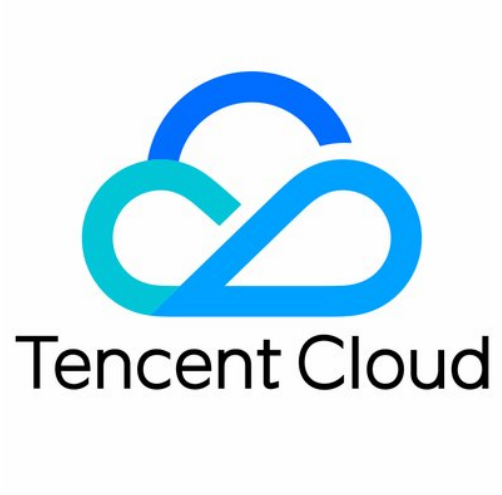
Cloud HDFS

운영 가이드

제품 문서



목록:

운영 가이드
CHDFS 생성

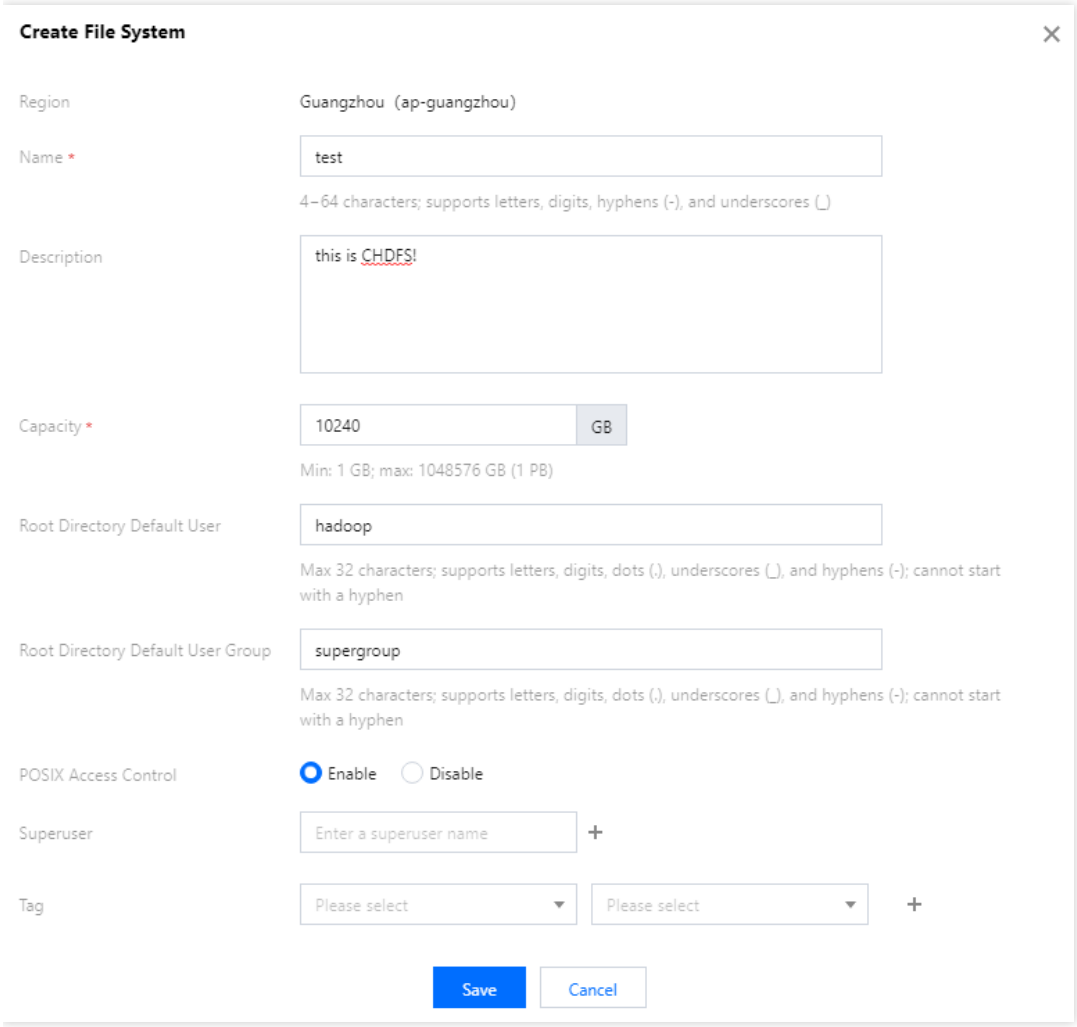
최종 업데이트 날짜: : 2022-03-30 09:30:25

본 문서는 CHDFS 콘솔에서 파일 시스템을 생성하는 방법을 소개합니다.

작업 절차

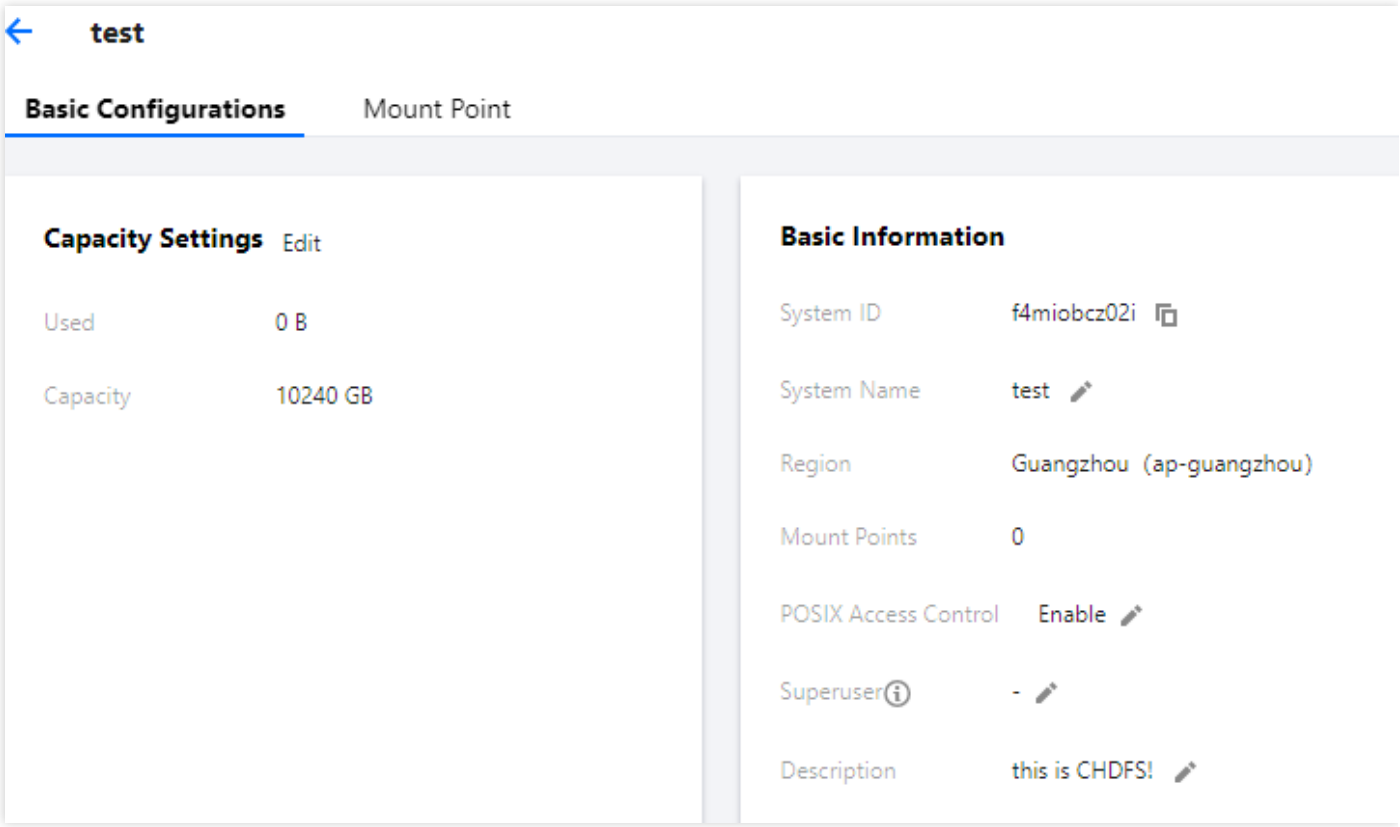
1. CHDFS 콘솔에 로그인한 후, 왼쪽 메뉴에서 [파일 시스템]을 클릭하고, 해당 리전을 선택합니다. 예: 광저우
2. [생성]을 클릭하고, 파일 시스템 생성 페이지에서 이름과 설명을 입력합니다.

- **이름**: 파일 시스템 이름으로 대소문자 알파벳, 숫자, - 또는 _ 조합만 지원하며 길이는 4~64자입니다.
- **설명**: 파일 시스템 설명 정보.



3. [저장]을 클릭하면 CHDFS를 생성할 수 있습니다.

4. CHDFS에서 [설정]을 클릭하면, 다음과 같이 CHDFS 기본 설정과 마운트 정보를 확인할 수 있습니다.



권한 그룹 생성

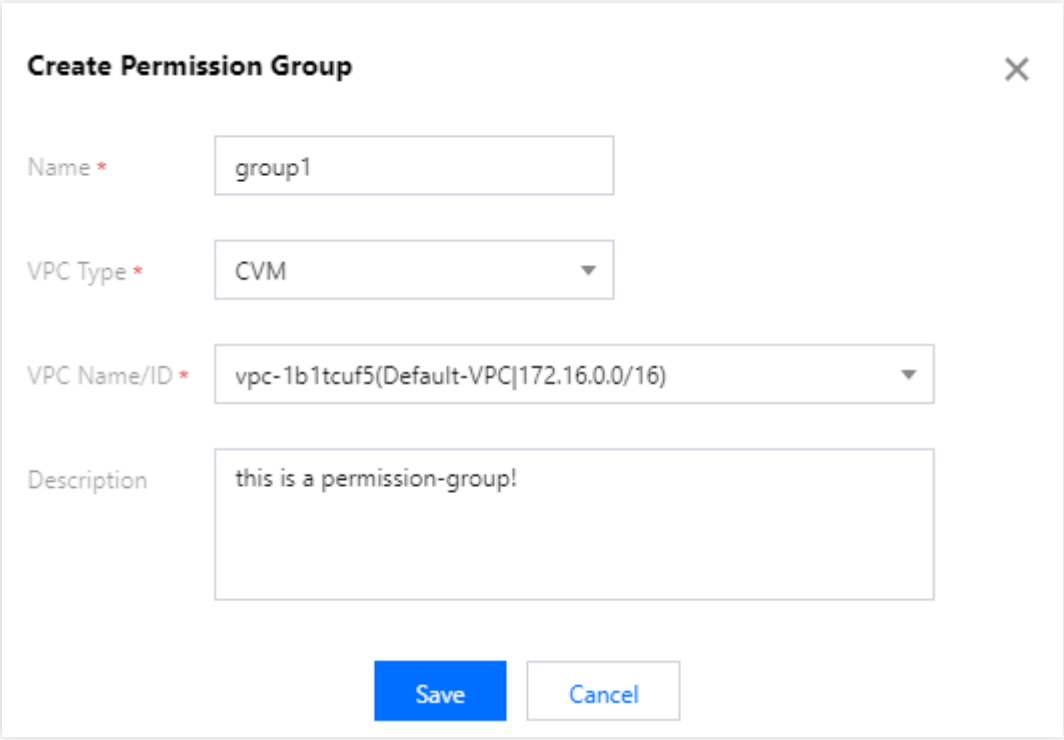
최종 업데이트 날짜: : 2022-03-30 09:30:25

CHDFS의 권한을 관리하는 권한 그룹은 CHDFS 사용 전에 생성해야 합니다. 본 문서는 권한 그룹 생성 방법을 소개합니다.

작업 절차

1. CHDFS 콘솔에 로그인한 후, 왼쪽 메뉴에서 [권한 그룹]을 클릭하고 해당 리전을 선택합니다. 예: 광저우
2. [생성]을 클릭하고 권한 그룹 생성 팝업창에 이름과 설명을 입력합니다.

- 이름: 파일 시스템 이름으로 대소문자 알파벳, 숫자, - 또는 _ 조합만 지원하며 길이는 4~64자 입니다.
- 설명: 권한 그룹 설명 정보.



3. [저장]을 클릭하면 권한 그룹이 생성됩니다.

권한 규칙 생성

최종 업데이트 날짜: : 2022-03-30 09:30:26

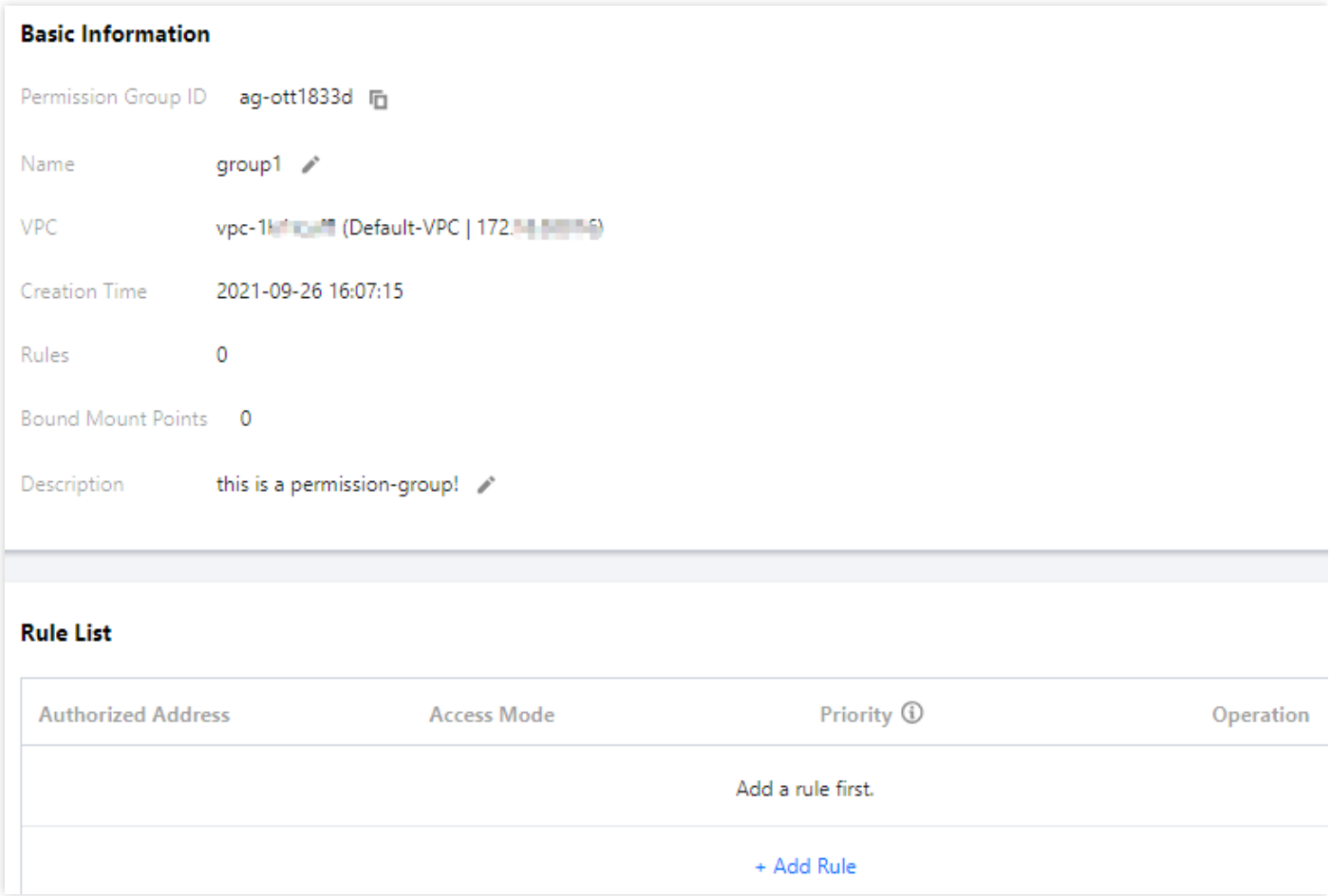
권한 그룹에는 다양한 권한 규칙이 포함되며 CHDFS 권한을 구체적으로 관리합니다. CHDFS를 사용하기 전에 기존에 생성한 권한 그룹에 권한 규칙을 생성해야 합니다. 본 문서는 권한 규칙 설정 방법을 소개합니다.

전제 조건

생성된 권한 그룹에 대한 자세한 내용은 권한 그룹 생성을 참고하십시오.

작업 절차

1. CHDFS 콘솔에 로그인한 후, 왼쪽 메뉴에서 [권한 그룹]을 클릭하고, 해당 리전을 선택합니다. 예: 광저우
2. 작업할 권한 그룹을 찾아 [규칙 추가]를 클릭하면 권한 그룹 설정 페이지로 이동하고 권한 그룹 기본 정보와 권한 규칙 리스트를 확인할 수 있습니다. 규칙 리스트에 권한 부여한 주소, 액세스 모드, 우선순위를 입력합니다.



- 라이선스 주소: IP 주소 또는 대역을 지원하며, 10.10.1.2 또는 10.10.1.2/20 같은 해당 IP 주소 또는 대역에 CHDFS 액세스 권한을 부여한다는 의미입니다.
- 액세스 모드: 읽기/쓰기 가능, 읽기 전용을 지원하며 CHDFS에 대해 읽기/쓰기 가능과 읽기 전용 액세스 권한을 부여한다는 의미입니다.
- 우선순위: 1~100까지 지원하며 1은 가장 높은 우선순위를 의미합니다. CHDFS가 여러 권한 규칙에 매칭되면 우선순위가 높은 규칙이 낮은 규칙을 덮어쓰기합니다.

3. [저장]을 클릭하면 권한 규칙이 생성됩니다.

마운트 포인트 생성

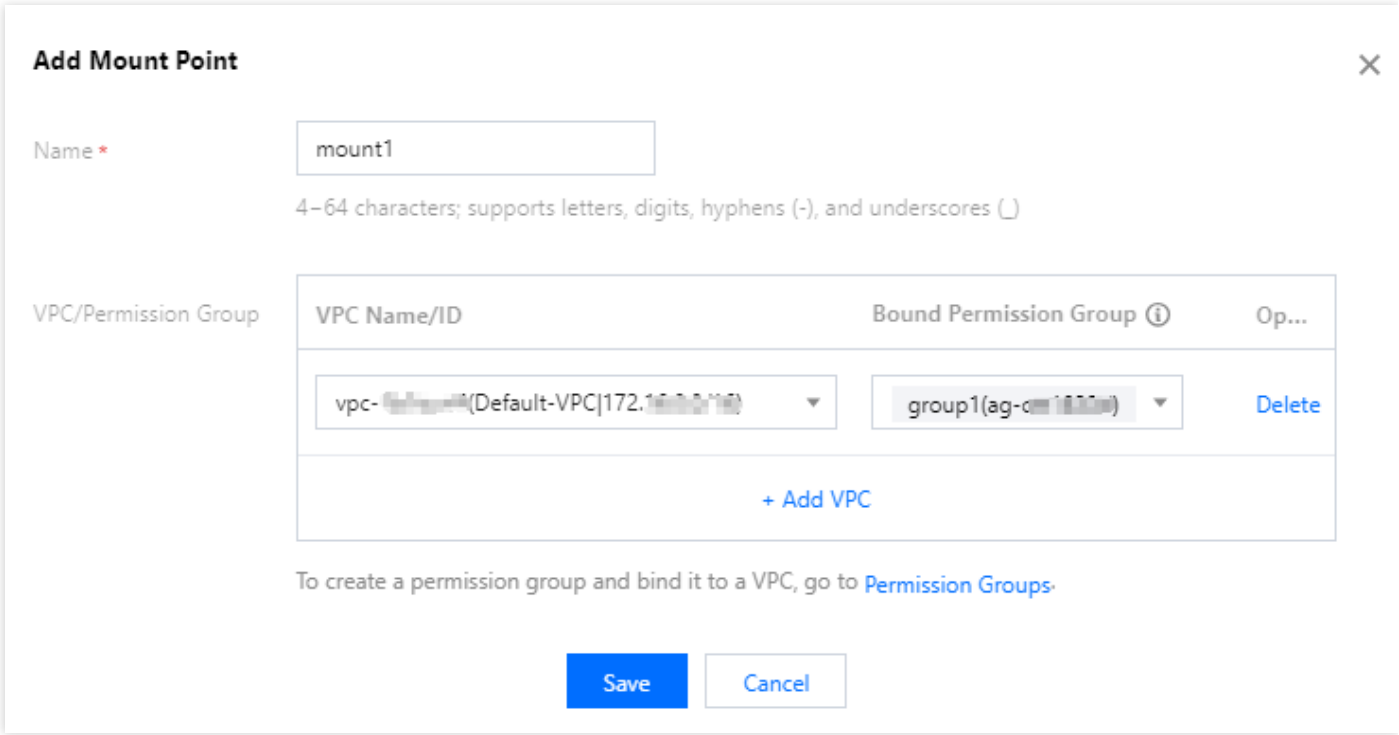
최종 업데이트 날짜: : 2022-03-30 09:30:26

CVM, CPM 2.0 또는 컨테이너는 마운트 포인트를 통해 CHDFS 데이터에 액세스합니다. 마운트 포인트는 VPC 내에서 CHDFS가 액세스하는 타깃 주소이며 모든 마운트 포인트는 하나의 도메인에 대응됩니다. 본 문서는 마운트 포인트 생성 방법을 소개합니다.

작업 절차

1. CHDFS 콘솔에 로그인한 후, 왼쪽 메뉴에서 [파일 시스템]을 클릭하고, 해당 리전을 선택합니다. 예: 광저우
2. 작업할 CHDFS를 찾아 [설정]을 클릭하면 해당 기본 설정과 마운트 정보를 확인할 수 있습니다.
3. [마운트 포인트]>[마운트 포인트 추가]를 선택한 후, 마운트 포인트 추가 페이지에서 이름을 입력하고 VPC 네트워크와 권한 그룹을 지정합니다.

- 이름: 마운트 포인트 이름으로 대소문자 알파벳, 숫자, - 또는 _ 조합만 지원하며 길이는 4~64자 입니다.
- VPC 네트워크 이름/ID: VPC 네트워크 선택.
- 권한 그룹: 권한 그룹을 선택하고 아직 생성되지 않았으면 권한 그룹 생성을 선택합니다.



3. [저장]을 클릭하면 마운트 포인트가 생성됩니다.

CHDFS 마운트

최종 업데이트 날짜: : 2022-03-30 09:30:26

CHDFS와 마운트 포인트 생성 후, 마운트 포인트를 통해 CHDFS를 마운트할 수 있습니다. 본 문서는 CHDFS 마운트 방법을 소개합니다.

전제 조건

- 마운트할 기기 또는 컨테이너에 Java 1.8이 설치되어 있어야 합니다.
- 마운트할 기기 또는 컨테이너의 VPC와 마운트 포인트 지정 VPC가 같아야 합니다.
- 마운트할 기기 또는 컨테이너의 VPC IP가 마운트 포인트 지정 권한 그룹중 하나의 권한 규칙 라이선스 주소에 매칭되어야 합니다.

작업 절차

1. CHDFS-Hadoop JAR 패키지를 다운로드합니다.

2. JAR 패키지를 해당 디렉터리로 이동합니다. EMR 클러스터의 경우 모든 노드의 `/usr/local/service/hadoop/share/hadoop/common/lib/` 디렉터리로 동기화할 수 있습니다.

3. core-site.xml 파일을 편집하고 다음 기본 설정을 추가합니다.

```
<!--chdfs 구현 유형-->
<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>
<!--로컬 cache의 임시 디렉터리로, 데이터 읽기/쓰기의 경우, 메모리 cache 부족 시, 로컬 디스크에 쓰기 하며, 이 경로가 존재하지 않을 경우 자동 생성됩니다. -->
<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. core-site.xml을 모든 hadoop 노드로 동기화 합니다.

> 설명 :
> EMR 클러스터의 경우, 앞의 절차 3, 4를 통해 EMR 콘솔 컴포넌트 관리에서 HDFS 설정을 수정할 수 있습니다.

5. hadoop fs 명령 라인 툴을 사용하여, `hadoop fs -ls ofs://${mountpoint}/` 명령을 실행하며, 이 때 mountpoint는 마운트 주소입니다. 파일 리스트가 정상적으로 표시되면 CHDFS 마운트가 완료되었음을 의미합니다.

6. 사용자도 hadoop 기타 설정 항목 또는 mr 작업을 통해 CHDFS에서 데이터 작업을 실행할 수 있습니다. mr 작업의 경우, `-Dfs.defaultFS=ofs://${mountpoint}/` 를 통해 이번 작업의 기본 입출력 FS를 CHDFS로 수정할 수 있습니다.

기타 설정 항목

설정 항목	설명	기본값	필수 입력 여부
fs.ofs.tmp.cache.dir	임시 데이터 저장	없음	예
fs.ofs.map.block.size	chdfs 파일 시스템의 block 크기, 단위는 바이트. 기본값은 128MB(map 분할에만 영향을 주며, chdfs 기본 스토리지 블럭 사이즈와 무관)	134217728	아니오
fs.ofs.data.transfer.thread.count	chdfs 데이터 전송 시의 병행 스레드 수	32	아니오
fs.ofs.block.max.memory.cache.mb	chdfs 플러그 인이 사용하는 메모리 buffer 크기, 단위는 MB. (읽기/쓰기에 모두 가속 효과)	16	아니오
fs.ofs.block.max.file.cache.mb	chdfs 플러그 인이 사용하는 디스크 buffer 크기, 단위는 MB임. (읽기/쓰기에 모두 가속 효과)	256	아니오

설정 항목	설명	기본값	필수 입력 여부
fs.ofs.prev.read.block.count	읽기 할 때, 미리 읽어 오는 chdfs block 수량 (chdfs의 기본 block 크기는 일반적으로 4MB 임)	4	아니오
fs.ofs.plugin.info.log	플러그 인의 디버깅 로그 출력 여부, 로그는 info 레벨로 출력. 선택값은 true、false	false	아니오

CAM 라이선스로 액세스

최종 업데이트 날짜: : 2022-03-30 09:30:26

CHDFS 정책 사전 설정

CHDFS 라이선스 정책 사전 설정은 다음과 같습니다.

정책	설명
QcloudCHDFSReadOnlyAccess	CHDFS 읽기 전용 액세스 권한
QcloudCHDFSFullAccess	CHDFS 관리 권한

CHDFS 라이선스 작업

Action	Resouce	설명
chdfs:CreateFileSystem	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/*	CHDFS 생성
chdfs:DeleteFileSystem	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	CHDFS 삭제
chdfs:ModifyFileSystem	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	CHDFS 속성 수정
chdfs:DescribeFileSystem	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	CHDFS 상세 정보 조회
chdfs:DescribeFileSystems	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	CHDFS 리스트 조회
chdfs:CreateMountPoint	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	마운트 포인트 생성
chdfs:DeleteMountPoint	qcs::chdfs:${region-id}:uin/${account-uin}:mountpoint/${mount-point-id}	마운트 포인트 삭제
chdfs:ModifyMountPoint	qcs::chdfs:${region-id}:uin/${account-uin}:mountpoint/${mount-point-id}	마운트 포인트 속성 수정
chdfs:DescribeMountPoint	qcs::chdfs:${region-id}:uin/${account-uin}:mountpoint/${mount-point-id}	마운트 포인트 상세 정보 조회

Action	Resouce	설명
chdfs:DescribeMountPoints	qcs::chdfs:${region-id}:uin/${account-uin}:mountpoint/${mount-point-id}	마운트 포인트 리스트 조회
chdfs:AssociateAccessGroups	qcs::chdfs:${region-id}:uin/${account-uin}:mountpoint/${mount-point-id}	권한 그룹 리스트 바인딩
chdfs:DisassociateAccessGroups	qcs::chdfs:${region-id}:uin/${account-uin}:mountpoint/${mount-point-id}	권한 그룹 리스트 바인딩 해제
chdfs:CreateAccessGroup	qcs::chdfs:${region-id}:uin/${account-uin}:vpc/${vpc-id} 또는 qcs::chdfs:${region-id}:uin/${account-uin}:unVpcId/${unVpcId}	권한 그룹 생성
chdfs:DeleteAccessGroup	qcs::chdfs:${region-id}:uin/${account-uin}:accessgroup/${access-group-id}	권한 그룹 삭제
chdfs:ModifyAccessGroup	qcs::chdfs:${region-id}:uin/${account-uin}:accessgroup/${access-group-id}	권한 그룹 속성 수정
chdfs:DescribeAccessGroup	qcs::chdfs:${region-id}:uin/${account-uin}:accessgroup/${access-group-id}	권한 그룹 상세 정보 조회
chdfs:DescribeAccessGroups	qcs::chdfs:${region-id}:uin/${account-uin}:accessgroup/${access-group-id}	권한 그룹 리스트 조회
chdfs:CreateAccessRules	qcs::chdfs:${region-id}:uin/${account-uin}:accessgroup/${access-group-id}	권한 규칙 일괄 생성
chdfs:DeleteAccessRules	qcs::chdfs:${region-id}:uin/${account-uin}:accessrule/${access-rule-id}	권한 규칙 일괄 삭제
chdfs:ModifyAccessRules	qcs::chdfs:${region-id}:uin/${account-uin}:accessrule/${access-rule-id}	권한 규칙 속성 일괄 수정
chdfs:DescribeAccessRules	qcs::chdfs:${region-id}:uin/${account-uin}:accessgroup/${access-group-id}	권한 규칙 리스트 조회
chdfs:CreateLifeCycleRules	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	라이프사이클 규칙 일괄 생성

Action	Resouce	설명
chdfs:DeleteLifeCycleRules	qcs::chdfs:${region-id}:uin/${account-uin}:lifecyclerule/${life-cycle-rule-id}	라이프사이클 규칙 일괄 삭제
chdfs:ModifyLifeCycleRules	qcs::chdfs:${region-id}:uin/${account-uin}:lifecyclerule/${life-cycle-rule-id}	라이프사이클 규칙 속성 일괄 수정
chdfs:DescribeLifeCycleRules	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	라이프사이클 규칙 리스트 조회
chdfs:CreateRestoreTasks	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	복구 작업 일괄 생성
chdfs:DescribeRestoreTasks	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	복구 작업 리스트 조회
chdfs:ModifyResourceTags	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	리소스 태그 리스트 수정
chdfs:DescribeResourceTags	qcs::chdfs:${region-id}:uin/${account-uin}:filesystem/${file-system-id}	리소스 태그 리스트 조회

CHDFS 라이선스 정책 예시

서브 계정에 CHDFS 관리 시스템 읽기 전용 권한을 부여하는 정책 예시는 다음과 같습니다.

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

서브 계정에 CHDFS 조회 권한을 부여하는 정책 예시는 다음과 같습니다.

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

Java 코드로 CHDFS 액세스

최종 업데이트 날짜: : 2022-03-30 09:30:26

작업 시나리오

Cloud HDFS(CHDFS)의 JAR 패키지를 설치하면 명령어, 빅 데이터 컴포넌트 등 방식 외에도 Java 코드로 CHDFS에 액세스할 수 있습니다. 본 문서는 Java 코드로 CHDFS 에 액세스하는 방법을 소개합니다.

전제 조건

- CHDFS 관련 JAR 패키지 설치. 자세한 내용은 CHDFS 마운트를 참고하십시오.
- Java 프로그램을 실행할 기기가 마운트 포인트 권한 그룹에서 액세스를 허용한 VPC에 있어야 합니다.

작업 절차

1. maven 프로젝트를 생성하고 maven pom.xml에 다음 종속 항목(본인의 실제 hadoop 환경에 맞춰 hadoop-common 패키지 버전 설정)을 추가합니다.

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

2. 다음 hadoop 작업 코드를 참고하여 수정합니다. 설정 항목 및 관련 설명은 CHDFS 마운트를 참고하십시오.
일부 자주 사용되는 파일 시스템 작업 인터페이스는 다음과 같습니다. 기타 인터페이스는 Hadoop FileSystem 인터페이스 문서를 참고하십시오.

```
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 구성 항목은 https://cloud.tencent.com/document/product/1105/36368을 참고하십시오.
// 다음 설정은 필수 항목입니다.
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");
// 기타 옵션 설정 항목은 https://cloud.tencent.com/document/product/1105/36368을 참고하십시오.
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
{
// 파일 생성 (이미 있는 경우 덮어쓰기)
// if the parent dir does not exist, fs will create it!
FSDataOutputStream out = fs.create(filePath, true);
try {
// 파일 입력
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 IOExc
eption {
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);
}

// 기본 인증 유형은 COMPOSITE-CRC32C임.
private static void getFileCheckSum(FileSystem fs, Path path) throws IOExceptio
n {
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 local
Path) throws IOException {
fs.copyFromLocalFile(localPath, chdfsPath);
}

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

private static void renamePath(FileSystem fs, Path oldPath, Path newPath) throw
s 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(), dirM
ember.getLen());
}
}

// 재귀 삭제 마크는 디렉터리 삭제에 사용됩니다.
// 재귀가 false 이고 dir가 비어있지 않으면 작업이 수행되지 않습니다.
private static void deleteFileOrDir(FileSystem fs, Path path, boolean recursiv
e) throws IOException {
fs.delete(path, recursive);
}

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

public static void main(String[] args) throws IOException {
// 파일 초기화
FileSystem fs = initFS();

// 파일 생성
Path chdfsFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, chdfsFilePath);

// 파일 불러오기
readFile(fs, chdfsFilePath);

// 파일 또는 디렉터리 조회
queryFileOrDirStatus(fs, chdfsFilePath);

// 파일 검사합 가져오기
getFileCheckSum(fs, chdfsFilePath);

// 로컬에서 파일 복사
Path localFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobject.
txt");
copyFileFromLocal(fs, chdfsFilePath, localFilePath);

// 파일을 로컬로 가져오기
Path localDownFilePath = new Path("file:///home/hadoop/ofs_demo/data/exampleobj
ect.txt");
copyFileToLocal(fs, chdfsFilePath, localDownFilePath);
```

```
// 이름 변경
Path newPath = new Path("/doc/example.txt");
renamePath(fs, chdfsFilePath, newPath);

// 파일 삭제
deleteFileOrDir(fs, newPath, false);

// 디렉터리 생성
Path dirPath = new Path("/folder");
mkdir(fs, dirPath);

//디렉터리에 파일 생성
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);

// 디렉터리 나열
listDirPath(fs, dirPath);

// 디렉터리 삭제
deleteFileOrDir(fs, dirPath, true);

//파일 시스템 비활성화
closeFileSystem(fs);
}
}
```

3. 컴파일 및 실행

설명 :

- 실행 전에 classpath를 정확히 설정했는지 확인하십시오. classpath에는 Hadoop common 패키지 및 CHDFS 패키지 경로가 포함되어야 합니다.
- EMR 환경의 경우, CHDFS 마운트에 따라 작업하면 Hadoop common 패키지는 일반적으로 `/usr/local/service/hadoop/share/hadoop/common/` 디렉터리에, CHDFS 패키지는 `/usr/local/service/hadoop/share/hadoop/common/lib/` 디렉터리에 위치합니다.

파일 시스템 삭제

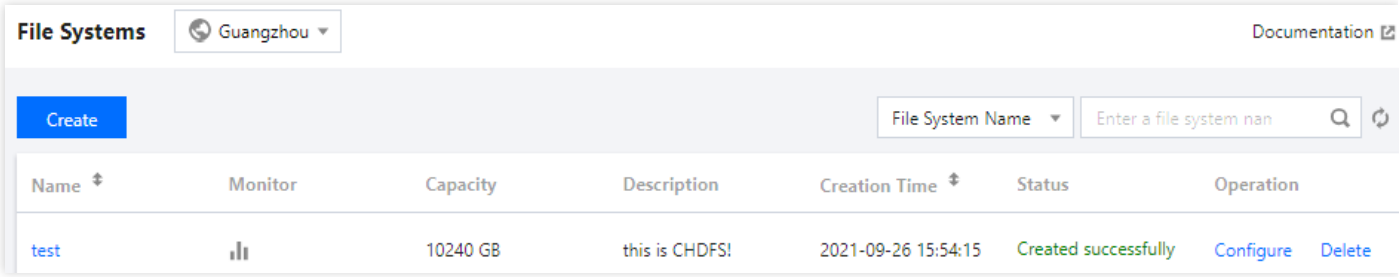
최종 업데이트 날짜: : 2022-03-30 09:30:26

본 문서는 CHDFS 콘솔에서 파일 시스템 삭제 방법을 소개합니다.

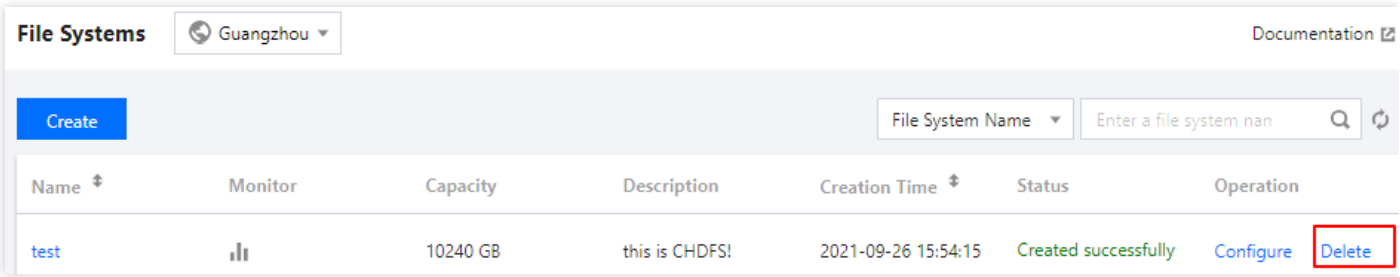
> 주의 :
> 파일 시스템을 삭제하면, 해당 시스템의 모든 설정이 삭제되며 복구할 수 없으니 신중히 작업하시기 바랍니다.

작업 절차

1. CHDFS 콘솔에 로그인합니다.
2. 왼쪽 메뉴에서 [파일 시스템]을 클릭합니다.
3. 파일 시스템 인터페이스에서 파일 시스템이 있는 리전을 선택합니다. 예: 광저우



4. 파일 시스템 리스트에서 삭제할 파일 시스템을 찾아 파일 시스템 우측의 [삭제]를 클릭합니다.



5. 팝업창에서 [삭제]를 클릭하면 파일 시스템이 삭제됩니다.