logging to /opt/hbase/logs/hbase-root-regionserver-M2020212185.out
root@M2020212185: # hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.5_r03b8c0cf426cbae3284225b73040ec574d5bac34, Tue Jul 27 09:44:16 PDT 2021
Took 0.0023 seconds
hbase:001:>

zhangyang@M2020212185:~$ sudo su
[sudo] password for zhangyang:
root@M2020212185:~/home/zhangyang# cd ~
root@M2020212185: # jps
Command 'jps' not found, but can be installed with:
apt install openjdk-11-jdk-headless # version 11.0.18+10-0ubuntu1
apt install openjdk-17-jdk-headless # version 17.0.6+10-0ubuntu1
apt install openjdk-18-jdk-headless # version 18.0.2+9-2-22
apt install openjdk-19-jdk-headless # version 19.0.2+7-0ubuntu1
apt install openjdk-8-jdk-headless # version 8u362-ga-0ubuntu1
apt install openjdk-13-jdk-headless # version 13.0.7+5-0ubuntu1
apt install openjdk-16-jdk-headless # version 16.0.1+9-1-20
root@M2020212185: # source /etc/profile
root@M2020212185: # jps
8912 HQuorumPeer
9649 Jps
5012 SecondaryNameNode
9159 HRegionServer
3463 JarBootstrapMain
3472 NodeManager
4080 NameNode
root@M2020212185:~|
```



# 完全分布式(Fully Distributed)下的HBase 安装配置与操作

## 1-安装HBase

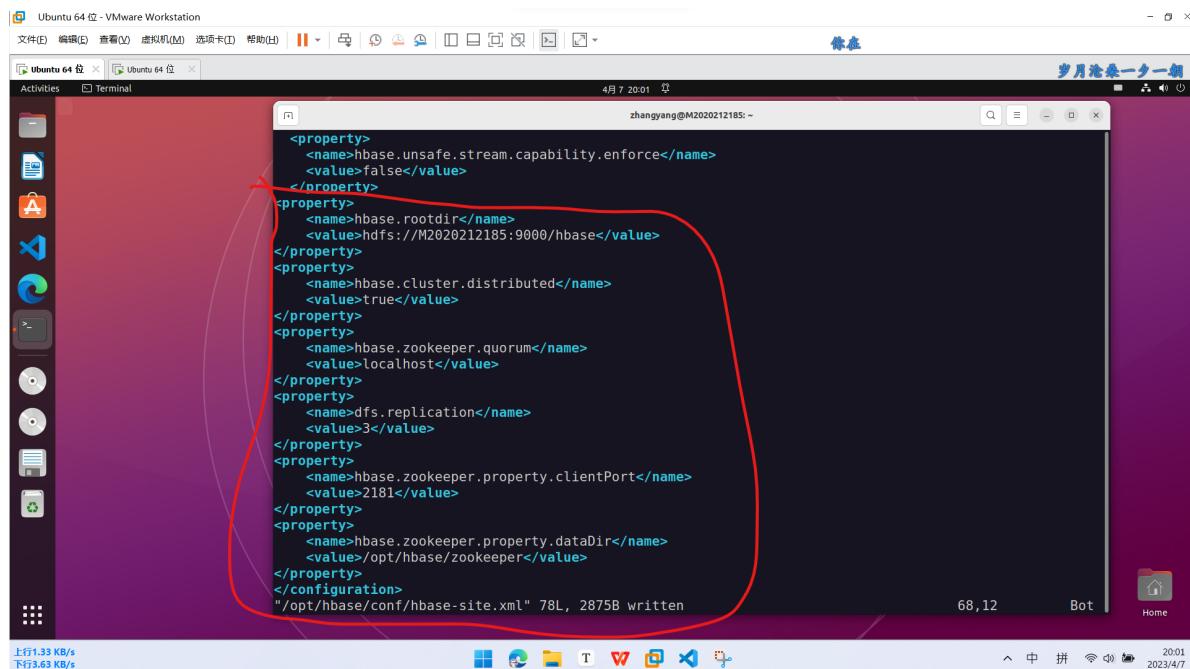
同伪分布，多两台slave

## 2-配置HBase

编辑 `/etc/profile` 同伪分布。

HBase默认写到 `/${java.io.tmpdir}/HBase-${user.name}` 。  
`{.io.tmpdir}` 通常映射到 /tmp 目录下。可以修改配置文件 `/opt/hbase/conf/hbase-site.xml` 以规定其映射目录；同时需要修改配置文件来指定内置zookeeper的运行文件保存位置：

张扬2020212185的截图：



```
1 <property>
2   <name>hbase.rootdir</name>
3   <value>hdfs://M2020212185:9000/hbase</value>
4 </property>
5 <property>
6   <name>hbase.cluster.distributed</name>
7   <value>true</value>
8 </property>
9 <property>
10  <name>hbase.zookeeper.quorum</name>
11  <value>localhost</value>
12 </property>
13 <property>
14   <name>dfs.replication</name>
15   <value>3</value>
16 </property>
17 <property>
18   <name>hbase.zookeeper.property.clientPort</name>
19   <value>2181</value>
20 </property>
21 <property>
22   <name>hbase.zookeeper.property.dataDir</name>
23   <value>/opt/hbase/zookeeper</value>
24 </property>
25 </configuration>
```

**hbase.rootdir** 属性的配置应在之前配置的HDFS的 **fs.defaultFS** 属性上添加 hbase 目录；在 **dfs.replication** 属性中设置副本数量。默认情况下，将副本设置为 1。在完全分布式模式下，存在多个数据节点，因此我们可以通过设置 **dfs.replication** 为超过1的值。

编辑 **hbase-env.sh** 文件，添加以下配置，来配置区域服务器的路径，并启动自动管理 **zookeeper**：

同伪分布。

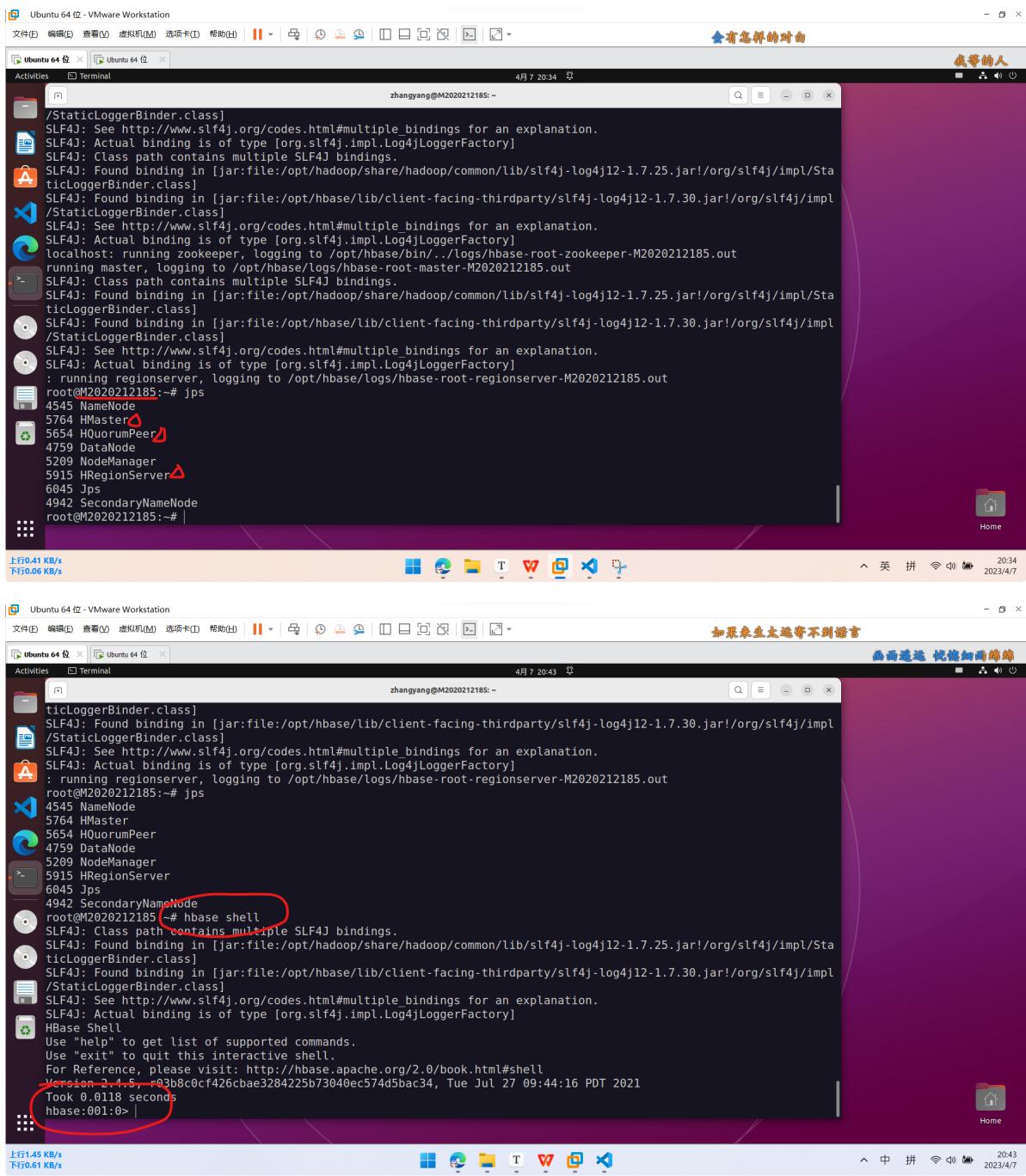
启动hbase进行测试，指令如下：

```
1 start-all.sh  
2 start-hbase.sh  
3 hbase shell
```

张扬2020212185的截图：

The screenshot shows a terminal window on an Ubuntu 64-bit VM. The terminal output is as follows:

```
*****  
SHUTDOWN MSG: Shutting down NameNode at M2020212185/10.241.164.219  
*****  
root@M2020212185:~# jps  
4347 Jps  
root@M2020212185:~# start-dfs.sh  
Starting namenodes on [M2020212185]  
M2020212185: starting namenode, logging to /opt/hadoop/logs/hadoop-root-namenode-M2020212185.out  
M2020212185: starting datanode, logging to /opt/hadoop/logs/hadoop-root-datanode-M2020212185.out  
S2020212194: starting datanode, logging to /opt/hadoop/logs/hadoop-root-datanode-S2020212194.out  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-root-secondarynamenode-M2020212185.out  
root@M2020212185:~# jps  
4545 NameNode  
5044 Jps  
4759 DataNode  
4942 SecondaryNameNode  
root@M2020212185:~# (start-yarn.sh  
starting yarn daemons  
starting resourcemanager, logging to /opt/hadoop/logs/yarn-root-resourcemanager-M2020212185.out  
S2020212194: starting nodemanager, logging to /opt/hadoop/logs/yarn-root-nodemanager-S2020212194.out  
M2020212185: starting nodemanager, logging to /opt/hadoop/logs/yarn-root-nodemanager-M2020212185.out  
root@M2020212185:~# jps  
4545 NameNode  
5091 ResourceManager  
4759 DataNode  
5209 NodeManager  
4942 SecondaryNameNode  
root@M2020212185:~#
```



## 出现的问题

## 一直没有 Hmaster

```
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:GBRqK05Xs7f+ewnuRK2c+x0bp9CJCxxUvJBA0uLTU.
This host key is known by the following other names/addresses:
  -./ssh/known_hosts:1: [hashed name]
  -./ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
localhost: Warning: Permanently added 'localhost' (ED25519) to the list of known
hosts.
localhost: running zookeeper, logging to /opt/hbase/bin/../logs/hbase-root-zooke
eper-M2020212185.out
running master, logging to /opt/hbase/logs/hbase-root-master-M2020212185.out
: running regionserver, logging to /opt/hbase/logs/hbase-root-regionserver-M2020
2185.out
root@M2020212185:~# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j1.2.17.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.5, r03b8c0cf426cbea3284225b73040ec574d5bac34, Tue Jul 27 09:44:16 PD
T 2021
Took 0.0023 seconds
hbase:001:0> |
```

```
zhangyang@M2020212185:~$ sudo su
[sudo] password for zhangyang:
root@M2020212185:/home/zhangyang# cd ~
root@M2020212185:~# jps
Command 'jps' not found, but can be installed with:
apt install openjdk-11-jdk-headless # version 11.0.18+10-0ubu
apt install openjdk-17-jdk-headless # version 17.0.6+10-0ubu
apt install openjdk-18-jdk-headless # version 18.0.2+9-2-22
apt install openjdk-19-jdk-headless # version 19.0.2+7-0ubu
apt install openjdk-8-jdk-headless # version 8u362-ga-0ubu
apt install openjdk-13-jdk-headless # version 13.0.7+5-0ubu
apt install openjdk-16-jdk-headless # version 16.0.1+9-1-20
root@M2020212185:~# source /etc/profile
8912 HQuorumPeer
9649 Jps
5012 SecondaryNameNode
9159 HRegionServer
3463 JarBootstrapMain
3472 NodeManager
4189 NameNode
root@M2020212185:~|
```

后来发现是因为当我们使用 `hadoop namenode -format` 格式化 namenode 时，会在 `namenode` 数据文件夹（这个文件夹为自己配置文件中 `dfs.name.dir` 的路径）中保存一个 `current/VERSION` 文件，记录 `clusterID`，`datanode` 中保存的 `current/VERSION` 文件中的 `clusterID` 的值是上一次格式化保存的 `clusterID`，这样，`datanode` 和 `namenode` 之间的 ID 不一致。导致 `datanode` 和 `namenode` 无法正常启动，从而 `Hmaster` 无法启动

需要在所有机器上，重启，删除临时文件，在 master 重新格式化 namenode

```
1 | rm -rf /tmp/*
2 | rm -rf /opt/hadoop/tmp/*
3 | hdfs namenode -format
```

成功解决！

## HBase Shell的操作

我们创建一个名为 `test` 的表，并创建一个名为 `data` 的列族(Column Family)，指令如下：

```
1 | create 'test', 'data'
```

张扬2020212185的截图：

The screenshot shows a terminal window titled "Ubuntu 64 位 - VMware Workstation". The terminal session is as follows:

```
zhangyang@M2020212185:~$ jps
4545 NameNode
5764 HMaster
5654 HQuorumPeer
4759 DataNode
5209 NodeManager
5915 HRegionServer
6045 Jps
4942 SecondaryNameNode
root@M2020212185:~# hbase shell
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
: running regionserver, logging to /opt/hbase/logs/hbase-root-regionserver-M2020212185.out
root@M2020212185:~# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.5, r03b8c0cf426cbae3284225b7304ec574d5bac34, Tue Jul 27 09:44:16 PDT 2021
Took 0.0118 seconds
hbase:001:0> create 'test', 'data'
Created table test
Took 2.1022 seconds
=> Hbase:002:0>
hbase:002:0>
```

注意：HBase的表中会有一个系统默认的属性作为行键，无需自行创建，默认为put命令操作中表名后第一个数据。

为了证明新表已成功创建，运行 `list` 命令，输出用户空间中的所有表：

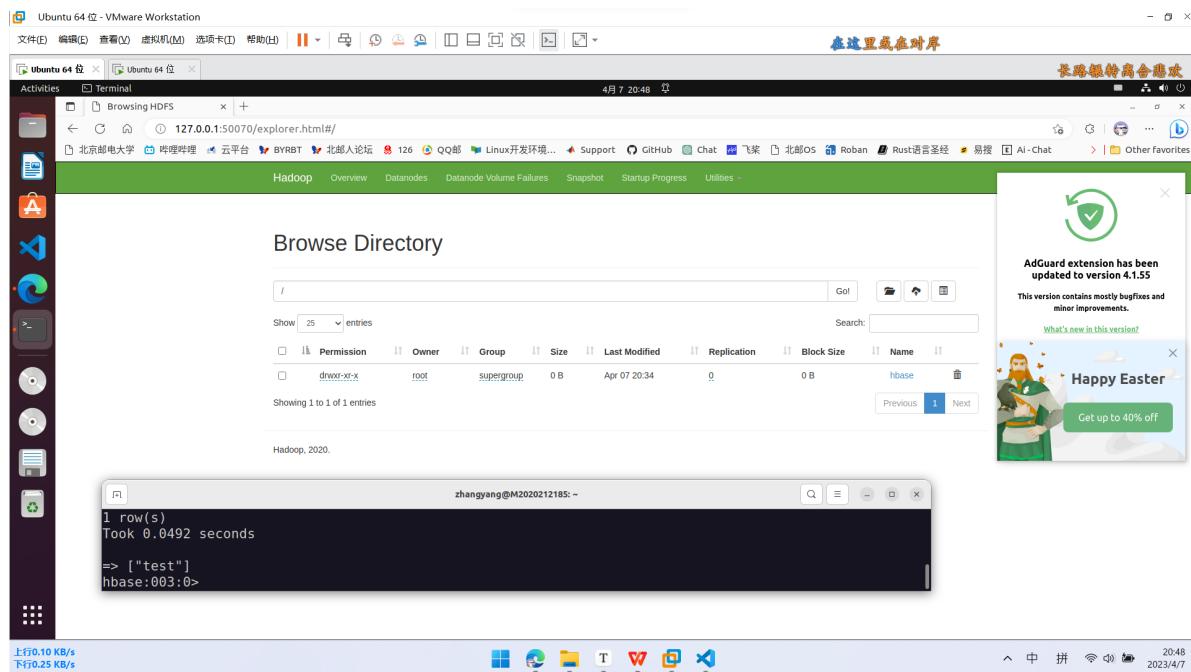
### 张扬2020212185的截图：

The screenshot shows a terminal window titled "Ubuntu 64 位 - VMware Workstation". The terminal session is as follows:

```
zhangyang@M2020212185:~$ jps
4759 DataNode
5209 NodeManager
5915 HRegionServer
6045 Jps
4942 SecondaryNameNode
root@M2020212185:~# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.5, r03b8c0cf426cbae3284225b7304ec574d5bac34, Tue Jul 27 09:44:16 PDT 2021
Took 0.0118 seconds
hbase:001:0> create 'test', 'data'
Created table test
Took 2.1022 seconds
=> Hbase:002:0>
hbase:002:0> list
TABLE
test
1 row(s)
Took 0.0492 seconds
=> ["test"]
hbase:003:0>
```

以上过程会将表 `test` 保存到HDFS上，位置为 `/hbase/data/default` 即我们在 `hbase-site.xml` 上配置的 `hbase.rootdir`。打开 `http://127.0.0.1:50070/`，选择 `utilities/Browse the file system` 查看：

### 张扬2020212185的截图：



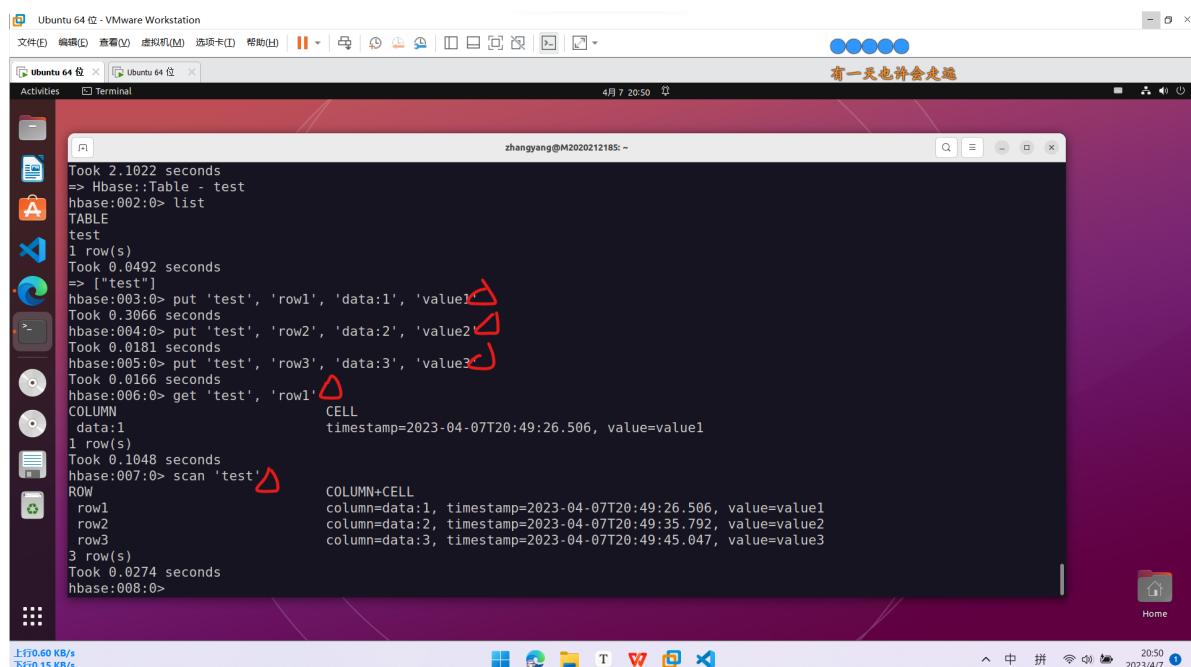
我们将数据插入到数据列族中的三个不同行和列中，获取第一行，然后列出表内容，指令如下：

```

1 put 'test', 'row1', 'data:1', 'value1'
2 put 'test', 'row2', 'data:2', 'value2'
3 put 'test', 'row3', 'data:3', 'value3'
4 get 'test', 'row1'
5 scan 'test'

```

### 张扬2020212185的截图：



对 row1 的 data1 进行修改：

```

1 put 'test', 'row1', 'data:1', 'new value'
2 get 'test', 'row1'

```

## 张扬2020212185的截图：

The screenshot shows a terminal window titled "Ubuntu 64 位 - VMware Workstation". The command history in the terminal is as follows:

```
=> ["test"]
hbase:003:0> put 'test', 'row1', 'data:1', 'value1'
Took 0.3066 seconds
hbase:004:0> put 'test', 'row2', 'data:2', 'value2'
Took 0.0181 seconds
hbase:005:0> put 'test', 'row3', 'data:3', 'value3'
Took 0.0166 seconds
hbase:006:0> get 'test', 'row1'
COLUMN                                CELL
data:1                               timestamp=2023-04-07T20:49:26.506, value=value1
1 row(s)
Took 0.1048 seconds
hbase:007:0> scan 'test'
ROW                                     COLUMN+CELL
row1                                  column=data:1, timestamp=2023-04-07T20:49:26.506, value=value1
row2                                  column=data:2, timestamp=2023-04-07T20:49:35.792, value=value2
row3                                  column=data:3, timestamp=2023-04-07T20:49:45.047, value=value3
3 row(s)
Took 0.0274 seconds
hbase:008:0> put 'test', 'row1', 'data:1', 'new value' ①
Took 0.0108 seconds
hbase:009:0> get 'test', 'row1' ②
COLUMN                                CELL
data:1                               timestamp=2023-04-07T20:51:55.855, value=new value
1 row(s)
Took 0.0147 seconds
hbase:010:0> |
```

如果想删除表，指令如下：

```
1 disable 'test'
2 drop 'test'
3 list
```

## 张扬2020212185的截图：

The screenshot shows a terminal window titled "Ubuntu 64 位 - VMware Workstation". The command history in the terminal is as follows:

```
data:1                               timestamp=2023-04-07T20:49:26.506, value=value1
1 row(s)
Took 0.1048 seconds
hbase:007:0> scan 'test'
ROW                                     COLUMN+CELL
row1                                  column=data:1, timestamp=2023-04-07T20:49:26.506, value=value1
row2                                  column=data:2, timestamp=2023-04-07T20:49:35.792, value=value2
row3                                  column=data:3, timestamp=2023-04-07T20:49:45.047, value=value3
3 row(s)
Took 0.0274 seconds
hbase:008:0> put 'test', 'row1', 'data:1', 'new value'
Took 0.0108 seconds
hbase:009:0> get 'test', 'row1'
COLUMN                                CELL
data:1                               timestamp=2023-04-07T20:51:55.855, value=new value
1 row(s)
Took 0.0147 seconds
hbase:010:0> disable 'test' ③
Took 0.7404 seconds
hbase:011:0> drop 'test' ④
Took 0.6953 seconds
hbase:012:0> list ⑤
TABLE
0 row(s)
Took 0.0073 seconds
=> []
hbase:013:0>
```

如图， test表已经被删除了。

在HDFS的 `/hbase/data/default` 目录下的test也被删除了。

离开HBase Shell的指令为 `exit` :

HBase的关闭方法为：

1 | stop-hbase.sh

### 张扬2020212185的截图：

```
zhangyang@M2020212185:~$ hbase:012:0> list
TABLE
0 row(s)
Took 0.0073 seconds
=> []
hbase:013:0> exit
root@M2020212185:~# stop-hbase.sh
stopping hbase.....
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
localhost: running zookeeper, logging to /opt/hbase/bin/../logs/hbase-root-zookeeper-M2020212185.out
localhost: stopping zookeeper.
root@M2020212185:~#
```

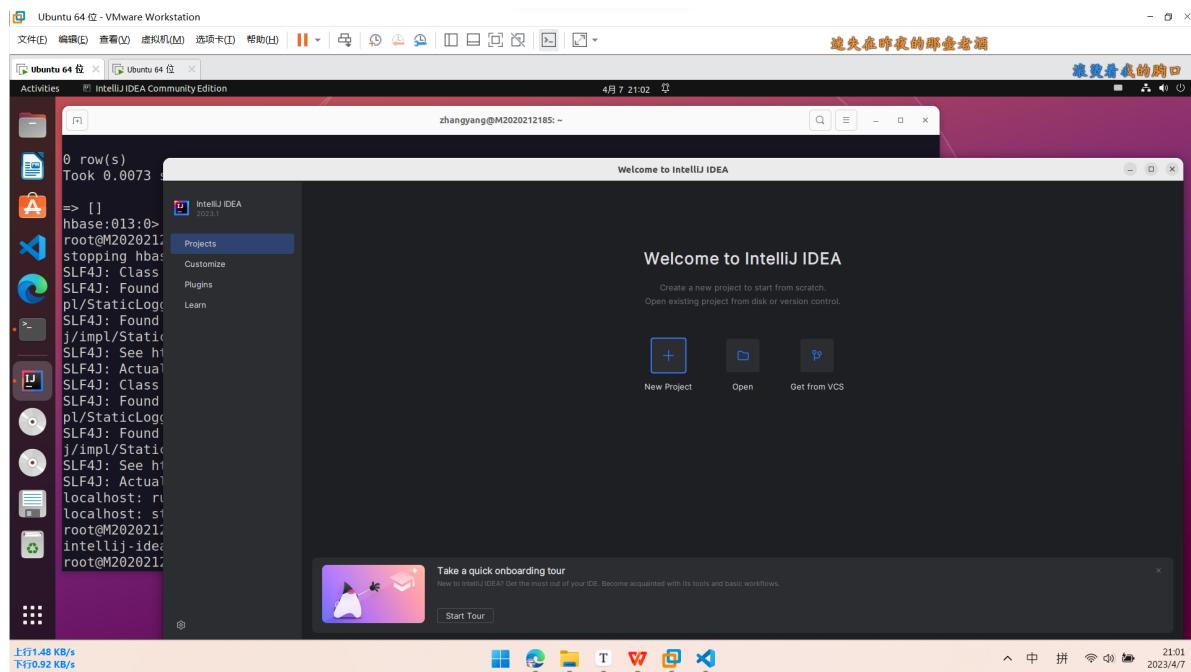
## 使用Java操作HBase

### 1-安装IDEA

1 | sudo snap install intellij-idea-community --classic

### 张扬2020212185的截图：

```
zhangyang@M2020212185:~$ TABLE
0 row(s)
Took 0.0073 seconds
=> []
hbase:013:0> exit
root@M2020212185:~# stop-hbase.sh
stopping hbase.....
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
localhost: running zookeeper, logging to /opt/hbase/bin/../logs/hbase-root-zookeeper-M2020212185.out
localhost: stopping zookeeper.
root@M2020212185:~# sudo snap install intellij-idea-community --classic
Download snap "intellij-idea-community" (427) from channel 'stable'
59% 10.3MB/s 40.6s
```



## 2-安装maven

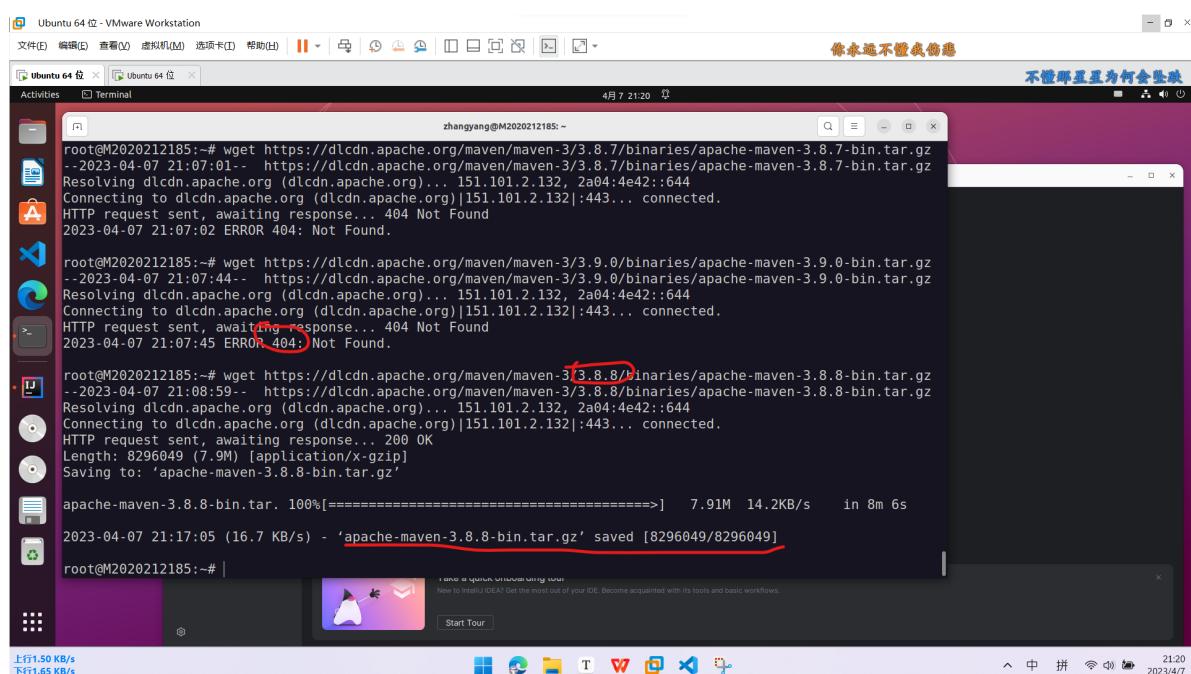
使用以下命令下载并解压maven3.9.0：

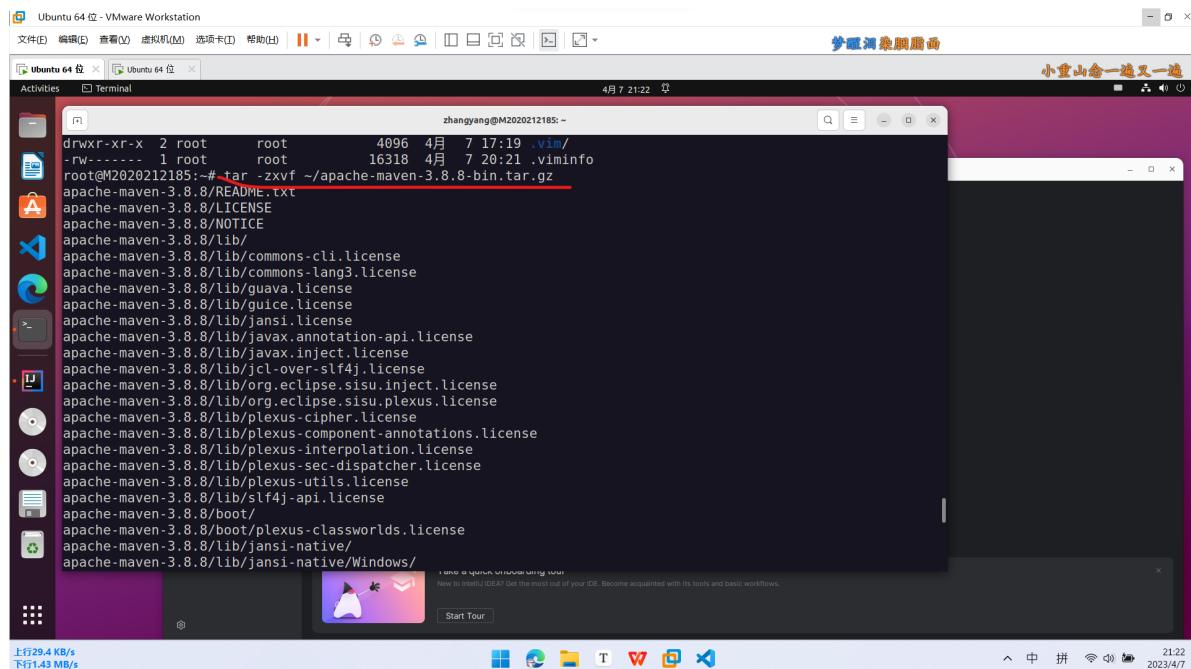
```
1 wget https://dlcdn.apache.org/maven/maven-3/3.8.7/binaries/apache-maven-3.8.7-bin.tar.gz
2 tar -zxvf ~/apache-maven-3.8.7-bin.tar.gz
```

这里会404，进官网发现没有3.8.7，改为3.8.8：

```
1 wget https://dlcdn.apache.org/maven/maven-3/3.8.8/binaries/apache-maven-3.8.8-bin.tar.gz
```

### 张场2020212185的截图：





修改 `/apache-maven-3.8.8/conf/settings.xml`，在其中添加如下配置：

```
1 <mirror>
2     <id>alimaven</id>
3     <mirrorOf>central</mirrorOf>
4     <name>aliyun maven</name>
5
6     <url>http://maven.aliyun.com/nexus/content/repositories/ce
7 ntral/</url>
8 </mirror>
9 <mirror>
10    <id>aliyunmaven</id>
11    <mirrorOf>*</mirrorOf>
12    <name>阿里云公共仓库</name>
13
14    <url>https://maven.aliyun.com/repository/public</url>
15 </mirror>
16 <localRepository>/opt/maven</localRepository>
```

张扬2020212185的截图：

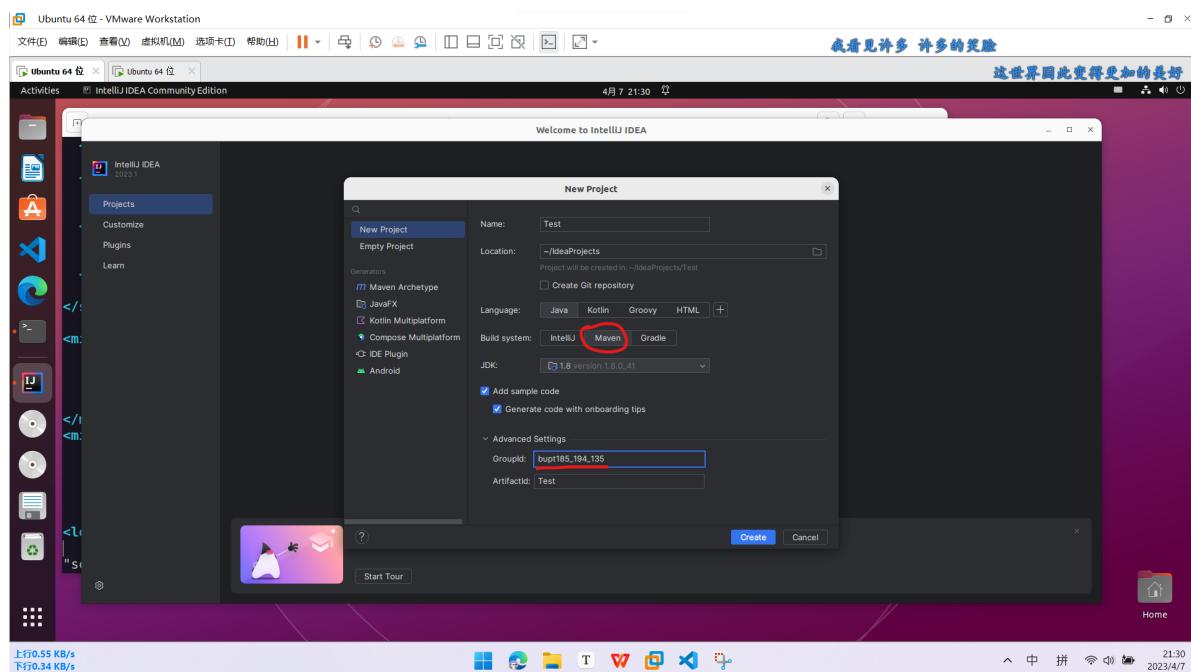
```
</profiles>
<!-- activeProfiles
   | List of profiles that are active for all builds.
   |
   <activeProfiles>
      <activeProfile>alwaysActiveProfile</activeProfile>
      <activeProfile>anotherAlwaysActiveProfile</activeProfile>
   </activeProfiles>
-->
</settings>
<mirror>
   <id>alimaven</id>
   <mirrorOf>central</mirrorOf>
   <name>aliyun maven</name>
   <url>http://maven.aliyun.com/nexus/content/repositories/central/</url>
</mirror>
<mirror>
   <id>aliyunmaven</id>
   <mirrorOf>*</mirrorOf>
   <name>阿里云公共仓库</name>
   <url>https://maven.aliyun.com/repository/public</url>
</mirror>
<localRepository>/opt/maven</localRepository>
```

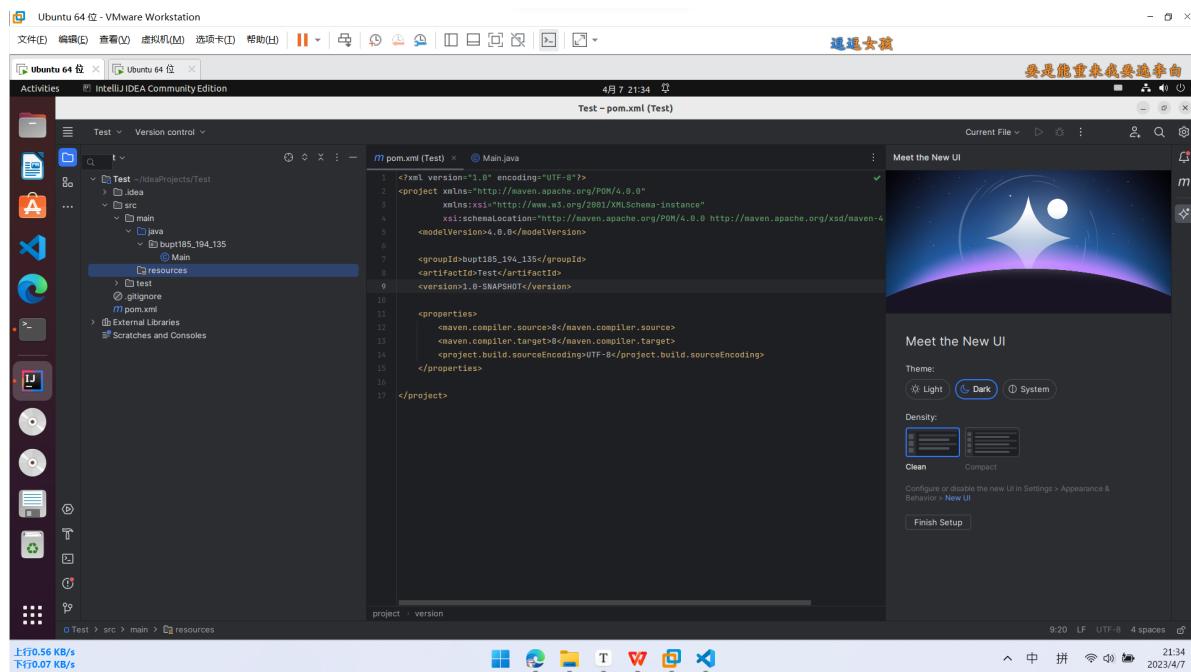
以上配置添加了一个镜像以提高下载速度，并指定了一个本地的maven仓库路径。

### 3-在IDEA中配置maven

首先创建一个java项目，并选择maven为构建工具：

张扬2020212185的截图：

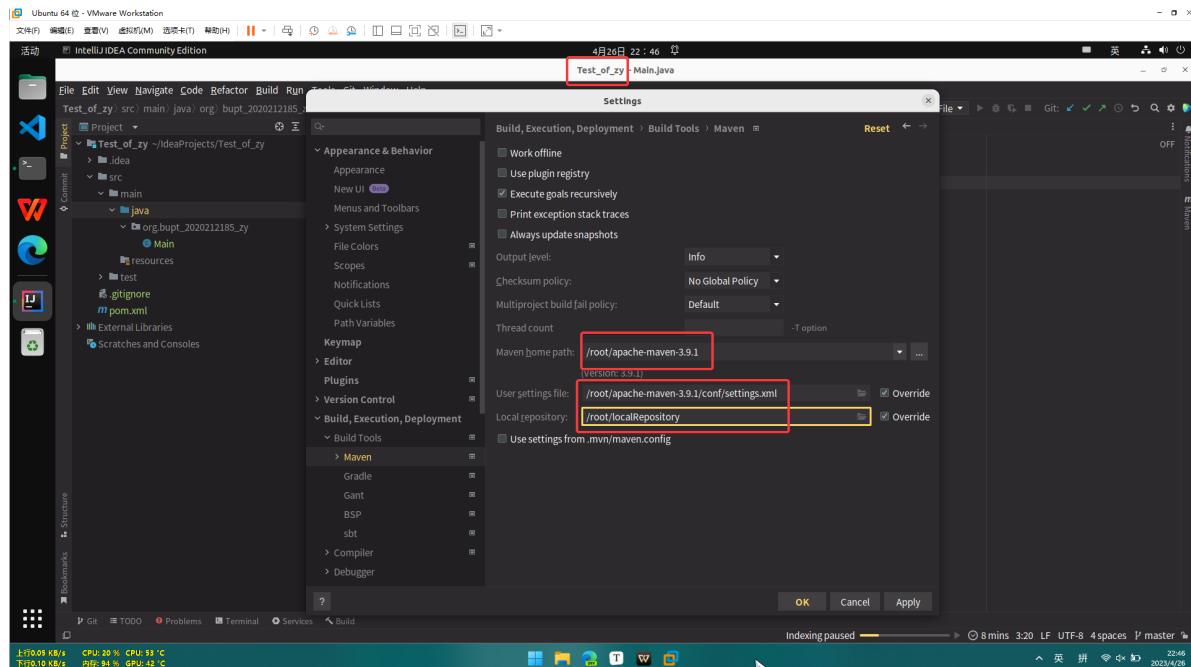




在IDEA中打开

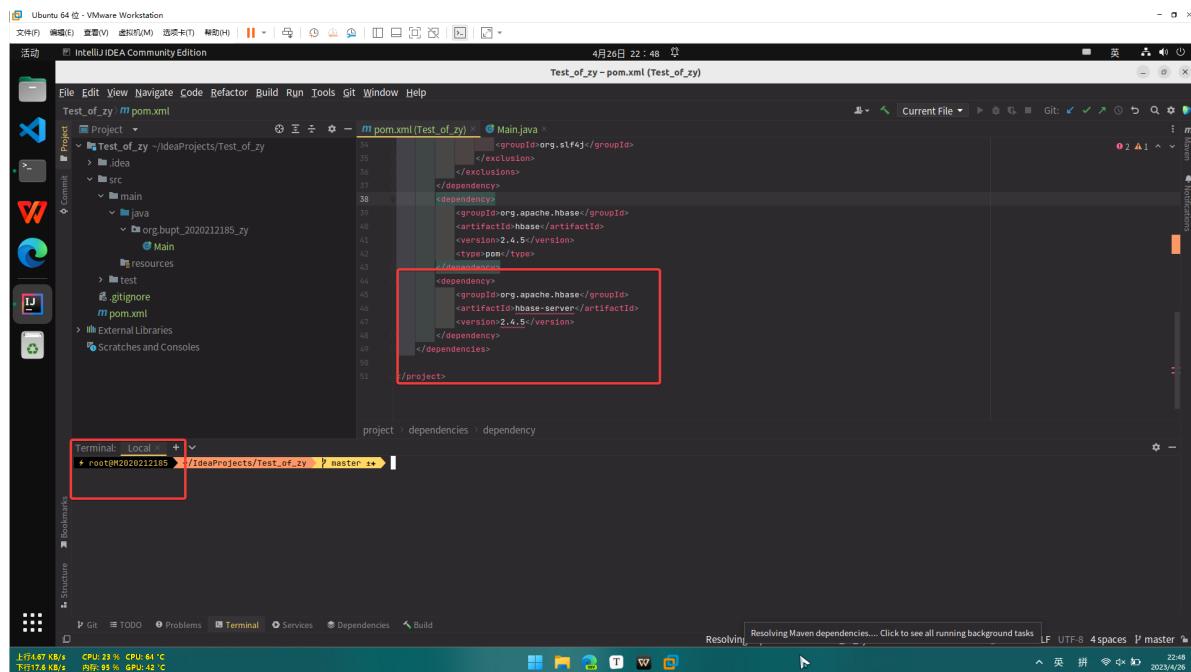
**File/Settings/Build,Execution,Deployment/BuildTools/Maven**，分别设置其中的 **Maven home path**、**User setting file** 和 **Local repository** **maven安装目录**、安装目录下的 **/apache-maven-3.8.7/conf/settings.xml** 和**自己的本地仓库路径**。

张扬2020212185的截图：



修改 **pom.xml** 文件如下：

问题：找不到 **hbase**



换了一个镜像 (?)

```
1 <mirror>
2   <id>alimaven</id>
3   <name>aliyun maven</name>
4
5     <url>http://maven.aliyun.com/nexus/content/groups/public/<url>
6       <mirrorOf>central</mirrorOf>
7 </mirror>
```

重新运行：

```
1 | mvn clean install
```

成功消除报错：

```

<dependency>
    <groupId>org.apache.hbase</groupId>
    <artifactId>hbase-client</artifactId>
    <version>2.4.5</version>
    <exclusions>
        <exclusion>
            <artifactId>slf4j-api</artifactId>
            <groupId>org.slf4j</groupId>
        </exclusion>
        <exclusion>
            <artifactId>slf4j-log4j12</artifactId>
            <groupId>org.slf4j</groupId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.apache.hbase</groupId>
    <artifactId>hbase</artifactId>
    <version>2.4.5</version>
    <type>pom</type>
</dependency>
<dependency>
    <groupId>org.apache.hbase</groupId>
    <artifactId>hbase-server</artifactId>
    <version>2.4.5</version>
</dependency>

```

标黄是版本的警告，不必在意。

## 4-配置log4j

在java项目的 resources 文件夹下添加 `log4j.properties` 文件：

内容如下：

张扬2020212185的截图：

```

log4j.rootLogger=debug, stdout, R
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
# Pattern to output the caller's file name and line number.
log4j.appender.stdout.layout.ConversionPattern=%5p [%t] (%F:%L) - %m%n
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=example.log
log4j.appender.R.MaxFileSize=100KB
# Keep one backup file
log4j.appender.R.MaxBackupIndex=5
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%p %c - %m%n

```

## 5-在java中调用HBase

实例代码Main.java如下：

张扬2020212185的截图

The screenshot shows the IntelliJ IDEA interface with the Main.java file open. The code implements a Main class with a static void main method that connects to a database and prints the connection URL. The project structure is visible on the left, and the terminal at the bottom shows Maven dependency download logs.

```
package org.bupt_2020212185.zy;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.ColumnFamilyDescriptorBuilder;
import org.apache.hadoop.hbase.client.ColumnDescriptor;
import org.apache.hadoop.hbase.client.TableDescriptorBuilder;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.ClientAdmin;
import org.apache.hadoop.hbase.client.ColumnFamilyDescriptorBuilder;
import org.apache.hadoop.hbase.client.ColumnDescriptor;
import org.apache.hadoop.hbase.client.TableDescriptorBuilder;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.commons.collections.CollectionUtils;
import org.apache.commons.lang.StringUtils;
import java.io.IOException;
import java.util.*;
public class Main {
    private static final Logger logger = LoggerFactory.getLogger(Main.class);
    new";
    public static void main(String[] args) {
        try (Connection connection = getConnection()) {
            Test_z_y connection = Test_z_y.createConnection();
        }
    }
}
```

Build Success

张扬2020212185的截图

The screenshot shows the IntelliJ IDEA interface during a build process. The terminal output at the bottom indicates a successful build of the Test\_of\_z\_y target, producing a JAR file named Test\_of\_z\_y-1.0-SNAPSHOT.jar. The build log also shows Maven dependency download logs.

```
[INFO] Building Jar /root/IdeaProjects/Test_of_z_y/target/Test_of_z_y-1.0-SNAPSHOT.jar
[INFO] --- install:3.1.0:install (default-install) @ test_of_z_y ...
[INFO] Installing /root/IdeaProjects/Test_of_z_y/pom.xml to /root/localRepository/org/bupt_2020212185.zy/Test_of_z_y/1.0-SNAPSHOT/Test_of_z_y-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 05:13 min
[INFO] Finished at: 2023-04-27T08:41:11+08:00
[INFO]
```

张扬2020212185的截图

log4j.properties	32
> test	33
< target	34
> classes	35
> generated-sources	36
> generated-test-sources	37
> maven-archiver	38
> maven-status	39
> test-classes	40
<b>Test_of_zy-1.0-SNAPSHOT.jar</b>	41
< .gitignore	42
<b>pom.xml</b>	43
	44
	45
	46