

对象存储服务

用户指南

文档版本 01
发布日期 2024-10-24



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

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，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 使用前须知	1
1.1 OBS 帮助文档概览	1
1.2 使用方式	2
1.3 访问规则	3
2 存储类别	9
3 桶管理	16
3.1 桶概述	16
3.2 创建桶	17
3.3 列举桶	18
3.4 查看桶信息	19
3.5 管理桶配额	19
3.6 删除桶	20
3.7 复制桶配置信息	21
4 对象管理	22
4.1 对象概述	22
4.2 上传	25
4.2.1 上传对象	25
4.2.2 多段上传	30
4.3 下载	34
4.4 文件夹管理	34
4.4.1 创建文件夹	35
4.4.2 分享文件夹	35
4.5 其他对象管理操作	37
4.5.1 列举对象	37
4.5.2 复制对象	37
4.5.3 查看对象信息	38
4.5.4 分享对象	39
4.5.5 管理对象元数据	41
4.5.6 恢复归档存储或深度归档存储对象	51
4.5.7 直读归档存储数据	52
4.6 删除	52
4.6.1 删除对象	53

4.6.2 取消删除对象.....	54
4.6.3 清理碎片.....	55
5 权限控制.....	57
6 数据管理.....	58
6.1 生命周期管理.....	58
6.2 标签.....	67
6.3 桶清单.....	68
6.4 事件通知.....	71
6.5 用量统计.....	72
7 数据访问.....	73
7.1 静态网站托管.....	73
7.2 数据回源.....	80
7.3 域名管理.....	82
7.3.1 自定义域名配置.....	83
8 数据安全.....	85
8.1 服务端加密.....	85
8.2 WORM.....	86
8.3 跨域资源共享.....	87
8.4 多版本控制.....	89
8.5 跨区域复制.....	92
8.6 防盗链.....	96
9 数据处理.....	99
9.1 图片处理.....	99
9.2 在线解压.....	99
10 监控与日志.....	102
10.1 监控.....	102
10.2 审计.....	103
10.3 日志记录.....	104
11 并行文件系统.....	109

1 使用前须知

1.1 OBS 帮助文档概览

对象存储服务OBS提供了一系列文档，帮助您由浅入深地了解和使用对象存储服务OBS。

建议您在使用对象存储服务OBS前，先阅读本文档和其他相关帮助文档（见[表1-1](#)），了解对象存储服务OBS的基本概念、使用场景、使用方式等，从而帮助您更快速、更轻松地使用对象存储服务OBS。

表 1-1 对象存储服务 OBS 帮助文档

文档	描述
最新动态	对象存储服务OBS功能新增和文档优化的最新动态展示页面，帮助您第一时间了解对象存储服务OBS的最新资讯，新上线功能和文档体验快人一步。
功能总览	对象存储服务OBS功能汇总。您可以清晰地看到OBS支持的功能及功能发布的区域等。
产品介绍	对象存储服务OBS的系统性介绍文档，从服务的常用概念、产品优势、应用场景、计费说明、权限管理等多个方面进行介绍，让您初步了解对象存储服务OBS。
快速入门	对象存储服务OBS的快速使用指导，帮助您快速完成创建桶、上传对象和下载对象的操作。
控制台指南	对象存储服务OBS控制台的详细使用指导，提供了控制台各个功能特性的使用说明，同时提供应用举例帮助您更好地使用对象存储服务OBS控制台。
工具指南	对象存储服务OBS工具（OBS Browser+、obsutil、obsfs）的详细介绍和使用指导，同时提供应用举例帮助您更好地使用对象存储服务OBS工具。

文档	描述
图片处理特性指南	对象存储服务OBS图片处理特性的详细使用指导，从图片处理的基本概念、功能、常见问题等方面，指导您如何通过控制台和API使用图片处理功能。
并行文件系统特性指南	对象存储服务OBS并行文件系统特性的详细使用指导，从并行文件系统的基本概念、应用场景、约束限制、使用方式、计费标准等多个方面进行介绍，帮助您更好的使用并行文件系统。
最佳实践	对象存储服务OBS在各个典型场景下的应用实践，帮助您在不同的应用场景下以更优的性能、更低的成本、更便捷地操作使用对象存储服务OBS完成您的业务目标。
API参考	对象存储服务OBS提供的REST API的详细说明文档，详细介绍了各个API的请求示例和参数说明、响应示例和参数说明等信息，帮助您顺利使用API完成二次开发。
SDK参考	对象存储服务OBS软件开发工具包（OBS SDK，Object Storage Service Software Development Kit）的详细使用指导，提供Java、Python、C、GO、Android、IOS等多种主流语言的安装、开发、参数、示例和常见问题处理等内容，帮助您顺利使用SDK完成二次开发。
常见问题	对象存储服务OBS使用过程中的常见问题分类汇总，解答您使用对象存储服务OBS过程中的各种疑问。
视频帮助	对象存储服务OBS介绍帮助视频，以比文字更加生动的方式向您介绍对象存储服务OBS。

1.2 使用方式

您可以通过以下方式对OBS资源进行管理。

工具	描述	使用方式	参考文档
管理控制台	管理控制台是网页形式的。通过管理控制台，您可以使用直观的界面进行相应的操作。	使用网页访问云服务控制台，通过账号名&密码或IAM的账号名&用户名&密码登录认证身份，即可访问OBS的全部资源。 详情请参见 登录OBS管理控制台 。	控制台指南
SDK	SDK是对OBS服务提供的REST API进行的封装，以简化用户的开发工作。用户直接调用SDK提供的接口函数即可实现使用OBS业务能力的目的。	环境准备时配置终端节点，并在初始化阶段设置访问密钥（AK/SK）。 详情请参见 SDK参考 。	SDK参考

工具	描述	使用方式	参考文档
API	OBS提供了REST（Representational State Transfer）风格API，支持您通过HTTP/HTTPS请求调用，实现创建、修改、删除桶，上传、下载、删除对象等操作。	环境准备时配置终端节点，计算签名时添加访问密钥（AK/SK）到请求中。如果您使用REST API自行开发程序，则需按照OBS定义的签名算法来计算签名并添加到请求中。 详情请参见 用户签名验证 。	API参考
OBS Browser+	OBS Browser+是一款运行在Windows系统上的对象存储服务管理工具，OBS Browser+的图形化界面可以非常方便地让用户在本地对OBS进行管理。	下载OBS Browser+工具，通过访问密钥（AK/SK）进行身份鉴权。OBS Browser+支持直接访问OBS，也支持通过配置服务器地址方式访问。 OBS Browser+的AK、SK和终端节点的配置详情请参见 在哪里可以获取AK和SK 。	OBS Browser+工具指南
obsutil	obsutil是一款用于访问管理OBS的命令行工具，您可以使用该工具对OBS进行常用的配置管理操作。对于熟悉命令程序的用户，obsutil是执行批量处理、自动化任务的最佳选择。	下载obsutil工具，配置服务器地址，通过访问密钥（AK/SK）进行身份鉴权。 obsutil的AK、SK和终端节点的配置详情请参见 初始化配置 。	obsutil工具指南
obsfs	obsfs是OBS提供的一款基于FUSE的文件系统工具，主要用于将并行文件系统挂载至Linux系统，让用户能够在本地像操作文件系统一样直接使用OBS海量的存储空间。	下载obsfs工具，通过访问密钥（AK/SK）进行身份鉴权。 obsfs的AK、SK的配置详情请参见 初始化配置 。	obsfs工具指南

1.3 访问规则

OBS 域名

在介绍OBS域名之前，您需要先了解两个概念：

- 终端节点（Endpoint）**：OBS为每个区域提供一个终端节点，终端节点可以理解为OBS在不同区域的区域域名，用于处理各自区域的访问请求。各区域的终端节点详情请参见[地区和终端节点](#)。
- 访问域名**：OBS会为每一个桶分配默认的访问域名。访问域名是桶在互联网中的域名地址，可应用于直接通过域名访问桶的场景，比如：云应用开发、数据分享等。

OBS桶访问域名的结构为：**BucketName.Endpoint**

其中**BucketName**为桶名称，**Endpoint**为桶所在区域的终端节点（区域域名）。

除了桶访问域名外，[表1-2](#)列出了与OBS相关的其他域名的结构、协议类型等信息，以便您全面地了解OBS域名。

表 1-2 OBS 域名组成规则

域名类型	域名结构	说明	协议类型
区域域名	【结构】 Endpoint 【示例】 obs.ap-southeast-1.myhuaweicloud.com	不同的区域分配各自对应的域名，即各区域的终端节点。各区域的终端节点详情请参见 地区和终端节点 。	HTTPS HTTP
桶访问域名	【结构】 BucketName.Endpoint 【示例】 bucketname.obs.ap-southeast-1.myhuaweicloud.com	桶创建成功后，可以使用桶访问域名来访问桶。您可以根据访问域名结构自行拼接，也可以通过在OBS管理控制台、OBS Browser+上 查看桶基本信息 获取。	HTTPS HTTP
对象访问域名	【结构】 BucketName.Endpoint/ObjectName 【示例】 bucketname.obs.ap-southeast-1.myhuaweicloud.com/object.txt	对象上传到桶中后，可以使用对象访问域名来访问桶中的指定对象。您可以根据访问域名结构自行拼接，也可以通过在OBS管理控制台、OBS Browser+上 查看对象属性 获取，或在SDK上通过调用GetObjectUrl接口获取。	HTTPS HTTP
静态网站访问域名	【结构】 BucketName.obs-website.Endpoint 【示例】 bucketname.obs-website.ap-southeast-1.myhuaweicloud.com	桶配置为静态网站托管时，桶的静态网站访问域名。	HTTPS HTTP
自定义域名	已通过工信部备案的域名	你可以为桶配置用户自定义的域名，通过用户自定义的域名访问桶。	HTTP

Endpoint 使用说明

OBS在不同区域提供不同的终端节点，在使用的时候应该如何选择呢？

一般情况下，向OBS资源发起请求时使用的Endpoint必须为被请求资源所在区域的Endpoint；但也存在某些特殊情况可以不用遵守此要求，可以使用任意区域的Endpoint。

1. 请求中Endpoint必须使用被请求资源所在区域Endpoint的场景

在访问桶和对象时，请求中Endpoint必须使用桶和对象所在区域的Endpoint。

例如桶mybucket所属区域是ap-southeast-1，列举桶mybucket中的对象的示例如下：

列举桶内对象正确示例：

【请求消息】

```
GET / HTTP/1.1
Host: mybucket.obs.ap-southeast-1.myhuaweicloud.com
Accept: */*
Date: Thu, 10 Mar 2016 08:51:25 GMT
Authorization: authorization
```

【响应消息】

```
HTTP/1.1 200 OK
x-obs-request-id: 0001EF710C000001536176DA465E4E6G
x-obs-id-2: Rdj0zZvRkihRcjcQUqjkDGt8JuAgi2CGuLiP7Pv/cYYplsS0xTFJQHP5vSg5yOYC
Content-Type: application/xml
Date: Thu, 10 Mar 2016 16:58:12 GMT
x-obs-bucket-location: ap-southeast-1
Content-Length: 259
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListBucketResult xmlns="http://obs.myhuaweicloud.com/doc/2015-06-30/">
  <Name>mybucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>object001</Key>
    <LastModified>2015-07-01T00:32:16.482Z</LastModified>
    <ETag>"2fa3bcaaec668adc5da177e67a122d7c"</ETag>
    <Size>12041</Size>
    <Owner>
      <ID>b4bf1b36d9ca43d984fbc9491b6fce9</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```

如果使用的Endpoint和桶所属区域对应的Endpoint不一致，会提示桶不存在。

上述例子中桶mybucket所属区域是ap-southeast-1，如果使用cn-south-1的Endpoint（mybucket.obs.cn-south-1.myhuaweicloud.com）访问桶，则会收到HTTP 404状态码，提示桶不存在。此时可以使用[获取桶区域位置](#)接口获取到桶所属区域，再次发起请求。

列举桶内对象错误示例：

【请求消息】

```
GET / HTTP/1.1
Host: mybucket.obs.cn-south-1.myhuaweicloud.com
Accept: */*
Date: Thu, 10 Mar 2016 08:51:25 GMT
Authorization: authorization
```

【响应消息】

```
HTTP/1.1 404 NoSuchBucket
x-obs-request-id: 0001EF710C000001536176DA465E4E6H
x-obs-id-2: Rdj0zZvRkihRcjcQUqjkDGt8JuAgi2CGuLiP7Pv/cYYplsS0xTFJQHP5vSg5yOYL
Date: Thu, 10 Mar 2016 08:51:30 GMT
Content-Length: 0
```

2. 请求中Endpoint可以使用任意区域Endpoint的场景

因为“获取桶列表”接口获取的是指定账号或用户下所有区域的OBS桶，“获取桶区域位置”接口是在所有区域中检索您请求的桶，所以对于这两个接口而言，请求中Endpoint可以使用任意区域的Endpoint。

获取桶区域位置示例：

例如桶mybucket位于ap-southeast-1区域，请求中使用cn-south-1区域的Endpoint，也能成功获取到其位置信息。

【请求消息】

```
GET /?location HTTP/1.1
Host: mybucket.obs.cn-south-1.myhuaweicloud.com
Accept: */*
Date: Thu, 10 Mar 2016 08:51:25 GMT
Authorization: authorization
```

【响应消息】

```
HTTP/1.1 200 OK
x-obs-request-id: 0001EF710C000001536176DA465E4E6G
x-obs-id-2: Rdj0zZvRkihRcjcQUqjkDGt8JuAgi2CGuLiP7Pv/cYypIsS0xTFJQHP5vSg5yOYC
Content-Type: application/xml
Date: Thu, 10 Mar 2016 16:58:12 GMT
Content-Length: length

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Location xmlns="http://obs.myhuaweicloud.com/doc/2015-06-30/">ap-southeast-1</Location>
```

通过公网访问 OBS

通过公网访问OBS产生的流入流量（数据上传到OBS等写操作）是免费的，流出流量（下载OBS数据等读操作）是收费的。

公网流出流量的收费标准请参见[产品价格详情](#)。

通过公网访问OBS可以直接通过URL的形式来表示OBS中的资源。OBS的URL构成如下：

Protocol://[BucketName.]Endpoint[:Port]/[Object][?Param]

表 1-3 OBS URL 中的参数

参数	描述	是否必选
Protocol	请求使用的协议类型，如HTTP、HTTPS。HTTPS表示通过安全的HTTPS访问该资源，对象存储服务支持HTTP，HTTPS两种传输协议。	必选
BucketName	请求使用的桶名称，在整个OBS中唯一标识一个桶。	可选
Endpoint	OBS中桶所在区域的区域域名（终端节点）。各区域域名详情请参见 地区和终端节点 。	必选
Port	请求使用的端口号。根据软件服务器的部署不同而不同。缺省时使用默认端口，各种传输协议都有默认的端口号，如HTTP的默认端口为80，HTTPS的默认端口为443。 OBS对象存储服务的HTTP方式访问端口为80，HTTPS方式访问端口为443。	可选

参数	描述	是否必选
Object	请求使用的对象资源路径，即对象上传到OBS后的访问路径。	可选
Param	请求使用的桶和对象的具体资源，缺省默认为请求桶或对象自身资源。	可选

示例：您在中国-香港（**ap-southeast-1**）区域有一个名为**mybucket**的桶，桶中有一个对象**myfolder/myfile.txt**，通过公网访问这个对象的URL地址为：

<https://mybucket.obs.ap-southeast-1.myhuaweicloud.com/myfolder/myfile.txt>

📖 说明

除“获取桶列表”之外的所有接口，都应当包含桶名BucketName。OBS基于DNS解析性能和可靠性的考虑，要求凡是携带桶名的请求，在构造URL的时候都必须将桶名放在Endpoint前面，形成三级域名形式，又称为虚拟主机访问域名。

通过内网访问 OBS

通过内网访问OBS主要指通过华为云服务之间的内部通信网络访问OBS。通过内网访问OBS产生的流入流量（数据上传到OBS等写操作）和流出流量（下载OBS数据等读操作）都是免费的。

例如在弹性云服务器（Elastic Cloud Server，ECS）上通过内网访问OBS，可以避免公网网络质量带来的不利影响，达到优化性能、节省开支的目的。

OBS向您提供了相关最佳实践的配置方法，详情请参见[在ECS上通过内网访问OBS](#)。

OBS 2.0 和 OBS 3.0 的判断方法

OBS存在两代架构，我们将他们称之为OBS 2.0 和OBS 3.0。新创建桶时桶默认会存储到OBS 3.0，桶的版本为OBS 3.0，但以前创建的桶仍存储在OBS 2.0。

OBS的基础特性和功能在OBS 3.0和OBS 2.0都支持。但一些新特性仅在OBS 3.0支持，比如图片处理、跨域区域复制等。

您可以通过OBS Console或Head Bucket接口来确认桶存储在OBS 2.0还是OBS 3.0，具体方法如下：

方法一：登录OBS Console，查看桶概览

如果“桶版本号”的值是“3.0”则说明桶存储在OBS 3.0，否则桶存储在OBS 2.0。

方法二：通过Head Bucket接口查看桶的版本号

请求消息示例

```
HEAD / HTTP/1.1
Host: bucketname.obs.ap-southeast-1.myhuaweicloud.com
Accept: */*
Date: WED, 01 Jul 2015 02:23:25 GMT
Authorization: auth string
```

响应消息示例

```
HTTP/1.1 200 OK
Server: OBS
x-obs-request-id: BF2600000163D80E4C5F20FDD5BD0085
Content-Type: application/xml
x-obs-version: 3.0
x-obs-id-2: 32AAAQAAEAABAAAQAAEAABAAAQAAEAABCS8wS9l00ll4oMWmdniV7XmdAvfewrQq
Date: WED, 01 Jul 2015 02:23:25 GMT
Content-Length: 0
```

响应消息里面x-obs-version: 3.0表示桶存储在OBS 3.0, 没有该头域或者该头域的值是其他值表示桶存储在OBS 2.0。

关于Head Bucket接口的更多信息请参见[获取桶元数据API](#)。

2 存储类别

使用场景

OBS支持4种存储类别：标准存储、低频访问存储、归档存储、深度归档存储（受限公测），从而满足客户业务对存储性能、成本的不同诉求。不同规格的存储类别计费参见[存储费用](#)。

不同的存储类别可以满足客户业务对存储性能、成本的不同诉求。

- 标准存储访问时延低和吞吐量高，因而适用于有大量热点文件（平均一个月多次）或小文件（小于1MB），且需要频繁访问数据的业务场景，例如：大数据、移动应用、热点视频、社交图片等场景。
- 低频访问存储适用于不频繁访问（平均一年少于12次）但在需要时也要求快速访问数据的业务场景，例如：文件同步/共享、企业备份等场景。
- 归档存储适用于很少访问（平均一年访问一次）数据的业务场景，例如：数据归档、长期备份等场景。归档存储安全、持久且成本极低，可以用来替代磁带库。为了保持成本低廉，数据恢复时间可能长达数分钟到数小时不等。
- 深度归档存储（受限公测）适用于长期不访问（平均几年访问一次）数据的业务场景，其成本相比归档存储更低，但相应的数据恢复时间将更长，一般为数小时。

存储类别对比

对比项目	标准存储	低频访问存储	归档存储	深度归档存储（受限公测）
特点	高性能、高可靠、高可用的对象存储服务	高可靠、较低成本的实时访问存储服务	归档数据的长期存储，存储单价更优惠	深度归档数据的长期存储，存储单价相比归档存储更优惠
应用场景	云应用、数据分享、内容分享、热点对象	网盘应用、企业备份、活跃归档、监控数据	档案数据、医疗影像、视频素材、带库替代	长期不访问的数据存档场景

对比项目	标准存储	低频访问存储	归档存储	深度归档存储 (受限公测)
设计持久性	99.999999999 %	99.9999999 99%	99.99999999 9%	99.999999999 %
设计持久性(多AZ)	99.999999999 9%	99.9999999 999%	不支持多AZ	不支持多AZ
设计可用性	99.99%	99%	99%	99%
设计可用性(多AZ)	99.995%	99.5%	不支持多AZ	不支持多AZ
最小计量单位 ^a	64KB	64KB	64KB	64KB
最低存储时间 ^b	无	30天	90天	180天
数据恢复	不涉及	按实际恢复数据量收费, 单位GB	分加急和标准两种恢复方式, 不同恢复方式消耗时间不同, 请参见 恢复归档存储或深度归档存储对象 。 按实际恢复数据量收费, 单位GB	分加急和标准两种恢复方式, 不同恢复方式消耗时间不同, 请参见 恢复归档存储或深度归档存储对象 。 按实际恢复数据量收费, 单位GB
图片处理	支持	支持	不支持	不支持

📖 说明

a: 最小计量单位是指对象的计费大小下限。对象大小小于最小计量单位时, 将按照最小计量单位进行计费。例如上传一个32KB的标准存储对象, 会按照64KB进行计费。

b: 最低存储时间是指对象的计费时间下限。对象存储时间小于最低存储时间时, 将按照最低存储时间计费。例如, 一个低频访问存储对象在OBS中存储了20天后删除, 会按照30天计费。

桶存储类别和对象存储类别

创建桶时支持设置桶存储类别, 也可在桶创建完成后修改桶存储类别。

上传对象时, 对象的存储类别默认继承桶的存储类别, 您也可以重新指定对象的存储类别, 同时也支持在对象上传完成后修改对象的存储类别。

修改桶的存储类别, 桶内已有对象的存储类别不会修改, 新上传对象时的默认对象存储类别随之修改。

📖 说明

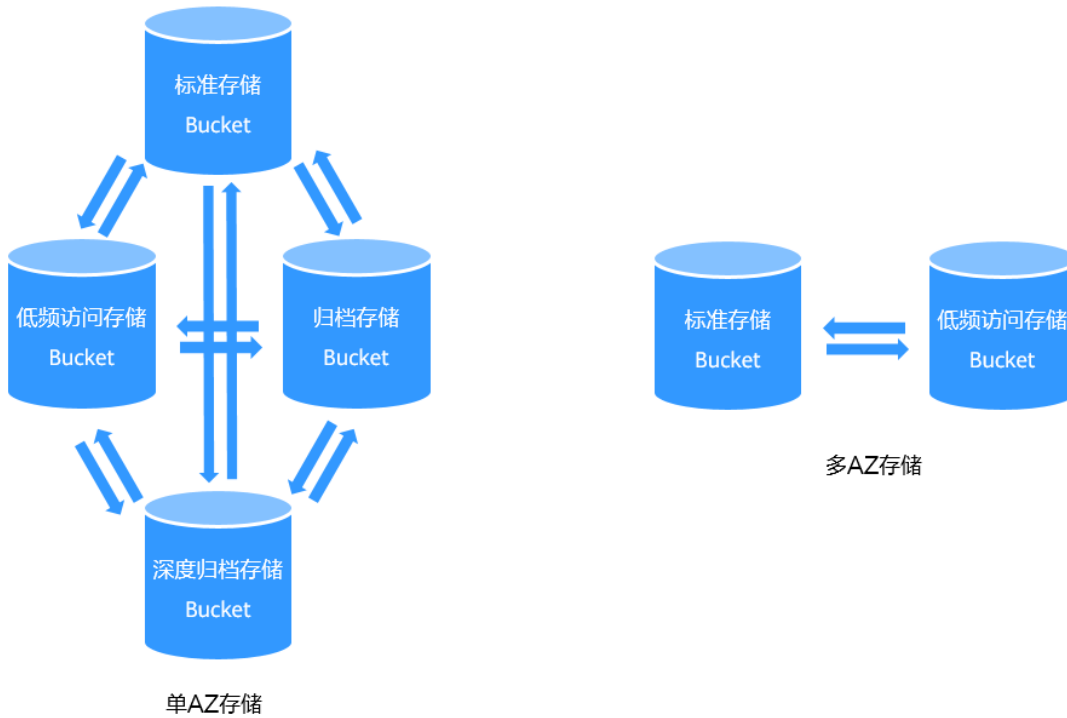
归档存储的对象, 恢复状态默认为未恢复, 此状态下无法直接下载。

桶存储类别转换

桶存储类别仅支持手动修改。

桶存储类别转换规则如图2-1所示。标准存储和低频访问存储支持单AZ存储和多AZ存储两种数据冗余存储策略，归档存储和深度归档存储仅支持单AZ存储。当一个桶的数据冗余存储策略为单AZ存储时，仅支持变更为单AZ存储的存储类别。多AZ存储同理。

图 2-1 桶存储类别转换规则



手动修改桶存储类别操作步骤：

1. 在OBS管理控制台左侧导航栏选择“对象存储”。
2. 在桶列表中，选择待修改存储类别的桶，并单击右侧操作栏的“修改存储类别”。
3. 选择存储类别，单击“确定”。

图 2-2 手动修改桶存储类别



注意事项：

- 修改桶的存储类别，不会影响桶中已有对象的存储类别，后续上传的对象默认与桶的存储类别相同。如果想批量修改桶中对象的存储类别，建议通过[生命周期规则](#)配置。
例：桶bucket1为标准存储，桶中有一个标准存储的对象object1。当桶bucket1转换为低频访问存储类别，object1依然为标准存储类别，新上传的object2则为低频访问存储类别。
- 桶的存储类别由归档存储或深度归档存储转换为标准存储或低频访问存储，不会自动恢复桶中存储类别为归档存储或深度归档存储的对象。
- 归档和深度归档存储不支持多AZ。因此不支持使用生命周期的存储类别转换功能，将多AZ桶的存储类别转化为归档或深度归档存储。

对象存储类别转换

对象存储类别支持手动修改和自动转换。对象存储类别转换规则如所示。

- 手动修改：上传对象后，手动修改对象的存储类别：
 - 标准存储的对象可以手动修改为低频访问存储、归档存储或深度归档存储的对象。
 - 低频访问存储的对象可以手动修改为标准存储、归档存储或深度归档存储的对象。
 - 归档存储的对象要先恢复，才可以手动修改为标准存储、低频访问存储和深度归档存储的对象。
 - 深度归档存储的对象要先恢复，才可以手动修改为标准存储、低频访问存储或归档存储的对象。

说明

对低频访问存储的对象、归档存储或深度归档存储的对象进行转换，会涉及恢复费用，请根据对象的使用频率和场景择优选择。

手动修改对象存储类别操作步骤：

- a. 在OBS[管理控制台](#)左侧导航栏选择“对象存储”。
- b. 在桶列表单击待操作的桶，进入“对象”页面。
- c. 修改对象存储类别支持单个修改和批量修改。
 - i. 单个修改：在对象列表中，选择待修改存储类别的对象，并单击右侧操作栏的“更多>修改存储类别”如[图2-3](#)所示。

图 2-3 单个修改对象存储类别



- ii. 批量修改：在对象列表中，勾选待修改存储类别的对象，并单击上方的“更多>修改存储类别”按钮，如[图2-4](#)所示。

说明

批量修改对象存储类别当前部分区域支持，请以管理控制台实际情况为准。

图 2-4 批量修改对象存储类别



d. 选择存储类别，单击“确定”，如图2-5所示。

图 2-5 选择对象存储类别

修改存储类别

已选择的对象：1个。 [查看](#)

存储类别 标准存储 低频访问存储 归档存储

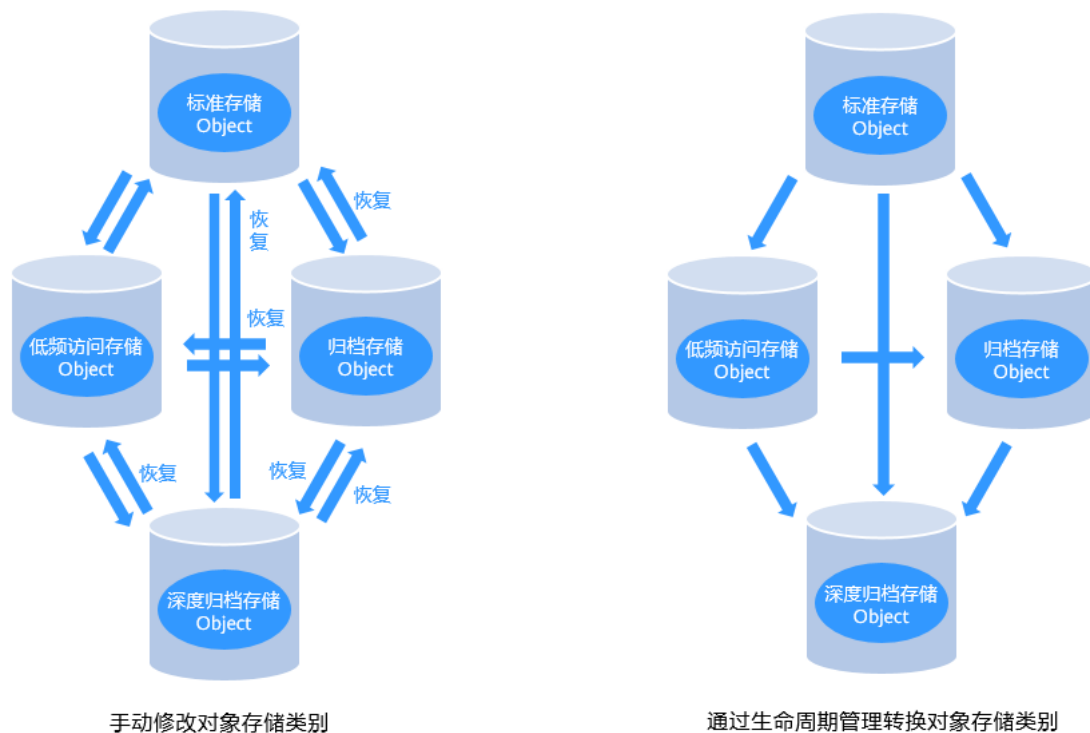
适用于有大量热点文件或小文件，且需要频繁访问（平均一个月多次）并快速获取数据的业务场景。

确定

取消

- 自动转换：通过OBS提供的生命周期功能实现存储类别自动转换，以降低存储成本。通过配置生命周期规则转换对象的存储类别，详情请参见[生命周期管理](#)。标准存储的对象可以自动转换为低频访问存储、归档存储或深度归档存储的对象。低频访问存储的对象可以自动转换为归档存储或深度归档存储的对象。归档存储的对象可以自动转换为深度归档存储的对象。

图 2-6 对象存储类别转换规则



说明

归档和深度归档存储对象手动进行存储类别转换时，需要先恢复对象，数据恢复后，才能更改对象存储类别。

注意事项：

- 标准存储的对象转换为低频访问存储或归档存储的对象，恢复状态为未恢复。
- 低频访问存储的最低存储时间为30天，归档存储的最低存储时间为90天，深度归档存储的最低存储时间为180天。如果对象经过规格转换，存储时间少于最低存储时间，需要补足剩余天数的存储费用。
- 归档和深度归档存储不支持多AZ。因此不支持使用生命周期的存储类别转换功能，将多AZ对象的存储类别转化为归档或深度归档存储。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式设置桶和对象的存储类别。

支持的使用方式	参考文档
控制台	创建桶时指定桶存储类别 上传文件时指定对象存储类别
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。

支持的使用方式	参考文档
API	设置桶默认存储类别 创建桶时指定桶存储类别 （添加x-obs-storage-class头域设置） PUT上传时指定对象存储类别 （添加x-obs-storage-class头域设置） POST上传时指定对象存储类别 （添加x-obs-storage-class头域设置）
OBS Browser+	-
obsutil	创建桶时指定桶存储类别 设置桶属性指定桶存储类别 上传对象时指定对象存储类别 设置对象属性指定对象存储类别

3 桶管理

3.1 桶概述

桶（Bucket）是OBS中存储**对象**的容器。对象存储提供了基于桶和对象的扁平化存储方式，桶中的所有对象都处于同一逻辑层级，去除了文件系统中的多层级树形目录结构。

每个桶都有自己的**存储类别**、访问权限、**所属区域**等属性，用户可以在不同区域创建不同存储类别和访问权限的桶，并配置更多高级属性来满足不同场景的存储诉求。

对象存储服务设置有四类存储类别，分别为：标准存储、低频访问存储、归档存储、深度归档存储（受限公测），从而满足客户业务对存储性能、成本的不同诉求。创建桶时可以指定桶的存储类别，桶的存储类别可以修改。

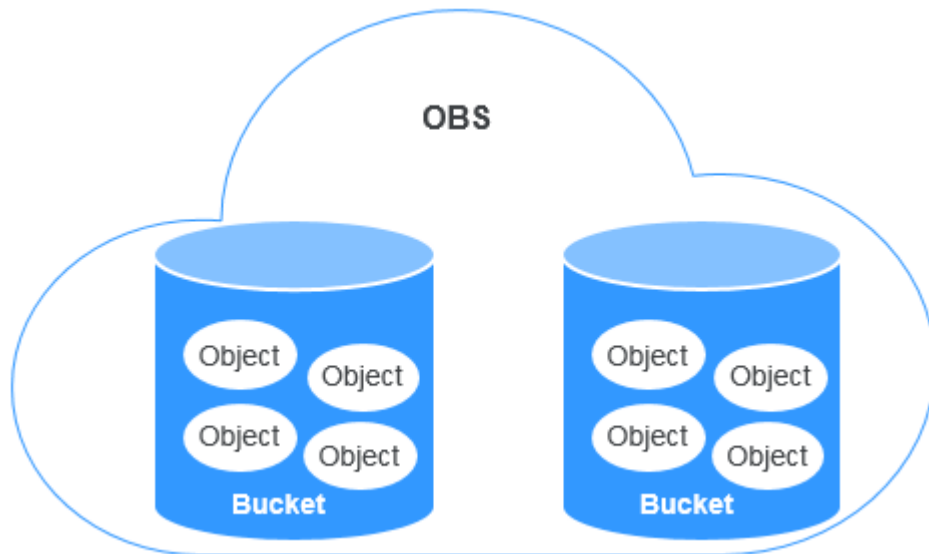
在OBS中，桶名必须是全局唯一的且不能修改，即用户创建的桶不能与自己已创建的其他桶名称相同，也不能与其他用户创建的桶名称相同。桶所属的区域在创建后也不能修改。每个桶在创建时都会生成默认的桶ACL（Access Control List），桶ACL列表的每项包含了对被授权用户授予什么样的权限，如读取权限、写入权限等。用户只有对桶有相应的权限，才可以对桶进行操作，如创建、删除、显示、设置桶ACL等。

一个账号及账号下的所有IAM用户一共可创建100个桶。建议结合OBS细粒度权限控制能力，合理进行桶规划和使用。例如，建议在桶内根据对象前缀划分不同的目录，通过**细粒度权限控制**实现不同目录在不同业务部门之间的权限隔离。每个桶中存放的对象的数量和大小总和没有限制，用户不需要考虑数据的可扩展性。

由于OBS是基于REST风格HTTP和HTTPS协议的服务，您可以通过URL（Uniform Resource Locator）来定位资源。

OBS中桶和对象的关系如**图3-1**所示：

图 3-1 桶和对象



您可以根据使用习惯、业务场景通过不同的[使用方式](#)来[创建桶](#)。桶创建成功后，您可以通过多种使用方式将[文件（数据）上传至桶](#)，OBS最终将这些文件以对象的形式存储在桶中。OBS中桶和对象按[区域](#)划分，无论您通过哪种方式使用OBS，都可以通过其他方式在同一区域访问相同的桶及桶内资源。

3.2 创建桶

使用场景

桶是OBS中存储对象的容器。创建桶之后，然后才能在OBS中存储数据。

前提条件

创建桶之前需要进行一些准备工作，如注册账号，充值，获取访问密钥（AK/SK）和终端节点（Endpoint）等，详情请参见[快速入门](#)。

约束与限制

- 桶创建成功后，不能修改名称和所属区域。创建时，请设置合适的桶名和区域。
- 一个账号及账号下的所有IAM用户一共可创建100个桶。建议结合OBS细粒度权限控制能力，合理进行桶规划和使用。例如，建议在桶内根据对象前缀划分不同的目录，通过[细粒度权限控制](#)实现不同目录在不同业务部门之间的权限隔离。每个桶中存放的对象的数量和大小总和没有限制，用户不需要考虑数据的可扩展性。
- 考虑到桶名会作为访问域名的一部分，需要参与域名解析，因此桶名需要满足[DNS域名规范](#)。OBS系统在接收创桶请求时，会对桶名进行严格的检查，具体规则如下：
 - 需全局唯一，不能与已有的任何桶名称重复，包括其他用户创建的桶。用户删除桶后，立即创建同名桶或并行文件系统会创建失败，需要等待30分钟才能创建。
 - 长度范围为3到63个字符，支持小写字母、数字、中划线（-）、英文句号（.）。

- 禁止两个英文句号 (.) 相邻，禁止英文句号 (.) 和中划线 (-) 相邻，禁止以英文句号 (.) 和中划线 (-) 开头或结尾。
- 禁止使用IP地址。

须知

当使用HTTPS协议访问OBS系统时，由于SSL的通配符证书仅匹配不包含"."的桶。这将导致桶名包含"."的桶在访问OBS系统时，客户端会提示证书校验存在风险，比如浏览器安全提示会呈现红色告警。因此如非必要，请尽量不要在桶名中包含"."。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式创建桶。

支持的使用方式	参考文档
控制台	创建桶
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	创建桶
OBS Browser+	创建桶
obsutil	创建桶

如何访问已创建的桶

桶创建成功后，可以使用桶访问域名来访问桶。您可以根据访问域名结构自行拼接，也可以通过在OBS管理控制台、OBS Browser+上[查看桶基本信息](#)获取。

访问域名的结构和示例如下：

【结构】 **BucketName.Endpoint**

【示例】 bucketname.obs.ap-southeast-1.myhuaweicloud.com

创建桶失败的原因和解决方法

请参见[创建桶失败](#)处理。

3.3 列举桶

使用场景

您可以通过列举桶查看已创建的桶列表。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式列举桶。

支持的使用方式	参考文档
控制台	登录控制台并选择“对象存储”，桶列表页将展示账户下所有的桶。
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	列举桶
OBS Browser+	登录OBS Browser+，桶列表页将展示账户下所有的桶。
obsutil	列举桶

3.4 查看桶信息

使用场景

创建桶之后，您可以通过OBS管理控制台直接查看某个桶的详情，包括基本信息上方的桶信息、桶的基本信息、用量分析、监报告警、常见场景操作指引、域名信息、基础配置、常见问题和了解更多模块。也可以将桶列表导出，在本地Excel查看账号下所有桶的基本信息。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil查看桶基本信息。

支持的使用方式	参考文档
控制台	查看桶的信息
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	获取桶元数据
OBS Browser+	查看桶的基本信息
obsutil	查询桶属性

3.5 管理桶配额

使用场景

默认情况下，OBS系统和单个桶都没有总数据容量和对象数量的限制。您可以通过为桶设置配额限制来控制桶内允许上传的对象总容量，超过设置的对象容量后，上传对象会失败。

桶配额只对设置生效后的对象上传操作有限制，如果设置的桶配额数值小于已经上传的对象容量，不会删除已有对象，但后续不能再上传任何对象。这种情况下只有删除部分已有对象，将已用空间释放到配额限制以下之后才能再次上传新对象。

📖 说明

- 桶配额值必须为非负整数，单位为字节，支持的最大值为 $2^{63} - 1$ 。
- OBS没有提供删除桶配额的接口，您可以将桶配额设置为0来取消配额限制。

使用方式

OBS支持通过API和SDK方式管理桶配额。

支持的使用方式	参考文档
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	获取桶配额

3.6 删除桶

使用场景

桶为空时，桶的拥有者可以根据需要删除桶，以免占用桶数量配额。

桶为空包含两方面含义：

- 桶内没有任何对象或对象的任何历史版本。
- 桶内没有任何未合并的多段上传任务，即桶内不存在碎片。

📖 说明

- 如果桶已开启多版本特性，请确保所有历史版本和删除标记（删除标记也视作一个历史版本）已删除。
- 删除桶后，需要等待30分钟才能创建同名桶。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式删除桶。

支持的使用方式	参考文档
控制台	删除桶
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	删除桶
OBS Browser+	删除桶

支持的使用方式	参考文档
obsutil	删除桶

删除桶失败的原因和解决方法

请参见[删除桶失败](#)处理。

3.7 复制桶配置信息

使用场景

OBS提供了桶配置信息复制功能，方便您在创建新桶之后，快速将已有桶的配置信息复制到新桶中。

支持复制的桶配置信息包括：

- 桶策略
- CORS规则
- 生命周期规则
- 数据回源规则
- 图片处理样式
- 在线解压规则

约束与限制

- 从源桶复制配置信息的操作为增量复制，不会删除当前桶已存在的配置信息，与已存在的配置信息冲突的规则不会复制。
- 源桶和目标桶的桶版本号都必须是3.0。
- 对象存储桶和并行文件系统之间可以相互复制，前提是该功能同时在对象存储桶和并行文件系统都支持。

使用方式

OBS支持通过控制台复制桶配置信息。

支持的使用方式	参考文档
控制台	<ul style="list-style-type: none">• 复制桶策略• 复制CORS规则• 复制生命周期规则• 复制数据回源规则• 复制图片处理样式• 复制在线解压规则

4 对象管理

4.1 对象概述

对象（Object）是OBS中数据存储的基本单位，一个对象实际是一个文件的数据与其相关属性信息（元数据）的集合体。用户上传至OBS的数据都以对象的形式保存在桶中。

对象包括了Key，Metadata，Data三部分：

- Key：键值，即对象的名称，为经过UTF-8编码的长度大于0且不超过1024的字符序列。一个桶里的每个对象必须拥有唯一的对象键值。
- Metadata：元数据，即对象的描述信息，包括系统元数据和用户元数据，这些元数据以键值对（Key-Value）的形式被上传到OBS中。
 - 系统元数据由OBS自动产生，在处理对象数据时使用，包括Date，Content-length，Last-modify，Content-MD5等。
 - 用户元数据由用户在上传对象时指定，是用户自定义的对象描述信息。
- Data：数据，即文件的数据内容。

通常，我们将对象等同于文件来进行管理，但是由于OBS是一种对象存储服务，并没有文件系统中的文件和文件夹概念。为了使用户更方便进行管理数据，OBS提供了一种方式模拟文件夹。通过在对象的名称中增加“/”，例如“test/123.jpg”。此时，“test”就被模拟成了一个文件夹，“123.jpg”则模拟成“test”文件夹下的文件名了，而实际上，对象名称（Key）仍然是“test/123.jpg”。

上传对象时，可以指定对象的**存储类别**，如果不指定，默认与桶的存储类别一致。上传后，对象的存储类别可以修改。

您可以根据使用习惯、业务场景通过多种**使用方式**将**文件（数据）上传至桶**，OBS最终将这些文件以对象的形式存储在桶中。OBS中桶和对象按**区域**划分，无论您通过哪种方式使用OBS，都可以通过其他方式在同一区域访问相同的桶及桶内资源。

对象键命名指导原则

虽然您可以在对象键名称中使用任何UTF-8字符，但是按照以下规则命名有助于确保与其他应用程序的最大兼容性。每个应用程序对特殊字符的分析方式可能不同。以下指导原则有助于最大程度符合DNS、Web安全字符、XML分析器和其他API的要求。

以下字符集通常可安全地用于键名称：

字母数字字符（通常也被称为非保留字符）	[0-9a-zA-Z]
特殊字符（通常也被称为保留字符）	感叹号（“!”） 连字符（“-”） 下划线（“_”） 句点（“.”） 星号（“*”） 单引号（“'”） 左括号（“（”） 右括号（“）”）

以下是有效对象键名称的示例：

```
4my-organization  
my.great_photos-2014/jan/myvacation.jpg  
videos/2014/birthday/video1.wmv
```

对保留字符的百分号编码

如果一个保留字符在URI中具有特殊含义（称作“reserved purpose”），且URI中必须使用该字符用于其它目的，那么该字符必须百分号编码。OBS服务对非ASCII字符编码必须采用UTF-8编码，否则最终上传至OBS的对象名可能与预期不符。例如，保留字符“/”，如果将其用作URI的路径成分的分隔符，则是具有特殊含义的保留字符（在OBS中用来分割桶名和对象名）。如果该字符需要出现在URI一个路径成分的内部，则需要使用三个字符“%2F”或“%2f”代替原本的“/”。保留字符“ ”，需要将其编码为“%20”，例如字符串“abc d”，应该编码为“abc%20d”。

可能需要特殊处理的字符

键名中的以下字符可能需要另外进行代码处理：

- 和符号（“&”）
- 美元（“\$”）
- 分号（“;”）
- 冒号（“:”）
- 加号（“+”）：OBS服务会将请求URI中的“+”解码成空格，因此如果原始的对象键名称中包含“+”，需要编码成%2B之后再放入请求URI中。
- 空格：大量连续空格可能会在某些使用情形中丢失（特别是多个空格）。
- 等于（“=”）
- At符号（“@”）
- 逗号（“,”）
- 问号（“?”）
- ASCII字符范围00-1F十六进制（0-31十进制）和7F（127十进制）

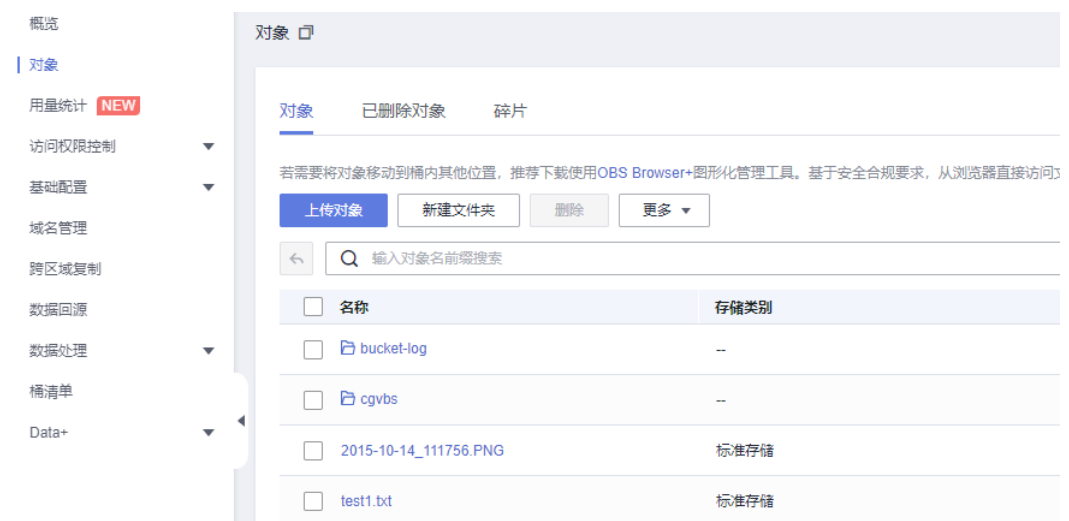
避免在键名中使用以下字符，因为这些字符需要进行大量的特殊处理，才能在所有应用程序间保持一致性。

- 反斜杠 (“\”)
- 左大括号 (“{”)
- 不可打印的ASCII字符 (128–255十进制字符)
- 插入符号 (“^”)
- 右大括号 (“}”)
- 百分比字符 (“%”)
- 重音符/反勾号 (“`”)
- 右方括号 (“]”)
- 引号
- “大于”符号 (“>”)
- 左方括号 (“[”)
- 波浪字符 (“~”)
- “小于”符号 (“<”)
- “井号”字符 (“#”)
- 竖线 (“|”)

请注意，OBS数据模型是一种扁平结构：您创建桶，桶存储对象。不存在子桶或子文件夹层次结构。但您可以使用键名称前缀和分隔符推断逻辑层次结构（如同OBS控制台一样），OBS控制台支持文件夹的概念。假设您的桶（companybucket）包含具有以下对象键的四个对象：

- bucket-log/log01.txt
- cgvbs/test1.txt
- 2015-10-14_111756.png
- test1.txt

控制台使用键名称前缀（bucket-log/和cgvbs/）和分隔符（“/”）呈现如下所示的文件夹结构：



2015-10-14_111756.png和test1.txt键没有前缀，因此其对象直接在桶的根级别出现。如果打开cgvbs/文件夹，则您会看到其中包含test1.txt对象。



假设您的桶（companybucket）包含以下对象键的两个对象：

- obj
- 1/../obj

当您直接访问API来获取对象（例如：使用SDK）则能正确获取，当使用console控制台来访问时，浏览器针对相对关系实际发送的请求会访问obj对象，因“../”在uri中特殊语义，请尽量避免出现在对象键中。

禁止创建以“..”开头的对象名。

4.2 上传

4.2.1 上传对象

使用场景

您可以根据需要上传文件或者文件夹至已有的OBS桶。上传文件时，如果您需要将文件归类处理，可以先新建文件夹，然后将相关的文件上传到文件夹中。

新建文件夹的详情请参见[创建文件夹](#)。

待上传的文件可以是任何类型：文本文件、图片、视频等。

须知

虽然OBS服务支持您将对象上传到指定region的桶里，但是华为云并不感知您对象的具体内容。如果您对象上传行为涉及跨境传输，请您确保使用本服务符合所适用的法律法规要求。

约束与限制

大小及数量限制：

- 通过OBS管理控制台上传的文件有大小和数量限制。

- 在部分支持批量上传的区域，每次最多支持100个文件同时上传，总大小不超过5GB。如果只上传1个文件，则这个文件最大为5GB。
- 在部分不支持批量上传的区域，每次只能上传1个文件，大小不超过50MB。
- 通过OBS Browser+和obsutil可以上传小于48.8TB的文件。OBS Browser+最多同时一次上传500个文件，obsutil没有上传文件数量限制。
- 通过SDK或API的PUT上传、POST上传和追加写，可以上传小于5GB的文件。
- 通过SDK或API的多段上传以及SDK的断点续传，可以上传小于48.8TB的文件。
- 您可以通过设置桶配额限制单个桶上传对象的总容量，详情请参见[管理桶配额](#)。

命名限制：

- 对象命名规则详见[对象键命名指导原则](#)。

批量操作限制：

- 支持批量上传功能需要满足以下条件：
 - a. OBS桶所在区域支持批量上传功能。当前支持的区域有：中国-香港、亚太-曼谷、亚太-新加坡。
 - b. OBS桶的版本号为“3.0”。桶版本号的查看方法请参见[OBS 2.0和OBS 3.0的判断方法](#)。

文件夹上传限制：

- 控制台、OBS Browser+、obsutil支持直接上传文件夹。

其他限制：

- 在未开启多版本控制功能的情况下，如果新上传的文件和桶内文件重名，则新上传的文件会自动覆盖老文件，且不会保留老文件的ACL等信息；如果新上传的文件夹和桶内文件夹重名，则上传后会将新老文件夹合并，合并过程如遇重名文件，会使用新上传的文件夹中的文件进行覆盖。
- 在开启了多版本控制功能的情况下，如果新上传的文件和桶内文件重名，则会在老文件上新增一个版本。关于多版本的详细介绍请参见[多版本控制](#)。

上传方式

上传方式	说明
PUT上传 (流式上传)	当上传文件小于5GB时使用。 两种上传方式的区别请参见 PUT上传和POST上传有什么区别?
POST上传 (表单上传)	
多段上传 (分段上传)	当上传文件大于5GB且小于48.8TB时使用。 详情请参见 多段上传 。 说明 如果您有大量的数据（大于48.8TB）需要上传至OBS，可以参考 搬迁本地数据至OBS 。

上传方式	说明
断点续传	<p>当上传大文件时，经常出现因网络不稳定或程序崩溃导致上传失败的情况。失败后再次重新上传不仅浪费资源，而且当网络不稳定时仍然有上传失败的风险。断点续传上传接口能有效地解决此类问题引起的上传失败，其原理是将待上传的文件分成若干个分段分别上传，不仅能在失败重传时节省资源提高效率，还因其对分段进行并发上传的机制能加快上传速度。</p> <p>各个使用方式支持的情况请参见OBS是否支持断点续传功能?</p>
增量同步上传	<p>将本地源路径下的所有内容同步到OBS指定目标桶，使两边内容保持一致。这里的增量同步有两层含义：1) 增量，依次比较源文件和目标对象，只上传存在变化的源文件；2) 同步，命令执行完成后，保证本地源路径是OBS指定目标桶的子集，即本地源路径下的所有文件均能在OBS指定目标桶中找到对应对象。</p> <p>详情请参见增量同步上传对象。</p>
追加写	<p>追加写对象操作是指在指定桶内的一个对象尾追加上传数据，不存在相同对象键值的对象则创建新对象。</p> <p>详情请参见追加写对象。</p>

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式上传对象。

支持的使用方式	参考文档
控制台	上传文件
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	对象操作
OBS Browser+	拖拽上传
obsutil	上传对象

相关操作

您可以在上传对象时指定对象的[存储类别](#)，也可以在上传后修改对象的存储类别。

- 标准存储的对象可以修改为低频访问存储、归档存储或深度归档存储的对象。低频访问存储的对象可以修改为标准存储、归档存储或深度归档存储的对象。归档存储的对象要先恢复，才可以修改为标准存储或低频访问存储。深度归档存储的对象要先恢复，才可以修改为标准存储、低频访问存储或归档存储。对低频访问存储的对象、归档存储的对象或深度归档存储的对象进行转换，会涉及恢复费用，请根据对象的使用频率和场景择优选择。

- 低频访问存储的最低存储时间为30天，归档存储的最低存储时间为90天，深度归档存储的最低存储时间为180天。如果对象经过转换，存储时间少于最低存储时间，需要补足剩余天数的存储费用。
- 您也可以通过配置生命周期规则转换对象的存储类别，详情请参见[生命周期管理](#)。

授权其他华为账号上传对象

本案例介绍如何为其他华为账号（包含账号下的IAM用户）授予OBS上传对象的权限。对其他华为账号授权，推荐使用桶策略。本示例以桶策略为例。

说明

在为其他华为账号下的IAM用户授予上传对象的权限之前，需要被授权IAM用户所属账号，为该IAM用户配置允许对指定桶执行指定操作的IAM权限，允许的操作要与桶策略相同，具体操作请参见[配置允许指定操作的IAM权限](#)。

如果为其他华为账号（Tenant Administrator，该角色用户拥有除IAM外其他所有服务的管理员权限）授权，可跳过该步骤。

1. 在OBS[管理控制台](#)左侧导航栏选择“对象存储”。
2. 在桶列表单击目标桶的桶名称，进入“对象”页面。
3. 在左侧导航栏，单击“访问权限控制>桶策略”，进入桶策略页面。
4. 在“桶策略”页面，单击“创建”。
5. 配置桶策略内容。

图 4-1 配置桶策略

创建桶策略

【创建桶】、【获取桶列表】两个服务级的操作权限，需要您通过统一身份认证进行配置。如何配置？

可视化视图 JSON视图

* 策略名称

* 效力 允许 拒绝

* 被授权用户 所有帐号 当前帐号 其他帐号

请输入帐号ID和IAM用户，ID格式：domainid/userid
可授权给多个IAM用户，每行一个

domainid"表示授权给帐号下的所有用户 如何查看【帐号ID】和【IAM用户ID】

[添加委托帐号](#)

* 授权资源 整个桶（包括桶内对象） 当前桶 指定对象

[添加资源路径](#)

* 授权操作 模板配置 自定义配置

授权条件（可选） 本规则生效的所需条件，以此限定规则的生效范围，通过键值表达式实现 [查看配置案例](#)

表 4-1 桶策略配置说明

参数		说明
策略配置方式		根据使用习惯进行选择，此处以“可视化视图”为例
策略名称		输入自定义的桶策略名称
策略内容	效力	允许
	被授权用户	<ul style="list-style-type: none">被授权用户：其他账号 说明 账号ID和IAM用户ID可在“我的凭证”页面获取。 输入格式：domainId/userId，可授权给多个账号，每行一个。 domainId/*表示授权给账号下的所有用户。
	授权资源	<ul style="list-style-type: none">资源范围：指定对象资源路径：输入对象前缀 说明 <ul style="list-style-type: none">支持输入多个资源路径，单击“添加资源路径”按钮即可。您可以指定资源路径为具体对象、对象集或目录，对象前缀为空表示当前桶，*表示桶内所有对象。 如果指定某个对象：对象名称。 如果指定某个对象集：“对象名称前缀” + “*”、“*” + “对象名后缀”或“*”。
授权操作	<ul style="list-style-type: none">动作范围：自定义配置选择动作：PutObject（上传对象） 说明 如果被授权账号（包含账号下的IAM用户）希望通过挂载外部桶的方式在OBS Browser+上访问此OBS桶，“授权资源”需要增加勾选“当前桶”，“授权操作”需要增加勾选自定义动作“ListBucket”。	

6. 核对权限配置信息，确认无误后单击“创建”，完成桶策略创建。
7. 配置成功后，被授权账号可以正常通过API或SDK完成上传操作，此外允许通过OBS Browser+挂载外部桶的方式完成上传操作，但还需要额外配置一条ListBucket的权限。暂不支持在OBS控制台访问非本账号的OBS桶。

配置ListBucket权限后，通过OBS Browser+访问添加的外部桶可能仍会出现无权限的相关提示信息。

报错原因：OBS Browser+桶详情页面的加载会调用一些其他的OBS接口，而授予的权限中并没有包含这些操作的权限，所以会提示“拒绝访问，请检查相应权限”，或者“不允许在请求的资源上执行此操作”，但并不影响已有权限。

上传对象失败的原因和解决方法

请参见[上传对象失败](#)处理。

4.2.2 多段上传

多段上传允许将单个对象作为一组段分别上传。每个段都是对象数据的连续部分。您可以独立上传以及按任意顺序上传这些对象的段。如果其中某个段传输失败，可以重新传输该段且不会影响其他段。当对象的所有段都上传后，OBS将这些段合并起来，然后创建对象。一般而言，如果您的对象大小达到了100MB，您应该考虑使用多段上传，而不是在单个操作中上传对象。例如：您有1个500MB大小的对象，需要上传到OBS桶中，您可以使用OBS Browser+工具进行多段上传，此工具会将该对象自动切分为多个小对象进行上传；或者使用OBS提供的多段上传API进行上传，最大程度提高上传效率同时避免您的任务失败。

使用多段上传可提供以下优势：

- 提高吞吐量：您可以并行上传段以提高吞吐量。
- 从任何网络问题中快速恢复：较小的段大小可以将由于网络错误而需重启失败的上传所产生的影响降至最低。
- 暂停和恢复对象上传：您可以随时上传对象段。启动多段上传后，不存在过期限；您必须显式地完成或取消多段上传任务。
- 在得知最终对象大小之前开始上传：您可以在创建对象的同时上传对象。

多段上传API允许您多段上传大型对象。您可以使用此API上传新的大型对象或创建现有对象的副本。

多段上传有三个步骤：开始上传（初始化上传段任务）、上传每个段，完成多段上传（合并段）。收到合并段请求后，OBS将构建由上传的段组成的对象，然后您可以像在您的桶中访问任何其他对象一样访问该对象。

您可以列出所有正在进行的多段上传任务，或获取特定多段上传任务已上传的段列表。以上每个操作都在本节中进行了说明。

初始化上传段任务

当您发送请求以开始多段上传时，OBS将返回具有上传ID的响应，此ID是多段上传的唯一标识符。无论您何时上传段、列举已上传的段、合并段或取消多段上传任务，您都必须包括此上传ID。

上传段

上传段时，除了指定上传ID，还必须指定段编号。您可以选择1和10000之间的任意段编号。段编号在您正在上传的对象中唯一地标示了段及其位置。如果您使用之前上传的段的同一段编号上传新段，则之前上传的段将被覆盖。无论您何时上传段，OBS都将在其响应中返回ETag标头。对于每个段上传任务，您必须记录每个段编号和ETag值。您需要在随后的请求中包括这些值以完成多段上传。

须知

初始化上传段任务并上传一个或多个段之后，您必须合并段或取消多段上传任务，才能停止收取上传的段的存储费用。仅当在合并段或取消多段上传任务之后，OBS才释放段存储并停止向您收取段存储费用。

多并发上传同一对象的同一段时，服务端遵循Last Write Win策略，但“Last Write”的时间定义为段元数据创建时间。为了保证数据准确性，客户端需要加锁保证同一对象的不同段并发上传不需要加锁。

拷贝段

多段上传任务创建后，用户可以通过指定多段上传任务号，为特定的任务上传段。添加段的方式还包括调用段拷贝接口。允许客户将已上传对象的一部分或全部拷贝为段。

须知

拷贝段的结果不能仅根据HTTP返回头域中的status_code来判断请求是否成功，头域中status_code返回200时表示服务端已经收到请求，且开始处理拷贝段请求。拷贝是否成功会在响应消息的body中，只有body体中有ETag标签才表示成功，否则表示拷贝失败。

将源对象object拷贝为一个段part1，如果在拷贝操作之前part1已经存在，拷贝操作执行之后老段数据part1会被新拷贝的段数据覆盖。拷贝成功后，只能列举到最新的段part1，老段数据将会被删除。因此在使用拷贝段接口时请确保目标段不存在或者已无价值，避免因拷贝段导致数据误删除。拷贝过程中源对象object无任何变化。

合并段和取消多段上传任务

合并段时，OBS通过按升序的段编号规范化多段来创建对象。如果在初始化上传段任务中提供了任何对象元数据，则OBS会将该元数据与对象相关联。成功完成请求后，段将不再存在。合并段请求必须包括上传ID以及段编号和相应的ETag值的列表。OBS响应包括可唯一地识别组合对象数据的ETag。此ETag无需成为对象数据的MD5哈希。您可以选择取消多段上传任务，取消多段上传任务之后无法再次使用该上传ID上传任何段。然后，OBS将释放被取消的多段上传任务中的每个段数据的所有存储。如果任何多段上传已在进行中，即使您已执行中止操作，它们仍可以上传成功或失败。如果要释放所有分段使用的所有存储，必须在完成所有多段上传后再取消多段上传任务。

须知

如果上传了10个段，但合并时只选择了9个段进行合并，那么未被合并的段将会被系统自动删除，未被合并的段删除后不能恢复。在进行合并之前请使用列出已上传的段接口进行查询，仔细核对所有段，确保没有段被遗漏。

列举已上传的段

您可以列出特定多段上传任务或所有正在进行的多段上传任务的分段。列举已上传的段操作将返回您已为特定多段上传任务而上传的段信息。对于每个列举已上传的段请求，OBS将返回有关特定多段上传任务的分段信息，最多为1000个分段。如果多段上传中的段超过1000个，您必须发送一系列列举已上传的段请求以检索所有段。请注意，返回的分段列表不包括已合并的分段。

须知

仅使用返回的列表进行验证。发送完成多段上传请求时，您不应使用此列表的结果。但是，当上传段和OBS返回的相应的ETag值时，将保留您自己的指定段编号的列表。

列举多段上传任务

通过列举桶中的多段上传任务，您可以获得已初始化多段上传任务的列表，已初始化多段上传任务是指初始化后还未合并以及未取消的多段上传任务。每个请求将返回最多1000个多段上传任务，如果正在进行的多段上传任务超过1000个，您需要发送其他请求以检索剩余的多段上传任务。

列举多段上传任务的限制如表4-2所示：

表 4-2 列举多段上传任务的限制

项目	限制
对象大小	最大48.8TB
每次上传的段的数量	最大10000
段编号	1到10000（含）
段大小	5MB到5GB，最后一个段大小是0到5GB
列举已上传的段请求返回的段数量	最大1000
列举已初始化多段任务请求返回的任务数量	最大1000

多段上传 API 和权限

用户必须具有使用多段上传操作的所需权限。您可以使用ACL、桶策略或用户策略来授予用户以执行这些操作的权限。下表列出了使用ACL、桶策略或用户策略时，各种多段上传操作的所需权限。

操作	所需权限
初始化上传段任务	您必须要有执行 <i>PutObject</i> 操作的权限，才能初始化上传段任务。 桶所有者可以允许其他委托人执行 <i>PutObject</i> 。
上传段	您必须要有执行 <i>PutObject</i> 操作的权限，才能上传段。 仅多段上传任务的发起者可以上传段。 桶所有者必须赋予多段任务发起者执行 <i>PutObject</i> 操作的权限，以便发起者可以上传该对象的段。

操作	所需权限
拷贝段	<p>您必须要有执行 <i>PutObject</i> 操作的权限，同时因为您正在将一个已存在的对象作为一个段来上传，因此您也必须要对拷贝源对象执行 <i>GetObject</i> 操作的权限才可以拷贝段。</p> <p>仅多段上传任务的发起者可以拷贝段。桶所有者必须赋予多段任务发起者对对象执行 <i>PutObject</i> 操作的权限，以便发起者可以上传该对象的段。</p>
合并段	<p>您必须要有执行 <i>PutObject</i> 操作的权限，才能进行合并段操作。</p> <p>仅多段上传任务的发起者可以合并段。桶所有者必须赋予多段任务发起者对对象执行 <i>PutObject</i> 操作的权限，以便发起者可以完成该对象的多段上传。</p>
取消多段上传任务	<p>您必须要有执行 <i>AbortMultipartUpload</i> 操作的权限，才能取消多段上传任务。</p> <p>默认情况下，允许桶所有者和多段上传任务的发起者执行此操作。除了这些默认情况之外，桶所有者可以允许其他委托人对对象执行 <i>AbortMultipartUpload</i> 操作。桶所有者可以拒绝任何委托人，使其无法执行 <i>AbortMultipartUpload</i> 操作。</p>
列举已上传的段	<p>您必须得到可以执行 <i>ListMultipartUploadParts</i> 操作的权限，才能在多段上传中列举已上传的段。</p> <p>在默认情况下，桶所有者有权为任何针对桶的多段上传列举已上传的段。多段上传任务的发起者有权为特定多段上传列举已上传的段。</p> <p>除了这些默认情况之外，桶所有者可以允许其他委托人对对象执行 <i>ListMultipartUploadParts</i> 操作。桶所有者也可以拒绝任何委托人，使其无法执行 <i>ListMultipartUploadParts</i> 操作。</p>
列举多段上传任务	<p>您必须得到可以对桶执行 <i>ListBucketMultipartUploads</i> 操作的权限，才能列出正在上传到该桶的多段上传。</p> <p>除了默认情况之外，桶所有者可以允许其他委托人对桶执行 <i>ListBucketMultipartUploads</i> 操作。</p>

使用适用于多段上传的 REST API

对象存储服务API参考中的以下章节介绍适用于多段上传的REST API。

- 列举桶中已初始化多段任务
- 初始化上传段任务
- 上传段
- 拷贝段
- 列举已上传的段
- 合并段
- 取消多段上传任务

4.3 下载

使用场景

您可以根据需要将存储在OBS中的对象下载到本地。

OBS支持批量下载对象，详情请参见[OBS批量下载文件操作指导](#)。

约束与限制

- 对于存储类别为归档存储或深度归档存储的对象，需要确认对象的状态为“已恢复”才能对其进行下载。

使用方式

OBS支持通过控制台、OBS Browser+、API、SDK、obsutil方式下载对象。其中，OBS Browser+、obsutil支持下载文件夹。

支持的使用方式	参考文档
控制台	下载文件
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	获取对象内容
OBS Browser+	下载文件或文件夹
obsutil	下载对象

4.4 文件夹管理

4.4.1 创建文件夹

使用场景

您可以在已创建的桶中新建一个文件夹，从而更方便的对存储在OBS中的数据进行分类管理。

由于OBS是一种对象存储服务，并没有文件系统中的文件和文件夹概念。为了使用户更方便进行管理数据，OBS提供了一种方式模拟文件夹。实际上在OBS内部是通过在对象的名称中增加“/”，将该对象在OBS管理控制台或者OBS Browser+上模拟成一个文件夹的形式展现。通过API等方式列举对象，获取到的对象名就是以“/”分隔的，最后一个“/”后的内容就是对象名。如果最后一个“/”后没有内容，则表示一个文件夹路径。文件夹的层级结构深度不会影响访问对象的性能。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式创建文件夹。

支持的使用方式	参考文档
控制台	新建文件夹
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。 说明 通过SDK创建文件夹，本质上来说是创建了一个大小为0且对象名以“/”结尾的对象。
API	- 说明 通过API创建文件夹，本质上来说是创建了一个大小为0且对象名以“/”结尾的对象。
OBS Browser+	新建文件夹
obsutil	创建文件夹

4.4.2 分享文件夹

使用场景

实际业务场景下，您可能会需要将整个文件夹临时分享给其他人或者将文件夹永久分享给其他人。

- 临时分享：

通过文件夹的临时URL直接将其分享给其他用户。文件夹分享强调临时性，所有分享的URL都是临时URL，存在有效期。

临时分享分为两种方式：提取码分享、直接分享。

提取码分享

分享者需要先设置一个6位数的提取码，再创建分享。创建成功后，OBS会自动将文件夹中的所有对象的下载链接汇总到一个静态网站中，并托管到一个公共的

OBS桶。所有用户均可使用创建分享时生成的临时URL和提取码，访问这个静态网站，并进行文件下载。

临时URL是由文件夹的访问域名和临时鉴权信息组成。示例如下：

```
https://e-share.obs-website.ap-southeast-1.myhuaweicloud.com:443/image.png?token=xxx
```

临时鉴权的方法及各参数的详细解释，请参见《对象存储服务API参考》的[URL中携带签名](#)章节。

直接分享

分享者输入有效期后直接分享链接给用户。用户通过一个签名即可访问文件夹下所有的对象。

访问文件夹内对象的临时URL是由桶域名（前缀）+对象路径+签名信息（后缀）组成。

- 永久共享：

通过将对象权限设置为匿名用户读取权限，通过分享对象URL，匿名用户通过分享的链接地址可访问对象数据。设置权限请参见[对匿名用户授予指定对象的公共读权限](#)。

获取对象访问URL请参见[如何获取对象访问路径？](#)

操作失败请参见[通过URL访问对象失败](#)进行处理。

约束与限制

- 仅部分区域支持文件夹分享功能，具体支持区域请参见[功能总览](#)。
- 归档存储或深度归档存储对象需恢复后才能分享。
- 仅桶版本号为3.0的桶支持文件夹临时分享功能。查询桶版本号方法请参见[OBS 2.0和OBS 3.0的判断方法](#)。
- 临时分享文件夹的URL有效期如下：

支持的使用方式	有效期
控制台	取值范围为1分钟到18小时。 当对对象执行“分享”之后，OBS就会以默认5分钟的有效期限获取临时鉴权信息，并生成分享链接，此时链接就已经生效并且开始计算时间了。每调整一次URL有效期，OBS就会重新获取一次鉴权信息以生成新的分享链接，新链接的有效期从调整的时候开始计算。
API	通过参数“Expires”设置临时授权失效的时间，临时授权失效的时间为24小时。
OBS Browser+	使用账号密码登录时，分享的有效期限最长只有24小时，默认10小时。 如果需要更长的有效期，请使用永久AK/SK方式登录。
obsutil	通过附加参数“vp”设置生成授权码的有效期限，默认值为1天。该参数支持多种时间单位的配置方式，包括：m（月）、w（周）、d（天）、h（小时）、min（分钟）、s（秒）。如果不带时间单位，则该参数为授权码有效的秒数。

使用方式

OBS支持通过控制台、API、OBS Browser+、obsutil方式分享文件夹。

支持的使用方式	参考文档
控制台	分享文件夹
API	URL中携带签名
OBS Browser+	分享文件或文件夹
obsutil	创建目录分享的授权码

4.5 其他对象管理操作

4.5.1 列举对象

使用场景

您可以通过列举对象查看已创建的对象列表。

使用方式

OBS支持通过控制台、OBS Browser+、API、SDK、obsutil方式列举对象。

支持的使用方式	参考文档
控制台	登录控制台并选择“对象存储”，单击“桶名称”后进入“对象”页面，即可查看桶内所有对象。
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	列举桶内对象
OBS Browser+	-
obsutil	列举对象

4.5.2 复制对象

使用场景

用户可以根据需要将存储在OBS上的对象复制到其他路径下。复制对象操作将创建需要复制的对象的副本。在单次操作中，您可以创建最大5GB的对象副本。对于复制大于5GB的对象副本，请您使用多段上传API。通过使用copy操作，您可以：

- 创建对象的其他副本。

- 通过复制对象并删除原始对象来重命名它们。
- 更改对象元数据。每个OBS对象都带有元数据，它是一组名称值对，您可以在上传对象时对其进行设置。上传对象后，您可以通过接口修改元数据，详情请参见[修改对象元数据](#)。也可以创建对象的副本并设置元数据，在复制操作中，设置与源和目标相同的对象。

每个对象都带有元数据。有些是系统元数据，而另外一些则是用户定义的元数据。用户可以控制某些系统元数据。复制对象时，还会复制用户控制的系统元数据和用户定义的元数据。OBS将重设系统控制的元数据。例如，在复制对象时，OBS将重设已复制的对象的创建日期。在复制请求中，您无需设置这些值。

复制对象时，您可能会决定更新某些元数据值。例如，如果您的源对象被配置为使用标准存储，您可能会为对象复制选择低频访问存储。您可能还会决定更改源对象上某些用户定义的元数据值。请注意，如果您选择在复制期间更新任意对象的用户可配置元数据（系统或用户定义的），则您必须显式地指定所有的用户可配置元数据，即使您只更改一个元数据的值（显示在请求中源对象上）。

如果您想将对象自动复制到其他区域，可以创建跨区域复制规则，以满足异地容灾等需求，详情请参见[跨区域复制](#)。

📖 说明

- OBS帮助您将存储于华为云对象存储的业务数据复制到指定Region，但华为云并不感知您的数据的具体内容，也不对您使用本服务的合法合规性负责。如果您的复制涉及跨境传输，请确保您使用本服务符合法律法规的要求。
- 在桶没有开启多版本的情况下，将源对象objecta复制为目标对象objectb，如果在复制操作之前对象objectb已经存在，复制操作执行之后老对象objectb则会被新复制对象objectb覆盖，复制成功后，只能下载到新的对象objectb，老对象objectb将会被删除。因此在使用copy接口时请确保目标对象不存在或者已无价值，避免因copy导致数据误删除。复制过程中源对象objecta无任何变化。
- 复制对象的结果不能仅根据HTTP返回头域中的status_code来判断请求是否成功，头域中status_code返回200时表示服务端已经收到请求，且开始处理复制对象请求。复制是否成功会在响应消息的body中，只有body体中有ETag标签才表示成功，否则表示复制失败。

使用方式

OBS支持通过API、SDK、OBS Browser+、obsutil方式复制对象。

支持的使用方式	参考文档
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	复制对象
OBS Browser+	复制文件或文件夹
obsutil	复制对象

4.5.3 查看对象信息

使用场景

上传对象后，您可以根据需要查看对象的信息（对象属性）。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式查看对象信息。

支持的使用方式	参考文档
控制台	-
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	获取对象元数据
OBS Browser+	-
obsutil	查询对象属性

4.5.4 分享对象

使用场景

实际业务场景下，您可能需要需要将文件临时分享给其他人或者将文件永久共享给其他人。

- 临时分享：

通过文件的临时URL直接将其分享给其他用户。文件分享强调临时性，所有分享的URL都是临时URL，存在有效期。

临时URL是由文件的访问域名和临时鉴权信息组成。示例如下：

```
https://bucketname.obs.ap-southeast-1.myhuaweicloud.com:443/image.png?  
AccessKeyId=xxx&Expires=xxx&x-obs-security-token=xxx&Signature=xxx
```

临时鉴权信息主要包含**AccessKeyId**、**Expires**、**x-obs-security-token**和**Signature**四个参数。其中**AccessKeyId**、**x-obs-security-token**和**Signature**用于鉴权，**Expires**定义鉴权的有效期。

临时鉴权的方法及各参数的详细解释，请参见《对象存储服务API参考》的[URL中携带签名](#)章节。

- 永久共享：

通过将对象权限设置为匿名用户读取权限，通过分享对象URL，匿名用户通过分享的链接地址可访问对象数据。设置权限请参见[对匿名用户授予指定对象的公共读权限](#)。

对象类型不同，通过浏览器访问该对象的方式不同。例如：.txt、.html等文件可直接通过浏览器打开浏览，而.exe、.dat等文件则会在浏览器打开时，自动下载到本地。

获取对象访问URL请参见[如何获取对象访问路径?](#)

操作失败请参见[通过URL访问对象失败](#)进行处理。

约束与限制

- 归档存储或深度归档存储对象需恢复后才能分享。
- 临时分享文件不需要授权码。临时分享文件夹中的提取码分享方式需要授权码，详情请参见[分享文件夹](#)。

- 仅桶版本号为3.0的桶支持临时文件分享功能。查询桶版本号方法请参见[OBS 2.0和OBS 3.0的判断方法](#)。
- 临时分享文件的URL有效期如下：

支持的使用方式	有效期
控制台	取值范围为1分钟到18小时。 当对对象执行“分享”之后，OBS就会以默认5分钟的有效期限获取临时鉴权信息，并生成分享链接，此时链接就已经生效并且开始计算时间了。每调整一次URL有效期，OBS就会重新获取一次鉴权信息以生成新的分享链接，新链接的有效期从调整的时候开始计算。
SDK	通过参数“Expires”设置临时授权失效的时间，临时授权失效的时间为24小时。
API	通过参数“Expires”设置临时授权失效的时间，临时授权失效的时间为24小时。
OBS Browser+	使用账号密码登录时，分享的有效期限最长只有24小时，默认10小时。 如果需要更长的有效期，请使用永久AK/SK方式登录。
obsutil	通过附加参数“e”设置生成的对象下载链接的过期时间（单位：秒），最小值为60秒，默认为300秒，设置的过期时间没有上限。

使用方式

OBS支持通过控制台、SDK、API、OBS Browser+、obsutil方式分享文件。

支持的使用方式	参考文档
控制台	分享文件
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	URL中携带签名
OBS Browser+	分享文件或文件夹
obsutil	生成对象的下载链接

相关操作

文件分享功能生成临时URL，使用wget下载时，需要通过“-O”指定文件名，否则会出错。

例如：

```
[root@ecs-test ~]# wget "临时URL" -O abc
```

其中，“abc”为新的文件名。

4.5.5 管理对象元数据

使用场景

元数据（Metadata）为描述对象属性的信息，是一组名称值对，用作对象管理的一部分。

当前仅支持系统定义的元数据。

系统定义的元数据又分为两种类别：系统控制和用户控制。如Last-Modified日期等数据由系统控制，不可修改；如为对象配置的ContentLanguage，用户可以通过接口进行修改。用户可控制修改的元数据描述如下：

表 4-3 OBS 的元数据

名称	说明
ContentDisposition	<p>为请求的对象提供一个默认的文件名赋值给该对象，当下载对象或者访问对象时，以默认文件名命名的文件将直接在浏览器上显示或在访问时弹出文件下载对话框。</p> <p>例如：元数据名称选择为“ContentDisposition”，元数据值填写为“attachment;filename="testfile.xls"”，当通过链接访问设置了该元数据的对象时，会直接弹出一个对象下载的对话框，且对象名称会被修改为“testfile.xls”。</p> <p>详情请参见HTTP协议中关于ContentDisposition的定义。</p>
ContentLanguage	<p>说明访问者希望采用的语言或语言组合，以根据自己偏好的语言来定制。详情请参见HTTP协议中关于ContentLanguage的定义。</p>
WebsiteRedirectLocation	<p>为对象提供重定向功能，重定向到其他对象或者外部的URL。重定向功能通过静态网站托管实现。</p> <p>例如，可根据如下步骤实现对象重定向功能。</p> <ol style="list-style-type: none">为桶“testbucket”根目录下的对象“testobject.html”设置元数据，元数据名称选择为“WebsiteRedirectLocation”，元数据值填写为“http://www.example.com” <p>说明</p> <p>OBS仅支持为桶根目录下的对象设置重定向，不支持为桶中文件夹下的对象设置重定向。</p> <ol style="list-style-type: none">在桶“testbucket”中配置静态网站托管，将该桶中的对象“testobject.html”设置为静态网站托管的“默认首页”。当通过静态网站托管页面上的“访问地址”访问对象“testobject.html”时，会直接重定向访问http://www.example.com。

名称	说明
ContentEncoding	指定对象被下载时的内容编码格式，可以设置如下类型： <ul style="list-style-type: none"> 标准定义：compress、deflate、exi、identity、gzip、pack200-gzip 其他：br、bzip2、lzma、peerdist、sdch、xpress、xz
CacheControl	指定对象被下载时的网页的缓存行为。 <ul style="list-style-type: none"> 可缓冲性：public、private、no-cache、only-if-cached 到期时间：max-age=<seconds>、s-maxage=<seconds>、max-stale[=<seconds>]、min-fresh=<seconds>、stale-while-revalidate=<seconds>、stale-if-error=<seconds> 重新验证和重新加载：must-revalidate、proxy-revalidate、immutable 其他：no-store、no-transform
Expires	设置缓存过期时间（GMT）。
ContentType	设置对象的文件类型。 详情请参见 对象元数据Content-Type介绍 。

📖 说明

- 当桶开启多版本控制时，最新版本的对象支持设置元数据，历史版本的对象不支持设置元数据。
- 对于存储类别为归档存储或深度归档存储的对象，不能对其设置对象的元数据。

对象元数据 Content-Type 介绍

使用控制台和工具上传到OBS中的对象，会根据对象的文件扩展名，自动匹配Content-Type值。使用浏览器访问对象时，会根据Content-Type类型来指定应用程序来打开。您可以根据对象的文件扩展来修改Content-Type。

📖 说明

调用API上传对象时，不会自动匹配Content-Type值，需要指定Content-Type值。

表 4-4 常见的 Content-Type 类型

文件扩展名	Content-Type	文件扩展名	Content-Type
.*（二进制流，不知道下载文件类型）	application/octet-stream	.7z	application/x-7z-compressed

文件扩展名	Content-Type	文件扩展名	Content-Type
.001	application/x-001	.301	application/x-301
.323	text/h323	.906	application/x-906
.907	drawing/907	.a11	application/x-a11
.acp	audio/x-mei-aac	.ai	application/ postscript
.aif	audio/aiff	.aifc	audio/aiff
.aiff	audio/aiff	.anv	application/x-anv
.asa	text/asa	.asf	video/x-ms-asf
.asp	text/asp	.asx	video/x-ms-asf
.au	audio/basic	.avi	video/avi
.awf	application/ vnd.adobe.workflo w	.biz	text/xml
.bmp	application/x-bmp	.bot	application/x-bot
.c4t	application/x-c4t	.c90	application/x-c90
.cal	application/x-cals	.cat	application/ vnd.ms-pki.seccat
.cdf	application/x- netcdf	.cdr	application/x-cdr
.cel	application/x-cel	.cer	application/x- x509-ca-cert
.cg4	application/x-g4	.cgm	application/x-cgm
.cit	application/x-cit	.class	java/*
.cml	text/xml	.cmp	application/x-cmp
.cmx	application/x-cmx	.cot	application/x-cot
.crl	application/pkix- crl	.crt	application/x- x509-ca-cert
.csi	application/x-csi	.css	text/css
.cut	application/x-cut	.dbf	application/x-dbf
.dbm	application/x-dbm	.dbx	application/x-dbx
.dcd	text/xml	.dcx	application/x-dcx
.der	application/x- x509-ca-cert	.dgn	application/x-dgn

文件扩展名	Content-Type	文件扩展名	Content-Type
.dib	application/x-dib	.dll	application/x-msdownload
.doc	application/msword	.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
.drw	application/x-drw	.dot	application/msword
.dwf	Model/vnd.dwf	.dtd	text/xml
.dwg	application/x-dwg	.dwf	application/x-dwf
.dxf	application/x-dxf	.dxb	application/x-dxb
.emf	application/x-emf	.edn	application/vnd.adobe.edn
.ent	text/xml	.eml	message/rfc822
.eps	application/x-ps	.epi	application/x-epi
.etd	application/x-ebx	.eps	application/postscript
.fax	image/fax	.exe	application/x-msdownload
.fif	application/fractals	.fdf	application/vnd.fdf
.frm	application/x-frm	.fo	text/xml
.gbr	application/x-gbr	.g4	application/x-g4
.gif	image/gif	.	application/x-
.gp4	application/x-gp4	.gl2	application/x-gl2
.hmr	application/x-hmr	.hgl	application/x-hgl
.hpl	application/x-hpl	.hpg	application/x-hpgl
.hrf	application/x-hrf	.hqx	application/mac-binhex40
.htc	text/x-component	.hta	application/hta
.html	text/html	.htm	text/html
.htx	text/html	.htt	text/webviewhtml
.ico	image/x-icon	.icb	application/x-icb

文件扩展名	Content-Type	文件扩展名	Content-Type
.iff	application/x-iff	.ico	application/x-ico
.igs	application/x-igs	.ig4	application/x-g4
.img	application/x-img	.iii	application/x-iphone
.isp	application/x-internet-signup	.ins	application/x-internet-signup
.java	java/*	.IVF	video/x-ivf
.jpe	image/jpeg	.jfif	image/jpeg
.jpeg	image/jpeg	.jpe	application/x-jpe
.jpg	application/x-jpg	.jpg	image/jpeg
.jsp	text/html	.js	text/javascript
.lar	application/x-laplayer-reg	.la1	audio/x-liquid-file
.lavs	audio/x-liquid-secure	.latex	application/x-latex
.lmsff	audio/x-la-lms	.lbm	application/x-lbm
.ltr	application/x-ltr	.ls	application/x-javascript
.m2v	video/x-mpeg	.m1v	video/x-mpeg
.m4e	video/mpeg4	.m3u	audio/mpegurl
.man	application/x-troff-man	.mac	application/x-mac
.mdb	application/msaccess	.math	text/xml
.mfp	application/x-shockwave-flash	.mdb	application/x-mdb
.mhtml	message/rfc822	.mht	message/rfc822
.mid	audio/mid	.mi	application/x-mi
.mil	application/x-mil	.midi	audio/mid
.mnd	audio/x-musicnet-download	.mml	text/xml
.mocha	application/x-javascript	.mns	audio/x-musicnet-stream
.mp1	audio/mp1	.movie	video/x-sgi-movie

文件扩展名	Content-Type	文件扩展名	Content-Type
.mp2v	video/mpeg	.mp2	audio/mp2
.mp4	video/mp4	.mp3	audio/mp3
.mpd	application/ vnd.ms-project	.mpa	video/x-mpg
.mpeg	video/mpg	.mpe	video/x-mpeg
.mpga	audio/rn-mpeg	.mpg	video/mpg
.mps	video/x-mpeg	.mpp	application/ vnd.ms-project
.mpv	video/mpg	.mpt	application/ vnd.ms-project
.mpw	application/ vnd.ms-project	.mpv2	video/mpeg
.mtx	text/xml	.mpx	application/ vnd.ms-project
.net	image/pnetvue	.mxx	application/x- mxx
.nws	message/rfc822	.nrf	application/x-nrf
.out	application/x-out	.odc	text/x-ms-odc
.p12	application/x- pkcs12	.p10	application/ pkcs10
.p7c	application/pkcs7- mime	.p7b	application/x- pkcs7-certificates
.p7r	application/x- pkcs7-certreqresp	.p7m	application/pkcs7- mime
.pc5	application/x-pc5	.p7s	application/pkcs7- signature
.pcl	application/x-pcl	.pci	application/x-pci
.pdf	application/pdf	.pcx	application/x-pcx
.pdx	application/ vnd.adobe.pdx	.pfx	application/x- pkcs12
.pgl	application/x-pgl	.pic	application/x-pic
.pko	application/ vnd.ms-pki.pko	.pl	application/x-perl
.plg	text/html	.pls	audio/scpls
.plt	application/x-plt	.png	image/png

文件扩展名	Content-Type	文件扩展名	Content-Type
.png	application/x-png	.pot	application/ vnd.ms- powerpoint
.ppa	application/ vnd.ms- powerpoint	.ppm	application/x-ppm
.pps	application/ vnd.ms- powerpoint	.ppt	application/ vnd.ms- powerpoint
.ppt	application/x-ppt	.pr	application/x-pr
.prf	application/pics- rules	.prn	application/x-prn
.prt	application/x-prt	.ps	application/x-ps
.ps	application/ postscript	.ptn	application/x-ptn
.pwz	application/ vnd.ms- powerpoint	.r3t	text/vnd.rn- realtext3d
.ra	audio/vnd.rn- realaudio	.ram	audio/x-pn- realaudio
.ras	application/x-ras	.rat	application/rat- file
.rdf	text/xml	.rec	application/ vnd.rn-recording
.red	application/x-red	.rgb	application/x-rgb
.rjs	application/ vnd.rn- realsystem-rjs	.rjt	application/ vnd.rn- realsystem-rjt
.rlc	application/x-rlc	.rle	application/x-rle
.rm	application/ vnd.rn-realmedia	.rmf	application/ vnd.adobe.rmf
.rmi	audio/mid	.rmj	application/ vnd.rn- realsystem-rmj
.rmm	audio/x-pn- realaudio	.rmp	application/ vnd.rn- rn_music_package

文件扩展名	Content-Type	文件扩展名	Content-Type
.rms	application/ vnd.rn-realmedia- secure	.rmvb	application/ vnd.rn-realmedia- vbr
.rmx	application/ vnd.rn- realsystem-rmx	.rnx	application/ vnd.rn-realplayer
.rp	image/vnd.rn- realpix	.rpm	audio/x-pn- realaudio-plugin
.rsml	application/ vnd.rn-rsml	.rt	text/vnd.rn- realtext
.rtf	application/ msword	.rtf	application/x-rtf
.rv	video/vnd.rn- realvideo	.sam	application/x-sam
.sat	application/x-sat	.sdp	application/sdp
.sdw	application/x-sdw	.sit	application/x- stuffit
.slb	application/x-slb	.sld	application/x-sld
.slk	drawing/x-slk	.smi	application/smil
.smil	application/smil	.smk	application/x-smk
.snd	audio/basic	.sol	text/plain
.sor	text/plain	.spc	application/x- pkcs7-certificates
.spl	application/ futuresplash	.spp	text/xml
.ssm	application/ streamingmedia	.sst	application/ vnd.ms- pki.certstore
.stl	application/ vnd.ms-pki.stl	.stm	text/html
.sty	application/x-sty	.svg	text/svg+xml
.swf	application/x- shockwave-flash	.tdf	application/x-tdf
.tg4	application/x-tg4	.tga	application/x-tga
.tif	image/tiff	.tif	application/x-tif
.tiff	image/tiff	.tld	text/xml

文件扩展名	Content-Type	文件扩展名	Content-Type
.top	drawing/x-top	.torrent	application/x-bittorrent
.tsd	text/xml	.txt	text/plain
.uin	application/x-icq	.uls	text/iuls
.vcf	text/x-vcard	.vda	application/x-vda
.vdx	application/ vnd.visio	.vml	text/xml
.vpg	application/x- vpeg005	.vsd	application/ vnd.visio
.vsd	application/x-vsdx	.vss	application/ vnd.visio
.vst	application/ vnd.visio	.vst	application/x-vst
.vsw	application/ vnd.visio	.vsx	application/ vnd.visio
.vtx	application/ vnd.visio	.vxml	text/xml
.wav	audio/wav	.wax	audio/x-ms-wax
.wb1	application/x-wb1	.wb2	application/x-wb2
.wb3	application/x-wb3	.wbmp	image/ vnd.wap.wbmp
.wiz	application/ msword	.wk3	application/x-wk3
.wk4	application/x-wk4	.wkq	application/x-wkq
.wks	application/x-wks	.wm	video/x-ms-wm
.wma	audio/x-ms-wma	.wmd	application/x-ms- wmd
.wmf	application/x-wmf	.wml	text/vnd.wap.wml
.wmv	video/x-ms-wmv	.wmx	video/x-ms-wmx
.wmz	application/x-ms- wmz	.wp6	application/x-wp6
.wpd	application/x-wpd	.wpg	application/x-wpg
.wpl	application/ vnd.ms-wpl	.wq1	application/x-wq1
.wr1	application/x-wr1	.wri	application/x-wri

文件扩展名	Content-Type	文件扩展名	Content-Type
.wrk	application/x-wrk	.ws	application/x-ws
.ws2	application/x-ws	.wsc	text/scriptlet
.wsdl	text/xml	.wvx	video/x-ms-wvx
.xdp	application/ vnd.adobe.xdp	.xdr	text/xml
.xfd	application/ vnd.adobe.xfd	.xdf	application/ vnd.adobe.xdf
.xhtml	text/html	.xls	application/ vnd.ms-excel
.xls	application/x-xls	.xlw	application/x-xlw
.xml	text/xml	.xpl	audio/scpls
.xq	text/xml	.xql	text/xml
.xquery	text/xml	.xsd	text/xml
.xsl	text/xml	.xslt	text/xml
.xwd	application/x-xwd	.x_b	application/x-x_b
.sis	application/ vnd.symbian.instal l	.sisx	application/ vnd.symbian.instal l
.x_t	application/x-x_t	.ipa	application/ vnd.iphone
.apk	application/ vnd.android.packa ge-archive	.xap	application/x- silverlight-app
.zip	application/zip	.rar	application/x-rar- compressed

对象自定义元数据介绍

OBS支持用户使用以“x-obs-meta-”开头的消息头来加入自定义的元数据，以便对对象进行自定义管理。当用户获取此对象或查询此对象元数据时，加入的自定义元数据将会在返回的消息头中出现。自定义元数据大小限制为8K，计算方式为：每个键和值的UTF-8 编码中的字节总数来测量用户定义的元数据的大小。

自定义元数据的key值不区分大小写，OBS统一转为小写进行存储。value值区分大小写。

示例：

```
PUT /key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-Test1: Test Meta1
```

```
HEAD /Key HTTP/1.1
Host: bucket01.obs.myhuaweicloud.com
x-obs-meta-test1: Test Meta1
```

自定义元数据key-value对都必须符合US-ASCII。如果一定要使用非ASCII码或不可识别字符，需要客户端自行做编解码处理，可以采用URL编码或者Base64编码，服务端不会做解码处理。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式配置对象自定义元数据。

支持的使用方式	参考文档
控制台	配置对象元数据
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	修改对象元数据
OBS Browser+	-
obsutil	上传对象

4.5.6 恢复归档存储或深度归档存储对象

使用场景

对于存储类别为归档存储或深度归档存储的对象，用户需要先恢复才能下载、通过对对象URL访问对象、设置对象元数据。

OBS提供加急和标准两种不同的恢复方式，加急恢复归档存储恢复耗时1~5 min，加急恢复深度归档（受限公测）存储恢复约耗时3~5 h；标准恢复对象归档存储恢复耗时3~5 h，标准恢复深度归档（受限公测）存储恢复约耗时5~12 h。

说明

对于深度归档对象，大批量恢复建议使用标准恢复，且对象最终恢复时间受对象大小以及取回数据量的影响。

数据恢复时间和收费详情请参见[产品价格详情](#)。

约束与限制

- 归档存储或深度归档存储的对象正在恢复的过程中，不允许暂停或删除恢复任务。
- 对象的恢复状态为恢复中时，对象不能再次恢复。
- 数据恢复后，会产生一个标准存储类型的对象副本，即对象同时存在标准存储类型的对象副本和归档存储或深度归档存储类型的对象。归档存储或深度归档存储对象恢复完成时，对象的恢复状态显示“已恢复”，生成的标准存储类型的对象副本不会在桶中展示。

在恢复的有效期内，会同时收取这份数据在标准存储和归档存储或深度归档存储中的存储费用。恢复有效期到期后标准存储类型的对象副本会自动删除。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式恢复归档存储对象。

支持的使用方式	参考文档
控制台	恢复归档存储或深度归档存储对象
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	恢复归档存储或深度归档存储对象
OBS Browser+	恢复归档存储对象
obsutil	恢复归档存储对象

4.5.7 直读归档存储数据

使用场景

实际业务场景下，您可能有不通过恢复而直接获取归档数据的诉求。如有时间要求，需要急速读取归档数据；或者在OBS与其他系统对接的情况下，因接口兼容问题，需要直接读取归档数据。

桶开启归档数据直读后，存储类别为归档存储的对象可以直接下载，无需提前恢复。

归档数据直读会收取相应的费用，详情请参见[产品价格详情](#)。

您可以在创建桶时开启归档数据直读，也可以在已创建的桶中根据需要开启归档数据直读。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+方式设置直读归档存储数据。

支持的使用方式	参考文档
控制台	归档数据直读
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶归档对象直读策略
OBS Browser+	-

4.6 删除

4.6.1 删除对象

使用场景

为节省空间和成本，您可以根据需要删除无用的文件或文件夹。

须知

桶没有开启多版本控制功能时，已删除的对象不可恢复，请谨慎操作。

手动删除和自动删除

您可以随时手动删除对象，也可以通过[生命周期管理](#)功能，定期自动删除桶中对象或者一次性清空桶中所有对象。

对于并行文件系统目录，大数据场景下（目录层级深、目录下文件多）的删除，可能会因超时而删除失败，建议使用：

1. hadoop 客户端（嵌套OBS客户端插件OBSA）删除目录：`hadoop fs -rmr obs://{并行文件系统名}/{目录名}`。
2. 给目录配置[生命周期](#)规则，通过生命周期后台删除。

单个删除和批量删除

OBS支持单个和批量删除对象，各个使用方式的批量删除方法请参见[批量删除对象](#)。

说明

批量删除的性能和单个请求内的对象数负相关，对于QPS的计算，删除N个对象，算N次操作。如果删除对象数量大并且对象前缀使用了字典序，可能导致大量对象的请求访问集中于某个特定分区，造成访问热点。热点分区上的请求速率受限，访问时延上升。

为解决以上问题，您可以考虑减少单个批量删除请求的对象数量，增加并发请求数，并将对象名的顺序前缀改为随机性前缀。

多版本控制功能启用时的对象删除机制

桶的多版本控制功能启用时，删除的目标不同，OBS会采取不同的处理方式：

- 删除文件或文件夹：文件或文件夹不会立即被彻底删除，而是保留在“已删除对象”列表中，同时会为文件打上删除标记。
 - 如果想要彻底删除，需要再到“已删除对象”列表进行删除。
 - 如果想要找回删除的文件，可以通过“取消删除”功能来找回。详情请参见[取消删除对象](#)。
- 删除文件的某个版本：该版本会被彻底删除且无法恢复。如果删除的是文件的最新版本，那么时间最近的那个历史版本将会变成最新版本。

在多版本控制功能启用的场景下，在“已删除对象”中的文件仍然会保留多版本，在对不同的版本进行删除时需要注意：

- 如果删除的是带“删除标记”的版本，实际上是找回该文件，等同于“取消删除”文件，而非彻底删除。详情请参见[取消删除对象](#)。

- 如果删除的是不带“删除标记”的版本，则会彻底删除该历史版本，即使后续该文件找回后，也无法恢复这个被彻底删除的历史版本。

更多关于找回已删除对象的内容请参见[多版本控制功能启用时的对象找回机制](#)。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式删除对象。

支持的使用方式	参考文档
控制台	删除文件或文件夹
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	删除对象
OBS Browser+	删除文件或文件夹
obsutil	删除对象

4.6.2 取消删除对象

使用场景

桶没有开启多版本控制功能时，已删除的对象不可恢复。开启时，如果想将删除的文件找回，可以通过“取消删除”功能来实现。多版本控制功能的详细介绍请参见[多版本控制](#)。

多版本控制功能启用时的对象找回机制

启用了多版本控制功能的OBS桶中的文件从“对象”列表删除后，OBS不会立即将其彻底删除，而是保留在“已删除对象”中，同时会为其打上删除标记。您可以通过“取消删除”功能来找回被删除的文件。

使用“取消删除”功能需要注意以下几点：

1. 只支持对文件“取消删除”，不支持对文件夹“取消删除”。
“取消删除”文件后，该文件会恢复到“对象”列表中，此时可以正常使用对象的基本功能。如果文件存放于某个文件夹下，“取消删除”文件后依然会保留原有的目录结构。
2. “已删除对象”中的文件仍然会保留多版本，在对不同的版本进行删除时需要注意：
 - 如果删除的是带“删除标记”的版本，实际上是找回该文件，等同于“取消删除”文件，而非彻底删除。
 - 如果删除的是不带“删除标记”的版本，则会彻底删除该历史版本。即使后续该文件找回后，也无法恢复这个被彻底删除的历史版本。
3. “已删除对象”中的文件至少需要保留一个不带“删除标记”的历史版本，否则无法执行“取消删除”操作。

使用方式

OBS支持通过控制台取消删除对象。

支持的使用方式	参考文档
控制台	取消删除文件

4.6.3 清理碎片

使用场景

桶中不完整的数据称之为碎片。OBS采用分块上传的模式上传数据，在下列情况下（但不仅限于此）通常会导致数据上传失败而产生碎片。

- 网络条件较差，与OBS的服务器之间的连接经常断开。
- 上传过程中，人为中断上传任务。
- 设备故障。
- 突然断电等特殊情况。

OBS中的碎片会占用存储空间，会按照存储空间计费项进行计费。当存在运行中断或失败的分段上传任务并产生碎片后，您可以继续运行中断或失败的上传任务来消除碎片，或直接将碎片删除以节省存储空间。

您可以根据需要选用合适的方式清理碎片，详情请参见[如何处理碎片](#)。

📖 说明

- 当您遇到删除桶失败的场景，请检查碎片列表中的对象是否已经全部删除。如果没有，请先删除碎片列表中的所有对象。
- 当您遇到桶内无对象，却还在扣费的场景，请检查桶内是否还存在碎片。如果存在，请删除碎片以节约存储成本。

使用方式

表 4-5 删除碎片的不同方式

工具	方法
OBS控制台	OBS控制台提供图形化的碎片列表界面，您可以一键批量删除桶中所选中的碎片。 具体操作请参见 清理碎片 。
OBS Browser+	您可以一键批量删除桶中所选中的碎片。也可以单击页面上方“删除所有”，清除所有碎片。
obsutil	obsutil分段上传任务产生的碎片，可以通过删除分段上传任务来删除碎片。如果一个桶内有多个分段任务，则需要删除多个任务来删除桶中所有碎片。 具体操作请参见 删除分段上传任务 。

工具	方法
OBS API	<p>您可以通过以下步骤来直接删除桶中碎片：</p> <ol style="list-style-type: none">1. 使用列举桶中已初始化多段任务接口，来列举所有分段上传任务，获取所有UploadId。2. 使用取消多段上传任务接口，来取消分段上传任务，即可删除所有碎片。
OBS SDK	<p>OBS SDK通常只有采用多段上传，并且最后没有合并多段上传任务时会产生碎片，您通过取消多段上传任务来删除桶内碎片。步骤如下：</p> <ol style="list-style-type: none">1. 使用ObsClient.listMultipartUploads列举所有分段上传任务，获取所有UploadId。2. 使用ObsClient.abortMultipartUpload取消分段上传任务，即可删除所有分段碎片。

5 权限控制

使用场景

默认情况下，OBS的资源（桶和对象）都是私有的，只有资源所有者可以访问OBS资源，其他用户在未经授权的情况下均无OBS访问权限。OBS的权限控制是指通过编写访问策略向其他账号或者IAM用户授予资源的控制权限。例如，你拥有一个桶，你可以授权一个其他的IAM用户上传对象到你的桶中；你也可以将桶开放给非公有云用户访问，即桶作为一个公共资源，能被互联网上任何人访问。OBS提供多种方式将OBS资源权限授予给他人，资源所有者可以根据业务需求制定不同的权限控制方案，从而确保数据安全。

OBS权限管理方式的详细介绍，请参见[权限配置指南](#)。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+、obsutil方式配置权限。

支持的使用方式	参考文档
控制台	权限控制
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	IAM权限控制 设置桶策略 设置桶ACL 设置对象ACL
OBS Browser+	-
obsutil	设置桶属性 设置对象属性

6 数据管理

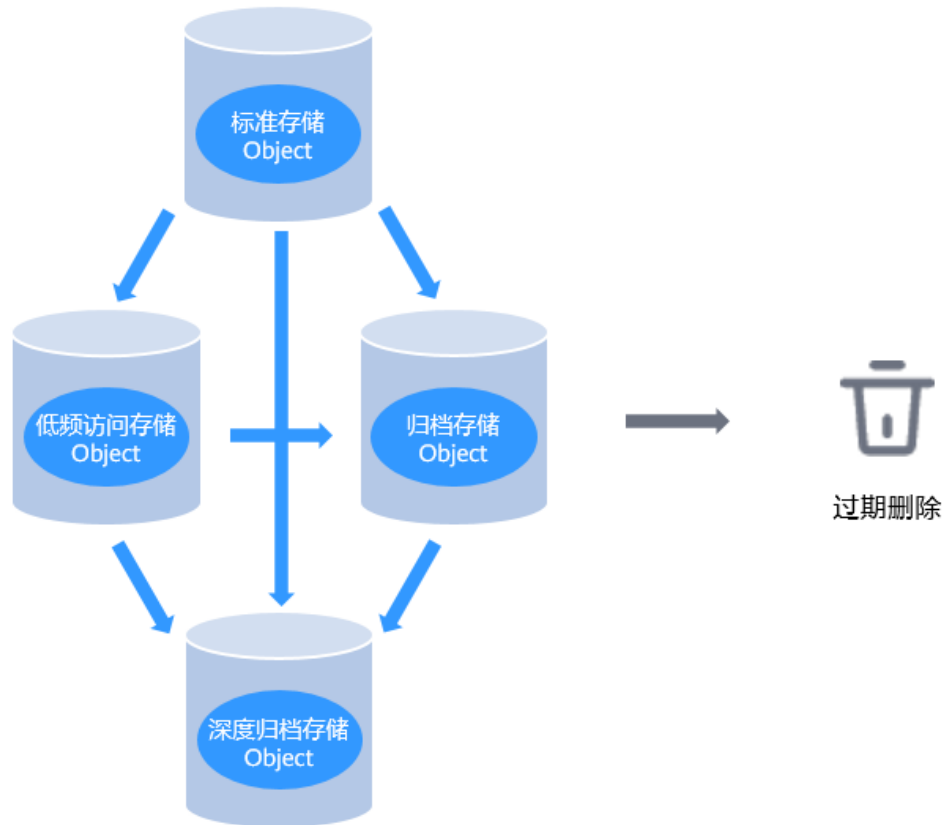
6.1 生命周期管理

生命周期管理是指通过配置指定的规则，实现在距离对象最后一次更新时间多少天后删除桶中的对象或者转换对象的存储类别。

说明

对象更新时间是指对象的上传时间；对于历史版本对象，对象更新时间是指变为历史版本的时间。

图 6-1 生命周期管理示意图



使用场景

生命周期管理可适用于以下典型场景：

- 周期性上传的日志文件，可能只需要保留一个星期或一个月。到期后要删除它们。
- 某些文档在一段时间内经常访问，但是超过一定时间后便可能不再访问了。这些文档需要在一定时间后转化为低频访问存储，归档存储或者删除。
- 为了存档目的而向OBS上传的某些类型的数据，包括数字媒体存档、金融和医疗记录、原始基因组序列数据、长期数据库备份以及为符合监管要求而必须保留的数据。
- 一次性删除桶中的大量文件。手动删除对象费时费力，且有数量限制。在桶中配置一条生命周期管理规则，设置定时删除所有文件即可。

对于上述场景中的对象，您可以定义用于识别这些对象的生命周期管理规则，通过这些规则实现对象的生命周期管理。

自动将某些不再频繁访问的对象转换为低频访问存储或者归档存储、深度归档存储，可以减少您的租赁费用。此处的“转换”表示在不复制原对象的情况下修改对象的存储类别。您可以在对象页面手动修改对象的存储类别，详情请参见[上传对象](#)的相关操作，也可以通过配置生命周期规则转换对象的存储类别。

您可以向不受版本控制的桶和启用了版本控制的桶添加生命周期配置。默认情况下，桶不受版本控制，您可以选择启用版本控制。启用了版本控制的桶会维护一个当前对象版本，以及零个或零个以上非当前对象版本。您可以结合使用版本控制和生命周期规则来帮助管理对象的存储成本。通过使用预定义的生命周期配置操作，可以管理当前和非当前对象版本的生命周期。

注意事项

- 单个桶的生命周期规则条数没有限制，但一个桶中所有生命周期规则的XML描述总大小不能超过20KB。
- 您最多可以在一个并行文件系统下配置20条生命周期管理规则，超过20条将提示不支持。
- 创建生命周期规则时，规则名称只能由大写或小写的英文字母、数字、英文句号（.）、下划线（_）和连字符（-）组成。
- 低频访问存储的最低存储时间为30天，归档存储的最低存储时间为90天，深度归档存储的最低存储时间为180天。如果对象经过转换，归档存储时间少于最低存储时间，需要补足剩余天数的存储费用。
- 对象存储类别转换限制：
 - 仅支持将标准存储对象转换为低频访问存储对象，低频访问存储对象转换为标准存储对象需手动转换。
 - 仅支持将标准存储或低频访问存储对象转换为归档存储对象。如果要将归档存储对象转换为标准存储或低频访问存储对象，需要手动恢复对象，然后手动转换存储类别。
 - 仅支持将标准存储、低频访问存储或归档存储对象转换为深度归档存储对象。如果要将深度归档存储对象转换为标准存储、低频访问存储对象或归档存储，需要手动恢复对象，然后手动转换存储类别。
 - 归档和深度归档存储不支持多AZ。因此不支持使用生命周期的存储类别转换功能，将多AZ桶或对象的存储类别转化为归档或深度归档存储。
- **对象过期删除：**

OBS提供Expiration操作，您可在生命周期配置中指定该操作以使对象过期。此外，对于启用了版本控制的桶，OBS还提供NoncurrentVersionExpiration操作，用于使非当前版本的对象过期。

 - 如果桶不受版本控制，则Expiration操作会使OBS永久删除对象。
 - 如果桶启用了版本控制（或版本控制已暂停），则Expiration操作会通过将删除标记添加为新的当前版本，在逻辑上删除当前版本。NoncurrentVersionExpiration操作永久删除非当前版本。
- 除OBS桶可以设置生存期外，上传对象时也可以设置生存期。对象的生存期优先于桶的生存期。

在对象的生存期结束后，OBS会将该对象加入删除队列并异步删除它。过期日期和OBS删除对象的日期之间可能会有一段滞后。对象被删除后，不会再向您收取相关的存储时间费用。

要找出对象计划过期的时间，您可以使用GET对象或HEAD对象API。这些API将返回可提供对象过期信息的响应标头。
- 对象更新后，系统会将下一个UTC零点作为对象存储的起始时间开始计算生命周期。生命周期规则执行最长耗时24小时。因此，存储类别的转换以及过期被删除可能会存在延时，且一般不会超过48小时。配置生命周期规则后，如果期间修改了生命周期配置，会重新计算生效时间。
- 修改生命周期配置后，配置不会立即对桶内所有对象生效。例如原有生命周期规则为1天过期删除，2021年1月3日已经开始删除2021年1月1日上传的满足过期条件的对象，如果此时修改生命周期规则为7天过期删除，那么新规则不会对2021年1月1日上传的对象生效，2021年1月1日上传的对象会在2021年1月3日被删除完成，新的规则只会对2021年1月2日及以后上传的对象生效。

生命周期规则

生命周期规则通常包含两个关键要素：

- 策略：生命周期规则匹配的对象。
 - 按前缀配置：指定对象名前缀来匹配受约束的对象，则匹配该前缀的对象将受规则影响。
 - 配置到整个桶：指定将生命周期管理规则配置到整个桶，则桶内所有对象都将受规则影响。
- 时间：设置对象的转换类别及转换时间。

您可以指定在对象最后一次更新后多少天，受规则影响的对象将转换为低频访问存储、归档存储、深度归档存储或者过期并自动被OBS删除。

 - 转换为低频访问存储：指定在对象最后一次更新后多少天，受规则影响的对象将转换为低频访问存储。
 - 转换为归档存储：指定在对象最后一次更新后多少天，受规则影响的对象将转换为归档存储。
 - 转换为深度归档存储：指定在对象最后一次更新后多少天，受规则影响的对象将转换为深度归档存储。
 - 过期删除：指定在对象最后一次更新后多少天，受规则影响的对象将过期并自动被OBS删除。

时间设置有如下约束限制：

- 仅设置一种转换：最少设置1天。
- 设置多种转换：后者时间要比前者时间最少长1天。

例：同时设置“转换为低频访问存储”和“过期删除”，不设置“转换为归档存储”。假设转换为低频访问存储设置为23天，则过期删除时间至少需要设置为24天。

说明

低频访问存储的最低存储时间为30天，归档存储的最低存储时间为90天，深度归档存储的最低存储时间为180天。如果对象在生命周期管理规则作用下，存储时间少于最低存储时间，需要补足剩余天数的存储费用，请您根据业务情况合理设置时间。

例：上传标准存储的对象后，不进行任何更新。生命周期管理规则的策略为配置到整个桶，3天转换为低频访问存储，4天转换为归档存储，7天过期删除。那么OBS服务将收取3天的标准存储费用，30天低频访问存储费用和90天的归档存储费用。

生命周期配置元素

您可将生命周期配置指定为XML。该配置包含一个或多个生命周期规则。

每个规则由以下内容组成：

- 规则元数据，包含规则ID以及用于指示规则是已启用还是已禁用的状态。如果规则处于禁用状态，OBS将不会执行规则中指定的任何操作。
- 筛选条件，用于标识即将应用规则的对象。您可以使用对象的前缀来指定筛选条件。
- 您希望OBS执行指定操作时的一个或多个转换或过期操作，带有位于对象的生命周期内的日志或时间段。

以下是两个配置示例：

例1：不受版本控制的桶的生命周期配置

默认情况下，您的桶不受版本控制。即桶中的每个对象只有一个版本。

假设您的桶未启用版本控制，且希望在创建30天后将对象名前缀为“documents/”的对象转为低频访问存储对象，创建60天后转为归档存储对象，创建1年之后永久删除。为此，您可以向桶附加以下生命周期配置。该生命周期配置定义的规则包含两个操作（Transition和Expiration），该规则应用于键前缀为“documents”（在Prefix元素中指定）的对象。

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
    <Transition>
      <Days>30</Days>
      <StorageClass>WARM</StorageClass>
    </Transition>
    <Transition>
      <Days>60</Days>
      <StorageClass>COLD</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

例2：启用了版本控制的桶的生命周期配置

您可以选择在桶上启用版本控制。如果您的桶启用了版本控制，则您有一个当前对象版本，以及零个或零个以上非当前版本。详情请参见[多版本控制](#)。通过对象版本控制，您可以维护对象版本的历史记录，而通过生命周期，您可以控制各版本存在的时间长度，或是各版本在转换为低频访问存储或归档存储类别以进行长期存档之前保持活动状态的时间长度。

对于启用了版本控制的桶，生命周期配置提供了更多预定义操作，可用于管理非当前对象版本。示例中，该生命周期配置有一个规则，它对键前缀为“logs/”的对象指定两个操作（NoncurrentVersionTransition和NoncurrentVersionExpiration）。NoncurrentVersionTransition操作在对象成为非当前版本30天之后，请求OBS将对象转换为低频访问存储类别，成为非当前版本60天后请求OBS转换为归档存储类别；NoncurrentVersionExpiration操作在对象成为非当前版本180天之后请求OBS永久删除对象。

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>180</NoncurrentDays>
    </NoncurrentVersionExpiration>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>WARM</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>60</NoncurrentDays>
      <StorageClass>COLD</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifecycleConfiguration>
```

您可以使用预定义NoncurrentVersionTransition和NoncurrentVersionExpiration操作管理桶中的非当前版本。

一般而言，每个生命周期配置规则都由以下内容组成：

- 规则元数据，包括规则ID（<ID>元素）以及用于指示规则是处于启用还是禁用状态（<Status>元素）。如果规则处于禁用状态，则OBS不会执行规则中指定的任何操作。
- 前缀（<Prefix>元素），用于通过键前缀标识要应用规则的对象。
- 您希望OBS对指定对象执行的操作（例如，上面示例中<NoncurrentVersionTransition>和<NoncurrentVersionExpiration>元素）。每个操作都需指定对象生命周期中您希望OBS执行指定操作的日期或时间段。

生命周期配置规则中元素介绍如下：

- **ID元素**

一个生命周期配置最多可以有1000个规则。ID元素唯一地标识规则。

- **Status元素**

Status元素值可以是Enabled或Disabled。如果规则处于禁用状态，则OBS不会执行在规则中定义的任何操作。

- **Prefix元素**

生命周期配置规则可应用于键名称以规则指定的前缀开头的单个对象或多个对象。例如，假设您拥有以下对象：

- logs/day1
- logs/day2
- logs/day3
- ExampleObject.jpg

如果您将ExampleObject.jpg指定为前缀，则规则仅应用于该特定对象。如果您将logs/指定为前缀，规则将应用于密钥名称以字符串“logs/”开头的三个对象。如果您指定空的前缀，该规则将应用于桶中的所有对象。

- **Action元素**

通过在生命周期配置规则中指定预定义操作，您可以请求OBS在对象的生命周期中执行特定操作。预定义操作包括：Transition、Expiration、NoncurrentVersionTransition和NoncurrentVersionExpiration。这些操作的效果取决于桶的版本控制状态。

说明

默认情况下，桶不受版本控制。您可以选择在桶上启用版本控制，在这种情况下，对象有一个当前版本和零个或零个以上非当前版本。您还可以临时暂停版本控制。详情请参见[多版本控制](#)。

- **Transition操作**

通过将对象的存储类别更改为低频访问存储或归档存储、深度归档存储来存档对象。达到对象生命周期中指定的日期或时间段时，OBS将适用的对象转换为低频访问存储或归档存储、深度归档存储对象。

- 不受版本控制的桶-Transition操作将对象的存储类别设置为低频访问存储或归档存储、深度归档存储。
- 启用了版本控制和暂停了版本控制的桶-Transition操作将当前版本对象的存储类别设置为低频访问存储或归档存储、深度归档存储。此操作不影响对象的非当前版本。

- **Expiration操作**

此操作可以使规则适用的对象过期。OBS使所有过期的对象不可用。是否永久删除对象取决于桶的版本控制状态。

📖 说明

Expiration操作不会删除未合并的多段对象。

- 不受版本控制的桶-Expiration操作永久删除对象，删除的对象无法恢复。
- 启用了版本控制的桶-此操作仅应用于当前对象版本，它不影响非当前版本。这种情况下的Expiration操作不会永久删除当前版本。它通过引入删除标记，将当前版本保留为非当前版本。如果当前版本已经带有删除标记，则不执行任何操作。如果当前版本是唯一的对象版本并且带有删除标记，则OBS将删除当前版本。清除删除标记时可能存在一些延迟，因为OBS需确保这是唯一的对象版本。

在不指定版本ID的情况下，对当前版本带删除标记的对象执行GET请求，则OBS将判断为对象已删除，返回404 Object Not Found错误。但是，您可以通过在GET请求中指定版本ID来恢复删除的对象。

例如，您可以设置一个规则，使名为photo.gif的对象创建五天之后过期。如果photo.gif在2016年1月1日上午10:30 UTC创建，则过期规则会在2016年1月7日00:00 UTC（对象创建五天之后）后的某个时间点执行，该时间点不会晚于2016年1月7日23:59 UTC。对于不受版本控制的桶，删除操作会将photo.gif永久删除；而对于启用了版本控制的桶，在过期规则执行之后，photo.gif（版本 111111）仍处于存储状态，可以在需要时进行访问，但是对象的当前版本（版本 4857693）现在带删除标记。原始photo.gif成为非当前版本。有关删除标记工作原理的信息，请参见[多版本控制](#)。

已暂停版本控制的桶：桶已暂停版本控制时，过期会导致使用null版本ID创建删除标记。任何现有null版本都会由新的null版本覆盖，与该版本关联的数据无法恢复。

特定于启用版本控制（或已暂停版本控制）的桶的操作

通过Transition和Expiration生命周期操作可以管理对象当前版本的生命周期。通过NoncurrentVersionTransition和NoncurrentVersionExpiration操作可以管理对象非当前（以前）版本的生命周期。

仅当桶是启用版本控制（或已暂停版本控制）的桶时，以下生命周期配置操作才适用。启用版本控制的桶可以有一个对象的许多版本，也就是一个当前版本和零个或多个以上非当前版本。您可以使用这些操作请求OBS对非当前对象版本执行特定操作。这些操作不会影响对象的当前版本。

NoncurrentVersionTransition操作-您可以使用此操作指定非当前版本在转换为WARM或COLD存储类型之前保留为STANDARD存储类型的时间长度。

NoncurrentVersionExpiration操作-您可以使用此操作指定在OBS永久删除非当前对象版本之前保留它们的时间长度。删除的对象无法恢复。

例如，如果您要启用一个为期五天的时段以更正任何意外删除或覆盖，您可以配置一个过期规则，用于在对象成为非当前版本五天之后删除非当前版本的对象。

在2016年1月1日上午10:30 UTC，您创建一个名为photo.gif（版本ID 111111）的对象。在2016年1月2日上午10:30 UTC，您意外删除了photo.gif（版本ID 111111），这会使用新版本ID（如版本ID 4857693）创建一个删除标记。您现在有五天时间可以在永久删除之前，恢复原始版本的photo.gif（版本ID 111111）。

在2016年1月8日00:00 UTC，过期生命周期规则执行并永久删除photo.gif（版本ID 111111）（在它成为非当前版本五天之后）。

OBS如何计算对象已成为非当前版本的时间长度

在启用版本控制的桶中，您可以有一个对象的多个版本，始终有一个当前版本和零个或零个以上非当前版本。每次上传对象时，当前版本都变为非当前版本，新上传的版本（后续者）成为当前版本。为了确定对象成为非当前版本的天数，OBS会查看新上传版本的创建时间。OBS使用新上传版本创建以来的天数作为对象成为非当前版本的天数。

在使用生命周期配置时还原对象的非当前版本

您可以通过以下两种方式还原对象的非当前版本：

1. 通过将对象的非当前版本复制到相同桶中。复制的对象将成为该对象的当前版本，且所有对象版本都保留。
2. 通过永久删除当前版本的对象。当您删除当前对象版本时，实际上是将非当前版本转换为该对象的当前版本。

推荐您使用第一种方法，即将对象的非当前版本复制到相同桶中。由于OBS的最终一致性语义，在更改传播之前，永久删除的当前版本可能不会消失（OBS可能不知道此删除操作）。同时，在您配置了使非当前对象过期的生命周期之后，可能会永久删除非当前对象，包括您要还原的对象。因此，复制非当前版本的方法更加安全。

表6-1 列出了对对象执行的生命周期配置规则操作与桶多版本控制状态的关系。

表 6-1 生命周期配置规则操作与桶多版本控制状态

操作	不受版本控制的桶	启用多版本控制的桶	已暂停多版本控制的桶
Transition 达到对象生命周期中指定的日期或时间段时	Transition转换对象，可以转换为WARM或COLD存储类别。	如果当前版本不带删除标记且是最新版本，则满足迁移时间可以迁移为WARM或COLD存储类别。	行为与启用了版本控制的桶相同。
Expiration 达到对象生命周期中指定的日期或时间段时	Expiration删除对象，删除的对象无法恢复。	如果当前版本不带删除标记，则生命周期创建删除标记（这会成为当前版本），现有当前版本保留为非当前版本。	桶已暂停版本控制时，过期会导致使用null版本ID创建删除标记。任何现有null版本都会由新的null版本覆盖，与该版本关联的数据无法恢复。
NoncurrentVersionTransition 对象成为非当前版本之后达到指定天数时	NoncurrentVersionTransition无效。	如果当前版本不带删除标记且不是最新版本，则满足转换时间可以转换为WARM或COLD存储类别。	行为与启用了版本控制的桶相同。
NoncurrentVersionExpiration 对象成为非当前版本之后达到指定天数时	NoncurrentVersionExpiration无效。	NoncurrentVersionExpiration操作删除对象的非当前版本，删除的对象无法恢复。	行为与启用了版本控制的桶相同。

基于特定日期的生命周期规则

使用Transition和Expiration操作时，您可以指定执行操作的日期（该日期必须符合ISO8601格式，时间始终为UTC午夜0点）。如果指定一个过去的日期，所有适用对象会立即执行操作。

不支持使用OBS控制台创建基于日期的生命周期规则。

警告

基于日期的操作并非一次性操作。即使过了该日期后，只要规则状态为“已启用”，OBS仍会继续应用该基于日期的操作。

例如，假设您指定一个基于日期的过期操作来删除所有对象（假设规则中未指定任何筛选条件）。在指定日期，OBS会使存储桶中的所有对象过期。此外，OBS还会继续使您在存储桶中创建的所有新对象过期。要终止生命周期操作，您必须从生命周期配置中删除操作、禁用规则或从生命周期配置中删除规则。

基于对象期限的生命周期规则

使用Transition和Expiration操作时，您可以指定执行操作的时间段（自对象创建以来的天数）。指定天数后，OBS从第二天的UTC午夜0点开始计算时间。例如，如果对象的创建时间是2016年1月15日上午10:30 UTC，并且您在转换规则中指定了3天，则对象的转换日期将计算为2016年1月19日00:00 UTC。

须知

OBS仅为每个对象保持上次修改日期。例如，OBS控制台在对象Properties(属性)窗格中显示LastModified(上次修改日期)日期。最初创建新对象时，此日期反映对象的创建日期。如果您替换对象，则此日期会相应地更改。

使用NoncurrentVersionTransition和NoncurrentVersionExpiration操作时，您可将对象版本成为非当前版本之后（即自对象覆盖或删除以来）的天数指定为对指定对象执行操作的时间段。

使用NoncurrentVersionTransition和NoncurrentVersionExpiration操作指定天数时，OBS将规则中指定的天数与对象版本成为非当前版本的时间相加，按照得到的时间从第二天的UTC午夜0点开始计算时间。例如，桶中某个对象当前版本的创建时间是2016年1月1日上午10:30 UTC，如果对象版本在2016年1月15日上午10:30 UTC成为非当前版本，并且您在转换规则中指定了3天，那么对象的转换日期计算为2016年1月19日00:00 UTC。

说明

配置生命周期规则时，在同一个规则内或前缀包含关系的规则间，如果配置Transition和Expiration，那么Date或Days应保持一致。

使用方式

OBS支持通过控制台、API、SDK方式配置生命周期规则。

支持的使用方式	参考文档
控制台	配置生命周期规则
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶的生命周期配置 获取桶的生命周期配置 删除桶的生命周期配置

6.2 标签

使用场景

标签用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按多种维度（如用途、所有者或环境）对云资源进行分类，从而帮助您对资源进行使用量分析或成本核算等。

对于OBS，标签用于标识桶，对OBS中的桶进行分类，从而帮助您对OBS资源进行相关分析等。

当为桶添加标签时，该桶上所有请求产生的计费话单中均会带上标签，从而可以使用标签对话单报表做分类筛选，进行更详细的成本分析。例如：某个应用程序在运行过程会往桶里上传数据，我们可以用应用名称作为标签，设置到被使用的桶上。在分析计费话单时，就可以通过该标签来分析此应用的成本。

约束与限制

- OBS以键值对的形式来描述标签。桶标签key和value区分大小写，
- 一个桶默认最大拥有10个标签。每个标签有且只有一对键值。
- 同一个桶标签的键不能重复，但是值可以重复，并且可以为空。
- 设置桶标签后，大约需要等待3分钟才能生效。

使用方式

OBS支持通过控制台、API、SDK方式设置标签。

支持的使用方式	参考文档
控制台	配置标签
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶标签

6.3 桶清单

使用场景

桶清单功能可以定期生成桶内对象的相关信息，保存在CSV格式的文件中，并上传到您指定的桶中，方便您管理桶内对象。目标桶和源桶可以是同一个桶。

- 指定的对象内容包括对象版本、大小、上次修改时间、存储类别、标签、加密状态等。
- 支持对桶清单加密，加密方式为SSE-KMS方式。
- 支持设定桶清单的生成频率：每天或每周。
- 支持将生成的桶清单存储到指定的目标桶。

约束与限制

桶版本限制：

- 只有OBS 3.0的桶支持配置清单，目标桶无OBS版本限制。

清单数量限制：

- 一个桶最多支持10条桶清单。

源桶和目标桶限制：

- 桶清单配置的源桶和目标桶必须归属同一个账号。
- 桶清单配置的源桶和目标桶必须归属同一个区域。
- 桶清单中配置的目标桶不能开启桶默认加密。

功能限制：

- 只支持生成CSV格式的清单文件。
- 桶清单筛选条件目前仅支持设置为所有对象或指定前缀的对象。
- 同一个桶中多条清单规则的筛选条件不能彼此包含：
 - 如果已经存在针对桶中所有对象的规则，则无法再创建按对象名前缀筛选的规则。如需创建，要先删除针对所有对象的规则。
 - 如果已经存在按对象名前缀筛选的规则，则无法再创建针对桶中所有对象的规则。如需创建，要先删除所有按对象名前缀筛选的规则。
 - 如果已经存在某个按对象名前缀筛选的规则（如前缀ab），则无法再创建与其存在包含或被包含关系的规则（如前缀a或前缀abc）。如需创建，要先删除存在包含或被包含关系的规则。
- 桶清单加密方式目前只支持SSE-KMS。

权限限制：

- 清单文件使用OBS系统用户上传到目标桶，目标桶必须给OBS系统用户写权限。

其他：

- 暂不对桶清单功能收费，桶清单生成后只按照存量计费。

如何配置桶清单

在介绍如何配置桶清单之前，先简要介绍一下清单的源桶和目标桶：

- 源桶：源桶为配置清单的桶，生成的清单文件中包含的为源桶中的对象。
- 目标桶：生成的清单文件存放到目标桶中，目标桶和源桶可以是同一个桶，如果要在目标桶中对生成的清单文件进行分类，可以在清单配置中指定目标前缀，生成的清单文件则会存放到以该前缀开头的目录下面，如果不指定前缀，则所有清单文件会默认存放到BucketInventory的目录下面。
 - 目标存储桶的限制
 - 和源桶属于同一个账号。
 - 和源桶在同一区域中。
 - 必须配置相应的桶策略，向OBS系统用户授予写入对象的权限，有关授权的更多信息，请参见[为目标桶添加桶策略](#)。
 - 目标存储桶包括的文件
 - 清单文件列表。
 - Manifest文件，Manifest文件中包含某条清单配置的所有清单文件列表，有关Manifest文件更多信息，请参见[manifest文件](#)。

配置桶清单

您可以通过OBS控制台配置桶清单，也可以通过REST API进行配置。如果通过控制台进行配置，控制台会自动生成目标桶的桶策略；如果通过REST API进行配置，需要手动为目标桶配置桶策略。

1. 为目标桶添加桶策略。

通过在目标桶上添加桶策略，向OBS系统用户授予将清单文件写入到目标桶的权限，桶策略的格式如下，修改其中的destbucket为目标桶的桶名。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Sid": "1",
      "Principal": {"Service": "obs"},
      "Resource": ["destbucket/*"],
      "Action": ["PutObject"]
    }
  ]
}
```

2. 配置桶清单。

您可以使用多种方式配置桶清单，详情请参见[使用方式](#)。

清单文件中包含的内容

清单文件中的内容由设置的清单配置而定，所有可能的字段见[表6-2](#)。

表 6-2 清单中列出的对象元数据

元数据	说明
Bucket	源桶的桶名。

元数据	说明
Key	键值，即对象的名称，一个桶里的每个对象拥有唯一的对象键值。（清单文件中的对象名采用UTF-8的URL编码，需解码后才能使用）。
VersionId	对象的版本id（如果清单配置中的IncludedObjectVersions值为Current时，则不包含此字段）。
IsLatest	如果对象的版本为最新，则设置为True（如果清单配置中的IncludedObjectVersions值为Current时，则不包含此字段）。
IsDeleteMarker	在桶开启多版本时，如果删除一个对象，会生成一条新的对象元数据，并将该条元数据的IsDeleteMarker标记为true（如果清单配置中的IncludedObjectVersions值为Current时，则不包含此字段）。
Size	对象大小（以字节为单位）。
LastModifiedDate	对象创建日期或上次修改日期。
ETag	对象MD5的16进制摘要。ETag是对象内容的唯一标识，可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A，下载对象时ETag为B，则说明对象内容发生了变化。
StorageClass	对象的存储类别。
IsMultipartUpload	对象是否是多段上传对象。
ReplicationStatus	对象的跨区域复制状态。
EncryptionStatus	对象的加密状态。

生成的清单文件对象的名称

生成的清单文件对象的名称如下：

```
destinationPrefix/sourceBucketName/inventoryId/yyyy-MM-dd'T'HH-mm'Z'/files/UUID_index.csv
```

- destinationPrefix：清单配置中的目标前缀，生成的清单文件以此前缀开头，可用于对清单文件进行分类存放，如果没有设置则默认为BucketInventory。
- sourceBucketName：清单文件对应的源桶，可以防止当不同的源桶都配置了相同的目标桶时产生冲突。
- inventoryId：可以防止当一个桶的多条清单配置中配置了相同的目标桶时产生冲突。
- yyyy-MM-dd'T'HH-mm'Z'：生成本次清单文件时开始扫描存储桶的开始时间和日期，清单文件中可能不会生成在此时间之后上传的对象。
- UUID_index.csv：清单文件之一。

manifest 文件

当桶内对象很多的时候，单条清单配置可能会生成多个清单文件，生成清单文件需要一定时间。例如，桶内有20万个对象时，生成所有清单文件需要约1.5分钟。在全部清

单文件生成完成之后1到2小时会生成一个manifest.json文件，manifest.json文件中包含了该条清单本次生成的所有清单文件的相关信息，具体信息如下：

- sourceBucket：源桶的桶名
- destinationBucket：目标桶的桶名
- version：清单的版本
- fileFormat：清单文件的格式
- fileSchema：清单文件中包含的对象元数据字段
- files：所有清单文件列表
- key：清单文件名称
- size：清单文件的大小（以字节为单位）
- inventoriedRecord：文件中包含的清单记录条数

下面是一个简单的manifest.json文件示例。

```
{
  "sourceBucket": "user001",
  "destinationBucket": "bucket001",
  "version": "2019-01-03",
  "fileFormat": "CSV",
  "fileSchema": "Bucket,Key,Size,LastModifiedDate,ETag,StorageClass,IsMultipartUploaded,ReplicationStatus,EncryptionStatus",
  "files": [
    {
      "key": "inventory/user001/test_id/2019-01-03T12-28Z/files/0000016813AF58E66806C1E2D7F15155_1.csv",
      "size": 6705647390,
      "inventoriedRecord": 70585762,
    }
  ]
}
```

生成的manifest文件对象的对象名如下，各个字段的含义见[生成的清单文件对象的名称](#)：

```
destinationPrefix/sourceBucketName/inventoryId/yyyy-MM-dd'T'HH-mm'Z'/manifest.json
```

使用方式

OBS支持通过控制台、API、SDK方式设置桶清单。

支持的使用方式	参考文档
控制台	配置桶清单
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶清单

6.4 事件通知

📖 说明

事件通知功能已下线。如有疑问，请[提交工单](#)联系华为云客服。

6.5 用量统计

OBS支持在对象存储控制台中查看桶的存储、流量和请求的情况。

使用场景

OBS为您提供了如表6-3所示的几种统计分析项，帮助您了解如下信息：

表 6-3 用量统计说明

统计类别	统计项	说明
存储	存储总趋势	包含桶中所有对象的存储总量以及桶中的文件总个数趋势。
	存储类型趋势	分别统计桶中标准存储、低频访问存储和归档存储三种存储类别的存储容量趋势。不同的存储类别详细介绍请参考 存储类别 。
	存储各类数量趋势	分别统计桶中标准存储、低频访问存储和归档存储三种存储类别的个数趋势。不同的存储类别详细介绍请参考 存储类别 。
流量	流量趋势	包含公网/内网流入流量和公网/内网流出流量的趋势。
请求	请求趋势	包含请求总数和总请求平均时延趋势。
	请求次数	分别统计产生的GET请求、PUT请求和DELETE请求的次数，以及所有请求的次数总和。
	请求使用率	分别统计请求的成功率、有效率和中断率。
	状态码分布	分别统计产生请求后服务端返回的状态码的分布情况。不同的状态码详细介绍请参考 状态码说明 。

约束与限制

- 可查询的时间跨度最长为30天。
- “用量统计”非实时刷新，大约有1小时左右的时延。

使用方式

OBS支持通过控制台查看桶的用量统计，具体操作请参考[查看用量统计](#)。

7 数据访问

7.1 静态网站托管

注意事项

基于安全合规要求，华为云对象存储服务OBS禁止通过OBS的默认域名（[桶访问域名](#)或[静态网站访问域名](#)）使用静态网站托管功能，即使用上述域名从浏览器访问网页类型文件时，不会显示对象内容，而是以附件形式下载。

各区域将自以下两个时间点起生效：

自2022年1月1日起生效：华北-北京一、华北-北京四、华东-上海一、华东-上海二、华南-广州

自2022年3月25日起生效：中国-香港、亚太-曼谷、亚太-新加坡、非洲-约翰内斯堡、拉美-墨西哥城一、拉美-墨西哥城二、拉美-圣保罗一、拉美-圣地亚哥

通过自定义域名使用静态网站托管功能不受影响，仍然支持文件内容的直接预览。详情请参见[如何在浏览器中在线预览OBS中的对象？](#)

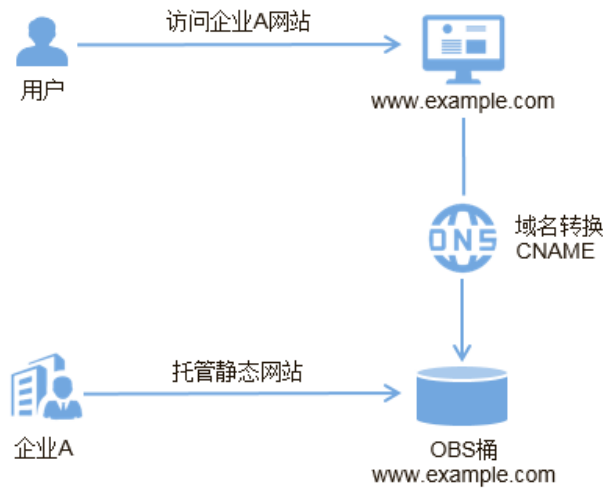
如果您的OBS桶开启了静态网站托管功能，请在配置CDN加速域名时勾选“静态网站托管”，这样用户访问时就不会显示桶中所有文件列表。

使用场景

OBS服务允许在桶上托管静态网站，支持索引页、错误页展示和页面跳转。您可以将静态网站文件上传至OBS的桶中，并对这些文件赋予匿名用户可读权限，然后将该桶配置成静态网站托管模式，就可以实现在OBS上托管静态网站了。

静态网站通常仅包含静态网页，以及可能包含部分可在客户端运行的脚本，如JavaScript、Flash等。相比之下，动态网站则依赖于服务器端处理脚本，包括PHP、JSP或ASP.Net等。OBS当前尚不支持服务器端运行脚本。

图 7-1 静态网站示意图



在设置了静态网站托管后，有以下两种方式访问OBS。

- 资源管理方式：资源管理方式通过普通的域名访问OBS。您可以使用API及SDK方式，结合带有终端节点（Endpoint）的OBS默认域名，实现普通的上传、下载、删除、列举等桶和对象操作。
- 静态网站方式：静态网站方式通过特定的域名访问OBS。使用静态网站访问时，根据配置好的域名托管规则，实现访问目录的索引页、错误页展示和访问页面资源的跳转。

修改静态网站托管设置后会在两分钟内生效，静态网站托管设置生效后，用户可以通过如下形式的网址访问静态资源：

https://静态网站访问域名/对象名
http://静态网站访问域名/对象名

其中静态网站访问域名的结构为：BucketName.obs-website.Endpoint，其中BucketName为桶名，Endpoint为终端节点，各区域的终端节点详情请参见[地区和终端节点](#)。例如您想使用静态访问域名访问位于中国-香港的名为testbucket桶，中国-香港的Endpoint为ap-southeast-1.myhuaweicloud.com，则此时对应的静态网站访问域名应该为testbucket.obs-website.ap-southeast-1.myhuaweicloud.com。

📖 说明

尽量避免目标桶名中带有“.”，否则通过HTTPS访问时可能出现客户端校验证书出错。

为了使您的客户可以访问网站终端节点上的内容，您必须使您的所有内容公开可读。这样，您可以使用对象上的桶策略或ACL来授予必要的权限。

资源管理方式和静态网站方式之间的主要差异如下：

主要差异	资源管理方式	静态网站方式
访问控制	同时支持公共内容和私有内容	仅支持公开可读的内容
错误消息处理	返回XML格式的错误响应	返回HTML文档

主要差异	资源管理方式	静态网站方式
重定向支持	不适用	同时支持对象级和桶级重定向
支持的请求	支持所有桶和对象操作	仅支持对象上的GET和HEAD请求
对桶根级的GET和HEAD请求的响应	返回桶中对象key的列表	返回配置中指定的索引文档

重定向简介

在使用静态网站托管功能时，OBS还支持配置重定向请求，即您可以将特定的请求或所有请求实施重定向。

当网站结构调整、网站地址变化或者网站的扩展名发生变化时，用户使用旧的网站地址（比如收藏夹中的地址）访问网站会访问失败，用户只能得到404页面错误信息。此时网站配置了重定向后，让访问这些域名的用户跳转到设定的页面以避免404错误访问。

重定向典型的应用场景包括：

- 重定向所有请求到另外一个站点。
- 设定特定的重定向规则，对特定的请求实施重定向。

设置静态网站托管

概述

如果要为静态网站托管配置桶，则要将网站配置添加到桶。该配置包含以下信息：

索引文档

当您键入诸如http://example.com的URL时，您不是在请求特定的页面。在这种情况下，Web服务器将提供默认页面，该页面包含存储了请求的网站内容的目录。此默认页面称为索引文档，大多数情况下，它被命名为index.html。当您为网站托管配置桶时，您必须指定索引文档。当对根域或任何子文件夹发出请求时，OBS将返回此索引文档。

错误文档

如果出现错误，则OBS将返回HTML错误文档。对于4XX类错误，您可以选择性地提供您自己的自定义错误文档，您可以在该文档中向您的用户提供其他指南。

重定向所有请求

如果根域为example.com，且您需要响应对http://example.com和http://www.example.com的请求，则可以创建名为example.com和www.example.com的两个桶，仅在一个桶中保留网站内容（如example.com），将另一个桶配置为将所有请求重定向到example.com桶。

高级条件重定向

您可以根据请求中的特定对象名称或前缀，或者根据响应代码按条件对请求进行重定向。例如，假设您在桶中删除或重命名了某个对象。您可以添加一个将请求重定向到其他对象的路由规则。

通过设置重定向规则特定请求，设置语法格式如下：

```
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix></IndexDocument>
  <ErrorDocument>
    <Key>SomeErrorDocument.html</Key>
  </ErrorDocument>
  <RoutingRules>
    .....RoutingRules.....
  </RoutingRules>
</WebsiteConfiguration>
```

RoutingRules语法格式如下，中括号[]中的内容为可选项：

```
<RoutingRules> =
  <RoutingRules>
    <RoutingRule>...</RoutingRule>
    [ <RoutingRule>...</RoutingRule> ... ]
  </RoutingRules>

<RoutingRule> =
  <RoutingRule>
    [ <Condition>...</Condition> ]
    <Redirect>...</Redirect>
  </RoutingRule>

<Condition> =
  <Condition>
    [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
    [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
  </Condition>

<Redirect> =
  <Redirect>
    [ <HostName>...</HostName> ]
    [ <Protocol>...</Protocol> ]
    [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith> ]
    [ <ReplaceKeyWith>...</ReplaceKeyWith> ] [
    <HttpRedirectCode>...</HttpRedirectCode> ]
  </Redirect>
```

重定向所有请求Website配置元素和设定重定向规则Website配置元素，详情请参见[设置桶的网站配置请求消息元素](#)。

以下是一些示例：

例1：修改对象名前缀进行重定向

假设您的桶包含了以下对象：

index.html

docs/article1.html

docs/article2.html

现在，您决定将该文件夹从docs/重命名为documents/。修改后，将对前缀/docs的请求重定向到documents/。例如，对docs/article1.html的请求需要重定向到documents/article1.html。

在这种情况下，您可以将以下路由规则添加到网站配置中：

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <KeyPrefixEquals>docs/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

例2：把发往已删除的文件夹的请求重定向到指定页面

假设您删除了images/文件夹（即，您删除了对象名前缀为images/的所有对象）。您可以添加路由规则，该规则将把对象名前缀为images/的所有对象的请求重定向到名为folderdeleted.html的页面。

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <KeyPrefixEquals>images/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

例3：为HTTP错误进行重定向

假设在未找到请求的对象时，需要将请求重定向到www.example.com。您可以添加重定向规则，以便当返回HTTP状态代码404（未找到）时，站点访问者可重定向到www.example.com。以下示例也将在重定向中插入对象名前缀report-404/。例如，如果您请求页面ExamplePage.html，且它导致了HTTP 404错误，该请求将重定向到www.example.com上的页面report-404/ExamplePage.html。如果没有路由规则，且发生了HTTP错误404，将返回在配置中指定的错误文档。

```
<RoutingRules>
  <RoutingRule>
    <Condition>
      <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
    </Condition>
    <Redirect>
      <HostName>www.example.com</HostName>
      <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
```

索引文档支持

索引文档是当对网站的根或任何子文件夹发出请求时返回的网页。例如，如果用户在浏览器中输入http://www.example.com，则该用户没有请求任何特定页面。在这种情况下，OBS将提供索引文档，该文档有时也称为默认页面。

当您将您的桶配置为网站时，您应该提供索引文档的名称。您必须上传具有此名称的对象并将其配置为公开可读。

根级URL的尾部斜杠是可选的。例如，如果您将具有index.html的网站配置为索引文档，以下任意一个URL将返回index.html。

```
http://bucketname.obs-website.example.com/
http://bucketname.obs-website.example.com
```

在OBS中，对象在桶中是水平存储的；它不会像计算机上的文件系统那样提供任何分层组织。您可以通过使用表示文件夹结构的对象名称创建逻辑层级结构。例如，考虑具有三个对象的桶和以下对象名。

- sample1.jpg
- photos/2006/Jan/sample2.jpg
- photos/2006/Feb/sample3.jpg

虽然它们没有按任何物理分层组织进行存储，但您可以从对象名推断以下逻辑文件夹结构。

- sample1.jpg对象位于桶的根级
- sample2.jpg对象位于photos/2006/Jan子文件夹中。
- sample3.jpg对象位于photos/2006/Feb子文件夹中。

自定义错误文档支持

表7-1列出了发生错误时OBS返回的HTTP响应代码的子集。

表 7-1 错误码列表

HTTP错误代码	说明
301 永久移动	当用户将请求直接发送到OBS网站终端节点时，OBS返回 301 Moved Permanently (301 永久移动) 响应。
302 Found (302 已找到)	当OBS收到对不包含尾部斜杠的键x的请求时，它首先查找键名称为x的对象。如果未找到对象，则OBS确定该请求是针对子文件夹x发出的，并通过在末尾添加斜杠重定向请求并返回 302 Found (302 已找到) 。
304 Not Modified (304 未修改)	OBS用户请求标头If-Modified-Since、If-Unmodified-Since以确定所请求的对象是否与客户端保存的缓存副本相同。如果对象相同，网站终端节点将返回 304 Not Modified 响应。
403 禁止访问	当用户请求转换为不可公开读取的对象时，网站终端节点的响应包含 403 Forbidden 。对象所有者必须使用桶策略或ACL使该对象公开可读。

HTTP错误代码	说明
404 未找到	<p>由于以下原因，网站终端节点的响应包含404 Not Found：</p> <ul style="list-style-type: none">• OBS确定网站URL引用了不存在的对象键• OBS推断该请求针对不存在的索引文档• 在URL中指定的桶不存在• 在URL中指定的桶存在，但它未配置为网站 <p>您可以创建为404 Not Found返回的自定义文档。确保该文档已上传到配置为网站的桶，且网站托管配置已设置为使用该文档。</p>
500 Service Error	当出现内部服务器错误时，网站终端节点的响应包含 500 Service Error 。
503 服务不可用	当OBS确定您需要降低请求频率时，网站终端节点的响应包含 503 Service Unavailable (503 服务不可用) 。

网站访问所需的权限

当您桶配置为网站时，您必须使要提供的对象公开可读。要执行此操作，您需要编写向每个人授予GetObject权限的桶策略。在网站终端节点上，如果用户请求的对象不存在，则OBS返回HTTP响应代码404 (Not Found)。如果该对象存在，但您未授予对对象的读取权限，则网站终端节点将返回HTTP响应代码403 (Access Denied)。用户可以使用该响应代码推断特定对象是否存在。如果您不需要此行为，则不要启用对桶的网站支持。

说明

静态网站托管场景需要设置匿名用户访问权限，当匿名用户访问静态网站文件时，会产生公网流出流量费用和请求费用。

以下示例桶策略向每个人授予了访问指定文件夹中的对象的权限。有关桶策略的更多信息，请参见[桶策略](#)。

```
{
  "Statement": [{
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": {"ID": "*" },
    "Action": ["GetObject"],
    "Resource": ["example-bucket/*" ]
  }
]
```

须知

桶策略仅适用于桶拥有者拥有的对象。如果桶包含非桶拥有者拥有的对象，则应使用对象ACL对这些对象授予公有READ权限。

您可以通过桶策略或对象ACL授予对您的对象的公开读取权限。如果要使用ACL使对象公开可读，您可以向Everyone授予READ权限，如以下授权元素所示。您可以将此授权元素添加到对象ACL。有关管理ACL的信息，请参见[ACL权限控制方式](#)。

```
<Grant>
  <Grantee>
    <Canned>Everyone</Canned>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
```

与其他功能的关系

功能	静态网站托管与该功能的关系	参考文档
跨域资源共享	默认情况下，托管在OBS的静态网站仅允许响应同域网站的访问请求。通过对保存静态网站资源的OBS桶进行跨域资源共享配置，可以实现跨域访问功能。	跨域资源共享
自定义域名配置	OBS支持通过自定义域名访问托管在OBS上的静态网站，用户无需修改网站代码，即可使用原域名访问网站。	自定义域名配置 使用自定义域名托管静态网站
防盗链	用户将网站托管在OBS后，为了防止网站数据被其他人盗链，OBS支持基于HTTP header中表头字段referer的防盗链方法。	防盗链

使用方式

OBS支持通过控制台、API、SDK方式配置静态网站托管。

支持的使用方式	参考文档
控制台	配置静态网站托管
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶的网站配置

7.2 数据回源

一般情况下，当客户端访问OBS时，如果OBS中没有被访问的数据，将会返回404错误。OBS提供数据回源功能，可以在被请求的数据不存在时，通过回源规则从源站获取对应数据。

镜像回源

用户为桶定义镜像回源规则后，如果客户端访问OBS桶中不存在的资源，且该资源符合镜像回源规则，OBS将以镜像回源的方式去数据源站获取资源，将该资源上传到OBS中并返回给客户端。整个过程不中断业务，适用于客户源站无缝迁移数据到OBS，用户可以在无感知的情况下，低成本地迁移业务到OBS上来。镜像回源流程如图7-2所示。

图 7-2 镜像回源流程



约束与限制

桶版本限制：

- 仅桶版本号为3.0及以上的桶支持镜像回源。

时间限制：

- 修改镜像回源规则，需要5分钟之后才能生效。

区域限制：

数据回源支持的区域请参见[功能总览](#)。

规则数量限制：

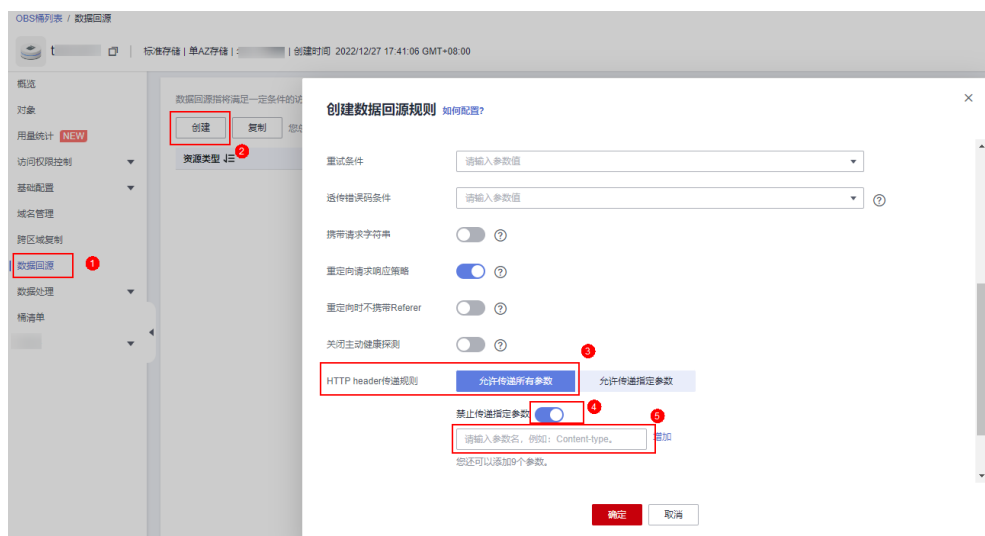
- 最多创建10条镜像回源规则。

功能限制：

- 不支持匿名用户配置镜像回源规则。
- 不支持给并行文件系统配置镜像回源规则。
- 不支持OBS静态网站，即通过OBS静态网站域名下载对象出现404，不会触发镜像回源规则。
- 不支持在镜像回源规则中指定当前OBS桶域名，即源桶与目标桶不能是同一个桶。
- 私有桶的镜像回源当前仅支持部分云厂商。
- 不支持源站使用Transfer-Encoding: chunked方式传输数据，即从源站下载对象的响应中必须包含Content-Length头域来指明源对象大小。

禁止源站使用Transfer-Encoding: chunked配置方式：创建/编辑数据回源规则 > HTTP header传递规则 > 允许传递所有参数 > 打开“禁止传递指定参数”开关 > 输入“Accept-Encoding”。

图 7-3 配置方式



权限限制:

- 需要使用IAM增加Tenant Administrator权限后才能设置、获取和删除镜像回源规则。
- 镜像回源需使用IAM创建云服务委托，委托OBS获取源站数据。委托需要给OBS服务授予obs:object:PutObject、obs:object:GetObject、obs:bucket:ListBucket、obs:object:AbortMultipartUpload权限。
- 如果桶开启了SSE-KMS服务端加密功能，对OBS的云服务委托中还需要使用IAM配置kms:cmk:get、kms:cmk:list、kms:cmk:create、kms:dek:create、kms:dek:crypto、kms:dek:crypto权限。

其他限制:

- 镜像回源暂时不收费。
- 不允许多条镜像回源规则配置的文件名前缀存在相互包含或重叠的关系，即对同一个对象的请求不能同时匹配两条镜像回源规则。

创建镜像回源规则

OBS支持通过控制台、API方式创建镜像回源规则。

支持的使用方式	参考文档
控制台	创建镜像回源规则
API	设置镜像回源规则

7.3 域名管理

7.3.1 自定义域名配置

使用场景

当用户需要将网站中的文件迁移到OBS，并且不想修改网页的代码（即保持网站的链接不变）时，可以使用自定义域名配置方案。

例如用户A的网站域名为www.example.com，网站文件为abc.html，网站链接为：
http://www.example.com/abc.html。配置流程如下：

1. 在OBS上创建一个桶，并上传abc.html网站文件到该桶中。
2. 通过OBS控制台，将www.example.com这个自定义的域名配置在已创建桶上。
3. 在域名服务器上，添加**CNAME规则**，将www.example.com映射成桶域名。
4. http://www.example.com/abc.html请求到达OBS后，OBS会找到www.example.com和桶域名的映射，转换变成访问桶的abc.html文件。即对http://www.example.com/abc.html的访问，经过OBS处理后，实际访问的是http://桶域名/abc.html。

约束与限制

桶版本限制：

- 桶版本号为3.0及以上的桶支持自定义配置域名功能。桶版本号可以在OBS管理控制台上，进入桶概览页后，在“基本信息”中查看。

域名数量限制：

- 每个桶默认最多配置20个自定义域名。

功能限制：

- OBS自定义域名配置暂时不支持HTTPS访问自定义域名，只支持HTTP访问自定义域名。

客户自定义域名配置成功后，如果想使用HTTPS进行访问，需同时使用CDN，通过CDN管理控制台进行HTTPS证书管理，即可使用HTTPS访问。

CDN管理控制台HTTPS证书管理方式，详情请参见[HTTPS配置](#)。

- 一个自定义域名只能配置到一个桶域名上。
- 暂不支持配置中文域名。
- 配置的自定义域名后缀目前支持的范围为2~6个英文大小写字母。
- 按照工信部要求，您配置自定义域名的桶如果在以下区域，需要提前完成[ICP备案](#)。

包括：华北-北京一、华北-北京四、华东-上海一、华东-上海二、华南-广州

📖 说明

如果使用了加速域名，为防止访问OBS桶中对象变成强制下载，配置自定义域名和加速域名后还需要参考[CDN加速场景下访问OBS桶内对象变成强制下载](#)进行处理。

使用方式

OBS支持通过控制台、API、SDK方式配置自定义域名绑定。

支持的使用方式	参考文档
控制台	配置自定义域名
API	设置桶的自定义域名

8 数据安全

8.1 服务端加密

使用场景

当启用服务端加密功能后，用户上传对象时，数据会在服务端加密成密文后存储。用户下载加密对象时，存储的密文会先在服务端解密为明文，再提供给用户。

OBS支持以下三种方式的服务端加密，三种方式都采用了行业标准的AES256加密算法。

- KMS托管密钥的服务端加密（SSE-KMS）

用户首先需要在KMS中创建密钥（或者使用KMS提供的默认密钥），当用户在OBS中上传对象时使用该密钥进行服务端加密。

在使用SSE-KMS方式加密时，您可以在创建桶时开启默认加密，开启后，所有上传到桶中的对象都会被加密；也可以在已创建的桶中根据需要开启默认加密，开启后，新上传到桶中的对象会被加密。

OBS仅会对开启桶默认加密之后上传的对象进行加密，不会改变开启前已经有对象的加密状态。关闭默认加密，也不会影响桶中已有对象的加密状态，关闭默认加密后可在上传对象时进行单独加密。

您可以通过控制台、API、SDK、OBS Browser+方式配置SSE-KMS加密。

- 客户提供加密密钥的服务端加密（SSE-C）

OBS使用用户提供的密钥和密钥的MD5值进行服务端加密。

您可以通过API、SDK方式配置SSE-C加密。

- OBS根密钥派生密钥的服务端加密（SSE-OBS）

OBS使用OBS根密钥派生的密钥对用户数据进行服务端加密。

您可以通过控制台配置SSE-OBS加密。

约束与限制

- 一个对象每次上传，只支持一种服务器端加密方式。
- 当桶或者桶内对象开启了SSE-KMS服务端加密功能，需要使用IAM为请求者配置kms:cmk:get、kms:cmk:list、kms:cmk:create、kms:dek:create、kms:dek:crypto权限，才能上传下载对象。

📖 说明

1. SSE-KMS服务端加密方式支持中国-香港、亚太-新加坡、拉美-墨西哥城一、拉美-圣保罗一、华南-广州、非洲-约翰内斯堡、亚太-曼谷、西南-贵阳一、亚太-雅加达、土耳其-伊斯坦布尔区域。
2. SSE-OBS服务端加密方式仅支持亚太-曼谷、亚太-雅加达区域。

背景知识

对于SSE-KMS方式，KMS通过使用硬件安全模块 (HSM) 保护密钥安全的托管，帮助用户轻松创建和控制加密密钥。用户密钥不会明文出现在HSM之外，避免密钥泄露。对密钥的所有操作都会进行访问控制及日志跟踪，提供所有密钥的使用记录，满足监督和合规性要求。

注意事项

桶加密开关关闭后，访问加密对象必须使用HTTPS协议。

使用方式

OBS支持通过控制台、API、SDK、OBS Browser+方式设置服务端加密。

支持的使用方式	参考文档
控制台	使用服务端加密方式上传文件 桶KMS加密
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	服务端加密SSE-KMS方式 服务端加密SSE-C方式 设置桶的加密配置
OBS Browser+	-

8.2 WORM

OBS提供WORM (Write Once Read Many) 功能，即一次写入多次读取，实现对象在用户指定的时间段内不能被删除和篡改。目前OBS支持桶级和对象级的WORM功能，提供合规模式一种对象锁定模式。

使用场景

OBS提供合规模式的WORM功能，合规模式是指如果您给一个对象应用了WORM，那么任何用户在指定时间内都不能覆盖或删除受保护的受保护的版本，包括账号中的根用户。

根据作用目标的范围不同，WORM可以分为桶级WORM和对象级WORM。桶级WORM以桶为单位锁定，保护范围覆盖桶中的所有对象。对象级WORM以对象为单位锁定，保护范围仅覆盖与WORM策略绑定的对象。无论是对象级WORM策略还是桶级

WORM策略，都要在开启桶级WORM开关后，才能生效，否则不会保护桶内对象。WORM策略只对策略生效之后新上传的对象生效。当对象同时被桶级WORM和对象级WORM覆盖时，优先遵从对象级WORM的保护策略。

注意事项

- 开启了桶级WORM开关，系统将自动开启多版本，并且不允许暂停多版本。WORM保护是基于对象版本号的，配置WORM的版本受到WORM保护，没有配置WORM的版本可正常删除。例如，test.txt 001受到WORM保护。此时再次上传同名文件，产生新的对象版本test.txt 002，test.txt 002并未配置WORM，那么test.txt 002就不受保护可以正常删除。当您下载对象时，不指定版本号下载的是最新对象，也就是test.txt 002。
- 处于保护中的对象不会被生命周期删除，但是能正常进行存储类别转换。受保护的對象不再受保护后，如果满足生命周期过期规则，将会被系统删除。
- 如果在创建桶时未开启WORM，那么后续不支持再开启WORM及配置WORM。如果您配置不了WORM，可能是由于您在创建桶时未开启WORM或者您的桶是在WORM特性上线前创建的，此时，如果您还需要使用WORM，请重新创建桶并开启WORM。
- 桶级WORM开关一旦打开，则无法为该桶禁用桶级WORM开关，也无法暂停桶的版本控制，但可以禁用桶级默认WORM策略。
- 开启桶级WORM开关的桶暂不支持跨区域复制能力。
- 在客户销户、欠费超期情况下，WORM未到期数据会被删除而不做保留。
- 不支持迁移场景。
- WORM不阻止修改对象元数据。

使用方式

OBS支持通过控制台、API方式进行WORM相关操作。

支持的使用方式	参考文档
控制台	创建桶 上传对象 配置WORM
API	创建桶 配置桶级默认WORM策略 获取桶级默认WORM策略 配置对象级WORM保护策略 通过 获取对象元数据 获取对象级WORM保护策略

8.3 跨域资源共享

跨域资源共享（Cross Origin Resource Sharing，CORS）是由W3C标准化组织提出的一种网络浏览器的规范机制，定义了一个域中加载的客户端Web应用程序与另一个域中的资源交互的方式。而在通常的网页请求中，由于同源安全策略（Same Origin Policy，SOP）的存在，不同域之间的网站脚本和内容是无法进行交互的。

OBS支持CORS规范，允许跨域请求访问OBS中的资源。CORS配置会在两分钟内生效。

OBS支持[静态网站托管](#)，而只有当对该桶设置了合理的CORS配置，OBS中保存的静态网站才能允许响应另一个跨域网站的请求。

须知

OBS系统提供的默认配置为根域名跨域访问允许所有域访问，会使客户端存在被攻击的风险。

客户端可通过在自己的桶内创建配有自己规则的crossdomain.xml，并在其flash代码中添加Security.loadPolicyFile("https://bucket.obs.ap-southeast-1.myhuaweicloud.com/crossdomain.xml")，从而避免被攻击的风险。其中，bucket.obs.ap-southeast-1.myhuaweicloud.com需根据实际情况替换为自己桶的访问域名。

背景知识

跨域是指不同域名之间相互访问。

跨域访问是浏览器出于安全考虑而设置的一个限制，即同源策略。由于JavaScript同源策略的限制，A域名下的JavaScript无法操作B域名或C域名下的对象。

同协议、同域名（或IP）、以及同端口视为同一个域。两个页面的协议、域名和端口（如果指定了端口）相同，则视为同源。为了更好的理解是否同源，您可以查看[表8-1](#)中相对https://support.huaweicloud.com/dir/test.html的同源检测示例。

表 8-1 同源检测示例

URL	访问结果	原因
https://support.huaweicloud.com/dir/other.html	成功	协议、域名、端口都相同
https://support.huaweicloud.com/dir/inner/other.html	成功	协议、域名、端口都相同
http://support.huaweicloud.com/dir/test.html	失败	域名、端口相同，协议不同
https://support.huaweicloud.com:81/dir/test.html	失败	协议、域名相同，端口不同
https://help.huaweicloud.com/dir/test.html	失败	协议、端口相同，域名不同

使用场景

CORS的典型应用场景包括：

- 通过CORS支持，使用JavaScript和HTML5来构建Web应用，直接访问OBS中的资源，而不再需要代理服务器做中转。

- 使用HTML5中的拖拽功能，直接向OBS上传文件，展示上传进度，或是直接从Web应用中更新内容。
- 托管在不同域中的外部网页、样式表和HTML5应用，现在可以引用存储在OBS中的Web字体或图片，让这些资源能被多个网站共享。

使用方式

OBS支持通过控制台、API、SDK方式配置跨域资源共享。

支持的使用方式	参考文档
控制台	配置跨域资源共享
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶的CORS配置

8.4 多版本控制

使用场景

利用多版本控制，您可以在一个桶中保留多个版本的对象，使您更方便地检索和还原各个版本，在意外操作或应用程序故障时快速恢复数据。

默认情况下，OBS中新创建的桶不会开启多版本功能，向同一个桶上传同名的对象时，新上传的对象将覆盖原有的对象。

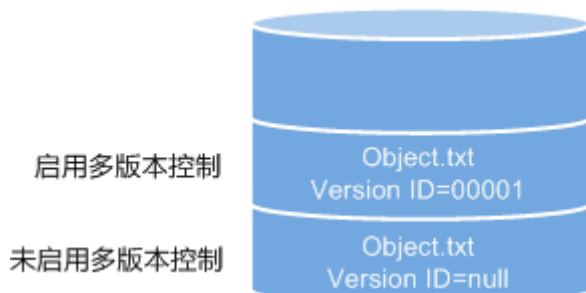
约束与限制

开启了桶级**WORM**开关，系统将自动开启多版本，并且不允许暂停多版本。

开启多版本控制

- 桶中已有对象版本ID（空）和内容都不会变化。再次上传该同名对象，对象版本示意图如[图8-1](#)所示。

图 8-1 多版本对象示意图（已有对象）



- 新上传对象，OBS自动为每个对象创建唯一的版本号。上传同名的对象将以不同的版本号同时保存在OBS中，如[图8-2](#)所示。

图 8-2 多版本对象示意图（新对象）

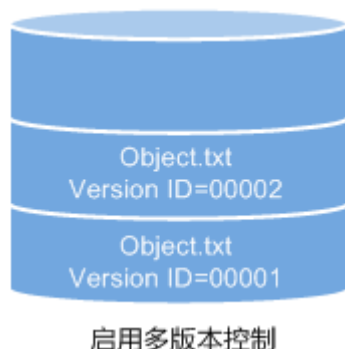


表 8-2 版本说明

版本	描述
最新版本	多版本控制开启后，同名对象多次操作，每次操作都会对应一个版本号进行保存。最后一次操作保存的版本号，为最新版本。
历史版本	多版本控制开启后，同名对象多次操作，每次操作都会对应一个版本号进行保存。除最后一次外的，其他保存的版本号为历史版本。

- 列出桶内对象列表时默认列出最新对象列表。
- 可以指定版本号下载对象，不指定版本号默认下载最新的对象。
- 可以选中目标对象，并单击右侧的“删除”删除对象。对象被删除后，OBS将插入一个删除标记，对象在“已删除对象”列表中呈现。此时如果访问该对象，会返回404错误。

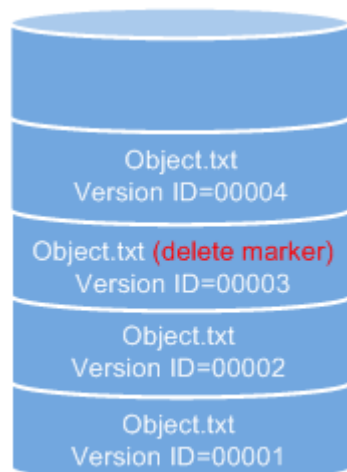
图 8-3 删除标记示意图



- 删除带删除标记的版本可恢复该对象。
- 在“已删除对象”列表，选中对象，可指定版本号彻底删除指定版本对象。
- 一个对象只会显示在对象列表或已删除对象列表中，不会同时出现。

例如，上传一个对象A后，将其删除，对象A将显示在已删除对象列表中。如果再次上传同名对象A，同名对象A会显示在对象列表中，显示在已删除对象列表中的原对象A将不会存在。对象A版本示意图如图8-4所示。

图 8-4 删除后再上传同名对象的版本示意图



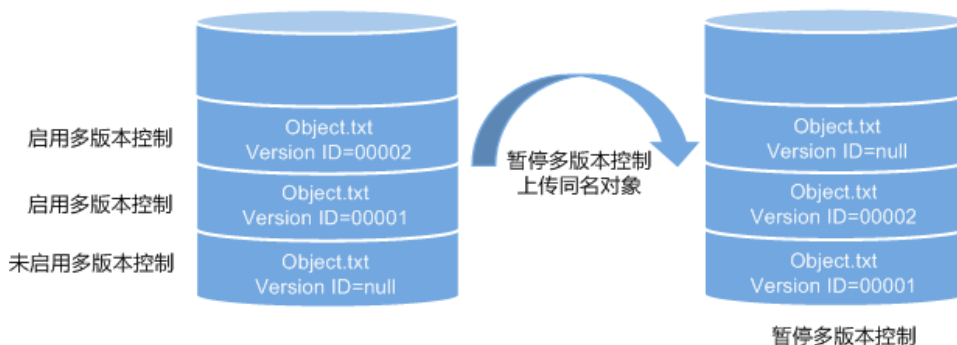
启用多版本控制

- 除带删除标记的版本外，其他每个版本的对象存储在OBS中均需要进行计费。

暂停多版本控制

多版本控制一旦启动，不可以关闭，只能暂停使用。暂停后，新上传的对象版本号为空。如果之前有空版本号同名对象，则会覆盖该带空版本号的对象。

图 8-5 暂停多版本控制后的对象版本示意图



当不需要对桶内对象进行版本控制时，可以暂停多版本控制：

- 历史版本将继续保留在OBS中，如果这些历史版本你不再需要，请手动删除。
- 仍可以指定版本号下载对象，不指定版本号默认下载最新的对象。
- 除带删除标记的版本外，其他历史版本的对象存储在OBS中仍需要进行计费。

暂停与未启用的区别

暂停多版本控制后，删除对象时，无论此对象是否存在历史版本，将会产生一个版本号为空的删除标记。而未启用多版本控制时，则不会产生删除标记。

说明

多版本开启后，由于对象的每个历史版本都被保存下来，每个版本都会占用存储空间，OBS会对所有版本收取存储费用。请谨慎启用多版本开关，以免产生额外的存储费用。

使用方式

OBS支持通过控制台、API、SDK方式配置多版本控制。

支持的使用方式	参考文档
控制台	配置多版本控制
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶的多版本状态

8.5 跨区域复制

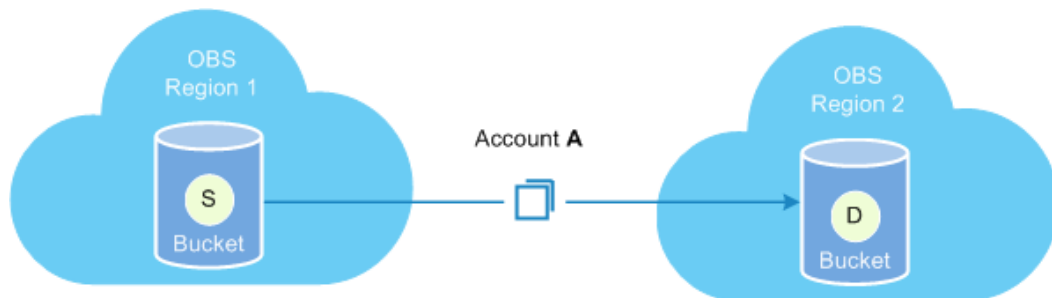
使用场景

跨区域复制能够为用户提供跨区域数据容灾的能力，满足用户数据复制到异地进行备份的需求。

跨区域复制是指通过创建跨区域复制规则，将一个桶（源桶）中的数据自动、异步地复制到不同区域的另外一个桶（目标桶）中，源桶和目标桶必须属于同一个账号，暂不支持跨账号复制。

在配置跨区域复制规则时，您可以按前缀匹配请求复制部分对象，也可以请求复制桶中的所有对象。复制到目标桶的对象是源桶中对象的精确副本。它们具有相同的对象名称和元数据，包括：对象内容、大小、最后修改时间、创建者、版本号、用户定义的元数据以及ACL。默认情况下复制对象的存储类别，与源对象保持一致。您也可以为复制对象指定一个存储类别。

图 8-6 跨区域复制示意图



- 合规性要求

虽然OBS默认跨多个地理位置较远的可用区存储数据，但是合规性要求所规定的数据存储距离可能更远。通过跨区域复制，可以在远距离OBS区域之间复制数据以满足这些合规性要求。

- **最大限度减少延迟**
客户需要在多地访问相同的OBS资源。为了最大限度缩短访问对象时的延迟，您可以使用跨区域复制，在离客户较近的区域中创建对象副本。
- **数据复制**
由于业务原因，您需要将OBS数据从一个区域的数据中心迁移至另一个区域的数据中心。
- **数据备份与容灾**
出于对数据安全性以及可用性的考虑，您希望对所有写入OBS的数据，都在另一个区域的数据中心显式地创建一个备份，以防止在数据发生不可逆损毁时，有安全、可用的备份数据。
- **易维护**
您在两个不同OBS区域中具有分析同一组对象的计算集群。您可能选择在这些区域中维护对象副本。

须知

OBS帮助您将存储于华为云对象存储的业务数据复制到指定Region，但华为云并不感知您的数据的具体内容，也不对您使用本服务的合法合规性负责。如果您的复制涉及跨境传输，请确保您使用本服务符合法律法规的要求。

复制的内容

启用跨区域复制规则后，符合以下条件的对象会复制到目标桶中：

- 新上传的对象（归档存储或深度归档存储对象除外）。
- 有更新的对象，比如对象内容有更新或者已复制成功的对象ACL有更新。
- 桶中的历史对象（需要开启“同步历史对象”功能，且归档存储或深度归档存储对象除外）。

例如，7月28号，源桶开启了跨区域复制，上传了对象A和B，此时目标桶同步复制了A和B。7月29号，源桶上传了对象C，这个时候目标桶只会复制C，不会再次复制A和B。7月30号，源桶修改了对象A，目标桶只会复制A，不会再次复制B和C。

📖 说明

跨区域复制不支持SSE-C加密的对象复制。

约束与限制

桶版本限制

- 桶版本号为3.0及以上的桶支持跨区域复制功能。桶版本号可以在OBS管理控制台上，进入桶概览页后，在“基本信息”中查看。

源桶和目标桶状态限制：

- 源桶和目标桶的多版本控制状态必须保持一致。

功能限制：

- 源桶中归档存储或深度归档存储类型的对象不支持跨区域复制到目标桶。
- 如果目标桶所属区域不支持存储类别，则复制后对象的存储类别均为标准存储类型。
- OBS目前仅支持一个源桶同时复制到一个目标桶，不支持一个源桶同时复制到多个目标桶。允许修改目标桶，但修改目标桶会更改所有已创建规则的目标桶。
- 开启跨区域复制功能后，源桶将不再支持追加写对象。
- 同一个源桶只能创建一条复制所有对象的跨区域复制规则，或多条（最多100条）按前缀匹配的跨区域复制规则。

时间限制：

- 跨区域复制不保证时效性，配置跨区域复制规则后，可能会出现对象不会立即进行复制的情况，请耐心等待。

区域限制：

- 源桶和目标桶必须属于不同的区域，同区域的桶不能进行数据复制。
- 跨区域复制支持的区域情况如#obs_41_0034/fig1152534613411所示。√表示两个区域可连通，x表示两个区域不可连通。

图 8-7 跨区域复制支持区域

目的区域 \ 源区域	华北—北京四	华东—上海一	华东—上海二	华北—北京二	华北—乌兰察布一	西南—贵阳一	华南—广州	中国—香港	亚太—曼谷	拉美—圣保罗一	非洲—约翰内斯堡	拉美—墨西哥城一	拉美—墨西哥城二	亚太—新加坡	亚太—雅加达	土耳其—伊斯坦布尔	华南—深圳	华南—广州友好环境	拉美—圣地亚哥
华北—北京四	-	✓	✓	✓	✓	✓	✓	×	×	×	×	×	×	×	×	×	×	×	×
华东—上海一	✓	-	✓	×	✓	✓	✓	×	×	×	×	×	×	×	×	×	×	×	×
华东—上海二	✓	✓	-	✓	✓	✓	✓	×	×	×	✓	×	×	×	×	×	×	×	×
华北—北京二	✓	×	✓	-	✓	✓	✓	×	×	×	×	×	×	×	×	×	×	×	×
华北—乌兰察布一	✓	✓	✓	✓	-	✓	✓	×	×	×	×	×	×	×	×	×	×	×	×
西南—贵阳一	✓	✓	✓	✓	✓	-	✓	×	×	×	×	×	×	×	×	×	×	×	×
华南—广州	✓	✓	✓	✓	✓	✓	-	×	×	×	×	×	×	×	×	×	×	×	×
中国—香港	×	×	×	×	×	×	×	-	✓	×	×	×	×	×	×	×	×	×	×
亚太—曼谷	×	×	×	×	×	×	×	×	-	×	×	×	×	×	×	×	×	×	×
拉美—圣保罗一	×	×	×	×	×	×	×	×	×	-	×	×	×	×	×	×	×	×	×
非洲—约翰内斯堡	×	×	✓	×	×	×	×	×	×	×	-	×	-	×	×	×	×	×	×
拉美—墨西哥城一	×	×	×	×	×	×	×	×	×	×	×	-	×	×	×	×	×	×	×
拉美—墨西哥城二	×	×	×	×	×	×	×	×	×	×	×	×	-	×	×	×	×	×	×
亚太—新加坡	×	×	×	×	×	×	×	×	×	×	×	×	×	-	×	×	×	×	×
亚太—雅加达	×	×	×	×	×	×	×	×	×	×	×	×	×	×	-	×	×	×	×
土耳其—伊斯坦布尔	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	-	×	×	×
华南—深圳	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	-	×	×
华南—广州友好环境	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	-	×
拉美—圣地亚哥	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	-

同步历史对象相关：

- 启用跨区域复制功能之前上传的对象，默认不会被复制到目标桶，除非开启了“同步历史对象”功能。
- 启用历史对象复制后，修改跨区域复制配置可能导致历史对象不复制，建议在历史对象复制完成前不要改变该桶的跨区域复制配置。

多版本控制相关：

- 当源桶、目标桶多版本控制处于开启或暂停状态，且跨区域复制规则处于开启状态时，在源桶中不指定版本删除对象，目标桶会同步删除此对象。

- 在启用跨区域复制过程中，如果您修改目标桶的多版本控制状态，会导致对象复制失败；如果您尝试修改源桶多版本控制状态，必须先删除复制配置，然后才能进行修改。

权限相关：

- 源桶或目标桶都需要一直保证桶所有者具有读写权限，以确保数据能够成功同步。如果源桶或目标桶的读写权限错误，导致系统没有读源对象或者写目标对象的权限，这种对象将一直复制不成功，即使将权限修改正确后，也不会重新复制。
- 不建议您对目标桶中的副本对象进行删除、覆盖或者修改ACL操作，此类操作可能导致目标桶中对象最新版本或者对象访问控制权限与源区域不一致。
- 如果已复制成功的源对象的ACL发生变化，在该对象匹配的复制策略未发生变化的情况下，这些变化会同步复制到对象副本，但已复制成功的历史对象不会同步源对象的ACL变化。

其他约束与限制

- 源桶中的对象只能被复制到一个目标桶中，且复制过去的对象不能再被复制到另外一个目标桶。例如有两个不同区域的桶A和桶B，桶A数据可以复制到桶B中，桶B数据也可以复制到桶A中，但桶B中存储的桶A数据的副本不会复制，同理桶A中存储的桶B数据的副本也不会复制。
- 在启用跨区域复制过程中，如果您删掉OBS云服务委托，会导致对象复制状态为FAILED。

使用方式

OBS支持通过控制台、API、SDK、obsutil方式配置跨区域复制。

支持的使用方式	参考文档
控制台	配置跨区域复制
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶的跨区域复制配置
obsutil	复制对象

8.6 防盗链

使用场景

一些不良网站想要扩充自己站点内容，但却不想增加成本，于是会盗用其他网站的链接。一方面损害了原网站的合法利益，另一方面又加重了服务器的负担。因此，产生了防盗链技术。

在HTTP协议中，通过表头字段referer，网站可以检测目标网页访问的来源网页。有了referer跟踪来源，就可以通过技术手段来进行处理，一旦检测到来源不是本站即进行阻止或者返回指定的页面。防盗链就是通过设置Referer，去检测请求来源的referer字

段信息是否与白名单或黑名单匹配，如果与白名单匹配成功则允许请求访问，否则阻止请求访问或返回指定页面。

为了防止用户在OBS的数据被其他人盗链，OBS支持基于HTTP header中表头字段referer的防盗链方法。OBS同时支持访问白名单和访问黑名单的设置。

Referer规则如下：

- 白名单Referer/黑名单Referer输入的字节数不能超过1024个字符。
- Referer格式：
 - Referer可以设置多个，多个Referer换行隔开；
 - Referer参数支持通配符（*）和问号（?），通配符可代替0个或多个字符，问号可代替单个字符；
 - 如果下载时Referer头域包含了http或https，则Referer设置必须包含http或https。
- 白名单Referer为空，黑名单Referer不空时，允许所有黑名单中指定网站以外的其他网站的请求访问目标桶中的数据。
- 白名单Referer不为空，黑名单Referer为空或不空时，允许在白名单且不在黑名单中的网站的请求访问目标桶中的数据。

说明

当白名单Referer与黑名单Referer内容有交集时，交集部分Referer被禁止。例如：当白名单Referer与黑名单Referer输入框中的referer字段都有“https://www.example.com”时，系统是阻止该网站的请求访问的。

- 黑名单Referer与白名单Referer都为空时，默认允许所有网站的请求访问目标桶中的数据。
- 判断用户是否有对桶及其内容访问的四种权限（读取权限、写入权限、ACL读取权限、ACL写入权限）之前，需要首先检查是否符合referer字段的防盗链规则。

白名单和黑名单设置方法如下：

- 白名单设置

通过设置白名单，用户可以指定允许来自白名单列表中的网站的请求访问，否则将进行拦截。

通过浏览器地址栏直接访问资源的请求，即HTTP请求中referer为空的场景，用户可以通过在Condition的"Referer"中添加\${null}字段，用于指定是否允许referer为空的请求访问。

白名单设置具体参考如下policy设置进行实现：

```
"Statement": [
  {
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"ID": ["*"]},
    "Action": "*",
    "Resource": ["bucket/*"],
  },
  {
    "Sid": "2",
    "Effect": "Deny",
    "Principal": {"ID": ["*"]},
    "Action": ["*"],
    "Resource": ["bucket/*"],
    "Condition": {
      "StringNotEquals": {
        "Referer": ["http://www.example01.com", "${null}"]
      }
    }
  }
]
```

如果按照此方式设置，只有referer为"www.example01.com"和referer为空的请求可以对桶bucket中的资源进行操作。

- 黑名单设置

使用访问黑名单功能可以参考如下policy设置进行实现：

```
"Statement":[
  {
    "Sid":"1",
    "Effect":"Deny",
    "Principal":{"ID":["*"]},
    "Action":["*"],
    "Resource":["bucket/*"],
    "Condition":{"
      "StringEquals":
        {"Referer":["http://www.example01.com","http://www.example02.com"]}
    }
  }
]
```

如果按照此方式设置，当referer为"www.example01.com"或"www.example02.com"时，就不能对桶bucket中的资源进行操作。

使用方式

OBS支持通过控制台、API方式设置防盗链。

支持的使用方式	参考文档
控制台	配置防盗链
API	设置防盗链白名单

9 数据处理

9.1 图片处理

图片处理特性（Image Processing）是对象存储服务（Object Storage Service, OBS）为用户提供稳定、安全、高效、易用、低成本的图片处理服务，包括：图片瘦身、图片剪切、图片缩放、图片水印、格式转换等。

图片处理特性提供OBS管理控制台和RESTful API接口的使用方法，使用户可以在任何时间和地点对存储在对象存储中的图片进行处理，并且可以快速获取到处理后的图片。

详情请参见[图片处理特性指南](#)。

9.2 在线解压

OBS支持ZIP包上传后自动解压。

您可以将批量文件打包成ZIP包后上传至OBS，上传之后压缩包可以自动解压。

使用场景

- 用户将ZIP包上传到OBS桶后，希望ZIP包能被自动解压并存放在指定目录。
- OBS批量上传文件时有一些限制，参见[OBS批量上传文件](#)。当您需要一次性上传大量的小文件时，可以将所有文件打包成ZIP包，上传后即可解压，省时省力。

前提条件

使用在线解压特性的账号或子账号应具备Tenant administrator的用户角色。

约束与限制

权限限制：

- 在线解压源需使用IAM创建云服务委托，委托OBS获取桶内数据，上传数据到桶内。委托需要给OBS服务授予obs:object:PutObject、obs:object:GetObject、obs:object:AbortMultipartUpload权限。

区域限制：

- 在线解压功能支持的区域请参见[功能总览](#)。

包及文件大小限制：

- 单个ZIP包大小上限为**1GB**。
- 单个ZIP包内最多可包含65536个文件。
- ZIP包内，单个文件解压后最大为40GB。

时间限制：

- ZIP包解压任务最大时长为10分钟。

功能限制：

- ZIP包的后缀必须为zip。
- ZIP包名称中不可带有中文标点符号、特殊符号和特殊编码。
- 针对ZIP包嵌套的场景，在线解压策略的事件类型需设置为ObjectCreated:*或ObjectCreated:CompleteMultipartUpload。
- 目前不支持解压任务完成后给用户发送消息通知。
- 目前仅支持解压压缩算法为deflated的ZIP包，此处的压缩算法与WINRAR等软件提供的Store、Normal等压缩方法不等同。
- 解压路径加上压缩包解压出文件的文件名，总字符长度不可超过512，否则会导致解压失败。
- 目前不支持解压加密ZIP包。

注意事项

- 建议设置精准的前缀，同一个桶下的不同触发条件不能包含或重叠，触发条件指策略配置项中的事件、前缀以及后缀。
例如，同一个桶下的两个解压策略event-0001和event-0002，假设event-0001的前缀配置为aa，则event-0002的前缀不可配置为aaaa，因为两者具有包含关系。
- 当前缀置空时，解压策略将默认匹配该桶下的全部ZIP包，可能会触发循环执行。
例如，ZIP包“AA.zip”中包含ZIP包“BB.zip”。如果前缀置空，当“AA.zip”解压完成后，系统会继续对解压出的“BB.zip”进行解压。设置前缀可以规避该问题。
- 目标目录：ZIP包解压生成文件的存放目录，如果不设置，函数计算会将生成的文件存放到当前桶的主目录下。
- 如果ZIP包中包含非UTF-8或GB 2312编码的文件名或文件夹（文件目录）名，可能导致解压生成文件的文件名或文件目录名出现乱码、解压过程中断等情况，请使用UTF-8或GB 2312编码命名您的文件或文件目录。
- 上传归档存储的ZIP包，不会触发ZIP包解压，如果桶已开启归档数据直读功能，则可以触发ZIP包解压。上传深度归档存储的ZIP包，不会触发ZIP包解压。
- ZIP包解压处理单个压缩包的最长处理时间是10分钟，超过10分钟未完成会解压失败。

使用方式

OBS支持通过控制台、API方式配置ZIP包在线解压。

支持的使用方式	参考文档
控制台	创建在线解压策略
API	设置在线解压策略

10 监控与日志

10.1 监控

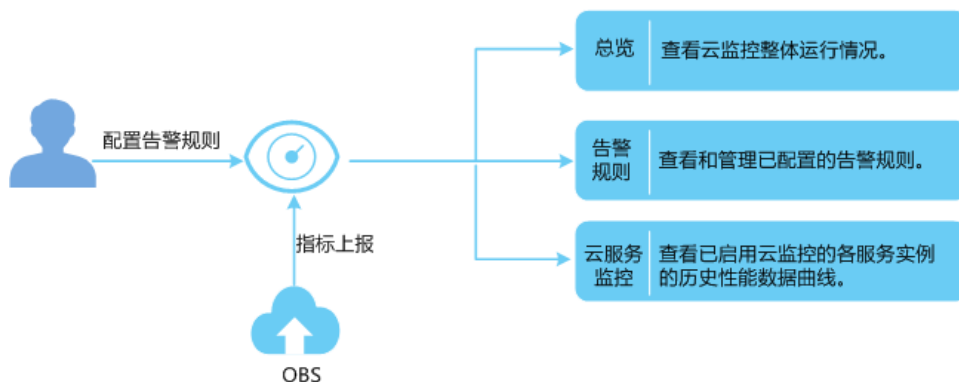
使用场景

用户在使用OBS的过程中会不断发送PUT、GET类的请求，产生上传、下载流量，也可能遇到服务端返回的错误响应。为使用户更好地掌握OBS桶的状态，华为云提供了云监控服务（Cloud Eye）。用户可使用该服务监控自己的OBS桶，执行自动实时监控、告警和通知操作，帮助您实时掌握桶中所产生的请求、流量和错误响应等信息。

云监控服务不需要开通，会在用户创建资源（如OBS桶）后自动启动。

关于云监控服务的更多介绍，请参见[云监控服务产品介绍](#)。

图 10-1 云监控示意图



设置告警规则

在自动实时监控的基础上，用户可以在云监控服务中设置告警规则，规定在某些特殊情况出现时向用户发送告警通知。

设置OBS监控信息告警规则的方法，请参见[创建告警规则](#)。

查看监控指标

云监控服务对OBS的监控指标进行实时监控，用户可以在云监控服务控制台查看各项指标的详细监控数据。

查看OBS监控指标详情的方法，请参见[查看云服务监控指标](#)。

监控指标

相关信息请参见[对象存储服务监控指标说明](#)。

10.2 审计

使用场景

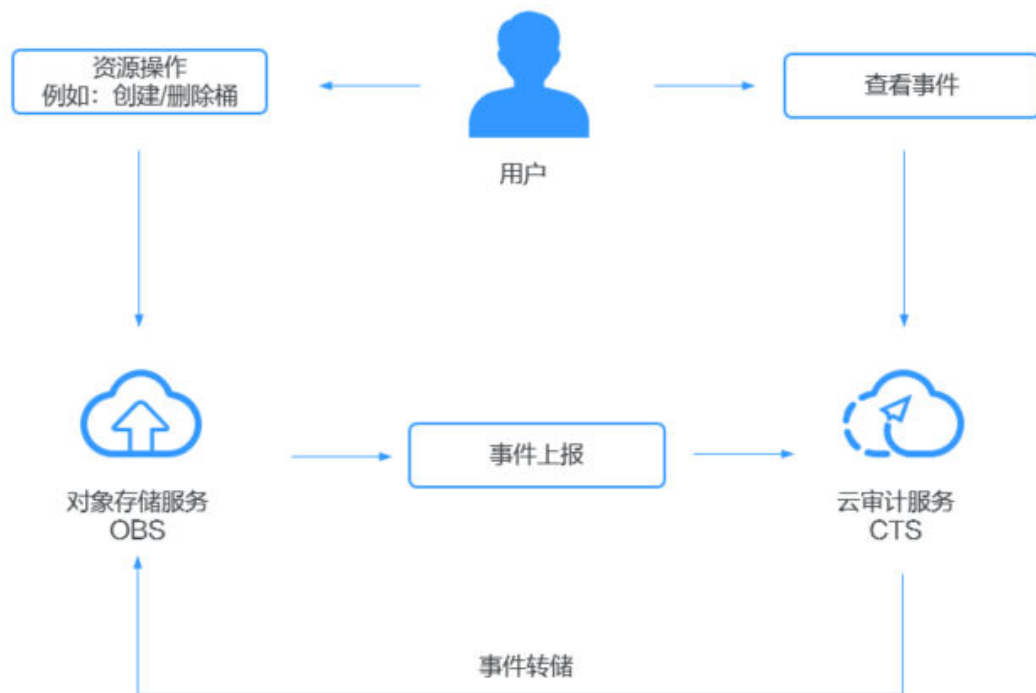
云审计服务（Cloud Trace Service, CTS），是华为云安全解决方案中专业的日志审计服务，提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。

用户开通云审计服务并创建和配置追踪器后，CTS可记录OBS的管理事件和数据事件用于审计。

CTS的详细介绍和开通配置方法，请参见[CTS快速入门](#)。

CTS支持追踪的OBS管理事件和数据事件列表，请参见[审计](#)。

图 10-2 云审计示意图



10.3 日志记录

使用场景

出于分析或审计等目的，用户可以开启日志记录功能。通过访问日志记录，桶的拥有者可以深入分析访问该桶的用户请求性质、类型或趋势。当用户开启一个桶的日志记录功能后，OBS会自动对这个桶的访问请求记录日志，并生成日志文件写入用户指定的桶中。

日志文件存放位置需要在开启桶日志功能时指定，可以存放到您拥有的，且与开启日志功能的桶位于同一区域的任一存储桶，当然也包括开启日志功能的桶本身。为了更有效的管理日志，建议您将日志存放到不同的桶中。当日志存储桶为开启日志功能的桶本身时，OBS将为写入该存储桶的日志创建额外的日志，这将增加您的存储账单费用，且由于额外日志的存在，将使您在寻找需要的日志时更加困难。

须知

- 桶日志上传会产生相应的PUT请求费用，PUT请求费用的具体说明请参考[OBS计费说明](#)。
- 日志记录设置成功后，大约15分钟后可在日志存储桶中查看到桶的操作日志。

OBS支持对桶的访问请求创建并保存访问日志记录，可用于进行请求分析或日志审计。

由于日志存储在OBS中也会占用用户租用的OBS存储空间，即意味着将产生额外的存储费用，默认情况下，OBS不会为用户的桶收集访问日志。

由于日志文件是OBS产生，并且由OBS上传到存放日志的桶中，因此OBS需要获得委托授权，用于上传生成的日志文件。所以在配置桶日志记录前，需要先到统一身份认证服务生成一个对OBS服务的委托，并在配置日志记录时添加该委托。默认情况下，在为委托配置权限时只需设置日志存储桶的上传对象权限，示例如下（其中mybucketlogs为日志存储桶的桶名）。如果日志存储桶开启了默认加密功能，还需要委托同时具有日志存储桶所在区域的KMS Administrator权限。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "obs:object:PutObject"
      ],
      "Resource": [
        "OBS:*:*:object:mybucketlogs/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

以下所示为在日志存储桶生成的桶访问日志文件记录：

```
787f2f92b20943998a4fe2ab75eb09b8 bucket [13/Aug/2015:01:43:42 +0000] xx.xx.xx.xx
787f2f92b20943998a4fe2ab75eb09b8 281599BACAD9376ECE141B842B94535B
REST.GET.BUCKET.LOCATION
- "GET /bucket?location HTTP/1.1" 200 - 211 - 6 6 "-" "HttpClient" - -
```

每个桶访问日志都包含以下信息：

表 10-1 Bucket Logging 格式

名称	示例	含义
BucketOwner	787f2f92b20943998a4fe2a b75eb09b8	桶的ownerId
Bucket	bucket	桶名
Time	[13/Aug/2015:01:43:42 +0000]	请求时间戳 (UTC)
Remote IP	xx.xx.xx.xx	请求IP
Requester	787f2f92b20943998a4fe2a b75eb09b8	请求者ID <ul style="list-style-type: none">当使用账号或IAM用户发起请求时，此ID为请求者所属账号的账号ID。当使用匿名用户发起请求时，取值为Anonymous。
RequestID	281599BACAD9376ECE141 B842B94535B	请求ID
Operation	REST.GET.BUCKET.LOCATI ON	操作名称 常见的Operation及其描述 请参见表10-2。
Key	-	对象名
Request-URI	GET /bucket?location HTTP/1.1	请求URI
HTTPStatus	200	返回码
ErrorCode	-	错误码
BytesSent	211	HTTP响应的字节大小
ObjectSize	-	对象大小 (bytes) 说明 <ul style="list-style-type: none">删除对象时，日志不会记录删除对象的大小。删除对象的日志中ObjectSize值为0。如果返回错误码4XX，ObjectSize值为“-”，不展示具体大小。
TotalTime	6	服务端处理时间 (ms)

名称	示例	含义
Turn-AroundTime	6	总请求时间 (ms) 说明 TotalTime和Turn-AroundTime是一个参数的两种叫法, 两者无区别。
Referer	-	请求的referer头域
User-Agent	HttpClient	请求的user-agent头域
VersionID	-	请求中带的versionId
STSLogUrn	-	联邦认证及委托授权信息
StorageClass	STANDARD_IA	当前的对象存储类型
TargetStorageClass	GLACIER	通过转换后的对象存储类型
DentryName	12456/file.txt	<ul style="list-style-type: none"> 对于并行文件系统, 是文件/目录的内部标识, 由父目录inode编号与文件/目录名称组成。 对于对象桶, 该字段为 "-"。
IAMUserID	8f3b8c53d29244a780084f2b8c106c32	IAM用户ID。 当使用匿名用户发起请求, 记为Anonymous。

表 10-2 常见的 Operation

Operation	描述	Operation	描述
REST.GET.SERVICE	获取桶列表	REST.GET.ENCRYPTI ON	获取桶的加密配置
REST.PUT.BUCKET	创建桶	REST.DELETE.ENCY PTION	删除桶的加密配置
REST.HEAD.BUCKE T	查看桶	REST.PUT.OTM_DIRE CT_COLD_ACCESS	设置桶归档对象直 读策略
REST.GET.BUCKET VERSIONS	列举桶内对象	REST.GET.OTM_DIRE CT_COLD_ACCESS	获取桶归档对象直 读策略
REST.GET.BUCKET	获取桶元数据	REST.DELETE.OTM_D IRECT_COLD_ACCES S	删除桶归档对象直 读策略
REST.GET.BUCKET. LOCATION	获取桶区域位置	REST.PUT.BUCKET.W EBSITE	设置桶的网站配置

Operation	描述	Operation	描述
REST.DELETE.BUCKET	删除桶	REST.GET.BUCKET.WEBSITE	获取桶的网站配置
REST.PUT.POLICY	设置桶策略	REST.DEL.BUCKET.WEBSITE	删除桶的网站配置
REST.GET.POLICY	获取桶策略	REST.PUT.BUCKET.CORS	设置桶的CORS配置
REST.DELETE.POLICY	删除桶策略	REST.GET.BUCKET.CORS	获取桶的CORS配置
REST.PUT.ACL	设置桶ACL、设置对象ACL	REST.DEL.BUCKET.CORS	删除桶的CORS配置
REST.GET.ACL	获取桶ACL、获取对象ACL	REST.OPTIONS.BUCKET	OPTIONS桶
REST.PUT.LOGGING_STATUS	设置桶日志管理配置	REST.OPTIONS.OBJECT	OPTIONS对象
REST.GET.LOGGING_STATUS	获取桶日志管理配置	REST.PUT.OBJECT	PUT上传
REST.PUT.BUCKET.LIFECYCLE	设置桶的生命周期配置	REST.POST.OBJECT	POST上传
REST.GET.LIFECYCLE	获取桶的生命周期配置	REST.COPY.OBJECT	复制对象
REST.DEL.LIFECYCLE	删除桶的生命周期配置	REST.GET.OBJECT	获取对象内容
REST.PUT.VERSIONING	设置桶的多版本状态	REST.HEAD.OBJECT	获取对象元数据
REST.GET.VERSIONING	获取桶的多版本状态	REST.DELETE.OBJECT	删除对象
REST.GET.BUCKET.STORAGE.POLICY	设置桶默认存储类别	REST.TRANSITION.STORAGECLASS.OBJECT	修改对象存储类别
REST.PUT.BUCKET.STORAGE.POLICY	获取桶默认存储类别	OP_MULTIPLE_DELETEOBJECT	批量删除对象
REST.PUT.REPLICATION	设置桶的跨区域复制配置	REST.POST.RESTORE	恢复归档存储对象
REST.DELETE.REPLICATION	删除桶的跨区域复制配置	REST.APPEND.OBJECT	追加写对象
REST.GET.REPLICATION	获取桶的跨区域复制配置	REST.MODIFY.OBJECT.META	修改对象元数据

Operation	描述	Operation	描述
REST.PUT.TAGGING	设置桶标签	REST.TRUNCATE.OBJECT	截断对象
REST.GET.TAGGING	获取桶标签	REST.RENAME.OBJECT	重命名对象
REST.DEL.TAGGING	删除桶标签	REST.GET.UPLOADS	列举桶中已初始化多段任务
REST.PUT.BUCKET_QUOTA	设置桶配额	REST.POST.UPLOADS	初始化多段上传任务
REST.GET.BUCKET_QUOTA	获取桶配额	REST.PUT.PART	上传段
REST.GET.BUCKET_STORAGEINFO	获取桶存量信息	REST.COPY.PART	拷贝段
REST.PUT.BUCKET_INVENTORY	设置桶清单	REST.GET.UPLOAD	列举已上传的段
REST.GET.BUCKET_INVENTORY	获取桶清单、列举桶清单	REST.POST.UPLOAD	合并段
REST.DELETE.BUCKET_INVENTORY	删除桶清单	REST.DELETE.UPLOAD	取消多段上传任务
REST.PUT.CUSTOMDOMAIN	设置桶的自定义域名	REST.CLEAR.EXPIRE.UPLOAD	清理过期的段
REST.GET.CUSTOMDOMAIN	获取桶的自定义域名	REST.DELETE.CUSTOMDOMAIN	删除桶的自定义域名
REST.PUT.ENCRYPTION	设置桶的加密配置	-	-

使用方式

OBS支持通过控制台、API、SDK方式配置日志记录。

支持的使用方式	参考文档
控制台	配置桶的日志记录
SDK	OBS支持多种语言SDK，请从 SDK概述 页面选择进入对应的开发指南查阅。
API	设置桶日志管理配置

11 并行文件系统

并行文件系统（Parallel File System）是OBS提供的一种经过优化的高性能文件系统，提供毫秒级别访问时延，以及TB/s级别带宽和百万级别的IOPS，能够快速处理高性能计算（HPC）工作负载。

作为对象存储服务的子产品，并行文件系统支持用户按照标准的OBS接口读取数据。也支持通过部署在弹性云服务器中的PFS客户端（obsfs工具），按照POSIX文件语义读写数据；通过obsfs用户可以将创建的并行文件系统挂载到云端Linux服务器上并能像操作本地文件系统一样对并行文件系统内的文件和目录进行在线处理，包括：创建、删除文件/目录，重命名文件/目录，修改写文件等操作。

详情请参见[并行文件系统特性指南](#)。