

Hadoop	<p>1、启动 hadoop 进程 start-all.sh 等价于 start-dfs.sh+start-yarn.sh(推荐后一种, 因为前一种在内部命令中存在一定的问题)</p> <p>2、查看指定目录内容 hdfs dfs -ls 目录 如: hdfs dfs -ls /usr</p> <p>3、打开某个文件 hdfs dfs -cat 文件路径</p> <p>4、本地文件存储值 hadoop, 本地路径也可以是目录 hdfs dfs -put 本地路径 hadoop 目录</p> <p>5、将 hadoop 上的文件下载至本地 hadoop dfs -get hadoop 路径 本地路径</p> <p>6、删除 hadoop 上指定文件 hdfs dfs -rm 文件路径</p> <p>7、删除 hadoop 上指定文件夹(递归删除) hdfs dfs -rmr 文件夹路径</p> <p>8、Hadoop 在指定目录上创建新的目录 hdfs dfs -mkdir 目录路径 递归创建目录 hdfs dfs -mkdir -p 目录路径</p> <p>9、hadoop 指定目录下新建一个空文件 hdfs dfs -touchz 文件路径</p> <p>10、移动文件 hdfs dfs -mv 源文件路径 目标文件路径</p> <p>11、查看帮助 hdfs dfs -help</p> <p>12、运行 mapreduce 示例程序 pi 计算样例: /share/hadoop/mapreduce 下 hadoop jar hadoop-mapreduce-examples-2.7.3.jar pi 5 5</p> <p>注: 一般也可以是/bin/yarn jar 命令</p> <p>wordcount 样例:</p>
--------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre>hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount 自定义 hadoop 输入目录 自定义 hadoop 输出目录</pre> <p>13、重启 NameNode</p> <pre>hadoop-daemon.sh stop namenode</pre> <pre>hadoop-daemon.sh start namenode</pre> <p>14、重启 DataNode</p> <pre>hadoop-daemon.sh stop datanode</pre> <pre>hadoop-daemon.sh start datanode</pre> <p>查看 hdfs 报告</p> <pre>hdfs dfsadmin -report</pre> <p>查看 hdfs 的空间情况</p> <pre>hadoop fs -df</pre>
git	<p>1、远程免密连接</p> <p>生成 rsa 密钥，复制公钥至 github 处</p> <p>连接测试：</p> <pre>ssh -T git@github.com</pre> <p>2、将当前目录初始化成本地仓库</p> <pre>git init</pre> <p>3、将文件添加/删除至本地仓库变更缓冲区</p> <pre>git add 文件名</pre> <pre>git add .      #提交被修改的和新建的文件，但不包括被删除的文件</pre> <pre>git add -u     #-update 更新所有改变的文件，即提交所有变化的文件</pre> <pre>git add -A     #-all 提交已被修改和已被删除文件，但是不包括新的文件 4</pre> <pre>git rm 文件名</pre> <p>4、配置提交人参数</p> <pre>git config --global user.name '名称' #配置提交人的名称</pre> <pre>git config --global user.email '邮箱' #配置提交人邮箱</pre>

5、提交变更

```
git commit -m "注释内容"
```

6、添加/删除一个远程服务器别名至本地

```
git remote add 别名 git@github.com:拥有者/仓库名.git
```

```
git remote rm 别名
```

7、提交本地文件变更

```
git push -u 远程服务器名或别名 分支(如: master)
```

8、获取远程服务器变更合并到本地

```
git pull git@github.com:拥有者/仓库名.git
```

9、克隆远程仓库至本地(一般克隆路径可以从 github 上复制到)

```
git clone git@github.com:拥有者/仓库名.git ##以 SSH 方式克隆到本地, 可以读写
```

```
git clone https://github.com/拥有者/仓库名.git ##以 https 方式克隆到本地, 可以读写
```

10、查看当前本地仓库状态

```
git status
```

11、查看所有历史版本信息

```
git log
```

只显示版本号与注释

```
git log --pretty=oneline
```

只查看涉及某个文件的变更

```
git log 文件名
```

操作日志

```
git reflog
```

12、回退版本

```
git reset --hard HEAD^, 回滚到上个版本
```

```
git reset --hard xxx(版本号或版本号前几位), 回滚到指定版本号, 如果是版本号前几位, git 会自动寻找匹配的版本号
```

```
git reset --hard xxx(版本号或版本号前几位) 文件名, 回滚某个文件到指定版本号 (需要进入该文件所在目录)
```

	<p>13、回退本地的单个文件常用最后两步</p> <pre>git reset 版本号 文件名</pre> <pre>git checkout -- 文件名</pre>
linux	<p>1、免密登录 ★ ★ ★</p> <pre>ssh-keygen -t dsa -P "" -f ~/.ssh/id_dsa</pre> <p>master:</p> <pre>cat ~/.ssh/id_dsa.pub &gt;&gt; ~/.ssh/authorized_keys</pre> <p>slaves:</p> <pre>ssh-copy-id -i ~/.ssh/id_dsa.pub master</pre> <p>master:</p> <pre>scp -r /root/.ssh/authorized_keys root@slave1:/root/.ssh/</pre> <pre>scp -r /root/.ssh/authorized_keys root@slave2:/root/.ssh/</pre> <p>权限:</p> <pre>all:</pre> <pre>chmod 777 authorized_keys</pre> <p>uname -a # 查看内核/操作系统/CPU 信息 ★ ★ ★</p> <p>cat /proc/cpuinfo # 查看 CPU 信息 ★ ★ ★</p> <p>hostname # 查看计算机名</p> <p>lspci -tv # 列出所有 PCI 设备</p> <p>lsusb -tv # 列出所有 USB 设备</p> <p>lsmod # 列出加载的内核模块</p> <p>env # 查看环境变量</p> <p>资源</p> <p>free -m # 查看内存使用量和交换区使用量 ★ ★ ★</p> <p>df -h # 查看各分区使用情况 ★ ★ ★</p> <p>du -sh &lt;目录名&gt; # 查看指定目录的大小</p> <p>grep MemTotal /proc/meminfo # 查看内存总量</p> <p>grep MemFree /proc/meminfo # 查看空闲内存量</p> <p>uptime # 查看系统运行时间、用户数、负载</p> <p>cat /proc/loadavg # 查看系统负载</p> <p>磁盘和分区</p>

	<p>mount   column -t # 查看挂接的分区状态</p> <p>fdisk -l # 查看所有分区</p> <p>swapon -s # 查看所有交换分区   ★ ★ ★</p> <p>hdparm -i /dev/hda # 查看磁盘参数(仅适用于 IDE 设备)</p> <p>dmesg   grep IDE # 查看启动时 IDE 设备检测状况</p> <p>网络</p> <p>ifconfig # 查看所有网络接口的属性   ★ ★ ★</p> <p>iptables -L # 查看防火墙设置</p> <p>route -n # 查看路由表</p> <p>netstat -ltnp# 查看所有监听端口 ★ ★ ★</p> <p>netstat -antp # 查看所有已经建立的连接 ★ ★ ★</p> <p>netstat -s # 查看网络统计信息</p> <p>进程</p> <p>ps -ef # 查看所有进程</p> <p>top # 实时显示进程状态   ★ ★ ★</p> <p>用户</p> <p>添加用户   ★ ★ ★</p> <p>useradd dashu 添加用户</p> <p>passwd dashu 指定密码</p> <p>su dashu 切换用户</p> <p>w # 查看活动用户</p> <p>id &lt;用户名&gt; # 查看指定用户信息</p> <p>last # 查看用户登录日志</p> <p>cut -d: -f1 /etc/passwd # 查看系统所有用户</p> <p>cut -d: -f1 /etc/group # 查看系统所有组</p> <p>chkconfig --list # 列出所有系统服务</p> <p>chkconfig --list   grep on # 列出所有启动的系统服务程序</p>
linux 定时任务	<p>每 6 个小时执行一次命令（假设文件名为 hdfs_balancer_cron.cron）</p> <p>0 */6 * * * 命令</p>

	<p>载入定时任务</p> <pre>crontab hdfs_balancer_cron.cron</pre> <p>查看是否载入成功</p> <pre>crontab -l</pre> <p>[删除指定用户的定时任务]</p> <pre>crontab [-u user] -r</pre> <p>执行定时任务</p> <pre>service crond start</pre> <p>停止定时任务</p> <pre>service crond stop</pre>
zookeeper	<p>查看 ZooKeeper 状态</p> <pre>zkServer.sh status</pre> <p>连接到 ZooKeeper 服务</p> <pre>zkCli.sh -server master:2181</pre>
hbase	<p>1、进入 hbase 命令行</p> <pre>hbase shell</pre> <p>2、查看数据库状态</p> <pre>status</pre> <p>3、查看版本</p> <pre>version</pre> <p>4、查看 hbase 中的表</p> <pre>list</pre> <p>5、创建表 score，默认命名空间为 default</p> <pre>create '表名','列族名 1','列族名 2',...</pre> <p>6、删除表</p> <pre>disable '表名'</pre> <pre>drop '表名'</pre> <p>7、显示表的信息</p> <pre>describe '表名'</pre> <p>8、添加记录</p> <pre>put '表名','rowkey','列族:列','值'</pre> <p>9、扫描表所有数据</p> <pre>scan '表名'</pre> <p>10、筛选</p>

	<p>前两条数据，指定列族，指定行键范围</p> <pre>scan '表名',{limit=&gt;2, columns=&gt;'列族名', startrow=&gt;'xxx', stoprow=&gt;'xxx'}</pre> <p>11、获取某个列族</p> <pre>get '表名','rowkey','列族[:列]'</pre> <p>或</p> <pre>get '表名','rowkey',{column=&gt;'列族[:列]'}</pre>
hive	<p>1、进入 hive</p> <p>安装好 hive 配置环境变量后直接 hive 进入</p> <p>2、退出 hive</p> <pre>exit;</pre> <p>3、显示所有数据库</p> <pre>show databases;</pre> <p>4、创建数据库</p> <pre>create database 数据库名;</pre> <p>5、使用某个数据库</p> <pre>use 数据库名;</pre> <p>6、查看数据库详细信息与文件系统位置</p> <pre>describe database 数据库名;</pre> <p>7、创建内部表示例</p> <pre>create table if not exists person(id string,name string,age int) row format delimited  fields terminated by '\,' stored as textfile</pre> <p>8、将数据从本地文件导入到表中</p> <pre>load data local inpath "/opt/test_data/hive_test_data.csv" overwrite into table person;</pre> <p>9、查看表结构</p> <pre>describe 表名</pre> <p>10、删除表</p> <pre>drop table 表名;</pre>

	<p>11、从 hadoop 的文件创建外部表</p> <pre>create external table if not exists book(id int,name string,price int) row format delimited fields terminated by '\,' stored as textfile location '/test_data'</pre> <p>其中，文件存在于 test_data 目录中</p>
sqoop	<p>1、查看 sqoop 帮助</p> <pre>sqoop help 或 sqoop-help</pre> <p>2、查看版本</p> <pre>sqoop version 或 sqoop- version</pre> <p>3、查询 MySQL 所有数据库</p> <pre>sqoop-list-databases --connect jdbc:mysql://master:3306/ --username root --password 123456</pre>