

CFS 文档



【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。 未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。 本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

【免责声明】

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。 如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

目录	
目录	2
动态与公告	5
功能发布记录	5
产品公告	6
关于CFS监控指标调整公告	6
产品描述	6
产品介绍	6
基本概念	7
产品特性	7
产品规格	8
使用限制与建议	8
应用场景	9
选型指导	11
如何选择CFS和PFS	11
如何选用CFS、BOS和CDS	12
产品定价	13
按需计费	13
存储包计费	13
余额不足提醒和欠费处理	15
计费案例	16
快速入门	18
入门概述	18
快速入门（Linux）	19
快速入门（Windows）	22
操作指南	25
开通服务	25
管理文件系统	26
创建文件系统	26
删除文件系统	27
查看文件系统详情	27
设置文件系统容量上限	29
挂载访问	30
添加挂载点	30
BCC挂载与卸载CFS	31
CCE使用CFS	41
权限组管理	41
多用户访问控制	43
数据加密	47
管理标签	48
备份	52

Baidu 百度智能云文档	目录
监控与报警	54
查看监控	54
报警管理	56
指标含义	58
云审计	59
典型实践	60
为不同目录设置不同的用户权限	60
跨地域或账号使用文件系统	61
使用SFTP上传下载CFS文件系统数据	66
使用rsync从旧文件系统同步到CFS文件系统	67
使用Terraform管理CFS最佳实践	69
性能测试方法	72
Linux系统性能测试方法	73
Windows系统性能测试方法	75
API参考	78
API功能更新记录	78
API概述	79
接口概览	79
通用说明	79
服务域名	81
错误码	81
文件系统相关接口	84
创建文件系统	84
更新文件系统	86
查询文件系统	87
查询挂载客户端	89
释放文件系统	91
更新文件系统标签	92
挂载点相关接口	93
创建挂载点	93
查询挂载点	94
删除挂载点	96
数据类型	97
Go-SDK	98
概述	98
初始化	98
使用STS创建CFS Client	99
设置网络参数	101
配置生成签名字符串选项	101
文件系统	102

Baidu 百度智能云文档	动态与公告
挂载点	103
错误处理	105
常见问题	105
常见问题总览	105
一般类问题	106
计费类问题	107
NFS协议问题	108
SMB协议问题	114
服务等级协议SLA	115
CFS服务等级协议（SLA）	115

动态与公告

功能发布记录

发布时间	功能概述
2024-06	容量型存储包 支持更多购买时长（6个月、1年）和规格（50T、100T、200T）；优化存储包购买交互体验。
2024-04	优化读/写吞吐、读/写延迟、读/写io和元数据读/写io等8项 监控指标 ，为用户提供更精确的监控指标数值。
2024-02	支持对挂载客户端进行 读写流量查询、排序 ；新增文件系统容量使用率监控指标。
2023-12	发布CFS GO SDK 。
2023-09	支持通过 云助手批量挂载/卸载CFS 。
2023-06	监控报警能力升级，新增读/写延时、挂载/卸载错误统计、元数据io等 监控指标 。
2022-11	支持 标签管理 ，可按照各种标准（如用途、所有者或项目）对资源添加标签，从而快速分类和管理资源。
2022-06	支持 CFS OpenAPI 。
2022-03	发布 CFS服务等级协议 （SLA）。
2021-01	发布文件存储新类型：性能型CFS，适用于高吞吐、业务弹性扩展、对时延有较高要求的文件共享存储服务。对于读写频繁、系统响应要求高的业务，有高性能优势。
2020-10	发布文件存储新类型——容量型CFS，适用于大容量高吞吐并对时延响应要求不高的计算型业务。对于读写不频繁的大多数文件共享业务，有高性价比优势。
2020-07	CFS 支持用户在创建文件系统时开启 数据加密 ，并支持根据用户需要进行密钥变更。
2020-06	CFS 支持用户在创建文件系统时自定义容量上限，并支持根据业务需求对容量进行扩/缩容操作。
2020-06	CFS 支持展示已挂载 客户端的信息列表 ，包括客户端IP、可用区、VPC ID、挂载点ID等信息。
2020-06	CFS 支持 备份 恢复功能，用户可根据需要对文件系统实例进行手动/自动备份、恢复操作。
2020-06	CFS 新增预付费（包年包月）计费模式，支持 存储包购买 并指定文件系统进行抵扣。
2020-01	CFS 支持 SMB/CIFS 协议，支持在Windows/Linux系统下 挂载SMB文件系统 。

2019-11	CFS 支持 权限组功能 ，支持用户对权限组和规则进行增删改查等管理操作，以实现IP/网段进行访问权限控制。
2019-07	新增“为不同目录设置不同的用户权限”和“跨地域或账号使用文件系统” 典型实践 。

产品公告

关于CFS监控指标调整公告

为了给您提供更好的监控告警服务，文件存储CFS对部分监控指标进行了优化。优化后的监控指标将于2024年4月10日19:00:00正式上线。届时，您将能够在[文件存储CFS控制台](#)和[云监控BCM控制台](#)查看到最新的监控指标，以获取更精准的监控指标数据。

调整时间

2024年4月10日19:00:00

调整区域

全部区域

调整内容

优化前			优化后		
指标中文名称	指标英文名称（Metric name）	单位	指标中文名	指标英文名称（Metric name）	单位
读吞吐	ReadThroughput	bytes/min	读吞吐	ReadThroughputNew	bytes/s
读延迟	ReadLatency	ms	读延迟	ReadLatencyNew	ms
读io	ReadIO	次每分钟	读iops	ReadIONew	次每秒
写吞吐	WriteThroughput	bytes/min	写吞吐	WriteThroughputNew	bytes/s
写延迟	WriteLatency	ms	写延迟	WriteLatencyNew	ms
写io	WriteIO	次每分钟	写iops	WriteIONew	次每秒
元数据读io	MetaReadQps	次每秒	元数据读qps	MetaReadQpsNew	次每秒
元数据写io	MetaWriteQps	次每秒	元数据写qps	MetaWriteQpsNew	次每秒

调整影响

优化后的监控指标数值将更加精确。其中：

- ① 读吞吐和写吞吐的单位由“bytes/min”优化为“bytes/s”，读IO和写IO的单位由“次每分钟”优化为“次每秒”。由于统计口径的变更，监控指标的数值将会变小。
- ② 读延迟、写延迟、元数据读IO、元数据写IO等监控指标的单位没有变更，但对精确度进行了提升。

如果您在云监控BCM中设置了上述优化后的监控指标，建议您参考 [报警管理](#) 文档，并根据实际业务情况及时调整设置的报警规则阈值大小，以防止报警失灵造成业务损失。

产品描述

产品介绍

百度智能云文件存储（Cloud File System）是一种安全高效的文件存储服务，为云上的虚机、容器资源提供了跨操作系统的文件存储及共享能力。CFS基于POSIX接口适配原生操作系统，兼容NFS协议和SMB协议。同时，百度智能云CFS提供简单、易操

作的对外接口，并支持按实际使用量计费，免去部署、维护费用的同时，最大化提升您的业务效率。



CFS支持丰富的应用场景，详情参见[应用场景](#)。

CFS提供了性能型和容量型存储类型，更多详情参见[产品规格](#)。

基本概念

文件存储

文件存储是云存储的一种，为云主机、Docker容器等提供标准文件访问接口（NFS、CIFS）的云存储服务，具备无限容量、高性能、多客户端共享、高可用等特性。

文件系统

文件系统是操作系统用来组织和管理存储于物理介质上数据的一种手段。

NFS协议

NFS（NetworkFileSystem）协议主要用于 Linux 及 Unix 客户端，CFS目前支持NFS V4.1协议。

SMB/CIFS协议

SMB（Server Message Block）协议是微软推出的共享网络资源的应用层通信协议，CIFS(Common Internet File System) 协议则是公共的或开放的 SMB 协议版本，其可以更好的支持 Windows 客户端的访问。CFS目前支持SMB 1、SMB 2和SMB 3等版本的主要功能。

挂载点

文件系统对于用户在不同专有网络或经典网络的访问需求，可分别生成专门的访问地址，每个访问地址即为一个挂载点，用户挂载时通过指定挂载点的域名来挂载对应的CFS文件系统到本地。

权限组

权限组用于配置通过不同挂载点访问文件系统的权限信息，包括授权IP地址、读写操作权限等。

产品特性

弹性扩展

CFS具有Scale-Out横向扩展能力，单文件系统最大支持100PB，最多支持1000亿个文件。

共享访问

多个计算节点可通过同时挂载一套文件系统，实现操作系统间的文件共有存储及高效共享。

协议兼容

CFS支持标准的NFS和SMB协议，支持Linux和Windows客户端使用。

安全控制

具备安全鉴权机制，支持VPC隔离及IP权限控制，保证数据安全。

数据加密

对文件系统中存储的数据进行自动加密解密，保证用户数据安全。

数据备份

支持用户通过自定义策略创建多个备份数据，在数据发生损坏时进行数据恢复。

稳定可靠

系统设计可用性达到99.9%，数据可靠性达到99.99999999%（10个9），保证数据安全可靠。

产品规格

文件存储CFS分为性能型和容量型两种。下表详细介绍了两种类型的特点、性能规格和应用场景，您可以参考以下参数根据业务需求选择合适的文件存储类型。

项目	性能型CFS	容量型CFS
容量	0~100PB	0~100PB
最大IOPS	300K	15K
最大吞吐	40GB/s（随容量线性扩展）	10GB/s（随容量线性扩展）
吞吐量计算公式	吞吐量（MB/s）= 存储量（GB）x 0.6 + 600	吞吐量（MB/s）= 存储量（GB）x 0.15 + 150
IOPS计算公式	IOPS = 存储量（GB）x 3 + 30K	-
时延	毫秒级	10毫秒级
应用场景	适用于高吞吐、业务弹性扩展、对时延有较高要求的文件共享存储服务。对于读写频繁、系统响应要求高的业务，有高性能优势。	适用于大容量高吞吐并对时延响应要求不高的计算型业务。对于读写不频繁的大多数文件共享业务，有高性价比优势。

使用限制与建议

使用限制

CFS产品有如下配额限制：

限制项	限额
单用户可创建文件系统数	20个
单个文件系统容量	100PB
单个文件系统文件数	1000亿个
单个文件大小	1PB
目录深度	1000级
软链接路径长度	4080字节
文件名长度	255字节
同时连接的客户端数量	数千个

使用建议

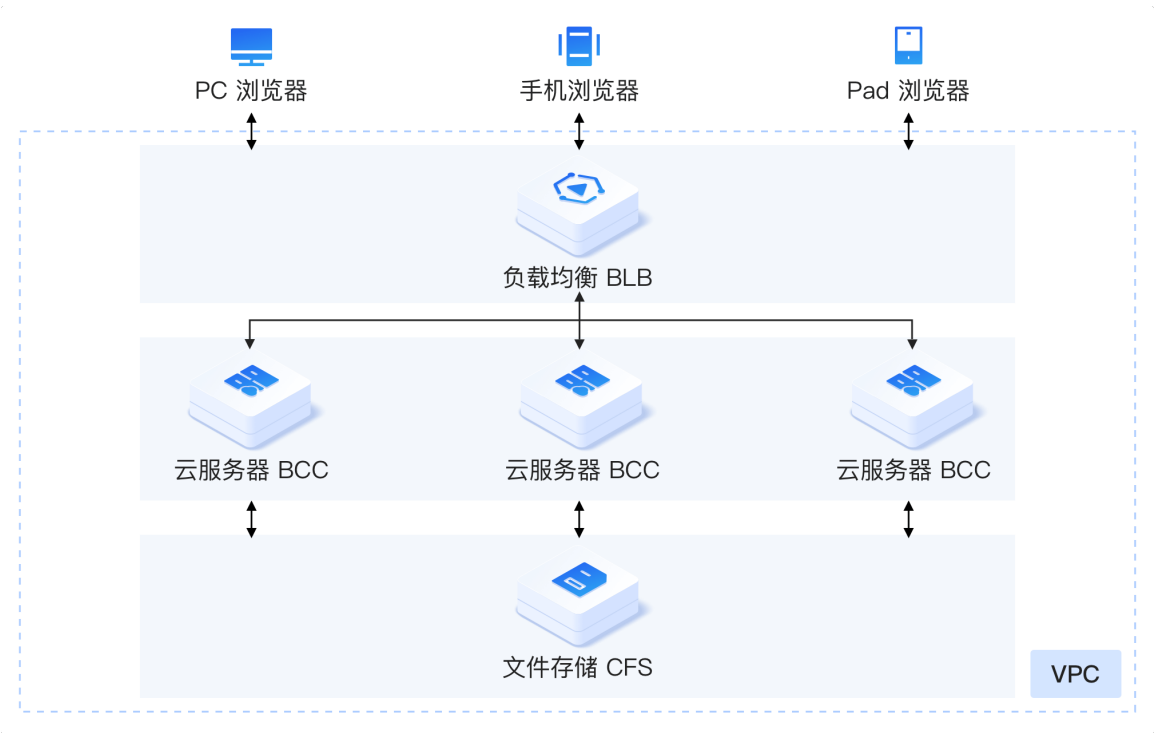
1.单目录下最大文件数或子目录数不要超过1000万。

说明：当单层目录下放置超过1000万的文件或子目录时，操作整个目录可能会导致NFS协议向服务端发送大量遍历请求，从而引发排队，导致请求耗时长或请求失败。

应用场景

内容管理及Web应用

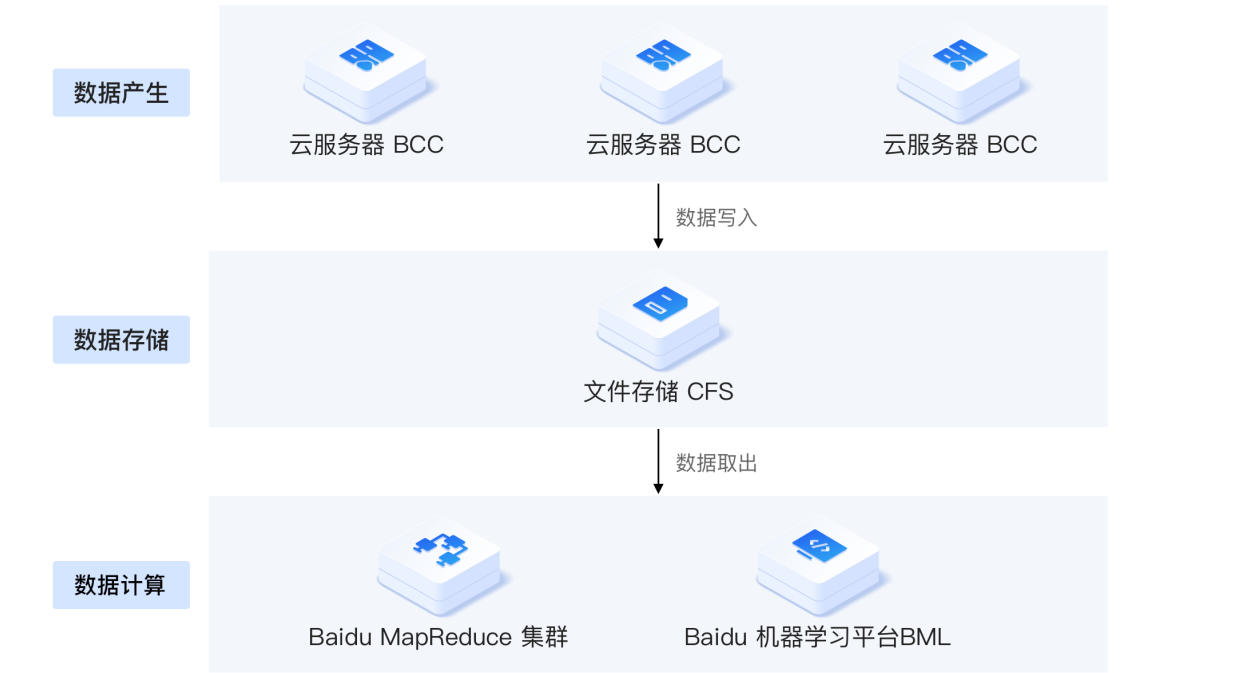
在大型网站、游戏、视频等领域，内容往往由服务器集群来提供，其中每台服务器都需要访问同一组文件时，则可通过挂载同一个百度智能云CFS的形式对相同的文件进行读写。而CFS高可扩展性、高可用性的特点，也可充分满足前端服务器的高IO读写的需要。



相关产品：[云服务器BCC](#)、[负载均衡](#)

大数据分析

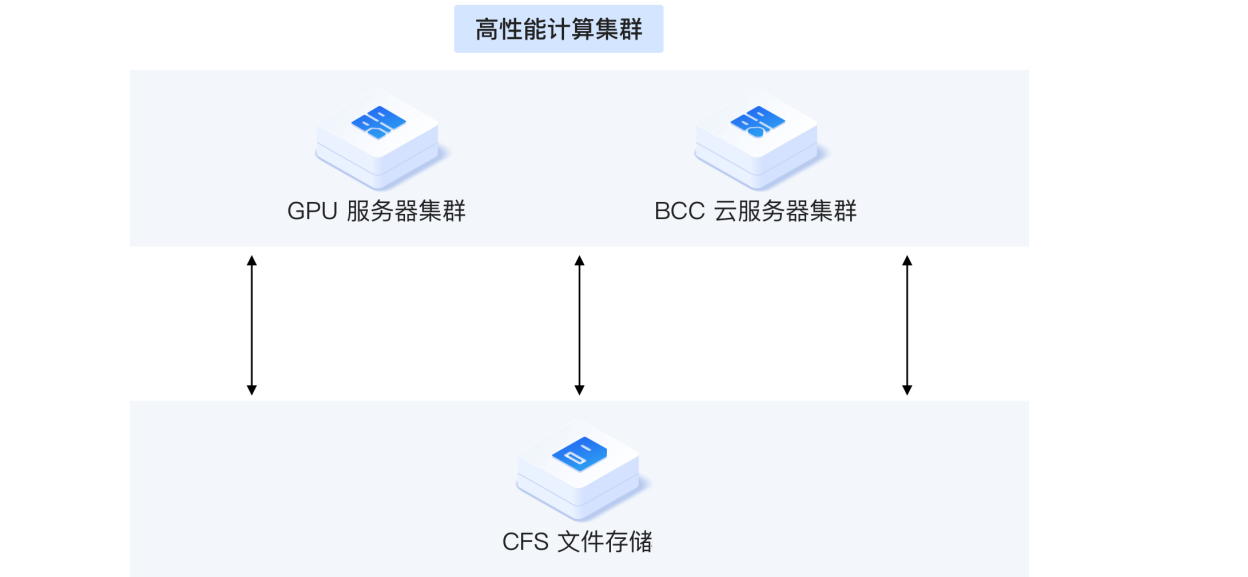
百度智能云CFS满足常见大数据应用在计算节点高吞吐量、写后读一致性及低延迟的需求，提供极易集成于大数据工作流的文件共享解决方案，是服务器日志分析和处理的最佳方案。



相关产品：[MapReduce](#)、[云服务器BCC](#)、[千帆BML全功能AI开发平台](#)

泛娱乐流媒体处理

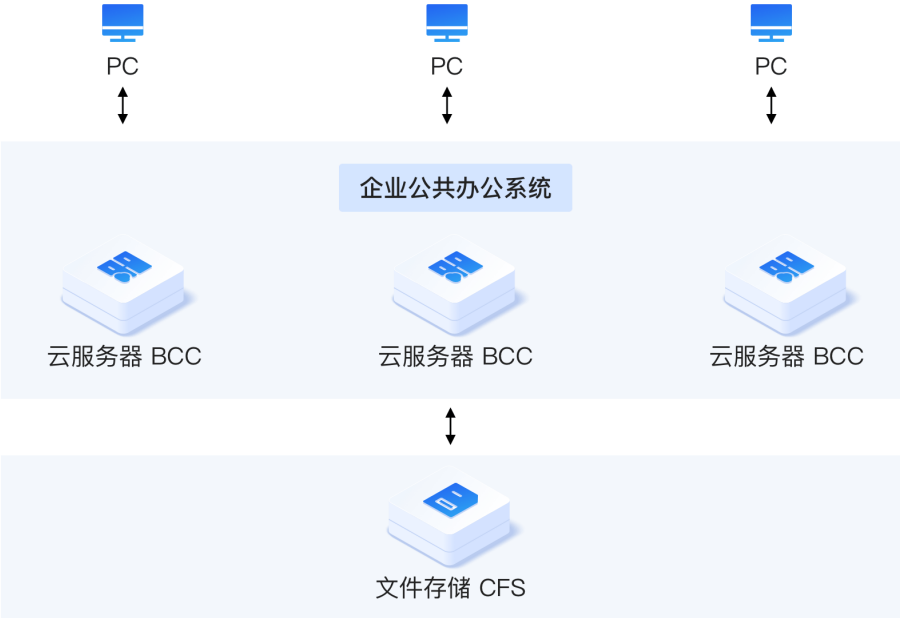
在音视频后期制作、广播媒体处理和渲染等场景中，一个典型的特点是待处理文件较大，多台服务器共同处理。百度智能云CFS提供了对大文件快速共享访问的能力。



相关产品：[GPU云服务器](#)、[云服务器BCC](#)

企业办公文件共享

在企业内部，员工需要从不同的终端访问企业公共办公系统中的同一文件，管理员可将各企业办公系统挂载到百度智能云CFS上，为企业提供统一共享的文件系统。



相关产品：[云服务器BCC](#)

选型指导

如何选择CFS和PFS

百度智能云文件存储 CFS 和并行文件存储 PFS 提供了多种规格的文件系统，本文从规格、计费、性能、功能、协议/接口、使用限制、应用场景等方面介绍了 CFS 和 PFS 的对比情况，您可以根据实际业务需求选择合适的文件系统。

类别		文件存储 CFS		并行文件存储 PFS	
概述		类 NAS 存储产品，支持标准文件访问协议（NFS 和 SMB），主要用于数据共享访问场景。		高性能文件存储，提供私有客户端，支持	
规格		容量型	性能型	标准型	增强型
计费	月单价(以30天为例)	0.35元/GB/月	1.8元/GB/月	0.83元/GB/月	1.4元/GB/月
		CFS定价详情及计费规则请参见： 文件存储CFS产品定价页 PFS定价详情及计费规则请参见： 并行文件存储PFS产品定价页			
	计量方式	Serverless，完全按使用量计费，弹性扩缩容		托管，有起步容量与扩容步长要求，按则	
	起步容量与扩容步长	无要求，最大容量100PB		起步容量：7TB 扩容步长：7TB 最大容量：200TB	起步容量：7TB 扩容步长：3TB 最大容量：1PB
性能	最大吞吐	随存储使用量线性扩展，最高可达10GB/s，详见 CFS产品规格	随存储使用量线性扩展，最高可达40GB/s，详见 CFS产品规格	随存储购买容量线性扩展（100MBps/TB），最高可达20GB/s，详见 PFS产品规格	随存储购买容量线性扩展（200MBps/TB），最高可达20GB/s，详见 PFS产品规格
	最大IOPS	15K，详见 CFS产品规格	随存储使用量线性扩展，最高可达30万，详见 CFS产品规格	随存储购买容量线性扩展（15K/TB），最高可达300万，详见 PFS产品规格	随存储购买容量线性扩展（30K/TB），最高可达300万，详见 PFS产品规格
	最低时延	10ms级	ms级	亚ms级	
功能		支持备份、监控报警、权限组管理、 云审计 等		支持数据流动、监控报警、 云审计 等	
协议/接口		NFS 4.1（支持 Linux） SMB 1、SMB 2和SMB 3（支持 Windows）		POSIX（支持 Linux）	
使用限制		参见 CFS使用限制		参见 使用限制 （标准型、增强型和极速型）	
适用场景		适用于大容量高吞吐并对时延响应要求不高的计算型业务。对于读写不频繁的大多数文件共享业务，有高性价比优势。	适用于高吞吐、业务弹性扩展、对时延有较高要求的文件共享存储服务。对于读写频繁、系统响应要求高的业务，有高性能优势。	AI 训练与推理、自动驾驶、高性能计算、	

更多产品介绍和操作见：[文件存储CFS](#) 和 [并行文件存储PFS](#)

如何选用CFS、BOS和CDS

本文介绍文件存储CFS与对象存储BOS、云磁盘CDS的区别，帮助您更好地选用合适的百度智能云存储服务，主要区别如下表所示。

对比项	文件存储CFS	对象存储BOS	云磁盘CDS
特点	文件存储是一种基于层级目录结构的存储方式，数据以文件的形式保存，并且通过目录和文件路径进行访问，类似传统本地文件系统。它支持POSIX标准的操作，如读写权限管理等。文件存储CFS具备数据共享、高吞吐、高IOPS、文件随机读写等特点。	对象存储采用扁平结构，数据被存储为独立的对象，每个对象拥有唯一的标识符（ID），并可以附加丰富的自定义元数据，这种结构使对象存储具备很高的扩展性和灵活性。对象存储BOS支持公网访问，具备海量数据低成本共享和存储的特点，适合存储读多写少、写入后较少修改的数据，比如图片、视频、备份文件等。	云磁盘是一种块存储，类似物理硬盘，支持分区格式化并建立文件系统。云磁盘CDS具备高性能、低时延、高效随机读写能力，其数据会在可用区内以多副本冗余方式存储，避免数据的单点故障风险，同时支持在线扩容和在线变更磁盘类型等方式对CDS进行弹性管理。
应用场景	主要应用于内容管理及Web应用、大数据分析、泛娱乐流媒体处理和企业办公文件共享等场景。详情参见： CFS应用场景 。	主要应用于存储数据全球分发、冷数据归档存储、数据的容灾备份、高性能大数据计算、多媒体数据处理等场景。详情参见： BOS应用场景 。	主要应用于云服务存储、关键应用数据存储和复杂环境的一致性复制等场景。详情参见： CDS应用场景 。
访问模式	云上BCC云服务器、GPU云服务器、CCE容器等可通过POSIX接口进行数据读写。	客户端可通过SDK、API和周边工具等进行文件并发读取，追加写入。	CDS是裸磁盘，需要对磁盘进行分区、创建文件系统（如linux系统的ext4、Windows系统的NTFS）并挂载到BCC后才能被正常读写数据。
最低时延	毫秒级，详情参见： CFS产品规格 。	几十毫秒级，详情参见： BOS使用限制 。	微秒级，详情参见： 磁盘性能 。
最大吞吐	40 GB/s，详情参见： CFS产品规格 。	公网带宽阈值为 10 Gbit/s（约1 GB/s），内网带宽阈值为 50 Gbit/s（约6 GB/s）。详情参见： BOS使用限制 。	数GB/s，详情参见： 磁盘性能 。
最大IOPS/QPS	最大IOPS 300K，详情参见： CFS产品规格 。	单Bucket设置的上传QPS 1500q/s，下载QPS不限制；对单个热点Object上传QPS 为 200q/s，下载QPS 为1000q/s；详情参见： BOS使用限制 。	最大IOPS 100K，详情参见： 磁盘性能 。
容量	<100PB	无上限	GB~TB
协议/接口	NFS和SMB	HTTP/HTTPS	虚拟块设备协议接口

产品定价

按需计费

百度智能云文件存储CFS为用户提供了两种灵活的购买方式，按需计费或存储包计费。

按需计费服务采用后付费方式，根据您的实际使用情况，自动扣费并生成账单；另外，对于不同存储类型的文件系统，单价不同。定价详情及计费规则请参见：[文件存储CFS产品定价页](#)。

存储包计费

百度智能云文件存储CFS为用户提供了两种灵活的购买方式，按需计费或存储包计费。

存储包服务采用预付费方式，通过购买存储包，您可提前预留云上存储资源，同时享受价格优惠，更大程度上帮您降低资源成本。

目前支持先创建文件系统再绑定购买的存储包，也支持在购买存储包的时候创建新的文件系统并绑定。购买存储包以后，所绑定文件系统的消费会首先使用存储包进行抵扣，超出部分按量付费。

🔗 存储包购买

您可以直接登录[文件存储系统CFS控制台](#)，点击左侧导航“存储包”页签购买存储包。购买时按照提示选择存储包类型、地域、所属文件系统、规格、购买时长、生效日期和是否自动续费即可。具体购买界面及选择参数如下：

购买须知

购买说明

CFS 存储包包含容量型存储包和性能型存储包，根据售卖地域的不同，存储包的价格会有相应的不同。具体价格详细信息请参见 [文件存储CFS价格说明](#)。

抵扣规则

每种类型存储包仅可用于抵扣指定计费项，且地域专属资源包不支持跨地域抵扣。如果您订购了多个相同类型的资源包，则抵扣顺序为：优先抵扣先到期的资源包。

选购信息

* 存储包类型：

容量型存储包

性能型存储包

用于抵扣使用文件存储时产生的容量存储费用，仅适用于性能型文件系统。

* 地域选择：

华北 - 北京

华北 - 保定

华北 - 阳泉

西南 - 成都

华南 - 广州

华东 - 苏州

华东 - 南京

华中 - 武汉

中国香港

* 所属文件系统：

dadsaq2261111 (cfs-enZ69Rdfv8) 已用64.00 GB

▼

存储包必须与一个文件系统绑定，且必须在同一个地域。可选择一个已存在的文件系统ID或 [创建新文件系统](#)。

* 规格：

10GB

50GB

100GB

500GB

1TB

5TB

10TB

50TB

100TB

200TB

当前绑定的文件系统已使用量为 64.00 GB

* 购买时长：

1个月

6个月

1年

* 规格说明：

数据总量：100GB。

自存储包生效起，时长 1年 内，您可以在CFS中存储 100GB 性能型存储包 存储数据，而不需要支付额外的容量存储包存储费用。超出的存储量部分将自动按量计费。

* 生效日期：

☒ 支付后立即生效

☐ 指定时间生效

* 自动续费：

开启

关闭

🔗

存储包示意图：

性能型存储包 100GB

2024-08-08 10:36:02

购买1年

2025-08-08 10:36:02

参数	说明
存储包类型	选择存储包的类型。包括性能型和容量型，不同类型的存储包适用于不同类型的文件系统。性能/容量型存储包只能绑定性能/容量型文件系统。
地域选择	选择存储包的所属地域。存储包只能绑定所属地域的文件系统。
所属文件系统	存储包必须选择一个文件系统进行绑定，且一个存储包仅可以绑定一个文件系统。可选择已经存在的文件系统或创建一个新的文件系统后进行绑定，务必注意存储包类型与绑定文件系统类型的一致性。
规格	根据需求选择需要购买的存储包规格。绑定的文件系统实际存储量超过存储包规格容量的部分进行按量收费。
购买时长	根据需求选择需要购买的存储包时长。存储包到期后，进行按量付费。
生效日期	根据需求选择需要购买的存储包生效时间。支持立即生效和指定时间（精确到分钟粒度）生效。
自动续费	是否开启自动续费。若开启自动续费，则该存储包到期时，系统会自动为您续费具有相同规格和有效期的存储包。

🔗 存储包价格

存储包包括：性能型存储包和容量型存储包。根据售卖地域的不同，存储包的价格会有相应的不同。具体价格详细信息请参见[文件存储CFS价格说明](#)

🔗 存储包状态

15

常见存储包状态如下：

存储包状态	状态说明
创建中	购买的存储包正在创建中
生效中	购买的存储包生效中，可抵扣绑定文件系统存储费用
待生效	购买的存储包待生效，到指定生效时间后将抵扣绑定文件系统存储费用
已失效	绑定的文件系统删除后，存储包会自动失效且无法绑定新的文件系统
已过期	购买的存储包已过期，您可以再次购买相同配置的存储包。
已注销	已购买的存储包发生退款或者存储包过期7天后，会被注销

🔗 计费规则

1. 存储包为预付费方式，需要一次性支付。
2. 有效期：购买存储包时，可选择立即生效或者指定时间（精确到分钟粒度）生效。
3. 在存储包生效期间，绑定存储包文件系统的费用会首先使用存储包进行抵扣，超出部分按量付费。举例：
 - 用户当前有两个 100 GB 的存储包处于生效状态，且绑定了文件系统 A，当前文件系统 A 的使用容量为 180 GB，由于 $180\text{ GB} < 200\text{ GB}$ ，用量在存储包规格之内，所以该计费项以预先购买的存储包抵扣即可；
 - 用户当前有两个 100GB 的存储包处于生效状态，且绑定了文件系统 A，当前文件系统 A 的使用容量为 250 GB，由于 $250\text{GB} > 200\text{ GB}$ ，所以需要收取超出的 50 GB 的费用，并按照“按量付费”的标准计费。
4. 如果存储包是在某一个小时中间开通，在下个整点账单时，存储包会按照时间比例抵扣，不会完整抵扣。
5. 用户可以购买多个存储包，同一时间段也可以有多个存储包处于生效状态。
6. 官方退款：用户购买存储包7天内且未使用（即未到达用户设置的生效时间）则可全额退款；若已使用（已生效）或超出 7 天，则不支持退款。
7. 到期提醒：到期的存储包立即失效，到期前7天和到期时发送用户提醒消息。

余额不足提醒和欠费处理

🔗 余额不足提醒

- 根据您最近3天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来3天的费用，若不足以支付，系统发送续费提醒。
- 根据您最近1天的账单金额来判断您的账户余额（含可用代金券）是否足以支付未来1天的费用，若不足以支付，系统发送续费提醒。

🔗 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次CFS账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后立即停服，系统会发送欠费停服通知。
- 欠费停服后停止运行的文件系统将为您保留15天，15天内未充值则释放，释放时会发送释放通知。
- 待账户充值成功且账户余额>0后，CFS服务将恢复正常运行并重新计费。

🔗 欠费停服影响

计费方式	是否绑定了存储包	存储包状态	欠费影响
按需计费	否	不涉及	对于未绑定存储包的文件系统，文件系统处于按需计费模式，因当前账号已欠费，无法支付文件系统按需计费产生的费用，文件系统状态会立即转为停止运行，期间您仅能对该文件系统查询、删除、编辑标签等基础操作，除此之外无法执行其他任何操作。
按需计费+包年包月	是	生效中且未用尽	对于已绑定存储包且存储包未用尽的文件系统，您已经预先支付了存储包的费用。在账号欠费情况下，该文件系统状态仍为运行中，可正常使用，但您无法进行新建CFS、备份和购买存储包等操作。
	是	生效中且已用尽	对于已绑定存储包但存储包已用尽的文件系统，因当前账号已欠费，无法支付超出存储包抵扣以外的按需计费产生的费用，文件系统状态会立即转为停止运行，期间您仅能对该文件系统查询、删除、编辑标签等基础操作，除此之外无法执行其他任何操作。
	是	未生效或已过期	对于已绑定存储包但存储包未生效或已过期的文件系统，当前存储包无法抵扣费用，文件系统处于按需计费模式，因当前账号已欠费，无法支付文件系统按需计费产生的费用，文件系统状态会立即转为停止运行，期间您仅能对该文件系统查询、删除、编辑标签等基础操作，除此之外无法执行其他任何操作。

计费案例

本文以案例形式介绍文件存储CFS费用的计算方法。

说明：以下计费示例仅供参考，实际价格以您CFS购买页面为准。

案例1：存储容量恒定型

容量型

陈先生在华北-北京可用区A创建了一个容量型的文件系统。假设在2024年9月份期间每小时的最大存储容量均为100 GB，以下分别以按需付费和存储包抵扣介绍陈先生9月的消费情况。

计费方式	费用情况	说明
按需计费	100 GB x 0.000486元/GB/小时 x 24小时 x 30天 = 35.00元	
存储包抵扣	30元	购买1个1个月时长的100GB容量型存储包

性能型

陈先生在华北-北京可用区A创建了一个性能型的文件系统。假设在2024年9月份期间每小时的最大存储容量均为100 GB，以下分别以按需付费和存储包抵扣介绍陈先生9月的消费情况。

计费方式	费用情况	说明
按需计费	100 GB x 0.0025元/GB/小时 x 24小时 x 30天 = 180.00元	
存储包抵扣	165元	购买1个1个月时长的100GB性能型存储包

容量型+性能型

陈先生在华北-北京可用区A创建了一个性能型的文件系统和一个容量型的文件系统。假设在2024年9月份期间性能型的文件系统每小时的最大存储容量均为100 GB，容量型的文件系统每小时的最大存储容量均为200GB，以下分别以按需付费和存储包抵扣介绍陈先生9月的消费情况。

计费方式	费用情况	说明
按需计费	- 性能型：100 GB x 0.0025元/GB/小时 x 24小时 x 30天 = 180.00元 - 容量型：200 GB x 0.000486元/GB/小时 x 24小时 x 30天 = 69.98元 - 总费用：180.00 + 69.98 =249.98元	
存储包抵扣	- 性能型：165元 - 容量型：30 * 2 = 60元 - 总费用：165 + 60 = 225元	购买1个1个月时长的100GB性能型存储包和2个1个月时长的100G容量型存储包

案例2：存储容量波动型

容量型

陈先生在华北-北京可用区A创建了一个容量型的文件系统。假设在2024年9月份期间每小时的最大存储容量均为100 GB，但其中有1天的最大存储容量为200 GB，以下分别以按需付费和存储包抵扣介绍陈先生9月的消费情况。

计费方式	费用情况	说明
按需计费	100 GB x 0.000486元/GB/小时 x 24小时 x 29天 + 200 GB x 0.000486元/GB/小时 x 24小时 x 1天 = 36.16元	
存储包抵扣+按需计费	(200 GB - 100 GB) x 0.000486元/GB/小时 x 24小时 x 1天 + 30元 = 31.17元	购买1个一个月时长的100GB容量型存储包，超出存储包的容量按需付费。

性能型

陈先生在华北-北京可用区A创建了一个性能型的文件系统。假设在2024年9月份期间每小时的最大存储容量均为100 GB，但其中有1天的最大存储容量为200 GB，以下分别以按需付费和存储包抵扣介绍陈先生9月的消费情况。

计费方式	费用情况	说明
按需计费	100 GB x 0.0025元/GB/小时 x 24小时 x 29天 + 200 GB x 0.0025元/GB/小时 x 24小时 x 1天 = 186.00元	
存储包抵扣+按需计费	(200 GB - 100 GB) x 0.0025元/GB/小时 x 24小时 x 1天 + 165元 = 171.00元	购买1个1个月时长的100GB性能型存储包，超出存储包的容量按需付费。

容量型+性能型

陈先生在华北-北京可用区A创建了一个性能型的文件系统和一个容量型的文件系统。假设在2024年9月份期间性能型的文件系统每小时的最大存储容量均为100 GB，但其中有1天的最大存储容量为200 GB；容量型的文件系统每小时的最大存储容量均为200 GB，但其中有1天的最大存储容量为400 GB；以下分别以按需付费和存储包抵扣介绍陈先生9月的消费情况。

计费方式	费用情况	说明
按需计费	- 性能型：100 GB x 0.000486元/GB/小时 x 24小时 x 29天 + 200 GB x 0.000486元/GB/小时 x 24小时 x 1天 = 186.00元 - 容量型：200 GB x 0.000486元/GB/小时 x 24小时 x 29天 + 400 GB x 0.000486元/GB/小时 x 24小时 x 1天 = 72.32元 - 总费用：186.00 + 72.32 =258.32元	
存储包抵扣+按需计费	- 性能型：(200 GB - 100 GB) x 0.0025元/GB/小时 x 24小时 x 1天 + 165元 = 171.00元 - 容量型：(400 GB - 200 GB) x 0.000486元/GB/小时 x 24小时 x 1天 + 30 * 2 = 62.33元 - 总费用：171 + 62.33 = 233.33元	购买1个1个月时长的100GB性能型存储包和2个1个月时长的100G容量型存储包，超出存储包的容量按需付费。

快速入门

入门概述

本文介绍从创建到使用CFS文件系统的全流程，帮助您快速上手文件存储CFS。

CFS概述

百度智能云文件存储（Cloud File System）是一种安全高效的文件存储服务，为云上的虚机、容器资源提供了跨操作系统的文件存储及共享能力。CFS基于POSIX接口适配原生操作系统，兼容NFS协议和SMB协议。同时，百度智能云CFS提供简单、易操作的对外接口，并支持按实际使用量计费，免去部署、维护费用的同时，最大化提升您的业务效率。

- [基本概念](#)：了解文件存储CFS在使用过程中遇到的基本概念。
- [产品特性](#)：了解文件存储CFS的产品特性。
- [产品规格](#)：文件存储CFS提供了性能型和容量型存储类型，可根据实际业务选择。
- [使用限制](#)：了解文件存储CFS使用限制。
- [应用场景](#)：了解文件存储CFS丰富的应用场景
- [如何选择CFS和PFS、如何选用CFS、BOS和CDS](#)：根据实际业务情况选用合适的百度智能云存储服务。

计费方式

文件存储CFS默认按需计费，同时也支持包年包月（即存储包）付费。

计费方式	说明
按需计费	按需计费是先使用，后付费，适用于业务用量经常有变化的场景。
存储包计费	预先购买针对不同规格的文件系统推出的优惠存储包，在费用结算时，优先从存储包抵扣用量，被抵扣的用量不会再算入按需计费账单。先购买，后抵扣，适用于业务用量相对稳定的场景。

入门示例

- [快速入门（Linux）](#)

- 快速入门 (Windows)

常用操作

如果您想…	您可以参考…
创建文件系统	创建文件系统
查看文件系统基本信息、挂载点信息或客户端信息	查看文件系统详情
添加挂载点	添加挂载点
创建权限组和规则	权限组管理
挂载/卸载文件系统	1. Linux系统挂载与卸载 单台BCC挂载与卸载NFS协议CFS 多台BCC批量挂载与卸载NFS协议CFS 单台BCC挂载与卸载SMB协议CFS 2. Windows系统挂载与卸载SMB协议CFS 3. CCE通过CSI挂载与卸载CFS
删除文件系统	删除文件系统
监测文件系统读写数据、元数据、用量等	查看监控
为监控指标设置报警策略	报警管理 指标含义
备份文件系统数据	备份
对文件系统中的数据加密访问	数据加密
为文件系统设置标签	管理标签
为不同子用户设置权限	多用户访问控制

快速入门 (Linux)

本文介绍如何在CFS控制台创建NFS协议文件系统，并将其挂载至云服务器BCC实例（Linux）上，从而实现数据读写。

前提条件

- 已开通文件存储CFS服务。首次登录[文件存储CFS控制台](#)时，需要根据页面引导开通CFS服务。
- 已创建目标BCC实例，用于挂载CFS文件系统。BCC创建方式请参见：[快速配置Linux云服务器](#)。

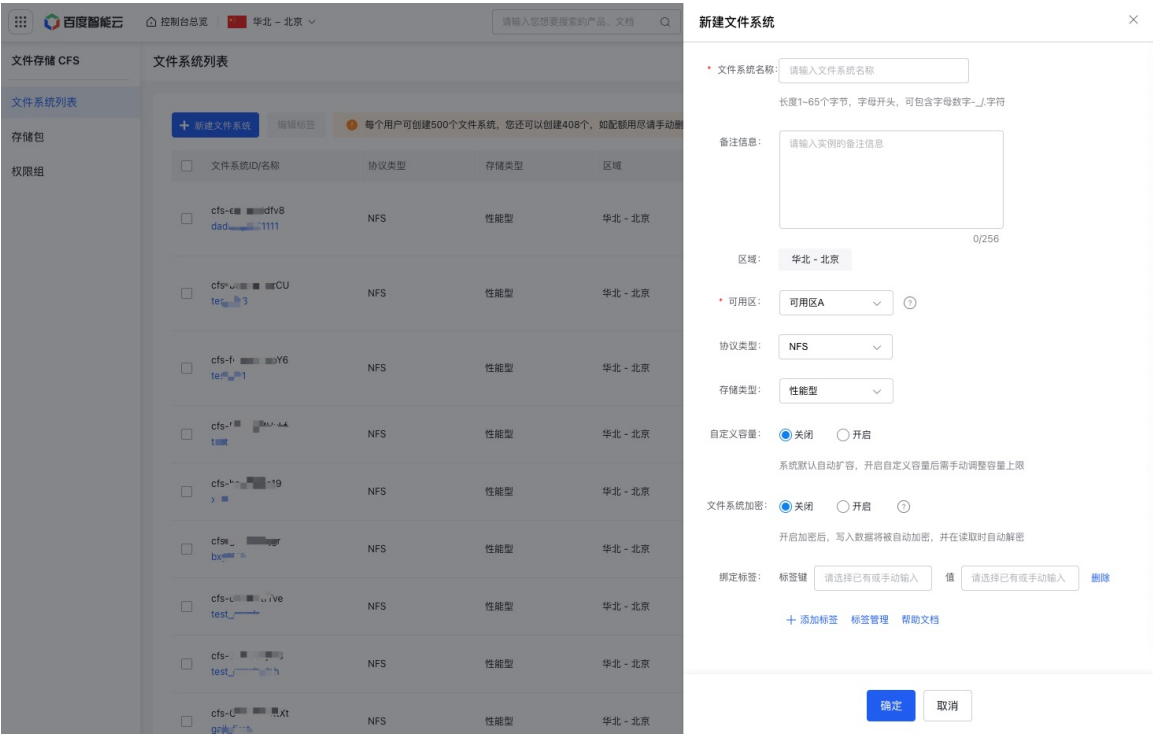
挂载操作步骤

步骤一：创建CFS文件系统

- 在CFS控制台，单击“新建文件系统”。
- 在新建文件系统面板，填写文件系统名称、可用区、协议类型、存储类型等信息。详细参数说明，请参考 [创建文件系统](#)。

说明：

- ①为获得最佳性能体验，建议CFS文件系统与BCC实例同一可用区
- ②本文选择NFS协议，代表装有Linux操作系统的BCC实例通过NFS协议挂载并访问CFS文件系统



3. 单击“确定”，根据页面提示，完成CFS文件系统创建。

步骤二：创建挂载点

- 1. 在CFS控制台，点击“文件系统名称”进入基本信息页
- 2. 点击“添加挂载点”，在弹出的页面中选择子网及权限组。更多权限组信息请参见：[权限组管理](#)



3. 点击“确认”后，即可查看挂载点信息。请记录此处的挂载地址（cfs-***.baidubce.com），用于后续的挂载。



步骤三：挂载CFS文件系统

1. 在BCC控制台登录目标BCC实例，本文以VNC方式登录，更多登录方式请参考 [云服务BCC登录方式](#)。
 2. 在目标BCC实例中安装NFS客户端，，下述命令均需要root权限。
- CentOS/RHEL 操作系统请运行以下命令：

```
yum install nfs-utils
```

- Ubuntu/Debian 操作系统请运行以下命令：

```
apt-get install nfs-common
```

3. 在目标BCC实例中执行如下挂载命令

```
mount -t nfs4 -o minorversion=1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport <挂载地址>:/<客户端本地路径>
```

说明：

- <挂载地址> 可以在文件系统的挂载点列表中查到(见步骤二)；
- CFS文件系统的路径：默认配置为CFS的根目录（例如：/），您可以修改成已存在的子目录（例如：/dir0）；
- <客户端本地路径> 是待挂载客户端的本地路径。必须是以/开头的绝对路径（例如：/mnt/cfs），且该路径必须已被创建。

示例：

```
mount -t nfs4 -o minorversion=1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport cfs-***.baidubce.com:/ /mnt/cfs
```

步骤四：验证挂载

1. 通过以下命令查询是否挂载成功

```
df -h
```

如果返回如下信息，则代表挂载成功

```
lroot@ ~# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.9G  0  3.9G   0% /dev
tmpfs           3.9G  0  3.9G   0% /dev/shm
tmpfs           3.9G  1.2M  3.9G   1% /run
tmpfs           3.9G  0  3.9G   0% /sys/fs/cgroup
/dev/ndal       20G   3.3G   16G  18% /
tmpfs           796M  0  796M   0% /run/user/0
cfs-yh1q1m1j0  18P   0  18P   0% /mnt
```

2. 挂载成功后，您可以在BCC上把CFS文件系统当作一个普通的本地目录来访问和使用。

🔗 卸载操作步骤

- 1. 登录到挂载CFS文件系统的[云服务器BCC](#)。
- 2. 使用umount命令卸载文件系统。

```
umount <目标路径>

##### 其中，<目标路径> 为需要卸载的CFS文件系统本地路径
```

3. 执行mount -l命令，查看卸载结果。如果回显中未找到您挂载的CFS文件系统信息，则表示该文件系统的挂载目录已经成功卸载。

快速入门（Windows）

本文介绍如何在CFS控制台创建SMB协议文件系统，并将其挂载至云服务器BCC实例（Windows）上，从而实现数据读写。

🔗 前提条件

- 1. 已开通文件存储CFS服务。首次登录[文件存储CFS控制台](#)时，需要根据页面引导开通CFS服务。
- 2. 已创建目标BCC实例，用于挂载CFS文件系统。BCC创建方式请参见：[快速配置Windows云服务器](#)。

🔗 挂载操作步骤

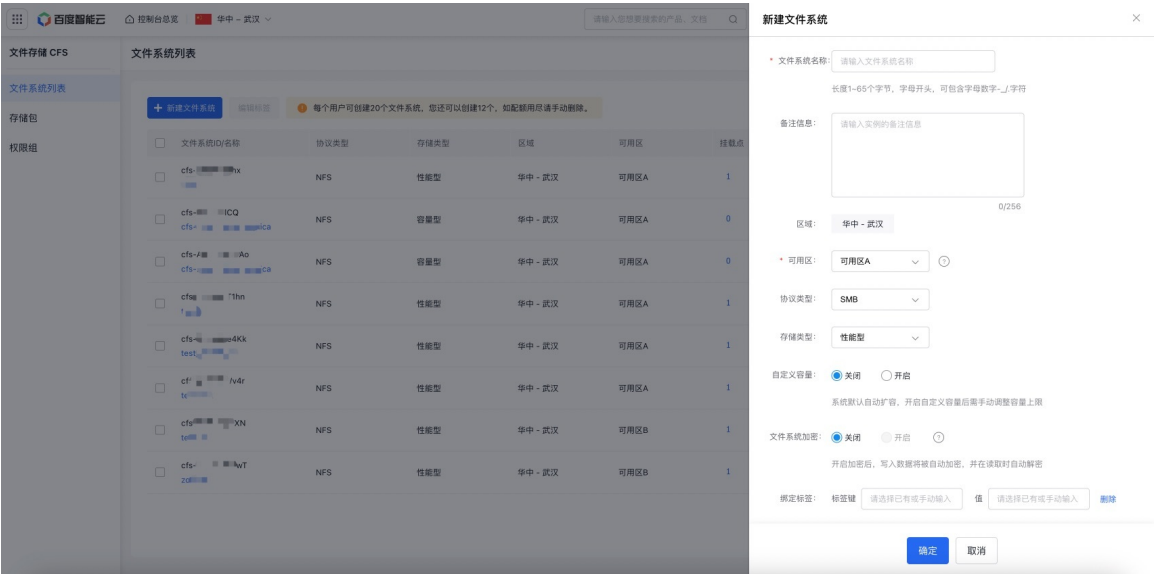
🔗 步骤一：创建CFS文件系统

- 1. 在CFS控制台，单击“新建文件系统”。
- 2. 在新建文件系统面板，填写文件系统名称、可用区、协议类型、存储类型等信息。详细参数说明，请参考 [创建文件系统](#)。

说明：

①为获得最佳性能体验，建议CFS文件系统与BCC实例同一可用区

②本文选择SMB协议，代表装有Windows操作系统的BCC实例通过SMB协议挂载并访问CFS文件系统



3. 单击“确定”，根据页面提示，完成CFS文件系统创建。

步骤二：创建挂载点

- 1. 在CFS控制台，点击“文件系统名称”进入基本信息页
- 2. 点击“添加挂载点”，在弹出的页面中选择子网及权限组。更多权限组信息请参见：[权限组管理](#)



3. 点击“确认”后，即可查看挂载点信息。请记录此处的挂载地址（cfs-***.baidubce.com），用于后续的挂载。



步骤三：挂载CFS文件系统

- 1. 在BCC控制台登录目标BCC实例，本文以VNC方式登录，更多登录方式请参考 [云服务BCC登录方式](#)。

2. 登录后，在目标BCC实例中打开Powershell或cmd命令行，执行如下挂载命令

```
net use <客户端本地盘符>: \\<挂载地址>\cfs
```

说明：

- <客户端本地盘符>是待挂载CFS文件系统的本地盘符（例如z），不能与已有盘符重复；
- <挂载地址> 可以在文件系统的挂载点列表中查到(见步骤二)；
- "cfs"为固定SMB share名称，不可修改。

示例：

```
net use z: \\cfs-***.lb-b768ef66.cfs.bj.baidubce.com\cfs
```

🔗 步骤四：验证挂载

1. 挂载完成后，可在Windows cmd命令行或Powershell中使用以下命令查看已挂载的文件系统。

```
net use
```

或直接打开Windows资源管理器查看网络位置中的已有盘符。



2. 挂载成功后，您可以在BCC上把CFS文件系统当作一个普通的本地目录来访问和使用。

🔗 卸载操作步骤

1. 登录到挂载CFS文件系统的云服务器BCC。
2. 使用以下命令卸载文件系统。

```
net use <目标盘符>: /del
```

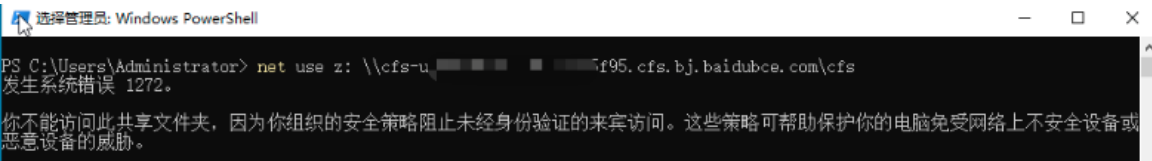
其中，<目标盘符>为需要卸载的CFS文件系统本地盘符。

3. 执行net use命令，查看卸载结果。如果回显中未找到您挂载的CFS文件系统信息，则表示该文件系统目录已成功卸载。

🔗 常见问题

🔗 系统错误1272该如何处理？

如果Windows Server出现如下报错信息（发生系统错误1272），是因为Windows系统因安全策略阻挡了以来宾访问权限（Guest Auth）访问SMB文件系统的用户



解决方案：以管理员身份运行如下powershell命令

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v "AllowInsecureGuestAuth" /t REG_DWORD /d 1 /f
net stop lanmanworkstation /y ; net start lanmanworkstation ; net start SessionEnv
```

或以管理员身份运行如下cmd命令：

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v "AllowInsecureGuestAuth" /t REG_DWORD /d 1 /f
net stop lanmanworkstation /y && net start lanmanworkstation && net start SessionEnv
```

操作指南

开通服务

文件存储 CFS (Cloud File Storage) 是百度智能云提供的安全、可扩展的文件存储服务。通过标准的文件访问协议，为云上的虚拟机、容器等计算资源提供无限扩展、高可靠、全球共享的文件存储能力。可用于内容管理及 Web 应用、大数据分析、泛娱乐和媒体应用等业务场景。

控制台为用户提供了界面化管理CFS服务的能力，在使用前请先[开通CFS服务](#)。

- 1. 注册并登录百度智能云平台，请参考[注册](#)和[登录](#)。
- 2. 如果未进行实名认证，点击控制台右上角的用户名的“账号管理”进入账号管理页面并点击“实名认证”。请根据实际情况选择“企业认证”或者“个人认证”。



- 3. 右上角选择“产品导览 > 文件存储 CFS”，进入“开通文件存储CFS”页面，勾选“我已阅读并同意《CFS服务等级协议SLA》”，点击“立即开通”。

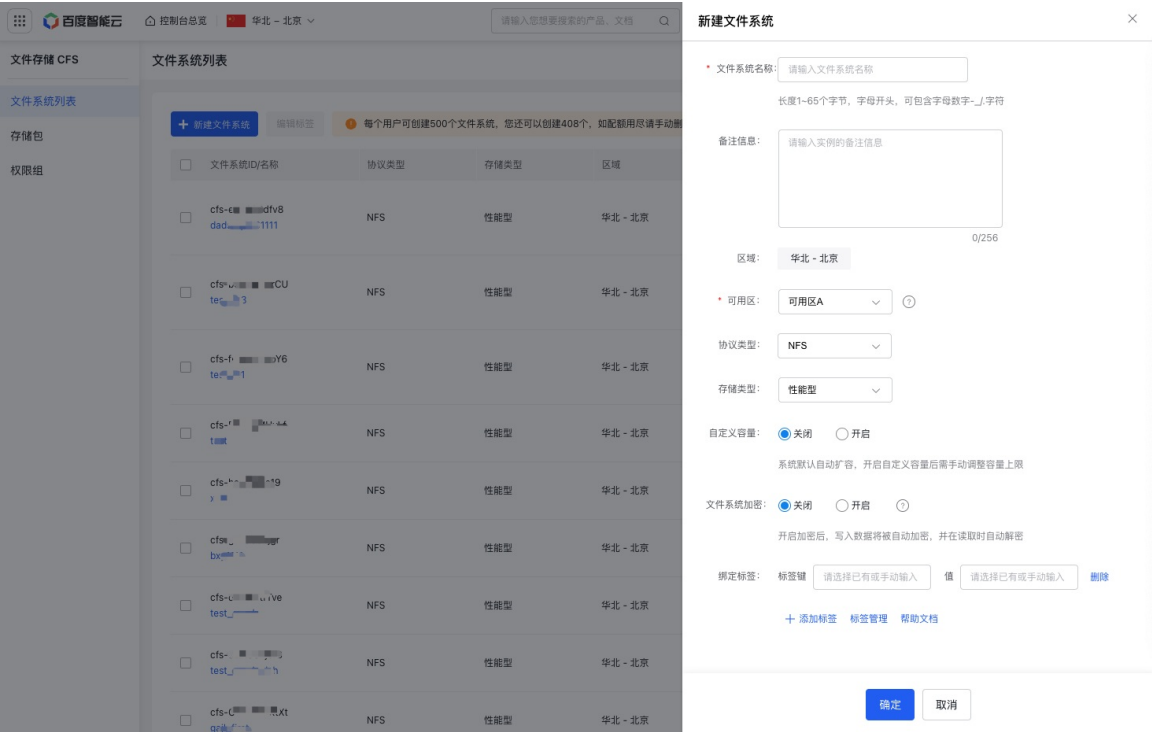


管理文件系统

创建文件系统

使用百度智能云文件系统 CFS 服务，需要先创建一个文件系统。

进入百度智能云管理控制台的“文件系统CFS->文件系统列表”页面，在页面左上角点击“新建文件系统”，在弹窗中进行创建文件系统的各项参数配置，完成文件系统的创建。具体购买界面及选择参数如下：



参数	说明
文件系统名称	用户根据需求设置需要创建文件系统的名称。
备注信息	用户根据需求设置实例备注信息，限制256字符。
区域	即该文件系统创建所在的地域，如需更改可以点击控制台左上角的区域名称来进行区域切换。
可用区	用户根据需求设置需要创建文件系统的可用区。每一个地域中都包含若干可用区，这里将会显示区域内所有已经部署CFS服务的可用区。文件系统可以跨可用区挂载使用，但是建议选择与虚拟机/容器在同一可用区，此时可避免跨可用区产生的时延长，性能最佳。
协议类型	用户根据需求设置需要创建文件系统的支持协议类型目前CFS支持NFS和SMB两种协议类型。
存储类型	文件存储CFS分为容量型和性能型两种，您可按需选择。
自定义容量	用户根据需要开启是否自定义容量，系统默认自动扩容，开启自定义容量后需手动调整容量上限。
容量上限	若用户开启“自定义容量”，则可根据需要设置该文件系统的容量上限（单个文件系统的最大容量），当实际使用容量达到该值时，将无法对文件系统进行写入操作，需要进行容量变更。
文件系统加密	开启加密后，写入数据将被自动加密，并在读取时自动解密。
绑定标签	通过给每个云资源绑定标签，可以实现快速分类和识别管理这些资源。

用户设置好文件系统的各项参数配置后，单击“确认”，则文件系统创建成功。

创建文件系统后，您需要继续创建挂载点才能使用文件系统。常见文件系统实例状态情况如下：

实例状态	状态说明
运行中	文件系统实例正常运行中
创建中	文件系统实例正在创建中
删除中	文件系统实例正在删除中
备份恢复失败	文件系统实例从源文件系统的备份副本中恢复失败，建议删除后重新进行备份恢复操作
备份恢复中	文件系统实例正在从源文件系统的备份副本中恢复，如需恢复的数据量较大，恢复时间会较长
停止运行	当账号欠费后，如果文件系统有以下情况之一将停止运行。 1. 文件系统未绑定存储包 2. 文件系统绑定存储包但存储包已用尽 3. 文件系统绑定存储包但存储包未生效或已过期 更多说明参见 余额不足提醒和欠费处理 。

删除文件系统

注意：需先删除文件系统已有的挂载点，才可对文件系统进行删除。

1. 删除挂载点：点击文件系统列表操作栏的“管理”按钮进入详情页，对文件系统现有挂载点进行删除操作。



2. 在文件系统列表页，点击操作栏的“删除”按钮，即可对对应文件系统进行删除。



查看文件系统详情

您可以通过文件存储 CFS 控制台查看当前已有文件系统的相关信息，并对已有文件系统进行管理操作，如查看文件系统基本信息、挂载点信息及挂载客户端信息等。

文件系统列表

在文件系统列表中，可以查看现有文件系统的 ID/名称、协议类型、存储类型、区域、可用区、挂载点、已用容量/容量上限、加密状态、创建时间、状态、标签等重要信息。同时可对选中的文件系统进行添加挂载点、管理、删除等操作。可通过列表右上角的搜索框，按照文件系统的名称、ID和标签快速搜索相关文件系统。



🔗 文件系统详情

在文件系统列表页面，点击文件系统的名称，则可进入该文件系统的详情页面。在文件系统的详情页面可以看到文件系统的基本信息、挂载点信息及已挂载客户端信息。

文件系统基本信息

文件系统基本信息中展示了实例ID、名称、状态、描述信息、协议类型、存储类型、地域、已用容量、容量上限、加密状态及创建时间等基本信息，您可以对实例名称、描述信息和容量上限进行修改。

基本信息					
实例ID：	cfs-enZ69Rdfv8	实例名称：	dadsaq2261111	状态：	运行中
描述信息：	111赛特斯吞吞吐吐吞吞吐吐推塔推塔推塔...	协议类型：	NFS	存储类型：	性能型
地域：	华北 - 北京 可用区D	已用容量：	64.00 GB	容量上限：	80.00 GB 容量变更
加密状态：	未加密	创建时间：	2023-03-30 14:15:32		

文件系统挂载点信息

文件系统挂载点信息中展示了该文件系统当前挂载点的ID、所在VPC ID、所在子网、挂载地址、权限组等信息，您可以对特定的挂载点进行修改权限组或删除操作。

挂载点信息					
+ 添加挂载点					
挂载点ID	所在VPCID	所在子网	挂载地址	权限组	操作
38776c6b-4e667054-50513d3d	vpc-120c0	sub-1bvw	cfs-en-d37a9-bce.com	默认权限组	修改权限组 删除
4b547c-72614c-5513d3d	vpc-1p6ej	sub-1py	cfs-en-8lb-de8-bce.com	long	修改权限组 删除
4e374e-507971-73d3d	vpc-120c0	sub-1iz0m	cfs-en-8lb-e53c-bce.com	默认权限组	修改权限组 删除

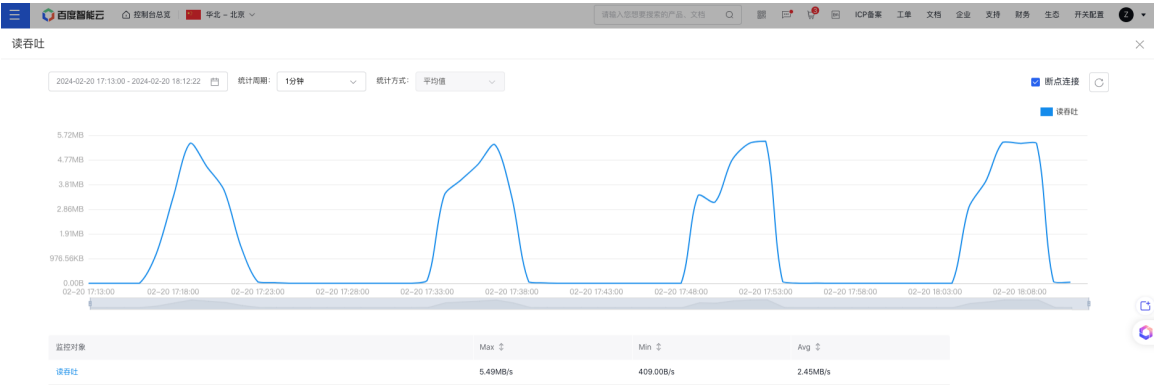
文件系统挂载客户端信息

文件系统挂载客户端信息中展示了该文件系统当前挂载客户端的IP、可用区、VPC ID、挂载点ID和读/写吞吐等信息。

挂载客户端信息					
请输入客户端IP进行搜索					
客户端IP	可用区	VPC ID	挂载点ID	读吞吐	写吞吐
192.168.1.116	zone8	vp-1p6ej	35714-6307c-5513d	2423 8.79KB/s	0.008/s

说明：文件系统挂载完客户端后，该客户端可用区信息需要10s后才能查询，请稍后刷新查看。

您可以点击 [📈](#) 按钮对文件系统挂载的所有客户端按读/写吞吐瞬时流量进行排序，也可以点击 [📊](#) 按钮查看具体挂载客户端的历史读/写吞吐情况。



设置文件系统容量上限

您可以根据需要为CFS文件系统设置自定义容量。系统默认关闭自定义容量，若您开启“自定义容量”，则可根据需要设置该文件系统的容量上限（单个文件系统的最大容量），当实际使用容量达到该值时，将无法对文件系统进行写入操作，需要进行容量变更。

设置容量上限

新建文件系统时，支持开启自定义容量，设置文件系统的容量上限（最大支持10PB）。

新建文件系统

* 文件系统名称: 请输入文件系统名称
长度1~65个字节，字母开头，可包含字母数字“-”/“_”字符

备注信息: 请输入实际的备注信息
0/256

区域: 华北 - 北京

* 可用区: 可用区A

协议类型: NFS

存储类型: 性能型

自定义容量: ☒ 关闭 ☐ 开启
系统默认自动扩容，开启自定义容量后需手动调整容量上限

文件系统加密: ☒ 关闭 ☐ 开启 ☐ 继承
开启加密后，写入数据将被自动加密，并在读取时自动解密

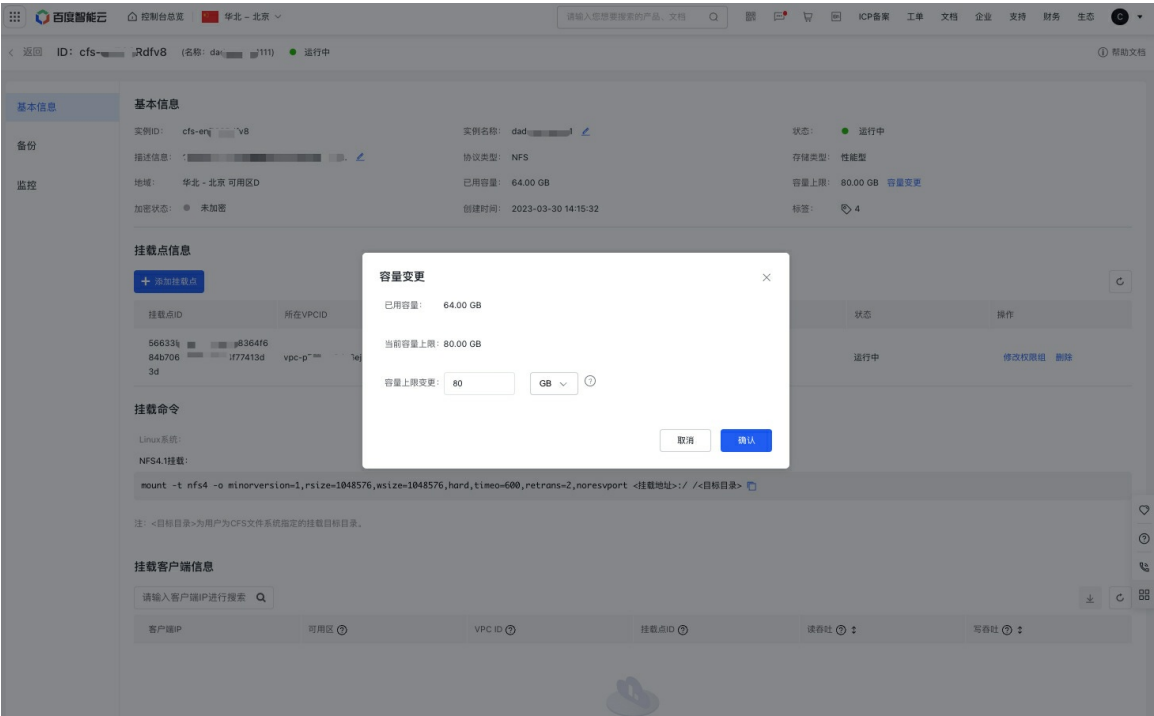
绑定标签: 标签键 请选择已有或手动输入 值 请选择已有或手动输入 删除
+ 添加标签 标签管理 帮助文档

确定 取消

调整容量上限

您的实际使用容量达到文件系统设置的容量上限后，将无法对文件系统进行写入操作，需要进行容量变更，变更操作如下：

1. 在文件系统列表找到对应的文件系统，点击管理进入基本信息页；
2. 点击容量变更，调整容量上限。



挂载访问

添加挂载点

文件系统创建完成后会在文件系统列表中显示。云上的虚拟机、容器等计算资源需要通过挂载点挂载至文件系统，所以需要为文件系统添加挂载点。

添加挂载点

点击文件系统列表中操作栏的“添加挂载点”按钮，或直接点击文件系统分列表中“挂载点”对应的数量统计，即可打开添加挂载点页面。



您需要为挂载点选择所对应的VPC及子网，百度智能云为您在每一个地域提供了一个默认VPC，您也可以按需购买VPC服务并创建新的VPC。

选择挂载点对应的VPC后，还需选择子网及权限组等信息。



注意：

- 同一VPC中的多个子网，可以分别添加不同的挂载点。
- VPC下的一个子网仅能添加一个挂载点，且子网的设备类型仅支持通用型。
- VPC下的所有子网都可以创建挂载点，您可以选择虚拟机/容器所在的可用区，以获得最佳的访问性能。

🔍 查询挂载点

创建挂载点后，可以进入到文件系统基本信息页查询挂载点信息，包括挂载点ID、所在VPCID、所在子网、挂载地址、权限组、状态等信息。

挂载点信息						
+ 添加挂载点						
挂载点ID	所在VPCID	所在子网	挂载地址	权限组	状态	操作
583676-4b6d4143-9e99773d3d	vpc-j6uwwu20c0	snb-pr-npshz0m	cfs-fHVe-26372.cfs.bj.ba.com	默认权限组	运行中	修改权限组 删除
74674431-547303-13a3d	vpc-j6uwwu20c0	snb-cl-bajie	cfs-fHVecD-934bfd.cfs.bj.ba.com	默认权限组	运行中	修改权限组 删除

挂载点常见状态说明：

状态	说明
运行中	挂载点正常运行
创建中	挂载点正在创建中
删除中	挂载点正在删除中

BCC挂载与卸载CFS

🔍 Linux系统挂载与卸载

🔍 单台BCC挂载与卸载NFS协议CFS

获取挂载地址

创建挂载点后就可以使用文件系统了。您可以点击文件系统列表中操作栏的“管理”按钮，即可打开挂载点信息页面，如下图所示：

挂载点信息

+ 添加挂载点

C

挂载点ID	所在VPCID	所在子网	挂载地址	权限组	操作
36634f8a-727157-704dc369-73d3d	vpc-jf-c0	sbn-hz0m	cfs-EU-664f-ib-3.com	默认权限组	修改权限组 删除

登录到虚拟机或者容器后，使用挂载地址来操作文件系统。

安装NFS客户端

多数操作系统默认不会安装NFS客户端，所以挂载文件系统前需要手动安装NFS客户端。

登录云服务器运行命令，安装NFS客户端。

- CentOS/RHEL 操作系统请运行以下命令：

```
yum install nfs-utils
```

- Ubuntu/Debian 操作系统请运行以下命令：

```
apt-get install nfs-common
```

上述命令均需要root权限。

手动挂载NFS文件系统

推荐使用以下命令挂载文件系统（需要 root 权限）：

```
mount -t nfs4 -o minorversion=1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport <挂载地址>: /<目标目录>
```

其中，<挂载地址> 可以在文件系统的挂载点列表中查到，<目标目录> 是挂载的目标目录。

命令中的参数解释参见下表（更多参数请参考 [NFS文档](#)）：

选项	说明
minorversion	挂载协议的小版本号。必须为 1，指定使用 NFS 4.1 协议。
rsize	设置客户端文件读请求可以接收的数据最大字节数。该值设置过小，会导致读吞吐下降。 推荐值：1048576。
wsiz	设置客户端文件写请求可以发送的最大字节数。该值设置过小，会导致写吞吐下降。 推荐值：1048576。
hard	客户端重试行为选项，有 hard 和 soft 两个可选值，不指定默认使用 hard。 hard 选项下，客户端在服务端回复前会无限重试，该选项能最大限度保证数据完整性。soft 选项下，客户端重试一段时间后放弃，同时返回错误给客户端，如果启用该选项，建议调大 timeo 值，来降低数据损坏风险。 推荐值：hard。
timeo	客户端在重试 NFS 请求之前等待响应的时间，单位是分秒（0.1s）。该值不应该低于 150，即 15 秒。 推荐值：600。
retrans	客户端重试请求次数，超过此次数之后将尝试进一步的恢复操作。 推荐值：2。
noresvport	是否使用特权端口号，有 resvport 和 noresvport 两个可选值，默认使用 resvport。 resvport 代表使用特权端口号（1-1023），noresvport 代表不使用特权端口号（>1023）。非特权端口号的范围更大，可以减少一些因为端口分配问题导致的服务不可用问题。 推荐值：noresvport。

自动挂载NFS文件系统

手动挂载的文件系统会在重启后丢失挂载信息，如果您需要在系统启动的时候自动挂载文件系统，可以按照下面的步骤操作。

修改 `/etc/fstab`，添加如下内容（需要 root 权限）：

```
<挂载点地址>:/ <目标目录> nfs4
minorversion=1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,_netdev 0 0
```

与手动挂载相比，自动挂载会多一个参数 `_netdev`。该参数要求操作系统在网络就绪后再尝试挂载文件系统。如果不指定该参数，会增加挂载失败的风险。

卸载NFS文件系统

使用 `umount` 命令卸载文件系统：

```
umount <目标目录>
```

查看已挂载的NFS文件系统

使用以下命令可以查看已经挂载的文件系统：

```
cat /proc/mounts
```

如果需要查看文件系统的容量信息，可以使用以下命令：

```
df -h
```

🔗 多台BCC批量挂载与卸载NFS协议CFS

批量挂载

如果您的业务遇到需要多台BCC实例共享同一CFS文件系统中的数据时，您可以通过云助手执行批量挂载命令，一键操作即可实现多台BCC实例批量挂载同一CFS文件系统。

前提条件

您在进行批量挂载前，需要满足以下条件：

- 已[创建文件系统](#)并[添加挂载点](#)。
- BCC实例和CFS挂载点在同一VPC中
- 已在CFS文件系统挂载点的权限组中为BCC实例授权，具体操作参考[权限组管理](#)。

使用限制

您在进行批量挂载前，需要清楚该操作存在的限制条件：

- 仅支持BCC实例中Linux操作系统挂载NFS协议的CFS文件系统。
- 仅支持状态为“运行中”的BCC实例。如果BCC实例状态为“启动中”、“停机”等状态，请您先修复实例状态。
- 云助手为正常状态。
- BCC实例必须和CFS文件系统在上一地域内，云助手不支持对不同地域的BCC实例执行挂载操作。

操作步骤

1. 登录[云服务器BCC控制台](#)，在左侧导航栏，选择【运维与监控-云助手】。
2. 进入到云助手控制台，在左侧导航栏选择【公共命令】后，选择【批量挂载CFS】命令，点击【执行】操作。
3. 进入到执行命令面板，配置相应命令参数，包括MountTargetDomain、LocalPath、CFSRemotePath、AutomountOnBoot。

命令参数：

参数名称	参数值	参数描述
MountTargetDomain		请输入文件系统挂载点地址。您可以在CFS控制台目标文件系统的挂载点列表查看挂载点地址，例如cfs****.baidubce.com
LocalPath		当前服务器上待挂载CFS文件系统的本地路径。必须是以/开头的绝对路径（例如：/mnt/cfs），如果该路径不存在会自动创建
CFSRemotePath		CFS文件系统的目录。您可以配置为CFS的根目录（例如：/）或已存在的子目录（例如：/dir0）
AutoMountOnBoot		是否开机自动挂载文件系统。取值：true或false。开启自动挂载后，当您重启BCC实例时，无需重复挂载操作

命令参数	参数描述
MountTargetDomain	输入文件系统挂载点地址，可以在CFS控制台目标文件系统的挂载点列表查看挂载点地址，例如cfs****.baidubce.com。具体操作，请参见 查看文件系统详情
LocalPath	当前服务器上待挂载CFS文件系统的本地路径。必须是以/开头的绝对路径（例如：/mnt/cfs），如果该路径不存在会自动创建。
CFSRemotePath	CFS文件系统的目录。您可以配置为CFS的根目录（例如：/）或已存在的子目录（例如：/dir0）。
AutomountOnBoot	是否开机自动挂载文件系统。取值：true或false。开启自动挂载后，当您重启BCC实例时，无需重复挂载操作。

4.根据业务需要通过手动选择实例、选择全部实例、指定实例标签和导入实例清单等方式，选择需要进行批量挂载的多台BCC实例。

选择实例

获取方式：

手动选择实例

在实例列表通过搜索...

选择全部实例

无需逐一勾选，即可...

指定实例标签

指定一个或多个标签...

导入实例清单

导入实例ID清单或导...

实例列表：

BCC

BBC

华北 - 保定

请选择

实例名称

请输入

Q

☐	实例名称/ID	状态	BSM_Agent状态	OS	创建时间	支付方式	IP地址/带宽	配置/类型	标签
☐	in- i- T	运行中	在线	CentOS 7.6	2022-09-21	后付费	17 - / -Mbps(公)	16核/64GB/40GB g4	默认项目：
☐	ins i- c	运行中	在线	CentOS 7.6	2022-09-21	后付费	17 - / -Mbps(公)	16核/64GB/40GB g4	默认项目：
☐	insta i-W	运行中	在线	CentOS 7.6	2022-09-21	后付费	17 - / -Mbps(公)	16核/64GB/40GB g4	默认项目：
☐	ins i-l	-	离线	CentOS 7.6	2022-09-21	后付费	17 - / -Mbps(公)	16核/64GB/40GB g4	默认项目：

5.点击执行后，您可在历史记录中查看执行情况。执行成功后各BCC实例会返回success状态，如失败可参考报错原因进行问题排查后重试。

云助手

实例列表

我的命令

公共命令

历史记录

历史记录

华北 - 北京

命令执行记录

文件上传记录

历史记录最长保存30天。

开始时间

结束时间

命令名称

请输入命令名称进行搜索

Q

C

执行ID	执行状态	命令名称/ID	命令内容	创建时间	执行情况	操作
r-g8 2hN2	执行中	批量挂载CFS c-Bx	#!/bin/bash function exit_with_msg() { err...	2024-01-26 15:47:55	总计实例数：2 执行失败实例数：0	详情

6.您可登录到具体的BCC实例验证挂载，通过以下命令查询，如返回如下信息，则代表挂载成功。

df -h

```
[root@instance-eeh5dznq ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                   3.9G         0   3.9G   0% /dev
tmpfs                      3.9G         0   3.9G   0% /dev/shm
tmpfs                      3.9G   1.2M   3.9G   1% /run
tmpfs                      3.9G         0   3.9G   0% /sys/fs/cgroup
/dev/vda1                  20G   4.0G   15G   21% /
cfs-zSUKPdTx6             10P         0   10P   0% /mnt
tmpfs                      795M         0   795M   0% /run/user/0
[root@instance-eeh5dznq ~]#
```

批量卸载

如果您多台BCC实例不再需要共享同一CFS文件系统的数据，也可以通过云助手执行批量卸载命令。

1. 在云助手控制台的公共命令页面，选择【批量卸载CFS】命令，点击【执行】操作。
2. 进入到执行命令面板，配置LocalPath命令参数。

命令参数：

参数名称	参数值	参数描述
LocalPath		当前服务器上待卸载CFS文件系统的本地路径。必须是以/开头的绝对路径（例如：/mnt/cfs）

3. 选择需要进行批量卸载的BCC实例，点击执行完成批量卸载。

单台BCC挂载与卸载SMB协议CFS

创建CFS SMB协议文件系统后，您需要使用云服务器BCC来挂载该文件系统，以实现多个BCC共享访问文件系统的目的。本文介绍如何通过Linux BCC实例挂载CFS SMB协议文件系统。

前提条件

1. 已创建目标BCC实例（Linux），用于挂载CFS文件系统。BCC创建方式请参见：[快速配置Linux云服务器](#)。

2. 已创建SMB协议文件系统并添加挂载点。具体操作请参见：[创建文件系统](#)、[添加挂载点](#)。

挂载操作步骤

步骤一：获取挂载地址

创建挂载点后就可以使用文件系统了。您可以点击文件系统列表中操作栏的“管理”按钮，即可打开挂载点信息页面，如下图所示：

挂载点信息

+ 添加挂载点

⌂

挂载点ID	所在VPCID	所在子网	挂载地址	权限组	操作
36634f704d...72715773d3d	vpc-j...c0	sbn-...hz0m	cfs-EU...lib-664f...b.com	默认权限组	修改权限组 删除

登录到虚拟机或者容器后，使用挂载地址来操作文件系统。

步骤二：安装SMB客户端

多数Linux操作系统默认不会安装SMB客户端，所以挂载文件系统前需要手动安装SMB客户端。登录云服务器运行命令，安装SMB客户端。

- CentOS/RHEL 操作系统请运行以下命令：

```
yum install cifs-utils
```

- Ubuntu/Debian 操作系统请运行以下命令：

```
apt-get install cifs-utils
```

步骤三：挂载SMB文件系统

执行以下命令，挂载SMB文件系统：

```
mount -t cifs -o vers=3.0,guest,mfsymlinks,rsize=1048576,wsiz=1048576,cache=strict //<挂载地址>/cfs <客户端本地路径>
```

- 说明：
- <挂载地址> 可以在文件系统的挂载点列表中查到(见步骤一)。
 - <客户端本地路径> 是待挂载客户端的本地路径，必须是以/开头的绝对路径（例如：/mnt/cfs），且该路径必须已被创建。

命令中的参数解释参见下表：

选项	说明
vers	SMB协议版本，支持1.0，2.0，2.1，3.0等SMB主要版本，由于Linux内核对部分高版本SMB协议的支持有限，推荐选择3.0版本（不指定vers时的默认值）。
rsize	设置客户端文件读请求可以接收的数据最大字节数。该值设置过小，会导致读吞吐下降。推荐值：1048576。
wsiz	设置客户端文件写请求可以发送的最大字节数。该值设置过小，会导致写吞吐下降。推荐值：1048576。
guest	SMB目前支持guest挂载来进行身份认证和文件访问。
mfsymli nks	用于支持符号链接功能。
cache	设置是否使用smb客户端缓存，可选值: strict和none。未设置时，默认为strict，表示使用客户端缓存。

示例：

```
mount -t cifs -o vers=3.0,guest,mfsymlinks,rsize=1048576,wsiz=1048576,cache=strict  
//cfs-***.lb-b768ef66.cfs.bj.baidubce.com/cfs /mnt/smb_dir
```

步骤四：查看已挂载的SMB文件系统

挂载完成后，使用以下命令可以查看已经挂载的文件系统：

```
cat /proc/mounts
```

若需要查看文件系统的容量信息，可使用以下命令：

```
df -h
```

卸载操作步骤

- 1. 登录到挂载CFS文件系统的云服务器BCC。
- 2. 使用umount命令卸载文件系统。

```
umount <目标路径>
```

其中，<目标路径> 为需要卸载的CFS文件系统本地路径

- 3. 执行mount -l命令，查看卸载结果。如果回显中未找到您挂载的CFS文件系统信息，则表示该文件系统的挂载目录已经成功卸载。

Windows系统挂载与卸载SMB协议CFS

创建CFS SMB协议文件系统后，您需要使用云服务器BCC来挂载该文件系统，以实现多个BCC共享访问文件系统的目的。本文介绍如何通过Windows BCC实例挂载CFS SMB协议文件系统。

前提条件

- 1. 已创建目标BCC实例（Windows），用于挂载CFS文件系统。BCC创建方式请参见：[快速配置Windows云服务器](#)。
- 2. 已创建SMB协议文件系统并添加挂载点。具体操作请参见：[创建文件系统](#)、[添加挂载点](#)。

挂载操作步骤

步骤一：获取挂载地址

创建挂载点之后，您就可以使用文件系统了。您可以点击文件系统列表中操作栏的“管理”按钮，即可打开挂载点信息页面，如

下图所示：

挂载点信息					
+ 添加挂载点					
挂载点ID	所在VPCID	所在子网	挂载地址	权限组	操作
36634f704d...72715773d3d	vpc-j...	sbn-...hz0m	cfs-EU...lb-664f...	默认权限组	修改权限组 删除

登录到虚拟机或者容器之后，使用挂载地址来操作文件系统。

步骤二：挂载SMB文件系统

在windows系统下，打开Powdershell或cmd命令行，使用以下命令挂载SMB文件系统：

```
net use <客户端本地盘符>: \\<挂载地址>\cfs
```

说明：

- <客户端本地盘符>是待挂载CFS文件系统的本地盘符（例如z），不能与已有盘符重复；
- <挂载地址> 可以在文件系统的挂载点列表中查到(见步骤一)；
- "cfs"为固定SMB share名称，不可修改。

示例：

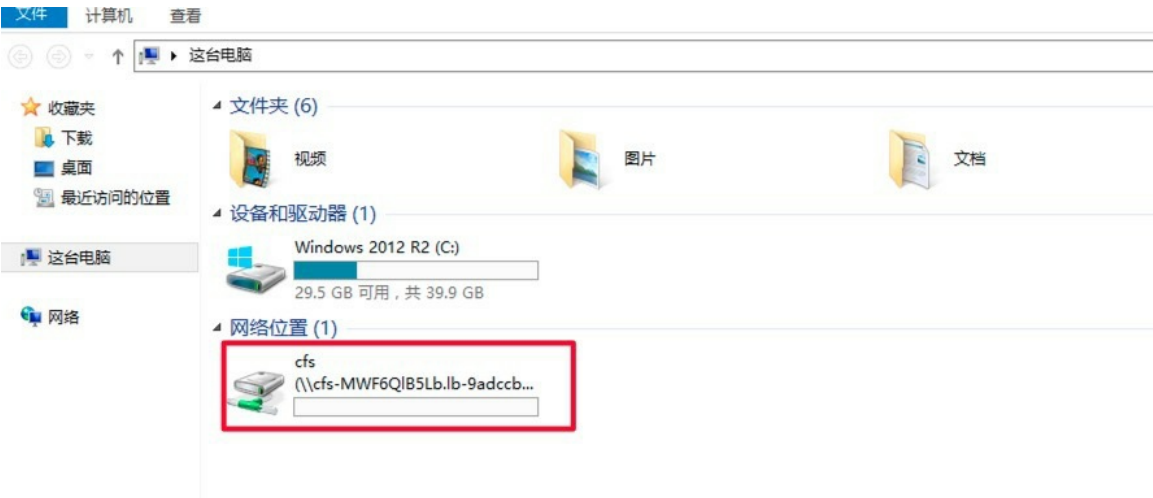
```
net use z: \\cfs-****.lb-b768ef66.cfs.bj.baidubce.com\cfs
```

步骤三：查看已挂载的SMB文件系统

挂载完成后，可在Windows cmd命令行或Powershell中使用以下命令查看已挂载的文件系统。

```
net use
```

或直接打开Windows资源管理器查看网络位置中的已有盘符。



卸载操作步骤

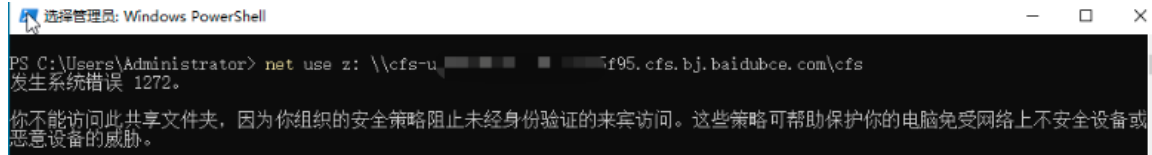
1. 登录到挂载CFS文件系统的云服务器BCC。
2. 使用以下命令卸载文件系统。

```
net use <目标盘符>: /del
```

其中，<目标盘符>为需要卸载的CFS文件系统盘符。

3. 执行net use命令，查看卸载结果。如果回显中未找到您挂载的CFS文件系统信息，则表示该文件系统目录已成功卸载。

常见问题 系统错误1272该如何处理？ 如果Windows Server出现如下报错信息（发生系统错误1272），是因为Windows系统因安全策略阻挡了以来宾访问权限（Guest Auth）访问SMB文件系统的用户



解决方案：以管理员身份运行如下powershell命令

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v "AllowInsecureGuestAuth" /t REG_DWORD /d 1 /f
net stop lanmanworkstation /y ; net start lanmanworkstation ; net start SessionEnv
```

或以管理员身份运行如下cmd命令：

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters" /v "AllowInsecureGuestAuth" /t REG_DWORD /d 1 /f
net stop lanmanworkstation /y && net start lanmanworkstation && net start SessionEnv
```

🔗 新购BCC时挂载CFS文件系统

本文档介绍如何在购买BCC时完成对CFS文件系统的挂载。 **前提条件**

- 已创建文件系统，具体操作请参见：[创建文件系统](#)。
- 已添加挂载点，具体操作请参见 [添加挂载点](#)。

使用限制

目前，新购BCC挂载CFS文件系统的配置存在如下限制：

1. 挂载目录：仅支持挂载CFS文件系统的根目录，不支持配置挂载CFS子目录。
2. 挂载数量：支持最多挂载5个CFS文件系统

说明：如果目前的BCC配置挂载CFS文件系统功能无法满足业务需要，请在BCC创建后，手动挂载CFS文件系统。具体操作可参见：[单台BCC挂载与卸载NFS协议CFS](#)、[多台BCC批量挂载与卸载NFS协议CFS](#)、[单台BCC挂载与卸载SMB协议CFS](#)、[Windows系统挂载与卸载SMB协议CFS](#)。

挂载CFS文件系统

1. 登录[云服务器BCC控制台](#)。
 2. 创建BCC实例，创建步骤请参见[通过购买页创建实例](#)。
- 在自定义购买页面存储模块添加文件存储。

存储

*系统盘：

*类型

通用型SSD

*容量

20

GIB

性能

基础性能：2400 IOPS 130 MBps

性能对比

数据盘：?

+ 创建数据盘 (0/5)

+ 从已有快照创建数据盘

如何选择云磁盘，请查看详细说明>>

共享盘：

请选择文件系统 ID/名称

请选择挂载地址

请输入挂载目录（例如：/mnt）

NFS4.1

C

-

+ 文件存储 (1/5)

去CFS创建文件系统 ?

绑定快照策略：

关

推荐开启

利用自动快照策略定期针对云盘进行备份，以应对病毒感染、数据误删等风险。[快照计费介绍 >>](#)
CDS磁盘快照已于2020年3月13日开始计费。若您不希望产生快照存储费用，请关闭自动快照策略
若您在后续使用过程中开启自动快照或创建手动快照，创建后将按照快照占用的存储空间进行计费，若不创建则不会产生快照存储费用。[查看详细说明 >>](#)

详细配置参数

配置参数	配置内容
文件系统	- Linux系统仅支持选择NFS协议的文件系统。 - Windows系统仅支持选择SMB协议的文件系统。
挂载地址	- 用于CFS和BCC间的网络连通，需要和BCC属于同一专有网络VPC。 - 如无选项，请为当前文件系统添加挂载点。具体操作，请参见添加挂载点。
挂载目录/盘符	从BCC上访问CFS的本地路径或本地盘符。例如，Linux系统可以填/mnt/cfs，Windows系统可以填z。
协议类型	- NFS文件系统选择NFSv4.1即可 - SMB文件系统选择SMB即可。

验证挂载结果 Linux系统

- 登录到挂载CFS文件系统的云服务器BCC。
- 通过以下命令查询是否挂载成功

```
df -h
```

如果返回如下信息，则代表挂载成功

```
root@ ~# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.9G   0  3.9G   0% /dev
tmpfs           3.9G   0  3.9G   0% /dev/shm
tmpfs           3.9G  1.2M  3.9G   1% /run
tmpfs           3.9G   0  3.9G   0% /sys/fs/cgroup
/dev/vda1       20G   3.3G   16G  18% /
tmpfs           796M   0  796M   0% /run/user/0
cfs-yhjlqMja    10P   0   10P   0% /mnt
```

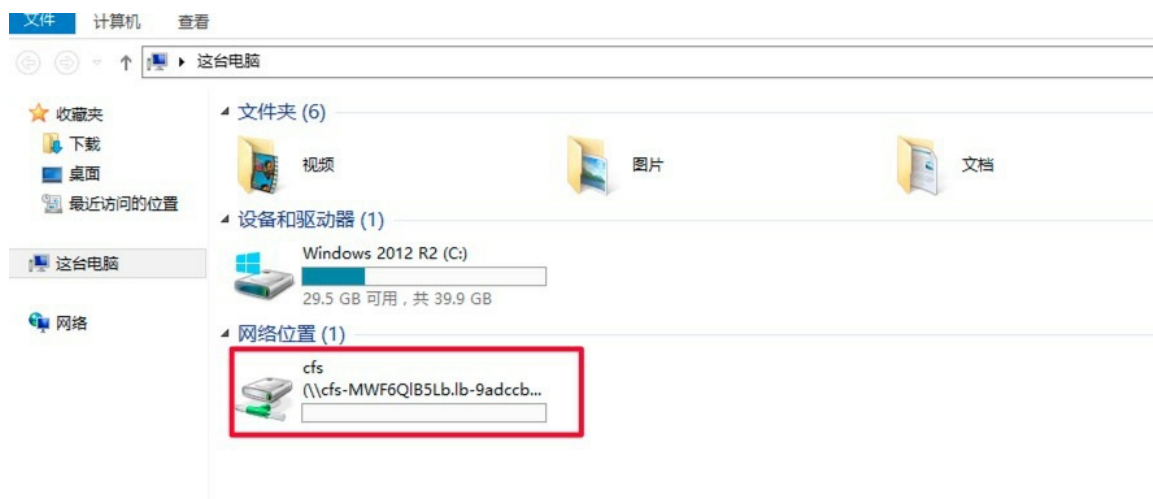
- 挂载成功后，您可以在BCC上把CFS文件系统当作一个普通的本地目录来访问和使用。

Windows系统

- 登录到挂载CFS文件系统的云服务器BCC。
- 可在Windows cmd命令行或Powershell中使用以下命令查看已挂载的文件系统。

```
net use
```

或直接打开Windows资源管理器查看网络位置中的已有盘符。



3. 挂载成功后，您可以在BCC上把CFS文件系统当作一个普通的本地目录来访问和使用。

CCE使用CFS

在CCE中使用CFS文件系统的详细操作，请参见 [使用文件存储CFS](#) 文档。

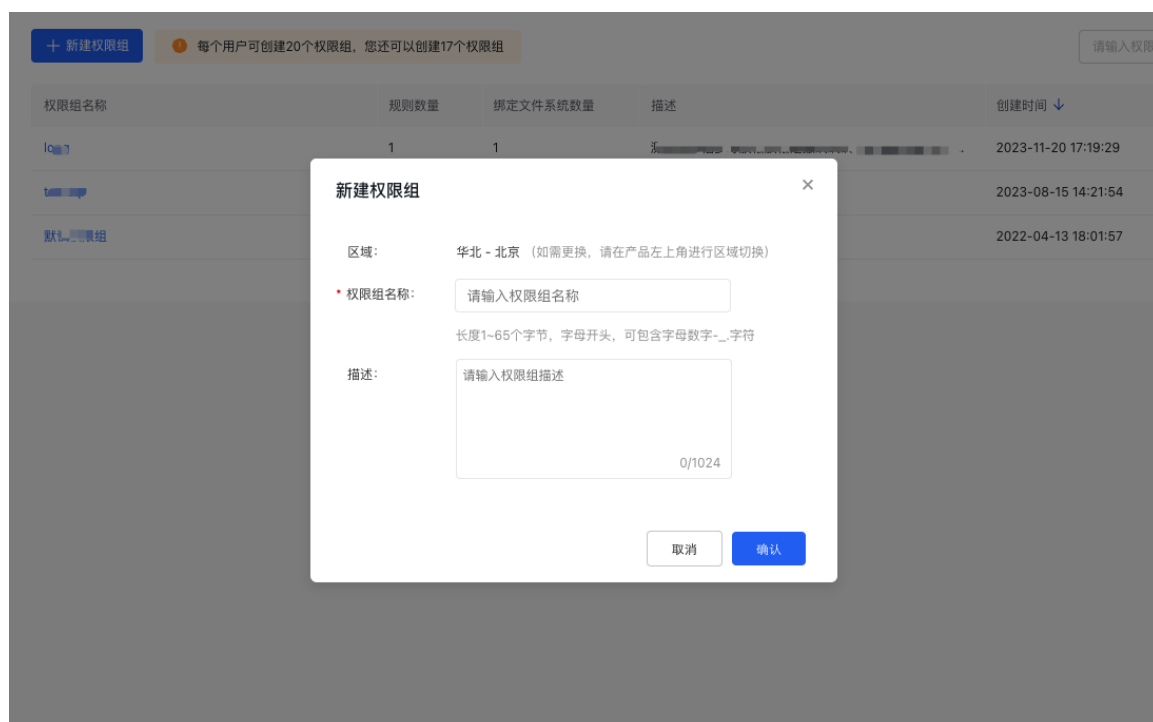
权限组管理

新建权限组

进入百度智能云 [CFS 服务管理控制台](#)，在左侧导航栏点击【权限组】，进"CFS权限组"页面。



在 CFS 权限组页面，点击【新建权限组】，在弹出的窗口中设置权限组名称和描述，完成权限组的创建。



新建好的权限组出现在权限组页面的列表中，支持对已有权限组的管理和删除。

权限组

帮助文档

+ 新建权限组

每个用户可创建20个权限组，您还可以创建16个权限组

请输入权限组名称进行搜索

权限组名称	规则数量	绑定文件系统数量	描述	创建时间	操作
test	0	0	--	2024-09-29 10:43:55	管理 删除
img	1	1		2023-11-20 17:19:29	管理 删除
testimg	0	1	--	2023-08-15 14:21:54	管理 删除
默认权限组	2	55	读写	2022-04-13 18:01:57	管理 删除

添加权限组规则

点击权限组名称，进入该权限组的详情页面，可以看到规则信息列表，支持对已有对规则进行编辑和删除。

返回 权限组名称: test

权限组信息

规则数量: 1 绑定文件系统数量: 0 创建时间: 2024-09-29 10:43:55

描述: 编辑

规则信息

权限组是一种白名单机制，您需要通过添加权限组规则授予指定的源IP地址访问文件系统的权限。

+ 新建规则

授权地址类型: IPv4

授权地址	权限	用户范围	优先级	操作
10.10.10.0/32	只读	限制所有用户	1	编辑 删除

每页展示 10 1 >

点击【新建规则】，在弹出的窗口中设置权限组规则的授权地址、权限、用户范围及优先级，完成权限组规则的创建。

绑定文件系统数量: 0 创建时间: 2024-09-29 10:43:55

权限组规则授予指定的源IP地址访问文件系统的权限。

权限

只读

新建规则

* 授权地址:

请输入授权地址

虚拟VPC IP, 允许单IP或网段, 支持IPv4和IPv6, 如10.10.1.123或2001::/16

* 权限:

只读

读写

* 用户范围:

限制所有用户

* 优先级:

1

优先级可选范围为1-100, 默认值为1, 即最高优先级

取消

确认

权限组规则设置的具体说明如下：

字段	说明
授权地址	虚拟机 VPC IP，可以填写单个 IP 或网段，如：10.10.1.123或192.168.3.0/24。
权限	只读或者读写，限制授权对象对文件系统的访问权限。
用户范围	限制来访用户对文件系统的访问权限。1) 不限制 root 用户：root 用户具有访问文件系统的权限；2) 限制 root 用户：root 用户访问时被映射为匿名用户；3) 限制所有用户：所有用户访问时都被映射为匿名用户。其中，匿名用户是 Linux 系统的默认用户，只可访问服务器上的公共内容。注意：该权限项仅对 NFS 协议的文件系统访问生效，SMB/CIFS 协议的文件系统不支持该权限项，配置后不生效。
优先级	优先级为1-100中的整数，数字越小优先级越高，高优先级规则覆盖低优先级规则；默认值为1，即为最高优先级。目前不支持多条规则共享同一优先级。

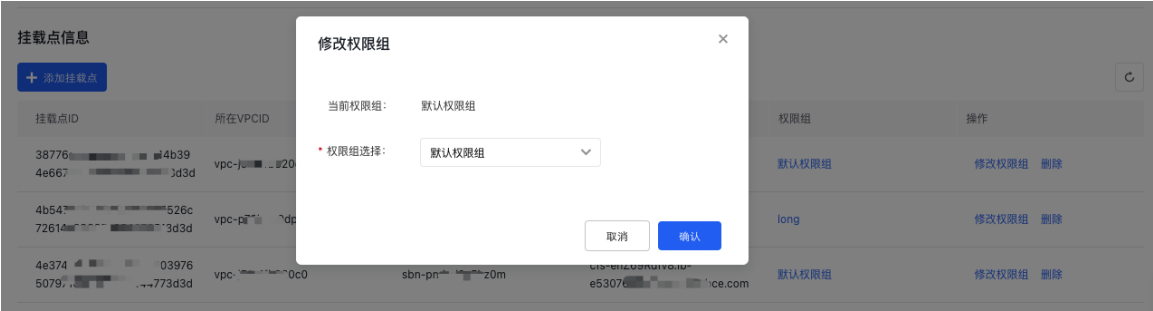
应用权限组规则

选择需要配置权限组的文件系统，进入文件系统详情，针对需要配置的挂载点进行权限组的绑定。目前支持两种配置方式：

1) 新建挂载点时选择已经配置好的权限组。



2) 创建挂载点时默认权限组，配置好权限组以后，点击挂载点信息列表的【修改权限组】操作，进行权限组的配置。



多用户访问控制

基本概念

文件存储CFS已经接入了百度智能云IAM平台，用户可以在IAM平台创建子用户，对云上业务进行细粒度管理和使用。

百度智能云IAM子用户有如下特点：

- 1. 所有IAM子用户的资源都隶属于主用户。尽管某些权限的子用户可以创建资源，但计费主体依然是主用户。
- 2. 子用户可以独立使用管理控制台和API。
- 3. 主用户可以对子用户进行授权，在IAM里授权是通过给予用户关联策略来完成的，一个子账号可以关联多个策略。

🔗 IAM子用户应用场景

用户可以在百度智能云上通过主子用户体系，灵活使用CFS资源。

- 企业子用户管理与分权
企业A使用百度智能云账号购买了多种云资源（如BCC实例/CFS实例/BLB实例/BOS存储/...），A的员工需要操作这些云资源包括购买、运维、线上应用等。不同岗位员工的工作职责不一样，需要的权限也不一样。出于安全考虑，A不希望将主账号的密钥直接公布给员工，而希望能给不同岗位员工创建相应的用户子账号。用户子账号只能在授权的前提下操作资源，不需要进行独立的计量计费，所有资源费用都归属主账号。A主账号随时可以撤销子账号的权限，也可以随时删除其创建的子账号。
- 企业之间的资源操作与授权管理
A和B代表不同的企业。A购买了多种云资源（如BCC实例/CFS实例/BLB实例/BOS存储/...）来开展业务。A希望能专注于业务系统，而将云资源运维监控管理等任务委托或授权给企业B。企业B也可以进一步将代运维任务分配给B的员工，即B为其员工创建相应的用户子账号供其使用。B可以精细控制其员工对A的云资源操作权限。如果A和B的这种代运维合同终止，A随时可以撤销对B的授权。

🔗 策略说明

CFS在云上提供了系统策略和自定义策略两种策略模式。

系统策略，主用户对子用户授予后，子用户可对主用户名下所有CFS资源进行相应的操作能力，共有如下三种系统策略：

- CFSFullControlAccessPolicy：管理百度智能云文件存储服务（CFS）的权限。
- CFSOperateAccessPolicy：运维百度智能云文件存储服务（CFS）的权限。
- CFSReadOnlyAccessPolicy：只读百度智能云文件存储服务（CFS）的权限。

三种系统策略对应的权限，与CFS的API对应关系如下：

系统策略	可操作API
管理	CFS所有API
运维	Updatefilesystem
	Describefilesystem
	Creatmounttarget
	Deletemounttarget
只读	Describefilesystem
	Describemounttarget

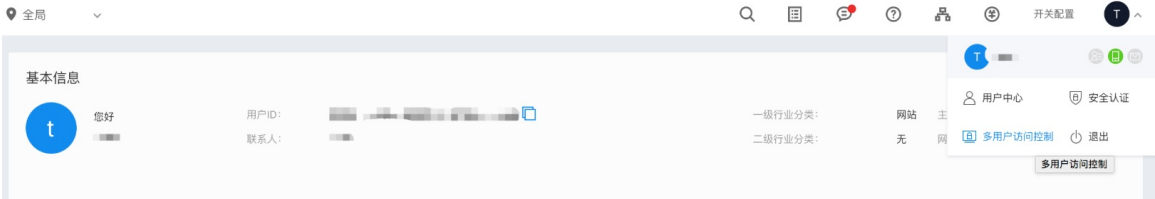
自定义策略，主用户可以按Region、按细粒度资源为子用户授权。可授予以下两种权限类型：

自定义策略	可操作API
运维	Updatefilesystem
	Describefilesystem
	Creatmounttarget
	Deletemounttarget
	Describemounttarget
只读	Describefilesystem
	Describemounttarget

🔗 操作步骤

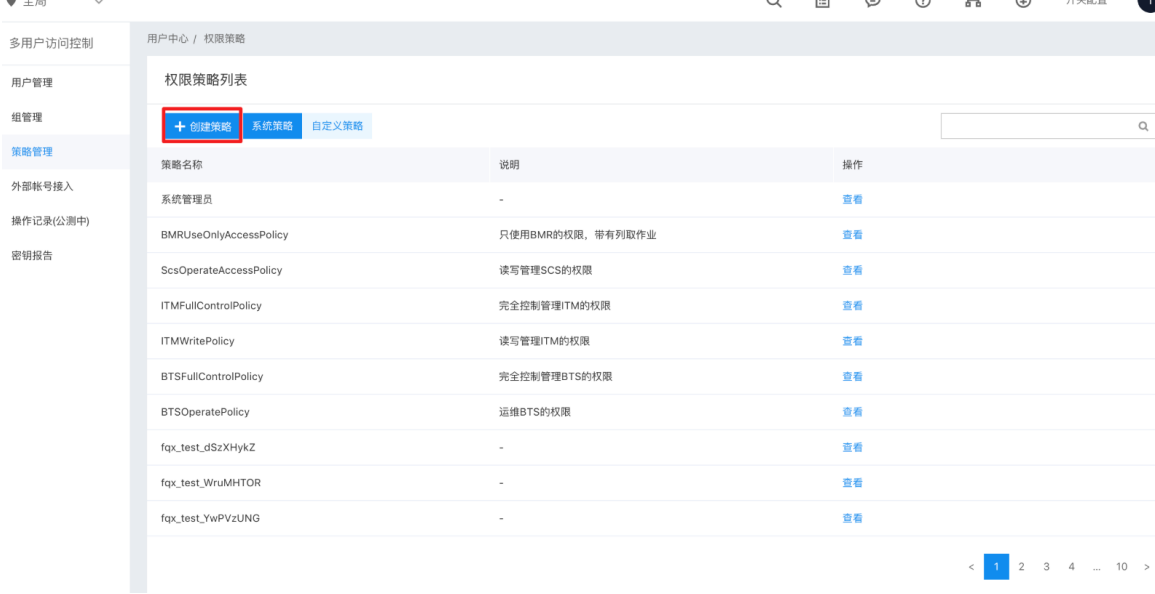
1. 策略创建

进入“管理控制台”，选择“多用户访问控制”



选择“策略管理”，搜索CFS，可以看到CFS相关系统策略。

当预期使用自定义策略时，进入策略管理页面，选择“创建策略”。



在自定义策略创建页面，为该自定义策略命名之后，需要筛选“服务类型”为CFS，并为预期授权的实例选择对应的权限。在资源选择方面，主用户可在列表中查看目前已创建的CFS实例，并选择预期授权的实例。

权限配置

* 服务类型：

文件存储 CFS

* 策略生成方式：

策略生成器

编辑策略文件

* 实例：

☒ 只读

☐ 运维

[权限说明](#)

* 资源选择：

*一次只能配置一个区域

☒ 华北 - 北京

请输入关键词进行搜索

Q

已选择 0 个资源

☐ 系统ID	系统名称	系统类型	系统协议
☐ cfs-6sGWPEJXqE	asdasdas	cap	nfs
☐ cfs-inPp4jGO15	adasdasd	cap	nfs

之后，在自定义策略列表中，即可看到新创建的自定义策略了。

注意：在使用CFS产品时，如果需要为CFS实例添加/修改挂载点时，会需要用户从名下已有的VPC、子网的列表中进行筛选，作为挂载点的挂载信息。因此，当主用户为子用户授予具有管理、运维权限的系统策略或自定义策略时，也需同时在百度智能云IAM平台为该子用户授予对VPC、子网产品具有只读或更高权限的策略。

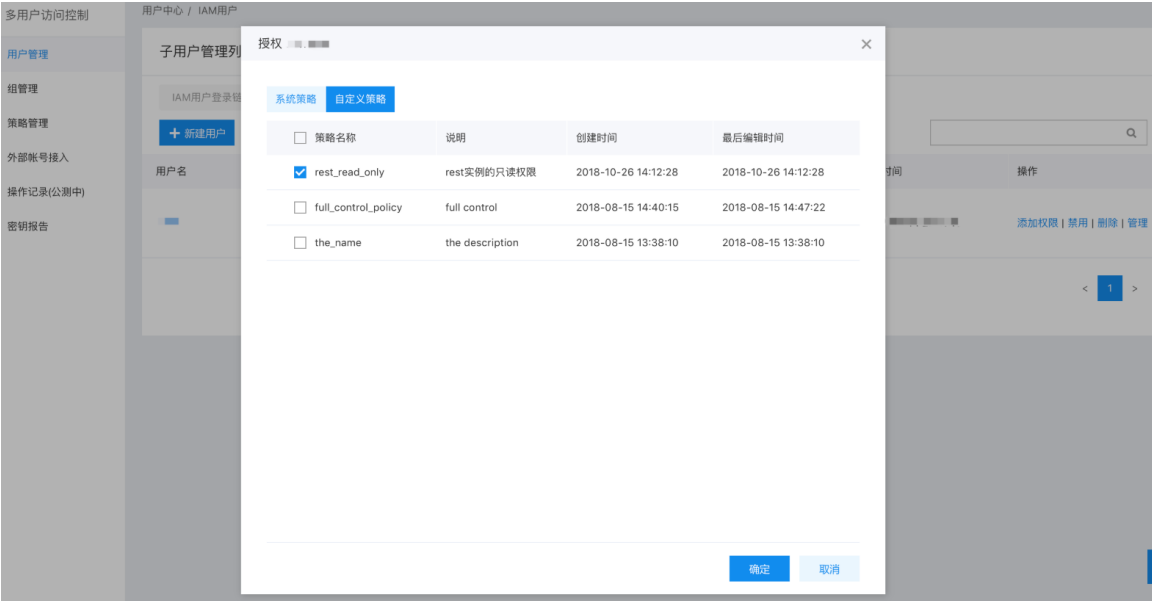
2. 为子用户授权

当策略创建完成后，即可为子用户授予策略。

在“用户管理”页签为子用户添加权限。



选择“系统策略”或“自定义策略”，获取相应策略列表，选择希望授予子用户的权限，点击“确定”即可将权限赋予子用户。



删除子用户授权：点击用户名，进入“子用户详情页面”，“权限信息”处可以看到此用户拥有的权限，点击“删除”可以将此权

限从用户权限中去掉

多用户访问控制

用户管理

组管理

策略管理

外部帐号接入

操作记录(公测中)

密钥报告

双因素认证

开启子用户短信验证

AccessKey

+ 新建AccessKey

AccessKeyID	创建时间	操作
AKIAI44QH8DHBEXAMPLE	2015-12-14 15:19:30	删除
AKIAI44QH8DHBEXAMPLE	2015-12-14 15:19:30	删除

权限信息

+ 授权

策略名称	策略描述	操作
rest_read_only	rest实例的只读权限	删除
rest_read_only	rest实例的只读权限	删除

操作日志

上次访问时间	资源名称	操作
2015-12-14 15:19:30	rest实例的只读权限	删除

数据加密

为了更好地保障CFS文件系统中存储数据的安全性和合规性，文件存储CFS支持文件系统加密。当您开启文件系统加密功能后，CFS会对写入文件系统的数据进行加密，读取数据时，CFS文件存储自动将加密数据解密后返回给用户。本文主要介绍文件系统加密的使用限制、加密方式和相关操作。

前提条件

首次使用KMS服务（KMS服务为付费产品，使用前请详细查看[KMS计费说明](#)）进行数据加密时，先[开通KMS服务](#)，并在KMS服务中创建密钥，授权CFS文件存储获取KMS密钥。

使用限制

- 设置文件系统加密仅支持在创建CFS文件系统时开启。
- 已开启文件系统加密功能的CFS文件系统暂时无法关闭。
- 文件系统加密功能仅支持在北京、广州、苏州地域使用。

加密方式

当您对CFS文件存储中存储数据有高安全性和合规性要求时，建议您开启文件系统加密功能。文件系统加密密钥支持行业常见标准AES、RSA、SM1、SM2、SM4等加密算法。文件系统加密密钥依托于KMS服务生成和管理，KMS服务是一款安全管理类服务，底层符合国家要求采用HSM设备安全妥善地保管密钥，能最大程度保障密钥的保密性、完整性和可用性。更多信息，请参考[百度智能云密钥管理服务](#)。

操作方式

开启文件系统加密

- 1.在[CFS文件存储控制台](#)新建文件系统时，选择开启文件系统加密功能。
- 2.选择一个“启用中”状态的KMS密钥，单击“确定”后创建加密文件系统。

新建文件系统×

可用区：

可用区A

?

协议类型：

NFS

存储类型：

性能型

自定义容量：

关闭

开启

系统默认自动扩容，开启自定义容量后需手动调整容量上限

文件系统加密：

关闭

开启

?

开启加密后，写入数据将被自动加密，并在读取时自动解密

选择KMS密钥：

32f4da9e-3 j14d6ea97109 启用中

绑定标签：

标签键 请选择已有或手动输入 值 请选择已有或手动输入 删除

+ 添加标签

标签管理

帮助文档

确定

取消

🔗 密钥变更

如果您在使用文件系统加密过程中，想要变更KMS密钥，可在对应的CFS文件系统基本信息页中进行密钥变更。

基本信息	基本信息		
备份	实例ID： cf- j6	实例名称：	状态： 运行中
监控	描述信息：	协议类型： NFS	存储类型： 性能型
	地域： 华北 - 北京 可用区D	已用容量： 0 bytes	容量上限： 容量变更
	加密状态： 已加密 密钥变更	创建时间： 2024-01-30 16:30:42	

管理标签

百度智能云提供标签管理功能，通过为CFS文件系统添加标签，从而帮助您对文件系统进行快速分类和识别管理。本文档主要介绍标签的使用限制、如何添加标签、查看标签、编辑标签、删除标签和通过标签筛选等。

🔗 使用限制

在使用标签前，请先了解以下使用限制：

- 标签：每个标签由键和值两部分组成，标签（键+值）唯一
- 每个用户最多可以创建200个标签
- 当一个文件系统添加多个标签时，标签键不允许重复
- 批量编辑标签时，选择CFS文件系统限制10个，添加标签限制100个。

49

添加标签

您可以按照以下步骤为新建的文件系统添加标签：

- 1. 登录[CFS控制台](#)，在文件系统列表页点击【新建文件系统】按钮进入新建文件系统弹窗。
- 2. 在绑定标签字段中，根据业务需要选择已有或手动输入合适的标签键和值，并根据实际情况添加多个标签。
- 3. 您可以点击【标签管理】进入到标签页，按需创建标签。

新建文件系统

备注信息：

请输入实例的备注信息

0/256

区域：

华北 - 北京

* 可用区：

可用区A

?

协议类型：

NFS

存储类型：

性能型

自定义容量：

☒ 关闭

☐ 开启

系统默认自动扩容，开启自定义容量后需手动调整容量上限

文件系统加密：

☒ 关闭

☐ 开启

?

开启加密后，写入数据将被自动加密，并在读取时自动解密

生命周期管理：

☐ 关

达到设定时间，则会被转化为“低频型”存储类型

绑定标签：

标签键

请选择已有或手动输入

值

请选择已有或手动输入

删除

+ 添加标签

标签管理

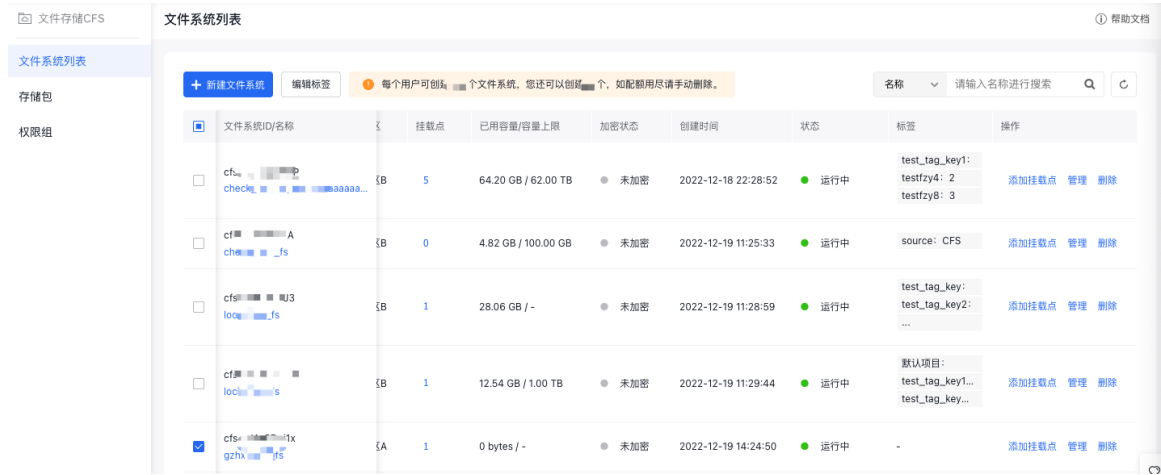
帮助文档

确定

取消

您也可以按照以下步骤为已建的文件系统添加标签：

- 1. 登录[CFS控制台](#)，在文件系统列表页选择未添加标签的文件系统（支持批量操作），点击【编辑标签】。



2. 在编辑标签弹窗中，根据业务需要选择已有或手动输入合适的标签键和值，并根据实际情况添加多个标签。



查看标签

在文件系统列表页，您可以在文件系统列表中查看标签。



编辑和删除标签

您可以按照以下步骤为已添加标签的文件系统进行重新编辑或删除：

- 1. 登录CFS控制台，在文件系统列表页选择需要重新编辑或删除标签的文件系统（不支持批量操作），点击【编辑标签】。
- 2. 在编辑标签弹窗中，对已有标签进行标签键和值修改、标签删除和标签添加等操作。

编辑标签

标签支持您按各种标准（如用途、所有者或项目）对资源进行分类；每个标签包含键和值两部分

标签键

test_tag_key1

值

请选择已有或手动输入

删除

标签键

testfzy4

值

2

删除

标签键

testfzy8

值

3

删除

+ 添加标签

标签管理

帮助文档

取消

确定

通过标签筛选

您可以通过以下步骤筛选包含指定标签键信息的文件系统：

1. 登录CFS控制台，在文件系统列表页上方，将筛选项切换成标签，并选择待筛选标签键。



2. 输入对应的标签值，点击【搜索】即可在列表中查看到符合搜索信息的文件系统。

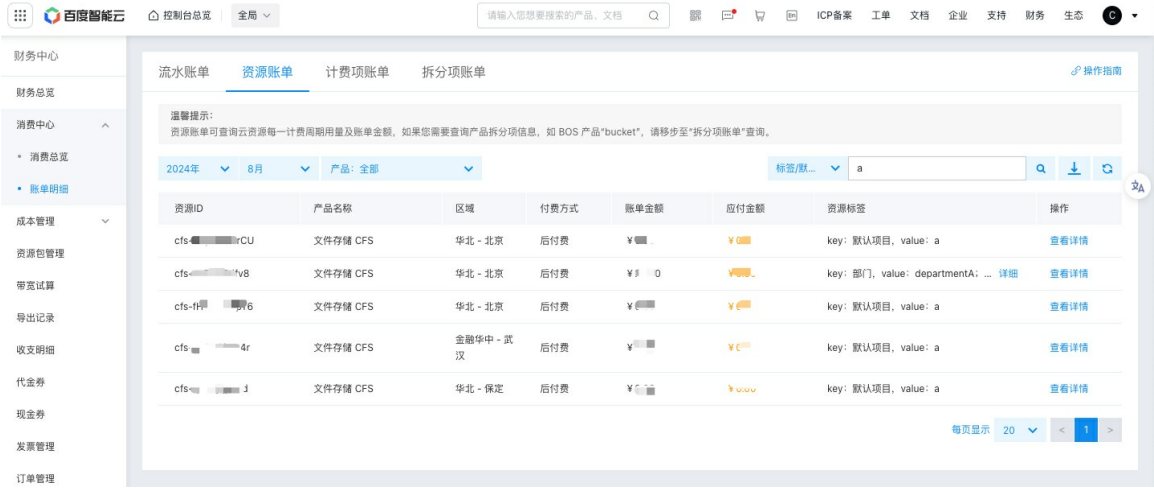
查看资源账单

资源账单是记录每个资源的生命周期与消费相关的数据的功能，方便用户以资源为维度查看资源及产品的消费数据。您可以按照以下步骤查看对应标签下PFS实例的资源账单：

1. 登录CFS控制台，在文件系统列表页选择一个文件系统，点击页面上方【编辑标签】
2. 在弹框中点击【标签管理】，进入百度智能云标签管理页面。
3. 在绑定文件存储CFS的操作栏中点击【查看账单】。



4. 进入资源账单页面，选择需要查看账单的年月，会显示当前标签资源的消费信息。还可以通过组合筛选条件执行标签搜索，按标签键和值进行选择其他标签的消费详情，并下载相关信息。



备份

为了保护重要业务资产，文件存储CFS支持备份与恢复功能。CFS在备份文件系统数据时不进行文件系统快照，而是基于高效的文件系统扫描机制。在CFS控制台通过设置备份策略生成多份备份副本数据（首次全量备份，后续增量备份），并在数据丢失或受损时及时恢复文件。

费用说明

CFS 的备份功能当前处于免费试用期，待免费试用期结束进入收费阶段前会统一发布具体计费说明。

备份

CFS备份支持手动备份和自动备份，它们之间的区别在于备份的触发和执行方式。手动备份需要用户手动触发和执行备份操作。自动备份是指备份过程的触发和执行是自动进行的，无需用户干预。用户可以事先设置备份计划、规则和参数，然后备份系统会根据设定的时间间隔或特定事件自动执行备份操作。

手动备份

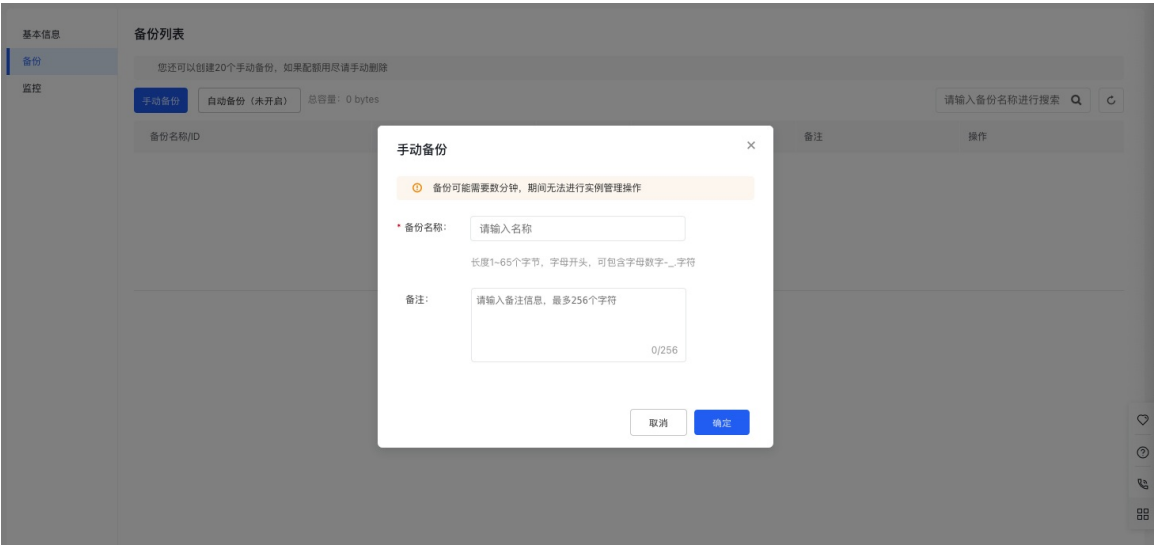
1. 进入CFS控制台，点击操作列的“管理”进入基本信息页。



2. 点击左侧“备份”，进入备份列表页。



3. 点击“手动备份”，在弹出的页面中填写“备份名称”和“备注”。



4. 点击“确定”后，能够看到备份任务详情。



🔗 自动备份

- 1. 进入备份页面（操作同手动备份），点击“自动备份（未开启）”。
- 2. 在弹出的页面中填写“备份时间”、“重复日期”、“保留时长”等信息。



- 3. 点击“确定”后，会发现自动备份已开启。
- 4. 如需修改备份策略或关闭自动备份，可再次点击“自动备份（已开启）”进行修改。

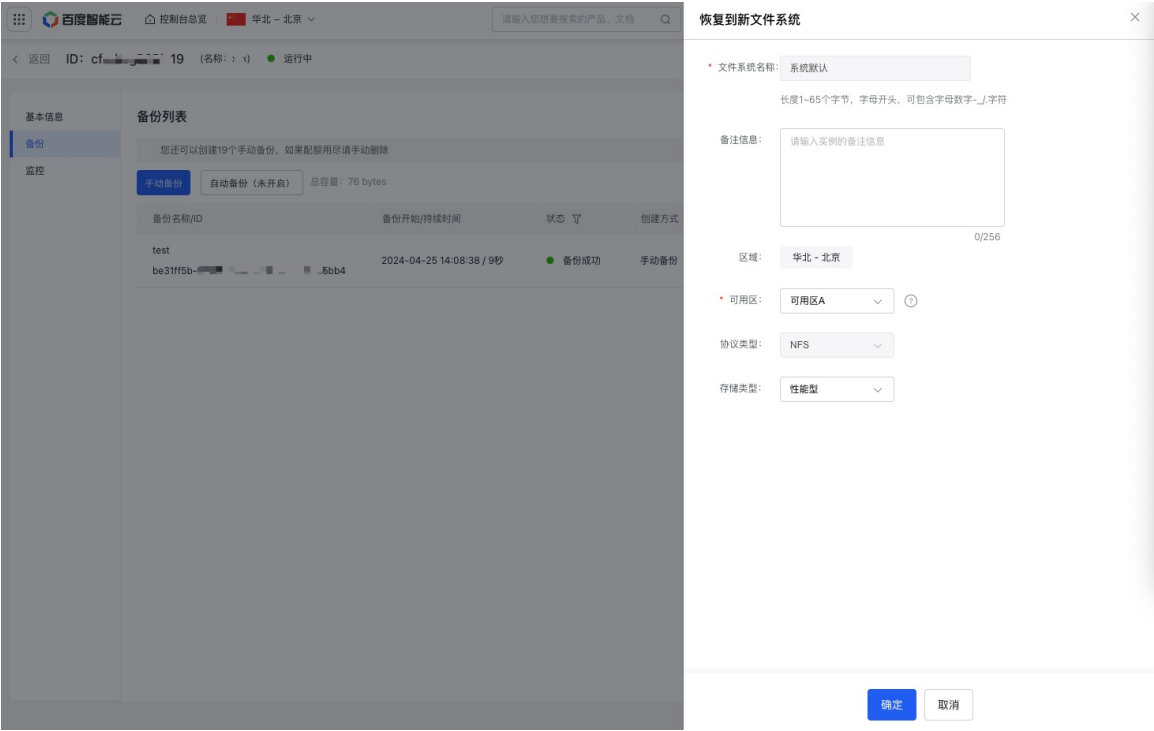
🔗 恢复

当前CFS仅支持文件系统全量恢复至新文件系统。具体操作如下：

- 1. 在备份列表页面，对于已完成的备份任务，可点击操作列的“备份恢复”进行文件系统恢复。



2. 在弹出的页面中选择要恢复的“可用区”及“存储类型”。



3. 点击确定，经一定时间后可看到恢复完成的新文件系统。



常见问题

1. CFS备份时是否会影响文件系统性能？

答：备份数据通过旁路进行，因此不会影响CFS文件系统的性能。

监控与报警

查看监控

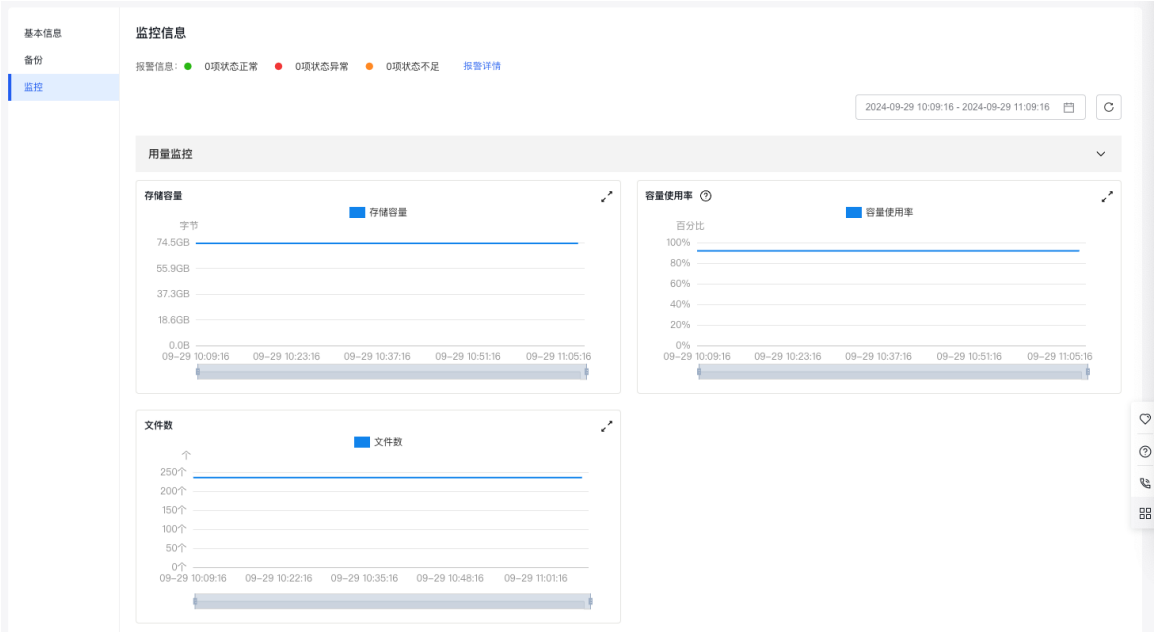
您可以进入[文件存储CFS控制台](#)或[云监控BCM控制台](#)查看监控数据。

通过CFS控制台查看

- 1. 进入[文件存储CFS控制台](#)，依次点击“文件系统列表 > 管理”



2. 点击“监控”，查看文件系统的监控图表

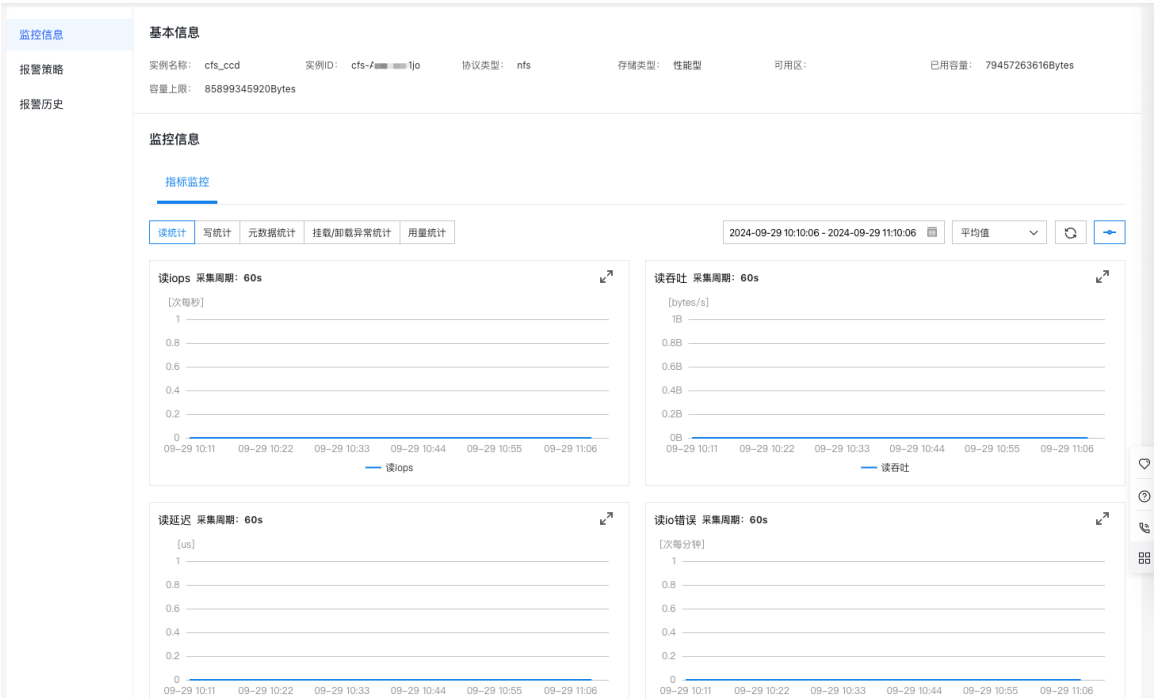


通过BCM控制台查看

1. 进入云监控BCM控制台，依次点击“云产品监控 > 实例监控 > 文件存储CFS”，然后点击对应的文件系统名称



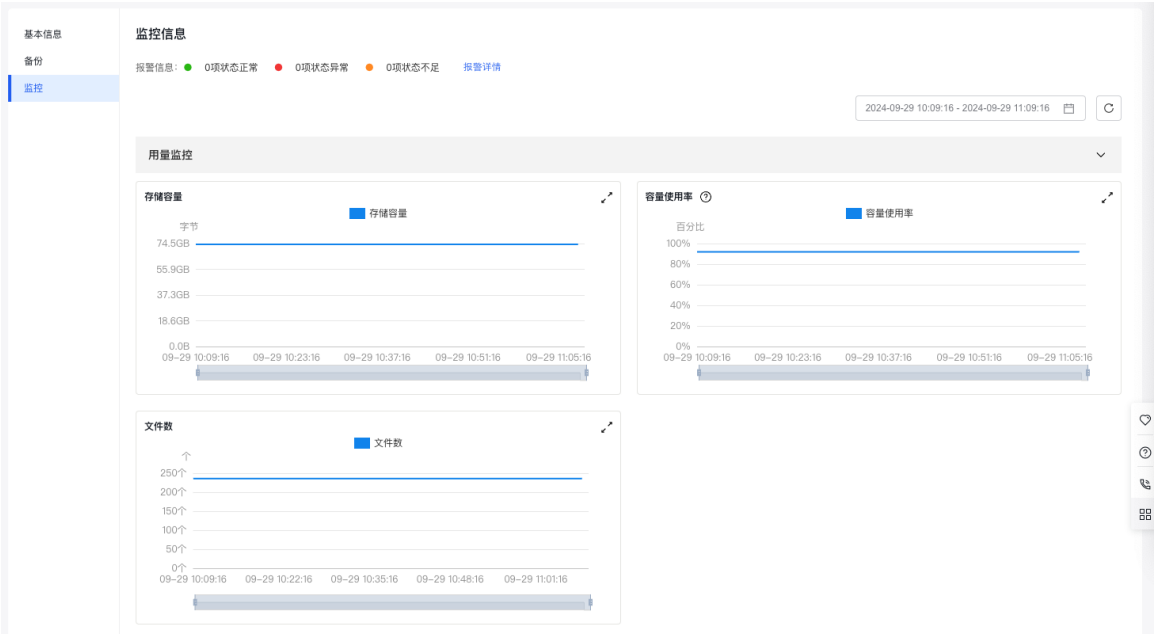
2. 通过“监控信息”查看文件系统的监控图表



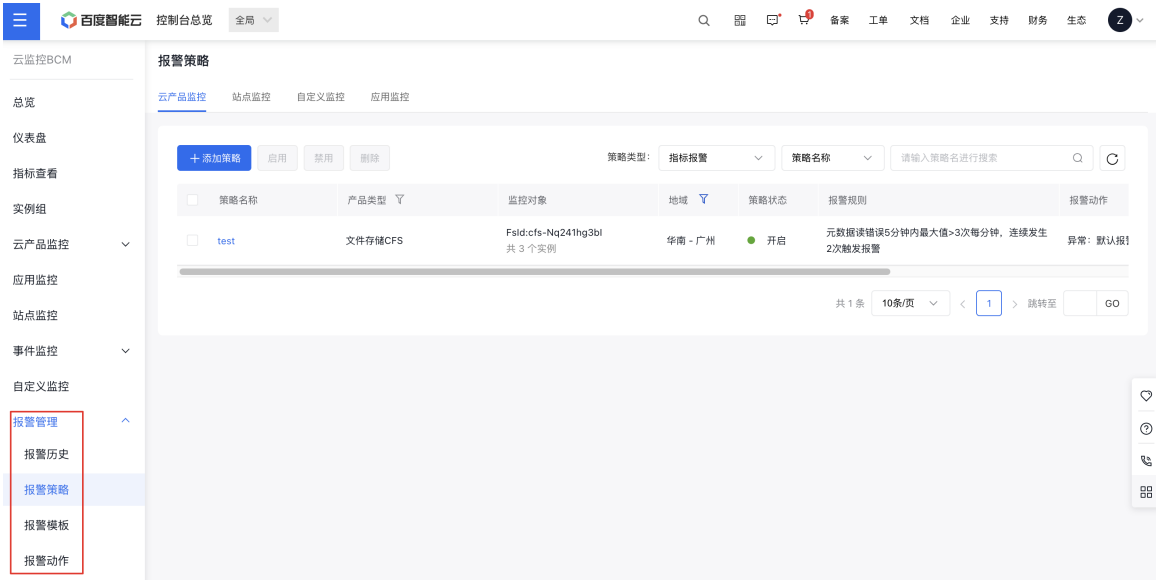
报警管理

查看报警详情

查看方式一：进入[文件存储CFS控制台](#)，依次点击“文件系统列表 > 管理 > 监控 > 报警详情”，进入报警管理子页面。

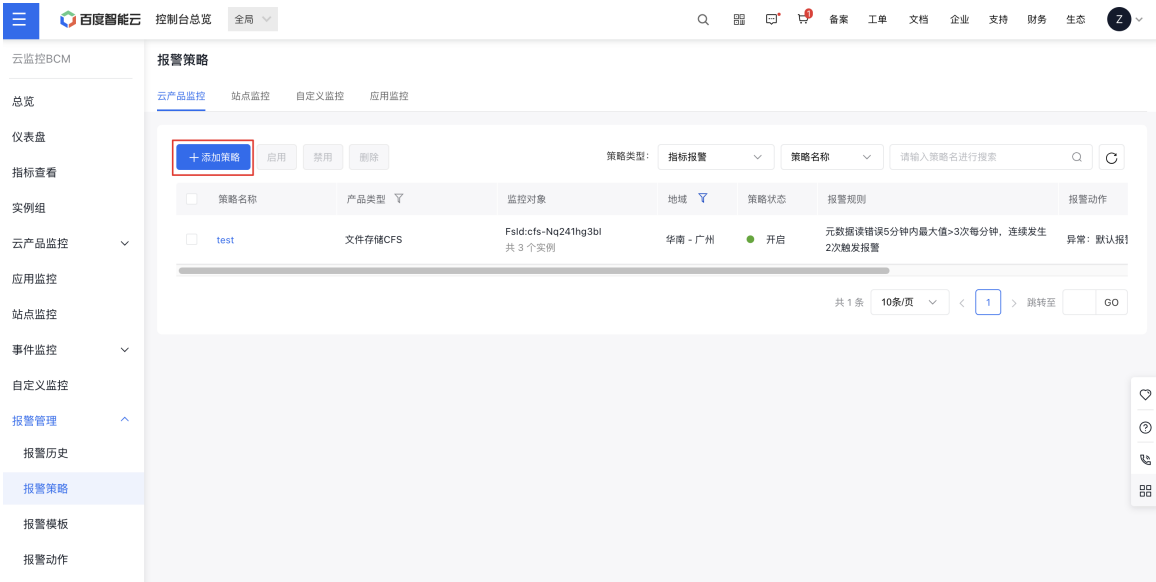


查看方式二：进入[云监控BCM控制台](#)，点击“报警管理”，可查看报警历史、报警策略、报警模板、报警动作等信息。

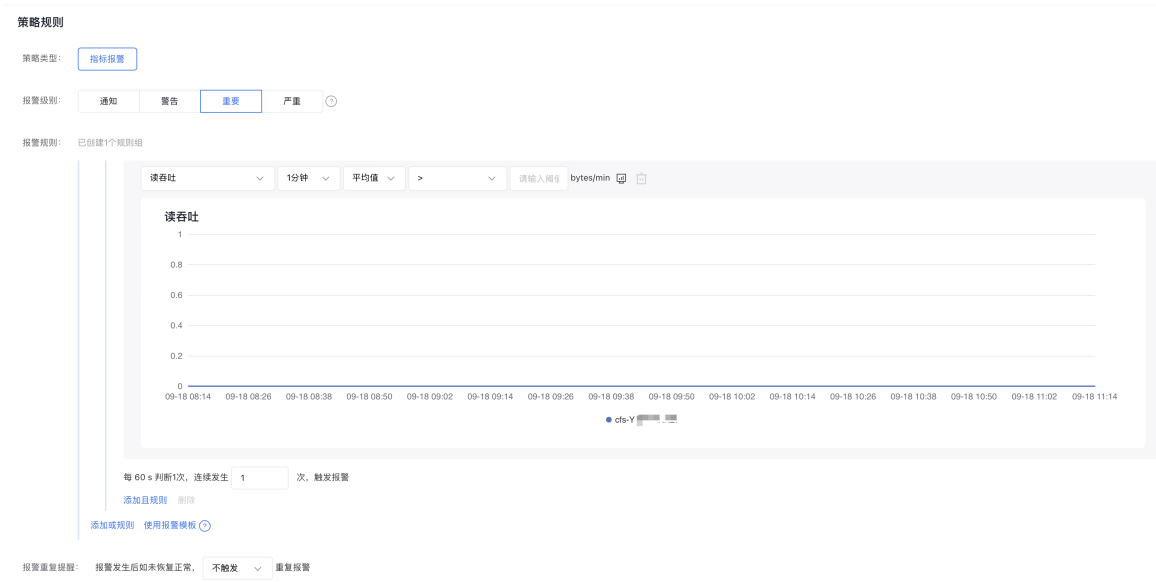


添加报警策略

1. 进入 [云监控BCM控制台](#)，依次点击“报警管理 > 报警策略 > 添加策略”，设定相应监控项的报警阈值和通知方式。具体操作请参考[报警动作](#)。



2. 在"添加策略"页面，您可以自定义报警策略。



管理报警策略

进入[云监控BCM控制台](#)，点击“**报警管理**”，您可以对已有的报警策略停用和删除，停用的报警策略可以再次启用。具体操作请参考[管理报警](#)。

指标含义

目前文件系统支持如下指标的监控与报警：

指标分类	指标英文名称 (Metric name)	指标中文名称	单位	说明
读统计	ReadThroughputNew	读吞吐	字节每秒	该文件系统在周期内的读吞吐
	ReadLatencyNew	读延迟	ms	该文件系统在周期内平均读延迟
	ReadIOWNew	读iops	次每秒	该文件系统在周期内的读io次数
	ReadIOErrorCount	读io错误	次每分钟	该文件系统在周期内读io错误次数
写统计	WriteThroughputNew	写吞吐	字节每秒	该文件系统在周期内的写吞吐
	WriteLatencyNew	写延迟	ms	该文件系统在周期内平均写延迟
	WriteIOWNew	写iops	次每秒	该文件系统在周期内的写io次数
	WriteIOErrorCount	写io错误	次每分钟	该文件系统在周期内写io错误次数
元数据统计	MetaReadQpsNew	元数据读qps	次每秒	该文件系统在周期内平均元数据读io次数
	MetaWriteQpsNew	元数据写qps	次每秒	该文件系统在周期内平均元数据写io次数
	MetaReadErrorCount	元数据读io错误	次每分钟	该文件系统在周期内元数据读io错误次数
	MetaWriteErrorCount	元数据写io错误	次每分钟	该文件系统在周期内元数据写io错误次数
挂载/卸载异常统计	MountErrorCount	挂载失败次数	次每分钟	该文件系统在周期内的挂载错误次数，仅统计服务端收到请求但未正常处理的情况。如果监控图表无数据展示，表示没有挂载失败
	UmountErrorCount	卸载失败次数	次每分钟	该文件系统在周期内的卸载错误次数，仅统计服务端收到请求但未正常处理的情况。如果监控图表无数据展示，表示没有卸载失败
用量统计	Capacity	存储容量	字节	该文件系统当前使用容量，不包括备份数据
	FileNum	文件数	个	该文件系统在周期内的文件数量，不包含目录
	UsagePercentage	容量使用率	%	如果设置了文件系统容量上限，此处的容量使用率=已用容量/容量上限；如果没有设置，此处的容量使用率=0

云审计

云审计（Baidu CloudTrail, 简称BCT）是一项云上的审计服务，通过记录用户对云资源的关键操作行为，支持对云账户进行合规

性检查、操作审核和事件回溯。

BCT提供通过云管理控制台、部分已接入服务API调用的操作记录查询，并支持将这些关键的用户活动长期存储到BOS中，以支持企业和机构的合规审计、事件追溯等。通过使用BCT，可以简化对你的云账户进行安全性分析、资源更改跟踪和故障排除工作。

文件存储CFS具体支持的审计事件，请参见 [文件存储CFS](#) 文档。

典型实践

为不同目录设置不同的用户权限

🔗 问题背景

在很多情况下，一个公司或组织内部的不同职能部门、业务需要共享数据，同时赋予不同的权限以保证数据的安全性。这个时候，用户希望像使用本地文件系统那样，为不同的用户设置不同的访问权限。

NFS 协议支持 UNIX 风格的权限认证机制，是以 UID、GID，而不是用户名、组名来进行权限验证的。因此，如果从两个不同的虚拟机访问 CFS 文件系统，根据用户的 UID 在这些虚拟机上是相同还是不同，会有不同的体验：

- 如果两个虚拟机上同一用户的 UID 相同，CFS 会将其视为同一用户；
- 如果两个虚拟机上同一用户的 UID 不相同，则 CFS 会将其视为不同的用户；
- 如果不同虚拟机上，两个不同用户的 UID 相同，则 CFS 会将其视为同一个用户。

除了极个别特殊的用户（如 root），不同机器上同一用户的 UID、GID 大概率是不一样的。所以，需要做一些特别的设置，以使访问权限能够正确工作。

🔗 统一用户的 UID 和 GID

您可以按照以下方法在每个虚拟机上设置 UID 和 GID：

方法一：使用 usermode 和 groupmode 修改 UID 和 GID

1. 如果用户不存在，创建用户，假设要创建的用户名字为 cfs，并通过 id 命令查看 UID 和 GID。命令和示例输出如下：

```
[root@test-cfs ~]# useradd cfs
[root@test-cfs ~]# id cfs
uid=500(cfs) gid=500(cfs) groups=500(cfs)
```

2. 通过 usermode 和 groupmode 命令修改 cfs 账户的 UID 和 GID 到一个预先设定的值，例如 888。命令和示例输出如下：

```
[root@test-cfs ~]# usermod -u 888 cfs
[root@test-cfs ~]# groupmod -g 888 cfs
[root@test-cfs ~]# id cfs
uid=888(cfs) gid=888(cfs) groups=888(cfs)
```

方法二：直接修改 /etc/passwd 和 /etc/group 文件

第一个步骤和方法一一样，创建用户，创建完之后，分别打开 /etc/passwd 和 /etc/group 编辑。

vi /etc/passwd，找到其中 cfs 用户所在的行，修改 UID 和 GID。

修改前：

```
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
cfs:x:500:500:~/home/cfs:/bin/bash
```

修改后：

```
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
cfs:x:888:888:~/home/cfs:/bin/bash
```

输入 `:wq` 退出编辑模式并保存修改。

vi `/etc/group`，找到其中 `cfs` 用户所在的行，修改其所在的组。

修改前：

```
rpcuser:x:29:
nfsnobody:x:65534:
cfs:x:500:
```

修改后：

```
rpcuser:x:29:
nfsnobody:x:65534:
cfs:x:888:
```

输入 `:wq` 退出编辑模式并保存修改。

创建初始目录

默认情况下，只有 `ROOT` 用户可以创建目录和文件。所以，还需要以 `root` 账户为创建每个用户设置自己的根目录，并 `chown` 修改权限后，用户才能以自己的身份才能使用。示例代码如下（假设挂载点为 `/mnt`）：

```
[root@test-cfs ~]# mkdir /mnt/cfs
[root@test-cfs ~]# chown -R cfs:cfs /mnt/cfs
[root@test-cfs ~]# ls -l /mnt/
total 8
drwxr-xr-x 2 cfs cfs 4096 Jul 24 15:38 cfs
drwxr-xr-x 2 root root 4096 Jul 24 15:39 root
```

内核禁止使用非 `root` 用户挂载文件系统，所以在用户使用之前，仍然需要先以 `root` 账号挂载文件系统。这个步骤可以通过“自动挂载”来实现自动化。同时挂载的时候，可以直接以用户的根目录路径挂载，将使用范围局限到用户自己。

跨地域或账号使用文件系统

问题背景

一些情况下，用户有跨账号、区域使用文件系统的需求。典型的场景如下：

- 用户有好几个百度云账号，一个账号购买了 CFS，想要共享给另外一个用户使用；
- 用户在一个地域购买了文件系统，想从另外一个地域访问。

通常情况下，我们不推荐这样的使用方式。因为解决该类问题的原理是打通不同账号或区域的网络，只要网络通了，文件系统就可以被使用。这里存在数据安全风险和访问性能问题，但如果用户确实有此类需求，可以按照本文档描述的步骤来操作。

🔗 原理

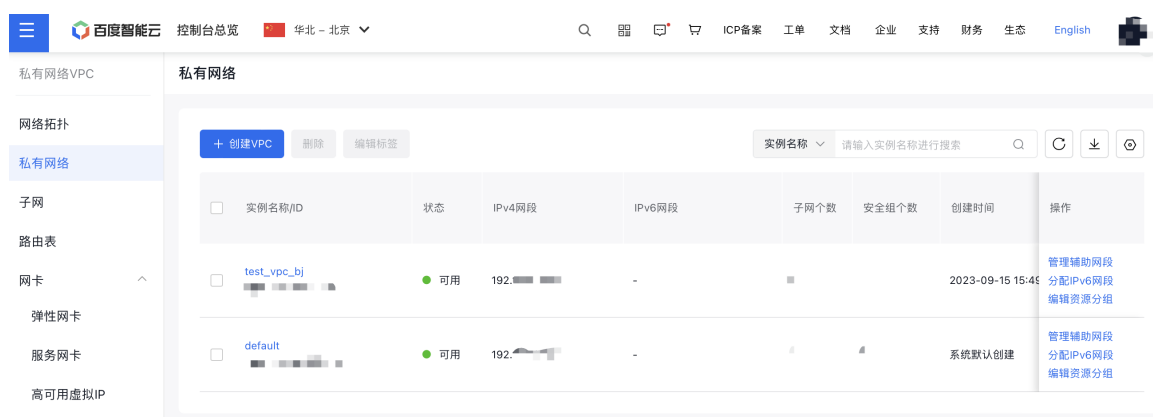
本文档所描述的方法，使用了 VPC 的“对等网络”功能，该功能的详细介绍参见 [文档1](#) 和 [文档2](#)。

🔗 跨区域使用文件系统操作步骤

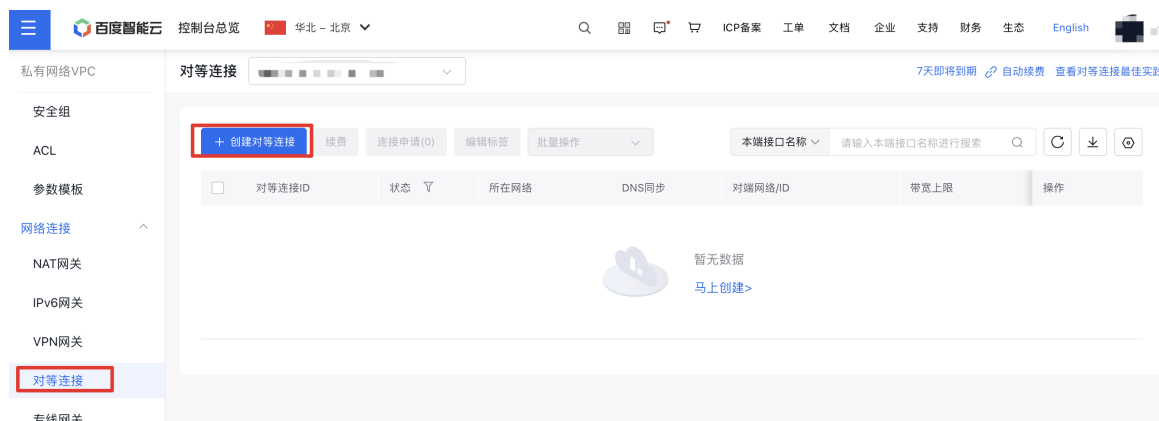
以下示例假设需要在“华北-北京”和“华南-广州”之间跨地域访问文件系统，创建的文件系统在“华南-广州”，两边需要打通的 VPC 名字分别为“test_vpc_bj”和“test_vpc_gz”。

🔗 选择“华北-北京”或“华南-广州”地域，为 VPC 创建对等连接

1. 进入 VPC 实例的管理界面，选择“test_vpc_bj”：



2. 在“对等连接”界面，点击一个“创建对等连接”：



3. 选择，连接类型为“本账号”，选择对端的地域、VPC，设置带宽上限：

华北 - 北京

当前地域: 华北 - 北京

配置信息

所在网络: test_vpc_bj (192.168.0.0/16)

本端端口名称: bj

连接类型: 本帐号 跨帐号

对端地域: 华南 - 广州

对端网络: test_vpc_gz

对端端口名称: gz

带宽上限: 1Mbps 500Mbps 1000Mbps

备注: 请输入备注, 长度0-200个字节

所选配置

当前地域: 华北 - 北京

对端地域: 华南 - 广州

带宽上限: 1000Mbps

购买配额: 1个

配置费用: 按日峰值带宽计费

下一步

这里需要注意, 当本端 VPC 和对端 VPC CIDR 重叠的时候, 虚拟机的子网段不能重合, 否则会出现网络问题。

4. 创建成功后, 点击对等连接ID即可进入详情页, 可查看本端地域与对端地域等详细信息。

百度智能云 控制台总览 华北 - 北京

私有网络VPC

网络拓扑

私有网络

子网

路由表

网卡

弹性网卡

服务网卡

对等连接

创建对等连接 续费 连接申请(0) 编辑标签 批量操作

本端接口名称 请输入本端接口名称进行搜索

对等连接ID	状态	所在网络	DNS同步	对端网络ID	带宽上限	操作
peercon	可用		未开启		10000Mbps	监控 查看详情 编辑资源分组

共 1 条 10条/页 1 跳转至 GO

百度智能云 控制台总览 华北 - 北京

返回 bj 可用

实例详情

监控

基本信息

对等连接ID: pec 状态: 可用 连接类型: 本帐号

带宽上限: 10000Mbps 带宽升级 创建时间: 2023-09-19 16:45:50 连接角色: 发起端

备注:

本端配置信息

本端地域: 华北 - 北京 本端VPC ID: vpc 本端接口名称: bj

本端接口ID: qplf-q5e2c2q7kme1

对端配置信息

对端地域: 华南 - 广州 对端VPC ID: vpc 对端账户ID: 4a37dc941

5. 为两个地域的 VPC，添加路由规则，目的地分别是对端的 VPC 网段或子网网段：

私有网络VPC

网络拓扑

私有网络

子网

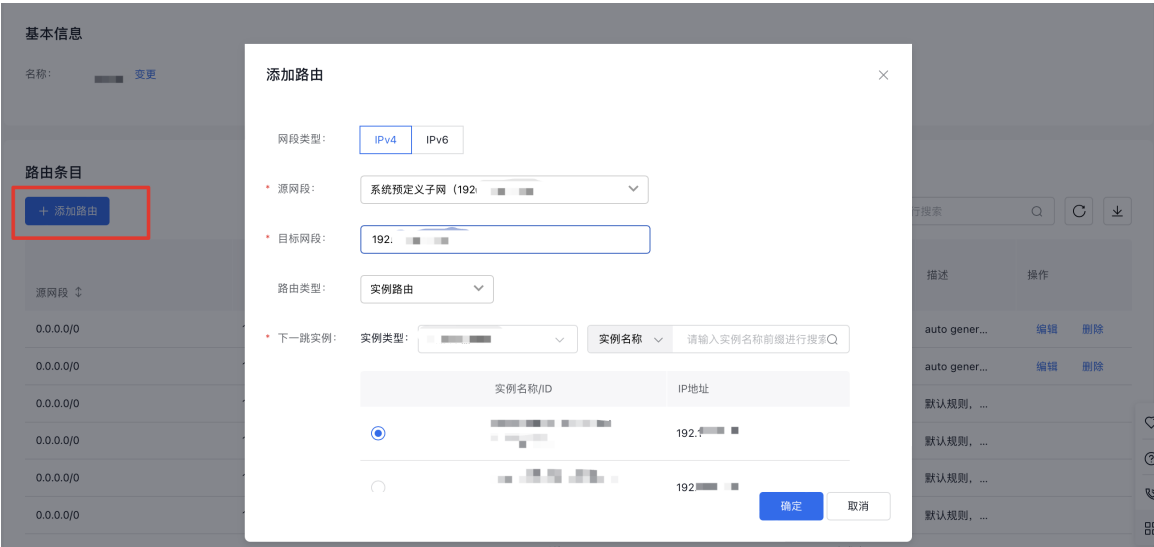
路由表

网卡

路由表

路由表名称/ID	所在网络	路由中继	操作
default	default	关	管理

共 1 条 10条/页 1 跳转至 GO



6. 在创建完对等连接后，可以在一个地域 VPC 内的虚拟机上，ping 另外一个地域 VPC 内的虚拟机内网 IP 地址，确认网络是否连通。如果不连通，请检查安全组规则：

```
[root@~]# ping 172.16.0.7
PING 172.16.0.7 (172.16.0.7) 56(84) bytes of data.
64 bytes from 172.16.0.7: icmp_seq=32 ttl=64 time=46.8 ms
64 bytes from 172.16.0.7: icmp_seq=33 ttl=64 time=46.0 ms
64 bytes from 172.16.0.7: icmp_seq=34 ttl=64 time=45.9 ms
64 bytes from 172.16.0.7: icmp_seq=35 ttl=64 time=45.9 ms
64 bytes from 172.16.0.7: icmp_seq=36 ttl=64 time=46.4 ms
```

☞ (推荐) 开启“DNS 同步”，使用挂载点地址，跨地域挂载文件系统

1. 为对等连接开启“DNS 同步”：



2. 在虚拟机上确认挂载点地址连通后，挂载文件系统：

```
[root@~]# host cfs-...cfs.gz.baidubce.com
cfs-...cfs.gz.baidubce.com has address 172.16.0.8
[root@~]# mount -t nfs4 -o minorversion=1,rsz=1048576,hard,timeo=600,retrans=2,noresvport cfs-...cfs.gz.baidubce.com:/ /mnt/
[root@~]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/vda1             41152716 1935812   37470500   5% /
devtmpfs              496932      0    496932    0% /dev
tmpfs                 507444      0    507444    0% /dev/shm
tmpfs                 507444    5128    456216   11% /run
tmpfs                 507444      0    507444    0% /sys/fs/cgroup
tmpfs                 101492      0    101492    0% /run/user/0
cfs-...cfs.gz.baidubce.com:/ 5368709120 0 5368709120  0% /mnt
```

☞ (可选) 不开启“DNS 同步”，使用挂载点 IP 地址，跨地域挂载文件系统

有可能存在某些特殊原因，用户无法为对等连接开启 DNS 同步，此时，可以使用 IP 地址直接挂载。该方法的问题在于，同一个 IP 地址，在挂载点删除之后，可能被其它新建的挂载点或服务占用，导致访问错误的文件系统或服务，请谨慎使用。

1. 在文件系统所在地域的虚拟机上，使用 host 命令解析出挂载点地址对应的 IP 地址：

```
[root@~]# host cfs-...cfs.gz.baidubce.com
cfs-...cfs.gz.baidubce.com has address 172.16.0.8
```

2. 在另外一个地域的虚拟机上，使用解析出的 IP 地址挂载文件系统：

```
[root@ ~]# mount -t nfs4 -o minorversion=1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport 172.16.0.8:/ /mnt/
[root@ ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        40G   1.9G   36G   5% /
devtmpfs         486M    0   486M    0% /dev
tmpfs            496M    0   496M    0% /dev/shm
tmpfs            496M   51M   446M   11% /run
tmpfs            496M    0   496M    0% /sys/fs/cgroup
tmpfs            100M    0   100M    0% /run/user/0
172.16.0.8:/     5.0T    0   5.0T    0% /mnt
```

跨用户使用文件系统操作步骤

跨用户使用文件系统的操作步骤和跨区域挂载基本一样，只有建立“对等连接”的步骤有区别。

在“对等连接”的创建界面，选择类型“跨账号”，填写“对端用户 ID”以及“对端网络”（即 VPC ID）。

百度智能云 控制台总览 华北 - 北京

配置信息

所在网络:

本端口名称: self

连接类型: 本帐号 跨帐号

本端连接角色: 发起端

对端账户ID:

对端网络:

对端地域:

计费方式: 按使用流量计费 按使用带宽计费

带宽上限: 1Mbps 5000Mbps 10000Mbps 10000 Mbps

备注: 请输入

创建对等连接之后，登录另外一个用户账号，在对应 VPC 的“对等连接”管理界面，会看到一个“连接申请”：

华南 - 广州

<返回实例列表

子网

安全组

ACL

路由表

NAT网关

VPN网关

对等连接

专线网关

产品服务 / 私有网络VPC-VPC实例 / 私有网络VPC-对等连接

对等连接 所在网络: 默认私有网络 (172.16.0.0/12)

+ 创建对等连接

续费

连接申请 (1)

编辑标签

更多操作

☐	对等连接ID	本端口名称/ID	状态	连接类型
--------------------------	--------	----------	----	------

接受连接申请，其余的步骤和跨地域使用一样：

连接申请列表

对等连接ID	对端区域	带宽上限	对端VPC ID	对端账户ID	本端端口名称 ?	操作 ?
peerconn-i...	华南 - 广州	1Mbps	vpc-		gzl	接受 拒绝

使用SFTP上传下载CFS文件系统数据

本文档介绍如何使用SFTP上传和下载CFS文件系统上的数据。

🔗 前提条件

在CFS文件系统所在地域，已购买[云服务器BCC](#)实例（以下介绍将以CentOS操作系统为例），详细请参见[创建BCC实例](#)。

🔗 背景信息

SFTP的传输速度依赖于BCC的外网带宽，请根据业务需求配置适当的网络带宽。

🔗 操作步骤

1. 登录云服务器BCC

2. 修改配置文件/etc/ssh/sshd_config

- 将 sshd_config 配置文件中的 Subsystem 行注释掉，新增一行 Subsystem sftp internal-sftp

```
# override default of no subsystems
#Subsystem sftp /usr/libexec/openssh/sftp-server
Subsystem sftp internal-sftp
```

- 在 sshd_config 文件末尾添加如下内容。其中 /usr/sftp 为sftp根目录，请根据实际情况修改

```
X11Forwarding no
AllowTcpForwarding no
ForceCommand internal-sftp
ChrootDirectory /usr/sftp
```

3. 设置相关权限

- 3.1 执行以下命令添加用户组。

```
groupadd sftp
```

- 3.2 执行以下命令添加用户并设置为SFTP组。

```
useradd -g sftp -s /sbin/nologin -M sftp
```

- 3.3 执行以下命令设置SFTP用户密码。

```
passwd sftp
```

- 3.4 创建SFTP用户的根目录、属主和属组，并修改权限（755）。

```
cd /usr
mkdir sftp
chown root:sftp sftp
chmod 755 sftp
```

- 在SFTP目录中，创建CFS的挂载目录，并修改权限

```
cd sftp/  
mkdir mnt  
chown sftp:sftp mnt
```

- 执行以下命令将CFS文件系统挂载到/usr/sftp/mnt 目录

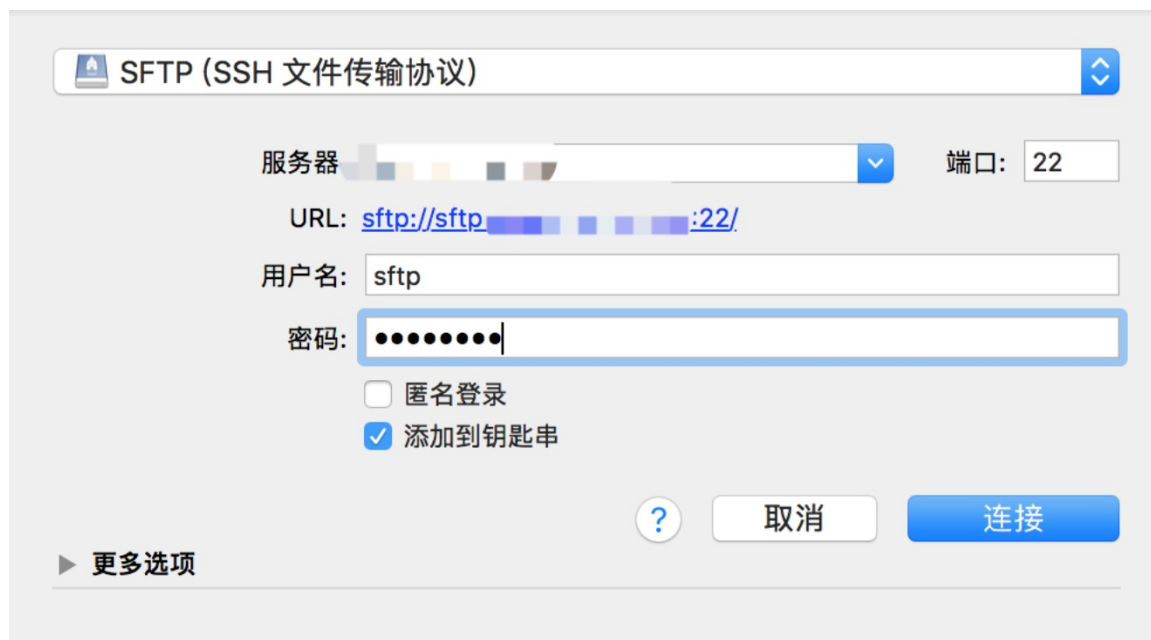
```
mount -t nfs4 -o minorversion=1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport cfs-  
q2BPKzxxxx.lb-5179xxxx.cfs.bj.baidubce.com:/ /usr/sftp/mnt
```

- 执行以下命令重启sshd服务

```
service sshd restart
```

- 登录SFTP服务，上传、下载文件

账号密码为步骤3中配置的用户和密码。此处以Cyberduck客户端为例，可根据实际情况使用支持SFTP协议的客户端进行连接。



使用rsync从旧文件系统同步到CFS文件系统

本文档提供了一种基于rsync的数据同步方案，帮助用户将存储在本地机房、其它云厂商的文件系统数据备份或迁移到百度智能云CFS中。

以下假设被同步的文件系统已经挂载在一台或者多台服务器上，路径为/old_fs。

操作步骤

- 登录云服务器

- 在云服务器BCC上绑定EIP，从而拥有公网IP。
- 确保远端服务器可以通过ssh方式登录到BCC的公网IP。

- 配置公钥

登录远端服务器，执行如下命令：

```
ssh-keygen -t rsa    #创建公钥, 会提示输入信息, 可以不用管, 一直敲回车即可
cat ~/.ssh/id_rsa.pub #查看生成的公钥, 将这里的内容拷贝, 准备输入下一步中的文件中
```

登录BCC, 编辑文件~/.ssh/id_rsa.pub, 将上一步拷贝的公钥, 输入到这里。

至此, 现在从远端服务器, 即可无需密钥登录并访问BCC。

3. 策略评估

用户需要评估以下因素, 以采用更合适的方式来同步数据:

1. 同步期间是否可以暂停对已有文件系统的写入。如果可以, 则只执行一次全量同步 (同步方案4.1)。如果不可以, 请评估同步期间的新写入数据量, 若数据量很少, 可以使用全量同步+一次增量同步 (同步方案4.2), 否则使用周期性同步方案4.3。
2. 文件系统的数据量。若已有文件系统数据量较大, 则分目录进行同步。每个子目录的同步, 是不相关的, 既可同时进行, 也可序列化执行。同时, CFS支持多个虚机同时挂载, 适当增加虚机的数量来并发执行子目录的同步可提高同步的速度。

4. 同步方案

同步过程会针对用户对旧文件系统是否有写操作而略有不同。下面分别讨论以下几种情况:

4.1 用户没有写IO请求到原有文件系统

首先确认CFS文件系统已经被mount到BCC上机器上的/mnt/cfs/或其他目录上, 接着在远端服务器上执行如下命令:

```
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_1
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_2
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_3
...
```

注释

-z: 允许压缩

-v: 冗长

-r: 递归

user: bcc的登录账号

bcc_ip: BCC的公有IP地址

4.2 用户有少量的写IO请求到原有文件系统

若用户有写IO的请求, 此时通过rsync命令, 不能保证原文件和CFS文件系统保持绝对同步。但rsync内部的算法可以保证两个文件系统是大部分同步的。用户可以在第一次rsync执行完毕后, 暂停应用程序的写操作IO到原有文件系统。具体步骤如下:

1. 用户对旧文件系统的写IO可以继续。在远端服务器上执行如下命令:

```
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_1
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_2
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_3
...
```

2. 暂停用户对旧文件系统的写IO。在远端服务器上执行如下命令:

```
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_1
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_2
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_3
...
```

第二次执行rsync操作时，由于已经做过一次同步操作，第二次会根据计算两个文件的差异，然后进行同步。因此，第二次执行rsync操作会快很多。

4.3 用户有大量的写IO请求到原有文件系统

当用户的应用程序有大量的写IO，同时又不能暂停IO的情况下。理论上无法通过同步保证旧文件系统和CFS的强一致性。此时可以考虑周期的运行rsync操作，尽可能让两个文件系统保持最大程度的一致。在Linux系统下，有多重周期执行命令的办法，这里我们推荐使用crontab，它可以方便的周期的调用rsync。例如：

```
crontab -e
```

接着将下列内容输入，其含义是每间隔1小时调动一次rsync命令。

```
* */1 * * * rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_1
```

可通过如下命令查看crontab命令是否配置成功。

```
crontab -l
```

当用户经过一段周期的从旧文件系统到百度CFS的同步后，选择一个合适时机，暂停应用程序对旧文件系统的写操作，同时删掉crontab的任务，操作如下：

```
crontab -e #根据提示，删除rsync任务并退出
```

最后再手动的执行以下rsync命令，再次确保两个文件系统完全同步。

```
rsync -zvr /old_fs/sub_folder_1/ user@bcc_ip:/mnt/cfs/sub_folder_1
rsync -zvr /old_fs/sub_folder_2/ user@bcc_ip:/mnt/cfs/sub_folder_2
rsync -zvr /old_fs/sub_folder_3/ user@bcc_ip:/mnt/cfs/sub_folder_3
...
```

此时，用户则可以从旧的服务器迁移应用程序到百度BCC服务器上。

5. 统计结果

每当执行完rsync后，会显示出这次同步的结果，其中包括：

1. 同步花费的时间；
2. 传送了多少字节；
3. IO传送的速度；
4. 一共有多少文件被同步。

使用Terraform管理CFS最佳实践

概览

百度智能云文件存储（Cloud File System）通过对标准NFS协议的支持，兼容POSIX接口，为云上的虚机、容器资源提供了跨操作系统的文件存储及共享能力。同时，百度智能云CFS提供简单、易操作的对外接口，并支持按实际使用量计费（公测期间免费），免去部署、维护费用的同时，最大化提升您的业务效率。

在目前的CFS使用过程中，开发和运维人员需要登录百度智能云的控制台，在控制台的页面内进行操作，从文完成CFS的创建、变更和释放等过程。这种做法很容易因为人为因素导致部署和管理上的错误，而且，资源不能进行批量的创建和管理、变更等

操作。

通过Terraform这种基础设施架构自动化编排工具，能够很好的解决上述问题，Terraform能够创建配置文件的模板，以可重复、可预测的方式定义和预配百度智能云上的资源，减少人为因素导致的部署和管理错误。能够多次部署同一模板，创建相同的开发、测试和生产环境。

需求场景

面向企业IT人员（提升云上资源的管理维护效率）

- 可以用代码来管理维护您的CFS资源。允许保存CFS的状态，从而使您能够跟踪对CFS的配置所做的更改，并与其他人共享这些配置。
- 您通过按需创建开发和部署环境来降低成本。并且，您可以在CFS配置更改之前进行评估。
- 大大降低企业在基础设施构建过程中的成本，同时也可以提高云基础设施开发人员在 IAC(基础设施即代码) 方面的技能，简化云上资源运维

配置前您需要了解

CFS产品有如下配额限制：

限制项	限额
单用户可创建文件系统数	20个
单个文件系统容量	100PB
单个文件系统文件数	10亿个
单个文件大小	1PB
目录深度	1000级
软链接路径长度	4080字节
文件名长度	255字节
同时连接的客户端数量	数百个

配置步骤

环境准备

Terraform：目前百度智能云已与Terraform集成，您可通过Terraform来创建、管理CFS、VPC、BCC等多种百度智能云资源，具体请参考[Terraform BaiduCloud Provider](#)。

初始化Terraform

1. 登录[Terraform](#)官网下载适用于您的操作系统的程序包。
2. 配置terraform baidu provider信息，配置文件可参考如下的代码，在当前文件夹内保存为version.tf。

```
terraform {
  required_providers {
    baiducld = {
      source = "baidubce/baiducld"
      version = "1.18.0" #版本号 从1.18.0版本开始支持文件存储CFS
    }
  }
}
```

3. 执行命令terraform init，Terraform会根据当前文件夹内的version.tf文件初始化工作空间。

4. 编辑Terraform配置文件。新建文件main.tf，在文件中指定VPC、CFS、CFS挂载点等元信息。请参考后续步骤中的配置示例。

更多CFS参数信息可参考[baidu provider官方文档](#)

🔗 CFS创建配置示例

```
resource "baiducloud_vpc" "vpc" {
  name = "terraform_vpc"
  cidr = "172.16.0.0/16"
}

resource "baiducloud_subnet" "subnet" {
  name       = "terraform_subnet-c"
  zone_name = "cn-bj-c"
  cidr       = "172.16.128.0/24"
  vpc_id     = baiducloud_vpc.vpc.id
  description = "terraform test subnet"
}

resource "baiducloud_cfs" "default" {
  # CFS名称
  name = "terraform_test"
  # 可用区
  zone = "zoneD"
  # 协议类型 默认为nfs
  protocol = "nfs"
  # 存储类型 默认为cap
  type = "cap"
}
```

🔗 挂载点创建配置示例

```
resource "baiducloud_cfs_mount_target" "mount1" {
  fs_id = baiducloud_cfs.default.id
  subnet_id = baiducloud_subnet.subnet.id
  vpc_id = baiducloud_vpc.vpc.id
}
```

🔗 应用配置

- 执行如下命令，可查看资源变更计划

```
$ terraform plan
```

- 执行如下命令，可应用资源变更计划，完成资源的创建或更新，如图所示

```
$ terraform apply
```

```
baiducloud_cfs.default: Creating...
baiducloud_vpc.vpc: Creating...
baiducloud_vpc.vpc: Creation complete after 2s [id=vp-██████████]
baiducloud_subnet.subnet: Creating...
baiducloud_subnet.subnet: Creation complete after 1s [id=sn-██████████]
baiducloud_cfs.default: Still creating... [10s elapsed]
baiducloud_cfs.default: Creation complete after 13s [id=fs-██████████]
data.baiducloud_cfss.default: Reading...
baiducloud_cfs_mount_target.mount1: Creating...
data.baiducloud_cfss.default: Read complete after 0s [id=terraform-20:██████████1000000001]
baiducloud_cfs_mount_target.mount1: Still creating... [10s elapsed]
baiducloud_cfs_mount_target.mount1: Still creating... [20s elapsed]
baiducloud_cfs_mount_target.mount1: Creation complete after 23s [id=65-██████████77783032754961654f4a5235673d3d]
data.baiducloud_cfs_mount_targets.default: Reading...
data.baiducloud_cfs_mount_targets.default: Read complete after 0s [id=terraform-20:██████████938491687000000002]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
```

🔗 CFS管理

🔗 更改CFS名称

如要对CFS进行名称的修改，只需要修改name字段的值，然后执行步骤7即可

🔗 查询CFS信息

```
data "baiducloud_cfss" "default" {
  # 可通过filter进行过滤 如下为过滤fs_id=fsxxxx的CFS
  filter{
    name = "fs_id"
    values = ["fsxxxx"]
  }
}
```

🔗 资源删除

- 若需要删除挂载点，只需要删除baiducloud_cfs_mount_target这一资源即可，可通过如下命令

```
# 通过-target指定资源进行删除
terraform destroy -target baiducloud_cfs_mount_target.mount1
```

注意：再删除挂载点之前要先进行卸载操作

- 若要删除CFS，只需删除baiducloud_cfs这一资源即可，可通过如下命令

```
# 通过-target指定资源进行删除
terraform destroy -target baiducloud_cfs.default
```

- 可通过如下命令释放当前main.tf中的所有资源

```
terraform destroy
```

🔗 测试验证

1. 在百度智能云控制台的产品界面查看对应的资源是否都创建成功。
2. 在完成了上述步骤的配置之后，可按照这篇文档中的步骤去使用CFS，查看是否能够正确工作。 [文档链接](#)

🔗 相关产品

[文件存储CFS](#)、[私有网络VPC](#)

性能测试方法

Linux系统性能测试方法

本文介绍如何在Linux系统中使用FIO工具对CFS进行数据读写（不涉及元数据）IOPS、吞吐和时延的性能测试。

前提条件

1. 您已创建待测试的CFS文件系统，并将CFS文件系统挂载到BCC云服务器上。
 - 创建CFS文件系统请见：[创建文件系统](#)。
 - Linux系统挂载NFS协议CFS文件系统请见：[Linux系统挂载与卸载NFS协议CFS](#)。
2. CFS所提供的性能规格，除时延参数外，均需要一定规模的并发压测才能达到最大值。建议压测时准备32C以上的BCC云服务器作为客户端，可满足大部分场景的压测需求。其他压测需求，可根据实际情况配置对应的BCC云服务器。
3. 进行性能压测时，尤其是时延测试，需要保证BCC云服务器（客户端）和 CFS 处于同可用区。跨可用区测试的性能结果会和标准值有较大差异，不具备参考价值，应尽量避免。

背景信息

推荐使用 FIO 工具测试CFS，不同工具测试的CFS性能会有差异，难以反映真实性能。本文示例的性能参数，均为Linux系统下采用FIO工具的测试结果，以此作为CFS产品性能指标参考。本次具体的性能指标说明如下：

- IOPS：每秒读写数据块的数量，单位 个/s。建议在数据块较小（512bytes/4k/8k/16k）场景下进行 IOPS 的基准测试；
- 吞吐：每秒读写数据量的大小，单位 GiB/s 或 MiB/s。建议在数据块较大（256k/512k/1M）场景下进行吞吐的基准测试；
- 时延：处理读写请求的耗时，单位 ms。主要针对单线程以及多线程并发读写时的IO时延进行基准测试；针对平均时延、最大时延以及99.9%分位时延（即99.0%的时延小于该值）进行基准测试。

测试步骤

1. 远程登录[BCC云服务器](#)（客户端）。
2. 安装FIO测试工具。

操作系统	安装命令
CentOS	sudo yum install fio
RHEL	
Ubuntu	## 依次执行以下安装命令： sudo apt-get update sudo apt-get install fio
Debian	

3. 在单台Linux BCC中执行以下命令测试性能。

测试性能	测试命令参考
IOPS	## 128并发下随机读写IOPS fio -direct=1 -ioengine=libaio -rw=randread -bs=4k -size=1G -runtime=60 -time_based=1 -group_reportin fio -direct=1 -ioengine=libaio -rw=randwrite -bs=4k -size=1G -runtime=60 -time_based=1 -group_report
吞吐	## 8并发下顺序读写吞吐。吞吐性能受限于BCC带宽，为了测试获得最准确吞吐性能，建议根据客户端配置情况合理配置 --numjobs fio -direct=1 -ioengine=libaio -rw=read -bs=1024k -size=1G -runtime=60 -time_based=1 -group_reportin fio -direct=1 -ioengine=libaio -rw=write -bs=1024k -size=1G -runtime=60 -time_based=1 -group_reporti
时延	## 单线程下读写时延 fio -direct=1 -ioengine=libaio -rw=randread -bs=4k -size=1G -runtime=60 -time_based=1 -group_reporti fio -direct=1 -ioengine=libaio -rw=randwrite -bs=4k -size=1G -runtime=60 -time_based=1 -group_report

说明：如需要进行多线程测试，建议您根据压测时客户端配置情况合理设置--numjobs参数。

附录 - FIO参数说明

测试命令中有关FIO各参数含义的说明如下表所示。

参数	说明
direct	表示是否使用direct I/O。默认值：1。 - 值为1：表示使用direct I/O，忽略I/O缓存，数据直写。 - 值为0：表示不使用direct I/O。
ioengine	表示测试时FIO选择哪种I/O引擎，通常选择libaio，更符合日常应用模式，更多的选择请查阅FIO官方文档。
rw	表示测试时的读写策略。您可以设置为： - randwrite：随机写。 - randread：随机读。 - read：顺序读。 - write：顺序写。 - randrw：混合随机读写。
bs	表示I/O单元的块大小（block size）。默认值：4 KiB。读取和写入的值可以以read、write格式单独指定，其中任何一个都可以为空以将该值保留为其默认值。
size	表示测试文件大小。 FIO会将指定的文件大小全部读/写完成，然后才停止测试，除非受到其他选项（例如运行时）的限制。如果未指定该参数，FIO将使用给定文件或设备的完整大小。也可以将大小作为1到100之间的百分比给出。例如指定size=20%，FIO将使用给定文件或设备完整大小的20%空间。
runtime	表示测试时间，即FIO运行时长。 如果未指定该参数，则FIO会持续将上述size指定大小的文件，以每次bs值为块大小读/写完。
time_based	如果指定了运行时间，即使文件已完全读写，也会按指定的持续时间运行。相同的工作负载将根据运行时间重复执行。
group_reporting	表示测试结果显示模式。如果指定该参数，测试结果会汇总每个进程的统计信息，而不是以不同任务来统计信息。
directory	表示待测试的文件系统挂载路径。例如/mnt/test_perf，请正确替换。
name	表示测试任务名称，可以随意设定。
numjobs	表示测试的并发线程数。

有关各参数的更多说明，请参见[FIO官方文档](#)。

Windows系统性能测试方法

本文介绍如何在Windows系统中使用FIO工具对CFS进行数据读写（不涉及元数据）IOPS、吞吐和时延的性能测试。

前提条件

1. 您已创建待测试的CFS文件系统，并将CFS文件系统挂载到BCC云服务器上。
 - 创建CFS文件系统请见：[创建文件系统](#)。
 - Windows系统挂载SMB协议CFS文件系统请见：[Windows系统挂载与卸载SMB协议CFS](#)。

2. CFS所提供的性能规格，除时延参数外，均需要一定规模的并发压测才能达到最大值。建议压测时准备32C以上的BCC云服务器作为客户端，可满足大部分场景的压测需求。其他压测需求，可根据实际情况配置对应的BCC云服务器。
3. 进行性能压测时，尤其是时延测试，需要保证BCC云服务器（客户端）和 CFS 处于同可用区。跨可用区测试的性能结果会和标准值有较大差异，不具备参考价值，应尽量避免。

🔗 背景信息

推荐使用 FIO 工具测试CFS，不同工具测试的CFS性能会有差异，难以反映真实性能。本文示例的性能参数，均为Windows系统下采用FIO工具的测试结果，以此作为CFS产品性能指标参考。本次具体的性能指标说明如下：

- IOPS：每秒读写数据块的数量，单位 个/s。建议在数据块较小（512bytes/4k/8k/16k）场景下进行 IOPS 的基准测试；
- 吞吐：每秒读写数据量的大小，单位 GiB/s 或 MiB/s。建议在数据块较大（256k/512k/1M）场景下进行吞吐的基准测试；
- 时延：处理读写请求的耗时，单位 ms。主要针对单线程以及多线程并发读写时的IO时延进行基准测试；针对平均时延、最大时延以及99.9%分位时延（即99.0%的时延小于该值）进行基准测试。

🔗 测试步骤

1. 远程登录[BCC云服务器](#)（客户端）。
2. 安装FIO测试工具，您可以下载msi格式的安装包进行安装。FIO [安装包](#)下载地址，请参见 [fio官网](#)。
3. 将CFS挂载到Z盘，FIO可执行文件安装在 `C:\Program Files\fio\fio.exe` 为例，在单台Windows BCC中执行以下命令测试性能。
4. 在单台Linux BCC中执行以下命令测试性能。

测试性能	测试命令参考
IOPS	<pre>## 128并发下随机读写IOPS ### 导航到FIO安装目录下 cd "C:\Program Files\fio" ### 运行fio命令 fio.exe -direct=1 -ioengine=windowsaio -rw=randread -bs=4k -size=1G -runtime=60 -time_based=1 -group fio.exe -direct=1 -ioengine=windowsaio -rw=randwrite -bs=4k -size=1G -runtime=60 -time_based=1 -grou</pre>
吞吐	<pre>## 8并发下顺序读写吞吐。吞吐性能受限于BCC带宽，为了测试获得最准确吞吐性能，建议根据客户端配置情况合理配置 --numjobs ### 导航到FIO安装目录下 cd "C:\Program Files\fio" ### 运行fio命令 fio.exe -direct=1 -ioengine=windowsaio -rw=read -bs=1024k -size=1G -runtime=60 -time_based=1 -group_ fio.exe -direct=1 -ioengine=windowsaio -rw=write -bs=1024k -size=1G -runtime=60 -time_based=1 -group</pre>
时延	<pre>## 单线程下读写时延 ### 导航到FIO安装目录下 cd "C:\Program Files\fio" ### 运行fio命令 fio.exe -direct=1 -ioengine=windowsaio -rw=randread -bs=4k -size=1G -runtime=60 -time_based=1 -group fio.exe -direct=1 -ioengine=windowsaio -rw=randwrite -bs=4k -size=1G -runtime=60 -time_based=1 -grou</pre>

说明：如需要进行多线程测试，建议您根据压测时客户端配置情况合理设置--numjobs参数。

附录 - FIO参数说明

测试命令中有关FIO各参数含义的说明如下表所示。

参数	说明
direct	表示是否使用direct I/O。默认值：1。 - 值为1：表示使用direct I/O，忽略I/O缓存，数据直写。 - 值为0：表示不使用direct I/O。
ioengine	表示测试时FIO选择哪种I/O引擎，通常选择libaio，更符合日常应用模式，更多的选择请查阅FIO官方文档。
rw	表示测试时的读写策略。您可以设置为： - randwrite：随机写。 - randread：随机读。 - read：顺序读。 - write：顺序写。 - randrw：混合随机读写。
bs	表示I/O单元的块大小（block size）。默认值：4 KiB。读取和写入的值可以以read、write格式单独指定，其中任何一个都可以为空以将该值保留为其默认值。
size	表示测试文件大小。 FIO会将指定的文件大小全部读/写完成，然后才停止测试，除非受到其他选项（例如运行时）的限制。如果未指定该参数，FIO将使用给定文件或设备的完整大小。也可以将大小作为1到100之间的百分比给出。例如指定size=20%，FIO将使用给定文件或设备完整大小的20%空间。
runtime	表示测试时间，即FIO运行时长。 如果未指定该参数，则FIO会持续将上述size指定大小的文件，以每次bs值为块大小读/写完。
time_based	如果指定了运行时间，即使文件已完全读写，也会按指定的持续时间运行。相同的工作负载将根据运行时间重复执行。
group_reporting	表示测试结果显示模式。如果指定该参数，测试结果会汇总每个进程的统计信息，而不是以不同任务来统计信息。
directory	表示待测试的文件系统挂载路径。例如/mnt/test_perf，请正确替换。
name	表示测试任务名称，可以随意设定。
numjobs	表示测试的并发线程数。

有关各参数的更多说明，请参见[FIO官方文档](#)。

API参考

API功能更新记录

发布时间	接口	更新介绍
2025-03-31	查询文件系统	支持批量查询、新增KMSKeyId（加密密钥Id）、createTime（创建时间）字段信息
2025-03-31	查询挂载客户端	新增查询文件系统挂载客户端接口
2024-12-20	查询文件系统 、 查询挂载点	新增vpclid（挂载点所在VPCID）字段信息
2024-10-09	创建文件系统 、 查询文件系统	新增zone（可用区）字段信息
2024-07-16	创建文件系统 、 更新文件系统 、 查询文件系统	新增capacityQuota（容量上限）字段信息
2024-07-05	更新文件系统标签	新增更新文件系统标签接口
2024-07-05	创建文件系统 、 查询文件系统	新增Tags（标签）字段信息

API概述

欢迎使用百度智能云文件存储CFS，您可以使用API对CFS做灵活的操作。如果您对CFS还不太了解，请参考[入门概述](#)。

如果您是初次调用百度智能云产品的API，可以观看[API入门视频指南](#)，快速掌握调用API的方法。

接口概览

CFS API提供如下接口类型：

接口类型	描述
文件系统接口	提供创建文件系统、更新文件系统、查询文件系统、查询挂载客户端、释放文件系统和更新文件系统标签等接口。
挂载点接口	提供创建挂载点、查询挂载点和删除挂载点等接口。

通用说明

API调用遵循HTTP协议，各Region采用不同的域名，具体域名为cfs.{region}.baidubce.com。数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。URL参数中所使用的fsId为文件系统唯一资源标识。

实名认证

使用CFS API的用户需要实名认证，没有通过实名认证的可以前往[百度智能云官网控制台](#)中的安全认证下的实名认证中进行认证。

百度智能云提供[个人认证](#)、[企业认证](#)两种认证方式，您可以根据实际情况选择一种进行认证。

API认证机制

所有API的安全认证一律采用Access Key与请求签名机制。Access Key由Access Key ID和Secret Access Key组成，均为字符串。对于每个HTTP请求，使用下面所描述的算法生成一个认证字符串。提交认证字符串放在Authorization头域里。服务端根据生成算法验证认证字符串的正确性。认证字符串的格式为bce-auth-v{version}/{accessKeyId}/{timestamp}/{expirationPeriodInSeconds}/{signedHeaders}/{signature}。

- version是正整数。
- timestamp是生成签名时的UTC时间。

- expirationPeriodInSeconds表示签名有效期限。
- signedHeaders是签名算法中涉及到的头域列表。头域名之间用分号（;）分隔，如host;x-bce-date。列表按照字典序排列。（本API签名仅使用host和x-bce-date两个header）
- signature是256位签名的十六进制表示，由64个小写字母组成。

当百度智能云接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。

鉴权认证机制的详细内容请参见[鉴权认证机制](#)。

🔗 通信协议

CFS API支持HTTP和HTTPS两种调用方式。为了提升数据的安全性，建议通过HTTPS调用。

🔗 请求结构说明

请求结构中包括的组成说明：

参数类型	说明
URI	通常用于指明操作实体,如:PUT /v1/cfs/{fsId}
Query String	URL中携带的请求参数
Header	通过HTTP头域传入,如x-bce-date
Request Body	通过JSON格式组织的请求数据体

🔗 响应结构说明

响应值分为两种形式：

返回内容	说明
HTTP Status Code	如200,400,403,404等
Response Body	JSON格式组织的响应数据体

🔗 公共请求头

下表列出了所有CFS API所携带的公共头域。HTTP协议的标准头域不在此处列出

头域（HEADER）	是否必须	说明
Authorization	是	签名字符串，生成签名字符串的方法请参考鉴权认证机制
Content-Type	是	application/json; charset=utf-8
x-bce-date	可选	签名日期

🔗 公共响应头

下表列出了所有CFS API的公共响应头域。HTTP协议的标准响应头域不在此处列出

头域（HEADER）	说明
Content-Type	application/json; charset=utf-8
x-bce-request-id	本次请求的requestId

🔗 幂等性

当调用某些接口时如果遇到了请求超时或服务器内部错误，用户可能会尝试重发请求，这时用户通过clientToken参数避免创建出比预期要多的资源，即保证请求的幂等性。

幂等性基于clientToken，clientToken是一个长度不超过64位的ASCII字符串，通常放在query string里，如http://bcc.bj.baidubce.com/v1/instance?clientToken=be31b98c-5e41-4838-9830-9be700de5a20。

如果用户使用同一个clientToken值调用创建接口，则服务端会返回相同的请求结果。因此用户在遇到错误进行重试的时候，可以通过提供相同的clientToken值，来确保只创建一个资源；如果用户提供了一个已经使用过的clientToken，但其他请求参数（包括queryString和requestBody）不同甚至url Path不同，则会返回IdempotentParameterMismatch的错误代码。

clientToken的有效期为24小时，以服务端最后一次收到该clientToken为准。也就是说，如果客户端不断发送同一个clientToken，那么该clientToken将长期有效。

🔗 日期与时间规范

日期与时间的表示有多种方式。为统一起见，除非是约定俗成或者有相应规范的，凡需要日期时间表示的地方一律采用UTC时间，遵循ISO 8601，并做以下约束：

- 1. 表示日期一律采用YYYY-MM-DD方式，例如2014-06-01表示2014年6月1日。
- 2. 表示时间一律采用hh:mm:ss方式，并在最后加一个大写字母Z表示UTC时间。例如23:00:10Z表示UTC时间23点0分10秒。
- 3. 凡涉及日期和时间合并表示时，在两者中间加大写字母T，例如2014-06-01T23:00:10Z表示UTC时间2014年6月1日23点0分10秒。

服务域名

CFS API的服务域名为：

Region	Endpoint	Protocol
北京（bj）	cfs.bj.baidubce.com	HTTP and HTTPS
保定（bd）	cfs.bd.baidubce.com	HTTP and HTTPS
成都（cd）	cfs.cd.baidubce.com	HTTP and HTTPS
广州（gz）	cfs.gz.baidubce.com	HTTP and HTTPS
苏州（su）	cfs.su.baidubce.com	HTTP and HTTPS
武汉（fwh）	cfs.fwh.baidubce.com	HTTP and HTTPS
香港（hkg）	cfs.hkg.baidubce.com	HTTP and HTTPS

说明：CFS API支持HTTP和HTTPS两种调用方式。为了提升数据的安全性，建议通过HTTPS调用。

错误码

🔗 错误码格式

当用户访问API出现错误时，会返回给用户相应的错误码和错误信息，便于定位问题，并做出适当的处理。请求发生错误时通过Response Body返回详细错误信息，遵循如下格式：

参数名	类型	说明
code	String	错误码
message	String	错误描述
requestId	String	本次请求的requestId

例如：

```
{
  "requestId" : "ae2225f7-1c2e-427a-a1ad-5413b762957d",
  "code" : "NoSuchKey",
  "message" : "The resource you requested does not exist"
}
```

公共错误码

下表列出了百度智能云API的公共错误码。

Code	Message	HTTP Status Code	说明
AccessDenied	Access denied.	403 Forbidden	无权限访问对应的资源。
InappropriateJSON	The JSON you provided was well-formed and valid, but not appropriate for this operation.	400 Bad Request	请求中的JSON格式正确，但语义上不符合要求。如缺少某个必需项，或者值类型不匹配等。出于兼容性考虑，对于所有无法识别的项应直接忽略，不应该返回这个错误。
InternalServerError	We encountered an internal error. Please try again.	500 Internal Server Error	所有未定义的其他错误。在有明确对应的其他类型的错误时（包括通用的和服务自定义的）不应该使用。
InvalidAccessKeyId	The Access Key ID you provided does not exist in our records.	403 Forbidden	Access Key ID不存在。
InvalidHTTPAuthHeader	The HTTP authorization header is invalid. Consult the service documentation for details.	400 Bad Request	Authorization头域格式错误。
InvalidHTTPRequest	There was an error in the body of your HTTP request.	400 Bad Request	HTTP body格式错误，例如不符合指定的Encoding等。
InvalidURI	Could not parse the specified URI.	400 Bad Request	URI形式不正确。例如一些服务定义的关键词不匹配等。对于ID不匹配等问题，应定义更加具体的错误码，例如NoSuchKey。
MalformedJSON	The JSON you provided was not well-formed.	400 Bad Request	JSON格式不合法。
InvalidVersion	The API version specified was invalid.	404 Not Found	URI的版本号不合法。
OptInRe		403	

quired	A subscription for the service is required.	Forbidden	没有开通对应的服务。
PreconditionFailed	The specified If-Match header doesn't match the ETag header.	412 Precondition Failed	详见ETag。
Request Expired	Request has expired. Timestamp date is XXX.	400 Bad Request	请求超时。XXX要改成x-bce-date的值。如果请求中只有Date，则需要将Date转换为datetime。
IdempotentParameterMismatch	The request uses the same client token as a previous, but non-identical request.	403 Forbidden	clientToken对应的API参数不一样。
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your Secret Access Key and signing method. Consult the service documentation for details.	400 Bad Request	Authorization头域中附带的签名和服务端验证不一致。

🔗 CFS错误码

下表列出了CFS API特有错误码。

错误码	错误描述	HTTP状态码	语义
InstanceNotFound	The specified CFS instance does not exist.	404 Not Found	指定的CFS实例不存在
InvalidFSType	The FS TYPE is invalid.	400 Bad Request	文件系统的类型错误
InvalidFSName	The FS NAME is invalid.	400 Bad Request	文件系统的名称不符合要求
InvalidFSProtocol	The FS PROTOCOL is invalid.	400 Bad Request	文件系统不支持该协议
AvailabilityZoneNotFound	The specified availability zone does not exist.	400 Bad Request	用户设置的分区（zone）不存在
InvalidVpcId	The vpc id is invalid.	400 Bad Request	vpc id 错误
CreateOrderFail	We encountered an internal error. Please try again.	500 Internal Server Error	创建订单失败，请稍候重试
DuplicateRequest	The request is duplicate.	400 Bad Request	重复的创建文件系统实例请求
FsNumQuotaExceeded	The number of CFS exceeded current quota.	413 Request Entity Too Large	CFS数量超过用户配额限制
DropCreatingFS	The fs is in creating, can not drop.	400 Bad Request	不能删除正在创建的文件系统实例
DropMountedFS	The fs has mount target , please drop before clear mount target.	400 Bad Request	不能删除有挂载点的文件系统实例
FSNotInService	The fs is not in service, not allow this operation.	400 Bad Request	文件系统服务不可用
InvalidSubnetId	The subnet id is invalid.	400 Bad Request	用户提供的 subnet id 不可用
InvalidVpcOrSubnet	Invalid VPC ID or Subnet ID.	400 Bad Request	创建挂载点时，提供的 vpc id 或者 subnet id 不正确
AlreadyMounted	The fs already has mount target in the same subnet	400 Bad Request	文件系统已经被挂载到目标子网

	说明	错误码	说明
DropCreatingMount	The mount target is in creating, can not drop.	400 Bad Request	不能删除正在创建的挂载点
MountTargetNotFound	The request mount target id is not found.	404 Not Found	没有找到目标挂载点
MissingParameter	A required parameter 'parameterName' is not supplied.	400 Bad Request	该请求缺少必要的参数“parameterName”
InvalidParameter	The parameter 'parameterName' is invalid.	400 Bad Request	"parameterName"参数不合法
ResourceNotExists	The resource does not exist.	404 Not Found	请求的host未在网关对应的region中配置
EOFException	Unexpected EOF read on the socket.	400 Bad Request	读取request 中inputstream异常，通常为eof
RequestTimeTooSkewed	The difference between the request time and the server's time is too large.	403 Forbidden	Authorization中timestamp无效
ServiceError	the dependency service is unavailable, host is {host}	424 Failed Dependency	服务处于熔断状态
UnsupportedProtocol	The protocol '{http}' is not supported.	403 Forbidden	协议校验不支持http或者https请求
QualifyNotPass	The User has not pass qualify.	403 Forbidden	未通过实名认证
RateLimit	There are too many connections. The host is {host}	421 Misdirected Request	请求被限速
IllegalRequestUri	The request URI contains some forbidden characters.	400 Bad Request	url合法性校验失败
RequestUriForbidden	The request URI is forbidden or not included.	400 Bad Request	请求的uri不在服务允许的uri
RequestEntityTooLargeException	Request entity is too large, should not exceed 50M.	413 Request Entity Too Large	multipart/ 类型的请求，文件大小大于50M
RequestEntityEmptyException	Request entity is empty.	400 Bad Request	multipart/ 类型的请求,文件字节流为0
ParseBodyException	Parsing request body exception	500 Internal Server Error	如果有映射文件，在映射body时异常
ServiceUnavailable	Service or dependent service is unavailable.	503 Service Unavailable	与后端请求连接异常或超时等网络错误
InvalidClientToken	The client token is invalid, client token must be a ascii string of no longer than 64 bits.	400 Bad Request	幂等请求的clientToken的值错误
MultiRequestConflict	The request is refused because of a previous is in progress.	409 Conflict	幂等请求时，前一个请求正在处理中，又收到其它请求

文件系统相关接口

创建文件系统

描述

- 创建一个cfs文件系统，返回分配的文件系统ID。
- 付费方式为后付费。

请求

请求结构

```
POST /v{version}/cfs HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string

{
  "fsName": name,
  "zone": zonB,
  "type": type,
  "protocol": protocol
  "tags": [
    {"tagKey": "aaa", "tagValue": "bbb"}
  ]
  "capacityQuota": 1075838976
}
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1。
fsName	String	是	RequestBody参数	FileSystem的名称，方便记忆。长度1~65个字节，字母开头，可包含字母数字_/.字符。
zone	String	是	RequestBody参数	FileSystem所在可用区，例如zoneB。
type	String	否	RequestBody参数	文件系统类型：1.cap(容量型) 2.ssd(性能型) 当前默认是容量型
protocol	String	否	RequestBody参数	协议类型：1.nfs 2.smb，默认nfs协议。
tags	List<Tag>	否	RequestBody参数	文件系统标签
capacityQuota	uint64	否	RequestBody参数	文件系统的quota值，不选或者值为0是表示无自定义硬限，则表示上限容量（设置时最小为1GB，最大为10PB），单位 bytes。

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数

参数名称	类型	描述
fsId	String	FileSystem的ID。后续针对该实例的操作，均需要在请求中带上此ID。

示例

请求：

```
POST /v1/cfs HTTP/1.1
HOST cfs.bj.baidubce.com
Authorization bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
{
  "fsName": "cfs-for-test"
}
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS
{
  "fsId": "cfs-jVDkWrBBoj"
}
```

更新文件系统

描述

更新一个filesystem的配置信息

请求

请求结构

```
PUT /v{version}/cfs/{fsId} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string
{
  "fsName": name
  "capacityQuota": 1075838976
}
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1。
fsId	String	是	URL参数	待更新的FileSystem的ID。
fsName	String	否	RequestBody参数	FileSystem的名称，方便记忆。长度1~65个字节，字母开头，可包含字母数字_/.字符。
capacityQuota	uint64	否	RequestBody参数	更新之后的quota值，不选或者值为0是表示无自定义硬限，否则表示上限容量（设置时最小为1GB，最大为10PB），单位 bytes。

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数：无

示例

请求：

```
PUT /v1/cfs/cfs-jVDkWrBBoj HTTP/1.1
HOST cfs.bj.baidubce.com
Authorization bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
{
  "fsName": "cfs-for-test"
}
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS
```

查询文件系统

描述

- 查询用户账户下的所有FileSystem信息
- 支持按fsId、userId，匹配规则支持部分包含
- 返回结果是多重查询条件的交集
- 结果支持marker分页，分页大小默认为1000，可通过maxKeys参数指定

请求

请求结构

```
GET /v{version}/cfs?fsId={fsId}&userId={userId}&marker={marker}&maxKeys={maxKeys}&filterTag={filterTag} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
userId	String	是	Query参数	要查询文件系统所属的用户ID
fsId	String	否	Query参数	要查询的FileSystem ID，支持批量查询，格式为fsId=fsIdA,fsIdB,fsIdC, 批量查询数量上限为100个。
marker	String	否	Query参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串
maxKeys	int	否	Query参数	每页包含的最大数量，最大数量不超过1000。缺省值为1000
filterTag	String	否	Query参数	{key}:{value}格式 查询key下所有值为 {key}:@@@

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数

参数名称	类型	描述
fileSystemList	List< FileSystemModel >	包含查询结果的列表
marker	String	标记查询的起始位置，若结果列表为空，此项不存在
isTruncated	boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现
maxKeys	int	每页结果包含的数量

示例

请求：

```
GET /v1/cfs?fsId=cfs-jVDkWrBBoj&maxKeys=2 HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host:x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
```

响应

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS

{
  "isTruncated": false,
  "maxKeys": 1,
  "fileSystemList": [
    {
      "fsId": "cfs-jVDkWrBBoj",
      "fsName": "test-cfs",
      "vpId": "",
      "type": "ssd",
      "protocol": "nfs",
      "fsUsage": "O",
      "zone": "zoneB",
      "status": "available",
      "KMSKeyId": "xxxxxxx",
      "createTime": "2023-11-29T11:19:43Z",
      "capacityQuota": 0,
      "mountTargetList": [
        {
          "accessGroupName": "DEFAULT_ACCESS_GROUP_NAME",
          "mountId": "664d336f357447706c794d48304f5a306a5369317a413d3d",
          "ovip": "192.168.5.242",
          "domain": "cfs-Y1uZ26QOoh.lb-f02bb5da.cfs.bce.sandbox.baidu.com",
          "subnetId": "sbn-2ge2ucwhf8ut",
          "vpId": "vpc-vjf1mdpq9pkf"
        }
      ],
      "tags": [
        {
          "tagKey": "默认项目",
          "tagValue": "123"
        }
      ]
    }
  ]
}
```

查询挂载客户端

描述

查询指定filesystem下挂载的客户端信息。

请求

请求结构

```
GET /v{version}/cfs/clients?fsId={fsId}&marker={marker}&maxKeys={maxKeys} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
fsId	String	是	Query参数	要查询的FileSystem ID
marker	String	否	Query参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串
maxKeys	Integer	否	Query参数	每页包含的最大数量，最大数量不超过1000。缺省值为1000

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数

参数名称	类型	描述
clientInfoList	List< ClientInfoModel >	包含查询结果的列表
marker	String	标记查询的起始位置
isTruncated	boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值
maxKeys	int	每页结果包含的数量

示例

请求：

```
GET /v1/cfs/clients?fsId=cfs-j6zB***Ovg HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS

{
  "isTruncated": false,
  "marker": "",
  "nextMarker": "",
  "maxKeys": 2,
  "clientInfoList": [
    {
      "zone": "zoneB",
      "vpclId": "vpc-vjf***pq9pkf",
      "mountId": "6f495573706f41326*****0706833347970646259773d3d",
      "clientIp": "2400:da00:e003:a806::4"
    },
    {
      "zone": "zoneB",
      "vpclId": "vpc-vjf***pq9pkf",
      "mountId": "6f495573706f41326*****0706833347970646259773d3d",
      "clientIp": "2400:da00:e003:a806::52"
    }
  ]
}
```

释放文件系统

描述

- 释放指定的FileSystem，被释放的FileSystem超过回收时间后会被永远删除，无法找回。

请求

请求结构

```
DELETE /v{version}/cfs/{fsId} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
fsId	String	是	URL参数	待释放的FileSystem的ID

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数：无

示例

请求：

```
DELETE /v1/cfs/cfs-jVDkWrBBoj HTTP/1.1
HOST cfs.bj.baidubce.com
Authorization bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS
```

更新文件系统标签

描述

- 更新文件系统标签
 - 指定单个FileSystem时，将此FileSystem标签全量更新为参数指定的标签 (不填标签则解绑该FileSystem所有标签)
 - 指定多个FileSystem时，将指定的FileSystem标签增加参数指定的标签 (不填标签则无实际作用)
 - 可传FileSystem个数限制为10，标签个数限制为100

请求

请求结构

```
POST /v{version}/cfs?tag HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string

{
  "fsId": [
    "cfs-xxxxx",
    "cfs-yyyyy"
  ],
  "tags": [
    { "tagKey": "aaa", "tagValue": "bbb" },
    { "tagKey": "bbb", "tagValue": "ccc" }
  ]
}
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
tag	None	是	Query参数	
fsId	List<String>	是	RequestBody参数	FileSystem ID 列表
tags	List<Tag>	否	RequestBody参数	

返回参数：除公共头域外，无其他特殊头域

示例

请求：

```
POST /v{version}/cfs?tag HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string

{
  "fsId": [
    "cfs-xxxxx",
    "cfs-yyyyy"
  ],
  "tags": [
    { "tagKey": "aaa", "tagValue": "bbb" },
    { "tagKey": "bbb", "tagValue": "ccc" }
  ]
}
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS
```

挂载点相关接口

创建挂载点

描述

- 创建一个文件系统的挂载点，返回domain。
- 付费方式为后付费。

请求

请求结构

```
POST /v{version}/cfs/{fsId} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string

{
  "subnetId": subnetId
}
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
fsId	String	是	URL参数	待创建MountTarget的FileSystem的ID
subnetId	String	是	RequestBody参数	MountTarget所属子网，subnet属于fs所在vpc，为短id
vpId	String	是	RequestBody参数	fs实例vip所属VPC的短Id

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数

参数名称	类型	描述
domain	String	分配的服务dns，通过此dns执行文件系统挂载，即可访问服务
mountId	String	mount点的ID

示例

请求：

```
POST /v1/cfs/cfs-jVDkWrBBoj HTTP/1.1
HOST cfs.bj.baidubce.com
Authorization bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02

{
  "subnetId": "sbn-qwkkxpvenm0z"
}
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS

{
  "mountId": "53585364776c6d577a2f424444796a2b6641584251773d3d"
  "domain": "cfs-jVDkWrBBoj-sdfoijnvsdf.bce.qa.sandbox.baidu.com"
}
```

查询挂载点

描述

- 查询指定文件系统下的所有MountTarget信息。
- 返回结果是多重查询条件的交集
- 结果支持marker分页，分页大小默认为1000，可通过maxKeys参数指定

请求

请求结构

```
GET /v{version}/cfs/{fsId}?&mountId={mountId}&marker={marker}&maxKeys={maxKeys} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string
```

请求头域：除公共头域外，无其他特殊头域

请求参数

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
fsId	String	是	URL参数	要查询的FileSystem ID
mountId	String	否	Query参数	要查询的MountTarget的ID
marker	String	否	Query参数	批量获取列表的查询的起始位置，是一个由系统生成的字符串
maxKeys	int	否	Query参数	每页包含的最大数量，最大数量不超过1000。缺省值为1000

响应

返回状态码：成功返回200，失败返回见[错误码](#)

返回头域：除公共头域外，无其他特殊头域

返回参数

参数名称	类型	描述
mountTargetList	List< MountTargetModel >	包含查询结果的列表
marker	String	标记查询的起始位置，若结果列表为空，此项不存在
isTruncated	boolean	true表示后面还有数据，false表示已经是最后一页
nextMarker	String	获取下一页所需要传递的marker值。当isTruncated为false时，该域不出现
maxKeys	int	每页包含的最大数量

示例

请求：

```
GET /v1/cfs/cfs-jVDkWrBBoj?maxKeys=100 HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS

{
  "mountTargetList": [
    {
      "accessGroupName"; "DEFAULT_ACCESS_GROUP_NAME",
      "domain": "cfs-nDODXHw25b.lb-24cbde9e.cfs.gz.baidubce.com",
      "mountId"; "362f315475536d46314c626e5a46724b7a7553432f413d3d",
      "ovip": "172.16.0.53",
      "subnetId": "sbn-d9uhhv2heg14"
      "vpclId": "vpc-jx648pkxw3tc"
    },
    {
      "accessGroupName"; "DEFAULT_ACCESS_GROUP_NAME",
      "domain": "cfs-jVDkWrBBoj-kweicskdgsev.bce.qa.sandbox.baidu.com",
      "mountId"; "36684e635250353569345454684e76364c36457057513d3d",
      "ovip": "128.1.0.1",
      "subnetId": "sbn-d9uhhv2heg14"
      "vpclId": "vpc-jx648pkxw3tc"
    }
  ],
  "marker": "kweicskdgsev",
  "nextMarker": "kweicskdgsev",
  "isTruncated": false,
  "maxKeys": 2
}
```

删除挂载点

描述

- 释放指定的FileSystem的MountTarget

请求

请求结构

```
DELETE /v{version}/cfs/{fsId}/{mountId} HTTP/1.1
Host: cfs.bj.baidubce.com
Authorization: authorization string
```

请求头域

除公共头域外，无其他特殊头域

参数名称	类型	是否必需	参数位置	描述
version	String	是	URL参数	API版本号，当前取值1
fsId	String	是	URL参数	待释放的target的FileSystem的ID
mountId	String	是	URL参数	待释放的target的ID

响应

返回状态码：成功返回200，失败返回见错误码

返回头域：除公共头域外，无其他特殊头域

返回参数：无

示例

请求：

```
DELETE /v1/cfs/cfs-jVDkWrBBoj/53585364776c6d577a2f424444796a2b6641584251773d3d HTTP/1.1
HOST cfs.bj.baidubce.com
Authorization bce-auth-v1/5e5a8adf11ae475ba95f1bd38228b44f/2016-04-10T08:26:52Z/1800/host;x-bce-date/ec3c0069f9abb1e247773a62707224124b2b31b4c171133677f9042969791f02
```

响应：

```
HTTP/1.1 200 OK
x-bce-request-id: 946002ee-cb4f-4aad-b686-5be55df27f09
Date: Wed, 10 Apr 2016 08:26:52 GMT
Transfer-Encoding: chunked
Content-Type: application/json;charset=UTF-8
Server: BWS
```

数据类型

🔗 Model对象定义

🔗 FileSystemModel

参数名称	类型	描述
fsId	String	文件系统的id
fsName	String	文件系统的名称
type	String	文件系统的存储类型
protocol	String	文件系统的协议类型
zone	String	文件系统所在可用区
capacityQuota	uint64	文件系统容量quota，单位 bytes
status	FsStatus	文件系统的状态
vpId	String	vpc短Id
mountTargetList	MountTargetModel	文件系统的挂载点列表
tags	List<Tag>	文件系统绑定的标签列表
fsUsage	String	文件系统的已用容量
KMSKeyId	String	文件系统使用的加密密钥Id
createTime	String	文件系统的创建时间 (UTC)

🔗 MountTargetModel

参数名称	类型	描述
subnetId	String	挂载点所在子网ID
domain	String	挂载地址
mountId	String	挂载点ID
ovip	String	挂载点IP
vpId	String	挂载点所在VPCID
accessGroupName	String	挂载点权限组名称

ClientInfoModel

参数名称	类型	描述
clientIp	String	客户端Ip
zone	String	客户端可用区。 - 部分情况可能查询不到可用区，如客户端非BCC - 刚挂载的客户端在约10s内无法返回zone信息
vpId	String	挂载点所在vpId
mountId	String	挂载点Id

状态编码定义

FsStatus

状态	描述
available	运行中
updating	更新中
paused	已欠费
unavailable	暂不可用

Go-SDK

概述

本文档主要介绍CFS GO SDK的使用。在使用本文档前，您需要先了解CFS的一些基本知识。如果您对CFS还不太了解，请参考[入门概述](#)。

初始化

确认Endpoint

在您使用SDK配置Endpoint时，可先阅读开发人员指南中关于[CFS访问域名](#)的部分，理解Endpoint相关的概念。百度云目前开放了多区域支持，请参考[区域选择说明](#)。

获取密钥

要使用百度云CFS，您需要拥有一个有效的AK(Access Key ID)和SK(Secret Access Key)用来进行签名认证。AK/SK是由系统分配给用户的，均为字符串，用于标识用户，为访问CFS做签名验证。

可以通过如下步骤获得并了解您的AK/SK信息：

- [注册百度云账号](#)
- [创建AK/SK](#)

🔗 新建CFS Client

CFS Client是CFS控制面服务的客户端，为开发者与CFS控制面服务进行交互提供了一系列的方法。

🔗 使用AK/SK新建CFS Client

通过AK/SK方式访问CFS，用户可以参考如下代码新建一个CFS Client：

```
import (
    "github.com/baidubce/bce-sdk-go/services/cfs"
)

func main() {
    // 用户的Access Key ID和Secret Access Key
    ACCESS_KEY_ID, SECRET_ACCESS_KEY := <your-access-key-id>, <your-secret-access-key>
    // 用户指定的Endpoint
    ENDPOINT := <domain-name>
    // 初始化一个CFSClient
    cfsClient, err := cfs.NewClient(ACCESS_KEY_ID, SECRET_ACCESS_KEY, ENDPOINT)
}
```

在上面代码中，ACCESS_KEY_ID对应控制台中的“Access Key ID”，SECRET_ACCESS_KEY对应控制台中的“Access Key Secret”，获取方式请参考《[如何获取AKSK](#)》。第三个参数ENDPOINT支持用户自己指定域名，如果设置为空字符串，会使用默认域名作为CFS的控制面服务地址。

注意：ENDPOINT参数需要用指定区域的域名来进行定义，如服务所在区域为北京，则为cfs.bj.baidubce.com。

使用STS创建CFS Client

申请STS token

CFS可以通过STS机制实现第三方的临时授权访问。STS（Security Token Service）是百度云提供的临时授权服务。通过STS，您可以为第三方用户颁发一个自定义时效和权限的访问凭证。第三方用户可以使用该访问凭证直接调用百度云的API或SDK访问百度云资源。

通过STS方式访问CFS，用户需要先通过STS的client申请一个认证字符串，申请方式可参见[百度云STS使用介绍](#)。

用STS token新建CFS Client

申请好STS后，可将STS Token配置到CFS Client中，从而实现通过STS Token创建CFS Client。

代码示例

GO SDK实现了STS服务的接口，用户可以参考如下完整代码，实现申请STS Token和创建CFS Client对象：

```

import (
    "fmt"
    "github.com/baidubce/bce-sdk-go/auth" //导入认证模块
    "github.com/baidubce/bce-sdk-go/services/cfs" //导入CFS服务模块
    "github.com/baidubce/bce-sdk-go/services/sts" //导入STS服务模块
)

func main() {
    // 创建STS服务的Client对象，Endpoint使用默认值
    AK, SK := <your-access-key-id>, <your-secret-access-key>
    stsClient, err := sts.NewClient(AK, SK)
    EndPoint := "cfs.bj.baidubce.com"
    if err != nil {
        fmt.Println("create sts client object :", err)
        return
    }

    // 获取临时认证token，有效期为60秒，ACL为空
    stsObj, err := stsClient.GetSessionToken(60, "")
    if err != nil {
        fmt.Println("get session token failed:", err)
        return
    }
    fmt.Println("GetSessionToken result:")
    fmt.Println("  accessKeyId:", stsObj.AccessKeyId)
    fmt.Println("  secretAccessKey:", stsObj.SecretAccessKey)
    fmt.Println("  sessionToken:", stsObj.SessionToken)
    fmt.Println("  createTime:", stsObj.CreateTime)
    fmt.Println("  expiration:", stsObj.Expiration)
    fmt.Println("  userId:", stsObj.UserId)

    // 使用申请的临时STS创建CFS控制面服务的Client对象，Endpoint使用默认值
    cfsClient, err := cfs.NewClient(stsObj.AccessKeyId, stsObj.SecretAccessKey, EndPoint)
    if err != nil {
        fmt.Println("create cfs client failed:", err)
        return
    }
    stsCredential, err := auth.NewSessionBceCredentials(
        stsObj.AccessKeyId,
        stsObj.SecretAccessKey,
        stsObj.SessionToken)
    if err != nil {
        fmt.Println("create sts credential object failed:", err)
        return
    }
    cfsClient.Config.Credentials=stsCredential
}

```

注意：目前使用STS配置CFS Client时，无论对应CFS服务的Endpoint在哪里，STS的Endpoint都需配置为 <http://sts.bj.baidubce.com>。上述代码中创建STS对象时使用此默认值。

配置HTTPS协议访问CFS

CFS支持HTTPS传输协议，您可以通过在创建CFS Client对象时指定的Endpoint中指明HTTPS的方式，在CFS GO SDK中使用HTTPS访问CFS服务：

```
import "github.com/baidubce/bce-sdk-go/services/cfs"
ENDPOINT := "https://cfs.bj.baidubce.com" //指明使用HTTPS协议
AK, SK := <your-access-key-id>, <your-secret-access-key>
cfsClient, _ := cfs.NewClient(AK, SK, ENDPOINT)
```

配置CFS Client

如果用户需要配置CFS Client的一些细节的参数，可以在创建CFS Client对象之后，使用该对象的导出字段Config进行自定义配置，可以为客户端配置代理，最大连接数等参数。

使用代理

下面一段代码可以让客户端使用代理访问CFS服务：

```
import "github.com/baidubce/bce-sdk-go/services/cfs"
//创建CFS Client对象
AK, SK := <your-access-key-id>, <your-secret-access-key>
ENDPOINT := "cfs.bj.baidubce.com"
client, _ := cfs.NewClient(AK, SK, ENDPOINT)

//代理使用本地的8080端口
client.Config.ProxyUrl = "127.0.0.1:8080"
```

设置网络参数

用户可以通过如下的示例代码进行网络参数的设置：

```
import "github.com/baidubce/bce-sdk-go/services/cfs"
AK, SK := <your-access-key-id>, <your-secret-access-key>
ENDPOINT := "cfs.bj.baidubce.com"
client, _ := cfs.NewClient(AK, SK, ENDPOINT)

// 配置不进行重试，默认为Back Off重试
client.Config.Retry=bce.NewNoRetryPolicy()

// 配置连接超时时间为30秒
client.Config.ConnectionTimeoutInMillis=30*1000
```

配置生成签名字符串选项

```
import "github.com/baidubce/bce-sdk-go/services/cfs"
AK, SK := <your-access-key-id>, <your-secret-access-key>
ENDPOINT := "cfs.bj.baidubce.com"
client, _ := cfs.NewClient(AK, SK, ENDPOINT)

// 配置签名使用的HTTP请求头为`Host`
headersToSign := map[string]struct{}{"Host": struct{}}
client.Config.SignOption.HeadersToSign = HeadersToSign
// 配置签名的有效期为30秒
client.Config.SignOption.ExpireSeconds=30
```

参数说明 用户使用GO SDK访问CFS时，创建的CFS Client对象的Config字段支持的所有参数如下表所示：

配置项名称	类型	含义
Endpoint	string	请求服务的域名
ProxyUrl	string	客户端请求的代理地址
Region	string	请求资源的区域
UserAgent	string	用户名称，HTTP请求的User-Agent头
Credentials	*auth.BceCredentials	请求的鉴权对象，分为普通AK/SK与STS两种
SignOption	*auth.SignOptions	认证字符串签名选项
Retry	RetryPolicy	连接重试策略
ConnectionTimeoutInMillis	int	连接超时时间，单位毫秒，默认20分钟

说明：

1. `Credentials`字段使用`auth.NewBceCredentials`与`auth.NewSessionBceCredentials`函数创建，默认使用前者，后者为使用STS鉴权时使用，详见“使用STS创建CFS Client”小节。
2. `SignOption`字段为生成签名字符串时的选项，详见下表说明：

名称	类型	含义
HeadersToSign	map[string]struct{}	生成签名字符串时使用的HTTP头
Timestamp	int64	生成的签名字符串中使用的时间戳，默认使用请求发送时的值

其中，`HeadersToSign`默认为`Host`，`Content-Type`，`Content-Length`，`Content-MD5`；`TimeStamp`一般为零值，表示使用调用生成认证字符串时的时间戳，用户一般不应该明确指定该字段的值；`ExpireSeconds`默认为1800秒即30分钟。

3. `Retry`字段指定重试策略，目前支持两种：`NoRetryPolicy`和`BackOffRetryPolicy`。默认使用后者，该重试策略是指定最大重试次数、最长重试时间和重试基数，按照重试基数乘以2的指数级增长的方式进行重试，直到达到最大重试次数或者最长重试时间为止。

文件系统

创建文件系统

通过以下代码，可以创建一个CFS文件系统，返回对应的文件系统ID。

```
args := &cfs.CreateFSArgs{
    ClientToken: "be31b98c-5e41-4838-9830-9be700de5a20",
    Name:       "sdkCFS", // 设置实例名称
    VpcID:      vpcId, // 设置实例所属vpc
    Protocol:   protocol, // 设置实例所属协议类型：1.nfs 2.smb
    Zone:       zone, // 设置实例所属可用区
}

result, err := cfsClient.CreateFS(args)
if err != nil {
    fmt.Println("create cfs failed:", err)
} else {
    fmt.Println("create cfs success: ", result)
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [创建文件系统](#)。

更新文件系统

通过以下代码，可以更新一个CFS文件系统的配置信息，如文件系统名称。

```
args := &cfs.UpdateFSArgs{
    FSID:    "cfs-xxxx", // 实例ID
    FSName:  "testSdk",
}
err := cfsClient.UpdateFS(args)
if err != nil {
    fmt.Println("update cfs failed:", err)
} else {
    fmt.Println("update cfs success")
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [更新文件系统](#)。

🔗 查询已有文件系统

通过以下代码，可以查询用户账户下所有CFS文件系统的信息

```
args := &cfs.DescribeFSArgs{}

// 支持按fsId、userId，匹配规则支持部分包含（不支持正则）
args.FSID = cfsId
args.UserId = userId
result, err := cfsClient.DescribeFS(args)
if err != nil {
    fmt.Println("list all cfs failed:", err)
} else {
    fmt.Println("list all cfs success: ", result)
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [查询文件系统](#)。

🔗 释放文件系统

通过以下代码，可以释放指定CFS文件系统，被释放的CFS文件系统无法找回(删除实例之前，需要先删除挂载点，否则会删除失败)。

```
args := &cfs.DropFSArgs{}
args.FSID = cfsId

err := client.DropFS(args)
if err != nil {
    fmt.Println("delete cfs failed:", err)
} else {
    fmt.Println("delete cfs success")
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [释放文件系统](#)。

挂载点

🔗 创建挂载点

通过以下代码，在指定CFS文件系统下，创建一个文件系统的挂载点，返回domain。

```
args := &cfs.CreateMountTargetArgs{
    // 所属文件系统实例ID
    FSID: cfsId,
    // 所属子网ID
    SubnetId: subnetId,
    // 所属vpc短ID
    VpcId: vpcId,
}
err := client.CreateMountTarget(args)
if err != nil {
    fmt.Println("create Mount Target failed:", err)
} else {
    fmt.Println("create Mount Target success")
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [创建挂载点](#)。

🔗 查询挂载点

通过以下代码，查询指定CFS文件系统下所有挂载点信息，支持按挂载点匹配查询，结果支持marker分页，分页大小默认为1000，可通过maxKeys参数指定。

```
args := &cfs.DescribeMountTargetArgs{
    // 要查询的文件系统实例id
    FSID: cfsId,
}
result, err := client.DescribeMountTarget(args)
if err != nil {
    fmt.Println("describe Mount Target failed:", err)
} else {
    fmt.Println("describe Mount Target success: ", result)
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [查询挂载点](#)。

🔗 删除挂载点

通过以下代码，释放指定CFS文件系统下的挂载点。

```
args:=&cfs.DropMountTargetArgs{
    // 要删除的文件系统实例ID
    FSID:  cfsId,
    MountId: mountId,
}
err := client.DropMountTarget(args)
if err != nil {
    fmt.Println("delete Mount Target failed:", err)
} else {
    fmt.Println("delete Mount Target success: ")
}
```

提示：详细的参数配置及限制条件，可以参考CFS API 文档 [删除挂载点](#)。

错误处理

GO语言以error类型标识错误，CFS支持两种错误见下表：

错误类型	说明
BceClientError	用户操作产生的错误
BceServiceError	CFS服务返回的错误

用户使用SDK调用CFS相关接口，除了返回所需的结果之外还会返回错误，用户可以获取相关错误进行处理。实例如下：

```
// cfsClient 为已创建的CFS Client对象
describeArgs := &DescribeFSArgs {
    FSID: CFS_ID,
}
cfsDetail, err := cfsClient.DescribeFS(describeArgs)
if err != nil {
    switch realErr := err.(type) {
    case *bce.BceClientError:
        fmt.Println("client occurs error:", realErr.Error())
    case *bce.BceServiceError:
        fmt.Println("service occurs error:", realErr.Error())
    default:
        fmt.Println("unknown error:", err)
    }
} else {
    fmt.Println("get cfs detail success: ", cfsDetail)
}
```

客户端异常

客户端异常表示客户端尝试向CFS发送请求以及数据传输时遇到的异常。例如，当发送请求时网络连接不可用时，则会返回BceClientError。

服务端异常

当CFS服务端出现异常时，CFS服务端会返回给用户相应的错误信息，以便定位问题。

常见问题

常见问题总览

一般类问题

- [CFS 支持哪些存储类型](#)
- [CFS 支持哪些协议类型](#)
- [每个用户可以创建多少个文件系统](#)
- [每个文件系统可以创建多少个挂载点](#)
- [挂载点可以跨可用区访问吗](#)
- [如何备份 CFS 的数据](#)

- [CFS 开放了哪些区域](#)
- [CFS 如何保证数据可用性及可靠性](#)
- [应该如何在对象存储 BOS、块存储 CDS 和文件存储 CFS 间进行选择](#)
- [如何查询客户端读/写吞吐流量](#)

🔗 计费类问题

- [CFS 如何收费](#)
- [一个文件系统用于计费的实际使用空间是如何计算的](#)
- [一个普通文件的存储空间是如何计算的](#)
- [文件空洞是否计费](#)
- [欠费之后是否还会继续累加计费](#)
- [购买存储包可以提升文件系统的最大吞吐吗](#)

🔗 NFS协议问题

- [支持的 NFS 协议版本](#)
- [为什么不支持 NFS 的其它版本](#)
- [推荐的操作系统版本](#)
- [不支持的特性](#)
- [如何解决在NFS协议文件系统中创建文件后报不存在问题？](#)
- [如何解决向NFS协议文件系统中写入数据延迟问题？](#)
- [挂载的时候报“mount: wrong fs type, bad option, bad superblock”](#)
- [遍历文件夹时得到的列表有漏项、重复、更新滞后等情况](#)
- [主机名和内网 IP 都相同的客户端请求相互干扰](#)
- [客户端无法连接挂载点，内核日志提示“nfs: server XXX not responding, still trying”](#)
- [目录下发现一些“.nfsXXXX”类似名字的文件](#)
- [客户端吞吐或 qps 较低](#)
- [多个客户端并发 append 写同一文件数据互相覆盖](#)

一般类问题

🔗 CFS 支持哪些存储类型

CFS 目前仅支持一种规格的存储类型，采用 SSD 介质存储数据。

🔗 CFS 支持哪些协议类型

CFS 目前支持使用比较广泛的 NFS 4.1 协议，SMB/CIFS 和 NFS 4.0 版本的支持开发中，敬请期待。

🔗 每个用户可以创建多少个文件系统

同一region内，每个用户最多可创建20个文件系统。

每个文件系统可以创建多少个挂载点

目前，对一个文件系统可以创建的挂载点总数量没有限制。但是，一个 VPC 内的子网只能创建一个挂载点。

挂载点可以跨可用区访问吗

可以，只要挂载点的地址和访问端网络是连通的即可。

CFS 的备份功能是否计费

CFS 的备份功能当前处于免费试用期，待免费试用期结束进入收费阶段前会统一发布具体计费说明。

CFS 开放了哪些区域

您可以在控制台界面通过切换地域查看某个地域是否提供 CFS 服务。

CFS 如何保证数据可用性及可靠性

CFS 中每一份数据均会冗余存储到多个副本，每个模块均有负载均衡、自动故障转移等功能，最大限度保证数据的可用性及可靠性。

应该如何在对象存储 BOS、块存储 CDS 和文件存储 CFS 间进行选择

您可以根据自己的使用场景来评估使用何种类型的存储，以下是几种产品的主要区别：

	BOS	CDS	CFS
接口形式	HTTP 接口	块设备接口	Posix 兼容接口
随机读写	不支持	支持	支持
延时	最高	最低	高于 CDS，低于 BOS
数据可靠性	多个副本，跨多个可用区存储	多个副本，单个可用区内存储	多个副本，单个可用区内存储
数据一致性	强一致	强一致	强一致
最大容量	无限	几十 TB	百 PB
客户端同时访问能力	可以被多个客户端同时访问	不可以被多个客户端同时访问	可以被多个客户端同时访问
客户端地域限制	可公网、任意地域访问	同一可用区内的虚拟机	同一区域内的虚拟机或容器
主要使用场景	海量数据的分发、备份，数据分析和数据处理	无需数据共享的场景，如虚拟机 root 盘，I/O 密集型业务（例如数据库）	数据共享数据的场景，如数据分析、数据处理、共享日志等，与容器配合较好

如何查询客户端读写吞吐流量

当您一个文件系统挂载了很多客户端，想快速查找出读/写吞吐瞬时流量top的客户端；或者您想知道文件系统挂载的具体客户端的历史读/写吞吐流量和统计情况，您可以参考[查看文件系统详情](#)文档进行操作。

计费类问题

CFS 如何收费

目前，CFS 提供按需计费与存储包计费两种方式，收费仅依据一个文件系统实际使用的空间，无最低费用、流量费和请求费用。有关 CFS 的定价信息，请访问 [CFS 计费](#) 页面。

一个文件系统用于计费的实际使用空间是如何计算的

CFS 的计费系统会周期采集一个文件系统的实际使用空间，按照一个小时内采集到的峰值作为计费依据。一个文件系统实际存储空间是所有普通文件的实际存储空间之和。CFS 不对除文件外的其它类型目录项，如文件夹、软链接、特殊文件等，进行收费。

☞ 一个普通文件的存储空间是如何计算的

CFS 将一个文件按照 1MB 作为一个基本的粒度进行分片。分片是 CFS 进行空间统计的一个基本单位，一个文件的存储空间就是所有分片存储空间的总和。

每个分片的存储空间按照如下规则进行计算：

- 规则1：如果文件的最后一个分片的大小不足 1MB，按照该分片的大小，对齐到 4KB，计算存储空间；
- 规则2：其它情况下，如果分片写过数据，按照整个分片的大小，即 1MB，计算存储空间；
- 规则3：一个文件最小存储空间大小为 4KB。

举几个例子说明：

- 例子一：一个文件的大小是 5KB，那么这个文件无论是否写过数据，按照规则 1，存储空间均为 8KB；
- 例子二：一个文件的大小是 1025KB。[1024KB, 1025KB) 这个分片存储空间为 4KB。那么，如果 [0, 1024KB) 这个分片写过数据，则存储空间为 1MB + 4KB = 1028KB，如果未写过数据，则存储空间为 0MB + 4KB = 4KB；
- 例子三：一个文件 1MB，从未写过数据，按照规则 3，存储空间为 4KB。

☞ 文件空洞是否计费

按照上述的存储空间计算规则，文件空洞不产生实际的存储空间，不收取费用。

☞ 欠费之后是否还会继续累加计费

在欠费之后，CFS 系统会停止服务，数据保留 7 天。停服期间，文件系统的数据不可访问，但是仍然会占用资源，所以会继续收费。停服期间，如果用户续费成功，服务恢复。否则，超过 7 天之后，文件系统将会被强制删除，资源释放，用户数据无法找回。

用户应该关注欠费通知，保持账户余额充足，避免文件系统停服，造成不必要的经济损失和数据丢失。

☞ 购买存储包可以提升文件系统的最大吞吐吗

不能。CFS 文件系统的最大吞吐随实际使用量线性增长，存储包只能抵扣使用费用，与最大吞吐无关。更多信息，请参见[产品规格](#)。

NFS协议问题

☞ 支持的 NFS 协议版本

目前，CFS 支持 NFS 4.1 协议。

☞ 为什么不支持 NFS 的其它版本

从 2.6 系列开始，Linux 内核就已经支持了 NFS 4.0，协议，所以支持 NFS 3 及更低版本的唯一理由可能就是 Windows 操作系统。但 Windows 操作系统对 NFS 协议的支持并不好。在 Windows 上，首选的共享文件系统是 SMB/CIFS 协议，该协议本身也被较新 Linux（内核版本 > 3.4）支持。所以，CFS 选择不支持 NFS 3 及更低版本的 NFS 协议。

NFS 4.X 系列中，使用较为广泛的是 4.0 和 4.1 版本。4.1 修正了 4.0 版本的一些设计缺陷，引入了 session、pNFS、directory delegation 等特性，尽管只有小版本号发生变化，但实际差异较大，所以 CFS 选择优先支持性能更佳的 NFS 4.1 协议。

☞ 推荐的操作系统版本

Linux 官方内核对 NFS 4.1 的开发从 2.6.31 版本开始，到 2.6.36 版本发布了实验性支持，最后到 3.7 版本，提供了正式支持。具体信息参见 [链接](#)。因此，用户应该尽量选择 3.7 及以上版本的内核。通常情况下，更新的稳定版本内核可以提供更好的性能和体验。

部分操作系统显示的内核版本，其代码与 Linux 官方的同版本内核并不完全一致，例如 CentOS/RHEL，这些发行版会自己做一些 bugfix、修改或者移植的工作，其代码通常超前于同版本的官方内核。因此，各个操作系统的实际情况以官方文档和源代码代码为准。

对于常见的操作系统，我们推荐的版本如下：

- CentOS/RHEL：推荐 6.6 及以上版本。参见 [链接](#) 获取该操作系统上更详细的 NFS 协议 bug 信息；
- Ubuntu：推荐 14.04 及以上版本；
- Debian：推荐 8 及以上版本；
- OpenSUSE：推荐 42.3 及以上版本。

不支持的特性

CFS 不支持的 Attributes 包括：FATTR4_ACL，FATTR4_ACLSUPPORT，FATTR4_ARCHIVE，FATTR4_HIDDEN、FATTR4_MIMETYPE、FATTR4_QUOTA_AVAIL_HARD、FATTR4_QUOTA_AVAIL_SOFT、FATTR4_QUOTA_USED、FATTR4_SYSTEM、FATTR4_TIME_BACKUP、FATTR4_TIME_CREATE、FATTR4_DIR_NOTIF_DELAY、FATTR4_DIRENT_NOTIF_DELAY、FATTR4_DACL、FATTR4_SACL、FATTR4_CHANGE_POLICY、FATTR4_FS_STATUS、FATTR4_LAYOUT_HINT、FATTR4_LAYOUT_ALIGNMENT、FATTR4_FS_LOCATIONS_INFO、FATTR4_MDSTHRESHOLD、FATTR4_RETENTION_GET、FATTR4_RETENTION_SET、FATTR4_RETENT EVT_GET、FATTR4_RETENT EVT_SET、FATTR4_RETENTION_HOLD、FATTR4_MODE_SET_MASKED、FATTR4_FS_CHARSET_CAP。

CFS 支持的文件锁为非阻塞（nonblocking）、建议性质，支持锁定整个文件和或者部分字节范围，其它种类的锁定目前不被支持，这意味着，在使用 3 种常见的锁定接口时需要注意：

- flock 需同时指定 LOCK_NB 参数；
- lockf 不支持 F_LOCK 选项；
- fcntl 不支持 F_SETLKW 选项。

CFS 不支持 Delegation、pNFS、访问控制列表（ACL）、Kerberos 安全等特性。

出于性能的考虑，只读操作时 CFS 不更新文件读取的 time_access 属性。

CFS 没有实现持久化 Session、Replay Cache 等信息，当发生故障转移时，客户端会重新建立 Session。该行为符合 RFC 5661 的规定，通常客户端可以自动处理。

如何解决在 NFS 协议文件系统中创建文件后报不存在问题？

问题现象：

现象一：BCC-1 创建了文件 file1，但是 BCC-2 需要过一段时间才能看到 BCC-1 创建的文件 file1，有时会延迟 1s，有时甚至会到 1 分钟。

现象二：BCC-1 创建了文件 file1，BCC-2 在一段时间后 Open 过该文件；BCC-1 删除 file1 后又创建了同名文件 file1，BCC-2 再次尝试访问该文件会报不存在。

问题原因：这两个现象都是 Lookup Cache 导致的。

对于现象一：BCC-2 在 BCC-1 创建文件 file1 前进行了访问，导致 BCC-2 发生文件不存在，于是缓存了一条文件 file1 不存在的记录。由于 FileAttr 还没有过期，BCC-2 再次访问时，仍会访问第一次缓存到文件 file1 不存在的记录（Negative Lookup Cache）。

对于现象二：BCC-2 访问过 file1，缓存了一条文件 file1 的 inode 记录（Positive Lookup Cache）。由于 BCC-1 删除并创建了同名

的file1后，BCC-2再次访问时，仍会根据相同路径拿到file1的inode记录，并向服务器发起请求。

解决方案：

方案一：对于现象一，关闭BCC-2的Negative Lookup Cache，不缓存不存在的文件。挂载时，添加lookupcache=positive（默认值lookupcache=all）字段，挂载命令如下所示：

```
mount -t nfs4 -o  
minorversion=1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport,lookupcache=positive <挂载地  
址>:/ <客户端本地路径>
```

说明：

- <挂载地址> 可以在文件系统的挂载点列表中查到(见[获取挂载点地址](#))；
- CFS文件系统的路径：默认配置为CFS的根目录（例如：/），您可以修改成已存在的子目录（例如：/dir0）；
- <客户端本地路径> 是待挂载客户端的本地路径，必须是以/开头的绝对路径（例如：/mnt/cfs），且该路径必须已被创建。

方案二：对于所有Lookup Cache导致的问题均有效。

关闭BCC-2的Lookup Cache，该方案会导致性能非常差，请根据业务实际情况选择合适的方案。挂载时，添加lookupcache=none字段，挂载命令如下所示：

```
mount -t nfs4 -o  
minorversion=1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport,lookupcache=none <挂载地址>:/  
<客户端本地路径>
```

同时也可以选择关闭所有缓存，该方案性能更差。挂载时，添加actimeo=0字段，挂载命令如下所示：

```
mount -t nfs4 -o minorversion=1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport,actimeo=0 <挂  
载地址>:/ <客户端本地路径>
```

说明

- Lookup Cache有三种选项：all、none、positive，分别含义为：
 - all：positive、negative均开启，缓存文件/目录存在、不存在的记录，此为默认选项
 - none：完全不开启Lookup Cache
 - positive：只记录文件/目录存在的记录

其中，Lookup Cache会根据文件/目录的mtime淘汰缓存，由于文件/目录本身有属性缓存，所以淘汰时间可以参考actimeo（acregmin、acregmax、acdirmin、acdirmax）、noac等参数的配置情况。

☞ 如何解决向NFS协议文件系统中写入数据延迟问题？

问题现象：BCC-1更新了文件file1，但是BCC-2立即去读它，仍然是旧的内容，这是为什么？

问题原因：涉及如下两个原因。

第一个原因：BCC-1写了file1后，不会立即flush，会先进行PageCache，依赖应用层调用fsync或者close。第二个原因：BCC-2存在文件Cache，可能不会立即去服务端取最新的内容。例如，BCC-2在BCC-1更新文件file1之时，就已经缓存了数据，当BCC-

2再次去读时，仍然使用了缓存中的内容。

解决方案：如果要保证BCC-1创建文件后，BCC-2立即就能看到它，可以使用如下方案：

方案一：利用CTO一致性，让BCC-1或BCC-2的读写符合CTO模式，则BCC-2一定能读到最新数据。具体来说，BCC-1更新文件后，一定要执行close或者执行fsync。BCC-2读之前，重新open，然后再去读。

方案二：关闭BCC-1和BCC-2的所有缓存。该方案会导致性能非常差，请根据业务实际情况选择合适的方案。

- 关闭BCC-1的缓存。挂载时，添加noac字段，保证所有写入立即落盘。挂载命令如下所示：

```
mount -t nfs4 -o minorversion=1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,noac <挂载地址>:/ <客户端本地路径>
```

说明

- 如果BCC-1的写操作完成后会调用fsync，或者使用sync写，可以将上面的noac替换为actimeo=0，性能会稍好一点。
- noac等价于actimeo=0加sync（即，强制所有写入都为sync写）。

其中，actimeo为文件、目录的缓存有效时间（单位为秒），其中包括：

- acregmin：文件属性最小有效时间，默认3秒
- acregmax：文件属性最大有效时间，默认60秒
- acdirmin：目录属性最小有效时间，默认30秒
- acdirmax：目录属性最大有效时间，默认60秒

acregmin、acregmax、acdirmin、acdirmax均可单独配置，如果配置了actimeo则上述4个参数的值跟actimeo保持一致。

- 关闭BCC-2的缓存。挂载时，添加actimeo=0字段，忽略所有缓存。挂载命令如下所示：

```
mount -t nfs4 -o minorversion=1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport,actimeo=0 <挂载地址>:/ <客户端本地路径>
```

🔗 挂载的时候报“mount: wrong fs type, bad option, bad superblock”

多数操作系统默认不会安装 NFS 客户端，需要手动安装。

- 如果您使用 CentOS/RHEL 操作系统，请运行以下命令：

```
yum install nfs-utils
```

- 如果您使用 Ubuntu/Debian 操作系统，请运行以下命令：

```
apt-get install nfs-common
```

上述命令均需要 root 权限。

🔗 遍历文件夹时得到的列表有漏项、重复、更新滞后等情况

当一个客户端通过编程接口 readdir/getdents/getdents64 或者 ls 命令，遍历一个文件夹时，如果有另外一个客户端同时在该文件夹下有并发的创建、删除、重命名操作，用户看到的列表可能会出现漏项和重复。这个属于并发操作互相干扰所致，根本原因是由于一次完整的遍历由若干次连续的获取操作组成，一次只能获取部分文件项，连续的操作中间目录发生了变化。

通常情况下，客户端会使用目录项及属性缓存来提升性能，但是缓存也意味着一个客户端的修改可能不会马上被其它的客户端看到。这就导致了更新滞后的情况。用户可以在挂载的时候使用 `noac` 选项来禁用缓存。该方式会导致 `OP_GETATTR` 远程调用的数量大大增加，对性能造成较大的伤害，我们不推荐使用。

🔗 主机名和内网 IP 都相同的客户端请求相互干扰

对于同一个文件系统的两个客户端，如果它们的主机名和内网 IP 地址均相同，CFS 无法区分，请求会相互干扰。通常在容器的场景下容易发生这样的问题，因为同一机器上的内网 IP 是相同的，如果主机名再相同，就会被认为是同一客户端。

部分较老的 Linux 内核在编码 NFS 协议客户端标示字符 `client owner` 的时候，会截断主机名，取前面的部分字符，所以，应该尽可能将主机名字的差异部分编码在比较靠前的位置。根据我们的经验，差异字符控制在前 32 个字符以内，是比较稳妥的。

总之，您应该避免出现主机名和内网 IP 都相同的情况。

🔗 客户端无法连接挂载点，内核日志提示 “nfs: server XXX not responding, still trying”

内核日志中 XXX 为挂载点地址，以下以 <挂载点地址> 代表。按照以下步骤排查该问题：

- ping <挂载点地址>，确认该地址可以连通。如果不能连通，检查网页上相应的挂载点地址是否还存在；
- 如果上述步骤无问题，telnet <挂载点地址> 2049，确认和后端服务 NFS 协议端口号是否连通。如不连通，排查安全组策略是否禁用了相关端口号；
- 如果上述步骤无问题，检查 mount 参数是否设置了 `noresvport`，如果未设置，尝试重新以新选项挂载。

如果仍然存在问题，您可以发工单描述相关问题，联系工程师排查。

🔗 目录下发现一些 “.nfsXXXX” 类似名字的文件

这些文件是所谓 “silly rename” 机制产生的。

在类 UNIX 的操作系统上，一个文件在打开之后被删除，只是变成不可见，进程仍然可以通过文件描述符继续读写。当不再有进程打开该文件时，文件才会真正删除。在早期的 NFS 版本（NFS 3 及以前）中，并不能保证服务端继续保留该文件，因此，内核采用了一个小技巧，在删除的时候将文件重命名为 “.nfsXXXX” 这样的隐藏文件，当不再使用的时候删除。这就是所谓的 “silly rename” 机制。如果发生客户端宕机或者网络中断，可能导致客户端没有成功删除掉 “.nfsXXXX” 文件，这些文件就残留了下来。

NFS 4.X 协议已经能够处理其中的问题，但是内核为了向后兼容，仍然使用了 “silly rename” 机制。

您在列举文件夹内项目的时候，可以过滤掉这样的文件。当您确认没有进程在使用这些文件时，可以安全的删除文件。

🔗 客户端吞吐或 qps 较低

客户端实际吞吐比较低，可能是由于当前session的slot设置导致的，挂载CFS实例后，可以通过以下步骤确认：

- 1、安装systool工具。CentOS/RHEL 操作系统，通过以下命令安装：

```
yum install sysfsutils
```

Ubuntu/Debian 操作系统，通过以下命令安装：

```
apt-get install sysfsutils
```

- 2、查看当前slot值，执行前请确保已经成功挂载CFS实例：

```
systool -v -m nfs
```

执行结果中“max_session_slot”对应值即是当前session的并发值，如果该值较低，可以调大到 128。

1、修改环境变量

```
echo options nfs max_session_slots=128 > /etc/modprobe.d/nfsclient.conf
```

2. 重启系统

3. 重启后再次通过systool确认是否修改成功。

```
systool -v -m nfs
```

🔗 多个客户端并发 append 写同一文件数据互相覆盖

NFS 协议不支持原子的 append 写。为了达到本地文件系统上追加写的效果，需要先锁定文件，seek 到正确的位置，再写入数据。示例代码如下：

```
int lock_file(int fd) {
    for (;;) {
        int rc = flock(fd, LOCK_EX | LOCK_NB);
        if (rc == 0) {
            return 0;
        }
        if (errno == EINTR) {
            continue; // interrupt by a signal, retry
        } else if (errno == EWOULDBLOCK) {
            usleep((rand() % 10) * 100); // sleep a while to retry
            continue;
        } else {
            break;
        }
    }
    return -1; // lock fail
}

int unlock_file(int fd) {
    for (;;) {
        int rc = flock(fd, LOCK_UN);
        if (rc == 0) {
            return 0;
        }
        if (errno == EINTR) {
            continue; // interrupt by a signal, retry
        } else {
            break;
        }
    }
    return -1; // unlock fail
}

int append_write(int fd, const char* data, ssize_t size) {
    int rc = lock_file(fd);
    if (rc != 0) {
        return rc;
    }
    off_t offset = lseek(fd, 0, SEEK_END);
    if (offset < 0) {
        unlock_file(fd);
        return -1;
    }
    while (size > 0) {
        ssize_t nwriten = pwrite(fd, buf, size, offset);
        if (nwriten >= 0) {
            size -= nwriten;
            offset += nwriten;
        } else if (errno == EINTR) {
            continue;
        } else {
            unlock_file(fd);
            return -1;
        }
    }
    return unlock_file(fd);
}
```

SMB协议问题

CFS目前支持SMB 1、SMB 2和SMB 3等版本的主要功能。

SMB分别支持哪些版本的操作系统？

由于SMB 1在安全性还是性能方面都存在诸多问题，因此强烈建议用户使用SMB 2及以上版本，即Windows Vista及之后版本的操作系统客户端。各Windows版本对SMB协议的支持情况具体可参考下表：

SMB版本	年代	相应操作系统版本
SMB 3.1.1	2015	Windows 10 / Windows Server 2016
SMB 3.0.2	2013	Windows 8.1 / Windows Server 2012 R2
SMB 3.0	2012	Windows 8 / Windows Server 2012
SMB 2.1	2009	Windows 7 / Windows Server 2008 R2
SMB 2.0.2	2008	Windows Vista SP1 / Windows Server 2008
SMB 2.0	2007	Windows Vista

CFS SMB不支持哪些特性？

目前CFS SMB暂不支持以下特性：

- 不支持跨协议访问，即不支持用户使用NFS和SMB访问同一文件系统。
- 不支持文件字节范围锁，以及基于Oplocks和Lease的客户端缓存。
- 不支持文件ACL控制。
- 不支持文件异步IO。
- 不支持多通道、Persistent File Handle、SMB Direct、SMB Directory Leasing等SMB 3高级特性。
- 不支持Sparse files、文件压缩、网卡状态查询、重解析点（Reparse Point）等IOCTL/FSCTL操作。

为什么Windows资源管理器中部分编辑器对文件属性的更新有所滞后？

以记事本为例：用户打开记事本写入内容后保存，可能发现文件的大小并未及时更新，此时右键查看常规属性或在Windows资源管理器中刷新即可更新显示大小。

该问题的产生与Windows下部分编辑器（如记事本）在文件修改过程中对于资源管理器显示的属性更新有关，此时文件自身的属性本身已经更新，只是资源管理器未更新显示而已，因此本身对正常使用并没有影响。对于写字板等某些文件编辑软件，则不存在该问题。

服务等级协议SLA

CFS服务等级协议（SLA）

协议生效时间：2019 年 03 月 01 日

本服务等级协议（Service Level Agreement，简称“SLA”）规定了百度智能云向客户提供的文件存储服务CFS（Cloud File System）的服务可用性等级指标及赔偿方案。

1. 定义

服务周期：一个服务周期为一个自然月。

服务周期总分钟数：按照单CFS文件系统实例服务周期内的总天数 * 24（小时） * 60（分钟）计算。**服务不可用分钟数：**在连续的一分钟内，若向某CFS文件系统实例的请求均失败，则视为该分钟内该CFS文件系统实例不可用。在一个服务周期内CFS

单个文件系统实例不可用分钟数之和即服务不可用分钟数。**月度服务费**：为客户在一个服务周期（即自然月）中就单CFS实例所支付的服务费用总额，如客户一次性支付多个月份的服务费，则将按照所购买的月数分摊计算月度服务费。

2. 服务可用性

2.1 服务可用性计算公式

服务可用性以CFS单个文件系统实例为维度，按照如下方式计算：

服务可用性=
$$\left(\frac{\text{服务周期总分钟数}-\text{服务不可用分钟数}}{\text{服务周期总分钟数}}\right) \times 100\%$$

2.2 服务可用性承诺

百度智能云文件存储CFS承诺在每个服务周期内的服务可用性不低于99.9%。

如CFS未达到上述服务可用性承诺，客户可以申请获得赔偿。

赔偿范围不包括以下原因所导致的服务不可用：

- (1) 百度智能云预先通知用户后进行系统维护所引起的，包括合理升级、变更、停机、割接、维修和模拟故障演练；
- (2) 任何百度智能云所属设备以外的网络、设备故障或配置调整引起的；
- (3) 客户的应用程序受到黑客攻击而引起的；
- (4) 客户维护不当或保密不当致使数据、口令、密码等丢失或泄漏所引起的；
- (5) 客户的疏忽或由客户授权的操作所引起的；
- (6) 客户未遵循百度智能云产品使用文档或使用建议引起的，如客户在控制台、API或者CLI等控制方式对CFS实例进行操作引起的不可用；
- (7) 由于客户所安装软件或者其他非百度智能云直接运营的第三方软件或者配置引起的错误；
- (8) 不可抗力以及意外事件引起的；
- (9) 其他非百度智能云原因所造成的不可用。

3. 赔偿方案

3.1 赔偿标准

对于单CFS实例，如服务可用性低于99.9%，可按照下表中的标准获得赔偿，赔偿方式仅限于用于购买CFS产品的代金券，且赔偿总额不超过未达到服务可用性承诺当月客户就该CFS实例支付的单实例月度服务费。

服务可用性	赔偿代金券金额
低于99.9%但等于或高于99%	该文件系统月度服务费的15%
低于99%但等于或高于95%	该文件系统月度服务费的30%
低于95%	该文件系统月度服务费的100%

3.2 赔偿申请时限

客户可以在每月第五（5）个工作日对上个月没有达到可用性的CFS实例提出赔偿申请。赔偿申请必须限于在CFS没有达到服务可用性的相关月份结束后两（2）个月内提出。超出申请时限的赔偿申请将不被受理。

4. 其他

- (1) 在法律法规允许的范围内，百度智能云负责对本协议进行解释说明。
- (2) 本协议一经公布立即生效，百度智能云有权对本SLA条款做出修改。如本SLA条款有任何修改，百度智能云将以网站公示或发送邮件的方式通知您。如您不同意百度智能云对SLA所做的修改，您有权停止使用对等连接服务，如您继续使用对等连接服务，则视为您接受修改后的SLA。
- (3) 本协议项下百度智能云CFS服务对于用户所有的通知均可通过网页公告、站内信、电子邮件、手机短信或其他形式等方式进行；该类通知于发送之日视为已送达收件人。
- (4) 本协议的订立、执行和解释及争议的解决均应适用中国法律并受中国法院管辖。如双方就本协议内容或其执行发生任何争议，双方应尽量友好协商解决；协商不成时，任何一方均可向北京市海淀区人民法院提起诉讼。

- (5) 本协议构成双方对本协议之约定事项及其他有关事宜的完整协议，除本协议规定的之外，未赋予本协议各方其他权利。
- (6) 如本协议中的任何协议无论因何种原因完全或部分无效或不具有执行力，本协议的其余协议仍应有效并且有约束力。
- (7) 关于用户约束条款，详见[百度智能云用户服务协议](#)中的"用户的权利与义务"相关条款内容。
- (8) 关于服务商免责条款，详见[百度智能云用户服务协议](#)中的"免责声明"相关条款内容。