# 技术组件选型

表格

描述已自动生成

表格

描述已自动生成

之前按照上面的安装了一部分集群环境，刚参考尚硅谷数仓6.0教程，有新的，用这个新的吧，毕竟新的肯定有很多好的地方。

下面hadoop的安装是3.1.3的版本，安装完成后我就换成3.3.4了。

## Hadoop安装

Hadoop3.1.3 下载：<https://hadoop.apache.org/release/3.1.3.html>

## Zookeeper

下载：<https://archive.apache.org/dist/zookeeper/zookeeper-3.4.10/>

## Mysql

下载：<https://downloads.mysql.com/archives/community/>

## Hive

下载：<https://dlcdn.apache.org/hive/hive-3.1.2/>

## Flume

下载：<https://archive.apache.org/dist/flume/1.9.0/>

## Kafka

下载：<https://archive.apache.org/dist/kafka/2.4.1/kafka_2.12-2.4.1.tgz>

Kafka Eagle： <https://codeload.github.com/azkaban/azkaban/tar.gz/refs/tags/3.84.4>

## Azkaban

下载：<https://github.com/azkaban/azkaban/archive/refs/tags/3.84.4.tar.gz>

## Spark

下载：<https://archive.apache.org/dist/spark/spark-3.0.0/>

## Hbase

下载：<https://archive.apache.org/dist/hbase/2.0.5/>

## Phoenix

下载：<http://archive.apache.org/dist/phoenix/apache-phoenix-5.0.0-HBase-2.0/bin/>

1. Sqoop

下载：<https://archive.apache.org/dist/sqoop/1.4.6/>

## Presto

下载：<https://repo1.maven.org/maven2/com/facebook/presto/presto-server/0.189/presto-server-0.189.tar.gz>

## Kylin

下载：<https://archive.apache.org/dist/kylin/apache-kylin-3.0.1/>

## Atlas

下载：<https://atlas.apache.org/#/Downloads>

## Range

下载：<https://www.apache.org/dyn/closer.lua/ranger/2.0.0/apache-ranger-2.0.0.tar.gz>

# 虚拟机创建

本地目录：C:\Users\jmd\Documents\软件资料

基础环境：Windows11，VMware Workstation Pro17，CentOS-7.5-x86\_64-DVD-1804.iso

创建虚拟机，典型：



稍后安装操作系统：

电脑萤幕的截图

描述已自动生成

电脑萤幕的截图

描述已自动生成

磁盘大小设置为100G，反正用不到，不用担心不够用，如果设置小了，再调整磁盘大小就很难了。

电脑萤幕的截图

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

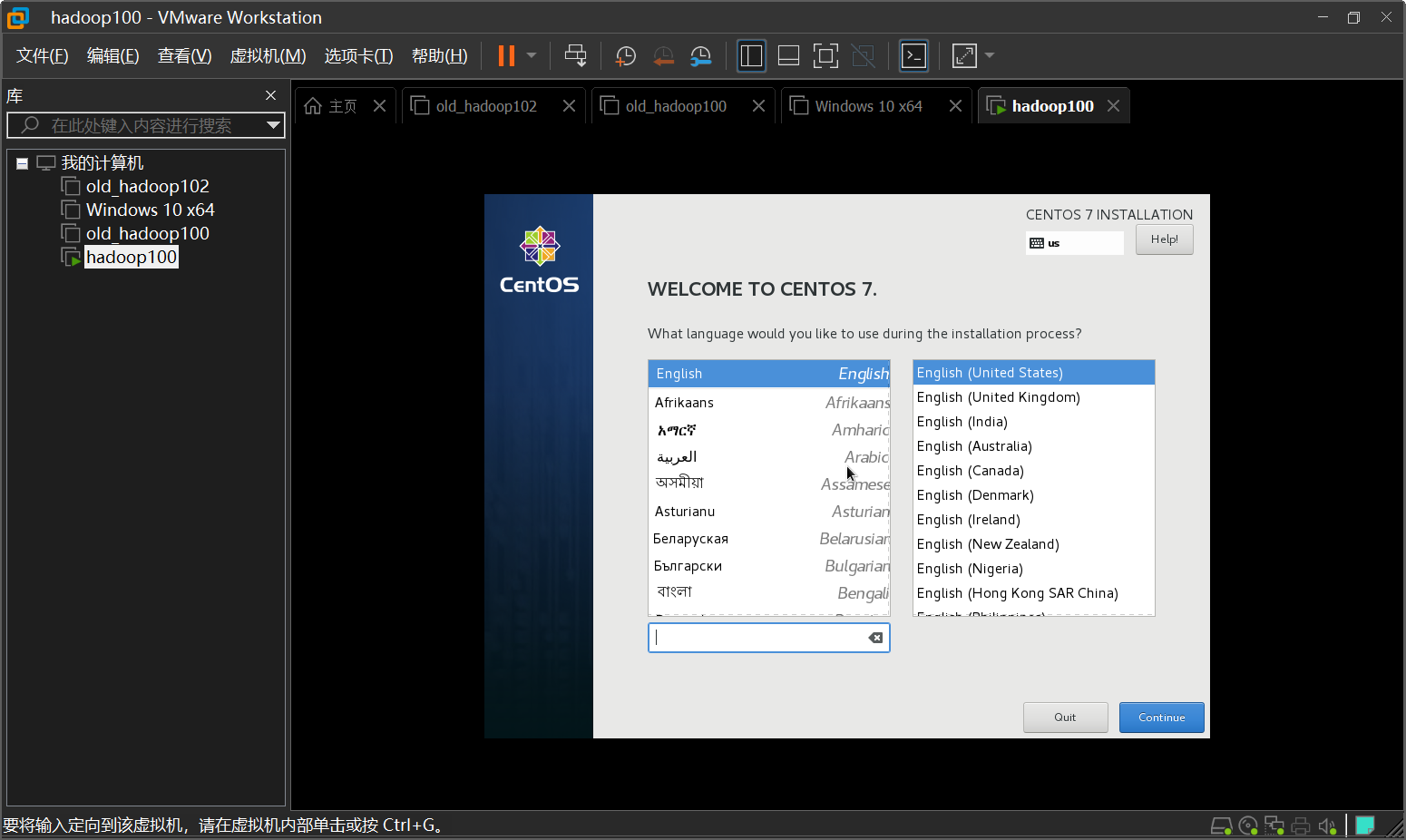
重命名为hadoop100，设置内存为6G，核心数量为4

设置系统文件路径：

图形用户界面, 应用程序

描述已自动生成

开启虚拟机：root用户密码888666，hadoop用户密码hadoop



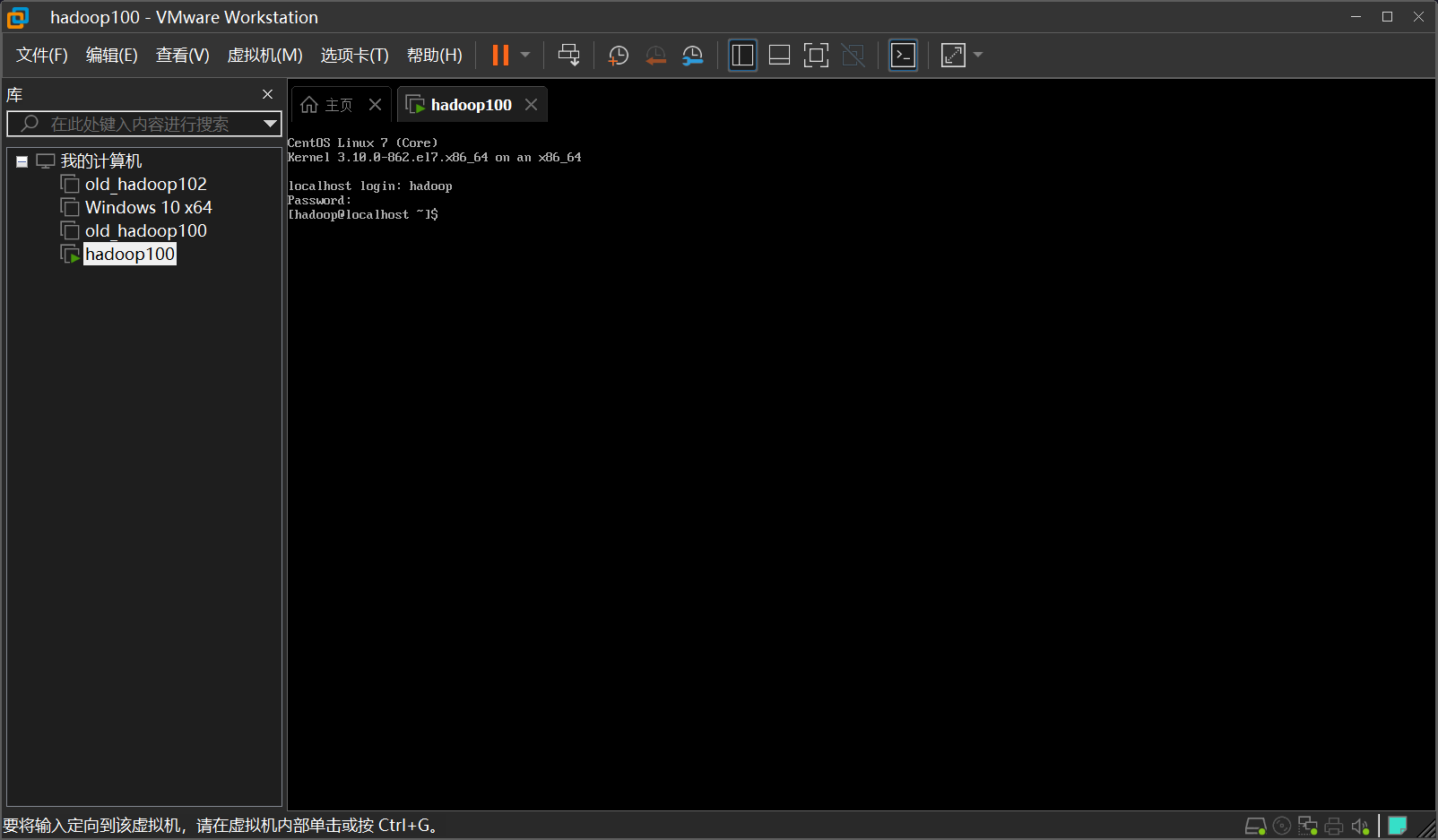
设置root用户和普通用户hadoop，分别设置密码为：888666，hadoop



2分钟后，显示安装完成后点击重启！

可以在安装环境时候分配更多的cpu和内存这样更快。

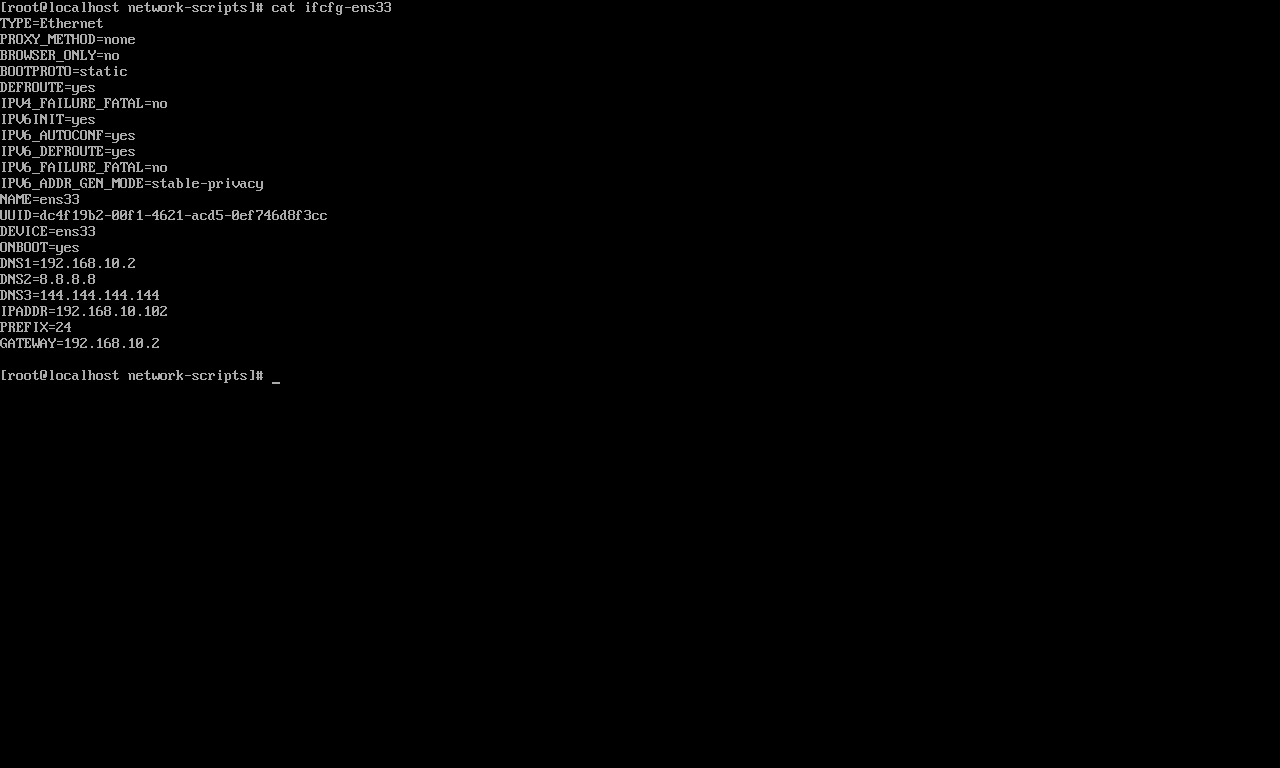
重启之后输入hadoop账号，密码，就可以进入系统了。



ping baidu.com，发现网络不通，设置下网卡配置文件

vi /etc/sysconfig/network-scripts/ifcfg-ens33

(IP地址配置应配置为192.128.10.100)



重启下网络或者直接reboot

发现直接使用xshell或者electerm这种远程终端工具可以通过ssh连接到虚拟机：

（有时需要开启虚拟机中的ssh服务）

1. 因为我安装的时候使用的是最小安装，所以需要安装一些工具软件：

安装 epel-release

yum install -y epel-release

注： Extra Packages for Enterprise Linux 是为“红帽系”的操作系统提供额外的软件包，

适用于 RHEL、 CentOS 和 Scientific Linux。相当于是一个软件仓库， 大多数 rpm 包在官方repository 中是找不到的）

文本

描述已自动生成

1. 如果 Linux 安装的是最小系统版，还需要安装如下工具；

如果安装的是 Linux  
桌面标准版，不需要执行如下操作  
net-tool：工具包集合，包含 ifconfig 等命令  
[root@hadoop100 ~]# yum install -y net-tools

图形用户界面, 文本

描述已自动生成  
vim：编辑器  
[root@hadoop100 ~]# yum install -y vim

文本

描述已自动生成

1. 关闭防火墙，关闭防火墙开机自启  
   [root@hadoop100 ~]# systemctl stop firewalld  
   [root@hadoop100 ~]# systemctl disable firewalld.service  
   注意：在企业开发时，通常单个服务器的防火墙时关闭的。公司整体对外会设置非常安全的防火墙。

文本

描述已自动生成

配置 hadoop用户具有 root 权限， 方便后期加 sudo 执行 root 权限的命令:

修改/etc/sudoers 文件，在%wheel 这行下面添加一行，如下所示：

|  |
| --- |
| sudo vim /etc/sudoers  ## Allow root to run any commands anywhere  root ALL=(ALL) ALL  ## Allows people in group wheel to run all commands  %wheel ALL=(ALL) ALL  hadoop ALL=(ALL) NOPASSWD:ALL  wq!保存退出 |

注意： hadoop这一行不要直接放到 root 行下面，因为所有用户都属于 wheel 组，你先  
配置了 hadoop具有免密功能，但是程序执行到%wheel 行时， 该功能又被覆盖回需要  
密码。所以 hadoop要放到%wheel 这行下面。

如果没配置这个，在hadoop这种普通用户空间的shell里执行sudo命令时就会报这样的错误：

文本

描述已自动生成

# Linux基础环境配置

1. 在/opt 目录下创建文件夹，并修改所属主和所属组

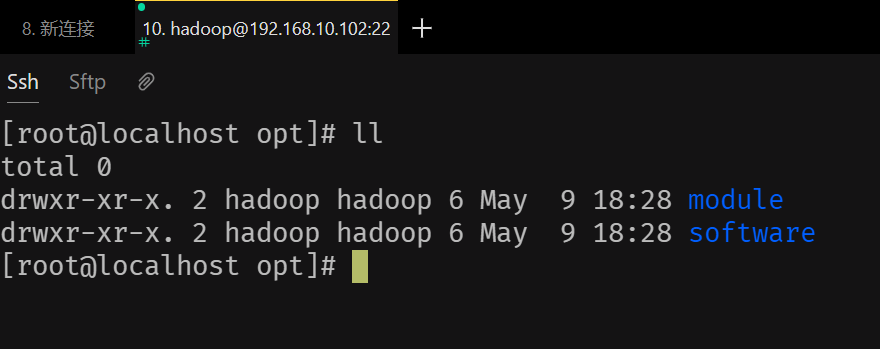
|  |
| --- |
| mkdir /opt/module && mkdir /opt/software |

1. 修改 module、 software 文件夹的所有者和所属组均为 hadoop 用户

|  |
| --- |
| chown hadoop:hadoop /opt/module && chown hadoop: hadoop /opt/softer |

1. 查看 module、 software 文件夹的所有者和所属组

|  |
| --- |
| cd /opt/  ll |



卸载虚拟机自带的 JDK

注意：如果你的虚拟机是最小化安装不需要执行这一步。

|  |
| --- |
| rpm -qa | grep -i java | xargs -n1 rpm -e –nodeps |

## windows主机上修改路由配置文件

C:\Windows\System32\drivers\etc\hosts

|  |
| --- |
| 192.168.10.100 hadoop100  192.168.10.102 hadoop102  192.168.10.103 hadoop103  192.168.10.104 hadoop104  192.168.10.105 hadoop105  192.168.10.106 hadoop106 |

## 将后续需要的安装包放到hadoop100上



修改主机名：

vim /etc/hostname

hadoop100

## 安装JDK8和JDK11（某些组件可能和JDK11不兼容）

将JDK8和JDK11的压缩包，解压到/opt/module目录

图形用户界面, 文本

描述已自动生成

并重命名

在vim /etc/profile.d/my\_env.sh 中设置Java路径

JAVA\_HOME，JAVA\_PATH，PATH

## 安装scala2.12.19

恰好看到2.12版本最后一个版本，就安装这个吧！

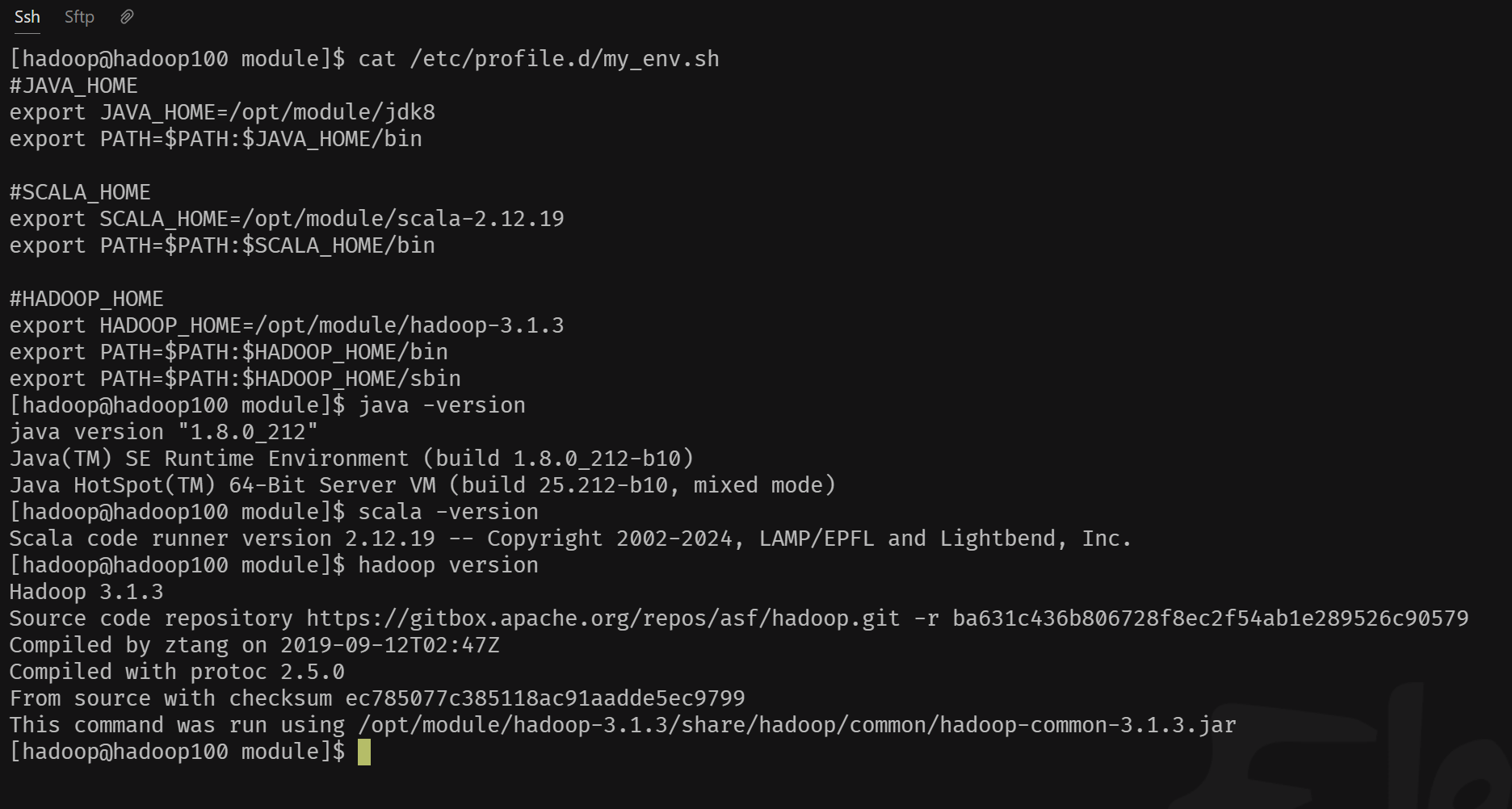
下载scala-2.12.19.tgz，解压缩到/opt/module目录下

配置SCALA\_HOME和PATH

## 安装Hadoop-3.1.3

将hadoop-3.1.3的压缩包解压到module中

按照下图配置HADOOP的环境变量



关闭该虚拟机hadoop100

vmware中操作克隆虚拟机命名为hadoop102, hadoop103, hadoop104

然后在hadoop103和104中修改ip和hostname，一个个改

vim /etc/sysconfig/network-scripts/ifcfg-ens33

vim /etc/hostname

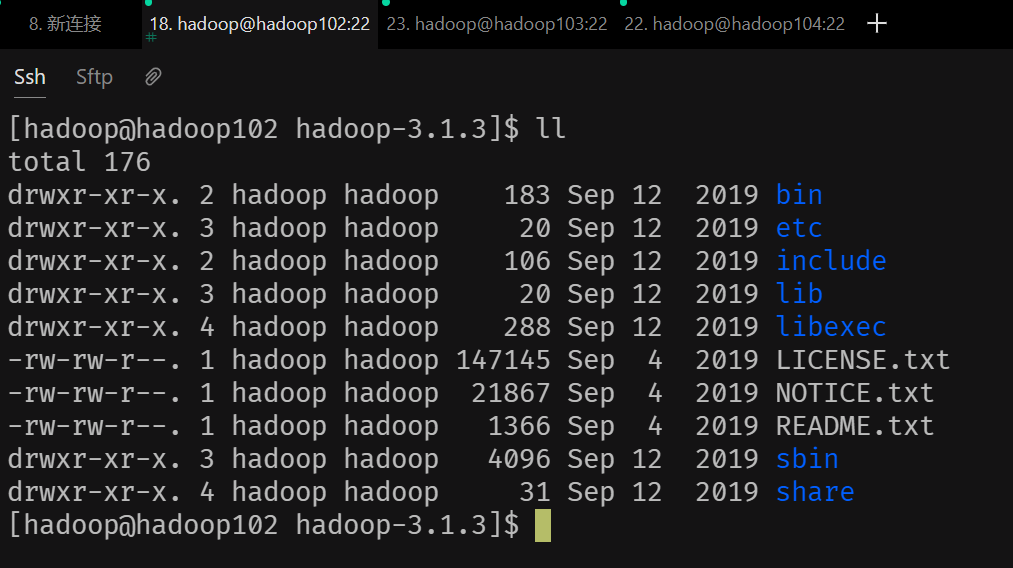
现在这三台虚拟机只有hostname和ip地址不同，都已经配置好了。

在vmware中分配给这三台机器的资源可以根据自己电脑调整下。

## 配置ssh远程免密登录

# Hadoop集群搭建和使用

上面已经在hadoop100中安装了Hadoop包，设置了路径。



（1） bin 目录：存放对 Hadoop 相关服务（hdfs， yarn， mapred）进行操作的脚本

（2） etc 目录： Hadoop 的配置文件目录，存放 Hadoop 的配置文件

（3） lib 目录：存放 Hadoop 的本地库（对数据进行压缩解压缩功能）

（4） sbin 目录：存放启动或停止 Hadoop 相关服务的脚本

（5） share 目录：存放 Hadoop 的依赖 jar 包、 文档、 和官方案例

Hadoop 运行模式

1） Hadoop 官方网站： http://hadoop.apache.org/

2） Hadoop 运行模式包括： 本地模式、 伪分布式模式以及完全分布式模式。

本地模式：单机运行，只是用来演示一下官方案例。 生产环境不用。

伪分布式模式： 也是单机运行，但是具备 Hadoop 集群的所有功能， 一台服务器模拟一个分布式的环境。 个别缺钱的公司用来测试，生产环境不用。

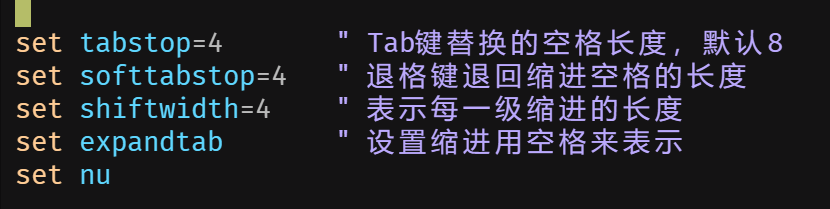
➢ 完全分布式模式： 多台服务器组成分布式环境。 生产环境使用。

1. 本地模式运行
2. 在 hadoop-3.1.3 文件下面创建一个 wcinput 文件夹
3. 在 wcinput 文件下创建一个 word.txt 文件
4. 编辑 word.txt 文件
5. 回到 Hadoop 目录/opt/module/hadoop-3.1.3
6. 执行示例程序

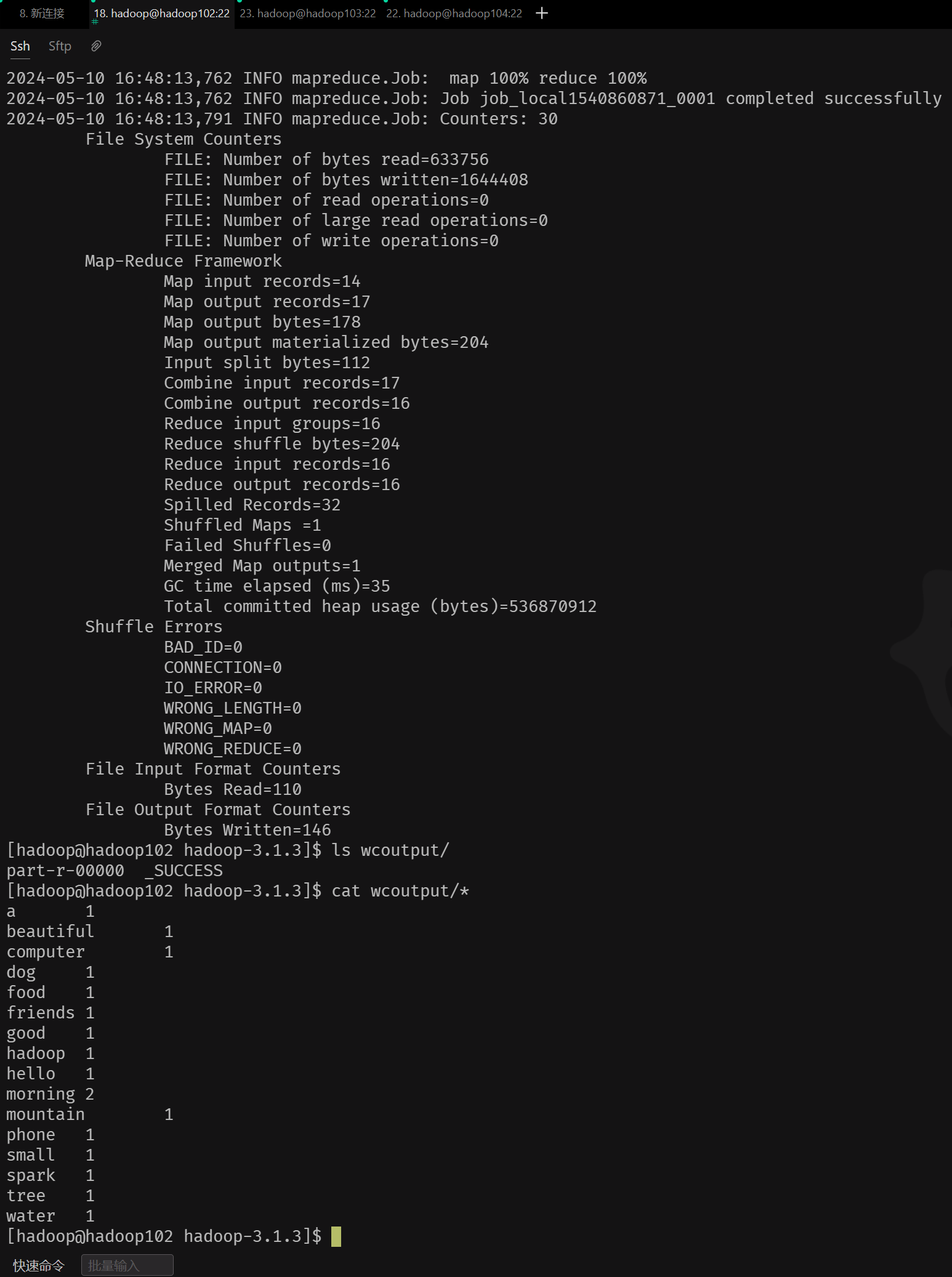
|  |
| --- |
| hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar wordcount wcinput wcoutput |

配置下vim

vi /etc/vim/vimrc



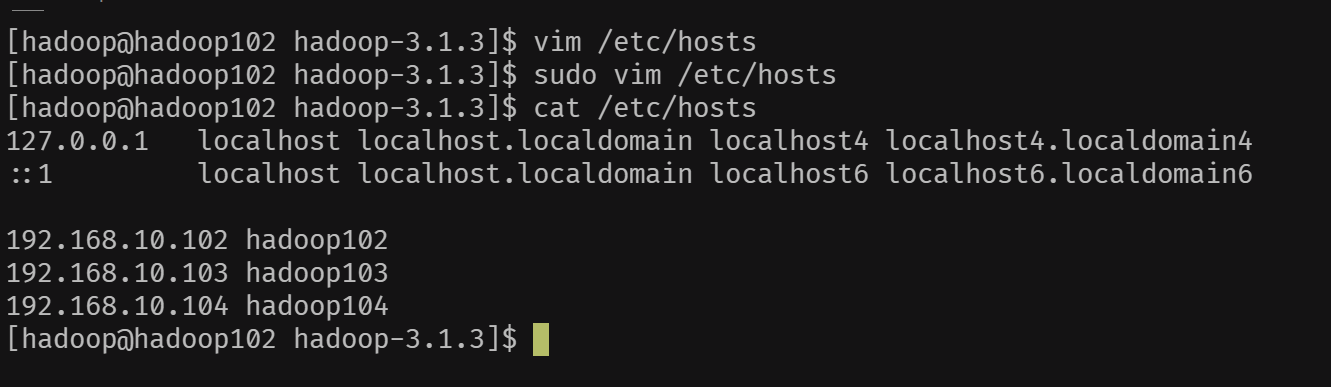
1. 查看结果



可以看到每个单词出现次数的统计结果

1. 完全分布式运行模式（开发重点）
2. 准备 3 台客户机（关闭防火墙、静态 IP、主机名称）

需要配置各个机器的ip到名字的映射，就可以用名字代替ip了。



2）安装 JDK

3）配置环境变量

4）安装 Hadoop

5）配置环境变量

6）配置集群

7）单点启动

8）配置 **ssh**

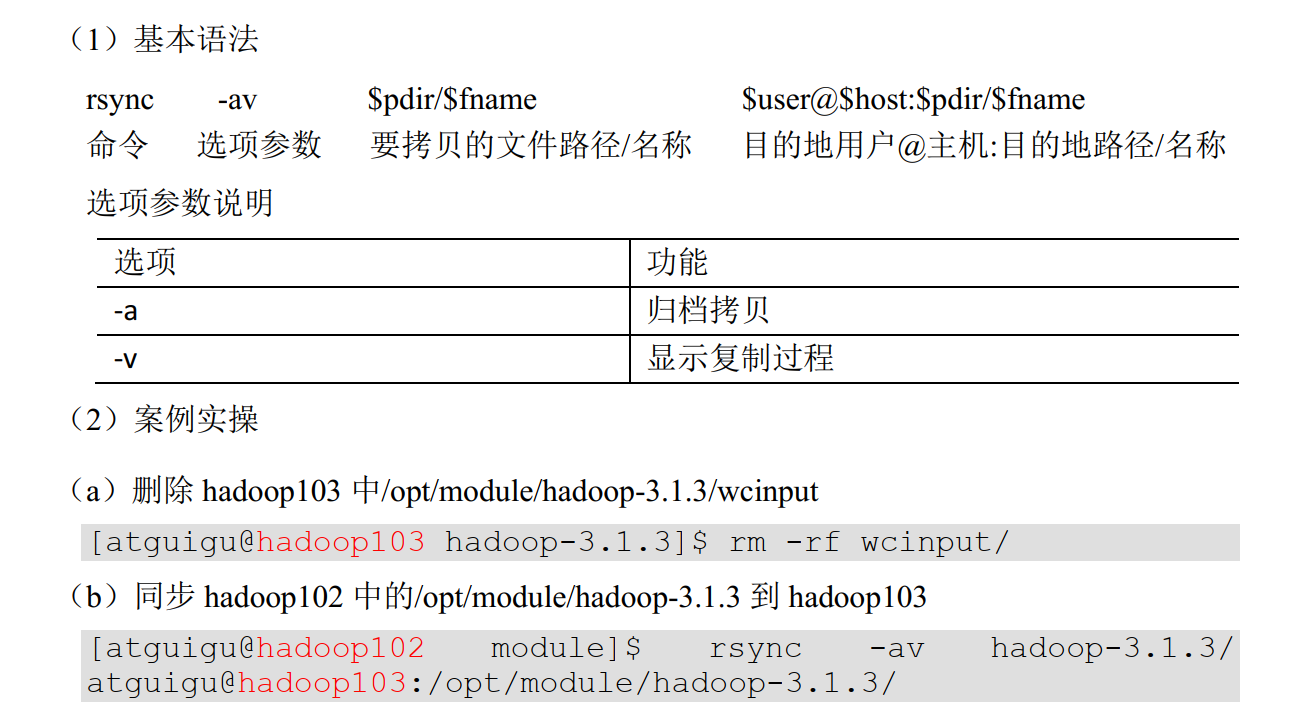
**9）**启动并测试集群

2.1 编写集群分发脚本 xsync

rsync 远程同步工具

rsync 主要用于备份和镜像。具有速度快、避免复制相同内容和支持符号链接的优点。

rsync 和 scp 区别： 用 rsync 做文件的复制要比 scp 的速度快， rsync 只对差异文件做更新。 scp 是把所有文件都复制过去。



编写xsync脚本（/home/hadoop/xsync）：



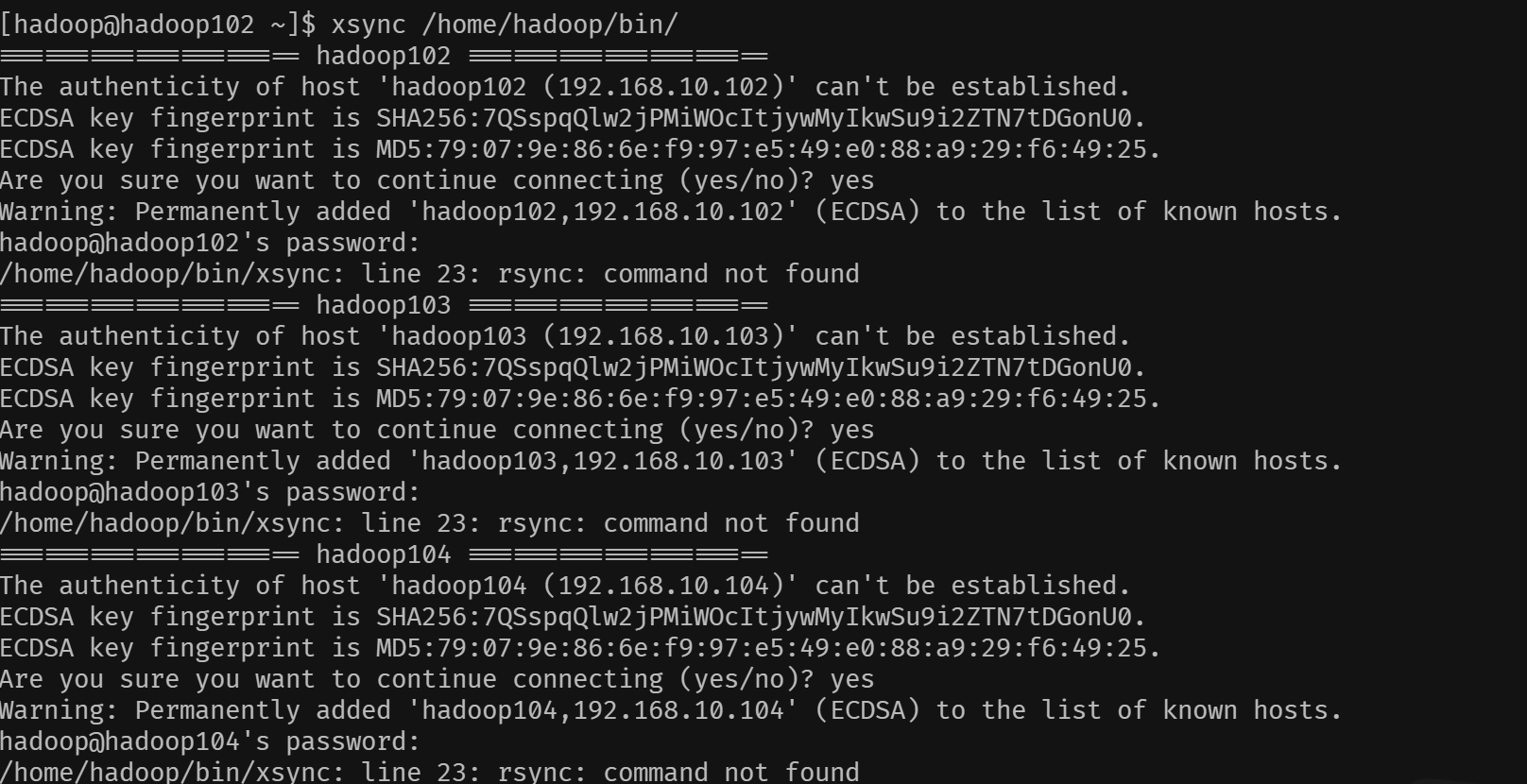
修改脚本 xsync 具有执行权限

chmod +x xsync

在my\_env.sh上将/home/hadoop/bin 设置到PATH

将这个脚本所在的目录同步到其它几个机器上

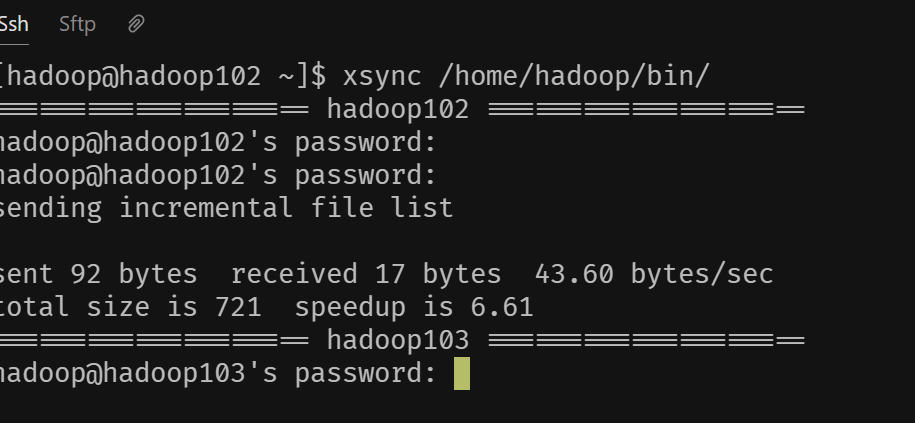
xsync /home/hadoop/bin



我的系统显示未安装rsync，安装下

sudo yum -y install rsync

然后就可以了：



需要到其它几个机器上安装下rsync

* 1. 配置SSH五无密钥登录

另一台电脑的 IP 地址

（2） ssh 连接时出现 Host key verification failed 的解决方法

[atguigu@hadoop102 ~]$ ssh hadoop103

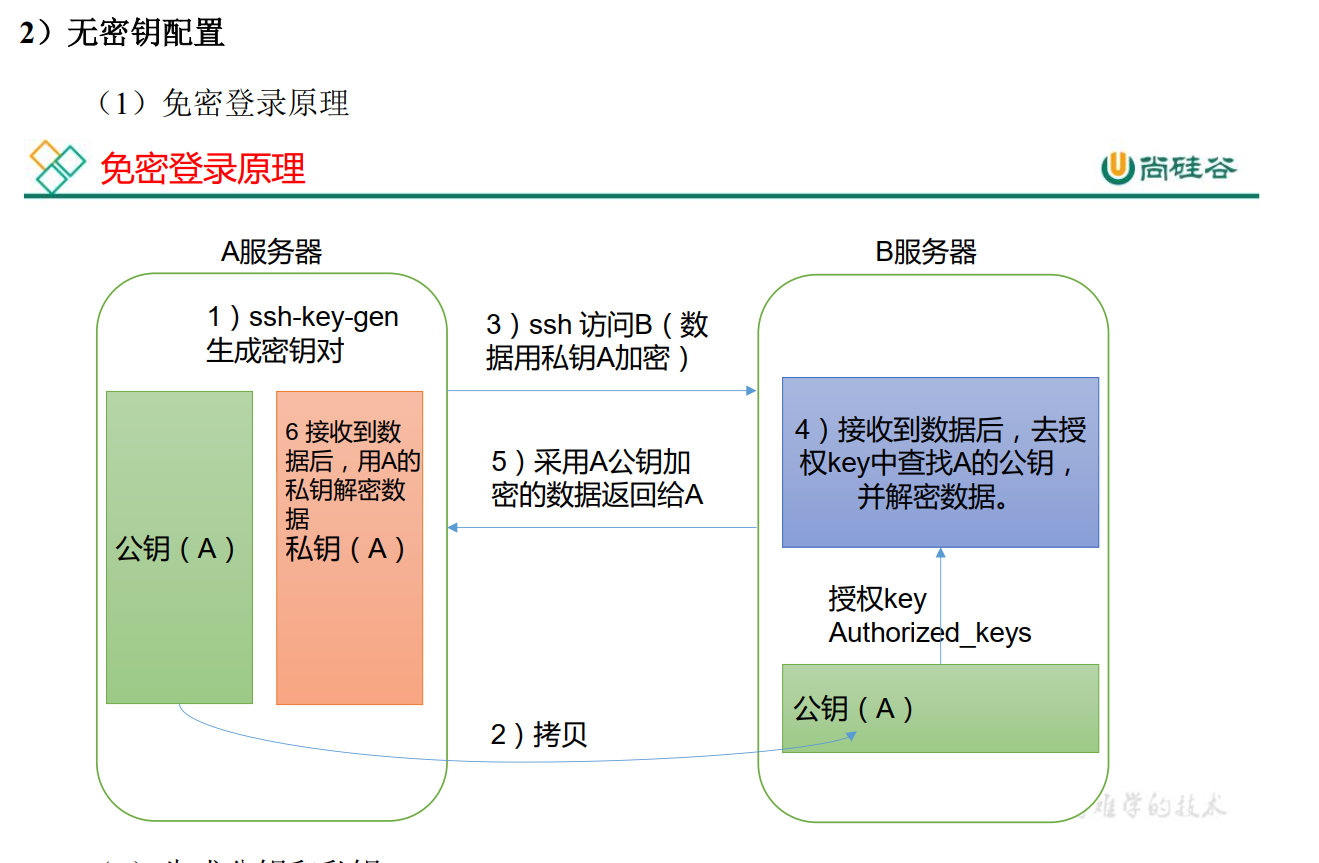
➢ 如果出现如下内容

Are you sure you want to continue connecting (yes/no)?

输入 yes，并回车

（3） 退回到 hadoop102

exit



生成公钥和私钥（在hadoop102的/home/hadoop/.ssh目录下）

ssh-keygen -t rsa

然后敲（三个回车），就会生成两个文件 id\_rsa（私钥）、 id\_rsa.pub（公钥）

将公钥拷贝到要免密登录的目标机器上

ssh-copy-id hadoop102

ssh-copy-id hadoop103

ssh-copy-id hadoop104

hadoop102上ssh hadoop104等服务器，不再需要输入密码即可。

文本

描述已自动生成

注意：

还需要在 hadoop103 上采用 hadoop账号配置一下无密登录到 hadoop102、 hadoop103、hadoop104 上。

还需要在 hadoop104 上采用 hadoop账号配置一下无密登录到 hadoop102、 hadoop103、hadoop104 上。

还需要在 hadoop102 上采用 root 账号，配置一下无密登录到 hadoop102、 hadoop103、hadoop104；

* 1. .ssh 文件夹下（~/.ssh） 的文件功能解释

文本, 表格

描述已自动生成

* 1. 集群部署规划

表格

描述已自动生成

* 1. 配置文件说明

Hadoop 配置文件分两类：默认配置文件和自定义配置文件，只有用户想修改某一默认

配置值时，才需要修改自定义配置文件，更改相应属性值。

（1）默认配置文件

表格

描述已自动生成

（2）自定义配置文件：

core-site.xml、 hdfs-site.xml、 yarn-site.xml、 mapred-site.xml 四个配置文件存放在

$HADOOP\_HOME/etc/hadoop 这个路径上， 用户可以根据项目需求重新进行修改配置。

* 1. 核心配置文件

cd $HADOOP\_HOME/etc/hadoop

1. 配置核心配置文件core-site.xml

|  |
| --- |
| <!-- 指定 NameNode 的地址 --> <property>  <name>fs.defaultFS</name>  <value>hdfs://hadoop102:8020</value> </property> <!-- 指定 hadoop 数据的存储目录 --> <property>  <name>hadoop.tmp.dir</name>  <value>/opt/module/hadoop-3.1.3/data</value> </property> <!-- 配置 HDFS 网页登录使用的静态用户为 hadoop --> <property>  <name>hadoop.http.staticuser.user</name>  <value>hadoop</value> </property> |

1. HDFS 配置文件

|  |
| --- |
| <!-- nn web 端访问地址--> <property>  <name>dfs.namenode.http-address</name>  <value>hadoop102:9870</value> </property> <!-- 2nn web 端访问地址--> <property>  <name>dfs.namenode.secondary.http-address</name>  <value>hadoop104:9868</value> </property> |

1. YARN 配置文件

|  |
| --- |
| <!-- 指定 MR 走 shuffle --> <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value> </property>  <!-- 指定 ResourceManager 的地址--> <property>  <name>yarn.resourcemanager.hostname</name>  <value>hadoop103</value> </property>  <!-- 环境变量的继承 --> <property>  <name>yarn.nodemanager.env-whitelist</name> <value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CO NF\_DIR,CLASSPATH\_PREPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAP RED\_HOME</value> </property> |

1. MapReduce 配置文件

|  |
| --- |
| <!-- 指定 MapReduce 程序运行在 Yarn 上 --> <property>  <name>mapreduce.framework.name</name>  <value>yarn</value> </property> |

使用xsync分发下更改的目录

xsync /opt/module/hadoop-3.1.3/etc/hadoop/

在其它机器上确认下：

cat /opt/module/hadoop-3.1.3/etc/hadoop/core-site.xml

* 1. 启动集群

1. 配置 workers

vim /opt/module/hadoop-3.1.3/etc/hadoop/workers

写入

hadoop102  
hadoop103  
hadoop104

注意：该文件中添加的内容结尾不允许有空格，文件中不允许有空行。

同步所有节点配置文件

xsync /opt/module/hadoop-3.1.3/etc

1. 启动集群
2. 格式化NameNode

如果集群是第一次启动，需要在 hadoop102 节点格式化 NameNode（注意： 格式化 NameNode， 会产生新的集群 id， 导致 NameNode 和 DataNode 的集群 id 不一致，集群找不到已往数据。 如果集群在运行过程中报错，需要重新格式化 NameNode 的话， 一定要先停止 namenode 和 datanode 进程， 并且要删除所有机器的 data 和 logs 目录，然后再进行格式化。）

Hadoop102上执行：hdfs namenode -format

1. 启动 HDFS

sbin/start-dfs.sh

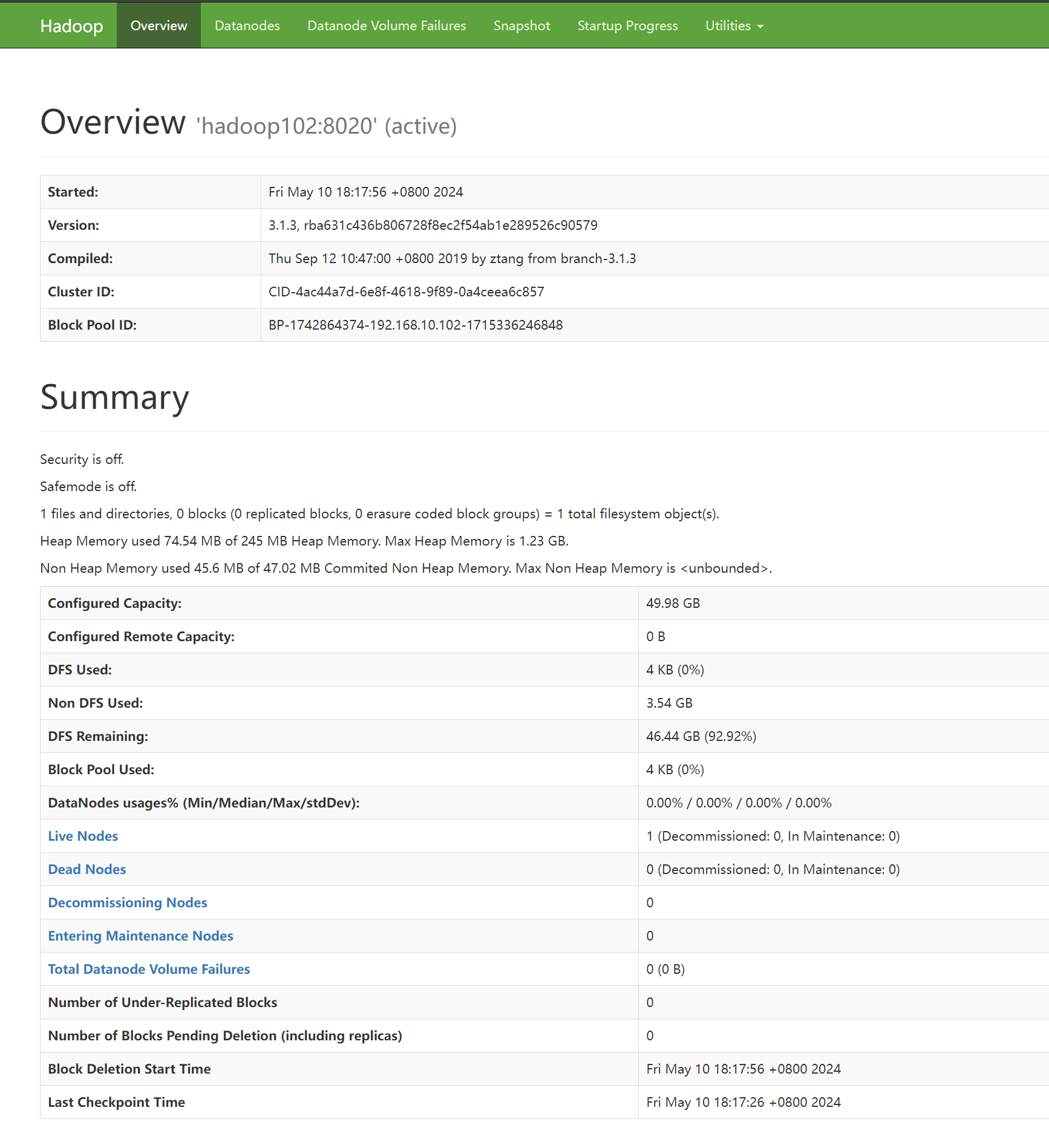
1. 在配置了 ResourceManager 的节点（hadoop103） 启动 YARN

sbin/start-yarn.sh

1. Web 端查看 HDFS 的 NameNode

浏览器中输入： http://hadoop102:9870

查看 HDFS 上存储的数据信息



1. Web 端查看 YARN 的 ResourceManager

浏览器中输入： http://hadoop103:8088

查看 YARN 上运行的 Job 信息

图形用户界面, 应用程序

描述已自动生成

# Kerberos安装和使用

# Zookeeper集群搭建和使用

# MySQL安装和使用

# MySQL集群搭建和使用

# Hive集群搭建和使用

# Spark集群搭建和使用

# Flink集群搭建和使用

# Kafka集群搭建和使用

# ClickHouse安装和使用

# Elasticsearch集群搭建和使用

# CDH-6.3.2集群搭建和使用

# Hue安装和使用

# ELK系统搭建和使用

# Redis安装（单机）和使用

# Redis集群搭建和使用

# MongoDB安装和使用

# Flume集群搭建和使用

# Sqoop安装和使用

# Maven使用