**大数据实训环境安装**

**Hadoop**

|  |  |  |
| --- | --- | --- |
|  | 文件标识： |  |
| 当前版本： | 1.0 |
| 作 者： | 马卫花 |
| 完成日期： | 2022-10 |

**目 录**

[1准备实验环境 2](#_Toc27258)

[2安装JDK 3](#_Toc9739)

[3 SSH配置 5](#_Toc4889)

[4安装Hadoop 7](#_Toc31838)

[4.1 Hadoop下载 7](#_Toc16432)

[4.2 Hadoop安装目录介绍 7](#_Toc776)

[4.3 核心配置文件 8](#_Toc10253)

[4.4 环境变量配置 11](#_Toc26372)

[4.5 格式化Hadoop 11](#_Toc31739)

[4.6 启动和停止Hadoop 11](#_Toc29614)

[5 HDFS常用操作命令 13](#_Toc9278)

[6 Hadoop案例 14](#_Toc22855)

# 1准备实验环境

本实验主要介绍Hadoop的单点配置和伪分布式集群部署，并通过示例了解Hadoop的运行。

该实验所需的安装文件: VMware Workstation、Centos7.5。

# 2安装JDK

Hadoop的运行需要Java环境，因此必须在所有节点上安装Java,并且对Java版本有如下要求：

Hadoop3.3 以上支持Java 8 and 11,但是编译时只支持Java 8

Hadoop3.0 -- 3.2 只支持Java 8

Hadoop2.7.x--2.x支持Java 7 and Java 8

**2.1 下载JDK**

下载将JDK的链接地址为:https

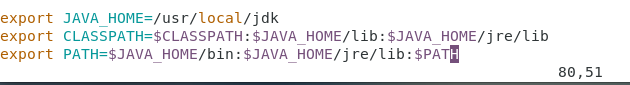
jdk-8u181-linux-x64.tar安装文件

**2.2 卸载系统原有的JDK**

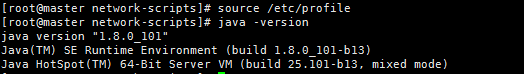
Centos中运行命令如下所示：

|  |
| --- |
| [root@hadoop ~]# rpm -qa | grep java  [root@hadoop ~]# rpm -qa | grep jdk  [root@hadoop ~]# rpm -qa | grep gcj  没有，如果有的话要卸载，卸载案例如下：  卸载linux自带open jdk，将前面三条命令检查出来的内容一一卸载：  [root@master ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.99-2.6.5.1.0.1.el6.x86\_64  [root@master ~]# rpm -e --nodeps tzdata-java-2016c-1.el6.noarch  [root@master ~]# rpm -e java-1.6.0-openjdk-1.6.0.38-1.13.10.4.el6.x86\_64  [root@master ~]# rpm -e java-1.7.0-openjdk-1.7.0.99-2.6.5.1.0.1.el6.x86\_64    卸载完成后应该再检查一次 |

**2.3 手动安装JDK**

JDK安装完成后，设置环境变量，如下所示。（**提示修改PATH**）

编辑完成后保存文件退出，然后输入source /etc/profile使环境变量设置生效。



# 3 SSH配置

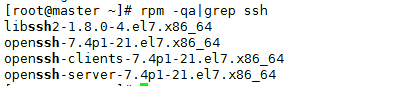
SSH免密登陆的步骤如下所示:

1) 测试SSH

集群、单节点模式都需要用到SSH登陆（类似于远程登陆，可以登陆某台Linux主机，并且在上面运行命令），一般情况下，Centos默认已安装了SSH Client、SSH Server打开终端执行如下命令进行检验。

[root@master ~]# rpm -qa|grep ssh

如果显示如下结果包含了SSH client、SSH Server则不需要安装。



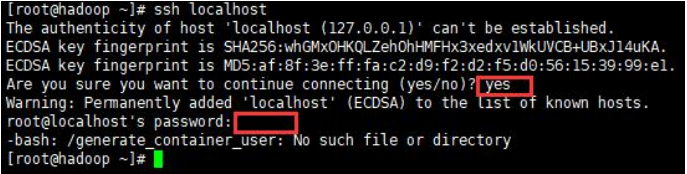
如需要安装，则可以通过yum进行安装。

yum install openssh-clients

yum install openssh-server

1. SSH无秘钥登陆

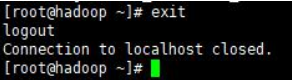
在centos命令窗口输入ssh localhost，根据提示输入密码显示效果如下图所示：



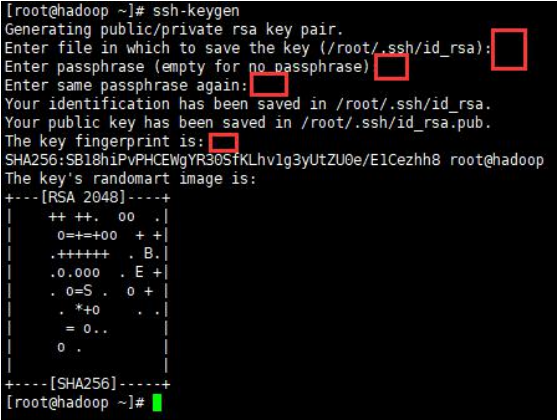
但是每次登陆都需要输入密码，因此本实训将SSH登陆配置为无密码登陆。

1. 输入exit退出登录

首先输入exit退出刚才SSH的登陆，就返回到原来的终端窗口。界面如下所示：



1. 生成公钥和私钥



在上面的几个位置**按回车**就可。

1. 把公钥发送给信任主机，本次实训是主机是本机。



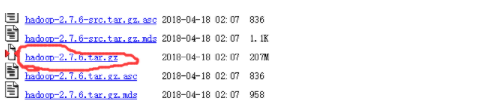
1. 免密验证

此时在输入ssh localhost，无需输入密码就可以直接登陆。

# 4安装Hadoop

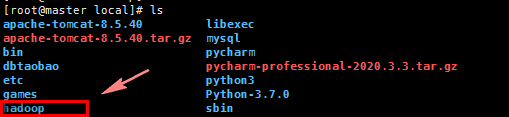
**4.1 Hadoop下载**

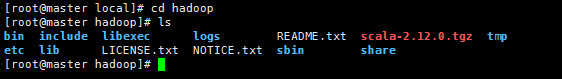
<http://archive.apache.org/dist/hadoop/core/hadoop-2.7.6/>



**4.2 Hadoop安装目录介绍**

将下载的hadoop-2.7.6解压到/usr/local目录下，解压后的目录如下所示,并在hadoop目录下新建tmp，然后在tmp下新建dfs/name,dfs/data目录。





Hadoop可以在单节点上以伪分布式的方式运行，Hadoop进程以分离Java进程来运行，节点即可以作为NameNode也可作为DataNode，同时读取的是HDFS中的文件。

解压后的Hadoop 目录说明。

|  |  |
| --- | --- |
| 目录名称 | 描述 |
| bin | Hadoop最基本的管理脚本和使用脚本所在的目录 |
| etc | Hadoop配置文件所在的目录。包括core-site.xml、Hdfs-site.xml、mapred-site.xml、yarn-site.xml等配置文件 |
| include | 对外提供的编程库头文件, 这些文件都是C++定义的,通常用于C++程序访问HDFS或者编写MapReduce等 |
| lib | 包含了Hadoop对外提供的编程动态库和静态库，与include目录中的头文件结合使用 |
| libexec | 各个服务对应的shell配置文件所在的目录，可用于配置日志输出目录、启动参数等基本信息。 |
| sbin | Hadoop管理脚本所在目录,主要包含HDFS和Yarn中各类服务启动、关闭的脚本 |
| share | Hadoop各个模块编译后的jar包所在目录，这个目录中包含了Hadoop文档 |

**4.3 核心配置文件**

1.core-site.xml文件

core-site.xml是Hadoop核心的配置文件，可在此文件中配置文件系统的访问端口和访问权限等。

vim /usr/local/haoop/core-site.xml

* fs.defaultFS

文件系统访问路径，准备将node01配置为NameNode节点，因此在伪分布式中配置以下值，本地文件系统配置为file:/// ,伪分布式可配置为hdfs://node01:9000

* hadoop.tmp.dir

Hadoop运行时的临时存储目录，默认值为/tmp/hadoop-${user.name}，配置后就会在Hadoop的安装目录下。

* io.file.buffer.size

用作序列化文件处理时读写buffer的大小，默认为4096,单位为byte，可以设置的大一些，较大的缓存都可以提供更高的数据传输，但是也可能带来更大的内存消耗和延迟。一般情况下可以设置为64KB

|  |
| --- |
| <configuration>  <!--Hadoop运行时临时的存储目录**必须配置**-->  <property>  <name>hadoop.tmp.dir</name>  <value>file:/usr/local/hadoop/tmp</value>  <description>Abase for other temporary directories.</description>  </property>  <！--Hadoop HDFS中namenode的URI和端口必须配置-->  <property>  <name>fs.defaultFS</name>  <value>hdfs://node01:9000</value>  </property>  </configuration> |

1. hdfs-site.xml文件

此配置文件是hdfs核心配置文件，对应于HDFS模块，此配置文件配置文件系统数据存储路径和SecondaryNameNode地址等.

* dfs.namenode.name.dir

HDFS中NameNode节点元数据存储目录，默认为file://${hadoop.tmp.dir}/dfs/name。

* dfs.datanode.data.dir

HDFS中DataNode节点的数据存储目录，是分布式文件系统中文件存放的真正地址，以block块的形式存储到该目录。

* dfs.replication

DataNode存储的Block的副本数量，不大于DataNode的个数就行，默认为3，实际生产环境中设置为3即可，过多则会更多地占用磁盘空间。

* dfs.namenode.checkpoint.dir

指定SecondaryNamenode的工作目录，即辅助名称节点在本地文件系统上存储要合并的临时fsimage的位置.

|  |
| --- |
| <!-- Put site-specific property overrides in this file. -->  <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> |

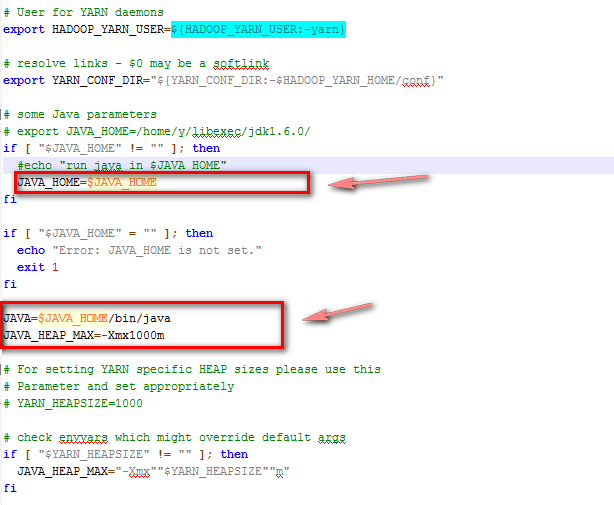
配置/hadoop/etc/hadoop/目录下hadoop-env.sh、yarn-env.sh的JAVA\_HOME如下图所示:

修改**hadoop-env.sh**



提示:java\_home为jdk1.8的安装路径。（eg:/usr/local/jdk1.8.0\_181）

修改**yarn-env.sh**的文件配置如下图所示。



提示：将JAVA\_HOME改为jdk的安装路径。

**4.4 环境变量配置**

配置Hadoop的环境变量如下所示,配置完成后，执行source /etc/profile,环境变量才能生效。

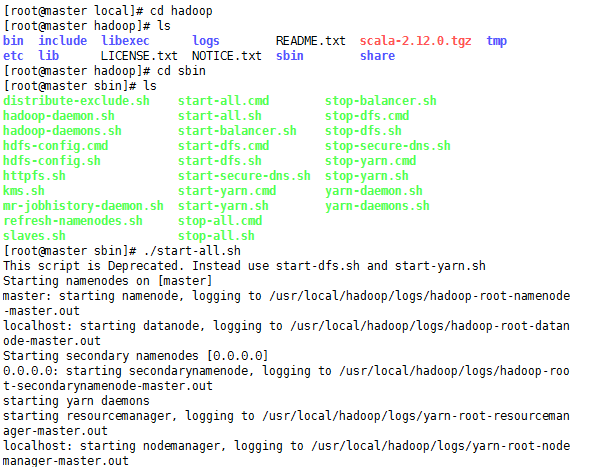




**4.5 格式化Hadoop**

格式:hdfs namenode -format

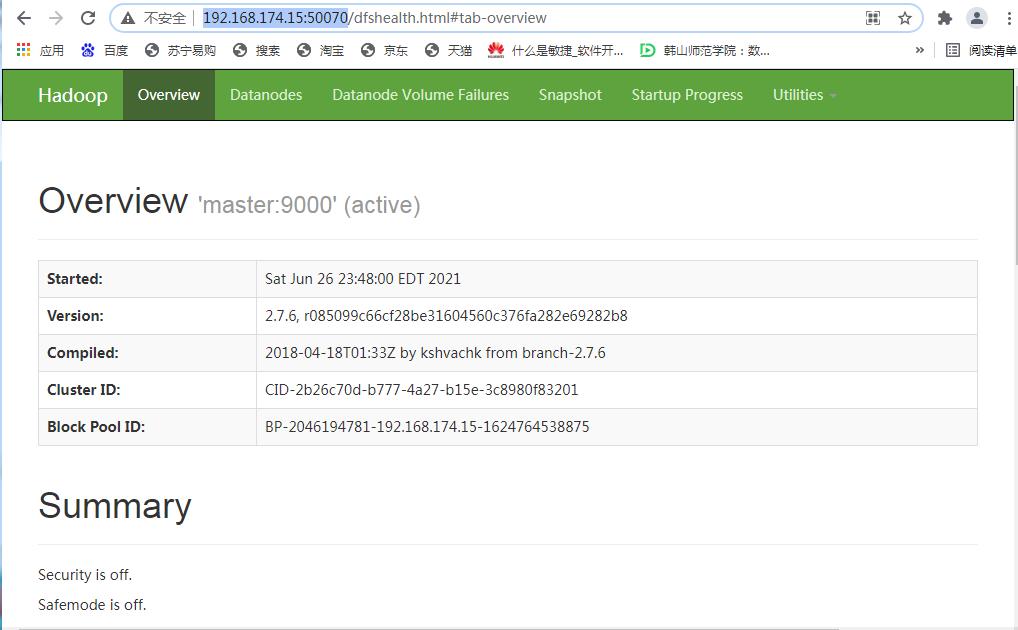
**4.6 启动和停止Hadoop**



hadoop启动成功后，输入jps命令查看系统的进程如下图所示:



打开浏览器输入如下的IP地址http://node01:50070/,显示如下图所示的界面:



停止Hadoop命令

[root@master local]# stop-all.sh

# 5 HDFS常用操作命令

Hdfs的命令和Linux命令很相似，都属于目录树的结构。

* 创建一个文件或目录

|  |
| --- |
| [root@master data]# hdfs dfs -mkdir /wordcount  创建多级目录结构的命令  [root@master data]# hdfs dfs -mkdir /wordcount |

* 查看HDFS文件目录

|  |
| --- |
| [root@master data]# hdfs dfs -ls / |

* 把本地文件上传至HDFS文件系统上

|  |
| --- |
| [root@master data]# hdfs dfs -put word.txt /wordcount |

* 下载文件word.txt

|  |
| --- |
| [root@master data]# hdfs dfs -get /wordcount/word.txt |

* 修改目录权限改为777

|  |
| --- |
| [root@master data]# hdfs dfs -chmod -R 777 /wordcount |

* 查看word.txt文件大小

|  |
| --- |
| [root@master data]# hdfs dfs -du /wordcount |

* 删除文件及目录

|  |
| --- |
| 删除文件  [root@master data]# hdfs dfs -rm /wordcount/\*  删除文件夹  [root@master data]# hdfs dfs -rm -r /wordcount |

# 6 Hadoop案例

1） 本地创建文件夹

|  |
| --- |
| [root@master data]# mkdir input  [root@master data]# cd input/  [root@master input]# echo "hello hadoop" > file1.txt  [root@master input]# echo "hello mapreduce" > file2.txt  [root@master input]# echo "hello spark" > file3.txt  [root@master input]# ls  file1.txt file2.txt file3.txt |

2） 将文件上传到HDFS

|  |
| --- |
| [root@master input]# hdfs dfs -mkdir /wordcount  [root@master input]# hdfs dfs -ls /  Found 2 items  drwxr-xr-x - root supergroup 0 2021-06-27 03:07 /userdata  drwxr-xr-x - root supergroup 0 2021-06-27 03:30 /wordcount  [root@master input]# cd ..  [root@master data]# hdfs dfs -put input/\* /wordcount  [root@master data]# hdfs dfs -ls /wordcount  Found 3 items  -rw-r--r-- 1 root supergroup 13 2021-06-27 03:32 /wordcount/file1.txt  -rw-r--r-- 1 root supergroup 16 2021-06-27 03:32 /wordcount/file2.txt  -rw-r--r-- 1 root supergroup 12 2021-06-27 03:32 /wordcount/file3.txt |

3） 启用hadoop自带单词统计

|  |
| --- |
| [root@master data]# cd /usr/local/hadoop  [root@master hadoop]# ls  bin include libexec logs README.txt scala-2.12.0.tgz tmp  etc lib LICENSE.txt NOTICE.txt sbin share  [root@master hadoop]# cd share  [root@master share]# ls  doc hadoop  [root@master share]# cd hadoop  [root@master hadoop]# ls  common hdfs httpfs kms mapreduce tools yarn  [root@master hadoop]# cd mapreduce/  [root@master mapreduce]# ls  hadoop-mapreduce-client-app-2.7.6.jar  hadoop-mapreduce-client-common-2.7.6.jar  hadoop-mapreduce-client-core-2.7.6.jar  hadoop-mapreduce-client-hs-2.7.6.jar  hadoop-mapreduce-client-hs-plugins-2.7.6.jar  hadoop-mapreduce-client-jobclient-2.7.6.jar  hadoop-mapreduce-client-jobclient-2.7.6-tests.jar  hadoop-mapreduce-client-shuffle-2.7.6.jar  hadoop-mapreduce-examples-2.7.6.jar  lib  lib-examples  sources  [root@master mapreduce]# hadoop jar hadoop-mapreduce-examples-2.7.6.jar wordcount /wordcount /print |

4） 查看结果

|  |
| --- |
| [root@master mapreduce]# hdfs dfs -ls /print  Found 2 items  -rw-r--r-- 1 root supergroup 0 2021-06-27 03:36 /print/\_SUCCESS  -rw-r--r-- 1 root supergroup 37 2021-06-27 03:36 /print/part-r-00000  [root@master mapreduce]# hdfs dfs -cat /print/part-r-00000  hadoop 1  hello 3  mapreduce 1  spark 1 |

【over】