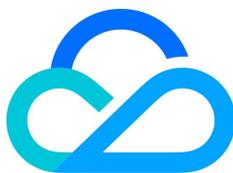


# 云 HDFS 操作指南



腾讯云

**【 版权声明 】**

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 商标声明 】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 服务声明 】**

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

**【 联系我们 】**

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

## 文档目录

### 操作指南

创建 CHDFS

创建权限组

创建权限规则

创建挂载点

挂载 CHDFS

使用 CAM 授权访问

通过 Java 代码访问 CHDFS

设置生命周期

删除文件系统

# 操作指南

## 创建 CHDFS

最近更新时间：2024-12-31 18:26:12

本文为您介绍如何在 CHDFS 控制台上创建文件系统。

### 注意：

- 在同一个地域下，CHDFS 最多支持创建5个文件系统。
- CHDFS 默认情况下不启用 ATime 功能。如果您希望使用 ATime，请注意，CHDFS 并不能保障 Atime 更新的准确性，因此依赖 ATime 特性的功能（如生命周期管理和智能分层）也可能受到此限制，您需要谨慎评估 ATime 不准确性所带来影响。如您经过评估仍希望开启此功能，请 [联系我们](#)。

## 操作步骤

- 登录 [云 HDFS 控制台](#)。
- 在左侧导航栏中，单击**文件系统**，选择所需地域，例如广州。
- 单击**新建**。
- 在弹出的窗口中，输入名称与描述，其他保持默认配置。

**新建文件系统** ×

所属地域 广州 (ap-guangzhou)

名称   
仅支持大小写字母、数字和 - 或 \_ 的组合，长度为 4-64 字符

描述   
请输入备注信息，长度不超过 32 个字符

容量  GB  
文件系统容量最小1GB，最大1PB

根目录默认用户   
仅支持数字、字母、.、\_、或者-，-不能位于首位，长度不超过 32 个字符

根目录默认用户组   
仅支持数字、字母、.、\_、或者-，-不能位于首位，长度不超过 32 个字符

POSIX 权限控制  开启  关闭

Ranger 校验  开启  关闭

超级用户  +

文件系统标签   +

- 名称：**文件系统名称，仅支持大小写字母、数字和 - 或 \_ 的组合，长度为4-64的字符。
- 描述：**文件系统描述信息。

- 单击**保存**，即可新建 CHDFS。

创建完成后，您可以在文件系统管理页面，单击该 CHDFS 的**配置**，查看 CHDFS 基础配置和挂载信息。

# 创建权限组

最近更新时间：2024-10-12 20:45:01

权限组实现对 CHDFS 的权限管理，在使用 CHDFS 之前，需要创建权限组，本文为您详细介绍如何创建权限组。

## 操作步骤

1. 登录 [CHDFS 控制台](#)。
2. 在左侧导航栏中，单击权限组，选择所需地域，例如广州。
3. 单击新建。
4. 在弹出的窗口中，填写如下信息：



- 名称：文件系统名称，仅支持大小写字母、数字和 - 或 \_ 的组合，长度为4-64的字符。
  - VPC网络类型：根据实际选择 VPC 网络类型。
  - VPC网络名称/ID：根据实际需求选择 VPC 网络名称/ID。
  - 描述：权限组描述信息。
5. 单击保存，即可新建权限组。

# 创建权限规则

最近更新时间：2024-10-12 20:45:01

权限组中包括各种权限规则，实现对 CHDFS 权限的具体管理，在使用 CHDFS 之前，需要在已创建权限组中创建权限规则，本文为您详细介绍如何创建权限规则。

## 前提条件

已创建权限组，详情请参见 [创建权限组](#)。

## 操作步骤

1. 登录 [CHDFS 控制台](#)。
2. 在左侧导航栏中，单击**权限组**，选择所需地域，例如广州。
3. 找到需要操作的权限组，单击**添加规则**，进入权限组规则配置页面。  
在该页面，可以查看权限组基本信息和权限规则列表。
4. 在规则列表中，单击**添加规则**，填写授权地址、访问模式、优先级。

### 基本信息

权限组ID	ag-z
名称	group1
类型	VPC
创建时间	2019-08-16 20:04:48
规则数目	0
绑定的挂载点数量	0
描述信息	this is a permission-group!

---

### 规则列表

授权地址	访问模式	优先级 <sup>①</sup>	操作
<input type="text"/>	只读	<input type="text" value="10"/>	<a href="#">保存</a> <a href="#">删除</a>

[+ 添加规则](#)

- 授权地址：支持 IP 地址或网段，含义为授权此 IP 地址或网段访问 CHDFS，例如10.10.1.2或者10.10.1.2/20。
- 访问模式：支持可读可写和只读，含义为授权对 CHDFS 可读可写和只读访问。
- 优先级：支持1 - 100的正整数，1为最高优先级，当 CHDFS 匹配到多条权限规则时，高优先级规则覆盖低优先级规则。

5. 单击**保存**，即可创建权限规则。

# 创建挂载点

最近更新时间：2024-10-12 20:45:01

CVM、CPM 2.0或者容器通过挂载点来访问 CHDFS 中数据，挂载点是 CHDFS 在 VPC 内访问的目标地址，同时每个挂载点对应一个域名，本文为您详细介绍如何创建挂载点。

## ⚠ 注意：

CHDFS 最多支持创建10个挂载点。

## 操作步骤

1. 登录 [CHDFS 控制台](#)。
2. 在左侧导航栏中，单击[文件系统](#)，选择所需地域，例如广州。
3. 找到需要操作的 CHDFS，单击[配置](#)，查看其基础配置和挂载信息。
4. 选择[挂载点](#)页签，单击[添加挂载点](#)。
5. 在弹出的窗口中，输入名称，添加需绑定的 VPC 网络和权限组。

### 添加挂载点

名称 \*

仅支持大小写字母、数字和 - 或 \_ 的组合，长度为 4-64 字符

绑定VPC/权限组	VPC网络名称/ID	VPC关联权限组 ①	操作
			暂无数据
			<a href="#">+ 添加 VPC</a>

如需创建权限组并关联VPC，请前往 [【权限组】](#) 页面，进行操作

[保存](#) [取消](#)

- 名称：挂载点名称，仅支持大小写字母、数字和 - 或 \_ 的组合，长度为4 - 64的字符。
  - VPC 网络名称/ID：选择 VPC 网络。
  - 绑定 VPC/权限组：单击[添加 VPC](#)，选择相应的 VPC 或权限组。若未新建，请先 [创建权限组](#)。
6. 单击[保存](#)，即可新建挂载点。

# 挂载 CHDFS

最近更新时间：2024-10-12 20:45:01

创建 CHDFS 及挂载点后，可以通过挂载点挂载 CHDFS，本文为您详细介绍如何挂载 CHDFS。

## 前提条件

- 确保挂载的机器或者容器内安装了 Java 1.8。
- 确保挂载的机器或者容器其 VPC，与挂载点指定 VPC 相同。
- 确保挂载的机器或者容器其 VPC IP，与挂载点指定权限组中有一条权限规则授权地址匹配。

## 操作步骤

1. 下载 **CHDFS-Hadoop** JAR 包。
2. 将 JAR 包放置对应的目录下，对于 EMR 集群，可同步到所有节点的 `/usr/local/service/hadoop/share/hadoop/common/lib/` 目录下。
3. 编辑 `core-site.xml` 文件，新增以下基本配置：

```
<!--chdfs 的实现类-->
<property>
  <name>fs.AbstractFileSystem ofs.impl</name>
  <value>com.qqcloud.chdfs.fs.CHDFSDelegateFSAdapter</value>
</property>

<!--chdfs 的实现类-->
<property>
  <name>fs ofs.impl</name>
  <value>com.qqcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter</value>
</property>

<!--本地 cache 的临时目录，对于读写数据，当内存 cache 不足时会写入本地硬盘，这个路径若不存在会自动创建-->
<property>
  <name>fs ofs.tmp.cache.dir</name>
  <value>/data/chdfs_tmp_cache</value>
</property>

<!--用户的 appId，可登录腾讯云控制台 (https://console.cloud.tencent.com/developer) 查看-->
<property>
  <name>fs ofs.user.appid</name>
  <value>1250000000</value>
</property>

<!--flush 语义，默认 false (异步刷盘)，请参考下图数据可见性与持久化详细说明 -->
<property>
  <name>fs ofs.upload.flush.flag</name>
  <value>>false</value>
</property>
```

4. 将 `core-site.xml` 同步到所有 hadoop 节点上。

### 说明

对于 EMR 集群，以上步骤3、4可在 EMR 控制台的组件管理中，修改 HDFS 配置即可。

5. 使用 `hadoop fs` 命令行工具，运行 `hadoop fs -ls ofs://{mountpoint}/` 命令，这里 `mountpoint` 为挂载地址。如果正常列出文件列表，则说明已经成功挂载 CHDFS。
6. 用户也可使用 `hadoop` 其他配置项，或者 `mr` 任务在 CHDFS 上运行数据任务。对于 `mr` 任务，可以通过 `-Dfs.defaultFS=ofs://{mountpoint}/` 将本次任务的默认输入输出 FS 改为 CHDFS。

## 场景说明

### 写入数据可见性与持久化

CHDFS 是云端分布式文件系统，通过 CHDFS 客户端打开一个文件，会返回一个输出流；通过输出流写入数据后，CHDFS 客户端会在满足一定大小后(通常为4MB)，异步刷盘。如果上层主动调用输出流的 flush 接口，默认的行为是忽略（和通常的 flush 同步刷盘语义不同）。等待后续满足写入量达到一定大小后异步刷盘。或者最后调用 close 的时候，也才会强制同步刷盘。close 成功的数据，表示已经持久化成功，可保证后续读取到。

CHDFS 客户端默认异步 flush，因为相比起本地磁盘，云端的访问时延较大，异步 flush 可以减少与云端的交互，避免 flush 卡住用户写入，显著提升写入性能，减少作业时间。风险点是如果最终没有主动调用 close，会存在未刷盘的数据丢失的风险。

对于需要 flush 后需要立刻可见以及持久化的场景，例如使用 CHDFS 存储 Hbase Wal Log，Journal 日志等，请开启同步刷盘。请参考配置项 fs.ofs.upload.flush.flag 说明。

### 配置项说明

配置项	说明	默认值	是否必填
fs.ofs.tmp.cache.dir	存放自动下载的相关 Jar 包的路径	无	是
fs.ofs.tmp.disk.cache.dir	上传过程中, 使用的磁盘 buffer 文件的产生路径	./tmp/chdfs-disk	否
fs.ofs.map.block.size	chdfs 文件系统的 block 大小, 单位为字节。默认为128MB (只对 map 切分有影响, 和 chdfs 底层存储切块大小无关)	134217728	否
fs.ofs.data.transfer.thread.count	chdfs 传输数据时的并行线程数	32	否
fs.ofs.block.max.memory.cache.mb	chdfs 插件使用的内存 buffer 的大小, 单位为 MB。(对读写都有加速作用)	16	否
fs.ofs.block.max.file.cache.mb	chdfs 插件使用的磁盘 buffer 的大小, 单位为 MB。(对写有加速作用)	256	否
fs.ofs.prev.read.block.count	读取时, 预读的 chdfs block 数量 (chdfs 的底层 block 大小一般为 4MB)	4	否
fs.ofs.plugin.info.log	是否打印插件的调试日志, 日志以 info 级别打印。可选值为 true、false	false	否
fs.ofs.bucket.region	文件系统或者元数据加速器 bucket 所在的地域, 如 ap-shanghai, ap-beijing, 配置项适用于 chdfs_hadoop_plugin_network 版本V2.7及其以上	false	否
fs.ofs.upload.flush.flag	写数据, 调用输出流 flush 接口时, 是否同步刷数据, 默认为异步刷数据 (false)。如需同步刷盘, 请设置为 true	false	否

# 使用 CAM 授权访问

最近更新时间：2024-12-19 15:36:12

## CHDFS 预设策略

CHDFS 预设授权策略如下：

策略	说明
QcloudCHDFSReadOnlyAccess	只读访问 CHDFS 的权限
QcloudCHDFSFullAccess	管理 CHDFS 的权限

## CHDFS 授权操作

Action	Resource	说明
chdfs:CreateFileSystem	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/*	创建 CHDFS
chdfs>DeleteFileSystem	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	删除 CHDFS
chdfs:ModifyFileSystem	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	修改 CHDFS 属性
chdfs:DescribeFileSystem	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	查看 CHDFS 详细信息
chdfs:DescribeFileSystems	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	查看 CHDFS 列表
chdfs:CreateMountPoint	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	创建挂载点
chdfs>DeleteMountPoint	qcs::chdfs:\${region-id}:uin/\${account-uin}:mountpoint/\${mount-point-id}	删除挂载点
chdfs:ModifyMountPoint	qcs::chdfs:\${region-id}:uin/\${account-uin}:mountpoint/\${mount-point-id}	修改挂载点属性
chdfs:DescribeMountPoint	qcs::chdfs:\${region-id}:uin/\${account-uin}:mountpoint/\${mount-point-id}	查看挂载点详细信息
chdfs:DescribeMountPoints	qcs::chdfs:\${region-id}:uin/\${account-uin}:mountpoint/\${mount-point-id}	查看挂载点列表
chdfs:AssociateAccessGroups	qcs::chdfs:\${region-id}:uin/\${account-uin}:mountpoint/\${mount-point-id}	绑定权限组列表
chdfs:DisassociateAccessGroups	qcs::chdfs:\${region-id}:uin/\${account-uin}:mountpoint/\${mount-point-id}	解绑权限组列表
chdfs:CreateAccessGroup	qcs::chdfs:\${region-id}:uin/\${account-uin}:vpc/\${vpc-id} 或qcs::chdfs:\${region-id}:uin/\${account-uin}:unVpcId/\${unVpcId}	创建权限组
chdfs>DeleteAccessGroup	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessgroup/\${access-group-id}	删除权限组
chdfs:ModifyAccessGroup	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessgroup/\${access-group-id}	修改权限组属性
chdfs:DescribeAccessGroup	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessgroup/\${access-group-id}	查看权限组详细信息

chdfs:DescribeAccessGroups	qcs::chdfs:\${region-id}:uin/\${account-uid}:accessgroup/\${access-group-id}	查看权限组列表
chdfs:CreateAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:accessgroup/\${access-group-id}	批量创建权限规则
chdfs>DeleteAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:accessgroup/\${access-group-id}	批量删除权限规则
chdfs:ModifyAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:accessgroup/\${access-group-id}	批量修改权限规则属性
chdfs:DescribeAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:accessgroup/\${access-group-id}	查看权限规则列表
chdfs:CreateLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	批量创建生命周期规则
chdfs>DeleteLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	批量删除生命周期规则
chdfs:ModifyLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	批量修改生命周期规则属性
chdfs:DescribeLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	查看生命周期规则列表
chdfs:CreateRestoreTasks	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	批量创建回热任务
chdfs:DescribeRestoreTasks	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	查看回热任务列表
chdfs:CreateInventoryConfig	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	创建清单配置
chdfs>DeleteInventoryConfig	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	删除清单配置
chdfs:ModifyInventoryConfig	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	修改清单配置属性
chdfs:DescribeInventoryConfigs	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	查看清单配置列表
chdfs:CreatePathProtectionRule	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	创建路径保护规则
chdfs>DeletePathProtectionRule	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	删除路径保护规则
chdfs:ModifyPathProtectionRule	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	修改路径保护规则属性
chdfs:DescribePathProtectionRules	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	查看路径保护规则列表
chdfs:ModifyResourceTags	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	修改资源标签列表
chdfs:DescribeResourceTags	qcs::chdfs:\${region-id}:uin/\${account-uid}:filesystem/\${file-system-id}	查看资源标签列表

## CHDFS 授权策略示例

授予子账号 CHDFS 管控系统只读权限的策略示例如下：

```
{
  "version": "2.0",
  "statement": [{
    "effect": "allow",
    "action": [
      "name/chdfs:Describe*"
    ],
    "resource": [
      "*"
    ]
  }]
}
```

授予子账号查看 CHDFS 的策略示例如下：

```
{
  "version": "2.0",
  "statement": [{
    "effect": "allow",
    "action": [
      "name/chdfs:DescribeFileSystem"
    ],
    "resource": [
      "qcs::chdfs::uin/ownerUin:filesystem/filesystemId"
    ]
  }]
}
```

# 通过 Java 代码访问 CHDFS

最近更新时间：2024-12-31 18:26:12

在部署云 HDFS（Cloud HDFS，CHDFS）的 JAR 包之后，除了可以使用命令行、大数据组件等方式操作 CHDFS，还可以通过 Java 代码来访问 CHDFS。本文指导您如何通过 Java 代码访问 CHDFS。

## 前提条件

- 确保已经部署了 CHDFS 的相关 JAR 包。详情请参见 [挂载 CHDFS](#)。
- 确保运行 Java 程序的机器处于挂载点权限组允许访问的私有网络 VPC 中。

## 操作步骤

1. 新建 maven 工程，并在 maven 的 pom.xml 中添加以下依赖项（请根据自己实际 hadoop 环境设置 hadoop-common 包的版本）。

```
<dependencies>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-common</artifactId>
    <version>2.8.5</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

2. 参考如下 hadoop 的代码进行修改，其中的配置项可参考 [挂载 CHDFS](#) 文档进行修改。以及重点关注其中数据持久化和可见性相关的说明。以下只列出了部分常见的文件系统的操作，其他的接口可参考 [Hadoop FileSystem 接口文档](#)。

```
package com.qcloud.chdfs.demo;

import org.apache.commons.io.IOUtils;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileChecksum;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;

import java.io.IOException;
import java.net.URI;
import java.nio.ByteBuffer;

public class Demo {
    private static FileSystem initFS() throws IOException {
        Configuration conf = new Configuration();
        // CHDFS 的配置项可参见 https://cloud.tencent.com/document/product/1105/36368
        // 以下配置是必填项

        conf.set("fs.ofs.impl", "com.qcloud.chdfs.fs.CHDFSHadoopFileSystemAdapter");
        conf.set("fs.AbstractFileSystem.ofs.impl", "com.qcloud.chdfs.fs.CHDFSDelegateFSAdapter");
        conf.set("fs.ofs.tmp.cache.dir", "/data/chdfs_tmp_cache");
        conf.set("fs.ofs.user.appid", "1250000000");
        // 其他可选配置项请参见 https://cloud.tencent.com/document/product/1105/36368

        String chdfsUrl = "ofs://f4maaabbb-ccdd.chdfs.ap-guangzhou.myqcloud.com/";
        return FileSystem.get(URI.create(chdfsUrl), conf);
    }

    private static void mkdir(FileSystem fs, Path filePath) throws IOException {
```

```
        fs.mkdirs(filePath);
    }

    private static void createFile(FileSystem fs, Path filePath) throws IOException {
        // 创建一个文件（如果存在则将其覆盖）
        // if the parent dir does not exist, fs will create it!
        FSDDataOutputStream out = fs.create(filePath, true);
        try {
            // 写入一个文件
            String content = "test write file";
            out.write(content.getBytes());
        } finally {
            // close 返回成功，表示数据写入成功，若抛出异常，表示数据写入失败
            out.close();
        }
    }

    private static void readFile(FileSystem fs, Path filePath) throws IOException {
        FSDDataInputStream in = fs.open(filePath);
        try {
            byte[] buf = new byte[4096];
            int readLen = -1;
            do {
                readLen = in.read(buf);
            } while (readLen >= 0);
        } finally {
            IOUtils.closeQuietly(in);
        }
    }

    private static void queryFileOrDirStatus(FileSystem fs, Path path) throws IOException {
        FileStatus fileStatus = fs.getFileStatus(path);
        if (fileStatus.isDirectory()) {
            System.out.printf("path %s is dir\n", path);
            return;
        }

        long fileLen = fileStatus.getLength();
        // CHDFS 默认情况下不启用 ATime 功能。如果您希望使用 ATime，请注意，CHDFS 并不能保障 Atime 更新的准确性，
        // 因此依赖 ATime 特性的功能（如生命周期管理和智能分层）也可能受到此限制，您需要谨慎评估 ATime 不准确性所带来的影响。如您经过
        // 评估仍希望开启此功能，请通过工单与我们联系。
        long accessTime = fileStatus.getAccessTime();
        long modifyTime = fileStatus.getModificationTime();
        String owner = fileStatus.getOwner();
        String group = fileStatus.getGroup();

        System.out.printf("path %s is file, fileLen: %d, accessTime: %d, modifyTime: %d, owner: %s, group: %s\n",
            path, fileLen, accessTime, modifyTime, owner, group);
    }

    // 默认的校验类型为 COMPOSITE-CRC32C
    private static void getFileChecksum(FileSystem fs, Path path) throws IOException {
        FileChecksum checksum = fs.getFileChecksum(path);
        System.out.printf("path %s, checksumType: %s, checksumCrcVal: %d\n",
            path, checksum.getAlgorithmName(), ByteBuffer.wrap(checksum.getBytes()).getInt());
    }
}
```

```
private static void copyFileFromLocal(FileSystem fs, Path chdfsPath, Path localPath) throws
IOException {
    fs.copyFromLocalFile(localPath, chdfsPath);
}

private static void copyFileToLocal(FileSystem fs, Path chdfsPath, Path localPath) throws
IOException {
    fs.copyToLocalFile(chdfsPath, localPath);
}

private static void renamePath(FileSystem fs, Path oldPath, Path newPath) throws IOException {
    fs.rename(oldPath, newPath);
}

private static void listDirPath(FileSystem fs, Path dirPath) throws IOException {
    FileStatus[] dirMemberArray = fs.listStatus(dirPath);

    for (FileStatus dirMember : dirMemberArray) {
        System.out.printf("dirMember path %s, fileLen: %d\n", dirMember.getPath(),
dirMember.getLen());
    }
}

// 递归删除标志用于删除目录
// 如果递归为 false 并且 dir 不为空, 则操作将失败
private static void deleteFileOrDir(FileSystem fs, Path path, boolean recursive) throws
IOException {
    fs.delete(path, recursive);
}

private static void closeFileSystem(FileSystem fs) throws IOException {
    fs.close();
}

public static void main(String[] args) throws IOException {
    // 初始化文件系统
    FileSystem fs = initFS();

    // 创建文件
    Path chdfsFilePath = new Path("/folder/exampleobject.txt");
    createFile(fs, chdfsFilePath);

    // 读取文件
    readFile(fs, chdfsFilePath);

    // 查询文件或目录
    queryFileOrDirStatus(fs, chdfsFilePath);

    // 获取文件校验和
```

```
getFileChecksum(fs, chdfsFilePath);

// 从本地复制文件
Path localFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
copyFileFromLocal(fs, chdfsFilePath, localFilePath);

// 获取文件到本地
Path localDownFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
copyFileToLocal(fs, chdfsFilePath, localDownFilePath);

// 重命名
Path newPath = new Path("/doc/example.txt");
renamePath(fs, chdfsFilePath, newPath);

// 删除文件
deleteFileOrDir(fs, newPath, false);

// 创建目录
Path dirPath = new Path("/folder");
mkdir(fs, dirPath);

// 在目录中创建文件
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);

// 列出目录
listDirPath(fs, dirPath);

// 删除目录
deleteFileOrDir(fs, dirPath, true);

// 关闭文件系统
closeFileSystem(fs);
}
}
```

### 3. 编译和运行。

#### ① 说明

- 运行前，请确保已正确设置 classpath。classpath 需包含 Hadoop common 包以及 CHDFS 包的路径。
- 对于 EMR 环境，如果您按照 [挂载 CHDFS](#) 逐步操作，那么 Hadoop common 包通常在 `/usr/local/service/hadoop/share/hadoop/common/` 目录下，CHDFS 包通常在 `/usr/local/service/hadoop/share/hadoop/common/lib/` 目录下。

# 设置生命周期

最近更新时间：2025-05-08 19:29:21

## 简介

当您需要定期对文件进行存储类型转换，或者希望定期删除文件以降低成本时，您可以使用生命周期功能。云 HDFS 会按照您设定的规则对指定的文件系统或目标路径下的文件在多少天后自动进行存储类型转换或删除。

### 说明：

- 生命周期的设置支持最长天数为3650天。
- 生命周期不会对小于64KB的文件执行转换操作。

## 操作步骤

- 登录 [云 HDFS 控制台](#)。
- 在左侧导航中，单击**文件系统**，进入文件系统列表页面。
- 找到需要开启生命周期功能的文件系统，单击其**名称**，进入文件系统详情页面。
- 单击上方的高级配置配置项，在**生命周期**的子页面，单击**添加规则**。



- 根据实际需求添加生命周期规则，配置项说明如下。

## 添加规则



**i** 为避免生命周期早于配置时间提前执行，使用生命周期沉降或删除文件（例如 .Trash）前，请您参考[回收站清空机制说明](#)。生命周期不会对小于64KB的文件执行存储类型的转换操作。

状态  开启  关闭

规则名称 \* rule1

仅支持大小写字母、数字和 - 或 \_ 的组合，长度为 4-64 字符

应用范围  整个CHDFS文件系统  目标路径

目标路径 \* / doc

支持目录或者文件路径

规则内容  文件更新时间的 30 天后沉降至低频存储

文件更新时间的 60 天后沉降至智能分层存储

文件更新时间的 90 天后沉降至归档存储

**!** 归档适用于极少访问的数据；访问归档数据需要先恢复，恢复需要耗费一定时间和成本。

文件更新时间的 360 天后删除

**!** 文件删除后不可恢复，请慎重按照业务需要进行选择。删除的时间必须大于转成归档的时间。

保存

取消

- **状态**：选择开启。
  - **规则名称**：输入您的生命周期规则名称。
  - **应用范围**：本生命周期规则可以作用于整个 CHDFS 文件系统下的文件；也可以作用于具体的目标路径下的文件，例如目标路径为 doc，则作用于 doc 目录下的文件。
  - **规则内容**：
    - **存储类型转换**：目前 CHDFS 支持将文件系统下的文件由标准存储沉降至低频存储、智能分层存储、归档存储，支持文件到期后删除。其中存储类型由热到冷分别为：**标准存储 > 低频存储 > 智能分层存储 > 归档存储**，存储类型转换只能由热到冷，不能反向进行。时间是以文件在云 HDFS 上的修改时间为标准开始计算，修改文件的行为等同于重新上传文件。
- ⚠ 注意：**  
归档适用于极少访问的数据；访问归档数据需要先恢复，恢复需要耗费一定时间和成本。
- **文件删除**：当您希望定期删除文件以降低成本时，可勾选该项。

**注意：**

文件删除后不可恢复，请慎重按照业务需要进行选择。删除的时间必须大于转成归档的时间。

- 文件更新时间：CHDFS 生命周期使用最近文件更新时间作为沉降或删除的依据。CHDFS 取文件 MTime、ATime、CTime 中的最近时间，作为文件更新时间。例如，若您6月1日创建文件 `text.txt`，6月10日再次访问该文件，后续对此文件无操作。6月11日配置生命周期规则，指定 `text.txt` 更新后10天沉降到低频。此时，该文件的 MTime 为6月1日；ATime 为6月10日，该文件的最近更新时间为6月10日，该文件将会于6月20日沉降为低频存储类型。

**注意：**

- 为保证读写性能最优，CHDFS 默认不开启 ATime 追踪。若您需要使用 ATime 功能，请 [联系我们](#)。
- 为避免误删数据，使用生命周期删除 .Trash 中的文件前，请您参见 [回收站清空机制说明](#)。

6. 配置好生命周期规则后，单击**保存**，您即可看到生命周期规则。

7. 当需要停止生命周期规则时，单击**编辑**，将生命周期规则的状态修改为**关闭**或者直接**删除**生命周期规则即可。

规则名称 ↓	应用范围	规则内容	状态	创建时间 ↓	操作
rule1	/doc	到期归档: 30天 到期删除: 180天	开启	2025-04-24 20:06:34	<a href="#">编辑</a> <a href="#">删除</a>

# 删除文件系统

最近更新时间：2024-10-12 20:45:01

本文为您介绍如何在 CHDFS 控制台上删除文件系统。

## 注意

文件系统删除后，该系统下的所有配置将会被清空，且无法恢复，请谨慎操作。

## 操作步骤

1. 登录 [CHDFS 控制台](#)。
2. 在左侧导航栏中，单击文件系统。
3. 在文件系统页面，选择文件系统所在的地域，例如广州。
4. 找到需要删除的文件系统，并单击文件系统右侧的删除。



5. 在弹出的提示框中，单击删除，即可删除文件系统。