

# Hadoop安装使用

**Author:LiYanHao1209**

## 0. Refs

1. 新增hadoop用户

2. 配置java环境

3. Hadoop安装

4. Hadoop配置

4.1. core-site.xml

4.2. hdfs-site.xml

4.3. mapred-site.xml

5. 环境变量配置

6. 连接到ssh

7. 格式化NameNode

8. 开启name/data node 守护进程

9. 运行测试例子

## 0. Refs

<https://dbllab.xmu.edu.cn/blog/2441/>

## 1. 新增hadoop用户

```
sudo useradd -m hadoop -s /bin/bash # 创建新用户
sudo passwd hadoop # 重设密码
sudo adduser hadoop sudo # 提升至管理员权限
```

最终切换用户到hadoop，GUI或CMD均可

```
su - hadoop
```

## 2. 配置java环境

先下载jdk 8u 162，官网:

```
https://www.oracle.com/java/technologies/javase/javase8-archive
```

解压:

```
cd /usr/lib
sudo mkdir jvm #创建/usr/lib/jvm目录用来存放JDK文件
cd ~ #进入hadoop用户的主目录
cd Downloads #注意区分大小写字母，刚才已经通过FTP软件把JDK安装包jdk-8u1
sudo tar -zxvf ./jdk-8u162-linux-x64.tar.gz -C /usr/lib/jvm #把J
```

配置环境变量：

```
cd ~
vim ~/.bashrc
```

```
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
```

```
source ~/.bashrc # 让配置生效
java -version # 查看是否安装成功
```

## 3. Hadoop安装

官网:

```
https://archive.apache.org/dist/hadoop/common/
```

解压：

```
sudo tar -zxvf ~/Downloads/hadoop-3.1.3.tar.gz -C /usr/local # 解
cd /usr/local/
sudo mv ./hadoop-3.1.3/ ./hadoop # 将文件夹名改为hadoop
sudo chown -R hadoop:hadoop ./hadoop # 修改文件权限
```

查看是否可用:

```
cd /usr/local/hadoop
./bin/hadoop version
```

## 4. Hadoop配置

### 4.1. core-site.xml

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

### 4.2. hdfs-site.xml

```
<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>
```

### 4.3. mapred-site.xml

```
<configuration>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
  </property>
</configuration>
```

这一部分教程里没提到，但是不配置就没有map-reduce应用框架的运行环境，我们必须告诉Hadoop，MapReduce应用框架的运行环境在什么位置上。也就是HADOOP\_MAPRED\_HOME。

## 5. 环境变量配置

```
sudo vim ~/.bashrc
```

增加以下内容：

```
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

export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

使其生效：

```
source ~/.bashrc
```

## 6. 连接到ssh

```
sudo apt-get install openssh-server # 先安装个这个软件
ssh localhost
```

但这样要密码，改成免密登录：

```
exit # 退出刚才的 ssh localhost
cd ~/.ssh/ # 若没有该目录，请先执行一次ssh local
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
```

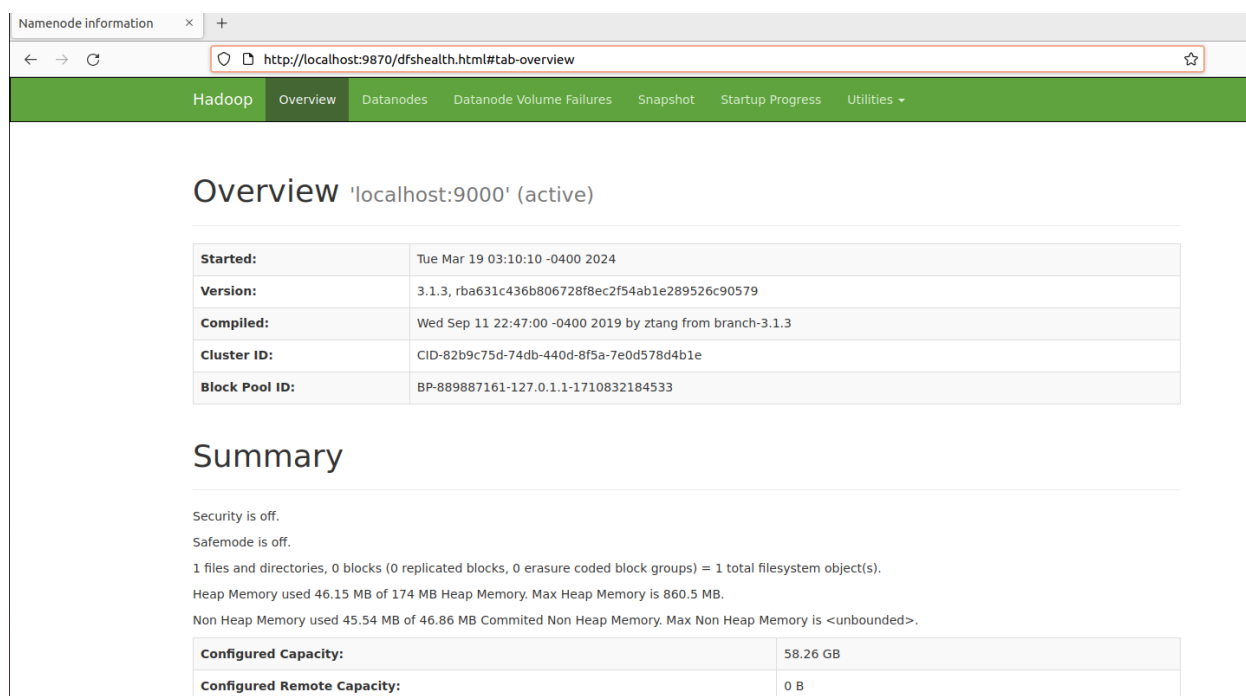
## 7. 格式化NameNode

```
hdfs namenode -format
```

## 8. 开启name/data node 守护进程

```
start-dfs.sh
```

成功启动后，可以访问 Web 界面 <http://localhost:9870> 查看 NameNode 和 Datanode 信息，还可以在线查看 HDFS 中的文件。



The screenshot shows the Hadoop web interface for a NameNode. The browser address bar displays <http://localhost:9870/dfshealth.html#tab-overview>. The interface has a green navigation bar with tabs: Hadoop, Overview (selected), Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area is titled "Overview 'localhost:9000' (active)".

Started:	Tue Mar 19 03:10:10 -0400 2024
Version:	3.1.3, rba631c436b806728f8ec2f54ab1e289526c90579
Compiled:	Wed Sep 11 22:47:00 -0400 2019 by ztang from branch-3.1.3
Cluster ID:	CID-82b9c75d-74db-440d-8f5a-7e0d578d4b1e
Block Pool ID:	BP-889887161-127.0.1.1-1710832184533

**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 46.15 MB of 174 MB Heap Memory. Max Heap Memory is 860.5 MB.  
Non Heap Memory used 45.54 MB of 46.86 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	58.26 GB
Configured Remote Capacity:	0 B

## 9. 运行测试例子

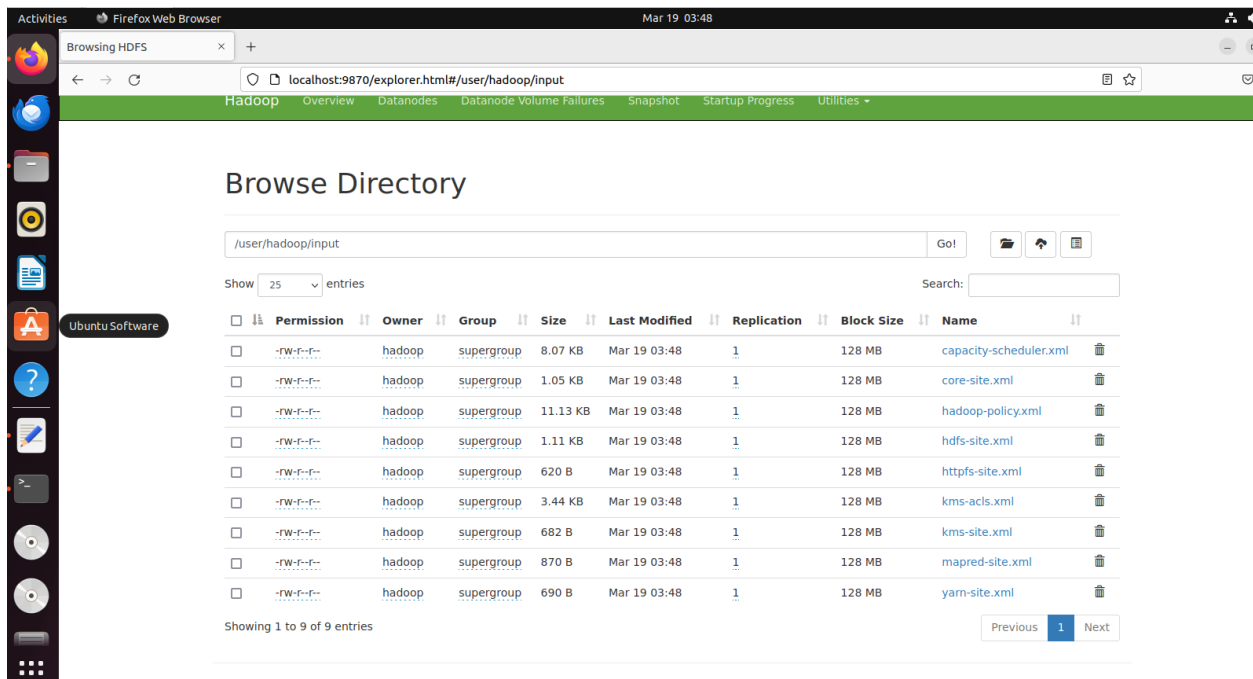
将 `./etc/hadoop` 中的 xml 文件作为输入文件复制到分布式文件系统中，即将 `/usr/local/hadoop/etc/hadoop` 复制到分布式文件系统 `/user/hadoop/input` 中。我们使用的是 `hadoop` 用户，并且已创建相应的用户目录 `/user/hadoop`，因此在命令中就可以使用相对路径如 `input`，其对应的绝对路径就是 `/user/hadoop/input`：

```
hdfs dfs -mkdir -p /user/hadoop/input
./bin/hdfs dfs -put ./etc/hadoop/*.xml input
```

复制完成后，可以通过如下命令查看文件列表：

```
./bin/hdfs dfs -ls input
```

或者通过web页面查看



可以看到我们把本地的hadoop的etc的所有.xml的配置文件全部放到了hdfs上。

伪分布式运行 MapReduce 作业的方式跟单机模式相同，区别在于伪分布式读取的是 HDFS中的文件,运行jar包的demo

```
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-exam
```

可以看到web界面中已经包含了最终的输出文件，汇总到了out文件夹下

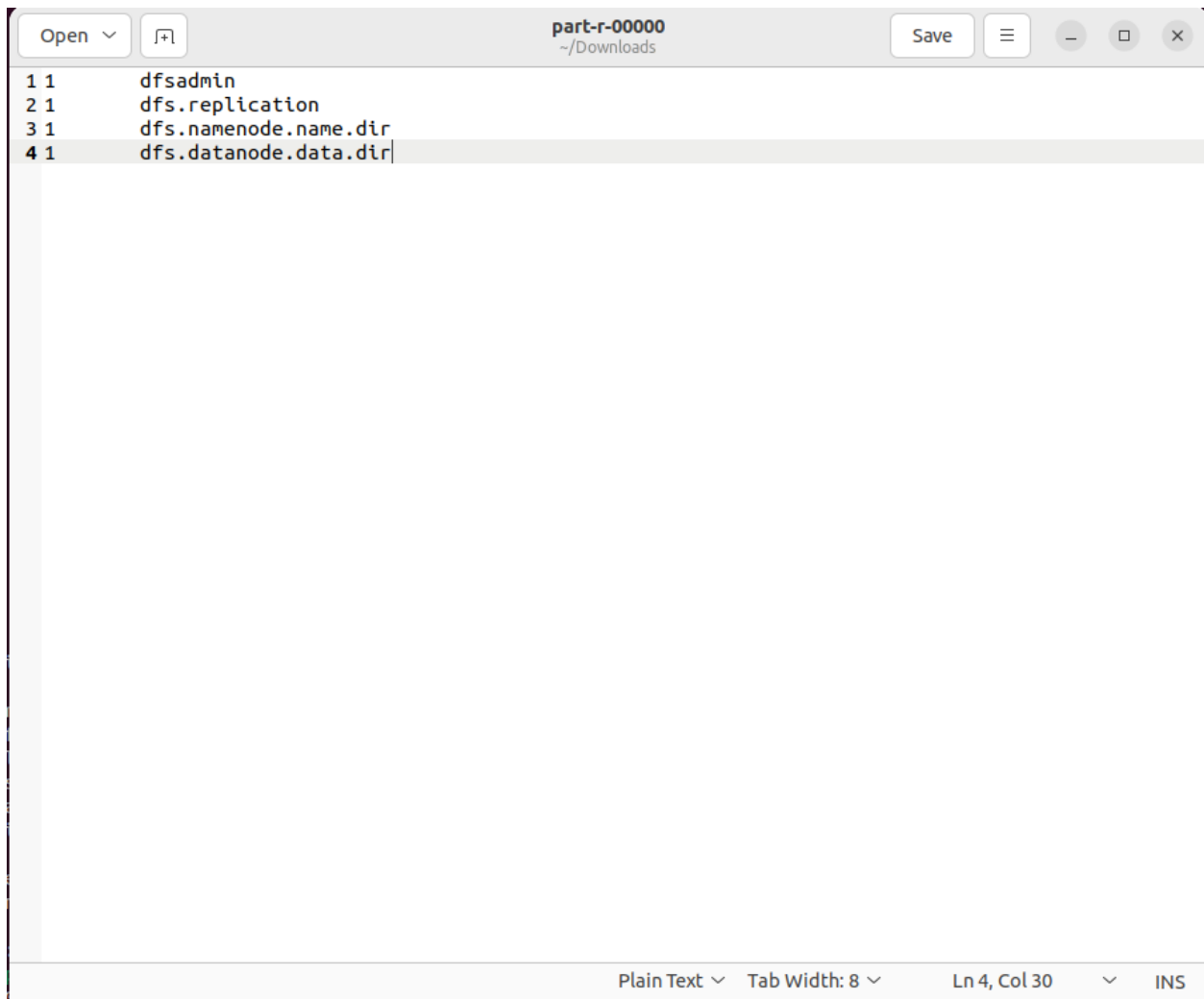
查看运行结果的命令（查看的是位于 HDFS 中的输出结果）：

```
./bin/hdfs dfs -cat output/*
```

```
Bytes Written=77
hadoop@hadoop: /usr/local/hadoop$ ./bin/hdfs dfs -cat output/*
2024-03-19 06:12:37,269 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2024-03-19 06:12:38,322 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
1 dfsadmin
1 dfs.replication
1 dfs.namenode.name.dir
1 dfs.datanode.data.dir
```

可以看到，我们的任务是找到配置文件中所有以dfs为前缀的字符串，并且统计这样的串的数量，mapReduce应用成功把他们统计了出来。

还可以通过get命令把output从hadoop服务器上抓下来放到本地文件系统。



和命令行的结果是一样的。