《云计算与云服务》

实验报告

实验 2 分布式存储基础实践

学期：2023-2024-2

姓名：胡肖安

班级：电信 2101

学号：0122120450310

一、 实验目的

（1） 掌握 Hadoop 的伪分布式安装方法；

（2） 理解 HDFS 在 Hadoop 体系结构中的角色；

（3） 熟练使用 HDFS 操作常用的 Shell 命令；

（4） 熟悉 HDFS 操作常用的 Java API。

二、 实验环境

- 操作系统：Linux (Ubuntu 20.04)
- Hadoop 版本：3.3.2
- JDK 版本：1.8 或者以上版本
- Java IDE：Eclipse

三、 实验内容

1、安装 Hadoop

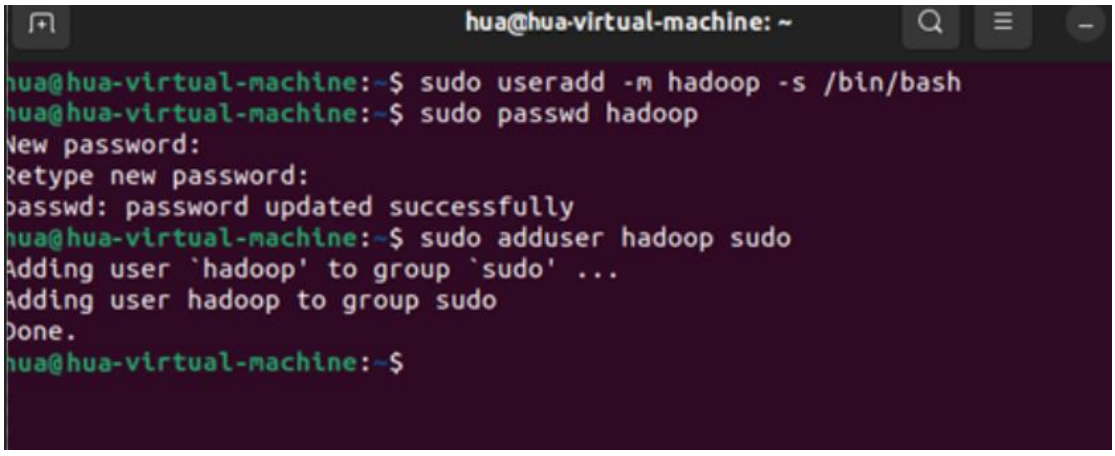
（1）创建 hadoop 用户（使用 /bin/bash 作为 Shell）、设置密码（建议简单）并为其添加管理员权限。具体命令如下：

sudo useradd -m hadoop -s /bin/bash

sudo passwd hadoop

sudo adduser hadoop sudo

完成上述步骤后，结果显示如下：



（2）切换到 hadoop 用户。

sudo su – Hadoop

```
hua@hua-virtual-machine:~$ sudo su - hadoop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

（3）安装 SSH server，并测试登陆。

sudo apt-get install openssh-server

ssh localhost

完成上述步骤后，结果如下：

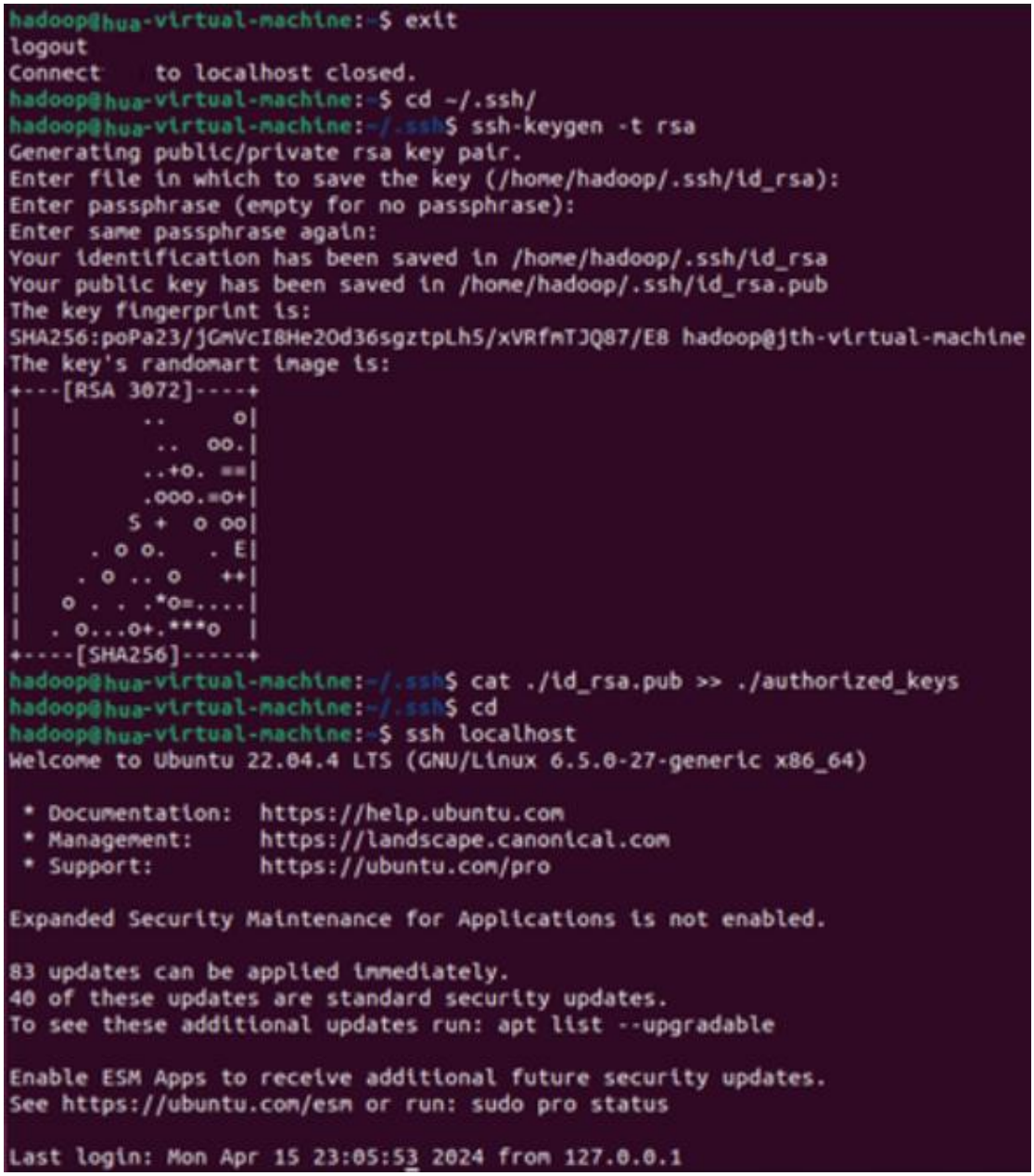
```
hadoop@hua-virtual-machine:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:nopO4KM1gRlxcgG9UYb1cYnTmqWI6Nkqm3BlXQh3N6U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
hadoop@localhost's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-27-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

83 updates can be applied immediately.
40 of these updates are standard security updates.
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


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.
```

在执行命令时，注意要按照指示输入 yes，再输入 hadoop 用户的密码。在执行结束后，即可通过 SSH 登陆到本机。由于 hadoop 需要通过 SSH 来控制集群，所以需要对 SSH 设置免密登陆，即允许名称节点可以无密码登录集群中的所有机器。

（4）设置 SSH 免密登录，利用 ssh-keygen 生成密钥并将秘钥加入到授权中。

```
exit                                    #退出刚才的 ssh localhost
cd ~/.ssh/                              #若没有该目录，请先执行一次 ssh localhost
ssh-keygen -t rsa                       #会有提示，都按回车就可以
cat ./id_rsa.pub >> ./authorized_keys   #加入授权
```

在执行 ssh-keygen -t rsa 命令时，每次按回车键即可。此时免密设置完成，使用 ssh localhost 进行验证。整个过程如下图所示：



（5）安装 Java 环境。Hadoop 3.3.2 需要 JDK 版本在 1.8 及以上。JDK 1.8 安装方式多种多样。这里采取手动安装，需要自行下载 JDK1.8 的安装包（见群文件）。将 JDK 安装包放在 Downloads 目录下，并且创建存放 JDK 文件的目录。如果在切换 Downloads 目录时出现找不到目录，可以重启系统，手动选择进入 hadoop 用户。

cd /usr/lib

sudo mkdir jvm　　#创建/usr/lib/jvm 目录用来存放 JDK 文件

cd ~/Downloads

sudo tar -zxvf ./jdk-8u162-linux-x64.tar.gz -C /usr/lib/jvm

#把 JDK 文件解压到/usr/lib/jvm 目录下

```
hadoop@hua-virtual-machine:~$ cd /usr/lib
hadoop@hua-virtual-machine:/usr/lib$ sudo mkdir jvm
[sudo] password for hadoop:
hadoop@hua-virtual-machine:/usr/lib$ cd ~/Downloads
hadoop@hua-virtual-machine:~/Downloads$ sudo tar -zxvf ./jdk-8u162-linux-x64.tar
.gz -C /usr/lib/jvm
jdk1.8.0_162/
jdk1.8.0_162/javafx-src.zip
jdk1.8.0_162/bin/
jdk1.8.0_162/bin/jmc
jdk1.8.0_162/bin/serialver
jdk1.8.0_162/bin/jmc.ini
jdk1.8.0_162/bin/jstack
jdk1.8.0_162/bin/rmiregistry
jdk1.8.0_162/bin/unpack200
jdk1.8.0_162/bin/jar
jdk1.8.0_162/bin/jps
jdk1.8.0_162/bin/wsimport
jdk1.8.0_162/bin/rmic
jdk1.8.0_162/bin/jdeps
jdk1.8.0_162/bin/jcontrol
```

（6）查看 JDK 是否解压到 jvm 文件夹中。

cd /usr/lib/jvm

ls

```
hadoop@hua-virtual-machine:~/Downloads$ cd /usr/lib/jvm
hadoop@hua-virtual-machine:/usr/lib/jvm$ ls
jdk1.8.0_162
```

（7）设置 java 环境变量使其生效，并查看版本号验证是否安装成功。

vim ~/.bashrc

上述命令使用 vim 编辑器打开了 hadoop 这个用户的环境变量配置文件.bashrc,请在这个文件的开头位置插入如下内容：

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
```

export PATH=${JAVA_HOME}/bin:$PATH

保存.bashrc 文件并退出 vim 编辑器。然后，继续执行如下命令让.bashrc 文件的配置立即生效：

source ~/.bashrc

这时，可以使用如下命令查看是否安装成功：

java -version

最终结果显示如下：

```
hadoop@hua-virtual-machine:/usr/lib/jvm$ vim ~/.bashrc
hadoop@hua-virtual-machine:/usr/lib/jvm$ source ~/.bashrc
hadoop@hua-virtual-machine:/usr/lib/jvm$ java -version
java version "1.8.0_162"
Java(TM) SE Runtime Environment (build 1.8.0_162-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)
```

至此，成功安装了 Java 环境。下面将进入 Hadoop 的安装。

（8）下载 hadoop-3.3.2 至 Downloads 目录（见群文件），并将 Hadoop 安装至/usr/local/中。

sudo tar -zxvf ~/Downloads/hadoop-3.3.2.tar.gz -C /usr/local #解压到/usr/local 中

cd /usr/local/

sudo mv ./hadoop-3.3.2/ ./hadoop　　#将文件夹名修改为 hadoop

sudo chown -R hadoop ./hadoop　　#修改文件权限

Hadoop 解压后即可使用。输入如下命令来检查 Hadoop 是否可用，成功则会显示 Hadoop 版本信息：

cd /usr/local/hadoop

./bin/hadoop version

结果如下图所示：

```
hadoop@hua-virtual-machine:/usr/lib/jvm$ cd /usr/local/
hadoop@hua-virtual-machine:/usr/local$ sudo mv ./hadoop-3.3.2/ ./hadoop
hadoop@hua-virtual-machine:/usr/local$ sudo chown -R hadoop ./hadoop
hadoop@hua-virtual-machine:/usr/local$ cd /usr/local/hadoop
hadoop@hua-virtual-machine:/usr/local/hadoop$ ./bin/hadoop version
Hadoop 3.3.2
Source code repository git@github.com:apache/hadoop.git -r 0bcb014209e219273cb6f
d4152df7df713cbac61
Compiled by chao on 2022-02-21T18:39Z
Compiled with protoc 3.7.1
From source with checksum 4b40fff8bb27201ba07b6fa5651217fb
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3
.3.2.jar
```

（9）Hadoop 伪分布式配置。对 core-site 和 hdfs-site 两个配置文件分别进行修改。

1）cd /usr/local/hadoop/etc/hadoop/

2）vim core-site.xml　　#使用 vim 打开配置文件 core-site.xml 并进行修改

将 core-site.xml 中的配置修改为如下内容：

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/hadoop/tmp</value>
        <description>Abase for other temporary directories.</description>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

3）vim hdfs-site.xml　　#使用 vim 打开配置文件 hdfs-site.xml 并进行修改

将 hdfs-site.xml 中的配置修改为如下内容：

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
```

<value>file:/usr/local/hadoop/tmp/dfs/data</value>

</property>

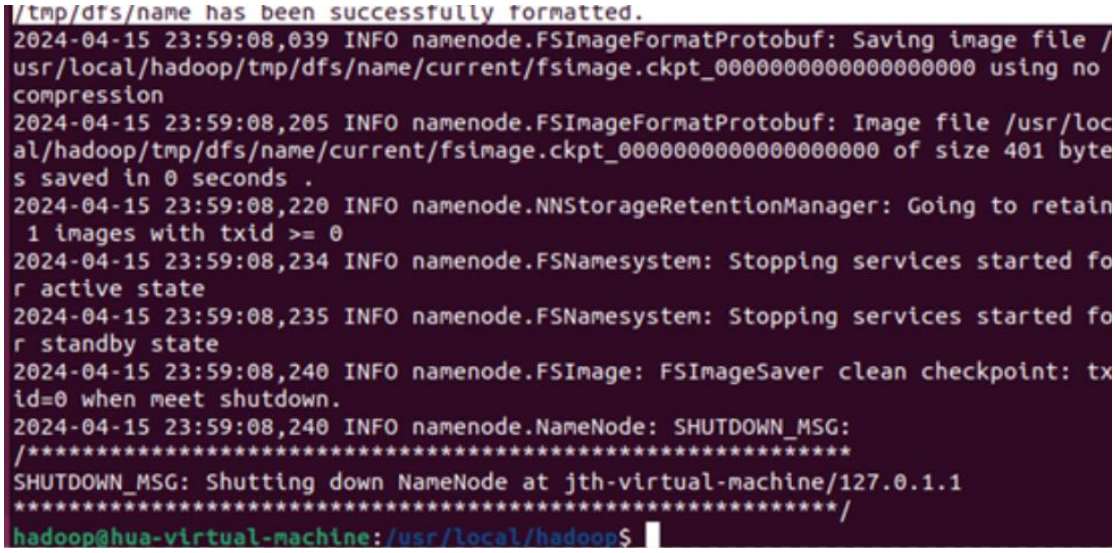
</configuration>

（10）对 NameNode 进行格式化。（在执行此步之前可以存快照）

cd /usr/local/hadoop

./bin/hdfs namenode -format

结果如下图所示（成功的话，会看到“successfully formatted”的提示）：



（11）开启 NameNode 和 DataNode 守护进程。在启动完成后使用 jps 命令来查看是否启动成功。若成功启动则会列出如下进程:NameNode, DataNode 和 SecondaryNameNode（如果 SecondaryNameNode 没有启动，请运行./sbin/stop-dfs.sh 关闭进程，然后再次尝试启动）。如果没有 NameNode 或 DataNode，那就是配置不成功，请仔细检查之前的步骤，或通过查看启动日志排查原因。

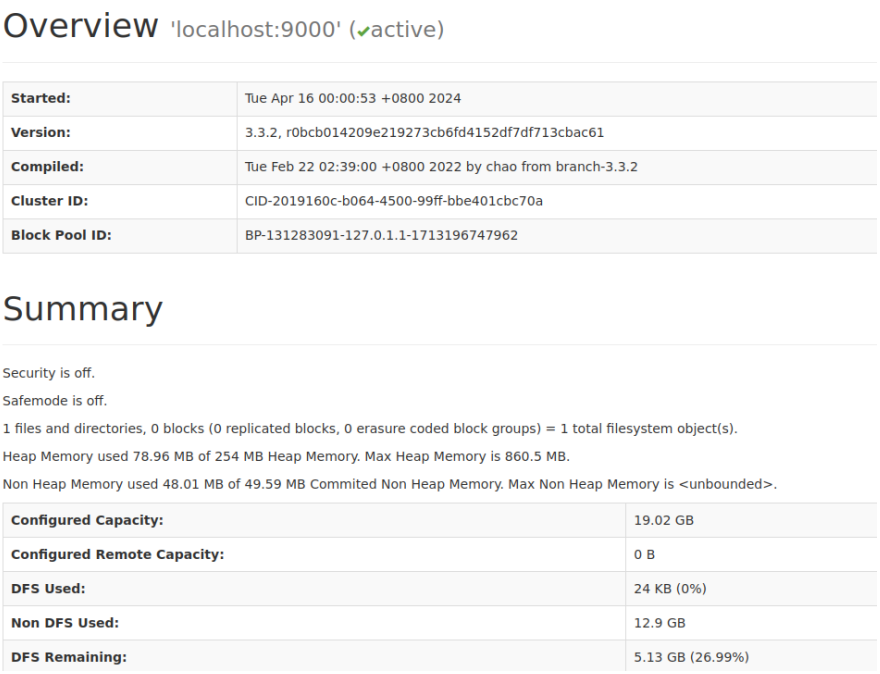
./sbin/start-dfs.sh

jps

结果如下图所示：

```
hadoop@hua-virtual-machine:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [jth-virtual-machine]
jth-virtual-machine: Warning: Permanently added 'jth-virtual-machine' (ED
to the list of known hosts.
hadoop@hua-virtual-machine:/usr/local/hadoop$ jps
7008 Jps
6578 NameNode
6697 DataNode
6877 SecondaryNameNode
```

（12）安装成功后，在 linux 浏览器中访问 http://localhost:9870 来查看 NameNode 和 DataNode 信息，还可以在线查看 HDFS 中的文件。如下图所示：



Overview 'localhost:9000' (active)

Started:	Tue Apr 16 00:00:53 +0800 2024
Version:	3.3.2, r0bcb014209e219273cb6fd4152df7df713cbac61
Compiled:	Tue Feb 22 02:39:00 +0800 2022 by chao from branch-3.3.2
Cluster ID:	CID-2019160c-b064-4500-99ff-bbe401cbc70a
Block Pool ID:	BP-131283091-127.0.1.1-1713196747962

Summary

Security is off.

Safemode is off.

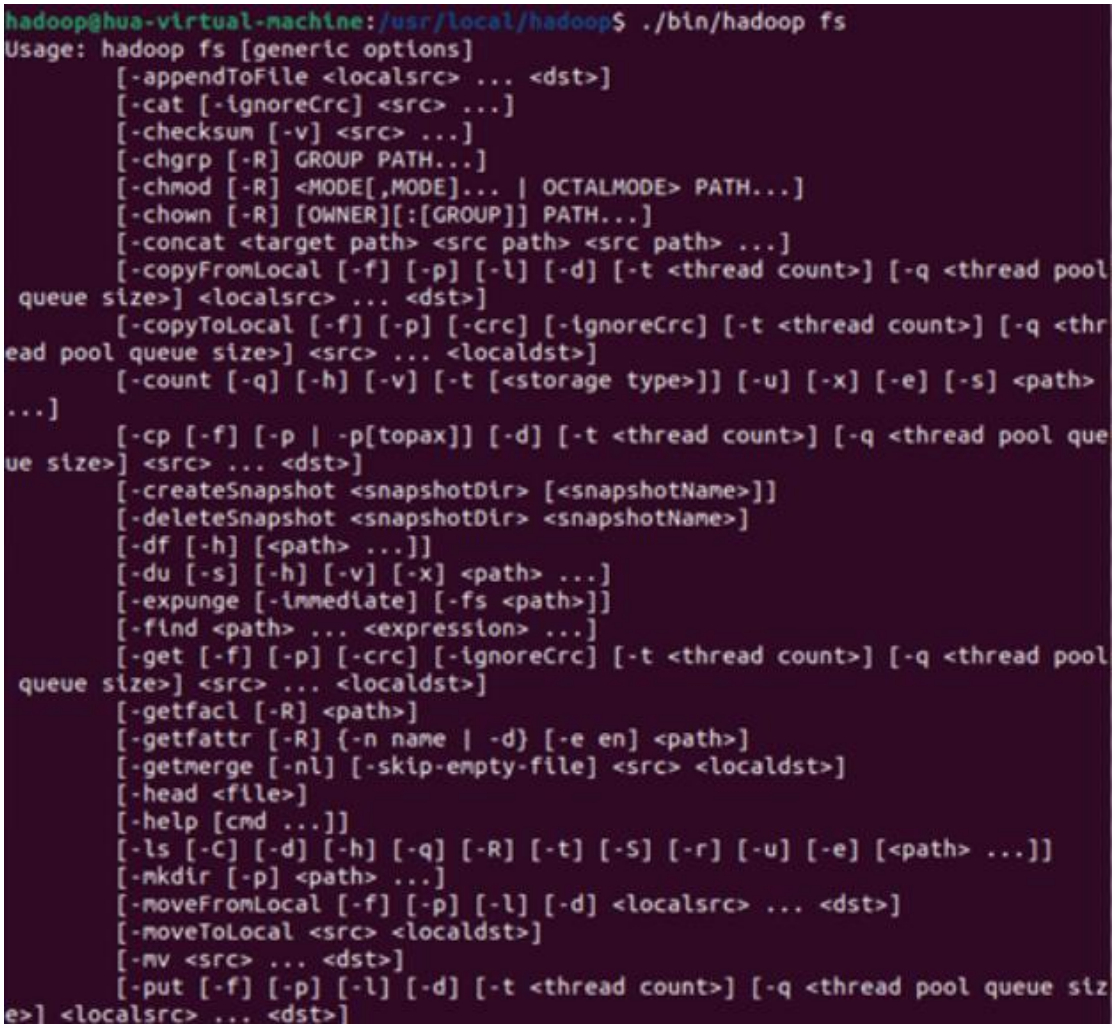
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 78.96 MB of 254 MB Heap Memory. Max Heap Memory is 860.5 MB.

Non Heap Memory used 48.01 MB of 49.59 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	19.02 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	12.9 GB
DFS Remaining:	5.13 GB (26.99%)

查看 fs 总共支持了哪些命令



查看 put 命令如何使用，可以输入如下命令

```
hadoop@hua-virtual-machine:/usr/local/hadoop$ ./bin/hadoop fs -help put\
>
-put [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <loca
lsrc> ... <dst> :
  Copy files from the local file system into fs. Copying fails if the file alrea
dy
  exists, unless the -f flag is given.
  Flags:


  -p                            Preserves timestamps, ownership and the mode.

  -f                            Overwrites the destination if it already exists.

  -t <thread count>             Number of threads to be used, default is 1.

  -q <thread pool queue size>   Thread pool queue size to be used, default is

                                1024.

  -l                            Allow DataNode to lazily persist the file to disk
.
                                Forces replication factor of 1. This flag will

                                result in reduced durability. Use with care.

  -d                            Skip creation of temporary file(<dst>._COPYING_).
```

2. 利用 Hadoop 提供的 shell 命令按照顺序依次完成以下任务：

（1）使用 hadoop 用户登录 Linux 系统，启动 Hadoop（Hadoop 的安装目录为/usr/local/hadoop），并使用 Web 界面查看 HDFS 信息；

```
hadoop@hua VirtualBox:/usr/local/hadoop$ cd /usr/local/hadoop
hadoop@hua VirtualBox:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 2528.  Stop it first and ensure /tmp/h
adoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 2672.  Stop it first and ensure /tmp/h
adoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [lrh-VirtualBox]
lrh-VirtualBox: secondarynamenode is running as process 2886.  Stop it first and
 ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
hadoop@hua VirtualBox:/usr/local/hadoop$
```

（2）为 hadoop 用户在 HDFS 中创建用户目录/user/hadoop；

（3）接着在 HDFS 的目录/user/hadoop 下，创建以自己名字和班级命名的文件夹，并查看文件列表

（4）将 Linux 系统本地的~/.bashrc (即/home/hadoop/.bashrc)文件上传到 HDFS 中由自己姓名和班级命名的文件夹，并查看是否上传成功；

（5）将 HDFS 文件夹复制到 Linux 系统本地文件系统的/usr/local/hadoop 目录下，并在 Linux 系统本地文件系统查看是否复制成功；

（6）在 HDFS 中删除文件夹中的所有文件，并显示是否删除成功；

（7）使用 vim（或者 gedit 或者 nano）编辑器，在本地 Linux 文件系统的

/home/hadoop/ 目录下创建一个文件 myLocalFile.txt，文件里面输入以下内容：

Hadoop and HDFS

dianxin2101ghm

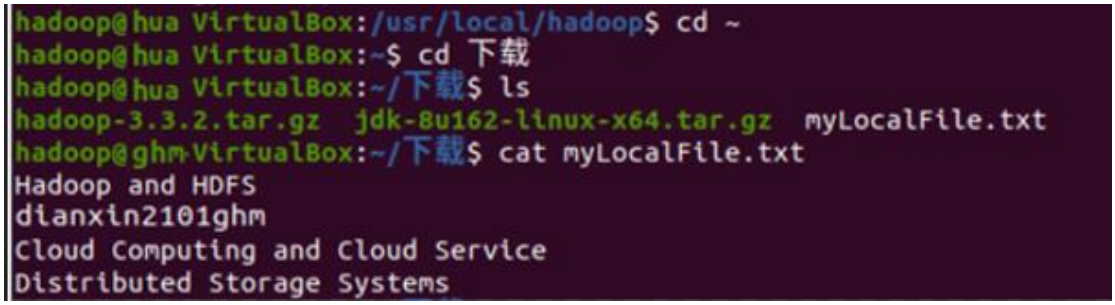
Cloud Computing and Cloud Service

Distributed Storage System

（7）将本地文件系统的“/home/hadoop/myLocalFile.txt”上传到 HDFS 中的“/user/hadoop/gaohaoming2101”目录下；

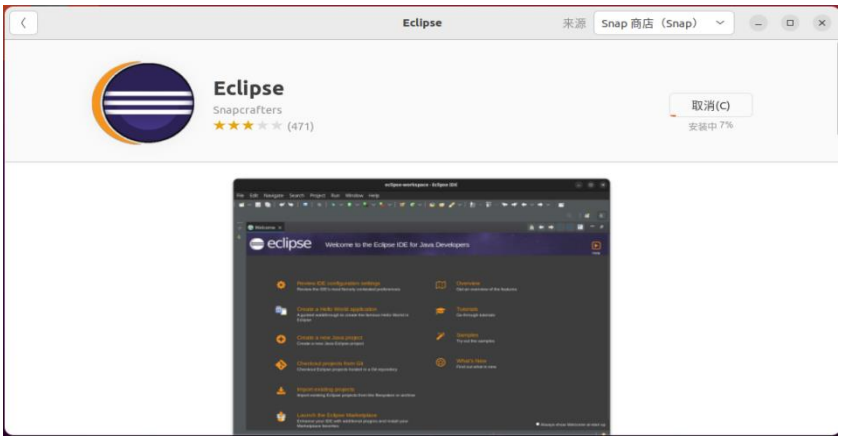
（8）查看文件 myLocalFile.txt 是否上传成功，如果上传成功，查看 HDFS 中的 myLocalFile.txt 这个文件的内容；

（9）将 HDFS 中的 myLocalFile.txt 文件下载到本地文件系统中的“/home/hadoop/下载”目录下，并在本地文件系统查看下载下来的文件

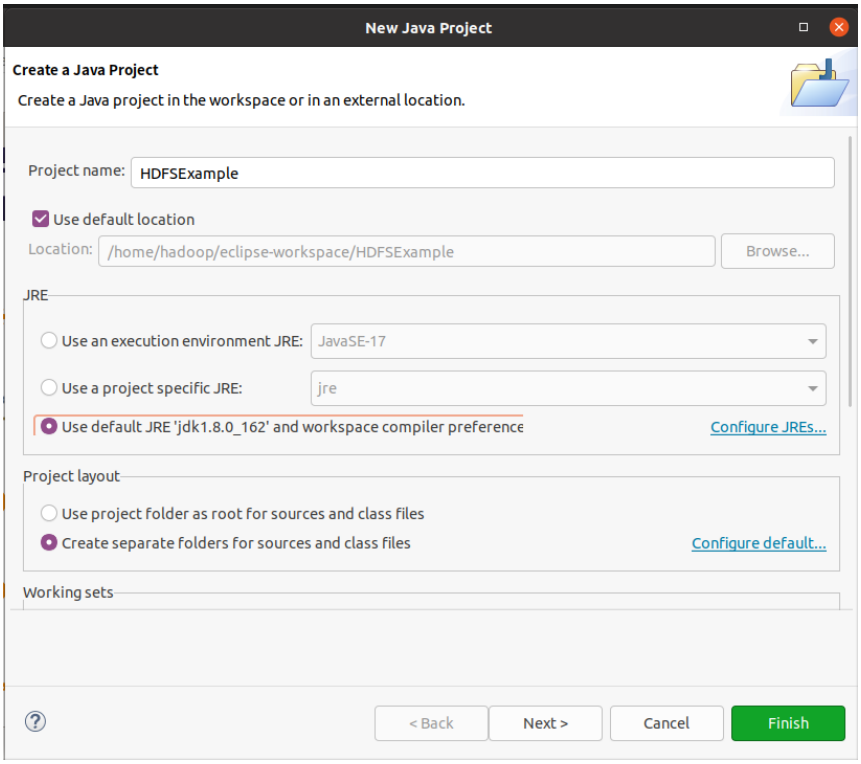


利用 Java API 与 HDFS 进行交互

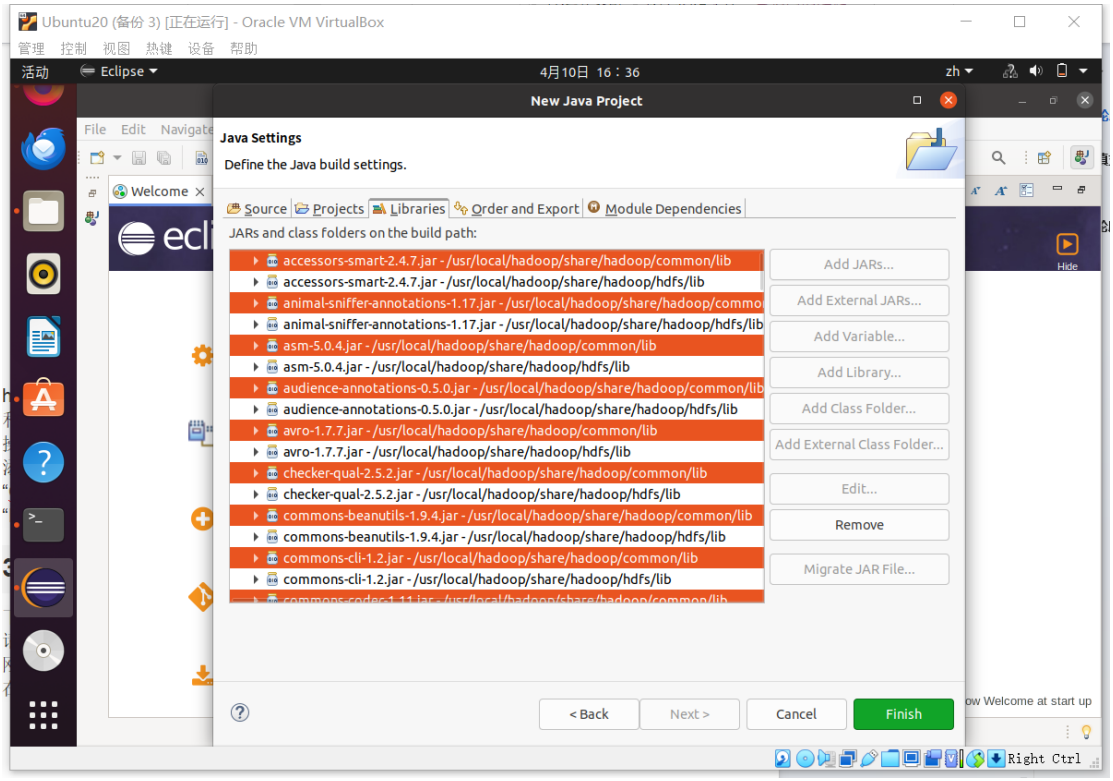
（1）在 Linux (Ubuntu)中安装 Eclipse；



（2）在 Eclipse 中创建项目；



（3）为项目添加需要用到的 JAR 包；



（4）用 Java 编写程序检测 HDFS 中是否存在文件 myLocalFile.txt；

（5）编译运行程序并显示检测结果。

四、 出现的问题与解决方案

1.在切换用户过程中，出现用户不存在或权限异常，阻碍了系统配置和运行。 解决方案：重新启动后，选择正确的用户登录，确保用户存在并避免权限问题。

2.安装 SSH server 时遇到依赖或配置问题，导致无法完成实验环境的搭建。 解决方案：放弃当前环境，重新建立实验环境并正确安装 SSH server。

3.编辑器无法正常使用，影响了对系统配置文件的编辑和管理。 解决方案：重新安装编辑器软件，确保能够顺利编辑配置文件。

4.实验中网络连接出现异常，导致无法访问所需资源。 解决方案：检查网络设置并重新连接，确保能够正常访问所需资源。

5.安装软件过程中出现版本不兼容或缺少依赖的问题，导致无法顺利进行后续操作。 解决方案：升级软件版本或安装所需的依赖，确保软件能够正常运行。

6.系统配置文件损坏或错误设置，导致系统启动异常或功能受限。 解决方案：恢复备份的配置文件或手动修改配置，确保系统配置正确。

7.实验环境中硬件故障或虚拟机配置不当，导致实验无法进行或数据丢失。解决方案：修复硬件故障或重新配置虚拟机，确保实验环境稳定运行。

五、实验总结

我认为：这次实验重点学习怎么安装 Hadoop 的伪分布式模式，然后搞明白 HDFS 在 Hadoop 里面的作用，还要学会用 Shell 命令操作 HDFS，最后也要熟悉 Java API 怎么用。我们在 Ubuntu 20.04 系统上搞了这个实验，用的是 Hadoop 3.3.2 和 JDK 1.8 或更高的版本。 开始的时候，我们先装了 JDK，然后下了 Hadoop 的东西，解压到对的地方。接着，我们创建了一个 Hadoop 的配置文件，叫做 hadoop-env.sh，然后设定了 JAVA_HOME 到我们装 JDK 的地方。还有，我们编辑了两个配置文件，core-site.xml 和 hdfs-site.xml，设置了 HDFS 的存储路径和其他设定。 通过这次实验，我们更加深入地了解了 Hadoop 和 HDFS 的基本原理，也学会了怎么装配置 Hadoop 环境，以及怎么操作 HDFS 和用 Java API 编程。

在实验过程中，我遇到了一些挑战，例如误卸载了重要软件导致系统受限、切换用户时权限问题、安装 SSH server 遇到依赖问题等实验中还涉及到网络连接、软件版本兼容性、配置文件管理等相关问题。通过自己的努力和查阅资料，我成功地解决了这些问题，也更加熟悉了软件配置和故障排除的方法。

总体来说，这次实验让我不仅加深了对 Hadoop 和云计算的理解，还提升了我的实践能力和问题解决能力。收获还是比较大的