

云 HDFS 操作指南 产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【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。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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模式的承诺或保证。

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

删除文件系统

操作指南

创建 CHDFS

最近更新时间：2024-02-29 17:48:52

本文为您详细介绍如何在 CHDFS 控制台上创建文件系统。

操作步骤

1. 登录 [CHDFS 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。
2. 单击【新建】，在新建文件系统页面，输入名称与描述。

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。

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

容量
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 +

3. 单击【保存】，即可新建 CHDFS。

4. 单击 CHDFS 上的【配置】，可查看 CHDFS 基础配置和挂载信息，如下图所示：

容量设置 [编辑](#)

已用容量 	0 B
存储容量 	10240 GB
标准存储容量 	0 B
低频存储容量 	0 B
归档存储容量 	0 B
深度归档容量 	0 B
智能分层容量 	0 B

基本信息

系统ID	f14mw6ewdvgh 
系统名称	test 
所属地域	广州 (ap-guangzhou)
挂载点数量	0
POSIX 权限控制	开启 
超级用户 	- 
描述信息	这是CHDFS 

创建权限组

最近更新时间：2024-02-29 17:53:42

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](#)，在左侧导航栏中单击【权限组】，选择所需地域，例如广州。
2. 单击【新建】，在弹出的新建权限组窗口中，输入名称与描述。

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：权限组描述信息。

新建权限组

名称 *

VPC网络类型 *

VPC网络名称/ID ⓘ *

描述

3. 单击【保存】，即可新建权限组。

创建权限规则

最近更新时间：2022-03-30 09:30:25

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

需已创建权限组，详情请参见 [创建权限组](#)。

操作步骤

1. 登录 [CHDFS 控制台](#)，在左侧导航栏中单击【权限组】，选择所需地域，例如广州。
2. 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，可以查看权限组基本信息和权限规则列表。在规则列表中填写授权地址、访问模式、优先级。

Basic Information

Permission Group ID `ag-ott1833d`

Name `group1`

VPC `vpc-1k...` (Default-VPC | 172.16.0.0/16)

Creation Time 2021-09-26 16:07:15

Rules 0

Bound Mount Points 0

Description `this is a permission-group!`

Rule List

Authorized Address	Access Mode	Priority	Operation
Add a rule first.			
+ Add Rule			

- 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 - 100，1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。

3. 单击【保存】，即可创建权限规则。

创建挂载点

最近更新时间：2022-03-30 09:30:26

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。

操作步骤

1. 登录 [CHDFS 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。
2. 找到需要操作的 CHDFS，单击其【配置】，可查看其基础配置和挂载信息。
3. 选择【挂载点】>【添加挂载点】，在添加挂载点页面，输入名称，指定 VPC 网络和权限组。

- 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。
- 权限组：选择权限组，若未新建，请先 [创建权限组](#)。

Add Mount Point ✕

Name *
4-64 characters; supports letters, digits, hyphens (-), and underscores (_)

VPC/Permission Group	VPC Name/ID	Bound Permission Group ⓘ	Op...
	<input type="text" value="vpc-12345678(Default-VPC 172.16.0.0/16)"/>	<input type="text" value="group1(ag-cm-12345678)"/>	Delete
+ Add VPC			

To create a permission group and bind it to a VPC, go to [Permission Groups](#).

3. 单击【保存】，即可新建挂载点。

挂载 CHDFS

最近更新时间：2022-03-30 09:30:26

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.qcloud.chdfs.fs.CHDFSDelegateFSAdapter</value>
</property>
<property>
<name>fs.ofs.impl</name>
<value>com.qcloud.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>
<!--appId-->
<property>
<name>fs.ofs.user.appid</name>
```

```
<value>1250000000</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。

其他配置项

配置项	说明	默认值	是否必填
<code>fs.ofs.tmp.cache.dir</code>	存放临时数据	无	是
<code>fs.ofs.map.block.size</code>	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
<code>fs.ofs.data.transfer.thread.count</code>	chdfs 传输数据时的并行线程数	32	否
<code>fs.ofs.block.max.memory.cache.mb</code>	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
<code>fs.ofs.block.max.file.cache.mb</code>	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
<code>fs.ofs.prev.read.block.count</code>	读取时，预读的 chdfs block 数量（chdfs 的底层 block 大小一般为4MB）	4	否
<code>fs.ofs.plugin.info.log</code>	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

使用 CAM 授权访问

最近更新时间：2022-03-30 09:30:26

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

Action	Resource	说明
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-uin}:accessgroup/\${access-group-id}	查看权限组列表
chdfs:CreateAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessgroup/\${access-group-id}	批量创建权限规则
chdfs>DeleteAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessrule/\${access-rule-id}	批量删除权限规则
chdfs:ModifyAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessrule/\${access-rule-id}	批量修改权限规则属性
chdfs:DescribeAccessRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:accessgroup/\${access-group-id}	查看权限规则列表
chdfs:CreateLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	批量创建生命周期规则
chdfs>DeleteLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:lifecyclerule/\${life-cycle-rule-id}	批量删除生命周期规则

Action	Resource	说明
chdfs:ModifyLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:lifecyclerule/\${life-cycle-rule-id}	批量修改生命周期规则属性
chdfs:DescribeLifeCycleRules	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	查看生命周期规则列表
chdfs:CreateRestoreTasks	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	批量创建回热任务
chdfs:DescribeRestoreTasks	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	查看回热任务列表
chdfs:ModifyResourceTags	qcs::chdfs:\${region-id}:uin/\${account-uin}:filesystem/\${file-system-id}	修改资源标签列表
chdfs:DescribeResourceTags	qcs::chdfs:\${region-id}:uin/\${account-uin}: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：2022-03-30 09:30:26

操作场景

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. 参考如下操作 hadoop 的代码进行修改，其中的配置项及其说明可参见 [挂载 CHDFS](#)。
以下只列出部分常见的文件系统操作接口，其他接口请参考 [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.CHDFSDelegateFS
Adapter");
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://f4maaabb-bcdd.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!
FSDataOutputStream out = fs.create(filePath, true);
try {
// 写入一个文件
String content = "test write file";
out.write(content.getBytes());
} finally {
IOUtils.closeQuietly(out);
}
}
private static void readFile(FileSystem fs, Path filePath) throws IOException {
FSDataInputStream 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.getLen();
    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
    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。

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。

删除文件系统

最近更新时间：2022-03-30 09:30:26

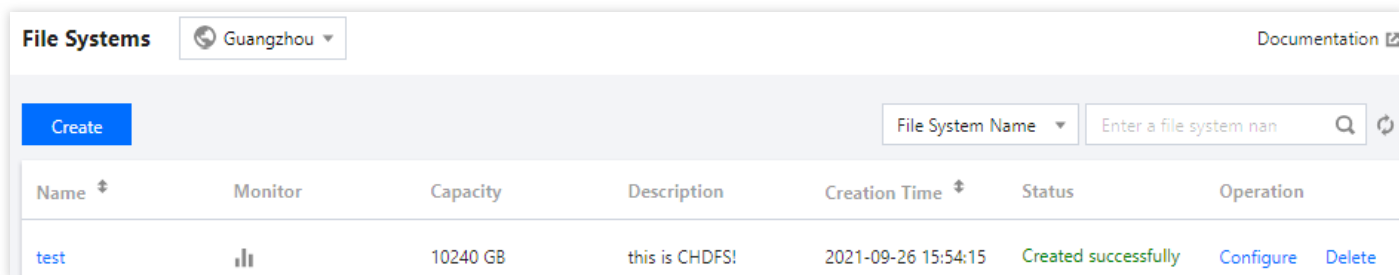
本文为您详细介绍如何在 CHDFS 控制台上删除文件系统。

注意：

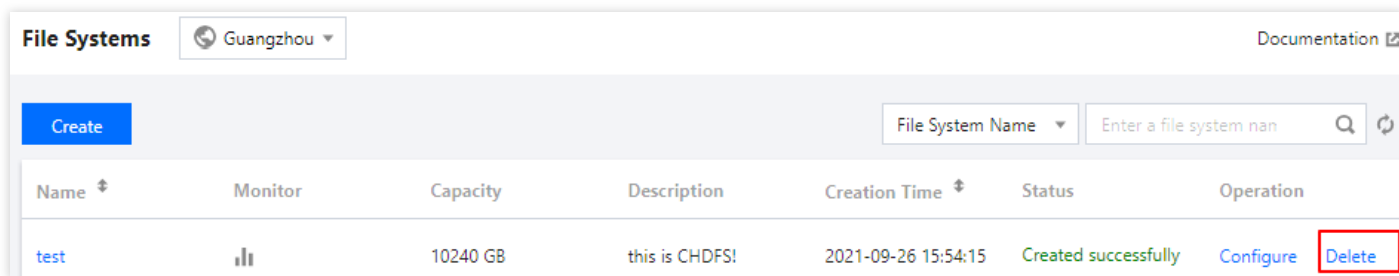
文件系统删除后，该系统下的所有配置将会被清空，且无法恢复，请谨慎操作。

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. 在文件系统界面中，选择文件系统所在的地域，例如广州。



4. 在文件系统列表中找到需要删除的文件系统，并单击文件系统右侧的【删除】。



5. 在弹出的窗口中，单击【删除】，即可删除文件系统。

