



# hadoop 入门手册

以 hadoop2.7.1 为例

由于 **hadoop** 版本更新比较快，大多数知识都是差不多的，但是最新版本更新，增加了一些命令，和配置变化，这里整理翻译，包括一些基础的配置知识，入门命令，**hdfs shell** 命令等，适合查找容易忘记的命令，配置等，同样适合新手入门，通过本入门手册，了解 **hadoop**。后续不断更新。如有错误不当之处，欢迎批评指正。

-----活到老，学到老

## hadoop 【2.7.1】【多节点】集群配置

### 【必知配置知识 1】

问题导读

**1.说说你对集群配置的认识？**

**2.集群配置的配置项你了解多少？**

**3.下面内容让你对集群的配置有了什么新的认识？**



### 目的

目的 1:

这个文档描述了如何安装配置 **hadoop** 集群，从几个节点到上千节点。为了学习 **hadoop**，你可能先从单节点入手（查看

Single Node Setup). 这里有中文版 [hadoop2.7【单节点】单机、伪分布、分布式安装指导](#)

<http://www.aboutyun.com/thread-12798-1-1.html>

这个文档不包括：hadoop 在安全模式下配置和 HA【高可用配置】，后面在更新

目的 2:

我们看了很多集群配置文档，你是否静下心来，想集群配置到底是怎么一回事。

## 准备

1.安装 Java

2.下载 hadoop 包

#####

包集合:

[hadoop 家族、strom、spark、Linux、flume 等 jar 包、安装包汇总下载\(持续更新\)](#)

<http://www.aboutyun.com/thread-8178-1-1.html>

#####

## 安装

安装 hadoop 集群包括：解压包，配置 hadoop，划分主节点和子节点。

集群中可以将 namenode 和 ResourceManager 分布在不同的机器上,这些称之为 **master**。其它服务例如: Web App Proxy

Server 和 MapReduce Job History server, 根据负载可以共享设施或则使用专用的机器。

集群其它机器作为 DataNode 和 NodeManager.这些是 slaves

## 配置 **hadoop** 【非安全模式】

hadoop 配置文件被分为两类:

1.只读默认配置, 有下列配置文件

core-default.xml, hdfs-default.xml, yarn-default.xml and mapred-default.xml.

2.定制配置, 有下列配置文件

etc/hadoop/core-site.xml, etc/hadoop/hdfs-site.xml, etc/hadoop/yarn-site.xml and etc/hadoop/mapred-site.xml.

另外你可以配置 **hadoop** 脚本, 在 **hadoop** 的 bin 目录下, 通过 `etc/hadoop/hadoop-env.sh` 和 `etc/hadoop/yarn-env.sh`

来指定值。

配置 hadoop 集群需要配置环境变量，Hadoop 守护进程执行以及 Hadoop 守护进程的配置参数。如果没有配置过集群，

可能对这个了解不多，具体参考

[hadoop（2.x）以 hadoop2.2 为例完全分布式最新高可靠安装文档](#)

<http://www.aboutyun.com/thread-7684-1-1.html>

hdfs 守护进程：

NameNode, SecondaryNameNode, 和 DataNode

YARN 守护进程：

ResourceManager, NodeManager, 和 WebAppProxy

如果运行 MapReduce ， MapReduce Job History Server 也会运行。前提需要配置并开启。对于比较大的集群安装，他

们分别运行在不同的客户端。

## 配置守护进程环境变量

管理员使用 `etc/hadoop/hadoop-env.sh` , `etc/hadoop/mapred-env.sh` 和 `etc/hadoop/yarn-env.sh` 脚本来定制

hadoop 守护进程变量。

至少需要指定 `JAVA_HOME` , 每个节点都必须指定。

管理员可以配置单独的守护进程, 使用下面的选项。

Daemon	Environment Variable
NameNode	HADOOP_NAMENODE_OPTS
DataNode	HADOOP_DATANODE_OPTS
Secondary NameNode	HADOOP_SECONDARYNAMENODE_OPTS
ResourceManager	YARN_RESOURCEMANAGER_OPTS
NodeManager	YARN_NODEMANAGER_OPTS
WebAppProxy	YARN_PROXYSERVER_OPTS
Map Reduce Job History Server	HADOOP_JOB_HISTORYSERVER_OPTS

例如：配置 Namenode 使用 parallelGC, 在 hadoop-env.sh 声明如下：export

```
HADOOP_NAMENODE_OPTS="-XX:+UseParallelGC"
```

查看 `etc/hadoop/hadoop-env.sh` 其它例子

其它可定义的有效参数包括

- HADOOP\_PID\_DIR - 存储守护进程 id 的文件目录。
- HADOOP\_LOG\_DIR - 存储守护进程日志文件目录。日志文件自动创建如果不存在。
- HADOOP\_HEAPSIZE / YARN\_HEAPSIZE -heapsize 内存最大配置.如果变量为 1000， heap 是 1000MB. 这是为守护进程配置 heap. 默认值 1000M.

在大多数情况下，应该指定 HADOOP\_PID\_DIR 和 HADOOP\_LOG\_DIR，他们由用户运行 `hadoop` 守护进程所写，否则可能是潜在攻击。

传统配置 HADOOP\_PREFIX 在系统级环境变量配置，例如一个简单的脚本 `/etc/profile.d:`

```
HADOOP_PREFIX=/path/to/hadoop
export HADOOP_PREFIX
```

Daemon	Environment Variable
ResourceManager	YARN_RESOURCEMANAGER_HEAPSIZE
NodeManager	YARN_NODEMANAGER_HEAPSIZE
WebAppProxy	YARN_PROXYSERVER_HEAPSIZE
Map Reduce Job History Server	HADOOP_JOB_HISTORYSERVER_HEAPSIZE

## 配置 **hadoop** 守护进程

本节涉及在给定的配置文件中指定的重要参数：

etc/hadoop/core-site.xml

Parameter	Value	Notes
-----------	-------	-------



fs.defaultFS	NameNode URI	<a href="#">hdfs://host:port/</a>
io.file.buffer.size	131072	读写序列文件缓存大小

- etc/hadoop/hdfs-site.xml

- 配置 NameNode:

[彻底了解 namenode](#)

Parameter	Value	Notes
dfs.namenode.name.dir	本地文件系统存储着命令空间和操作日志	如果含有多个目录，是冗余的【多个目录以逗号隔开】

dfs.hosts / dfs.hosts.exclude	列出排除 DataNodes.	如何需要使用这一功能来控制 datanode 的访问
dfs.blocksize	268435456	大文件系统 HDFS blocksize 256MB
dfs.namenode.handler.count	100	更多 NameNode server 线程来处理来自 datanode 的 RPCs 请求

- 配置 DataNode:

Parameter	Value	Notes
dfs.datanode.data.dir	存储 blocks 的本地路径列表， 用逗号隔开	这是一个逗号分隔的目录列表，数据将被存储在所有被命名的目录 中，通常在不同的设备上。

- etc/hadoop/yarn-site.xml
- 配置 ResourceManager 和 NodeManager:

Parameter	Value	Notes
yarn.acl.enable	true / false	是否启用 ACLs，默认为 false 不启用
yarn.admin.acl	Admin ACL	admin.acl 设置 YARN 的集群管理员，. 值为*表示任何人都可以. 仅指定的可以访问
yarn.log-aggregation-enable	false	配置启用或则禁用日志

- 配置 ResourceManager:

Parameter	Value	Notes
yarn.resourcemanager.address	ResourceManager host:port	如果配置 host:port ，会覆盖在

	为客户端提交 job.	yarn.resourcemanager.hostname 设置 的  hostname
yarn.resourcemanager.scheduler.address	ResourceManager 对  ApplicationMaster 暴露的访  问地址。ApplicationMaster  通过该地址向 RM 申请资  源、释放资源等。	如 果 配 置 host:port , 会 覆 盖 在  yarn.resourcemanager.hostname 设置 的  hostname
yarn.resourcemanager.resource-tracker.address	ResourceManager 对  NodeManager 暴露的地址。  NodeManager 通过该地址向  RM 汇报心跳，领取任务等。	如 果 配 置 host:port , 会 覆 盖 在  yarn.resourcemanager.hostname 设置 的  hostname
yarn.resourcemanager.admin.address	ResourceManager 对管理员  暴露的访问地址。管理员通  过该地址向 RM 发送管理命	如 果 配 置 host:port , 会 覆 盖 在  yarn.resourcemanager.hostname 设置 的  hostname

	令等	
yarn.resourcemanager.webapp.address	ResourceManager web-ui host:port （ResourceManager 访问端口）	如果配置 host:port ，会覆盖在 yarn.resourcemanager.hostname 设置的 hostname
yarn.resourcemanager.hostname	ResourceManager 客户端.	host Single hostname that can be set in place of setting all yarn.resourcemanager*address resources. Results in default ports for ResourceManager components.
yarn.resourcemanager.scheduler.class	ResourceManager 调度类 .	CapacityScheduler (recommended), FairScheduler (also recommended), or FifoScheduler
yarn.scheduler.minimum-allocation-mb	单个 container 可申请的最小 内存资源量。比如设置为	In MBs

	<p>1024，则运行 MapRedce 作</p> <p>业时，每个 Task 最少可申请</p> <p>1024MB 内存</p>	
yarn.scheduler.maximum-allocation-mb	<p>单个 container 可申请的最大</p> <p>内存资源量。比如设置为</p> <p>3072，则运行 MapRedce 作</p> <p>业时，每个 Task 最多可申请</p> <p>3072MB 内存。</p>	In MBs
yarn.resourcemanager.nodes.include-path  /yarn.resourcemanager.nodes.exclude-path	NodeManager 黑白名单。	<p>NodeManager 黑白名单。如果发现若干个</p> <p>NodeManager 存在问题，比如故障率很高，</p> <p>任务运行失败率高，则可以将之加入黑名单</p> <p>中。注意，这两个配置参数可以动态生效。（调</p> <p>用一个 refresh 命令即可）</p> <p>默认值：""</p>

- 配置 NodeManager:

Parameter	Value	Notes
yarn.nodemanager.resource.memory-mb	NodeManager 总的可用物理内存。	定义了资源总量的 nodemanager 可用运行 containers
yarn.nodemanager.vmem-pmem-ratio	每个任务的虚拟内存的使用可能会超过其物理内存每使用 1MB 物理内存，最多可用的虚拟内存数。	每个任务的虚拟内存的使用可能会超过其物理内存的限制，这个比例。通过对 nodemanager 任务使用的虚拟内存总量可能超过物理内存使用的比率。
yarn.nodemanager.local-dirs	中间结果存放位置	这个参数通常会配置多个目录，分摊磁盘 IO 负载。
yarn.nodemanager.log-dirs	日志存放路径	这个参数通常会配置多个目录，分摊磁盘 IO 负载。
yarn.nodemanager.log.retain-seconds	10800	NodeManager 上日志最多存放时间（不启用日志聚集功能时有效）。

yarn.nodemanager.remote-app-log-dir	/logs	当应用程序运行结束后，日志被转移到的 HDFS 目录  (启用日志聚集功能时有效)。
yarn.nodemanager.remote-app-log-dir-suffix	logs	远程日志目录子目录名称 (启用日志聚集功能时有效)。
yarn.nodemanager.aux-services	mapreduce_shuffle	NodeManager 上运行的附属服务。需配置成 mapreduce_shuffle，才可运行 MapReduce 程序。

- 配置 History Server (Needs to be moved elsewhere):

Parameter	Value	Notes
yarn.log-aggregation.retain-seconds	-1	参数解释：在 HDFS 上聚集的日志最多保存多长时间。  默认值：-1



yarn.log-aggregation.retain-check-interval-seconds	-1	<p>参数解释：多长时间检查一次日志，并将满足条件的删除，如果是 0 或者负数，则为上一个值的 1/10。</p> <p>默认值：-1</p>
--	----	---

- etc/hadoop/mapred-site.xml
- 配置 MapReduce Applications:

mapreduce.map.memory.mb

mapreduce.reduce.memory.mb

说明：这两个参数指定用于 MapReduce 的两个任务（Map and Reduce task）的内存大小，其值应该在 RM 中的最大最小 container 之间。如果没有配置则通过如下简单公式获得：

$\max(\text{MIN\_CONTAINER\_SIZE}, (\text{Total Available RAM}) / \text{containers}))$

一般的 reduce 应该是 map 的 2 倍。注：这两个值可以在应用启动时通过参数改变；

mapreduce.map.java.opts

mapreduce.reduce.java.opts

说明：这两个参主要是为需要运行 JVM 程序（java、scala 等）准备的，通过这两个设置可以向 JVM 中传递参数的，与内存有关的是，-Xmx，-Xms 等选项。此数值大小，应该在 AM 中的 map.mb 和 reduce.mb 之间。

Parameter	Value	Notes
-----------	-------	-------

mapreduce.framework.name	yarn	执行框架设置为 Hadoop YARN.
mapreduce.map.memory.mb	1536	maps 资源限制
mapreduce.map.java.opts	-Xmx1024M	maps 的 child jvms heap-size
mapreduce.reduce.memory.mb	3072	reduces 资源限制
mapreduce.reduce.java.opts	-Xmx2560M	reduces 的 child jvms heap-size
mapreduce.task.io.sort.mb	512	任务内部排序缓冲区大小
mapreduce.task.io.sort.factor	100	排序文件的时候一次同时最多可并流的个数，这里设置 100。
mapreduce.reduce.shuffle.parallelcopies	50	reuduce shuffle 阶段并行传输数据的数量。

- 配置 MapReduce JobHistory Server:

Parameter	Value	Notes
mapreduce.jobhistory.address	MapReduce JobHistory Server 地址  【host:port】	默认端口号 10020.
mapreduce.jobhistory.webapp.address	MapReduce JobHistory Server Web UI 地址  【 host:port】	默认端口号 19888.
mapreduce.jobhistory.intermediate-done-dir	/mr-history/tmp	MapReduce 作业产生的日志存放位置。
mapreduce.jobhistory.done-dir	/mr-history/done	MR JobHistory Server 管理的日志的存放位置。

原文:

hadoop 【2.7.1】【多节点】集群配置【必知配置知识 1】

<http://www.aboutyun.com/thread-15609-1-1.html>

## hadoop 【2.7.1】【多节点】集群配置

### 【必知配置知识 2】

问题导读

- 1.hadoop daemonlog 管理员命令的作用是什么?
- 2.hadoop 如何运行一个类, 如何运行一个 jar 包?
- 3.hadoop archive 的作用是什么?



## 概述

hadoop 命令被 bin/hadoop 脚本调用。运行 hadoop 脚本不带任何命令将打印命令相关描述。

Usage: hadoop [--config confdir] [--loglevel loglevel] [COMMAND] [GENERIC\_OPTIONS] [COMMAND\_OPTIONS]

FIELD	Description
--config confdir	覆盖默认配置文件目录。默认路径\${HADOOP_HOME}/conf.

--loglevel loglevel	覆盖日志级别。有效日志级别为 FATAL, ERROR, WARN, INFO, DEBUG, 和 TRACE。默认是 INFO。
GENERIC_OPTIONS	多项命令共同支持的选项
COMMAND_OPTIONS	hadoop 子项目文档描述了 hadoop 命令的选项。hdfs 和 YARN 在其它文档那个

## Hadoop 通用选项

Many subcommands honor a common set of configuration options to alter their behavior:

GENERIC_OPTION	Description
-archives <comma separated list of archives>	Specify comma separated archives to be unarchived on the compute machines. Applies only to job.
-conf <configuration file>	指定应用程序配置文件
-D <property>=<value>	使用一个给定的属性值
-files <comma separated list of files>	指定文件复制到 mapredue 集群。仅适用于 job.
-jt <local> or <resourcemanager:port>	指定 ResourceManager. 仅适用于 job.
-libjars <comma separated list of jars>	指定 jar 文件包括所在 classpath. 仅适用于 job.

## Hadoop 通用命令

所有这些命令都是从 Hadoop 的 shell 命令执行。他们被分成用户命令和管理员命令【 User Commands 和 Administration Commands.】

### 用户命令

用于 Hadoop 集群用户命令。

#### archive

创建一个 Hadoop 档案，更多信息查看 [Hadoop Archives Guide](#).

#### checknative

用法: `hadoop checknative [-a] [-h]`

COMMAND_OPTION	Description
-a	检查所有库
-h	打印帮助

这个命令检查 Hadoop 本地代码的可用性。更多信息查看 [#NativeLibraries.html](#)。默认，此命令只检查 `libhadoop` 的可用性。

### classpath

用法: `hadoop classpath [--glob |--jar <path>] [-h] [--help]`

COMMAND_OPTION	Description
--glob	通配符
--jar <i>path</i>	jar 路径
-h, --help	打印帮助

打印 `classpath` 需要 Hadoop 的 `jar` 和需要的库类路径。如果不带参数调用，然后打印的命令脚本设置 `classpath`。这可能包含通配符

### credential

用法: `hadoop credential <subcommand> [options]`

COMMAND_OPTION	Description
create alias [-provider provider-path]	提示证书被存储为指定别名的用户。如果没有 <code>-provider</code> 选项的话，那么将会默认使用 <code>core-site.xml</code> 文件中 <code>hadoop.security.credential.provider.path</code> 项对应的值。
delete alias [-provider provider-path] [-f]	删除与所提供的别名对应的证书文件。如果没有 <code>-provider</code> 选项的话，那么将会默认使用 <code>core-site.xml</code> 文件中 <code>hadoop.security.credential.provider.path</code> 项对应的值。这项操作需要通过用户的确认，除非使用了 <code>-f</code> 选项。
list [-provider provider-path]	列出所有的证书别名。如果没有 <code>-provider</code> 选项的话，那么将会默认使用 <code>core-site.xml</code> 文件中 <code>hadoop.security.credential.provider.path</code> 项对应的值。

该命令在凭证提供者内部管理凭证（credentials），密码（passwords）和秘密（secrets）。

Hadoop 的 `CredentialProvider` API 支持应用程序拆分，并且要求拆分后的应用如何储存所需的密码（passwords）和秘密（secrets）。为了指明一个 `Provider` 的位置和类型，需要在 `core-site.xml` 添加 `hadoop.security.credential.provider.path` 配置项，或者通过指令中 `-provider` 命令选项进行设置。`Provider` 路径是一串以逗号分割的 URL 字符串。这些字符串会说明 `Provider` 的类型和位置，举个例子：

```
user:///jceks://file/tmp/test.jceks,jceks://hdfs@nn1.example.com/my/path/test.jceks
```

指示当前用户的凭证，需要通过 `User Provider` 咨询。存储在本地文件系统的文件 `/tmp/test.jceks` 是一个 `Java Keystore Provider`，相应的存储在 `hdfs` 上的文件 `nn1.example.com/my/path/test.jcek` 也是一个 `Java Keystore Provider`。

当使用 `credential` 命令时，它通常要提供密码（password）或秘密（secret）给一个特定的凭证存储 `provider`。为了清晰的表明要显示使用哪个 `provider` 存储，可以在命令中使用 `-provider` 选项。否则，给定多个 `provider` 的时候，则使用的哥非持久的 `provider`，这可能不是你预期的

例如：

```
hadoop credential list -provider jceks:///file/tmp/test.jceks
```

### **distcp**

递归的拷贝文件或者目录。更多信息查看 [Hadoop DistCp Guide](#)。

### **fs**

这个命令在文档 [File System Shell Guide](#) 。 和 `hdfs` 脚本的 `dfs` 类似

### **jar**

用法: `hadoop jar <jar> [mainClass] args...`

运行一个 `jar` 文件使用 `yarn` 启动 `yarn` 应用程序

### **key**

通过 `KeyProvider` 管理密钥

### **trace**

查看和修改 Hadoop 跟踪（tracing）设置。查看：[跟踪（tracing）指南](#)。

[Tracing Guide](#)。

### **version**

用法: `hadoop version`

打印版本

### **CLASSNAME**

用法: `hadoop CLASSNAME`

运行一个类

## 管理员命令

集群管理员命令

### **daemonlog**

用法:

`hadoop daemonlog -getlevel <host:httpport> <classname>`

`hadoop daemonlog -setlevel <host:httpport> <classname> <level>`

COMMAND_OPTION	Description
<code>-getlevel host:httpportclassname</code>	打印运行在<host:port>的守护进程的日志级别。这个命令内部会连接 <code>http://&lt;host:port&gt;/logLevel?log=&lt;name&gt;</code>
<code>-setlevel host:httpportclassname level</code>	设置运行在<host:port>的守护进程的日志级别。这个命令内部会连接 <code>http://&lt;host:port&gt;/logLevel?log=&lt;name&gt;</code>

设置或获取指定后台进程的日志级别

Example: `$ bin/hadoop daemonlog -setlevel 127.0.0.1:50070 org.apache.hadoop.hdfs.server.namenode.NameNode`

DEBUG

原文链接:

hadoop【2.7.1】【多节点】集群配置【必知配置知识2】

<http://www.aboutyun.com/thread-15650-1-1.html>



# Hadoop 【2.7.1】初级入门之命令指南

问题导读

**1.hadoop daemonlog** 管理员命令的作用是什么？

**2.hadoop** 如何运行一个类，如何运行一个 **jar** 包？

**3.hadoop archive** 的作用是什么？



## 概述

hadoop 命令被 bin/hadoop 脚本调用。运行 hadoop 脚本不带任何命令将打印命令相关描述。

Usage: hadoop [--config confdir] [--loglevel loglevel] [COMMAND] [GENERIC\_OPTIONS] [COMMAND\_OPTIONS]

FIELD	Description
--config confdir	覆盖默认配置文件目录。默认路径\${HADOOP_HOME}/conf.
--loglevel loglevel	覆盖日志级别。有效日志级别为 FATAL, ERROR, WARN, INFO, DEBUG, 和 TRACE. 默认是 INFO.
GENERIC_OPTIONS	多项命令共同支持的选项
COMMAND_OPTIONS	hadoop 子项目文档描述了 hadoop 命令的选项。hdfs 和 YARN 在其它文档那个

## Hadoop 通用选项

Many subcommands honor a common set of configuration options to alter their behavior:

GENERIC_OPTION	Description
-archives <comma separated list of archives>	Specify comma separated archives to be unarchived on the compute machines. Applies only to job.
-conf <configuration file>	指定应用程序配置文件
-D <property>=<value>	使用一个给定的属性值

-files <comma separated list of files>	指定文件复制到 mapredue 集群。仅适用于 job.
-jt <local> or <resourcemanager:port>	指定 ResourceManager. 仅适用于 job.
-libjars <comma seperated list of jars>	指定 jar 文件包括所在 classpath. 仅适用于 job.

## Hadoop 通用命令

所有这些命令都是从 Hadoop 的 shell 命令执行。他们被分成用户命令和管理员命令【 User Commands 和 Administration Commands.】

### 用户命令

用于 Hadoop 集群用户命令。

#### archive

创建一个 Hadoop 档案，更多信息查看 [Hadoop Archives Guide](#).

#### checknative

用法: `hadoop checknative [-a] [-h]`

COMMAND_OPTION	Description
-a	检查所有库
-h	打印帮助

这个命令检查 Hadoop 本地代码的可用性。更多信息查看 [#NativeLibraries.html](#)。默认，此命令只检查 libhadoop 的可用性。

#### classpath

用法: `hadoop classpath [--glob] [--jar <path>] [-h] [--help]`

COMMAND_OPTION	Description
----------------	-------------

<code>--glob</code>	通配符
<code>--jar <i>path</i></code>	jar 路径
<code>-h, --help</code>	打印帮助

打印 `classpath` 需要 Hadoop 的 jar 和需要的库类路径。如果不带参数调用，然后打印的命令脚本设置 `classpath`。这可能包含通配符

## credential

用法: `hadoop credential <subcommand> [options]`

COMMAND_OPTION	Description
<code>create alias [-provider provider-path]</code>	提示证书被存储为指定别名的用户。如果没有 <code>-provider</code> 选项的话，那么将会默认使用 <code>core-site.xml</code> 文件中 <code>hadoop.security.credential.provider.path</code> 项对应的值。
<code>delete alias [-provider provider-path] [-f]</code>	删除与所提供的别名对应的证书文件。如果没有 <code>-provider</code> 选项的话，那么将会默认使用 <code>core-site.xml</code> 文件中 <code>hadoop.security.credential.provider.path</code> 项对应的值。这项操作需要通过用户的确认，除非使用了 <code>-f</code> 选项。
<code>list [-provider provider-path]</code>	列出所有的证书别名。如果没有 <code>-provider</code> 选项的话，那么将会默认使用 <code>core-site.xml</code> 文件中 <code>hadoop.security.credential.provider.path</code> 项对应的值。

该命令在凭证提供者内部管理凭证（credentials），密码（passwords）和秘密（secrets）。

Hadoop 的 `CredentialProvider` API 支持应用程序拆分，并且要求拆分后的应用如何储存所需的密码（passwords）和秘密（secrets）。为了指明一个 `Provider` 的位置和类型，需要在 `core-site.xml` 添加 `hadoop.security.credential.provider.path` 配置项，或者通过指令中 `-provider` 命令选项进行设置。`Provider` 路径是一串以逗号分割的 URL 字符串。这些字符串会说明 `Provider` 的类型和位置，举个例子：

```
user:///jceks://file/tmp/test.jceks,jceks://hdfs@nn1.example.com/my/path/test.jceks
```

指示当前用户的凭证，需要通过 `User Provider` 咨询。存储在本地文件系统的文件 `/tmp/test.jceks` 是一个 `Java Keystore Provider`，相应的存储在 `hdfs` 上的文件 `nn1.example.com/my/path/test.jcek` 也是一个 `Java Keystore Provider`。

当使用 `credential` 命令时，它通常要提供密码（password）或秘密（secret）给一个特定的凭证存储 `provider`。为了清晰的表明要显示使用哪个 `provider` 存储，可以在命令中使用 `-provider` 选项。否则，给定多个 `provider` 的时候，则使用的哥非持久的 `provider`，这可能不是你预期的

例如：

```
hadoop credential list -provider jceks:///file/tmp/test.jceks
```

## distcp

递归的拷贝文件或者目录。更多信息查看 [Hadoop DistCp Guide](#).

### **fs**

这个命令在文档 [File System Shell Guide](#) 。 和 hdfs 脚本的 dfs 类似

### **jar**

用法: `hadoop jar <jar> [mainClass] args...`

运行一个 jar 文件使用 yarn 启动 yarn 应用程序

### **key**

通过 KeyProvider 管理密钥

### **trace**

查看和修改 Hadoop 跟踪（tracing）设置。查看：[跟踪（tracing）指南](#)。

[Tracing Guide](#).

### **version**

用法: `hadoop version`

打印版本

### **CLASSNAME**

用法: `hadoop CLASSNAME`

运行一个类

## 管理员命令

集群管理员命令

### **daemonlog**

用法:

`hadoop daemonlog -getlevel <host:httpport> <classname>`

`hadoop daemonlog -setlevel <host:httpport> <classname> <level>`

COMMAND_OPTION	Description
<code>-getlevel host:httpportclassname</code>	打印运行在<host:port>的守护进程的日志级别。这个命令内部会连接 <code>http://&lt;host:port&gt;/logLevel?log=&lt;name&gt;</code>
<code>-setlevel host:httpportclassname level</code>	设置运行在<host:port>的守护进程的日志级别。这个命令内部会连接 <code>http://&lt;host:port&gt;/logLevel?log=&lt;name&gt;</code>

设置或获取指定后台进程的日志级别

Example: `$ bin/hadoop daemonlog -setlevel 127.0.0.1:50070 org.apache.hadoop.hdfs.server.namenode.NameNode`

DEBUG

## Hadoop 【2.7.1】初级入门之命令：

### 文件系统 shell1

问题导读

- 1.Hadoop 文件系统 shell 与 Linux shell 有哪些相似之处？
- 2.如何改变文件所属组？
- 3.如何改变 hdfs 的文件权限？
- 4.如何查找 hdfs 文件，并且不区分大小写？



## 概述

文件系统 (FS) shell 包括各种类似的命令直接与 Hadoop Distributed File System (HDFS)交互。hadoop 也支持其它文件系统,比如 Local FS, HFTP FS, S3 FS, 和 其它的。FS shell 被下面调用：

`bin/hadoop fs <args>`

所有的 FS shell 命令带有 URIs 路径参数。The URI 格式是://authority/path。对 HDFS 文件系统，scheme 是 hdfs。其中 scheme 和 authority 参数都是可选的

如果没有指定，在文件中使用默认 scheme。一个 hdfs 文件或则目录比如 /parent/child，可以是

hdfs://namenodehost/parent/child 或则简化为/parent/child（默认配置设置成指向 hdfs://namenodehost）。大多数 FS shell 命令对应 Unix 命令。每个命令都有不同的描述。将错误信息发送到标准错误输出和输出发送到 stdout。

### **appendToFile【添加文件】**

用法: `hadoop fs -appendToFile <localsrc> ... <dst>`添加单个 src,或则多个 srcs 从本地文件系统到目标文件系统。从标准输入读取并追加到目标文件系统。

- `hadoop fs -appendToFile localfile /user/hadoop/hadoopfile`
- `hadoop fs -appendToFile localfile1 localfile2 /user/hadoop/hadoopfile`
- `hadoop fs -appendToFile localfile hdfs://nn.example.com/hadoop/hadoopfile`
- `hadoop fs -appendToFile - hdfs://nn.example.com/hadoop/hadoopfile` Reads the input from stdin.

返回代码:

返回 0 成功返回 1 错误

### **cat**

用法: `hadoop fs -cat URI [URI ...]`

将路径指定文件的内容输出到 stdout

例子:

- `hadoop fs -cat hdfs://nn1.example.com/file1 hdfs://nn2.example.com/file2`
- `hadoop fs -cat file:///file3 /user/hadoop/file4`

返回代码:

返回 0 成功返回 1 错误

### **checksum**

用法: `hadoop fs -checksum URI`

返回 checksum 文件信息

例子:

- `hadoop fs -checksum hdfs://nn1.example.com/file1`
- `hadoop fs -checksum file:///etc/hosts`

### **chgrp**

用法: `hadoop fs -chgrp [-R] GROUP URI [URI ...]`

改变文件所属组，必须是文件所有者或则超级用户，更多信息在 [Permissions Guide](#)。

选项

- 使用 `-R` 将使改变在目录结构下递归进行

### **chmod**

用法: `hadoop fs -chmod [-R] <MODE[,MODE]... | OCTALMODE> URI [URI ...]`

更改文件的权限，使用 `-R` 将使改变在目录结构下递归进行。必须是文件所有者或则超级用户，更多信息在 [Permissions Guide](#)。

选项

- 使用 `-R` 将使改变在目录结构下递归进行。

### **chown**

用法: `hadoop fs -chown [-R] [OWNER][:[GROUP]] URI [URI ]`

更改文件的所有者，使用 `-R` 将使改变在目录结构下递归进行。必须是文件所有者或则超级用户，更多信息在 [Permissions Guide](#)。

选项

- 使用 `-R` 将使改变在目录结构下递归进行。

### **copyFromLocal**

用法: `hadoop fs -copyFromLocal <localsrc> URI`

类似 `put` 命令，需要指出的是这个限制是本地文件

选项:

- `-f` 选项会重写已存在的目标文件

### **copyToLocal**

用法: `hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>`

与 `get` 命令类似，除了限定目标路径是一个本地文件外。

### **count**

用法: `hadoop fs -count [-q] [-h] [-v] <paths>` 统计目录个数，文件和目录下文件的大小。输出列: `DIR_COUNT`, `FILE_COUNT`, `CONTENT_SIZE`, `PATHNAME`

【目录个数，文件个数，总大小，路径名称】

输出列带有 `-count -q` 是: `QUOTA`, `REMAINING_QUOTA`, `SPACE_QUOTA`, `REMAINING_SPACE_QUOTA`, `DIR_COUNT`, `FILE_COUNT`, `CONTENT_SIZE`, `PATHNAME`

【配置，其余指标，空间配额，剩余空间定额，目录个数，文件个数，总大小，路径名称】

The `-h` 选项，`size` 可读模式。

The `-v` 选项显示一个标题行。

Example:

- `hadoop fs -count hdfs://nn1.example.com/file1 hdfs://nn2.example.com/file2`
- `hadoop fs -count -q hdfs://nn1.example.com/file1`
- `hadoop fs -count -q -h hdfs://nn1.example.com/file1`
- `hdfs dfs -count -q -h -v hdfs://nn1.example.com/file1`

返回代码:

返回 0 成功返回 1 错误

```
aboutyun123@aboutyun:~$ hadoop fs -count -q -h /user/aboutyun123/input/
      none      inf      none      inf      1      30      76.7 K /user/aboutyun123/input
aboutyun123@aboutyun:~$ hadoop fs -count -q -h /user/aboutyun123
      none      inf      none      inf      3      32      77.0 K /user/aboutyun123
aboutyun123@aboutyun:~$ hadoop fs -ls /user/aboutyun123
Found 2 items
drwxr-xr-x - aboutyun123 supergroup      0 2015-07-08 00:31 /user/aboutyun123/input
drwxr-xr-x - aboutyun123 supergroup      0 2015-07-08 00:39 /user/aboutyun123/output
aboutyun123@aboutyun:~$ hadoop fs -ls /user
Found 1 items
drwxr-xr-x - aboutyun123 supergroup      0 2015-07-08 00:39 /user/aboutyun123
aboutyun123@aboutyun:~$ hadoop fs -count -q -h /user
      none      inf      none      inf      4      32      77.0 K /user
aboutyun123@aboutyun:~$
```

## cp

用法: `hadoop fs -cp [-f] [-p | -p[topax]] URI [URI ...] <dest>` 复制文件, 这个命令允许复制多个文件到一个目录。

'raw.\*' 命名空间扩展属性被保留

- (1) 源文件和目标文件支持他们 (仅 hdfs)
- (2) 所有的源文件和目标文件路径在 `/.reserved/raw` 目录结构下。

决定是否使用 `raw.*` 命名空间扩展属性依赖于 `-P` 选项

选项:

- `-f` 选项如果文件已经存在将会被重写。
- `-p` 选项保存文件属性 [topx] (timestamps, ownership, permission, ACL, XAttr). 如果指定 `-p` 没有参数, 保存 timestamps, ownership, permission. 如果指定 `-pa`, 保留权限 因为 ACL 是一个权限的超级组。确定是否保存 raw 命名空间属性取决于是否使用 `-p` 决定

例子:

- `hadoop fs -cp /user/hadoop/file1 /user/hadoop/file2`
- `hadoop fs -cp /user/hadoop/file1 /user/hadoop/file2 /user/hadoop/dir`

返回代码:

返回 0 成功返回 1 错误

## createSnapshot

查看 HDFS Snapshots Guide.



## deleteSnapshot

查看 HDFS Snapshots Guide.

## df【查看还剩多少 hdfs 空间】

用法: `hadoop fs -df [-h] URI [URI ...]`

显示剩余空间

选项:

- `-h` 选项会让人更加易读 (比如 `64.0m` 代替 `67108864`)

Example:

- `hadoop dfs -df /user/hadoop/dir1`

```
aboutyun123@aboutyun:~$ hadoop fs -df /
Filesystem              Size        Used      Available   Use%
hdfs://localhost:9000 20334034944 1019904 12705988608    0%
aboutyun123@aboutyun:~$ hadoop fs -df -h /
Filesystem              Size        Used      Available   Use%
hdfs://localhost:9000 18.9 G    996 K      11.8 G      0%
```

## du

用法: `hadoop fs -du [-s] [-h] URI [URI ...]`显示给定目录的文件大小及包含的目录, 如果只有文件只显示文件的大小

选项:

- `-s` 选项汇总文件的长度, 而不是现实单个文件.
- `-h` 选项显示格式更加易读 (例如 `64.0m` 代替 `67108864`)

例子:

- `hadoop fs -du /user/hadoop/dir1 /user/hadoop/file1 hdfs://nn.example.com/user/hadoop/dir1`

返回代码:

返回 `0` 成功返回 `1` 错误

```
aboutyun123@aboutyun:~$ hadoop fs -du -s /
702582 /
aboutyun123@aboutyun:~$ hadoop fs -du -s /user
78822 /user
aboutyun123@aboutyun:~$ hadoop fs -du -s -h /user
77.0 K /user
```

## dus

用法: `hadoop fs -dus <args>`

显示统计文件长度

注意:这个命令已被启用, `hadoop fs -du -s` 即可

## expunge

用法: `hadoop fs -expunge`

清空垃圾回收站, 涉及 [HDFS Architecture Guide](#) 更多信息查看回收站特点

## find

用法: `hadoop fs -find <path> ... <expression> ...` 查找与指定表达式匹配的所有文件, 并将选定的操作应用于它们。如果没有指定路径, 则默认查找当前目录。如果没有指定表达式默认 `-print`

下面主要表达式:

- `-name` 模式
- `-iname` 模式

如果

值为 `TRUE` 如果文件基本名匹配模式使用标准的文件系统组合。如果使用 `-iname` 匹配不区分大小写。

- `-print`
- `-print0` Always

值为 `TRUE`. 当前路径被写至标准输出。如果使用 `-print0` 表达式, ASCII NULL 字符是追加的。

下面操作:

- `expression -a expression`
- `expression -and expression`
- `expression expression`

`and` 运算符连接两个表达式, 如果两个字表达式返回 `true`, 则返回 `true`. 由两个表达式的并置暗示, 所以不需要明确指定。

如果第一个失败, 则不会应用第二个表达式。

例子:

`hadoop fs -find / -name test -print`

返回代码:

返回 0 成功返回 1 错误

```
aboutyun123@aboutyun:~$ hadoop fs -find /user/aboutyun123/input -iname yarn* -print0
/user/aboutyun123/input/yarn-env.cmd/user/aboutyun123/input/yarn-env.sh/user/aboutyun123/input/yarn-site.xmlaboutyun123@aboutyun:~$ h
aboutyun123@aboutyun:~$ hadoop fs -find /user/aboutyun123/input -iname yarn* -print
/user/aboutyun123/input/yarn-env.cmd
/user/aboutyun123/input/yarn-env.sh
/user/aboutyun123/input/yarn-site.xml
```

## get

用法: `hadoop fs -get [-ignorecrc] [-crc] <src> <localdst>` 复制文件到本地文件。

复制文件到本地文件系统。【CRC 校验失败的文件复制带有 `-ignorecrc` 选项（如翻译有误欢迎指正）】

Files that fail the CRC check may be copied with the `-ignorecrc` option.

文件 CRC 可以复制使用 CRC 选项。

例子：

- `hadoop fs -get /user/hadoop/file localfile`
- `hadoop fs -get hdfs://nn.example.com/user/hadoop/file localfile`

返回代码：

返回 0 成功返回 1 错误

## Hadoop【2.7.1】初级入门之命令：文件系统 shell2

问题导读

**1.改变 hdfs 文件的权限，需要修改哪个配置文件？**

**2.获取一个文件的或则目录的权限，哪个命令可以实现？**

**3.哪个命令可以实现设置访问控制列表（ACL）的文件和目录？**



接上篇：

**Hadoop【2.7.1】初级入门之命令：文件系统 shell1**

<http://www.aboutyun.com/thread-15824-1-1.html>

## **getfacl**

用法: `hadoop fs -getfacl [-R] <path>`

显示访问控制列表（ACL）的文件和目录。如果一个目录有默认的 ACL，`getfacl` 也显示默认的 ACL。

选项：

- `-R`: 递归目录和列出所有文件的 ACLs。
- `path`: 文件或目录列表。

例子：

- `hadoop fs -getfacl /file`
- `hadoop fs -getfacl -R /dir`

返回代码:

返回 0 成功返回 非 0 错误

```
aboutyun123@aboutyun:/usr/hadoop-2.7.0/sbin$ hadoop fs -getfacl /test
# file: /test
# owner: aboutyun123
# group: supergroup
user::rw-
group::r--
other::r--
```

### getfattr

用法: `hadoop fs -getfattr [-R] -n name | -d [-e en] <path>`

显示文件和目录扩展属性名字和值[如果有的话]

选项:

- `-R`: 递归显示文件和目录属性.
- `-n name`: Dump the named extended attribute value.
- `-d`: Dump all extended attribute values associated with pathname.
- `-e encoding`: 检索后的值进行编码。 有效的编码是 "text", "hex", and "base64". 值编码作为文本字符串是用双引号括起来的 (""),

值编码作为 16 进制和 64 进制, 前缀分别为 0x 和 0s

- `path`: 文件或则目录

例子:

- `hadoop fs -getfattr -d /file`
- `hadoop fs -getfattr -R -n user.myAttr /dir`

返回代码:

返回 0 成功返回 非 0 错误

### getmerge

用法: `hadoop fs -getmerge <src> <localdst> [addnl]`

源目录和目标文件作为输入和连接文件合并到本地目标文件。`addnl` 选项可以设置在文件末尾添加一个换行符。

## help

用法: `hadoop fs -help`

返回使用输出。

## ls

用法: `hadoop fs -ls [-d] [-h] [-R] [-t] [-S] [-r] [-u] <args>`

选项:

- `-d`: 目录被列为纯文件。
- `-h`: 文件格式变为易读 (例如 `67108864` 显示 `64.0m`)。
- `-R`: 递归子目录列表中。
- `-t`: 按修改时间排序输出 (最近一次)。
- `-S`: 按文件大小排序输出。
- `-r`: 倒序排序
- `-u`: 对使用时间显示和排序而不是修改时间

文件返回下面信息:

permissions	number_of_replicas	userid	groupid	filesize	modification_date	modification_time	filename
权限	副本数	用户名	所属组	文件大小	修改日期	修改时间	文件名

目录返回下面信息

权限	用户	所属组	修改日期	修改时间	目录名
----	----	-----	------	------	-----

目录内的文件默认按文件名排序

例子:

- `hadoop fs -ls /user/hadoop/file1`

退出代码:

返回 0 成功, 返回-1 错误

### **lsr**

用法: `hadoop fs -lsr <args>`

ls 递归

注意: 这个命令被启用的, 替换为 `hadoop fs -ls -R`

### **mkdir**

用法: `hadoop fs -mkdir [-p] <paths>`

以 URI 的路径作为参数并创建目录。

选项:

- `-p` 选项与 Linux `-p` 功能一样, 会创建父目录

例子:

- `hadoop fs -mkdir /user/hadoop/dir1 /user/hadoop/dir2`
- `hadoop fs -mkdir hdfs://nn1.example.com/user/hadoop/dir hdfs://nn2.example.com/user/hadoop/dir`

退出代码:

返回 0 成功, -1 错误

### **moveFromLocal**

用法: `hadoop fs -moveFromLocal <localsrc> <dst>`

类似 `put` 命令, 但是它是本地源文件复制后被删除

### **moveToLocal**

用法: `hadoop fs -moveToLocal [-crc] <src> <dst>`

显示 “Not implemented yet” 消息

### **mv**

用法: `hadoop fs -mv URI [URI ...] <dest>` 移动文件, 这个命令允许移动多个文件到某个目录

例子:

- `hadoop fs -mv /user/hadoop/file1 /user/hadoop/file2`
- `hadoop fs -mv hdfs://nn.example.com/file1 hdfs://nn.example.com/file2 hdfs://nn.example.com/file3 hdfs://nn.example.com/dir1`

退出代码:

返回 0 成功, -1 错误

### put

用法: `hadoop fs -put <localsrc> ... <dst>`

复制单个或则多个源文件到目标系统文件。从 `stdin` 读取输入并写入到目标文件系统。

- `hadoop fs -put localfile /user/hadoop/hadoopfile`
- `hadoop fs -put localfile1 localfile2 /user/hadoop/hadoopdir`
- `hadoop fs -put localfile hdfs://nn.example.com/hadoop/hadoopfile`
- `hadoop fs -put - hdfs://nn.example.com/hadoop/hadoopfile` 从 `stdin` 读取输入。

退出代码:

返回 0 成功, -1 错误

### renameSnapshot

See HDFS Snapshots Guide.

### rm

用法: `hadoop fs -rm [-f] [-r | -R] [-skipTrash] URI [URI ...]`

删除指定的参数文件。

选项:

- `-f` 选项 如果该文件不存在, 则该选项将不显示诊断信息或修改退出状态以反映错误。
- `-R` 选项递归删除目录下任何内容
- `-r` 与 `-R` 效果一样
- `-skipTrash` 选项绕过垃圾回收器, 如果启用, 将会立即删除指定文件。这是非常有用对于超过配额的目录

例子:

- `hadoop fs -rm hdfs://nn.example.com/file /user/hadoop/emptydir`

退出代码:

返回 0 成功, -1 错误

### rmdir

用法: `hadoop fs -rmdir [--ignore-fail-on-non-empty] URI [URI ...]`

删除目录



选项:

- `--ignore-fail-on-non-empty`: 当使用通配符, 一个目录还包含文件, 不会失败.

例子:

- `hadoop fs -rmdir /user/hadoop/emptydir`

## **rmdir**

用法: `hadoop fs -rmdir [-skipTrash] URI [URI ...]`

递归删除

说明:这个命令被弃用了, 而是使用 `hadoop fs -rm -r`

## **setfacl**

用法: `hadoop fs -setfacl [-R] [-b |-k -m |-x <acl_spec> <path>] [--set <acl_spec> <path>]`

设置访问控制列表 (ACL) 的文件和目录。

选项:

- `-b`: 移除所有除了基本的 ACL 条目。用户、组和其他的条目被保留为与权限位的兼容性。
- `-k`: 删除默认的 ACL。
- `-R`: 递归应用于所有文件和目录的操作。
- `-m`: 修改 ACL。新的项目添加到 ACL, 并保留现有的条目。
- `-x`: 删除指定的 ACL 条目。其他保留 ACL 条目。
- `--set`: 完全替换 ACL, 丢弃所有现有的条目。`acl_spec` 必须包括用户, 组, 和其他有权限位的兼容性。
- `acl_spec`: 逗号分隔的 ACL 条目列表。
- `path`: 修改文件或目录。

例子:

- `hadoop fs -setfacl -m user:hadoop:rw- /file`
- `hadoop fs -setfacl -x user:hadoop /file`
- `hadoop fs -setfacl -b /file`
- `hadoop fs -setfacl -k /dir`
- `hadoop fs -setfacl --set user::rw-,user:hadoop:rw-,group::r--,other::r-- /file`
- `hadoop fs -setfacl -R -m user:hadoop:r-x /dir`
- `hadoop fs -setfacl -m default:user:hadoop:r-x /dir`

退出代码:

返回 0 成功, 非 0 错误

以上需要开启 acl:

开启 acls, 配置 hdfs-site.xml

```
vi etc/hadoop/hdfs-site.xml
```

```
<property>
  <name>dfs.namenode.acls.enabled</name>
  <value>true</value>
</property>
```

### setfattr

用法: `hadoop fs -setfattr -n name [-v value] | -x name <path>`

设置一个文件或目录的扩展属性名和值。

选项:

-b: 移除所有的条目除了基本的 ACL 条目。用户、组和其他的条目被保留为与权限位的兼容性。

-n name: 扩展属性名。

-v value: 扩展属性值。有三种不同编码值, 如果该参数是用双引号括起来的, 则该值是引号内的字符串。如果参数是前缀 0x 或 0X, 然后作为一个十六进制数。如果参数从 0 或 0, 然后作为一个 base64 编码。

-x name: 移除所有属性值

path: 文件或则路径

例子:

- `hadoop fs -setfattr -n user.myAttr -v myValue /file`
- `hadoop fs -setfattr -n user.noValue /file`
- `hadoop fs -setfattr -x user.myAttr /file`

退出代码:

返回 0 成功, 非 0 错误

### setrep

用法: `hadoop fs -setrep [-R] [-w] <numReplicas> <path>`

更改文件的备份。如果是一个目录, 会递归改变目录下文件的备份。

选项:

-w 标识, 要求备份完成, 这可能需要很长时间。

-R 标识, 是为了兼容, 没有实际效果

例子:

- `hadoop fs -setrep -w 3 /user/hadoop/dir1`

退出代码:

返回 0 成功, 非 0 错误

### stat

用法: `hadoop fs -stat [format] <path> ...`按指定格式打印文件/目录的打印统计。

格式接受文件块 (%b), 类型 (%F), group 拥有者 (%g), 名字 (%n), block size (%o), replication (%r), 用户拥有者 (%u), 修改日期 (%y, %Y). %y 显示 UTC 日期如 "yyyy-MM-dd HH:mm:ss" 和 %Y 1970 年 1 月 1 日以来显示毫秒 UTC. 如果没有指定, 默认使用 %y.

例子:

- `hadoop fs -stat "%F %u:%g %b %y %n" /file`

退出代码:

返回 0 成功

返回-1 错误

### tail

用法: `hadoop fs -tail [-f] URI`

显示文件内容, 最后千字节的文件发送到 `stdout`,

选项:

-f 选项将输出附加数据随着文件的生长, 如同 Unix

例子:

- `hadoop fs -tail pathname`

退出代码:

返回 0 成功

返回-1 错误

### test

用法: `hadoop fs -test -[defsz] URI`

选项:

-d: 如果路径是一个目录, 返回 0

-e: 如果路径已经存在, 返回 0

-f: 如果路径是一个文件, 返回 0

-s: 如果路径不是空, 返回 0

-z: 如果文件长度为 0, 返回 0

例子:

- `hadoop fs -test -e filename`

### **text**

用法: `hadoop fs -text <src>`

一个源文件，以文本格式输出文件。允许的格式是 zip 和 `textrecordinputstream`。

### **touchz**

用法: `hadoop fs -touchz URI [URI ...]`

创建一个零长度的文件。

例子:

- `hadoop fs -touchz pathname`

退出代码: 返回 0 成功, `-1error`

### **truncate**

用法: `hadoop fs -truncate [-w] <length> <paths>`

截断指定文件模式指定的长度匹配的所有文件。

选项:

`-w` 选项需要等待命令完成块恢复。如果没有 `-w` 选项，在恢复的过程中可能是未闭合的

例子:

- `hadoop fs -truncate 55 /user/hadoop/file1 /user/hadoop/file2`
- `hadoop fs -truncate -w 127 hdfs://nn1.example.com/user/hadoop/file1`

### **usage**

用法: `hadoop fs -usage command`

返回单个命令的帮助。

## about 云介绍

为热爱云开发技术人员提供最全面的信息传播和服务平台，为大家提供云技术文档，视频、云技术学习指导，解疑等。

内容包括：**hadoop** 视频，**Nosql**,虚拟化，**OpenStack**，云平台等相关技术。

**about** 云，本着活到老学到老的精神，为了广大云技术爱好者获取更多知识，在文章开头，都有几个问题，因此 **about**

云亦为学问社区。同样我们准备了每日一读，为了就是每天进步一点点，每天能够学到新的内容。

## 关注本站：

### 1.欢迎加入 **about** 云群

大数据群 **432264021**、**425860289**

云技术 **openstack** 群 **322273151**

### 2.关注腾讯认证空间

[about 云腾讯认证空间](#)

### 3.惯用手机的用户可关注 **about** 云微信地址：

搜索:

[www.aboutyun.com](http://www.aboutyun.com)



#### 4.关注微博:

[新浪微博](#)



#### 5.邮件订阅

[邮件订阅](#)

## 捐助：大数据、云技术视频

[hadoop 生态系统零基础入门及大数据实战【后续不断更新】](#)





# 为何学习大数据

- 1.工资高
- 2.大数据在中小企业逐渐普及，人才需求旺盛
- 3.两年之内，大数据全面普及，不懂大数据程序员可能会被淘汰或则被迫转型

# 如何学习大数据

更多内容：

<https://item.taobao.com/item.htm?spm=a1z10.1-c.w4023-4627152319.4.0qf3BM&id=52041335>

[5976](#)

#####

[about 云零基础开发、部署+理论 openstack 入门视频【J 版及 K 版】](#)

# 如何学习云技术

## 入门云技术

对于云技术，[openstack](#) 是最火的，那么该如何入门 [openstack](#)。

我们或多或少应该了解过 [openstack](#)，它有很多的组件。

问题 1：

具体组件是什么，该如何配置，它的作用是什么，我们可能是不太清楚。

问题 2：

我们知道了 [keystone](#) 是什么，[nova](#) 是什么，该如何部署？

问题 3：

我们部署过程中，会遇到各种各样的问题，这些问题该如何解决？

问题 4：

这些问题解决之后，我们终于创建实例成功，可是 ping 不通外网，这又该如何解决？↵

上面的四个问题，相信如果你不花费一周或则两周的时间，根本是解决不了的。并且这是理想的情况下，如果非理想的情况，我们可能会花费一个月甚至更长时间。如果你亲自学习部署过 [openstack](#)，相信会有自己的体验。↵

特别是创建实例，不能 ping 通外网，这个根本没有错误，如果不懂网络，无从下手。↵

about 云 [openstack juno](#) 解决了上面问题，如果完全按照视频，认识了解 [openstack](#)，部署成功，并且创建实例能 ping 通外网，这完全没有问题的。↵

about 云其它文档

链接: <http://pan.baidu.com/s/1gd0OaEv> 密码: dr4h