# Hadoop安装d

## 下载安装文件

jdk-8u161-linux-x64.tar.gz

hadoop-2.9.0.tar.gz

## Hadoop安装

### 创建工作目录（root）

|  |
| --- |
| # 使用root用户  mkdir -p /app/data/hadoop  # 改变属主  chown -R bigdata:bigdata /app |

### 解压安装文件（bigdata）

|  |
| --- |
| # 解压JDK文件  tar -zxf jdk-8u161-linux-x64.tar.gz -C /app  # 解压Hadoop文件  tar -zxf hadoop-2.9.0.tar.gz -C /app |

### 配置环境变量（root）

|  |
| --- |
| # 环境变量：  vi /etc/profile.d/app.sh  export JAVA\_HOME=/app/jdk1.8.0\_161  export HADOOP\_HOME=/app/hadoop-2.9.0  export PATH=$JAVA\_HOME/bin:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin:$PATH  . /etc/profile.d/app.sh |

### Hadoop配置（bigdata）

|  |
| --- |
| **# 进入配置文件目录**  cd /app/hadoop-2.9.0/etc/hadoop  **# 配置core-site.xml**  <property>  <name>hadoop.tmp.dir</name>  <value>/app/hadoop/tmp</value>  </property>  <property>  <name>fs.defaultFS</name>  <value>hdfs://192.168.66.11:9000</value>  </property>    **# 配置hdfs-site.xml**  <property>  <name>dfs.namenode.name.dir</name>  <value>file:///app/data/hadoop/name</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:///app/data/hadoop/data</value>  </property>    **# 配置yarn-site.xml**  <property>  <name>hadoop.tmp.dir</name>  <value>/app/hadoop/tmp</value>  </property>  <property>  <name>yarn.resourcemanager.hostname</name>  <value>192.168.66.11</value>  </property> |

### 初始化Hadoop（bigdata）

|  |
| --- |
| # 格式化namenode  hadoop namenode -format |

### 停止防火墙（root）

|  |
| --- |
| # 停止  service firewalld stop  # 禁用  systemctl disable firewalld |

## Hadoop维护

### 启动/停止Hadoop（bigdata）

|  |
| --- |
| # 启动  start-dfs.sh  # 停止  stop-dfs.sh |

### 验证是否启动成功

1．浏览器里面输入：

<http://192.168.66.11:50070/>

2．在命令行输入jps

看NameNode、DataNode、SecondaryNameNode是否存在

## HDFS Shell操作

上传文件

hadoop fs –put <本地文件> <hdfs路径>

查看文件列表

hadoop fs –ls <hdfs路径>

下载文件

hadoop fs –get <hdfs文件路径> <本地路径>

重命名

hadoop fs –mv <hdfs原文件名> <hdfs新文件名>

删除文件

hadoop fs –rm <hdfs文件名>

# 创建文件夹

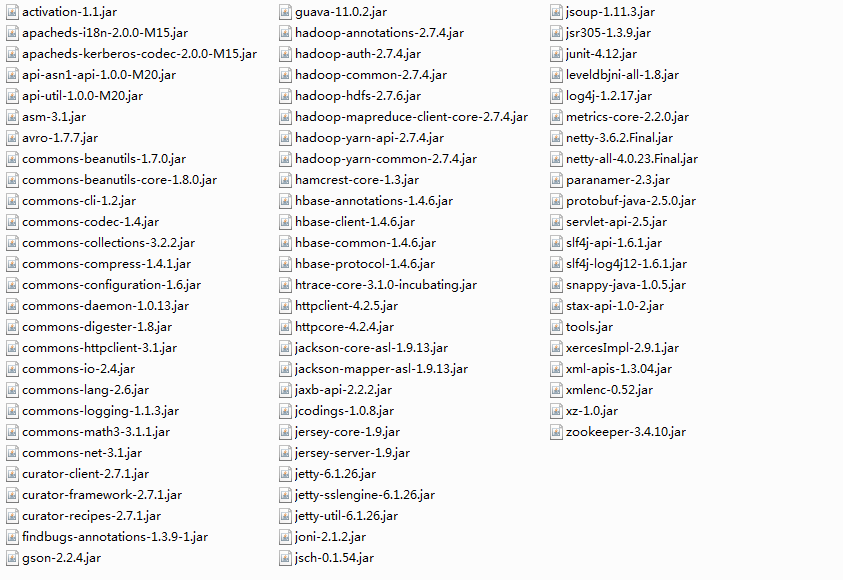
hadoop fs –mkdir <目录名>

## HDFS Java API实现

### 调用Hadoop步骤

#### 导入Hadoop相关依赖包

将以下jar包拷贝到项目的WEB-INFO/lib下



#### 初始化Hadoop配置

Configuration conf = new Configuration();

#### 获取客户端

FileSystem fs = FileSystem.get(new URI(“hdfs://192.168.66.11:9000”), conf);

#### HDFS具体操作

##### 上传文件

fs.copyFromLocalFile(false, false, new Path(localFile), new Path(hfdsFile));

参数说明：

第一个参数，是否删除本地文件

第二个参数，是否覆盖Hdfs的文件

第三个参数，本地文件路径

第四个参数，hdfs文件路径

##### 下载文件

fs.copyToLocalFile(false, new Path(hdfsFile), new Path(localFile), true);

参数说明：

第一个参数，是否删除hdfs文件

第二个参数，hdfs文件路径

第三个参数，本地文件路径

第四个参数，是否使用本地原文件系统(解决本地权限问题)

##### 获取文件列表

FileStatus[] files = fs.listStatus(**new** Path(path));

// 4.循环处理文件属性

// 4.1 实例化返回列表

fileList = **new** ArrayList<FileBean>();

// 4.2 循环处理文件列表属性

**for**(FileStatus file : files) {

// 文件名

String fileName = file.getPath().getName();

// 路径

String filePath = file.getPath().getParent().toString();

filePath = filePath.replace(*hdfsUri*, "");

// 时间

**long** datetmp = file.getModificationTime();

String date = **new** SimpleDateFormat("yyyy-MM-dd hh:mm").format(datetmp);

// 类型

String type = "文件";

**if**(file.isDirectory()) {

type = "文件夹";

}

// 大小

String size = file.getLen() + "";

}

##### 删除文件

// 删除文件

fs.delete(**new** Path(file), **false**);

参数说明：

第一个参数，hdfs文件

第一个参数，是否强制删除(递归删除)

##### 创建文件夹

fs.cre