对象存储

开发者指南

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

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

【商标声明】

腾讯云

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

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

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

文档目录

开发者指南
- 创建请求
  - 创建请求概述
- 存储桶
  - 存储桶概述
  - 创建存储桶
  - 删除存储桶
- 对象
  - 对象概述
  - 存储类型
    - 存储类型概述
    - 智能分层存储简介
    - 深度归档存储简介
  - 上传对象
    - 简单上传
    - 分块上传
    - 预签名授权上传
  - 下载对象
    - 简单下载对象
    - 预签名授权下载
  - 列出对象键
  - 复制对象
    - 简单复制
    - 分块复制
  - 删除对象
    - 删除单个对象
    - 删除多个对象
- 数据管理
  - 生命周期管理
    - 生命周期概述
    - 生命周期配置元素
    - 配置生命周期
  - 托管静态网站
  - 清单功能概述
  - 存储桶标签概述
  - 对象标签概述
  - 事件通知
  - 数据检索
    - Select 概述
    - Select 命令

SQL 函数
保留字段
数据类型
运算符
日志管理
日志管理概述
日志管理限制
使用 COS 存储云产品日志
数据容灾
版本控制
版本控制概述
删除标记
使用版本控制
版本控制配置
存储桶复制
存储桶复制概述
复制行为说明
存储桶复制配置
多 AZ 特性概述
数据安全
服务端加密概述
存储桶加密概述
访问管理
访问权限配置说明
访问控制概述
访问控制基本概念
COS 授权与身份认证流程
最小权限原则说明
访问策略评估流程
权限控制方式介绍
存储桶策略
用户策略
ACL
基于标签管理项目资源
如何选择授权方式
访问策略语言
访问策略语言概述
生效条件
请求方式介绍
使用永久密钥访问 COS
使用临时密钥访问 COS
使用预签名 URL 访问 COS

匿名访问 COS
使用 CDN 加速访问
CDN 加速概述
CDN 加速配置
COS 源站域名升级须知
单链接限速
批量处理
批量处理概述
批量处理操作
批量复制对象
批量恢复归档对象
批量删除对象
管理批量处理任务
全球加速
全球加速概述
内网全球加速
数据工作流
数据工作流概述
监控与告警

开发者指南
创建请求
创建请求概述

最近更新时间：2024-11-11 16:15:42

基本概念

腾讯云对象存储 COS 是一个使用 HTTP/HTTPS 协议访问的 Web 存储服务，您可以使用 REST API 或 COS SDK 来访问 COS。
当您发起访问 COS 请求，需要经过 COS 认证和鉴权后，才可以对资源进行操作。因此根据身份是否可识别，访问 COS 的请求分两种类型：匿名请求和签名请求。

- 匿名请求：请求未携带 Authorization 或相关参数，又或者相关字符无法识别出用户身份特征，此时请求就会被视为匿名请求而进行鉴权。
- 签名请求：签名的请求需要在 HTTP 头部或者请求包中包含 Authorization 字段，该字段的内容是结合腾讯云的安全凭证 SecretID、SecretKey 和请求的一些特征值，通过加密算法生成。

如果您使用 COS SDK 访问，只需配置好您的安全凭证即可发起请求。使用 REST API 访问，则需要参见 请求签名 文档自行计算请求签名或通过 COS 签名工具 直接生成。
此外，请求路径中 COS 对于多斜线（例如，`bucket-125000000.cos.ap-beijing.myqcloud.com/test//1.jpg`）默认采取归并处理（即转换为，`bucket-125000000.cos.ap-beijing.myqcloud.com/test/1.jpg`），以兼容大部分客户的使用习惯。如用户需要保持多斜线不归并处理，可 联系我们 开通白名单。

获取安全凭证

腾讯云 访问管理（Cloud Access Management，CAM）为对象存储 COS 提供了账号和安全凭证的相关功能和服务，主要用于帮助客户安全管理腾讯云账户下的资源的访问权限。用户可以通过 CAM 创建、管理和销毁用户（组），并使用身份管理和策略管理控制其他用户使用腾讯云资源的权限。

主账号的安全凭证

在登录主账号后，可通过访问管理中的 云 API 密钥 页面管理和获取您的主账号安全凭证 SecretID 和 SecretKey 密钥。以下是一组密钥示例：

- 36 个字符的访问密钥 ID（SecretID）：**********
- 32 个字符的访问密钥 Key（SecretKey）：**********

访问密钥可用于标识唯一的账户，使用密钥签名后发送请求，腾讯云将识别请求发起者的身份，并在认证后，对身份、资源、操作、条件等进行鉴权，来判断是否允许执行此操作。

> **注意：**
> 主账号密钥具备其名下所有资源的所有操作权限，密钥的泄露可能造成您的云上资产损失，强烈建议您创建子账户并分配合理的权限，使用子账户的密钥创建请求来访问和管理资源。

子账号的安全凭证

当您需要在多个维度管理名下的用户和云资源时，您可以在主账号名下创建多个子账号，实现分管人员对资源权限的功能。有关创建子账号的说明请参阅访问管理 用户管理 的相关文档。
使用子账号发起 API 请求前，您需要为子账号创建安全凭证，随后子账号也将具备独特的密钥对，以便于识别身份。您可以对不同的子账号编写用户策略，来控制其对资源的访问权限。您也可以创建用户组，并对用户组添加统一的访问策略，便于对人员的分组和资源的统筹管理。

> **注意：**
> 子账号被分配对应权限后，可以创建或修改资源，此时资源仍然属于主账号，主账号需要支付该资源产生的费用。

临时安全凭证

除了使用主账号或子账号的安全凭证访问资源外，腾讯云还支持通过创建角色，并使用临时安全凭证管理腾讯云资源。有关角色的基本概念和使用方法，请参阅访问管理 角色管理 文档。

由于角色是一个虚拟身份，因此角色不具备永久密钥，腾讯云的 CAM 提供了一套 STS API 用于生成临时安全凭证。临时安全凭证通常只包括**有限策略**（操作、资源、条件）和**有限时间**（有效起止时间），因此生成的临时安全凭证可自行分发或直接使用。临时安全凭证的使用方法和示例详情请参阅 使用角色 文档，或参阅 创建角色 文档查看生成临时安全凭证的方式。

调用生成临时安全凭证的接口，您将获得一对临时密钥（tmpSecretId/tmpSecretKey）和一个安全令牌（sessionToken），其构成了可用于访问对象存储 COS 的安全凭证，示例如下：

- 安全令牌（sessionToken）：**********，长度和绑定的策略有关，最长不超过4096字节。
- 临时访问密钥 ID（tmpSecretId）：**********，最长不超过1024字节。
- 临时访问密钥 Key（tmpSecretKey）：**********，最长不超过1024字节。

该接口还会通过 `expiration` 字段返回临时安全凭证的有效时间，这意味着您只能在此时间内使用这组安全凭证来发起请求。腾讯云对象存储 COS 提供了简单的服务端 SDK 可用于生成临时密钥，可访问 COS STS SDK 获取。在得到临时安全凭证后，使用 REST API 发起请求，您需要在 HTTP 头部或 POST 请求包的 form-data 中传入 `x-cos-security-token` 字段标识该请求使用的安全令牌，再使用临时访问密钥对来计算请求签名。使用 COS SDK 发起访问请参考各 SDK 文档中的相关部分。

访问域名

REST API

我们在 地域和访问域名 文档中列出了可用于发起 REST API 访问的域名列表。
COS 推荐使用虚拟托管型域名访问存储桶，在发起 HTTP 请求时直接通过 `Host` 头部带入需要访问的存储桶，例如 `<BucketName-APPID>.cos.<Region>.myqcloud.com` 。使用虚拟托管型的域名实现了类似虚拟服务器「根目录」的功能，可用于托管类似 favicon.ico、robots.txt、crossdomain.xml 的文件，这些文件是很多应用程序在识别托管网站时，会默认从虚拟服务器「根目录」位置检索的内容。
您也可以使用路径型请求来访问存储桶，例如使用 `cos.<region>.myqcloud.com/<BucketName-APPID>/` 的路径来访问存储桶。相应的，请求 Host 和签名需要使用 `cos.<region>.myqcloud.com` ，我们的 SDK 默认不支持此种访问方式。

静态网站域名

当您开启存储桶的静态网站功能时，我们将分配一个虚拟托管型的域名供相关功能的使用。静态网站域名的表现与 REST API 有所不同，除了特定的索引页、错误页和跳转配置外，静态网站域名只支持 GET/HEAD/OPTIONS Object 等几种操作，不

支持上传或配置资源的操作。
静态网站域名例如：`<BucketName-APPID>.cos-website.<Region>.myqcloud.com`，您也可以通过控制台的存储桶「基础配置 – 静态网站配置」模块获取此域名。

内网访问

腾讯云对象存储 COS 的访问域名使用了智能 DNS 解析，通过互联网在不同的运营商环境下，我们会检测并指向最优链路供您访问 COS。
如果您在腾讯云内部署 CVM 服务用于内网访问 COS，那么首先需确保 CVM 与 COS 存储桶的所属地域相同，然后通过在 CVM 上使用 `nslookup` 命令解析 COS 域名，若返回内网 IP，则表明 CVM 和 COS 之间是内网访问，否则为外网访问。
如果您在腾讯云内部署的 CVM 服务，其地域与 COS 存储桶的所属地域不同，但属于 COS 可用地域范围，可通过 COS 内网全球加速域名访问文件，实现 CVM 与 COS 的跨地域内网访问。通过内网全球加速域名实现跨地域内网访问会产生额外的费用，详情请参见 内网全球加速 。

内网访问判断方法

相同地域内腾讯云产品访问，可实现内网连接，产生的内网流量不计费。因此选购腾讯云不同产品时，建议尽量选择相同地域，减少您的费用。

> **注意：**
> 公有云地域和金融云地域内网不互通。

确认是否内网访问请参考如下方法：
以腾讯 CVM 访问 COS 为例，判断是否使用内网访问 COS ，可以在 CVM 上 使用 `nslookup` 命令解析 COS 域名，若返回内网 IP，则表明 CVM 和 COS 之间是内网访问，否则为外网访问。

> **说明：**
> 内网 IP 地址一般形如 `10.*.*.*` 、 `100.*.*.*` ，VPC 网络一般为 `169.254.*.*` 等，这两种形式的 IP 都属于内网。

假设 `examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com` 为目标存储桶地址，执行 `nslookup` 命令后可以看到以下信息：

```
nslookup examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Server:   10.138.224.65
Address:  10.138.224.65  #53
Name:     examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Address:  10.148.214.13
Name:     examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Address:  10.148.214.14
```

其中 `10.148.214.13` 和 `10.148.214.14` 这两个 IP 就代表了是通过内网访问的 COS。

测试连通性

基本连通测试

COS 使用了 HTTP 协议对外提供服务，您可使用最基本的 `telnet` 工具来对 COS 访问域名的80端口发起访问测试。
通过外网访问的示例如下：

```
telnet examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com 80
Trying 14.119.113.22...
Connected to gz.file.myqcloud.com.
Escape character is '^]'.
```

通过同地域的腾讯云 CVM （基础网络）访问示例如下：

```
telnet examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com 80
Trying 10.148.214.14...
Connected to 10.148.214.14.
Escape character is '^]'.
```

通过同地域的腾讯云 CVM （VPC 网络）访问示例如下：

```
telnet examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com 80
Trying 169.254.0.47....
Connected to 169.254.0.47.
Escape character is '^]'.
```

无论处于何种访问环境，命令返回 `Escape character is '^]'.` 字段则说明可成功连通。

通过互联网测试的说明

由于通过互联网访问 COS 会经过运营商网络，运营商网络可能禁止您使用 ICMP 协议的 `ping` 或 `traceroute` 等工具来测试连通性，因此建议使用 TCP 协议的工具来测试连通性。

> **注意：**
> 通过互联网访问可能受到多种网络环境的影响，如有访问不畅的情况，请排查本地网络链路或联系当地运营商进行反馈。

若您的运营商允许 ICMP 协议，您可以使用 `ping` 、 `traceroute` 或 `mtr` 工具来检视您的链路状况。运营商不允许使用 ICMP 协议，您可以使用 `psping` （Windows 环境，请前往微软官网下载）或 `tcping` （跨平台软件）等工具进行时延测试。

通过内网测试的说明

如果您通过同地域的腾讯云 VPC 网络来访问对象存储 COS，则可能无法使用 ICMP 协议的 `ping` 或 `traceroute` 等工具来测试连通性。建议您使用基本连通测试中的 `telnet` 命令进行测试。
您亦可尝试使用 `psping` 或 `tcping` 等工具直接对访问域名的80端口进行时延测试，请在测试前确保已通过 `nslookup` 命令查询并确认访问域名已正确解析至内网地址。

存储桶
存储桶概述

最近更新时间：2024-09-13 17:33:31

简介

存储桶（Bucket）是对象的载体，可理解为存放对象的“容器”，且该“容器”无容量上限。对象以扁平化结构存放在存储桶中，无文件夹和目录的概念，用户可选择将对象存放到单个或多个存储桶中。

> **说明**
> 每个存储桶可容纳任意数量的对象，但同个主账号下存储桶数量最多仅能创建200个。

存储桶命名规范

存储桶的命名由存储桶名称（BucketName）和 APPID 两部分组成，两者以中划线“-”相连。存储桶的命名格式为 `<BucketName-APPID>` 。

- 存储桶名称（BucketName）：由用户手动输入的一串字符，命名规范如下：
  - 仅支持小写英文字母和数字，即[a-z，0-9]、中划线“-”及其组合。
  - 存储桶名称的最大允许字符受到 地域简称 和 APPID 的字符数影响，组成的完整请求域名字符数总计最多60个字符。例如请求域名 `12345678901234567890 1-1250000000.cos.ap-beijing.myqcloud.com` 总和为60个字符。
  - 存储桶命名不能以“-”开头或结尾。
- APPID：是您在成功申请腾讯云账户后所得到的账号，由系统自动分配，具有固定性和全局唯一性，可在 账号信息 中查看。通过控制台创建存储桶时，无需用户输入，而在使用工具、API、SDK 时则需要指定 APPID。

命名示例

例如存储桶 `examplebucket-1250000000` ，其中 examplebucket 为 BucketName（由用户手动输入的一串字符）；1250000000 为 APPID（用户申请腾讯云账号时，由系统自动分配）。
以下是有效的存储桶命名示例：

- examplebucket-1-1250000000
- mybucket123-1250000000
- 1-newproject-1250000000

存储桶所属地域

地域（Region）指对象存储 COS 的数据中心所在的地域。对象存储允许用户在不同地域上创建存储桶，您可以选择在离您业务最近的地域上创建存储桶，以满足低延迟、低成本以及合规性要求。
例如，当您的业务分布在华南地区，那么选择在广州地域创建存储桶可以提高对象的上传、下载速度。更多地域信息，请参见 地域和访问域名 文档。

> **注意**
> 地域是在创建存储桶时必须指定的，一旦指定之后将不允许更改。该存储桶下所有的对象都存储在对应的数据中心，目前不支持对象级别的地域设置。

权限类别

存储桶默认提供两种权限类型：公共权限和用户权限。

> **说明**
> - 当存储桶为私有读写或对指定账号授予用户权限，请求对象时需携带签名用于身份验证，关于签名说明，请参见 请求签名 。
> - 当存储桶为公有读私有写或公有读写时，请求对象时无需携带签名，匿名用户可直接通过链接访问对象，您的数据会存在泄露风险，请谨慎设置。

公共权限

公共权限包括：私有读写、公有读私有写和公有读写。其访问权限可通过对象存储控制台上的存储桶的**权限管理**进行修改，更多访问权限的说明，请参见 访问控制基本概念 。

- 私有读写
  只有该存储桶的创建者及有授权的账号才对该存储桶中的对象有读写权限，其他任何人对该存储桶中的对象都没有读写权限。存储桶访问权限默认为私有读写，推荐使用。
- 公有读私有写
  任何人（包括匿名访问者）都对该存储桶中的对象有读权限，但只有存储桶创建者及有授权的账号才对该存储桶中的对象有写权限。
- 公有读写
  任何人（包括匿名访问者）都对该存储桶中的对象有读权限和写权限，不推荐使用。

用户权限

主账号默认拥有存储桶的所有权限（即完全控制）。另外 COS 支持添加子账号有数据读取、数据写入、权限读取、权限写入，甚至完全控制的最高权限。
以下视频为您详细介绍存储桶权限：
观看视频

存储桶操作

用户可以通过腾讯云控制台、工具、API、SDK 等多种方式管理存储桶以及配置属性。例如，配置存储桶用于静态网站托管、配置存储桶的访问权限等。下面列举了部分功能配置指引，有关存储桶功能配置的更多信息，请参见 存储桶概述 。

- 创建存储桶
- 设置静态网站
- 设置访问权限
- 设置防盗链

相关说明

- 对象存储以扁平化结构来存放对象，无文件夹概念。详情请参见 对象概述 文档中的“文件夹和目录”部分。
- 每个主账号（即同一个 APPID）可以创建多个存储桶，数量上限是200个（不区分地域），但是存储桶中的对象数量没有限制。
- 在腾讯云中，同一个 APPID 下的存储桶名称是全局唯一的，不能重名。

- 存储桶一旦创建后，将无法重命名。您只能删除后重新创建再命名存储桶。
- 用户在创建存储桶时，请确认好所属地域，地域一旦设置后将无法修改。

创建存储桶

最近更新时间：2024-12-20 11:29:53

将文件存放到对象存储（Cloud Object Storage，COS）时，您需要先创建一个存储桶用于存放对象，您可以通过控制台、工具、API 或 SDK 的方式来创建存储桶。
创建存储桶后，您可将对象上传至该存储桶，此外还可以为存储桶配置其他功能，例如 设置静态网站 、设置存储桶标签 、设置存储桶加密 等，更多配置指引请参见 控制台概述 。

限制说明

- 同个主账号下仅允许创建最多200个存储桶。
- 存储桶一旦创建成功，存储桶名称和所属地域不能修改。
- 同一主账号下所有存储桶名称唯一且不支持重命名。
- 存储桶名称仅支持小写英文字母和数字，即[a-z，0-9]、中划线“-”及其组合。存储桶名称的最大允许字符受到 地域简称 和 APPID 的字符数影响，组成的完整请求域名字符数总计最多60个字符。例如请求域名 `12345678901234567890 1-1250000000.cos.ap-beijing.myqcloud.com` 总和为60个字符。
- 存储桶命名不能以“-”开头或结尾。

使用方法

使用对象存储控制台

您可以使用对象存储控制台创建存储桶，详情请参见 创建存储桶 控制台文档。

使用工具

您可以使用工具创建存储桶，例如 COSBrowser 工具、COSCMD 工具等，详情请参见 工具概览 。

使用 REST API

您可以直接使用 REST API 发起创建存储桶请求，详情请参见 PUT Bucket API 文档。

使用 SDK

您可以直接调用 SDK 创建存储桶，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- Node.js SDK
- PHP SDK
- Python SDK

- 小程序 SDK

删除存储桶

最近更新时间：2024-12-20 11:29:53

当您在某些情况下需要删除存储桶时，您可以通过控制台、工具、API 或 SDK 的方式来删除存储桶。

> **注意：**
> 删除存储桶后，将无法恢复，请谨慎操作。

限制说明

- 目前仅支持删除数据已被清空的存储桶，如果存储桶中仍有对象或文件碎片，将会删除失败。请在执行删除存储桶前确保存储桶内已清空对象，详情请参见 清空存储桶 。
- 当删除存储桶时，您需要确保操作的身份已被授权该操作，并确认传入了正确的存储桶名称（Bucket）和地域（Region）参数。
- 删除存储桶后，需要等待数小时（通常为1到2小时）才能再次创建同名的存储桶。

使用方法

使用对象存储控制台

您可以使用对象存储控制台删除存储桶，详情请参见 删除存储桶 控制台指南文档。

使用工具

您可以使用工具删除存储桶，例如 COSBrowser 工具、COSCMD 工具等，详情请参见 工具概览 。

使用 REST API

您可以直接使用 REST API 发起删除存储桶请求，详情请参见 DELETE Bucket API 文档。

使用 SDK

您可以直接调用 SDK 的删除存储桶方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

对象
对象概述

最近更新时间：2024-10-14 09:47:31

简介

对象（Object）是对象存储（Cloud Object Storage，COS）的基本单元，可理解为任何格式类型的数据，例如图片、文档和音视频文件等。存储桶（Bucket）是对象的载体，每个存储桶可容纳任意数量的对象。对象在 COS 中可指定为不同的存储类型，详情请参见 存储类型概述 。
每个对象都由对象键（ObjectKey）、对象值（Value）、和对象元数据（Metadata）组成。

- 对象键（ObjectKey）：对象键是对象在存储桶中的唯一标识，可以通俗的理解为文件路径。在 API、SDK 示例中，对象的命名格式为 `<ObjectKey>` 。
- 对象值（Value）：即上传的对象本身，可以通俗的理解为文件内容（Object Content）。
- 对象元数据（Metadata）：是一组键值对，可以通俗的理解为文件的属性，例如文件的修改时间、存储类型等，您可以在上传对象后对其进行查询。

对象键

腾讯云 COS 中的对象需具有合法的对象键，对象键（ObjectKey）是对象在存储桶中的唯一标识。
例如：在对象的访问地址 `examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/folder/picture.jpg` 中，对象键为 `folder/picture.jpg` 。

命名规范

- 对象键的名称可以使用任何 UTF-8 字符，为了确保名称与其他应用程序的最大兼容性，推荐使用英文大小写字母、数字，即[a-z，A-Z，0-9]及其组合。
- 编码长度最大为850个字节。
- 不允许以正斜线 `/` 或者反斜线 `\` 开头。
- 不允许携带以下字符串： `%0a` 、 `%0d` 。
- 对象键中不支持 ASCII 控制字符中的字符上(↑)，字符下(↓)，字符右(→)，字符左(←)，分别对应 CAN(24)，EM(25)，SUB(26)，ESC(27)。
- 对于特殊字符，例如 `*` 、 `%` 等字符，尽量避免直接作为文件名使用。

> **说明**
> 如果用户上传的文件或文件夹的名字带有中文，在访问和请求这个文件或文件夹时，中文部分将按照 URL Encode 规则转化为百分号编码。
> 例如：对 `文档.doc` 进行访问的时候，对象键为： `文档.doc` ，实际读取的按 URL Encode 规则转化的百分号编码为： `%e6%96%87%e6%a1%a3.doc` 。

以下是有效的对象键命名示例：

- doc/exampleobject
- my.great_photos-2016/01/me.jpg

- videos/2016/birthday/video.wmv

特殊字符

有些字符在对象键中可能需要以十六进制形式在 URL 中编码或引用，其中有些甚至是不可被打印的，因此浏览器可能无法处理它们，对于这些字符需要进行特殊处理。可能需要特殊处理的字符如下：

,	:	;	=
&	$	@	+
?	ASCII 字符范围：00-1F 十六进制（0-31 十进制）以及7F（127 十进制）	（空格）	

还有一些字符因为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，所以建议直接避免使用。需要避免的字符如下：

`	^	"	\
{	}	[	]
~	%	#	|
*	>	<	ASCII 128-255 十进制

对象访问地址

对象的访问地址由存储桶访问地址和对象键组成，其结构形式为 `<存储桶域名>/<对象键>` 。
例如：上传对象 `exampleobject.txt` 到广州（华南）的存储桶 `examplebucket-1250000000` 中，那么 `exampleobject.txt` 的访问地址是：
`examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/exampleobject.txt` 。

文件夹和目录

对象存储中本身是没有文件夹和目录的概念的，对象存储不会因为上传对象 `project/a.txt` 而创建一个 `project` 文件夹。为了满足用户使用习惯，对象存储在控制台、COS browser 等图形化工具中模拟了「文件夹」或「目录」的展示方式，具体实现是通过创建一个键值为 `project/` ，内容为空的对象，展示方式上模拟了传统文件夹。
例如：通过 API、SDK 上传对象 `project/doc/a.txt` ，分隔符 `/` 会模拟「文件夹」的展示方式，于是可以看到控制台上出现「文件夹」 `project` 和 `doc` ，其中 `doc` 是 `project` 下一级「文件夹」，并包含了 `a.txt` 。

> **注意**
> 存储桶中不同对象是扁平的分布到不同的分布式集群中的，因此无法直接获取某对象键前缀容量的大小，只能通过累加各对象的大小得到。

对于文件夹和目录进行删除操作，情况会比较复杂，详情如下：

删除途径	删除对象或文件夹	结果
控制台	文件夹 `project`	对象键前缀为 `project/` 的全部对象都会被删除

控制台	对象 `project/doc/a.txt`	`project` 和 `doc` 文件夹仍会保留
API、SDK	对象 `project/` 或 `project/doc/`	对象 `project/doc/a.txt` 仍会保留。如果想将文件夹内的对象一并删除，则需要用代码遍历实现删除文件夹内对象的功能

对象元数据

对象元数据在对象中是一组键值对，是服务器以 HTTP 协议传 HTML 资料到浏览器前所送出的字符串，又称为 HTTP Header。通过在上传对象时修改 HTTP Header，可以改变页面的响应形式，或者传达配置信息，例如修改缓存时间。对象元数据包括有两种元数据：系统元数据和用户定义的元数据。

> **说明**
> 修改对象的 HTTP Header 不会修改对象本身。

系统元数据

指对象的属性信息，例如对象修改时间、对象大小、存储类型等。

名称	说明
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）
Last-Modified	对象创建日期或上次修改日期（以较晚者为准）
x-cos-version-id	对象版本 ID。在存储桶上启用版本控制时，返回对象的版本 ID，用于标识同一对象的历史版本
ETag	如通过 PUT Object 上传的对象，则为上传文件内容的 MD5 值，如通过分块上传或使用历史版本 API 上传的对象，为上传文件内容的唯一 ID，不具备校验功能

用户定义元数据

指的是对象的可自定义参数，如 Content-Type，Cache-Control，Expires，x-cos-meta-* 等。具体信息请参见 自定义对象 Headers 。

名称	说明
Cache-Control	RFC 2616 中定义的缓存策略，将作为对象元数据保存
Content-Disposition、Content-Encoding、Content-Type	RFC 2616 中定义的文件名称/编码格式/内容类型（MIME），将作为对象元数据保存
Expires	RFC 2616 中定义的缓存失效时间，将作为对象元数据保存

版权所有：腾讯云计算（北京）有限责任公司

x-cos-acl	定义对象的访问控制列表（ACL）属性。有效值：private，public-read-write，public-read；默认值：private
x-cos-grant-*	赋予被授权者某种权限
x-cos-meta-*	允许用户自定义的头部信息，将作为对象元数据返回（大小限制为2K）
x-cos-storage-class	设置对象的存储级别，枚举值：STANDARD，STANDARD_IA，ARCHIVE；默认值：STANDARD
x-cos-server-side-encryption	指定是否为对象启用服务端加密及加密的方式

对象操作

用户可以通过腾讯云控制台、工具、API、SDK 等多种方式管理对象。

> **注意**
> 按照支持的最大上传文件大小，划分为两种上传方式：简单上传 和 分块上传 。
> - 使用简单上传，对象大小限制在5GB以内。
> - 使用分块上传，每块的大小限制在5GB以内，分块数量需要小于10000，即最大上传对象约为48.82TB，更多限制说明，请参见 规格与限制 。

上传对象完成后，用户可以通过控制台对对象进行相关配置，具体请参见以下内容：

- 搜索对象
- 查看对象信息
- 设置对象的访问权限
- 设置自定义 Headers

对象子资源

COS 有与存储桶和对象相关联的子资源。子资源从属于对象，即子资源不会自行存在，它始终与某些其他实体（例如对象或存储桶）相关联。 访问控制列表（Access Control List）是指特定对象的访问控制信息列表，它是 COS 中对象的子资源。访问控制列表包含可以识别被授权者和其被授予的许可的授权列表，来实现对对象的访问控制。创建对象时，ACL 将识别可以完全控制对象的对象所有者。用户可以检索对象 ACL 或将其替换为更新的授权列表。

> **说明**
> 对 ACL 的任何更新都需要替换现有 ACL。

访问权限类型

对象存储 COS 支持对象设置两种权限类型：**公共权限**和**用户权限**。

公共权限：包括继承权限、私有读写和公有读私有写。

- 继承权限：对象继承存储桶的权限，与存储桶的访问权限一致。当访问对象时，COS 读取到对象权限为继承存储桶权限，会匹配存储桶的权限，来响应访问。任何新对象被添加时，默认继承存储桶权限。

- 私有读写：当访问对象时，COS 读取到对象的权限为私有读写，此时无论存储桶为何种权限，对象都需要通过 请求签名 才可访问。
- 公有读私有写：当访问对象时，COS 读取到对象的权限为公有读，此时无论存储桶为何种权限，对象都可以被直接下载。

用户权限：主账号默认拥有对象所有权限（即完全控制）。另外 COS 支持添加子账号有数据读取、数据写入、权限读取、权限写入，甚至完全控制的最高权限。

适用场景

在私有读写的存储桶中对特定对象设置允许公有访问，或在公有读写存储桶中对特定对象设置需要鉴权才可访问。

存储类型
存储类型概述

最近更新时间：2025-02-18 18:17:33

存储类型可体现对象在对象存储（Cloud Object Storage，COS）中的存储级别和活跃程度。按照访问频度的高低和容灾程度划分，COS 提供多种对象的存储类型：标准存储（多 AZ）、低频存储（多 AZ）、智能分层存储（多 AZ）、智能分层存储、标准存储、低频存储、归档存储、深度归档存储。每种存储类型拥有不同的特性，例如对象访问频度、数据持久性、数据可用性和访问时延等。用户可根据自身场景选择以哪种存储类型将数据上传至 COS。

> **注意：**
> - 上传对象时，如不设定存储类型，则默认存储类型为**标准存储**。
> - 有关多 AZ 的相关介绍，请参见 多 AZ 特性概述 。

标准存储（多 AZ）/标准存储

标准存储（多 AZ）（MAZ_STANDARD）和标准存储（STANDARD）均属于热数据类型，两者都拥有低访问时延、高吞吐量的性能，可为用户提供高可靠性、高可用性、高性能的对象存储服务。
与标准存储（STANDARD）相比，标准存储（多 AZ）拥有更高的数据持久性和服务可用性，标准存储（多 AZ）采用不同的存储机制，将数据存储于同一城市的不同机房，不受同一地域单机房故障影响，可进一步保障用户业务稳定性。
适用场景
标准存储（多 AZ）、标准存储均适用于实时访问大量热点文件、频繁的数据交互等业务场景，例如热点视频、社交图片、移动应用、游戏程序、静态网站等。
标准存储（STANDARD）涵盖大多数使用场景，存储成本比标准存储（多 AZ）低，属于一种通用型存储类型。
而标准存储（多 AZ）拥有更高的数据持久性和服务可用性，适用于更高要求的业务场景，例如重要文件、商业数据、敏感信息等。

低频存储（多 AZ）/低频存储

低频存储（多 AZ）（MAZ_STANDARD_IA）和低频存储（STANDARD_IA）均可为用户提供高可靠性、较低存储成本和较低访问时延的对象存储服务。两者在降低存储价格的基础上，保持首字节访问时间在毫秒级，保证用户在取回数据的场景下无需等待，高速读取。与标准存储有明显区别的是，用户访问数据时会收取数据取回费用。
低频存储（多 AZ）与低频存储相比，低频存储（多 AZ）采用不同的存储机制，将数据存储于同一城市的不同机房，可进一步保障用户业务稳定性不受单机房故障影响。
适用场景
低频存储（多 AZ）和低频存储均适用于较低访问频率（例如平均每月访问频率1到2次）的业务场景，例如网盘数据、大数据分析、政企业务数据、低频档案、监控数据。

> **注意：**
> 低频存储（多 AZ）和低频存储类型都对存储时间和存储单元有最低限制，存储时间不足30天，按30天计算。单个存储文件不足64KB，按64KB计算，大于或等于64KB，按实际大小计算。有关定价信息，请参见 COS 定价 。

智能分层存储（多 AZ）/智能分层存储

智能分层存储（多 AZ）（MAZ_INTELLIGENT_TIERING）类型的对象可存放在标准存储（多 AZ）层和低频存储（多 AZ）层两个存储层，智能分层存储（INTELLIGENT_TIERING）类型的对象可存放在标准存储层和低频存储层两个存储层，COS 可根据智能分层存储（多 AZ）/智能分层存储类型对象的访问频次自动在对应的两个存储层之间变换，无数据取回费用，可降低用户的存储成本。关于智能分层存储的更多说明，请参见 智能分层存储简介 。
智能分层存储（多 AZ）与智能分层存储相比，智能分层存储（多 AZ）采用不同的存储机制，将数据存储于同一城市的不同机房，可进一步保障用户业务稳定性不受单机房故障影响。

适用场景

智能分层存储（多 AZ）和智能分层存储均适用于数据访问模式不固定的场景，如果您的业务对成本要求较为严格，且对文件读取性能较不敏感，您可以使用该存储类型来降低使用成本。

> **注意：**
> 对于智能分层存储（多 AZ）和智能分层存储类型的文件。单个存储文件不论大小，均按实际数据大小计费。有关定价信息，请参见 COS 定价 。

归档存储

归档存储（ARCHIVE）属于冷数据类型，数据取回时需要提前恢复（解冻），可为用户提供高可靠性、极低存储成本和长期保存的对象存储服务。归档存储有最低90天的存储时间要求，并且在读取数据前需要先进行数据恢复（解冻）。
恢复操作支持三种模式：

- 快速取回模式：恢复任务在1 – 5分钟内可完成。
- 标准取回模式：恢复任务在3 – 5小时内完成。
- 批量取回模式：恢复任务在5 – 12小时内完成。

> **说明：**
> - 关于恢复操作指引，请参见 恢复归档对象 。
> - 数据恢复请求存在 QPS 限制，限制为100次/秒。

适用场景

归档存储适用于需要长期保存数据的业务场景，例如档案数据、医疗影像、科学资料等合规性文件归档、生命周期文件归档、操作日志归档以及异地容灾。

> **注意：**
> 归档存储对存储时间和存储单元有最低限制，存储时间不足90天，按90天计算。单个存储文件不足64KB，按64KB计算，大于或等于64KB，按实际大小计算。有关定价信息，请参见 COS 定价 。

深度归档存储

深度归档存储（DEEP_ARCHIVE）可为用户提供高可靠性、比其他存储类型都低的存储成本和长期保存的对象存储服务。深度归档存储有最低180天的存储时间要求，并且在读取数据前需要先进行数据恢复。关于深度归档存储的更多说明，请参见 深度归档存储简介 。
恢复操作支持两种模式：

- 标准取回模式：恢复时间为12 – 24小时。
- 批量取回模式：恢复时间为24 – 48小时。

> **说明：**
> 数据恢复请求存在 QPS 限制，限制为100次/秒。

适用场景

深度归档存储适用于需要长期保存数据的业务场景，例如医疗影像数据、视图数据、日志数据。

> **注意：**
> 深度归档存储对存储时间和存储单元有最低限制，存储时间不足180天，按180天计算。单个存储文件不足64KB，按64KB计算，大于或等于64KB，按实际大小计算。有关定价信息，请参见 COS 定价 。

存储类型对比

对比项	标准存储（多AZ）	低频存储（多AZ）	智能分层存储（多AZ）	智能分层存储	标准存储	低频存储	归档存储	深度归档存储
存储类型参数	MAZ_STANDARD	MAZ_STANDARD_IA	MAZ_INTELLIGENT_TIERING	INTELLIGENT_TIERING	STANDARD	STANDARD_IA	ARCHIVE	DEEP_ARCHIVE
数据持久性	99.99999999%	99.99999999%	99.9999999999%	99.9999999999%	99.9999999999%	99.9999999999%	99.999999999%	99.99999999999%
服务可用性	99.995%	99.95%	随层级的转换而改变。当处于高频访问层时，服务可用性可参考标准存储（多AZ）；当处于低频访问层时，服务可用性可参	随层级的转换而改变。当处于高频访问层时，服务可用性可参考标准存储；当处于低频访问层时，服务可用性可参考低	99.99%	99.90%	99.00%	99.00%

			考低频存储（多AZ）。	频存储。				
响应	毫秒级	毫秒级	毫秒级	毫秒级	毫秒级	毫秒级	需提前申请恢复，恢复操作支持三种模式： • 快速取回模式：恢复任务在1 – 5分钟内可完成。 • 标准取回模式：恢复任务在3 – 5小时内完成。 • 批量取回模式：恢复任务在5 – 12小时内完成。	需提前申请恢复，恢复操作支持两种模式： • 标准取回模式，恢复时间为12 – 24小时。 • 批量取回模式，恢复时间为24 – 48小时。
对象最小计量大小	按对象实际大小计算	64KB	按对象实际大小计算	按对象实际大小计算	按对象实际大小计算	64KB	64KB	64KB
最短计费时间	不限制	30天	不限制	不限制	不限制	30天	90天	180天
支持地域	当前仅支持北京、上海、广州、	当前仅支持北京、上海、广州、	当前仅支持北京、上海、广州、新加坡地域	当前仅支持北京、南京、上海、广州、成都、重庆、新	全部地域	全部地域	适用于公有云地域（雅加达除外）、深圳金融地域	当前仅支持北京、南京、上海、广州、成都、重庆、东京、新加坡地域

	中国香港、新加坡地域	中国香港、新加坡地域		加坡、东京、法兰克福、弗吉尼亚地域				
存储费用	较高	较高	与转换后的存储类型一致	与转换后的存储类型一致	标准	低	极低	极低
数据取回费用	无	较低，按实际读取的数据量收取	无	无	无	较低，按实际读取的数据量收取	较高，按实际恢复的数据量收取，不同恢复模式费用不同	高，按实际恢复的数据量收取，不同恢复模式费用不同
请求费用	标准	较高	较高，此外还收取智能分层对象监控费用	较高，此外还收取智能分层对象监控费用	标准	较高	标准（数据需恢复到标准存储）	高（数据需要恢复到标准存储），此外对深度归档存储类型的数据进行取回时，还会额外收取数据取回请求费用
数据处理	支持	支持	支持	支持	支持	支持	支持，但需先恢复	支持，但需先恢复

存储类型的转换

COS 提供了多种对象的存储类型，可体现对象在 COS 中的存储级别和活跃程度。在实际使用当中，用户仍然可以按实际需求修改对象的存储类型，或者将对象沉降到较低活跃度的存储类型（例如低频存储、归档存储或深度归档存储）。

> **说明：**
> - 转换存储类型时，请先确认当前对象的所在存储桶已支持对应的存储类型。
> - 若对象属于**归档存储/深度归档存储**类型，则需要恢复到标准存储类型才可修改其他存储类型，详情请参见 恢复归档对象 。

关于存储类型的转换情况如下表所示：

存储类型	可修改的存储类型	可沉降的存储类型和沉降优先级
标准存储（多	低频存储（多 AZ）、智能分层存储（多	标准存储（多 AZ）> 低频存储（多 AZ）>

AZ）	AZ）	智能分层存储（多 AZ）
低频存储（多 AZ）	标准存储（多 AZ）、智能分层存储（多 AZ）	低频存储（多 AZ）> 智能分层存储（多 AZ）
智能分层存储（多 AZ）	标准存储（多 AZ）、低频存储（多 AZ）	无
智能分层存储	标准存储、低频存储、归档存储、深度归档存储	智能分层存储 > 归档存储 > 深度归档存储
标准存储	智能分层存储、低频存储、归档存储、深度归档存储	标准存储 > 低频存储 > 智能分层存储 > 归档存储 > 深度归档存储
低频存储	智能分层存储、标准存储、归档存储、深度归档存储	低频存储 > 智能分层存储 > 归档存储 > 深度归档存储
归档存储	恢复后，可修改为智能分层存储、标准存储、低频存储、深度归档存储	归档存储 > 深度归档存储
深度归档存储	恢复后，可修改为智能分层存储、标准存储、低频存储、归档存储	无

具体操作指引请参见以下文档：

- 修改存储类型
- 设置生命周期

智能分层存储简介

最近更新时间：2025-02-18 18:17:33

简介

智能分层存储类型为数据提供了冷热分层机制，能够根据用户数据的访问模式，自动地转换数据的冷热层级，从而降低用户数据的存储成本。
智能分层存储适用于访问模式不固定或者无法预估访问模式的数据，COS 会监控对象的访问情况，存储费用将按照数据实际存储的层级收费（高频访问层、低频访问层、归档层、深度归档层）。用户可以按照自身业务需求，将访问模式不固定的数据转换为智能分层存储类型，降低云上存储成本。

> **注意：**
> - 智能分层存储类型当前仅支持北京、南京、上海、广州、成都、重庆、新加坡、东京、法兰克福、弗吉尼亚地域。
> - 智能分层存储类型为独立的存储类型，使用时将产生智能分层存储容量费用和智能分层对象监控费用，其中智能分层存储容量费用，您可以选择智能分层存储容量包进行抵扣。更多计费信息请参见 产品定价 。
> - MAZ 存储桶不支持开启智能分层归档/深度归档层配置。

优势

当用户上传数据时选择以智能分层存储类型存放到 COS，COS 将周期性地监测数据访问次数，在持续一段时间没有数据访问时，将数据转移至存储成本更低的访问层。如果数据重新被访问，则会被重新转移到高频访问层上，保障数据读取性能。通过数据冷热分层存储，智能分层能够帮助用户在存储成本和读写性能之间寻找平衡点。使用智能分层存储具有以下优势：

- **成本集约**：当数据持久化存储为智能分层存储类型时，存储时间越长，则相较存储于标准存储的成本越低，最多可节约20%左右的存储成本。智能分层存储类型还参与对象存储生命周期流程，用户可以按需将智能分层存储沉降到归档存储中，进一步降低数据存储成本。
- **稳定持久**：智能分层存储提供与标准存储一致的低时延和高吞吐体验。同时，智能分层存储采用纠删码冗余存储的方式，提供了高达99.999999999%（11个9）的数据可靠性；数据分块存储，并发读写，提供高达99.99%的业务可用性。多 AZ 架构已同步智能分层特性，数据设计可靠性可高达99.9999999999%（12个9），业务设计可用性可高达99.995%。
- **便捷易用**：只需为数据指定对象存储类型，即可应用智能分层存储特性。智能分层存储作为一种存储类型，天然适配 COS 的 API、SDK、工具以及生态应用，方便用户按需管理存储在云上的数据。

支持的存储层级

> **说明：**
> 64KB以下小文件将保留在标准层，不沉降到低频访问层、归档层、深度归档层。

用户可以通过 API HeadObject 返回的头部 x-cos-storage-tier 查看智能分层对象处在哪个层级。此外，调用 API GetBucket 和 GetBucketVersions 返回的对象列表中，智能分层对象将返回字段 StorageTier 用于查看对象所在的层级。

访问层级	x-cos-storage-tier
高频访问层	FREQUENT

低频访问层	INFREQUENT
归档层	ARCHIVE_ACCESS
深度归档层	DEEP_ARCHIEVE_ACCESS

高频访问层（默认开启）

上传 INT 对象后，对象默认处于高频访问层（FREQUENT）状态。当对象处于本层时，存储费用按当前地域标准存储刊例价进行计费。

低频访问层（默认开启）

为存储桶开启智能分层配置时，需要选取转换为低频访问层的天数，可选项包括30、60、90天。天数一旦设置后，不支持修改。
当对象连续30天（或60天、90天）无访问时，对象将从高频访问层转换为低频访问层（INFREQUENT）。当对象处于本层时，存储费用按当地域低频存储刊例价进行计费。该层对象被访问后，将重新回到高频访问层。

归档层（可选）

由于归档层对象需要先恢复再访问。因此，归档层的开启为可选项，用户可通过添加一条或多条智能分层归档配置规则，针对指定前缀、标签的对象开启归档层，并配置转换天数。归档层的转换天数至少为91天，最大值为730天。

> **说明：**
> - 若当前地域不支持归档存储，控制台不支持添加归档配置规则，智能分层对象不会沉降到归档层。
> - 当智能分层对象处于归档层，存储费用按当前地域归档存储刊例价进行计费，且不会产生提前删除费用。

恢复归档层的对象
当对象连续 N 天无访问时，对象将从低频访问层转换到归档层。一旦进入归档层，需要通过 PostObjectRestore 恢复后才能读取数据。

- 与普通的归档类型不同，INT 归档层对象的恢复，不会产生标准类型的副本，而是对象本身直接回到高频访问层。
- 与普通的归档类型相同，INT 归档层对象的支持快速、标准、批量三种取回模式。

在计费上：

- **INT 归档取回费用：**
  - **标准取回，批量取回不计费。**
  - **快速取回计费，刊例价与同地域归档存储快速取回费用相同。**
- **INT 归档取回请求费用：**
  - **标准取回请求、批量取回请求不计费。**
  - **快速取回请求计费，刊例价与同地域归档存储快速取回请求费用相同。由于归档存储快速取回请求暂不计费，所以这里 INT 归档快速取回请求也暂不计费，保持一致。**

深度归档层（可选）

同样的，深度归档层对象需要先恢复再访问。因此，深度归档层的开启为可选项，用户可通过添加一条或多条智能分层深度归档配置规则，针对指定前缀、标签的对象开启深度归档层。可以在同一条规则中，同时配置归档层和深度归档层的转换。需要注意的是，深度归档层的转换天数至少为180天，最大值为730天，**且必须大于归档层的转换天数**。

> **说明：**
> - 若当前地域不支持深度归档存储，控制台不支持添加深度归档配置规则，智能分层对象不会沉降到深度归档层。
> - 当 INT 对象处于深度归档层，存储费用按当前地域深度归档存储刊例价进行计费，且不会产生提前删除费用。

恢复深度归档层的对象

当对象连续 M 天无访问时，对象将从低频访问层/归档层转换到深度归档层。一旦进入深度归档层，需要通过 PostObjectRestore 恢复后才能读取数据。

- 与普通的深度归档类型不同，INT 深度归档层对象的恢复，不会产生标准类型的副本，而是对象本身直接回到高频访问层。
- 与普通的深度归档类型相同，INT 深度归档层对象支持标准、批量两种取回模式，**不收取回费用、取回请求费用。**

开启智能分层归档层及深度归档层配置

规则说明

当存储桶开启智能分层配置后，用户上传的智能分层对象默认仅在高频访问层、低频访问层间切换，在智能分层配置中添加归档及深度归档层规则后，可以开启归档、深度归档层的切换。每个存储桶最多支持添加1000条归档及深度归档层的配置规则，规则包含的元素如下。规则详情可参考 API 文档 PUT Bucket IntelligentTiering 。

规则名称（Id）

用于唯一标识归档及深度归档规则。

状态（Status）

归档层、深度归档层配置支持开启（Enabled）或关闭（Disabled），即使配置了归档/深度归档层规则，在关闭状态下，也不会被真实转换到归档或深度归档层级。

应用范围（Filter）

规定规则的生效范围，支持前缀筛选和标签筛选，前缀数量不超过1个，标签数量不超过10个。

层级设置

支持在一条规则中设置归档层和深度归档层的转换时间，设定好相应的规则后，INT 对象才会开启沉降到归档、深度归档层的转换，否则只会在高频访问层、低频访问层间互相转换。

- 归档层（ARCHIVE_ACCESS）：支持设置91天 – 730天。
- 深度归档层（DEEP_ARCHIVE_ACCESS）：支持设置180天 – 730天。

> **说明：**
> 如果在同一条规则中同时配置归档层、深度归档层的转换规则，则深度归档的转换天数必须大于归档的转换天数。

存储层级转换顺序

用户可以单独开启归档层的转换、单独开启深度归档层转换、同时开启归档和深度归档层的转换。按照配置的低频访问层、归档层、深度归档层的转换天数，将依次逐层沉降。

配置示例	配置详情	生效结果
示例1	低频访问层30天，归档层100天，深度归档层190天	• 连续30天无访问，从标准访问层转换为低频访问层。 • 连续100天无访问，从低频访问层转换为归档层。 • 连续190天无访问，从归档层转为深度归档层。
示例2	低频层30天，深度归档层190天	• 连续30天无访问，从标准访问层转换为低频层。 • 连续190天无访问，从低频层转换为深度归档层。
示例3	低频层60天，归档层91天	• 连续60天无访问，从标准访问层转换为低频层。 • 连续91天无访问，从低频层转换为归档层。

恢复归档及深度归档层的 INT 对象

当对象连续多天无访问时，对象将从低频访问层转换到归档层/深度归档层。一旦进入归档层、深度归档层，需要通过 PostObjectRestore 恢复后才能读取数据。

- 与普通归档类型、深度归档类型对象的恢复不同，INT 归档层、深度归档层对象的恢复，**不会产生标准类型的副本，而是对象本身直接回到高频访问层。因此，针对 INT 归档/深度归档层对象发起取回请求时，不需要指定取回天数。**

- 与普通的归档类型相同，INT 归档层对象支持快速、标准、批量取回模式；与普通的深度归档类型相同，INT 深度归档层对象支持标准、批量两种取回模式。
- **在收费上，除归档层快速取回费用外（定价与同地域归档存储快速取回费用相同），不收取任何其他取回费用、取回请求费用、回热副本存储费用等。**

		归档存储类型	智能分层对象-归档层	深度归档存储类型	智能分层对象-深度归档层
存储费用		按当前地域归档存储类型存储费用收费		按当前地域深度归档存储类型存储费用收费	
取回是否生成副本		是，需要指定保留天数，按标准存储定价产生副本存储费用。	否，无副本存储费用。	是，需要指定保留天数，按标准存储定价产生副本存储费用。	否，无副本存储费用。
取回模式		快速模式 标准模式 批量模式	快速模式 标准模式 批量模式	标准模式 批量模式	标准模式 批量模式
取回费用（单位：元/GB）	快速	归档快速取回费用	智能分层归档层快速取回费用（定价同归档存储）	\	\
	标准	归档标准取回费用	免费	深度归档标准取回费用	免费
	批量	归档批量取回费用	免费	深度归档批量取回费用	免费
取回请求费用（单位：元/万次）		暂不计费	暂不计费	深度归档标准取回请求费用 深度归档批量取回请求费用	免费
读写请求费用		与恢复后的标准存储费用一致	智能分层读写请求费用	深度归档读写请求费用	智能分层读写请求费用

恢复过程中，可以通过 HeadObject 查看 INT 归档层/深度归档层对象的恢复状态。

- 处于恢复状态中，HEAD Object的响应头部将包含 x-cos-restore 和 x-cos-restore-status。例如，x-cos-restore：ongoing-request="true"，cos-restore-status：tier="bulk"; request-date="Mon, 18 Nov 2019 09:34:50 GMT"。
- 恢复完毕后，INT 对象直接回到高频访问层，HEAD Object 的相应头部 x-cos-storage-tier 为 FREQUENT。

使用方法

将数据以智能分层存储类型存放到 COS，首先需要为存储桶开启智能分层配置。开启后，用户在上传对象时将**存储类型**指定为智能分层存储类型即可。

使用对象存储控制台

上传对象时设置为智能分层存储

用户可以参照以下步骤将对象保存为智能分层存储类型：

1. 在存储桶配置页面，开启智能分层存储配置，详细流程请参见 设置智能分层存储 文档。
2. 上传文件，并在上传时指定文件存储类型。文件的上传指引请参见 上传对象 文档。

> **注意：**
> 存储桶的智能分层存储配置开启后，将无法关闭，请谨慎配置。

将云上数据转换为智能分层

用户可以参照以下步骤将已上传的存量数据转换为智能分层存储类型：
1. 在存储桶配置页面，创建生命周期规则，详细流程请参见 设置生命周期 文档。
2. 设置指定的规则应用范围，将数据沉降为智能分层存储。

开启智能分层归档及深度归档层

用户可按需为存储桶开启智能分层归档及深度归档层配置。详情请参见 设置智能分层存储 文档。

使用 REST API

您可以直接通过以下 API 配置智能分层存储：
1. 首先使用 REST API 为存储桶开启智能分层存储，请参见以下 API 文档：
   - 开启智能分层、配置低频层转换天数：
     - PUT Bucket IntelligentTiering（id=default）
     - GET Bucket IntelligentTiering（id=default）
   - 配置、删除归档及深度归档层规则：
     - PUT Bucket IntelligentTiering （id 非 default）
     - GET Bucket IntelligentTiering（id 非 default）
     - Delete Bucket IntelligentTiering（id 非 default）
2. 存储桶开启智能分层存储，您可以参见以下 API 文档，将对象上传为智能分层存储类型：
   - PUT Object
   - PUT Object - Copy
   - POST Object
   - Initiate Multipart Upload
3. 如需查询对象的存储类型和所处的存储层，请参见以下 API 文档：
   - GET Object
   - HEAD Object
4. 您可以直接使用 REST API 删除智能分层存储类型的对象，请参见以下 API 文档：
   - DELETE Object
   - DELETE Multiple Objects
   - PUT Bucket lifecycle
   - Delete Bucket IntelligentTiering
5. 取回智能分层归档层、深度归档层的对象，请参考 API： PostObjectRestore 。

使用 SDK

当前 COS 所发布的 SDK 都支持使用智能分层存储类型，具体方法是在文件上传时，将 StorageClass 参数设置为 INTELLIGENT_TIERING 和 MAZ_INTELLIGENT_TIERING，实现直传智能分层存储和智能分层存储（多 AZ）。关于上传对象 SDK 文档，请参见 SDK 概览 。

使用限制

使用智能分层存储存在以下限制：

- **配置限制**：首次配置后不可更改，如需更改，请 联系我们 。转换为低频访问层的天数，可选值为30、60和90。
- **初始存储层限制**：智能分层存储类型的新增对象，默认处于高频访问层；持续一段时间无访问后才会转换至低频访问层。
- **最小存储单元限制**：小于64KB的对象只能持久存储于高频访问层中，不会在高频访问层和低频访问层之间转换。单个存储文件不论大小，均按实际数据大小计费。
- **操作限制**：不支持通过追加上传接口将对象上传为智能分层存储类型。
- **生命周期限制**：智能分层存储类型仅可转换为归档存储或者深度归档存储类型。标准存储类型沉降为智能分层存储类型时，将存储在高频访问层；低频存储类型沉降为智能分层存储类型时，将存储在低频存储访问层。
- **存储桶复制限制**：存储桶复制时，如果目标存储桶未开启智能分层存储配置，则无法将对象复制为智能分层存储类型。

常见问题

智能分层存储是如何计费的？

智能分层存储包含**智能分层存储容量费用**和**智能分层对象监控费用**。其中：

- 智能分层存储容量费用会根据文件所处的存储层收取不同的存储费用。
  - 当文件处于高频层时，按照标准存储容量费用收费。
  - 当文件处于低频层时，按照低频存储容量费用收费。

> **说明：**
> - 标准存储、低频存储容量费用在不同公有云地域有不同的定价，具体定价请参见 产品定价 。
> - 上传和下载文件过程中还会产生请求费用和流量费用，这些费用计算示例请参见 流量费用计费案例 和 请求费用计费案例 。

- 智能分层对象监控费用按存储的文件数来计算，小于64KB的文件不收取，具体定价请参见 产品定价 。

示例

假设企业拥有1TB文件，每个文件均大于64KB，总共10万个文件，数据以智能分层存储类型存储在北京地域，并指定转换为低频访问层的时间为30天。假设每30天都会有20%的文件（即2万个文件）沉降到低频层，那么**每30天**的对象监控费用和存储费用，如下表所示：

> **说明：**
> 下面表格中北京地域的对象监控费用月单价为 0.175元/每万个对象，根据“日单价 = 月单价 / 30”的换算逻辑，则日单价为0.00583333元/万个对象/日。

存储天数	每30天对象监控费用（元）	每30天智能分层存储费用（元）	每30天标准存储费用（元）
30 x 1	0.175元/每万个对象 / 30 x 30 x	1024 x 0.118 / 30 x 30 = 120.83	1024 x 0.118 / 30 x

	10万 = 1.75		30 = 120.83
30 x 2	0.175元/每万个对象 / 30 x 30 x 10万 = 1.75	819.2 x 0.118 / 30 x 30 + 204.8 x 0.08 / 30 x 30 = 113.05	1024 x 0.118 / 30 x 30 = 120.83
30 x 3	0.175元/每万个对象 / 30 x 30 x 10万 = 1.75	655.36 x 0.118 / 30 x 30 + 368.64 x 0.08 / 30 x 30 = 106.82	1024 x 0.118 / 30 x 30 = 120.83
30 x 4	0.175元/每万个对象 / 30 x 30 x 10万 = 1.75	524.288 x 0.118 / 30 x 30 + 499.712 x 0.08 / 30 x 30 = 101.84	1024 x 0.118 / 30 x 30 = 120.83
30 x 5	0.175元/每万个对象 / 30 x 30 x 10万 = 1.75	419.4304 x 0.118 / 30 x 30 + 604.5696 x 0.08 / 30 x 30 = 97.86	1024 x 0.118 / 30 x 30 = 120.83
30 x 6	0.175元/每万个对象 / 30 x 30 x 10万 = 1.75	335.54432 x 0.118 / 30 x 30 + 688.45568 x 0.08 / 30 x 30 = 94.67	1024 x 0.118 / 30 x 30 = 120.83

可以看到，随着存储时间越长，每30天只需要付出少量的监控成本，即可带来明显的成本降低。

智能分层适用于什么类型文件?

智能分层存储适用于音视频、日志等平均文件较大，且访问模式不固定的文件。平均文件容量越大，意味着您平均每GB文件需要付出的监控费用越少；如果您的业务访问模式比较固定，则意味着您可以通过生命周期配置好指定时间沉降到低频存储上，而不需要使用智能分层存储。

如何将文件存储为智能分层存储?

您可以通过以下两种方式将文件存储为智能分层存储：

- 增量文件：您只需要在上传时指定好存储类型为智能分层存储，即可将文件存储为智能分层存储。
- 存量文件：您可以通过 COPY 接口将文件存储类型修改为智能分层存储类型；或者通过生命周期功能，将标准存储，低频存储类型沉降为智能分层存储类型。

> **注意：**
> 小于64KB 的智能分层类型的文件会一直存储在标准层中。因此小于64KB的文件推荐按需直接上传标准/低频/归档/深度归档等存储类型，这样能够降低成本。

如何关闭智能分层存储配置?

智能分层存储配置开启后**无法关闭**，如果您无需将文件存储为智能分层存储，您只需要在上传文件时将文件存储类型指定为标准存储、低频存储、归档存储或者深度归档存储等非智能分层存储类型即可。
智能分层归档层/深度归档层规则支持删除，您可以将对应的规则删除，避免新的 INT 对象沉降到归档层/深度归档层。

智能分层的归档和深度归档层如何收费?

- **存储费用**：按照同地域归档和深度归档存储收费。
- **取回费用、取回请求费用**：除归档层快速取回费用外，不收取其他取回费用、取回请求费用。

- **回热副本费用：**不收取。

深度归档存储简介

最近更新时间：2024-11-06 14:44:02

简介

深度归档存储（Deep Archive）是对象存储（Cloud Object Storage，COS）提供的可让海量数据长期归档的存储服务。深度归档存储提供了磁带存储级别的存储单价，为用户数据长期存储提供了低成本方案。用户无需在本地维护复杂的磁带库配置，无需关注底层存储介质的演进，通过对象存储 COS 提供的 API、SDK、生态工具和控制台等丰富的人机交互手段，即可实现便捷、低成本地管理数据。

深度归档存储适用于数据访问频率极低，但需要长期保留的场景。在日志冷备场景中，企业需要按照当地法律法规要求，将每天产生的日志数据进行冷备存储，以便追溯和分析。在视图数据和自动驾驶等业务中，企业沉淀了大量的图片、视频等媒体文件，在数据被使用过后仍然需要长期归档保存。通过深度归档存储，企业可以将这些数据存储在云上，仅在需要的时候恢复，降低存储成本和运维管理难度。

深度归档存储提供了99.999999999%（11个9）的数据持久性以及99.00%的业务设计可用性，用户还可以利用对象存储 COS 提供的版本控制、跨地域复制等功能进一步保障数据安全。

使用方法

1. 使用控制台上传
   1.1 登录 COS 控制台 。
   1.2 单击**存储桶列表**，创建所支持地域（例如北京地域）的存储桶。
   1.3 创建完成后，在**文件列表**界面中，单击**上传文件**。
   1.4 在弹出的窗口中，单击**选择文件**或**选择文件夹**，确认好待上传的本地文件/文件夹后，单击**参数配置**中将存储类型选择为**深度归档存储**，详情请参见 上传对象 。
2. 使用 API 上传
   通过在 PUT Object 、POST Object 或 Initiate Multipart Upload 接口中将 `x-cos-storage-class` 设置为 `DEEP_ARCHIVE` ，实现直传深度归档。
3. 使用 SDK 上传
   当前 COS 所发布的 SDK 都支持直传深度归档，具体方法是在文件上传时，将 `StorageClass` 参数设置为 `DEEP_ARCHIVE` ，实现直传深度归档。
4. 使用工具上传
   COSBrowser、COSCMD 工具支持直传深度归档，具体方法是在文件上传时，通过增加头部字段 `x-cos-storage-class` ，并设置为 `DEEP_ARCHIVE` ，实现直传深度归档。

使用限制

深度归档存储具有以下限制，用户在使用过程中，请留意相关限制对存储成本和性能带来的影响：

- **最小存储单元限制**：最小存储单元为64KB，小于64KB的对象按64KB计算存储容量。
- **最短存储时间限制**：最短存储时间为180天，低于180天按照180天收费。

> **说明**
> 存储天数为逻辑天（满24小时为1天），计时开始时间以文件的修改时间计算。

- **取回请求 QPS 限制**：100次/秒。
- **地域限制**：当前仅支持北京、南京、上海、广州、成都、重庆、东京、新加坡地域，其他地域将陆续支持。
- **使用限制**：深度归档存储暂不支持多 AZ，即开启了多 AZ 配置的存储桶，暂时无法使用深度归档存储类型。
- **操作限制**：不支持追加上传对象为深度归档存储类型。
- **取回请求限制**：每个对象每次只能执行一个取回请求，多个重复的取回请求会被合并并按照最快的恢复模式处理。如果第N+1次发起的取回请求的恢复模式快于第N次，则按照新的取回请求执行，否则第N+1次取回请求失败。
- **数据取回**：深度归档存储类型的文件支持标准取回模式和批量取回模式。在标准模式下恢复时间为12 – 24小时；在批量模式下恢复时间为24 – 48小时。

版权所有：腾讯云计算（北京）有限责任公司

上传对象
简单上传

最近更新时间：2022-12-01 17:32:42

简单上传指用户使用 PUT Object 接口上传对象。简单上传方式适用于在单个请求中，上传不超过 5GB 的单个对象。
如需上传大于 5GB 的单个对象，可使用以下方式：

- 使用控制台上传：控制台可支持上传最大512GB 的单个对象，详情请参见 上传对象 控制台指南文档。
- 使用 API/SDK 的分块上传：可支持上传最大48.82TB（即50000GB）的单个对象，请参见 分块上传 。
- 使用 COSCMD 工具支持上传最大 40TB 的单个对象，详情请参见 COSCMD 工具 。

> **说明**
> 在发起上传请求时，如需上传到指定的文件夹或路径，可通过 `/` 实现，例如将 picture.png 上传到 doc 文件夹，则对象键设置为 doc/picture.png。

适用场景

简单上传方式适用于上传不超过 5GB 对象的场景。
在高宽带或弱网络环境中，若您要上传的对象较大，例如超过100MB（即便小于5GB），我们建议您优先使用 分块上传 的方式。因为分块上传可以并行传输多个分块。在高宽带环境中，可以有效利用资源；在弱网络环境中，单一分块上传失败不会影响其他已上传的分块，通过简单的重试即可重传失败分块，从而提高整体的上传成功率。有关移动端在弱网络环境的上传，可参见 弱网分块续传实践 。

使用方法

使用 REST API

您可以直接使用 REST API 发起简单上传对象请求，详情请参见 PUT Object API 文档。

使用 SDK

您可以直接调用 SDK 的简单上传对象方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK

- 小程序 SDK

分块上传

最近更新时间：2024-10-14 11:40:01

简介

分块上传可以实现将整个对象切分为多个分块，然后再将这些分块上传到对象存储（Cloud Object Storage，COS）。上传时，这些分块将会按连续的序号编号，您可以独立上传或者按照任意顺序上传各个分块，最终 COS 将会根据分块编号顺序重新组合出该对象。任意分块传输失败，都可以重新传输当前分块，不会影响其他分块和内容完整性。一般在弱网络环境中，单个对象大于20MB 可优先考虑分块上传，在大带宽环境中可将超过100MB 的对象进行分块上传。
分块上传最多支持将较大的对象切分为10000个分块，切分的分块大小范围一般为1MB - 5GB，最后一个分块可以小于1MB。

> **说明**
> 简单上传的方式只支持最大5GB 的文件上传，而通过分块上传的方式可上传大于5GB 的文件。

适用场景

分块上传适合于在弱网络或高带宽环境下上传较大的对象。
分块上传优势如下：

- 在弱网络环境中，使用较小的分块可以将网络失败导致的中断影响降低，实现对象续传。
- 在高带宽环境中，并发上传对象分块能充分利用网络带宽，乱序上传并不影响最终组合对象。
- 使用分块上传，您可以随时暂停和恢复单个大对象的上传。除非发起终止操作，所有未完成的对象将可随时继续上传。
- 分块上传也适用于在未知对象总大小的情况下上传对象，您可以先发起上传，再组合对象以获得完整大小。

使用方法

使用 REST API

您可以直接使用 REST API 发起分块上传请求，详情请参见以下 API 文档：

- Initiate Multipart Upload
- Upload Part
- Complete Multipart Upload
- Abort Multipart Upload

使用 SDK

您可以直接调用 SDK 分块操作，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK

- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

预签名授权上传

最近更新时间：2024-06-25 14:09:31

适用场景

在默认情况下，存储桶和对象都是私有的。如果您希望第三方可以上传对象到存储桶，又不希望对方使用 CAM 账户或临时密钥等方式时，您可以使用预签名 URL 的方式将签名提交给第三方，以供完成临时的上传操作。收到有效预签名 URL 的任何人都可以上传对象。
预签名 URL 时，您可以在签名中设置将对象键包含在签名中，只许可上传为指定的对象键。您也可以在程序中指定预签名 URL 的有效时间，以保证超时后该 URL 不会被未授权方使用。
当指定预签名 URL 的有效时间过期后，请求会中断；申请新的签名后，需要重新执行失败请求，不支持断点续传。

> **注意：**
> - 由于访问 CDN 域名需要遵循 CDN 的鉴权过程，无法使用 COS 签名，因此预签名 URL 不支持使用 CDN 域名。
> - 使用永久密钥来生成预签名，注意权限范围仅限于上传或下载操作，并且所生成的签名有效时长设置为完成本次上传或下载操作所需的最短期限。
> - 仅支持对单个对象进行预签名，不支持多个对象。
> - 使用预签名URL上传时，最大支持上传 5GB 文件。

使用方法

使用 SDK

您可以直接调用 SDK 的预签名 URL 方法，请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Flutter SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- React Native SDK
- 小程序 SDK

下载对象
简单下载对象

最近更新时间：2024-10-14 11:40:01

适用场景

您可以直接发起请求下载对象存储（Cloud Object Storage，COS）中的对象，下载对象支持以下功能：

- 下载完整的单个对象：直接发起 GET 请求即可下载完整的对象数据。
- 下载单个对象的部分内容：可在 GET 请求中传入 Range 请求头部，支持检索一个特定的字节范围，不支持检索多个范围。

对象的元数据将会作为 HTTP 响应头部随对象内容一同返回，GET 请求支持使用 URL 参数的方式覆盖响应的部分元数据值。例如 Content-Disposition 的响应值。支持修改的响应头部包括：

- Content-Type
- Content-Language
- Expires
- Cache-Control
- Content-Disposition
- Content-Encoding

使用方法

使用对象存储控制台

您可以使用对象存储控制台下载对象，详情请参见 下载对象 控制台指南文档。

使用 REST API

您可以直接使用 REST API 发起下载对象请求，详情请参见 GET Object API 文档。

使用 SDK

您可以直接调用 SDK 下载对象，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK

- 小程序 SDK

预签名授权下载

最近更新时间：2023-08-11 09:20:11

适用场景

在默认情况下，存储桶和对象都是私有的。如果您希望第三方可以下载对象，又不希望对方使用 CAM 账户或临时密钥等方式时，您可以使用预签名 URL 的方式将签名提交给第三方，以供完成下载操作。收到有效预签名 URL 的任何人都可以下载对象。
预签名 URL 时，您可以在签名中设置将对象键包含在签名中，只允许下载指定的对象。您也可以在程序中指定预签名 URL 的有效时间，以保证超时后该 URL 不会被未授权方使用。

> **注意**
> - 由于访问 CDN 域名需要遵循 CDN 的鉴权过程，无法使用 COS 签名，因此预签名 URL 不支持使用 CDN 域名。
> - 建议用户使用临时密钥生成预签名，通过临时授权的方式进一步提高预签名上传、下载等请求的安全性。申请临时密钥时，请遵循 最小权限指引原则，防止泄露目标存储桶或对象之外的资源。
> - 如果您一定要使用永久密钥来生成预签名，建议永久密钥的权限范围仅限于上传或下载操作，以规避风险。并且所生成的签名有效时长设置为完成本次上传或下载操作所需的最短期限，因为，当指定预签名 URL 的有效时间过期后，请求会中断；申请新的签名后，需要重新执行失败请求，不支持断点续传。
> - 仅支持对单个对象进行预签名，不支持多个对象。

使用方法

使用 SDK

您可以直接调用 SDK 的预签名 URL 方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Flutter SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- React Native SDK
- 小程序 SDK

列出对象键

最近更新时间：2024-10-14 11:40:01

对象键（ObjectKey） 是对象在存储桶中的唯一标识，可以通俗的理解为文件路径。例如对象键为 doc/picture.jpg 则表示为图片文件 picture.jpg 存放在对象存储（Cloud Object Storage，COS）的 doc 路径（或文件夹）下。
列出对象键即按照指定的对象键查找特定对象。此外，也可以按对象键前缀查找对象，即指定对象键前面的一部分（例如 doc）查找具有相同前缀 doc 的所有对象。

适用场景

腾讯云 COS 支持按照前缀顺序列出对象键，您也可以在对象键中使用 `/` 字符来实现类似传统文件系统的层级结构，COS 也支持按照分隔符来做层级结构的选择和浏览。
您可以列出单个存储桶中的所有对象键，根据前缀的 UTF-8 二进制顺序列出，或选择指定前缀过滤对象键的列表。例如加入参数 `t` 将列出 `tencent` 的对象，而跳过以 `a` 或其他字符为前缀的对象。
加入 `/` 分隔符可将根据此分隔符重新组织对象键，您可以结合前缀和分隔符来实现类似文件夹检索的功能。
腾讯云 COS 在单个存储桶中支持无限数量的对象，因此对象键列表可能非常大。为了管理方便，单个列出对象接口将最多返回1000个对象列表，同时会返回指示器来告知是否存在截断，若存在截断，则表示下一页还存在对象列表。您可以根据指示器和分隔符多次发起列出对象键请求，实现列出所有对象键，或查询您所需要的内容。

使用方法

使用对象存储控制台

您可以使用控制台搜索对象，详情请参见 搜索对象 控制台指南文档。

使用 REST API

您可以直接使用 REST API 发起列出对象键请求，详情请参见 GET Bucket（List Objects） API 文档。

使用 SDK

您可以直接调用 SDK 查询对象列表，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

复制对象
简单复制

最近更新时间：2024-11-29 11:00:53

适用场景

您可以在对象存储（Cloud Object Storage，COS）中将已存储的对象通过简单的复制操作，创建一个新的对象副本。在单个操作中，您可以复制最大5GB 的对象。当对象超过5GB 时，您必须使用 分块复制 接口来实现复制。复制对象有以下功能：

- 创建一个新的对象副本。
- 复制对象并更名，删除原始对象，实现重命名。
- 修改对象的存储类型，在复制时选择相同的源和目标对象键，修改存储类型。
- 在不同的腾讯云 COS 地域复制对象。
- 修改对象的元数据，在复制时选择相同的源和目标对象键，并修改其中的元数据。

复制对象时，默认将继承原对象的元数据，但创建日期将会按新对象的时间计算。

> **说明：**
> - 不支持对归档类型（例如归档存储、深度归档存储）的对象发起 PUT Object - Copy 请求。
> - 开启了多 AZ 配置的存储桶，不支持将多 AZ 存储类型复制为单 AZ 存储类型。
> - 子账号复制对象，需要拥有这三个权限：PutObject、GetObject、GetObjectACL。

使用方法

使用对象存储控制台

您可以使用对象存储控制台复制对象，详情请参见 复制对象 控制台指南文档。

使用 REST API

您可以直接使用 REST API 发起复制对象请求，详情请参见 PUT Object - Copy API 文档。

使用 SDK

您可以直接调用 SDK 的设置对象复制方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK

- PHP SDK
- Python SDK
- 小程序 SDK

分块复制

最近更新时间：2024-11-29 11:00:53

适用场景

当需要复制一个超过5GB 的对象时，您需要选择分块复制的方法来实现。使用分块上传的 API 来创建一个新的对象，并使用 Upload Part - Copy 的功能，携带 x-cos-copy-source 头部来指定源对象，流程包括：

1. 初始化一个分块上传的对象。
2. 复制源对象的数据，可指定 x-cos-copy-source-range 头部，每次只可复制最多5GB 数据。
3. 完成分块上传。

> **说明**
> 使用腾讯云 COS 提供的 SDK 可以轻松完成分块复制的功能。

使用方法

使用 REST API

您可以直接使用 REST API 发起分块复制请求，请参见以下 API 文档：

- Initiate Multipart Upload
- Upload Part - Copy
- Complete Multipart Upload
- Abort Multipart Upload

使用 SDK

您可以直接调用 SDK 的复制分块方法，请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET(C#) SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

删除对象
删除单个对象

最近更新时间：2023-01-31 15:27:40

适用场景

腾讯云对象存储（Cloud Object Storage，COS）支持直接删除一个或多个对象，当仅需要删除一个对象时，您只需要提供对象键，就可以调用一个 API 请求来删除它。

使用方法

使用对象存储控制台

您可以使用对象存储控制台删除对象，详情请参见 删除对象 控制台指南文档。

使用 REST API

您可以直接使用 REST API 发起删除单个对象请求，详情请参见 DELETE Object API 文档。

使用 SDK

您可以直接调用 SDK 的删除单个对象方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

删除多个对象

最近更新时间：2024-10-14 11:41:11

适用场景

腾讯云对象存储（Cloud Object Storage，COS）支持批量删除多个对象。您可以通过控制台、 API、SDK 等多种方式批量删除对象。
默认情况下，当删除任务都成功完成时，返回的内容通常为空。若有发生错误，则会返回错误信息。

> **注意**
> 单次请求最多可删除1000个对象，若需要删除更多对象，请将列表拆分后分别发送请求。

使用方法

使用对象存储控制台

您可以使用对象存储控制台批量删除多个对象，详情请参见 删除对象 控制台指南文档。

使用 REST API

您可以直接使用 REST API 发起删除多个对象请求，详情请参见 DELETE Multiple Objects API 文档。

使用 SDK

您可以直接调用 SDK 删除多个对象，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

版权所有：腾讯云计算（北京）有限责任公司

数据管理
生命周期管理
生命周期概述

最近更新时间：2025-04-09 19:46:02

概述

对象存储（Cloud Object Storage，COS）支持基于对象的生命周期配置，其通过对存储桶下发指定的描述语言，可以让符合规则的对象在指定的条件下自动执行一些操作。

> **说明：**
> 每个存储桶最多可添加1000条生命周期规则。

适用场景

日志记录

如果用户使用对象存储来存储日志数据，可以通过生命周期配置，使得日志数据在30天后自动归档，或者2年后自动删除。

冷热分层

热数据往往在上传后，短时间内被大量访问而热度升高，一段时间后热度逐渐降低或者不再需要被实时访问。您可以通过生命周期规则将30天前的数据转换为低频存储，进一步可以将60天前的数据转换为归档存储，这个过程称之为数据沉降。

> **说明：**
> 目前对象存储支持智能分层，能够根据用户数据的访问模式，自动地转换数据的冷热层级，从而降低用户数据的存储成本。详情请参见 智能分层存储简介 。

存档管理

使用对象存储进行文件存档管理时，往往根据金融、医疗等合规性要求，需要长期保存文件的所有历史版本，此时可以使用生命周期功能，对历史版本的文件执行沉降至归档的操作。

配置元素

创建一个生命周期规则，需要配置以下元素。

资源范围

指定生命周期规则执行的资源范围，支持自定义生命周期的应用范围和范围内覆盖的数据类型。生命周期执行时将扫描用户指定的应用范围，并对范围内用户配置的数据类型执行操作。其中，应用范围可指定存储桶的所有对象，也可以通过以下规则指定部分对象：

- 按对象前缀指定：支持按照目录前缀或文件名称前缀进行匹配。
- 按对象标签指定：通过对象标签筛选数据。

支持配置的数据类型包括：

- 当前版本文件：存储桶中最新版本的对象。
- 历史版本文件：开启版本控制后存储的历史版本对象。关于版本控制的更多信息请参见 版本控制概述 。
- 删除标记：“对象已被删除的标记”，生命周期支持在历史版本都删除后，自动移除该标记。关于删除标记的更多描述请参见 删除标记文档 。
- 碎片文件：分块上传任务未完成时产生的碎片。

操作

指定命中对象时，执行的操作：沉降数据和过期删除。

沉降数据

将对象在最后一次修改时间起，经过多少天后沉降为低频存储、智能分层存储、归档存储和深度归档存储类型。

- 支持地域
  支持公有云地域，金融云地域仅支持将数据沉降至低频存储类型。
- 单向原则
  沉降数据是单向的，只允许从标准存储 > 低频存储 > 智能分层存储 > 归档存储 > 深度归档存储，也支持跳级沉降（例如标准存储 > 归档存储），不支持逆向。您只能通过调用 PUT Object - Copy（针对非归档存储/深度归档存储类型），或 POST Object restore （仅适用于归档存储/深度归档存储类型）来将较冷存储类型的数据恢复至较热存储类型。
- 最终一致性
  如果对同一组的对象配置了多条规则，且存在冲突性情况（不含过期删除配置），对象存储会优先执行**沉降至最冷存储类型**的规则。
  例如，规则 A 配置了文件修改90天后**沉降至低频存储**，规则 B 配置了文件修改90天后**沉降至归档存储**，且上述规则都命中了同一对象 test.txt，则优先执行规则 B。

规则	资源	操作	时间条件	执行情况
规则 A	test.txt	沉降至低频存储	文件修改时间的90天	规则冲突，执行失败
规则 B	test.txt	沉降至归档存储	文件修改时间的90天	执行成功

注意：

- 生命周期不会对小于64KB的对象执行转换操作。
- 腾讯云 COS 强烈提醒您不要针对同一组对象配置多个含冲突条件的生命周期规则，冲突执行可能导致不同的费用表现。
- 沉降数据不会改变对象原先的上传或修改时间。

过期删除

设置对象的过期时间，使对象到期后被自动删除。

- 处理逻辑
  当对象匹配了指定的生命周期过期删除的规则时，腾讯云会将对象加入异步的删除队列，实际发生的删除时间将会与创建时间有一定的延时。您将可以通过 GET 或 HEAD Object 操作来获取对象的当前状态。

- 最终一致性
  如果对同一组的对象配置了多条规则，且存在冲突性情况，对象存储会以最短过期时间为准执行，且**过期删除的执行效力大于转换存储类型**。
  例如，规则 C 配置了文件修改180天后**沉降至低频存储**，规则 D 配置了文件修改180天后**删除对象**，且上述规则都命中了同一对象 test.txt，则优先执行规则 D。

规则	资源	操作	时间条件	执行情况
规则 C	test.txt	沉降至低频存储	文件修改时间的180天	规则冲突，执行失败
规则 D	test.txt	删除对象	文件修改时间的180天	执行成功

> **注意：**
> 腾讯云 COS 强烈提醒您不要针对同一组对象配置多个含冲突条件的生命周期规则，冲突执行可能导致不同的费用表现。

时间条件

指定触发上述操作的时间条件，有两种情况：

- 对于未开启版本控制的存储桶，或已开启版本控制存储桶的最新版本对象。以对象的最后一次修改时间（Last-Modified）为起始时间，经过多少天后执行相应操作。
- 对于已开启版本控制的存储桶，历史版本的沉降和删除，以对象成为历史版本的时间计算，**而非根据对象的最后修改时间计算**。

> **说明：**
> 支持设置最长天数为3650天。

文件修改时间说明

生命周期支持按照对象的修改时间触发规则执行。部分操作会更改文件修改时间，例如对文件的写操作，PUT Object、PUT Object - Copy、POST Object、Complete Multipart Upload 等接口会更新对象的修改时间，通过生命周期沉降对象不会更新修改时间。

执行天数说明

规则中制定的天数以24小时为准，未满24小时不算为1天。
例如，您在1号下午3点配置了一条文件修改后1天就删除的生命周期规则，那么，生命周期任务会在2号0点开始扫描在2号0点以前距离最终修改时间已经超过1天的文件，并执行删除任务。对于在1号当天上传的文件，由于距离最终修改时间没有超过1天，并不会被删除，而是需要等到3号0点，才会被扫描记录并执行删除。

使用方法

关于生命周期的使用方法，请参见 配置生命周期 。

规则生效时间说明

生命周期规则的生效分为每日扫描和执行两个操作：

- 扫描：COS 将在北京时间（GMT+8）每日0点拉取生命周期规则，扫描命中应用范围的所有对象。
- 执行：若扫描到对象达到规则指定日期，执行沉降或删除操作。
  例如，某用户2023年1月20日配置了规则 A，指定 test.txt 修改时间后10天删除。则从2023年1月21日0点起，每日0点扫描 test.txt 的修改时间。假设该文件的最后修改时间为2023年1月15日，则2023年1月26日0点进行的扫描任务将判断该文件已满足删除条件，扫描完成后将执行删除操作。

> **注意：**
> 请勿在规则扫描和执行期间变更规则状态。变更会导致原规则终止，将无法保证沉降或删除操作正确执行。

费用说明

请求次数费用

- 生命周期在执行删除、沉降操作时，会产生请求次数费用。
  - 生命周期执行一次删除对象操作，会收取一次请求次数费用。
  - 生命周期执行一次沉降对象操作（例如，从标准存储转换为低频存储类型），会收取一次请求费用。
- 如果两条或多条生命周期规则，同时命中了同一个对象，并在同一天执行，生命周期优先执行删除或沉降类型最冷的规则（详情参见 最终一致性 ）。同一天内，仅执行一次沉降/删除操作，仅收取一次请求费用。例如：
  - 一条规则将对象沉降为低频存储，一条规则沉降为归档存储，则生命周期仅执行将对象沉降为归档的规则，收取一次请求费用。
  - 一条规则将对象沉降为深度归档存储，一条规则将对象删除，则生命周期仅执行删除规则，收取一次请求费用。
- 如果两条或多条生命周期规则，同时命中了同一个对象，但在不同天执行，生命周期将执行多次操作，相应会收取多次请求费用。

提前删除费用

请注意低频存储存储至少30天、归档存储类型需存储至少90天、深度归档存储类型需存储至少180天。**如果在最短保留时间之前，通过生命周期提前删除或沉降了对象，需要收取提前删除费用。**腾讯云 COS 不会检查生命周期中是否包含少于30/90/180天的规则，所有配置都将按照您的要求执行。

案例1	生命周期操作	收费
低频存储的对象，已存储20天	生命周期执行删除操作	需要额外收取10天的低频存储费用，作为提前删除费用
低频存储额对象，已存储10天	生命周期执行沉降操作，转换为归档存储	需要额外收取20天的低频存储费用，作为提前删除费用

执行说明

- 目前腾讯云对于生命周期的执行效力不提供账单承诺，即对象的计费将会在生命周期执行完成时发生改变。
- 生命周期执行效力不包含意外情况或存储桶中包含大量存量对象的情况，若因为其他情况没有完成，您将可以通过 GET 或 HEAD Object 操作来获取对象的当前状态。
- 大小限制。在低频存储、归档存储和深度归档存储类型分别制定了对象最小占用空间限制。例如，在低频存储中上传小于64KB的文件，将按照64KB计算。为了降低用户成本，生命周期不会对小于64KB对象执行存储类型的转换操作。

生命周期配置元素

最近更新时间：2024-07-05 11:30:41

基本结构

生命周期配置使用 XML 描述方法，其可以配置一条或多条生命周期规则，基本结构如下：

```
<LifecycleConfiguration>
    <Rule>
        <ID>**your lifecycle name**</ID>
        <Status>Enabled</Status>
        <Filter>
            <And>
                <Prefix>projectA/</Prefix>
                <Tag>
                    <Key>key1</Key>
                    <Value>value1</Value>
                </Tag>
            </And>
        </Filter>
        **transition/expiration actions**
    </Rule>
    <Rule>
        ...
    </Rule>
</LifecycleConfiguration>
```

其中每一条规则包含如下内容：

- ID（可选）：可自定义的描述规则的内容。
- Status：可选择规则启用 Enabled 或禁用 Disabled 的状态。
- Filter：用于指定需要操作的对象的筛选条件。
- 操作：需要对符合以上描述的对象执行的操作。
- 时间：支持根据最后修改时间指定天数 Days，或指定某个具体的日期 Date 前修改的对象。

规则描述

Filter 元素

针对存储桶中的所有对象

指定空的筛选条件，将会应用于存储桶中的所有对象。

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
        </Filter>
```

```
        <Status>Enabled</Status>
        **transition/expiration actions**
    </Rule>
</LifecycleConfiguration>
```

针对指定的对象键前缀

指定对象前缀，可以对一部分符合前缀描述的对象组执行操作，例如设置以 logs/ 为前缀的所有对象。

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>logs/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        **transition/expiration actions**
    </Rule>
</LifecycleConfiguration>
```

排除指定的对象键前缀

筛选 Prefix 不匹配该字符串的对象。PrefixNotEquals 最多只能有一个，且覆盖范围应是Prefix的子集。

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>logs/</Prefix>
                <PrefixNotEquals>logs/111</PrefixNotEquals>
            </And>
        </Filter>
        <Status>Enabled</Status>
        **transition/expiration actions**
    </Rule>
</LifecycleConfiguration>
```

针对指定的对象标签

指定某些符合对象标签的 key 和 value 为筛选条件，对特定标签的对象执行操作，例如设置标签的 key=type 和 value=image 为筛选条件的对象：

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>type</Key>
              <Value>image</Value>
```

```
            </Tag>
        </Filter>
        <Status>Enabled</Status>
        **transition/expiration actions**
    </Rule>
</LifecycleConfiguration>
```

针对指定大小的对象生效

- ObjectSizeGreaterThan 规定规则所适用的对象的大小必须大于某值（不包括等于）,单位为 Bytes。ObjectSizeGreaterThan 的值必须大于等于65536。
- ObjectSizeLessThan 指定规则所适用的对象必须小于某值（不包括等于），单位为 Bytes。ObjectSizeLessThan 的值必须大于等于65537，并且 ObjectSizeLessThan 的值必须大于 ObjectSizeGreaterThan。

例如，针对文件大小大于 1MB （1048576 Bytes）、小于100MB （104857600 Bytes）的对象执行操作：

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>logs/</Prefix>
                <ObjectSizeGreaterThan>1048576</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>104857600</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        **transition/expiration actions**
    </Rule>
</LifecycleConfiguration>
```

合并使用多个筛选条件

腾讯云对象存储（Cloud Object Storage，COS）支持通过 AND 的逻辑来合并使用多个筛选条件，例如设置以 logs/ 为前缀，同时对象标签的 key=type 和 value=image 为筛选条件的对象：

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>logs/</Prefix>
                <Tag>
                      <Key>type</Key>
                      <Value>image</Value>
                </Tag>
            </And>
        </Filter>
        <Status>Enabled</Status>
        **transition/expiration actions**
```

```
    </Rule>
</LifecycleConfiguration>
```

操作元素

在生命周期规则中，可以对符合条件的一组对象执行一个或多个操作。

转换操作

指定 Transition 操作可以使对象从一个存储类型转换到另一个存储类型，如果存储桶开启了版本控制，其只对当前版本执行操作。最短的 Transition 设置时间为0天。例如，设置30天后沉降至归档存储：

```
<Transition>
    <StorageClass>ARCHIVE</StorageClass>
    <Days>30</Days>
</Transition>
```

过期删除

指定 Expiration 操作可以使符合规则的对象执行过期删除操作，如果存储桶从未启用过版本控制，则将永久删除对象。如果存储桶启用过版本控制，则将为过期的对象**添加一个 DeleteMarker 标记**，并将其设置为当前版本。例如，设置30天后删除对象：

```
<Expiration>
    <Days>30</Days>
</Expiration>
```

未完成的分块上传

> **注意：**
> 不支持在同一条生命周期规则内同时设置指定对象标签和清理未完成上传的文件碎片。

指定 AbortIncompleteMultipartUpload 操作可以允许分块上传的指定 UploadId 任务在保持一段时间后删除，其不再提供续传或可被检索的特性。例如，设置7天后清除未完成的分块上传任务：

```
<AbortIncompleteMultipartUpload>
   <DaysAfterInitiation>7</DaysAfterInitiation>
</AbortIncompleteMultipartUpload>
```

非当前版本的对象

在启用过版本控制的存储桶中，转换只会对最新版本执行，过期操作只会添加删除标记，因此 COS 对非当前版本的对象提供了如下操作：
指定 NoncurrentVersionTransition 可以将非当前版的对象在指定时间转换到另一个存储类型。例如，设置历史版本在30天后沉降至归档存储：

```
<NoncurrentVersionTransition>
    <StorageClass>ARCHIVE</StorageClass>
    <Days>30</Days>
</NoncurrentVersionTransition>
```

指定 NoncurrentVersionExpiration 可以将非当前版本的对象在指定时间内过期删除。例如，设置历史版本在30天后删除：

```
<NoncurrentVersionExpiration>
    <Days>30</Days>
</NoncurrentVersionExpiration>
```

指定 ExpiredObjectDeleteMarker 用于清理剩余的删除标记，可填入的值为 true 或 false，此选项的生效由清理过期的历史版本的动作（NoncurrentVersionExpiration）触发。开启了这一功能后，当生命周期删除对象的最后一个历史版本时，将自动清理剩余的多个删除标记。
具体情况如下，当清理过期的历史版本这个动作（NoncurrentVersionExpiration）生效时：

- 若剩余删除标记只有一个，无论是否开启 ExpiredObjectDeleteMarker，都会自动清理最后一个删除标记。
- 若剩余删除标记有多个，COS 会检查是否开启了 ExpiredObjectDeleteMarker。如果是 true，会自动清理剩余的多个删除标记；如果是 false，剩余的多个删除标记不会被清理。

```
<ExpiredObjectDeleteMarker>
    <Days>true|false</Days>
</ExpiredObjectDeleteMarker>
```

时间元素

时间条件的计算，有两种情况：

1. 对于未开启版本控制的存储桶，或已开启版本控制存储桶的最新版本对象。**以对象的最后一次修改时间（Last-Modified）为起始时间，经过多少天后执行相应操作。**
2. 对于已开启版本控制的存储桶，历史版本的沉降和删除，以对象成为历史版本的时间计算，**而非根据对象的最后修改时间计算。**

按天数计算

使用 Days 指定天数，是按照对象的最后修改时间或成为历史版本的时间来计算的。

- 最新版本对象
  - 例如，设置一个对象在1天后转换为归档存储类型，对象于2018-01-01 23:55:00 GMT+8上传，则其将于2018-01-03 00:00:00 GMT+8起进入转换存储类型的处理队列，最晚于2018-01-03 23:59:59 GMT+8前完成转换。
  - 例如，设置一个对象在2天后过期删除，对象于2018-01-01 23:55:00 GMT+8上传，则其将于2018-01-04 00:00:00 GMT+8起进入过期删除的处理队列，最晚于2018-01-04 23:59:59 GMT+8前完成删除。
- 历史版本对象
  - 例如，设置历史版本对象在1天后转换为归档存储。对象于2018-01-01 23:55:00 GMT+8上传，于2018-02-01 23:55:00 GMT+8 成为历史版本，则其将于2018-02-03 00:00:00 GMT+8起进入转换存储类型的处理队列，最晚于2018-02-03 23:59:59 GMT+8前完成转换。

- 例如，设置历史版本对象在2天后过期删除。对象于2018-01-01 23:55:00 GMT+8上传，于2018-02-01 23:55:00 GMT+8 成为历史版本，则其将于2018-02-04 00:00:00 GMT+8起进入转换存储类型的处理队列，最晚于2018-02-04 23:59:59 GMT+8前完成转换。

按指定日期

使用 Date 指定日期，将会在到达特定日期时，对符合筛选条件的所有对象执行该操作。目前仅支持指定 GMT+8 时区，设置为0时的 ISO8601 格式的时间。
例如，2018年01月01日，描述为：2018-01-01T00:00:00+08:00。

配置生命周期

最近更新时间：2023-08-09 18:08:52

适用场景

利用生命周期设置，可以让符合规则的对象在指定的条件下自动执行一些操作。例如：

- 转换存储类型：将创建的对象在指定时间后转换为低频存储类型（STANDARD_IA）、智能分层存储（INTELLIGENT_TIERING）、归档存储类型（ARCHIVE）和深度归档存储类型（DEEP_ARCHIVE）。
- 过期删除：设置对象的过期时间，使对象到期后被自动删除。

详情请参见 生命周期概述 文档和 生命周期配置元素 文档。

使用方法

使用对象存储控制台

您可以使用对象存储控制台配置生命周期，详情请参见 设置生命周期 控制台指南文档。

使用 REST API

您可以直接使用 REST API 配置和管理存储桶中对象的生命周期，详情请参见以下 API 文档：

- PUT Bucket lifecycle
- GET Buket lifecycle
- DELETE Bucket lifecycle

使用 SDK

您可以直接调用 SDK 的生命周期方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

版权所有：腾讯云计算（北京）有限责任公司

托管静态网站

最近更新时间：2024-12-31 17:43:12

静态网站指包含静态内容（例如 HTML）或客户端脚本的网站，用户可以通过 对象存储控制台 对已绑定自定义域名的存储桶，配置静态网站。而动态网站的内容包含 PHP、JSP 或 ASP.NET 等服务器端脚本，需要依赖服务器端处理。腾讯云对象存储（Cloud Object Storage，COS）支持静态网站的托管，不支持服务器端脚本编写。当您需要部署动态网站时，推荐使用 云服务器（Cloud Virtual Machine，CVM） 进行服务端代码部署。

本文将以以下为例介绍托管静态网站的能力。
用户创建名为 examplebucket-1250000000 的存储桶，上传了如下文件：

```
index.html
404.html
403.html
test.html
docs/a.html
images/
```

静态网站

开启前：使用如下默认访问域名访问存储桶，弹出下载提示，可以保存 `index.html` 文件到本地。

```
https://examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/index.html
```

开启后：使用如下访问节点访问存储桶，可以直接在浏览器中查看 `index.html` 的页面内容。

```
https://examplebucket-1250000000.cos-website.ap-guangzhou.myqcloud.com/index.html
```

> **注意：**
> 2024年1月1日后创建的存储桶，不再支持通过 COS 默认域名（包括静态网站域名）直接访问对象，详情请参见 COS 存储桶域名使用安全管理通知（2024年1月施行）。您可通过自定义域名访问对象。详情请参见 开启自定义源站域名。

强制 HTTPS

开启前：请求来源为 HTTP 时，访问节点 URL 保持 HTTP 未加密的传输协议。

```
http://examplebucket-1250000000.cos-website.ap-guangzhou.myqcloud.com
```

开启后：无论请求来源为 HTTP 或 HTTPS，访问节点始终保持 HTTPS 加密的传输协议。

```
https://examplebucket-1250000000.cos-website.ap-guangzhou.myqcloud.com
```

索引文档

索引文档即静态网站的首页，是当用户对网站的根目录或任何子目录发出请求时返回的网页，通常此页面被命名为 `index.html` 。
当用户使用存储桶访问域名（例如 `https://examplebucketbucket-1250000000.cos-website.ap-guangzhou.myqcloud.com` ）访问静态网站时，且未请求特定的页面。在这种情况下，Web 服务器将返回首页。
您的用户访问存储桶包括根目录在内的任何目录，URL 地址以 `/` 为结尾的会优先自动匹配该目录下的索引文档。根级 URL 的 `/` 是可选的，以下任意一个 URL 将返回索引文档。

```
http://www.examplebucket.com/
http://www.examplebucket.com
```

> **注意：**
> 如果存储桶中创建了文件夹，则需要在每个文件夹层级上都添加索引文档。

错误文档

假设您在配置错误文档前，访问以下页面，将返回404状态码，页面上显示为默认的错误页面信息。

```
https://examplebucket-1250000000.cos-website.ap-
guangzhou.myqcloud.com/webpage.html
```

配置错误文档后，访问以下页面，同样返回404状态码，但页面上将显示您所指定的错误页面信息。

```
https://examplebucket-1250000000.cos-website.ap-
guangzhou.myqcloud.com/webpage.html
```

重定向规则

> **说明：**
> 托管静态网站配置重定向规则，替换文档路径必须是存储桶内的对象路径。

配置错误码重定向

假设您为 webpage.html 这个文档设置了**私有读写**的公共访问权限，用户访问该文件时，将返回403错误。
配置403错误码重定向至 403.html 后：浏览器将返回 403.html 的内容。
如果您未配置 403.html 文档，浏览器将返回错误文档或默认错误信息。

配置前缀匹配

> **注意：**
> 前缀匹配不支持通配符。如您希望对前缀为 index1/、index2/ 的两个文件夹进行重定向，不能使用 `index*/` 作为匹配规则，需要分别创建对应的匹配规则。

- 当您将文件夹从 `docs/` 重命名为 `documents/` 后，用户在访问 `docs/` 文件夹会产生错误。所以，您可以将前缀 `docs/` 的请求重定向至 `documents/` 。

- 当您删除了 `images/` 文件夹（即删除了具有前缀 `images/` 的所有对象）。您可以添加重定向规则，将具有前缀 `images/` 的任何对象的请求重定向至 `test.html` 页面。

清单功能概述

最近更新时间：2024-08-28 11:52:51

什么是清单

清单是一种帮助用户管理存储桶中对象的功能，可以有计划地取代对象存储同步 List API 操作。对象存储（Cloud Object Storage，COS）可根据用户的清单任务配置，每天、每周或每月定时扫描用户存储桶内指定的对象或拥有相同对象前缀的对象，并输出一份清单报告，以 CSV 格式的文件存储到用户指定的存储桶中。也支持发起一次性清单任务，立刻产出一份清单报告。文件中会列出存储的对象及其对应的元数据，并根据用户的配置信息，记录用户所需的对象属性信息。
您可以使用清单功能实现包括但不限于以下基本用途：

- 审核并报告对象的复制和加密状态。
- 简化并加快业务工作流和大数据作业。

> **注意：**
> - 用户可以在一个存储桶中配置多个清单任务，清单任务执行过程中并不会直接读取对象内容，仅扫描对象元数据等属性信息。
> - 目前暂不支持金融云地域。

清单参数

用户配置一项清单任务后，COS 将根据配置定时扫描用户存储桶内指定的对象，并输出一份清单报告，清单报告支持 CSV 格式文件。目前 COS 清单报告中支持记录以下信息：

清单信息	描述
AppID	账号的 ID
Bucket	执行清单任务的存储桶名称
fileFormat	文件格式
listObjectCount	列出的对象数量，费用按此项计费，详情请查看 管理功能费用 中的清单功能费用说明
listStorageSize	列出的对象大小
filterObjectCount	筛选的对象数量
filterStorageSize	筛选的对象大小
Key	存储桶中的对象文件名称。使用 CSV 文件格式时，对象文件名称采用 URL 编码形式，必须解码然后才能使用
VersionId	对象版本 ID。在存储桶上启用版本控制后，COS 会为添加到存储桶的对象指定版本号。如果列表

	仅针对对象的当前版本，则不包含此字段
IsLatest	如果对象的版本为最新，则设置为 True。如果列表仅针对对象的当前版本，则不包含此字段
IsDeleteMarker	如果对象是删除标记，则设置为 True。如果列表仅针对对象的当前版本，则不包含此字段
Size	对象大小（以字节 Byte 为单位）
LastModifiedDate	对象的最近修改日期（以日期较晚者为准）
ETag	实体标签是对象的哈希。ETag 仅反映对对象的内容的更改，而不反映对对象的元数据的更改。ETag 可能是也可能不是对象数据的 MD5 摘要。是与不是取决于对象的创建方式和加密方式
StorageClass	用于存储对象的存储类，有关更多信息，请参见 存储类型
IsMultipartUploaded	如果对象以分块上传形式上传，则设置为 True，有关更多信息，请参见 分块上传
Replicationstatus	用于标记对象复制中源端文件和副本文件的状态。源端文件标记：PENDING（待复制）、COMPLETED（复制完成）、FAILED（复制失败）；副本文件标记：REPLICA（已完成复制，生成副本文件）。有关更多信息，请参见 复制行为说明
Tag	对象的标签
Crc64	对象的CRC64信息。
x-cos-meta-*	对象的自定义元数据信息，例如 `x-cos-meta-testheader` 。清单会将相应的对象元数据输出，若对象不包含该元数据，则为空。

周期性清单和一次性清单

清单支持两种类型：周期性清单和一次性清单。

- 周期性清单：支持用户设置好清单规则后，按月、按周或按天产出清单报告。其中，按月需要指定产出的日期（支持设置1~28日），按周的情况下，规则是周几配置的后续则继续按照周几产出。
- 一次性清单：发起一次性清单任务，产出一次报告。当前 COS 不支持对一次性清单任务进行记录查询，因此请注意保存好发起任务的规则 id、清单存储路径等信息，方便后续查找清单报告。

如何配置清单

在配置清单前，您需要了解两个概念：

- 源存储桶：想要开通清单功能的存储桶。
  - 包含清单中所列出的对象。
  - 包含清单的配置。
- 目标存储桶：存储清单的存储桶。
  - 包含清单列表文件。
  - 包含 Manifest 文件，描述清单列表文件的位置。

配置清单主要分为以下步骤：

指定源存储桶中待分析的对象信息

您需要告知 COS 您需要分析哪一些对象信息。因此，在配置清单功能时，需要在源存储桶中配置以下信息：

- 选择对象版本：列出所有对象版本或者仅列出当前版本。若您选择了列出所有对象版本，则 COS 将会把您同名对象的所有历史版本均列入清单报告中，仅选择当前版本，则 COS 仅会记录您的最新版本对象。
- 配置所需分析的对象属性：您需要告知 COS 需要将对象属性中的哪些信息记录到清单报告中，目前支持的对象属性包括账号 ID，源存储桶名称，对象文件名称，对象版本 ID、是否最新版本、是否删除标记，对象大小，对象最新修改日期，ETag，对象的存储类，跨地域复制标记以及是否属于分块上传文件。

配置清单报告的存储信息

您需要告知 COS 按照何种频率导出清单报告，清单报告要存储至哪个存储桶中，并决定是否需要对清单报告进行加密，需要配置信息如下：

- 选择清单导出频率：每日或者每周。您可以通过此配置告知 COS 应该按照何种频率执行清单功能。
- 选择清单加密：不加密或者 SSE-COS。如您选择了 SSE-COS 加密，我们将会对生成的清单报告进行加密。
- 配置清单的输出位置：您需要指定清单报告需要存储的存储桶。

> **注意：**
> 目标存储桶必须和源存储桶位于同一地域，两者可以是同一存储桶。

使用方法

通过控制台配置清单

您可以参见 开通清单功能 控制台文档，了解如何通过控制台配置清单功能。

通过 API 配置清单

使用 API 为指定存储桶开启清单功能时，请参考以下步骤：

1. 创建 COS 角色。
2. COS 角色绑定权限。
3. 开启清单功能。

1. 创建 COS 角色

创建 COS 角色，具体接口信息请参见 CreateRole 。
其中，roleName 必须为：COS_QcsRole。
policyDocument 为：

```
{
    "version": "2.0",
    "statement": [{
        "action": "name/sts:AssumeRole",
        "effect": "allow",
        "principal": {
            "service": "cos.cloud.tencent.com"
        }
    }]
```

```
}
```

2. COS 角色绑定权限

角色权限绑定权限，具体接口信息参见 AttachRolePolicy 。
其中，policyName 为：QcloudCOSFullAccess，roleName 为第1步中的 COS_QcsRole，也可以使用创建 roleName 时返回的 roleID。

3. 开启清单功能

存放清单文件的目标存储桶和源存储桶需要在同一地域，调用接口，开启清单功能，具体接口信息如下：

- 周期性清单
  - 配置周期性清单：PUT Bucket inventory
  - 查看周期性清单规则：GET Bucket Inventory
  - 删除周期性清单规则：DELETE Bucket Inventory
  - 列出周期性清单规则：LIST Bucket Inventory
- 发起一次性清单任务： POST Bucket Inventory

清单报告存储路径

清单报告及相关的 Manifest 相关文件会发布在目标存储桶中，其中清单报告会发布在以下路径：

```
destination-prefix/appid/source-bucket/config-ID/
```

Manifest 相关文件会发布在目标存储桶以下路径：

```
destination-prefix/appid/source-bucket/config-ID/YYYYMMDD/manifest.json
destination-prefix/appid/source-bucket/config-ID/YYYYMMDD/manifest.checksum
```

> **注意：**
> 发起清单任务后，清单结果会逐步更新，待清单目录下的manifest.json文件生成之后，清单结果才会完全生成。

路径中所代表的含义如下：

- **destination-prefix**：是用户在配置清单时设置的“目标前缀”，可用于对目标存储桶中的公共位置的所有清单报告进行分组。
- **source-bucket**：是清单报告对应的源存储桶名称，加入这个文件夹是为了避免在多个源存储桶将各自清单报告发送至同一目标存储桶时可能造成的冲突。
- **config-ID**：是用户在配置清单时设置的“清单名称”，当同一源存储桶设置多个清单报告并将其发送至同一目标存储桶时，可以用 config-ID 区分不同的清单报告。
- **YYYYMMDD**：时间戳，包含生成清单报告时开始存储桶扫描的时间和日期。
- **manifest.json**：指 Manifest 文件。
- **manifest.checksum**：指 manifest.json 文件内容的 MD5。

其中，Manifest 相关文件总共包含两份文件 manifest.json 和 manifest.checksum。

说明：

有关 Mainfest 文件的介绍如下：

- manifest.json 和 manifest.checksum 都属于 Manifest 文件，manifest.json 描述清单报告的位置，manifest.checksum 是作为 manifest.json 文件内容的 MD5。每次交付新的清单报告时，均会带有一组新的 Manifest 文件。
- manifest.json 包含的每个 Manifest 均提供了有关清单的元数据和其他基本信息，这些信息包括：
  - 源存储桶名称。
  - 目标存储桶名称。
  - 清单版本。
  - 时间戳，包含生成清单报告时开始扫描存储桶的日期与时间。
  - 清单文件的格式与架构。
  - 目标存储桶中清单报告的对象键，大小及 md5Checksum。

以下是 CSV 格式清单的 manifest.json 文件中的 Manifest 示例：

```
{
 "sourceAppid": "1250000000",
 "sourceBucket": "example-source-bucket",
 "destinationAppid": "1250000000",
 "destinationBucket": "example-inventory-destination-bucket",
 "fileFormat": "CSV",
 "listObjectCount": "13",
 "listStorageSize": "7212835",
 "filterObjectCount": "13",
 "filterStorageSize": "7212835",
 "fileSchema": "Appid, Bucket, Key, Size, LastModifiedDate, ETag, StorageClass,
IsMultipartUploaded, ReplicationStatus",
 "files": [
  {
   "key":
"cos_bucket_inventory/1250000000/examplebucket/inventory01/04d73d9debc73d9f0bf85af
461abde6c.csv.gz",
   "size": "502",
   "md5Checksum": "7d40288a09c25b302ad6cb5fced54f35"
  }
 ]
}
```

清单一致性

COS 的清单报告提供了新对象和覆盖的 PUT 的最终一致性，并提供了 DELETE 的最终一致性，因此清单报告中可能不包含最近添加或删除的对象。例如，COS 在执行用户配置的清单任务的过程中时，用户执行了上传或删除对象的操作，则这些操作的结果可能不被反映在清单报告中。

如果您需要在对象执行操作之前验证对象的状态，建议用 HEAD Object API 来检索对象元数据，或在对象存储控制台中检查对象属性。

存储桶标签概述

最近更新时间：2024-08-09 15:49:41

概述

存储桶标签是一个键值对（key = value），由标签的键（key）和标签的值（value）与“=”相连组成，例如 group = IT。它可以作为管理存储桶的一个标识，便于用户对存储桶进行分组管理。您可以对指定的存储桶进行标签的设定、查询和删除操作。

规格与限制

标签键限制

- 以 qcs:、project、项目等开头的标签键为系统预留标签键，系统预留标签键禁止创建。
- 支持 UTF-8 格式表示的字符、空格和数字以及特殊字符 `+ - = .\_ : / @` 。
- 标签键长度为1 – 127个字符（采用 UTF-8 格式）。
- 标签键区分英文字母大小写。

标签值限制

- 支持 UTF-8 格式表示的字符、空格和数字以及特殊字符 `+ - = .\_ : / @` 。
- 标签值长度为1 – 255个字符（采用 UTF-8 格式）。
- 标签值区分英文字母大小写。

标签数量限制

- 存储桶维度：一个资源最多50个不同的存储桶标签。
- 标签维度：
  - 单个用户最多1000个不同的 key。
  - 一个 key 最多有1000个 value。
  - 同个存储桶下不允许有多个相同的 key。

使用方法

您可以通过控制台、API 的方式设置存储桶标签。

使用对象存储控制台

您如需使用对象存储控制台设置存储桶标签，请参见 设置存储桶标签 控制台指南文档。

使用 REST API

您可以直接通过以下 API 管理存储桶标签：

- PUT Bucket tagging
- GET Bucket tagging
- DELETE Bucket tagging

对象标签概述

最近更新时间：2023-09-21 13:10:44

功能概述

对象标签功能的实现是通过为对象添加一个键值对形式的标识，协助用户分组管理存储桶中的对象。对象标签由标签的键（`tagKey`）和标签的值（`tagValue`）与`=`相连组成，例如`group = IT`。用户可以对指定的对象进行标签的设定、查询、删除操作。

> **注意**
> 对象标签功能为计费项，详细的定价请参见 产品定价 文档。

使用方法

使用对象存储控制台

您可以通过对象存储（Cloud Object Storage，COS）控制台管理对象标签，详情请参见 设置对象标签 。

使用 REST API

您可以通过以下 API 管理对象标签：

- PUT Object tagging
- GET Object tagging
- DELETE Object tagging

规格与限制

标签键限制

- 支持 UTF-8 格式表示的字符、空格和数字[0-9]以及特殊字符 `+ - = .\_ : / @` 。
- 标签键长度范围为1-127个字符（采用 UTF-8 格式）。
- 标签键区分英文大小写。

标签值限制

- 支持 UTF-8 格式表示的字符、空格和数字[0-9]以及特殊字符 `+ - = .\_ : / @` 。
- 标签值长度范围为1-255个字符（采用 UTF-8 格式）。
- 标签值区分英文大小写。

标签数量限制

- 对象维度：一个对象最多10个不同的对象标签。
- 标签维度：无限制。

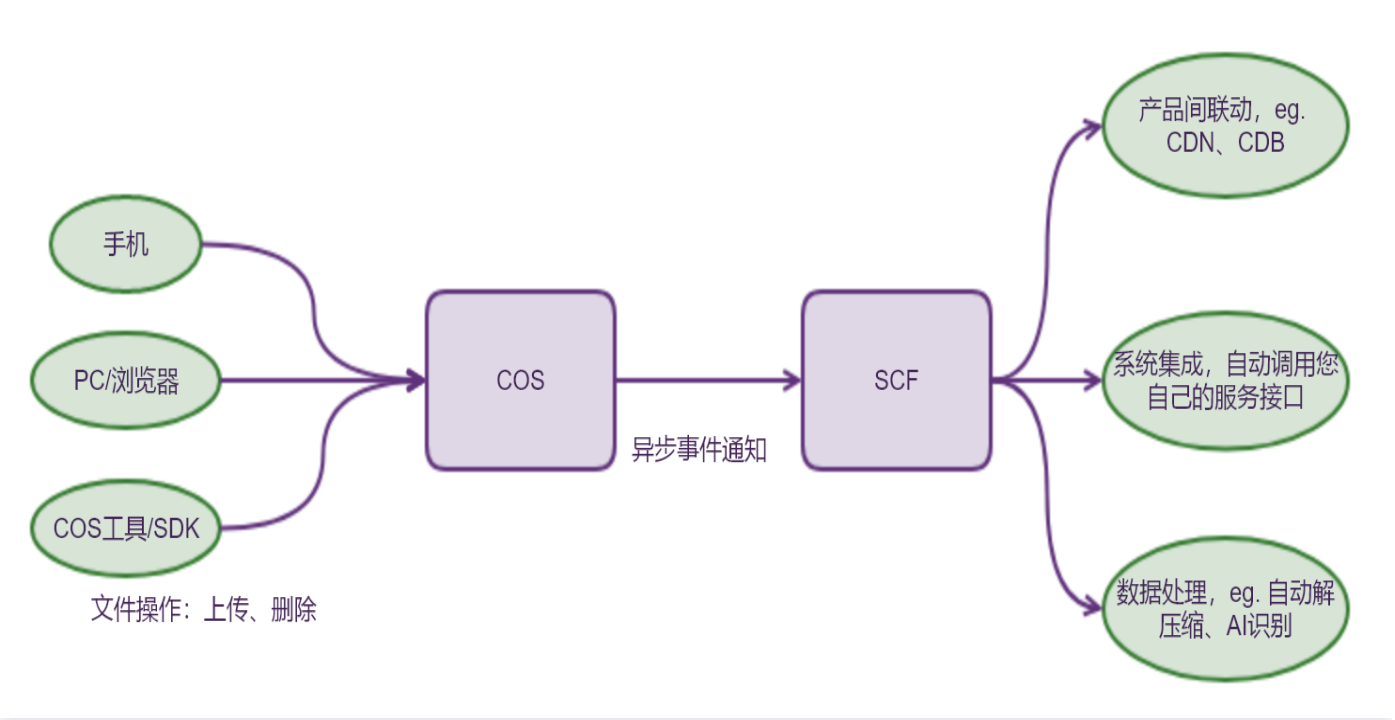
事件通知

最近更新时间：2025-02-26 17:08:12

概述

当对象存储（Cloud Object Storage，COS）资源发生变动（例如新文件上传、文件删除），您可以及时收到通知消息。事件通知可以结合 云函数（Serverless Cloud Function，SCF）实现更丰富的应用场景：

- **产品间联动**：例如，当新文件上传到 COS 后，自动刷新 CDN 缓存。新文件上传到 COS 后，自动更新数据库。
- **系统集成**：当 COS 上的文件发生变更（新建、删除、覆盖），自动调用您自己的服务接口。在 UGC（User Generated Content）场景下，您就可以基于事件通知功能，完成移动端和服务端的联动。
- **数据处理**：对 COS 上的文件进行自动处理，例如，自动解压缩、AI 识别等。



COS 事件通知具有以下特点：

- 异步处理：发送通知不会影响正常的 COS 操作。
- 通知目标：仅支持通知发送至同地域的 SCF 函数。

目前支持以下 COS 事件：

事件类型	描述
cos:ObjectCreated:*	以下提到的所有上传事件完成后，均可触发云函数
cos:ObjectCreated:Put	使用 PUT Object 接口创建文件完成后触发云函数
cos:ObjectCreated:Post	使用 POST Object 接口创建文件完成后触发云函数
cos:ObjectCreated:Copy	使用 PUT Object - Copy 接口创建文件完成后触发云函数

cos:ObjectCreated:Complete MultipartUpload	使用 Complete Multipart Upload 接口创建文件完成后触发云函数
cos:ObjectCreated:Origin	发生镜像回源完成后触发云函数
cos:ObjectCreated:Replication	通过跨地域复制创建对象完成后触发云函数
cos:ObjectRemove:*	以下提到的所有删除事件完成后，均可触发云函数
cos:ObjectRemove:Delete	在未开启版本控制的存储桶下，使用 DELETE Object 接口删除对象，或者使用 versionid 删除指定版本的对象完成后触发云函数
cos:ObjectRemove:DeleteMarkerCreated	在开启或者暂停版本控制的存储桶下，使用 DELETE Object 接口删除对象完成后触发云函数
cos:ObjectRestore:Post	创建了归档恢复的任务完成后触发云函数
cos:ObjectRestore:Completed	完成归档恢复任务完成后触发云函数
cos:TaskComplete:LifecycleCompleted	生命周期执行完成时触发云函数。
cos:TaskComplete:BatchTaskCompleted	批量处理任务执行完成时触发云函数。
cos:InventoryReportCreated:Put	完成清单文件生成时触发云函数。
cos:ObjectAcl:*	全部对象中任何一个被设置访问控制事件时触发云函数。
cos:ObjectAcl:Put	设置某个对象的访问控制事件时触发云函数。
cos:ObjectTagging:*	以下提到的所有对象标签事件均会触发云函数。
cos:ObjectTagging:Put	设置对象标签事件时触发云函数。
cos:ObjectTagging:Delete	删除对象标签事件时触发云函数。

使用 COS 事件通知

使用 COS 事件通知包含以下步骤：

1. 创建 SCF 函数。
   - 您可以通过 SCF 控制台 或 CLI 创建函数。创建函数过程中需要选择运行环境（根据您后续编写函数所使用的语言选择）、提交函数代码（支持在线编辑或本地上传代码包）。
   - 您也可以使用 SCF 预置的模板简化创建流程，详情请参见 创建函数 。不同编程语言的函数写法有所区别，详情请参见 云函数 文档。
2. 测试函数。

   函数创建完成后，您可以使用测试模板功能进行初步测试。测试模板可以模拟 COS 事件，并触发函数执行，详情请参见 测试函数 。
3. 添加触发器。

初步测试完成后，您可以通过创建 COS 触发器来实现绑定 SCF 函数和存储桶。您可以通过控制台或命令行添加触发器，详情请参见 创建触发器 文档。

4. 实际验证。

   完成以上步骤后，您可以操作 COS 中的存储桶，并验证整体流程是否正常。例如，您可以通过控制台、COS Browser 等工具上传、删除文件，并前往 SCF 控制台 > **函数服务** > **对应函数名** > **日志查询**，验证是否正常工作。

关于 SCF、COS 触发器的更多详情，请参见 COS 触发器 文档。

数据检索
Select 概述

最近更新时间：2024-10-14 18:57:21

COS Select 功能通过结构化查询语句（SQL）筛选存储在对象存储（Cloud Object Storage，COS）上的对象，以便检索对象并获取用户所需的数据。通过 COS Select 功能筛选对象数据，您可以减少 COS 传输的数据量，这将降低检索此数据所需的成本和延迟。
COS Select 功能目前支持检索以 CSV、JSON 和 Parquet 格式存储的对象，支持检索通过 GZIP 或 BZIP2 压缩的对象（仅对于 CSV、JSON 格式的对象）。此外，COS Select 功能还支持将结果的格式指定为 CSV 或 JSON，并且可以确定结果中记录的分隔方式。
您可以在请求中将 SQL 表达式传递给 COS。COS Select 目前只支持部分 SQL 表达式。有关 COS Select 支持的 SQL 表达式的更多信息，请参见 SQL 函数 文档。
您可以使用 COS 控制台、API、SDK、COSCMD 等方式执行 SQL 查询。需要注意，使用 COS 控制台进行文件检索存在一定限制：最大支持检索128M文件，返回的数据量限定为40MB。若需检索更多数据，请使用其他方式进行。

> **说明**
> - COS Select 所支持的数据类型和当前的保留字段，详情请参见 数据类型 和 保留字段 。
> - 目前检索功能仅支持中国大陆公有云地域，其他地域暂不支持此功能。

使用限制

使用 COS Select 时存在以下限制：
- 您必须拥有所查询对象的 `cos:GetObject` 权限，主账号默认拥有该权限。
- 仅支持标准存储类型的对象检索。
- SQL 表达式的最大长度为256KB。
- 检索结果中单条记录的最大长度为1MB。

COS Select 功能目前支持的 SQL 子句：
- SELECT 语句
- FROM 子句
- WHERE 子句
- LIMIT 子句

> **说明**
> 有关 SQL 子句的详细信息，请参见 Select 命令 。

COS Select 目前支持的函数例如下：
- 聚合函数：例如 AVG 函数、COUNT 函数、MAX 函数、MIN 函数、SUM 函数。
- 条件函数：例如 COALESCE 函数、NULLIF 函数。
- 转换函数：例如 CAST 函数。

- 日期函数：例如 DATE_ADD 函数、DATE_DIFF 函数、EXTRACT 函数、TO_STRING 函数、TO_TIMESTAMP 函数、UTCNOW 函数。
- 字符串函数：例如 CHAR_LENGTH 函数、CHARACTER_LENGTH 函数、LOWER 函数、SUBSTRING 函数、TRIM 函数、UPPER 函数。

> **说明**
> 有关 SQL 函数的详细信息，请参见 SQL 函数 。

COS Select 目前支持以下运算符：

- 逻辑运算符： `AND，NOT，OR`
- 比较运算符： `<，>，<=，>=，=，<>，!=，BETWEEN，IN`
- 模式匹配运算符： `LIKE`
- 数学运算符： `+，-，*，%`

> **说明**
> 有关运算符的详细信息，您可以参见 运算符 。

发起检索请求

您可以使用控制台、API、SDK 等多种方式发起检索请求：

- 使用控制台，您可参见 检索数据 文档进行操作。
- 使用 API，请参见 SELECT Object Content API 文档。
- 使用 SDK，您可以前往 SDK 概览 选择所需的 SDK 接口。

常见问题

如果在尝试执行查询时遇到问题，COS Select 将返回错误代码和关联的错误消息。有关错误代码和描述的列表，请参见 特殊错误码 。

Select 命令

最近更新时间：2022-04-14 21:42:54

概述

对象存储（Cloud Object Storage，COS）Select 功能仅支持 SELECT SQL 查询指令，以便检索所需的部分数据，减少传输的数据量。这样可以减少成本，同时降低请求延时。以下是 SELECT 查询支持的标准子句：

- SELECT 语句
- WHERE 子句
- LIMIT 子句

> **注意**
> COS Select 当前暂不支持子句查询或者 joins。

SELECT 语句

SELECT 语句可以实现从 COS 对象中检索到您所希望看到的数据，您可以以列的名称、函数或者表达式等维度进行查询，并以一个列表的形式返回查询结果。SELECT 语句调用格式如下：

```
SELECT *
SELECT projection [ AS column_alias | column_alias ] [, ...]
```

第一句 SELECT 语句带有 `*` （星号），将返回 COS 对象中的所有列。第二句 SELECT 语句使用用户自定义输出标量表达式，**projection**为每列创建自定义名称的输出列表。

WHERE 子句

WHERE 子句使用以下语法：

```
WHERE condition
```

WHERE 子句通过 **condition** 进行过滤。**condition** 是一种可以返回布尔结果的表达式，只有返回值为 TRUE 的行才会在结果中输出。

LIMIT 子句

LIMIT 子句使用以下语法：

```
LIMIT number
```

LIMIT 子句限制每次查询返回的记录数量，您可以通过 **number** 参数指定这个限制。

访问属性

SELECT 和 WHERE 子句可以通过以下任意方式选择查询的字段，您可以根据文件格式是 CSV 还是 JSON 进行选择。

CSV

- **列编号**：您可以通过 `_N` 指定查询第 N 列的数据。对于任意一份 CSV 文件，列编号从1开始递增。如第一列编号为 `_1` ，第二列编号为 `_2` 。在 SELECT 和 WHERE 子句中，通过 `_N` 或者 `alias._N` 指定需要查询的列均为合法的方式。
- **列表头**：如果待查询的 CSV 文件中有表头， SELECT 和 WHERE 子句中可以通过这些表头指定需要查询的列。在 SQL 语句中，您可以在 SELECT 和 WHERE 子句中通过 `alias.column_name` 或者 `column_name` 的方式指定。

JSON

- **文档（Document）**：您可以通过 `alias.name` 的方式访问 JSON 文档。嵌套数组则可以通过如 `alias.name1.name2.name3` 的方式访问。
- **列表（List）**：您可以通过索引访问列表中的元素，索引从0开始编号，并使用 `[]` 操作符。例如，您可以通过 `alias[1]` 访问 JSON 列表中的第2个元素。如果您需要访问嵌套数组，也可以通过如 `alias.name1.name2[1].name3` 这样的方式进行访问。

示例

以下为该示例的数据样本：

```
{
    "name": "Leon",
    "org": "Tencent",
    "projects":
        [
         {"project_name":"project1", "completed":true},
         {"project_name":"project2", "completed":false}
        ]
}
```

- 示例 1：以下为在数据样本中查询 name 的 SQL 语句和查询结果：

```
Select s.name from COSObject s
```

```
{"name":"Leon"}
```

- 示例 2：以下为在数据样本中查询 project_name 的 SQL 语句和查询结果：

```
Select s.projects[0].project_name from COSObject s
```

```
{"project_name":"project1"}
```

表头和属性名称的大小写敏感性

您可以使用双引号来标注 CSV 文件的表头和 JSON 文件的属性名称是否大小写敏感。如果不添加双引号，则说明表头/属性名是大小写不敏感的。在您设定不明确的情况下，COS Select 可能会抛出异常。

- 示例 1：查询表头/属性名称中有"NAME"的对象。
  以下 SQL 示例未使用双引号，表明是大小写不敏感的，由于表中有这个表头，所以最终会成功返回数值。

```
SELECT s.name from COSObject s
```

  以下 SQL 示例使用双引号，表明是大小写敏感的。由于表中实际上未包含这个表头，所以最终会返回400错误 `SQLParsingError` 。

```
SELECT s."name" from COSObject s
```

- 示例 2：查询表头/属性名称中有"NAME"和"name"的对象。
  以下 SQL 示例未使用双引号，表明是大小写不敏感的，由于表中同时存在"NAME"和"name"两个表头，查询指令设定不明确，会抛出异常 AmbiguousFieldName。

```
SELECT s.name from COSObject s
```

  以下 SQL 示例使用了双引号，表明是大小写敏感的，由于表中已经存在"NAME"这个表头，会成功返回查询结果。

```
SELECT s."NAME" from COSObject s
```

使用保留字段作为用户自定义字段

COS Select 的 SQL 表达式具有一些保留字段，包含了函数名称，数据类型，操作符等。在某些情况下，用户可能会使用这些保留字段作为 CSV 文件的列表头或者 JSON 文件的属性名称，这时候可能与保留字段存在冲突。在这种情况下，您可以使用双引号来表明您正在使用自定义字段，否则 COS 将返回 `400 parse error` 。
如您需要查阅完整的保留字段列表，请参见 保留字段 。

- 示例：待查询的对象的表头/属性名称具有一个保留字段"CAST"。
  以下 SQL 示例使用了双引号表明 CAST 是用户自定义字段，将成功返回查询结果。

```
SELECT s."CAST" from COSObject s
```

  以下 SQL 示例未使用双引号表明 CAST 是用户自定义字段，COS 将作为保留字段处理，将返回 `400 parse error` 。

```
SELECT s.CAST from COSObject s
```

标量表达式

在 SELECT 语句和 WHERE 子句中，您可以使用 SQL 标量表达式（返回标量的表达式）。目前 COS Select 支持以下形式：

- **literal**：SQL 文本。
- **column_reference**：column_name 或者 alias.column_name。
- **unary_opexpression**： SQL 一元运算符。
- **expressionbinary_opexpression**：SQL 二元运算符。

- **func_name**：被调用的标量函数的名称。
- **expression** [NOT] BETWEEN **expression** AND **expression**
- **expression** LIKE **expression** [ESCAPE **expression**]

SQL 函数

最近更新时间：2024-12-10 11:38:22

聚合函数

COS Select 支持以下聚合函数：

函数名	参数类型	返回类型
AVG(expression)	INT，FLOAT，DECIMAL	入参为整型时返回 DECIMAL，入参为浮点型时返回 FLOAT，其余情况下返回值与入参一致
COUNT	-	INT
MAX(expression)	INT，DECIMAL	返回值与入参一致
MIN(expression)	INT，DECIMAL	返回值与入参一致
SUM(expression)	INT，FLOAT，DOUBLE，DECIMAL	入参为整型时返回 INT，入参为浮点型时返回 FLOAT，其余情况下返回值与入参一致

条件函数

COS Select 支持以下条件函数。

COALESCE

COALESCE 函数按顺序判断输入的参数，并返回第一个非空的参数值。如果输入参数中不包含非空参数，则函数返回 null 值。

语法

```
COALESCE ( expression, expression, ... )
```

> **说明**
> expression 参数支持传入 INT、String、Float 类型数值、数组或者嵌套函数。

示例

```
COALESCE(1)                 -- 1
COALESCE(1, null)           -- 1
COALESCE(null, null, 1)     -- 1
COALESCE(missing, 1)        -- 1
COALESCE(null, 'string')    -- 'string'
```

```
COALESCE(null)              -- null
COALESCE(null, null)        -- null
COALESCE(missing)           -- null
COALESCE(missing, missing) -- null
```

NULLIF

NULLIF 函数判断两个传入的参数之间的差异，如果两个入参均为相同的值，则返回 NULL，否则返回第一个入参的值。

语法

```
NULLIF ( expression1, expression2 )
```

> **说明**
> expression 参数支持传入 INT、String、Float 类型数值、数组或者嵌套函数。

示例

```
NULLIF(1, 2)              -- 1
NULLIF(1, '1')            -- 1
NULLIF(1, NULL)           -- 1
NULLIF(1, 1)              -- null
NULLIF(1.0, 1)            -- null
NULLIF(missing, null)     -- null
NULLIF(missing, missing) -- null
NULLIF([1], [1])          -- null
NULLIF(NULL, 1)           -- null
NULLIF(null, null)        -- null
```

转换函数

COS Select 支持以下转换函数。

CAST

CAST 函数可以将一种实例转化为另一种实例。这个实例可以是数值，也可以是可以计算成某个确定数值的函数。

语法

```
CAST ( expression AS data_type )
```

> **说明**
> - expression 参数可以是数值、数组、运算符或者可以计算成某个确定数值的 SQL 函数。
> - data_type 参数是转换后的数据类型，例如 INT 类型。目前 COS Select 支持的数据类型，请参见 数据类型 。

示例

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

日期函数

COS Select 支持以下日期函数。

DATE_ADD

DATE_ADD 函数可以为指定时间戳的某个部分（年、月、日、时、分、秒）加上指定时间间隔，并返回一个新的时间戳。

语法

```
DATE_ADD( date_part, quantity, timestamp )
```

说明

- date_part 参数指定时间戳中需要修改的部分，可以是年、月、日、时、分、秒。
- quantity 参数代表需要增加的数值，必须为正整数。
- timestamp 参数为需要修改的时间戳。

示例

```
DATE_ADD(year, 5, `2010-01-01T`)                -- 2015-01-01
DATE_ADD(month, 1, `2010T`)                     -- 2010-02T
DATE_ADD(month, 13, `2010T`)                    -- 2011-02T
DATE_ADD(hour, 1, `2017T`)                      -- 2017-01-01T01:00-00:00
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)          -- 2017-01-02T04:04Z
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:05:05.006Z
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:04:06.006Z
```

DATE_DIFF

DATE_DIFF 函数比较两个合法的时间戳并返回两个时间戳的差值，这个差值可以以指定的时间单位显示。当 timestamp1 的 date_part 值比 timestamp2 大时，返回值为正数。反之，则返回负数。

语法

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

说明

- date_part 参数指定两个时间戳比较的时间单位，可以是年、月、日、时、分、秒。
- timestamp1 参数是输入的第一个时间戳。

- timestamp2 参数是输入的第二个时间戳。

示例

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)            -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)                     -- 0
DATE_DIFF(month, `2010T`, `2011T`)                       -- 12
DATE_DIFF(month, `2011T`, `2010T`)                       -- -12
DATE_DIFF(day, `2010-01-01T23:00T`, `2010-01-02T01:00T`) -- 0
```

EXTRACT

EXTRACT 函数从给定的时间戳中，抽取指定时间单位的数值。

语法

```
EXTRACT( date_part FROM timestamp )
```

说明

- date_part 参数指定需要抽取的时间单位，可以是年、月、日、时、分、秒。
- timestamp 参数是输入的时间戳。

示例

```
EXTRACT(YEAR FROM `2010-01-01T`)                          -- 2010
EXTRACT(MONTH FROM `2010T`)                               -- 1
EXTRACT(MONTH FROM `2010-10T`)                            -- 10
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`)            -- 3
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`)          -- 4
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`)   -- 7
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`) -- 8
```

TO_STRING

TO_STRING 函数可以将时间戳转换成指定格式的时间字符串。

语法

```
TO_STRING ( timestamp time_format_pattern )
```

说明

- timestamp 参数指定需要转换的时间戳。
- time_format_pattern 参数指定转换时间格式。

格式	描述	示例
yy	2位数年份	98
y	4位数年份	1998
yyyy	固定4位数年份，不足的以0填充	0199
M	月份	1
MM	固定2位数月份，不足的以0填充	01
MMM	月份的英文缩写	Jan
MMMM	月份的英文全称	January
MMMMM	月份的首字母缩写	J（不适用于 to_timestamp 函数）
d	一个月中的某天（1~31）	1
dd	固定2位数的天数（1~31）	01
a	早上或者下午的标识（AM/PM）	AM
h	时钟，12小时制	1
hh	固定2位数的时刻，12小时制	01
H	时钟，24小时制	1
HH	固定2位数的时刻，24小时制	01
m	分钟（00-59）	1
mm	固定2位数的分钟，24小时制	01
s	秒钟（00-59）	1
ss	固定2位数的时刻，24小时制	01
S	秒钟的小数部分（精度0.1，范围0.0 – 0.9）	0
SS	秒钟的小数部分（精度0.01，范围0.00 – 0.99）	6
SSS	秒钟的小数部分（精度0.001，范围0.000 – 0.999）	60
...	...	...
SSSSSSSSS	秒钟的小数部分（精度0.000000001，范围0.000000000 – 0.999999999）	60000000
n	纳秒	60000000
X	时钟级的偏移，如果偏移为0则为"Z"	+01或者 Z

XX 或者 XXXX	时钟或者分钟级的偏移，如果偏移为0则为"Z"	+0100或者 Z
xxx 或者 xxxxx	时钟或者分钟级的偏移，如果偏移为0则为"Z"	+01:00或者 Z
x	时钟级的偏移	1
xx 或者 xxxx	时钟或者分钟级的偏移	0100
xxx 或者 xxxxx	时钟或者分钟级的偏移	01:00

示例

```
TO_STRING(`1998-07-20T20:18Z`,  'MMMM d, y')                    -- "July 20, 1998"
TO_STRING(`1998-07-20T20:18Z`, 'MMM d, yyyy')                   -- "Jul 20, 1998"
TO_STRING(`1998-07-20T20:18Z`, 'M-d-yy')                        -- "7-20-69"
TO_STRING(`1998-07-20T20:18Z`, 'MM-d-y')                        -- "07-20-1998"
TO_STRING(`1998-07-20T20:18Z`, 'MMMM d, y h:m a')               -- "July 20, 1998
8:18 PM"
TO_STRING(`1998-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX')           -- "1998-07-
20T20:18:00Z"
TO_STRING(`1998-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX')     -- "1998-07-
20T20:18:00Z"
TO_STRING(`1998-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX')   -- "1998-07-
20T20:18:00+0800"
TO_STRING(`1998-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX')  -- "1998-07-
20T20:18:00+08:00"
```

TO_TIMESTAMP

TO_TIMESTAMP 函数可以将时间字符串转换为时间戳。

语法

```
TO_TIMESTAMP ( string )
```

> **说明**
> string 参数为输入的时间字符串。

示例

```
TO_TIMESTAMP('2007T')                          -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

UTCNOW 函数可以返回当前 UTC 制时间戳。

语法

```
UTCNOW()
```

示例

```
UTCNOW() -- 2019-01-01T14:23:12.123Z
```

字符串函数

COS Select 支持以下字符串函数。

CHAR_LENGTH, CHARACTER_LENGTH

CHAR_LENGTH 和 CHARACTER_LENGTH 均可以计算字符串中的字符数量，这两个函数名的语义相同。

语法

```
CHAR_LENGTH ( string )
```

> **说明**
> string 参数指定需要计算字符数的字符串。

示例

```
CHAR_LENGTH('null')       -- 4
CHAR_LENGTH('tencent')    -- 7
```

LOWER

LOWER 函数可以将指定字符串中的所有大写字母转换为小写字母，所有非大写字母将保持不变。

语法

```
LOWER ( string )
```

> **说明**
> string 参数指定一个需要将大写字母转为小写字母的字符串。

示例

```
LOWER('TENcent') -- 'tencent'
```

SUBSTRING

SUBSTRING 函数可以返回一个字符串的子串。您可以指定一个索引， SUBSTRING 函数将从这个索引开始，按照指定的子串长度截取原字符串的剩余内容，并将结果返回。

> **说明**
> 如果输入的字符串只有1个字符，且索引设置大于1，则 SUBSTRING 函数将自动将其切换至1。

语法

```
SUBSTRING( string FROM start [ FOR length ] )
```

> **说明**
> - string 参数指定需要截取子串的字符串。
> - start 参数为字符串的某个索引值，指定截取的位置。
> - length 参数指定子串长度，如未特殊指定将截取字符串剩余所有内容。

示例

```
SUBSTRING("123456789", 0)       -- "123456789"
SUBSTRING("123456789", 1)       -- "123456789"
SUBSTRING("123456789", 2)       -- "23456789"
SUBSTRING("123456789", -4)      -- "123456789"
SUBSTRING("123456789", 0, 999) -- "123456789"
SUBSTRING("123456789", 1, 5)    -- "12345"
```

TRIM

TRIM 函数可以删除指定字符串首字符前或末字符后的所有字符。默认删除字符为" "。

语法

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

> **说明**
> - string 参数指定需要操作的字符串。
> - `LEADING | TRAILING | BOTH` 参数指定需要删除字符串前（LEADING）或者字符串后（TRAILING）或者同时删除（BOTH）的多余字符。

- remove_chars 参数指定需要删除何种多余字符。remove_chars 参数可以是一个长度大于1的字符串，TRIM 函数只要在 string 参数前后发现与之匹配的多余字符串均会做删除处理。

示例

```
TRIM('        foobar          ')                -- 'foobar'
TRIM('       \tfoobar\t          ')             -- '\tfoobar\t'
TRIM(LEADING FROM '        foobar          ')  -- 'foobar'
TRIM(TRAILING FROM '        foobar          ') -- 'foobar'
TRIM(BOTH FROM '        foobar          ')     -- 'foobar'
TRIM(BOTH '12' FROM '1112211foobar22211122') -- 'foobar'
```

UPPER

UPPER 函数可以将所有小写字符转换为大写字符，非小写字符将不会被修改。

语法

```
UPPER ( string )
```

说明
string 参数指定需要转换为大写字符的字符串。

示例

```
UPPER('tenCENT') -- 'TENCENT'
```

保留字段

最近更新时间：2024-10-14 19:10:01

为了保证功能的正常使用和后续扩展，对象存储（Cloud Object Storage，COS）Select 功能制定了以下保留字段，包含函数名称、数据类型、运算符等字段。您在使用 SQL 查询语句时可能使用这些字段辅助您的查询。

序号	字段	序号	字段	序号	字段	序号	字段	序号	字段
1	absolute	51	create	101	goto	151	octet_length	201	struct
2	action	52	cross	102	grant	152	of	202	substring
3	add	53	current	103	group	153	on	203	sum
4	all	54	current_date	104	having	154	only	204	symbol
5	allocate	55	current_time	105	hour	155	open	205	system_user
6	alter	56	current_timestamp	106	identity	156	option	206	table
7	and	57	current_user	107	immediate	157	or	207	temporary
8	any	58	cursor	108	in	158	order	208	then
9	are	59	date	109	indicator	159	outer	209	time
10	as	60	day	110	initially	160	output	210	timestamp

11	asc	61	deallocate	111	inner	161	overlaps	211	timezone_hour
12	assertion	62	dec	112	input	162	pad	212	timezone_minute
13	at	63	decimal	113	insensitive	163	partial	213	to
14	authorization	64	declare	114	insert	164	pivot	214	trailing
15	avg	65	default	115	int	165	position	215	transaction
16	bag	66	deferrable	116	integer	166	precision	216	translate
17	begin	67	deferred	117	intersect	167	prepare	217	translation
18	between	68	delete	118	interval	168	preserve	218	trim
19	bit	69	desc	119	into	169	primary	219	true
20	bit_length	70	describe	120	is	170	prior	220	tuple
21	blob	71	descriptor	121	isolation	171	privileges	221	union
22	bool	72	diagnostics	122	join	172	procedure	222	unique
23	boolean	73	disconnect	123	key	173	public	223	unknown

24	both	74	distinct	124	language	174	read	224	unpivot
25	by	75	domain	125	last	175	real	225	update
26	cascade	76	double	126	leading	176	references	226	upper
27	cascaded	77	drop	127	left	177	relative	227	usage
28	case	78	else	128	level	178	restrict	228	user
29	cast	79	end	129	like	179	revoke	229	using
30	catalog	80	end-exec	130	limit	180	right	230	value
31	char	81	escape	131	list	181	rollback	231	values
32	char_length	82	except	132	local	182	rows	232	varchar
33	character	83	exception	133	lower	183	schema	233	varying
34	character_length	84	exec	134	match	184	scroll	234	view
35	check	85	execute	135	max	185	second	235	when
36	clob	86	exists	136	min	186	section	236	whenever

37	close	87	external	137	minute	187	select	237	where
38	coalesce	88	extract	138	missing	188	session	238	with
39	collate	89	false	139	module	189	session_user	239	work
40	collation	90	fetch	140	month	190	set	240	write
41	column	91	first	141	names	191	sexp	241	year
42	commit	92	float	142	national	192	size	242	zone
43	connect	93	for	143	natural	193	smallint	–	–
44	connection	94	foreign	144	nchar	194	some	–	–
45	constraint	95	found	145	next	195	space	–	–
46	constraints	96	from	146	no	196	sql	–	–
47	continue	97	full	147	not	197	sqlcode	–	–
48	convert	98	get	148	null	198	sqlerror	–	–
49	corresponding	99	global	149	nullif	199	sqlstate	–	–

版权所有：腾讯云计算（北京）有限责任公司

50	count	100	go	150	numeric	200	string	-	-

数据类型

最近更新时间：2022-04-14 21:40:47

概述

对象存储（Cloud Object Storage，COS）Select 支持多种基元数据类型。

> **说明**
> 编译器直接支持的数据类型称为**基元数据类型**。

数据类型转换

COS Select 通过 CAST 函数确定您输入数据的数据类型，一般而言，如果您未通过 CAST 函数进行数据类型指定，COS Select 将把输入数据类型视为 string 类型。
如您需要了解更多有关 CAST 函数的信息，可参见 SQL 函数文档中的 CAST 章节。

支持的数据类型

COS Select 支持以下基元数据类型。

名称	描述	示例
bool	TRUE/FALSE	FALSE
int，integer	8字节有符号整数 范围为 -9,223,372,036,854,775,808 - 9,223,372,036,854,775,807	100000
string	UTF-8编码的字符串，字符长度范围为1 - 2,147,483,647	'xyz'
float	8字节浮点数	CAST(0.456 AS FLOAT)
decimal，numeric	十进制数值，最大精度为38位小数，数值范围为 -2^{31} - $2^{31}-1$	123.456
timestamp	时间戳代表了某个确定的时刻，可以支持任意精度。文本格式的时间戳遵循 W3C 规范，但需要以“T”结尾（除非是以“天”为记录单位）。 您使用小数秒时，至少保持1位小数点的精度，小数点后可保持任意位数。 本地时间偏移可以用与 UTC 相比的时分偏移来表示，或者使用"Z"代表与 UTC 相比的本地时间偏移。时间偏移在仅记录日期时不需要展示。	CAST('2007-04-05T14:30Z' AS TIMESTAMP)

运算符

最近更新时间：2022-04-14 21:39:37

对象存储（Cloud Object Storage，COS）Select 支持以下运算符：

运算符种类	运算符
逻辑运算符	AND，NOT，OR
比较运算符	<，>，<=，>=，=，<>，!=，BETWEEN，IN。例如 `IN ('a', 'b', 'c')`
模式匹配运算符	LIKE
数学运算符	+，-，*，%

运算符优先顺序

下表按照降序展示运算符的操作顺序：

运算符/元素	关联性	使用场景
-	右	一元减法
*, /, %	左	乘、除、取模
+, -	左	加、减
IN	-	设置成员资格
BETWEEN	-	在（）范围内
LIKE	-	字符串模式匹配
<>	-	小于、大于
=	右	等价，赋值
NOT	右	逻辑非
AND	左	逻辑与
OR	左	逻辑或

版权所有：腾讯云计算（北京）有限责任公司

日志管理
日志管理概述

最近更新时间：2025-04-03 18:24:22

简介

日志管理功能能够记录对于指定源存储桶的详细访问信息，并将这些信息以日志文件的形式保存在指定的存储桶中，以实现对存储桶更好的管理。
在目标存储桶中，日志记录路径为：

```
目标存储桶/路径前缀{YYYY}/{MM}/{DD}/{time}_{random}_{index}
```

日志每5分钟生成一次，一条记录为一行，每条记录包含多个字段，字段之间以空格分割。需要注意的是，单个日志文件最大为256MB，如果您在这5分钟内产生的日志量超过256MB，那您的日志会被分割成多份日志文件。目前支持的日志字段如下：

字段序号	名称	含义	示例
1	eventVersion	记录版本	1.0
2	bucketName	存储桶名称	examplebucket-1250000000
3	qcsRegion	请求地域	ap-beijing
4	eventTime	事件时间（请求结束时间，UTC 0时 时间戳）	2018-12-01T11:02:33Z
5	eventSource	用户访问的域名	examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
6	eventName	事件名称，即 API 操作名，详情请参见 操作列表	UploadPart
7	remoteIp	来源 IP	192.168.0.1
8	userSecretKeyId	用户访问 KeyId	**********
9	reservedFiled	保留字段	保留字段，显示为 -
10	reqBytesSent	请求字节数（Bytes）	83886080
11	deltaDataSize	请求对存储量的改变（Bytes）	808
12	reqPath	请求的文件路径	/folder/text.txt
13	reqMethod	请求方法	put

14	userAgent	用户 UA	cos-go-sdk-v5.2.9
15	resHttpCode	HTTP 返回码	404
16	resErrorCode	错误码	NoSuchKey
17	resErrorMsg	错误信息	The specified key does not exist.
18	resBytesSent	返回字节数（Bytes）	197
19	resTotalTime	请求总耗时（毫秒，等于响应末字节的时间-请求首字节的时间）	4295
20	logSourceType	日志源类型	USER（用户访问请求），CDN（CDN 回源请求）
21	storageClass	存储类型	STANDARD，STANDARD_IA，ARCHIVE
22	accountId	存储桶所有者 ID	100000000001
23	resTurnAround Time	请求服务端耗时（毫秒，等于响应首字节的时间-请求末字节的时间）	4295
24	requester	访问者账号。 如果访问者为主账号，则账号格式为 `主账号 UIN:主账号 UIN`； 如果访问者为子账号，则账号格式为 `主账号 UIN:子账号 UIN`； 如果访问者为 服务账号，则账号格式为 `被授权的主账号 UIN:角色 ID`； 如果是匿名访问，则显示为 `-`	100000000001:100000000001
25	requestId	请求 ID	NWQ1ZjY4MTBfMjZiMjU4NjRfOWI1N180NDBiYTY=
26	objectSize	对象大小（Bytes）	808，如果您使用分块上传，objectSize 字段只会在完成上传的时候显示，各个分块上传期间该字段显示 `-`
27	versionId	对象版本 ID	随机字符串
28	targetStorageClass	目标存储类型，发起复制操作的请求会记录该字段	STANDARD，STANDARD_IA，ARCHIVE
29	referer	请求的 HTTP referer	`*.example.com` 或者111.111.111.1
30	requestUri	请求 URI	"GET /fdgfdgsf%20/%E6%B5%AE%E7%82%B9%E6%95%B0 HTTP/1.1"

			说明： 如果签名是使用 query_string 形式，那么在 requestUri 中将展示签名，否则 requestUri 中不会包含签名信息。
31	vpcId	VPC 请求 ID	"0"(非VPC)/"12345"(VPC，为不为"0"的 string)

注意：

- 目前 COS 的日志管理功能支持的地域包括北京、南京、上海、广州、成都、重庆、深圳金融、上海金融、中国香港、新加坡、首尔、曼谷、东京、硅谷、弗吉尼亚、法兰克福。
- 日志管理功能要求源存储桶与目标存储桶必须在同一地域。
- 存放日志的目标存储桶可以是源存储桶本身，但不推荐。
- 目前只有 XML API 以及基于 XML API 实现的 SDK、工具等进行的请求访问存储桶时，才会记录日志。JSON API 以及基于 JSON API 实现的 SDK、工具等进行的访问，不会记录日志。
- 根据用户需求和业务发展情况，COS 可能在访问日志中新增字段，请在解析日志时务必做好相应处理。

启用日志管理

使用控制台

用户可以通过控制台快速开启日志管理功能。操作指引请参见 设置日志存储 控制台指南。

使用 API

使用 API 为指定存储桶开启日志管理功能时，请参考以下步骤：

1. 创建日志角色

创建日志角色，具体接口信息请参见 CreateRole 。
其中，roleName 必须为：CLS_QcsRole。
policyDocument 为：

```
{
    "version": "2.0",
    "statement": [{
        "action": "name/sts:AssumeRole",
        "effect": "allow",
        "principal": {
            "service": "cls.cloud.tencent.com"
        }
    }]
}
```

2. 日志角色绑定权限

角色权限绑定权限，具体接口信息参见 AttachRolePolicy 。
其中，policyName 为：QcloudCOSAccessForCLSRole。roleName 为第1步中的 CLS_QcsRole，也可以使用创建 roleName 时返回的 roleID。

3. 开启日志管理

调用接口，开启日志管理功能，具体接口信息请参见 PUT Bucket logging 。其中，要求存放日志的目标存储桶和源存储桶在同一地域。

日志管理限制

最近更新时间：2023-12-07 14:21:21

日志管理功能够针对指定源存储桶记录其详细访问信息，并将这些信息以日志文件的形式保存在指定的存储桶中，以实现对存储桶更好的管理。

目前访问日志管理功能的使用限制如下：

- 投递频率限制：开启日志管理功能后，预计需要1小时完成首批日志投递。每批次的日志文件中按五分钟顺序生成一条记录。
- 投递日志文件大小限制：每次投递的日志文件最大为256MB，超过该限制时将以新的文件投递。
- 投递日志格式：一条记录为一行，每条记录包含多个字段，字段之间以空格分割。
- 投递字段限制：有关投递字段，请参见 日志管理概述 中的说明。
- 无效字段说明：如果您的日志中存在 `-` 这一字符，说明该字段为无效或者缺省记录。

日志记录的内容

- 用户发起的上传、下载、删除对象，以及创建、删除存储桶，修改存储桶配置等请求。
- 用户通过内容分发网络（Content Delivery Network，CDN）分发内容，由 CDN 回到对象存储（Cloud Object Storage，COS）源站拉取数据产生的请求。

日志不记录的内容

- 离线回源请求：用户配置回源后，如果 COS 中无对象，将会到用户指定的源站下载数据，这一下载操作即离线回源请求，不会被记录到日志中。
- 静态网站内部重定向操作：用户在静态网站功能中配置了重定向时，如果访问 index.html 可能会被重定向到其他页面，这类重定向操作不会被记录到日志中。
- 生命周期沉降、删除等操作：如果用户设置了生命周期功能，对对象进行过期沉降或者删除，这些沉降和删除操作由 COS 后端实现，不会记录到日志中。
- 清单列出对象和上传清单报告操作：清单功能代替用户周期性列出存储桶内全部或者指定对象，并将生成的清单报告投递到用户存储桶中，列出对象和投递清单报告的操作将不会被记录到日志中。
- 跨地域复制对象操作：跨地域复制功能需要从源存储桶中获取对象并将对象上传到目标存储桶中，这些操作由 COS 后端实现，不会记录到日志中。
- COS Select 功能中下载对象的操作：COS Select 功能协助用户检索对象，需要先从存储设备中获取数据后才能进行检索，下载对象的操作由 COS 后端实现，不会记录到日志中。

版权所有：腾讯云计算（北京）有限责任公司

使用 COS 存储云产品日志

最近更新时间：2024-11-29 11:00:53

简介

使用腾讯云产品时会产生大量日志，这些日志记录了您的业务情况，有助于您分析业务情况，为您的业务发展和决策提供辅助。您可以利用对象存储（Cloud Object Storage，COS）的存储能力持久化存储云产品日志，同时您可以通过 API、SDK 或者工具等方式，方便快捷地从 COS 上获取日志并进行分析。

使用 COS 存储云产品日志，可以帮您解决以下问题：

- **持久存储**：COS 提供稳定持久的存储服务，您可以以极低的成本将您的日志存放到 COS 实现持久化存储。当您的业务需要基于日志进行分析或者决策时，您可以通过 COS 随时随地获取任意时间段的日志。
- **数据检索**：COS 提供 Select 功能，您可以使用该功能对存储在 COS 上的日志进行简单的检索和抽取工作。您能够结合日志的字段，帮助您检索所需的信息，减少数据下载流量。
- **数据分析**：您可以使用 Sparkling 产品对存储在 COS 上的日志进行分析，您可以选择一个或者多个日志文件，利用 Sparkling 分析日志，根据分析结果进行决策。

日志投递方式

您可以通过两种方式将腾讯云产品的日志存储在 COS 上：

- 使用云产品自带的日志投递功能：例如对象存储 COS、操作审计 CA 等产品，可直接将日志投递到 COS。
- 使用日志服务 CLS 的投递功能：将投递到 CLS 的云产品日志，通过 CLS 投递到 COS 上进行持久化存储。

目前腾讯云产品对这两种方式的支持情况如下：

云产品名称	是否支持直接投递到 COS	是否支持投递到 CLS
操作审计 CA	是	否
负载均衡 CLB	否	是
消息队列 CKafka	是	否
API 网关 APIGateway	否	是
云函数 SCF	否	是
容器服务 TKE	否	是
云直播 CSS	否	是
云开发 TCB	否	是，但不支持通过 CLS 投递到 COS
对象存储 COS	是	支持申请开通白名单，请您加入 技术支持微信群 联系我们开通白名单

直接投递日志到 COS

以下腾讯云产品拥有直接投递日志到 COS 的能力，您可以按照对应的产品文档指引，配置日志投递规则，将日志投递到 COS 。

云产品名称	日志投递文档	日志投递间隔	日志投递路径
操作审计 CA	点此查阅	10-15分钟	cloudaudit/custompreffix/timestamp
消息队列 CKafka	点此查阅	5分钟 – 60分钟 可指定投递间隔	instance id/topic id/timestamp
对象存储 COS	点此查阅	5分钟	路径前缀可自行指定，推荐设置可识别的路径，例如 cos_bucketname_access_log/timestamp

> **说明**
> 消息队列支持投递该产品上产生的消息数据，如果您需要获取创建 CKafka 实例等行为日志，可以选择投递操作审计产品的日志。

通过日志服务（CLS）投递日志到 COS

以下腾讯云产品支持投递日志到日志服务 CLS 中，以便用户进行检索分析。日志服务同时提供了投递到 COS 的产品能力，方便用户持久化存储日志。对于支持投递日志到 CLS 的产品，您也可以通过在 CLS 处开启投递日志到 COS 的方式，将数据持久化存储以降低您的存储成本，方便进一步离线分析。当前支持直接投递日志到 CLS 的产品如下：

云产品名称	日志投递文档
API 网关 API Gateway	点此查阅
容器服务 TKE	点此查阅
云直播 LVB	点此查阅

CLS 投递到 COS 支持如下三种方式投递：

- 通过分隔符格式投递：可以将数据按照分隔符格式投递到 COS，详情请参见 分隔符格式投递 。
- 通过 JSON 格式投递：可以将数据按照 JSON 格式投递到 COS，详情请参见 JSON 格式投递 。
- 通过原文格式投递：可以将数据按照原文格式进行投递，支持单行全文、多行全文投递，部分支持 CSV 格式投递，详情请参见 原文格式投递 。

通过 CLS 投递日志到 COS，您需要执行的操作如下：

1. 根据您业务需要选择对应的产品，按照上述提供的产品日志投递文档链接指引，配置日志集和日志主题，将您业务产生的数据对接到 CLS 中。
2. 此后，根据您业务需要，选择合适的格式将数据投递到 COS 上。在将日志投递到 COS 时，我们建议您按照产品名称作为路径前缀以作为不同产品日志的区分。例如 TKE 的日志，您可以命名为 `TKE_tkeid_log/timestamp` 。
3. 配置好投递规则后，您还可以额外在 SCF 产品下配置文件上传的事件通知，当日志数据投递到 COS 后，您可以根据事件通知执行下一步操作，详情请参见 事件通知 。

对日志进行分析

下载日志到本地进行离线分析

如果您需要下载日志数据到本地，可以通过控制台、SDK、API 或者工具等多种方式进行下载，以下为各下载方式的使用文档说明，您可以参照下述文档，将代码中涉及到文件路径的部分替换为您日志存储路径，即可下载日志数据到本地：

下载方式	使用说明
控制台	点此查阅
cosbrowser	点此查阅
coscmd	点此查阅
Android SDK	点此查阅
C SDK	点此查阅
C++ SDK	点此查阅
.NET SDK	点此查阅
Go SDK	点此查阅
iOS SDK	点此查阅
Java SDK	点此查阅
JavaScript SDK	点此查阅
Node.js SDK	点此查阅
PHP SDK	点此查阅
Python SDK	点此查阅
小程序 SDK	点此查阅
API	点此查阅

使用 COS Select 对日志进行分析

您也可以通过 COS Select 功能直接检索和分析存储在 COS 上的日志文件，前提是日志文件以 CSV 或者 JSON 格式存储。通过 COS Select 功能，您可以筛选出您所需要的日志字段，这将很大程度降低 COS 传输的日志数据量，以减小您的使用成本，同时提高您的数据获取效率。如需详细了解 COS Select 功能，请参见 Select 概述 。
目前您可以通过控制台或者 API 的方式使用 COS Select 功能：

使用方式	使用说明
控制台	点此查阅
API	点此查阅

数据容灾
版本控制
版本控制概述

最近更新时间：2022-04-13 10:21:13

简介

版本控制用于实现在相同存储桶中存放同一对象的多个版本。例如，在一个存储桶中，您可以存放多个对象键同为 picture.jpg 的对象，但其版本 ID 不同，例如100000、100101和120002等。用户在为某一存储桶开启版本控制功能后，可以根据版本 ID 查询、删除或还原存放在存储桶中的对象。这有助于恢复被用户误删或应用程序故障而丢失的数据。例如，用户在对版本控制的对象进行删除操作时：

- 如果需要删除对象（非完全删除），对象存储（Cloud Object Storage，COS）会为被删除的对象插入删除标记，该标记将作为当前对象版本，您可以根据**删除标记**恢复以前的版本。有关删除标记的详细说明，请参见 删除标记 文档。
- 如果需要替换对象，对象存储会为新上传的对象插入新的版本 ID，您仍然可以根据版本 ID 恢复被替换前的对象。

版本控制状态

存储桶可处于如下三种版本控制状态：

- **未启用版本控制状态：**指存储桶的默认初始状态，此时版本控制功能关闭。
- **启用版本控制状态：**指开启存储桶版本控制功能，此时为版本控制开启状态，版本控制状态将应用到该存储桶中的**所有对象**。您对存储桶首次启用版本控制后，新增上传到该存储桶中的对象将拥有唯一的版本 ID。
- **暂停版本控制状态：**指存储桶的版本控制由开启状态变更为暂停状态（无法返回未启用版本控制状态），此后往存储桶中上传的对象将不再存放版本控制的对象。

> **注意**
>
> 1. 一旦您对存储桶启用了版本控制，它将无法返回到未启用版本控制状态（初始状态）。但是，您可以对该存储桶暂停版本控制，这样后续新上传的对象将不会产生多个版本。
> 2. 在启用版本控制之前，存储在存储桶中的对象的版本 ID 均为 null。
> 3. 启用或暂停版本控制时，会改变对象存储在处理这些对象的请求方式，不会改变对象本身。
> 4. 只有主账号和授权子账号可以暂停存储桶的版本控制。

管理版本控制状态下的对象

存储桶处于不同的版本控制状态下，您均可对不同状态的存储桶中的对象进行上传、查询和删除操作。除了未启用版本控制状态，启用版本控制状态和暂停版本控制状态下，查询存储桶中的对象和删除对象的操作还包括不指定版本 ID 和指定版本 ID。

- 未启用版本控制状态下：上传、查询和删除对象等操作方式不变，详情请参见 对象管理 目录下的文档。
- 启用和暂停版本控制状态下：上传、查询和删除对象等操作方式，与以往方式的差别在于引入了版本 ID。其中执行删除对象操作还有“删除标记”的概念。

管理启用版本控制状态下的对象

启用存储桶版本控制前，已存储在存储桶中的对象，其版本 ID 为 null。启用版本控制后，不会改变存储桶中已有的对象，只会改变 COS 处理已有对象的方式（如请求方式）。此时，新上传的同名对象将以不同的版本存在于同一个存储桶中。以下将介绍在已启用版本控制的存储桶中如何管理对象：

> **注意**
> 用户在未启用版本控制和启用版本控制的存储桶中上传对象的方式都是相同的，但其版本 ID 不同。通过第二种方式，对象存储会为对象分配特定版本 ID，而前一种方式上传的对象，其版本 ID 始终为 null。

上传对象

对存储桶启用版本控制后，当用户执行 PUT、POST 或 COPY 操作时，COS 会为存放到该存储桶中的对象自动添加唯一的版本 ID。
如下图所示，在启用了版本控制的存储桶中上传对象时，对象存储为该对象添加唯一的版本 ID。

列出版本控制对象

对象存储在与存储桶关联的 versions 参数中存储对象版本信息。COS 按照存储时间的先后顺序返回对象版本，最先返回最近存储的版本。

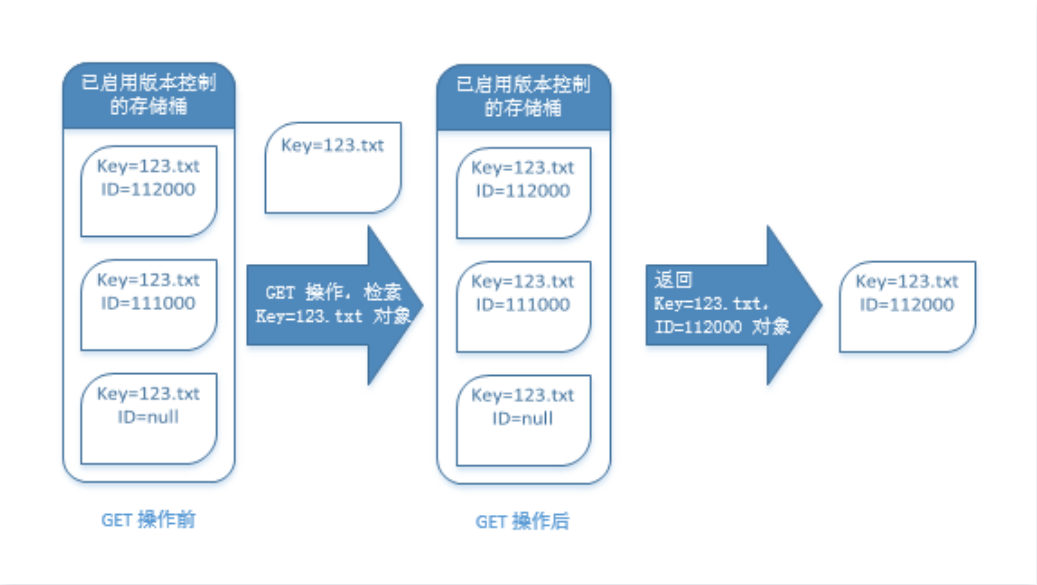
查询特定对象的所有版本

您可以通过以下过程，使用 versions 参数和 prefix 请求参数查询某对象的所有版本。有关 prefix 的更多信息，请参见 GET Bucket Object versions 文档。
查询某一对象的所有版本，请求示例如下：

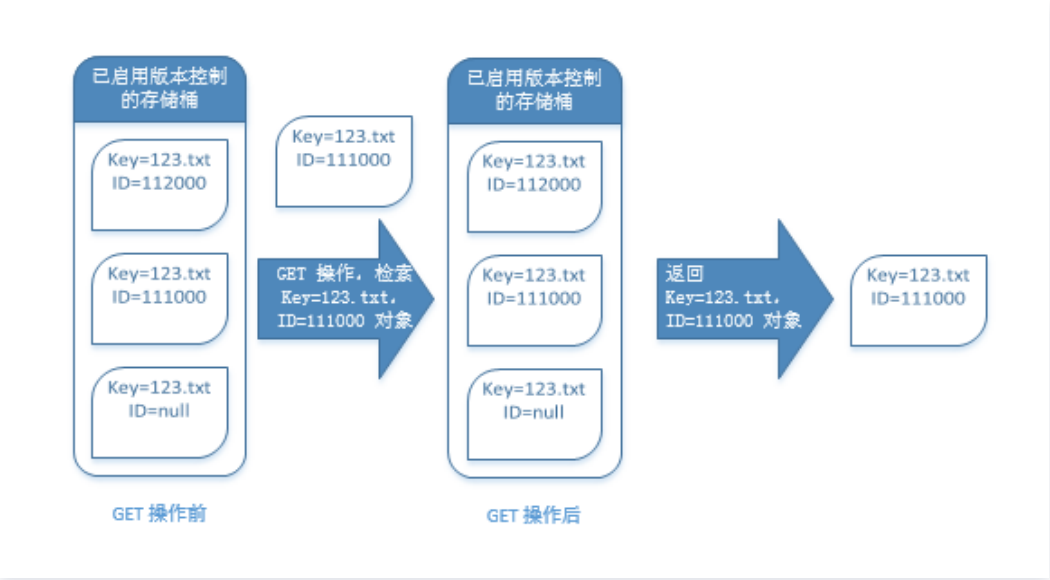
```
GET /?versions&prefix=ObjectKey HTTP/1.1
```

查询数据元版本

用户使用 GET 请求时无指定版本 ID，将查询对象的当前版本。如下图所示，GET 请求将返回 123.txt 对象的当前版本（最近版本）。



如用户使用 GET 请求时指定其版本 ID，将查询指定版本 ID 的对象。如下图所示，GET versionId 请求查询指定版本（可以是当前版本）的对象。



查询对象版本的元数据

如果您只需查询对象的元数据（而不是其内容），您可以使用 HEAD 操作。默认情况下，您将获得最新版本的元数据。如要查询指定对象版本的元数据，则发送请求时需要指定其版本 ID。

查询指定版本的对象的元数据步骤如下：

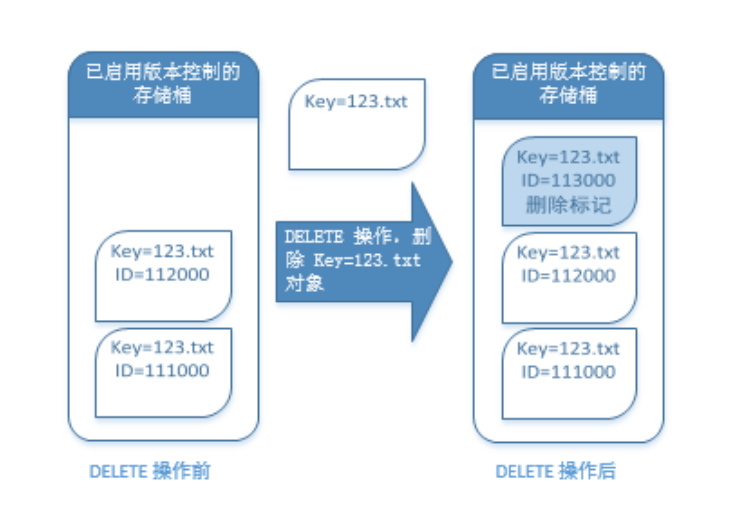
- 将 versionId 配置为被查询对象元数据的版本 ID。
- 发送指定 versionId 的 HEAD 操作请求。

删除对象

您可以根据需要，随时删除不必要的对象版本。用户在已启用版本控制状态下，使用 DELETE 请求有以下两个场景：

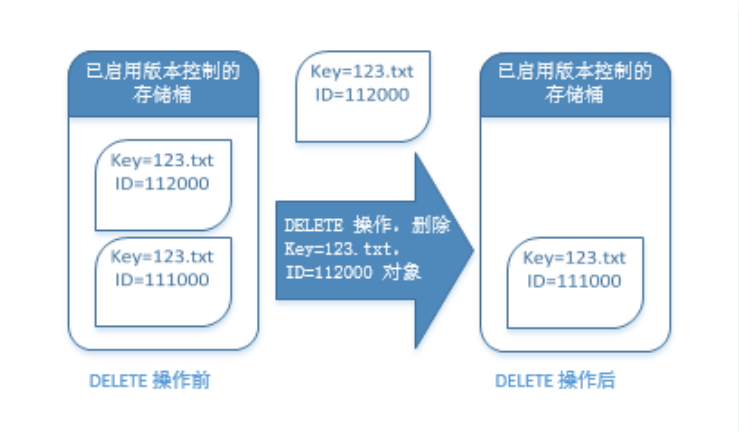
1. 用户未指定版本 ID，执行一般 DELETE 操作。
   此操作场景类似于将被删除对象放到了“回收站”，但没有完全移除对象，后续用户如有需要仍然可以恢复数据。
   如下图所示，用户在 DELETE 操作时不指定版本 ID，实际上不会删除 Key=123.txt 的对象，而是插入一个新的删除标

记，并添加新的版本 ID。



> **注意**
> COS 将在存储桶中为被删除对象插入一个拥有新版本 ID 的删除标记，该删除标记将成为被删除对象的当前版本。当您尝试对该删除标记的对象执行 GET 操作时，对象存储会认为该对象不存在，并返回404错误。

2. 用户指定版本 ID，执行操作删除对象版本，此场景可以永久删除版本控制的对象。



还原早期版本

版本控制能够用来还原对象的早期版本，有两种方法可执行该操作：

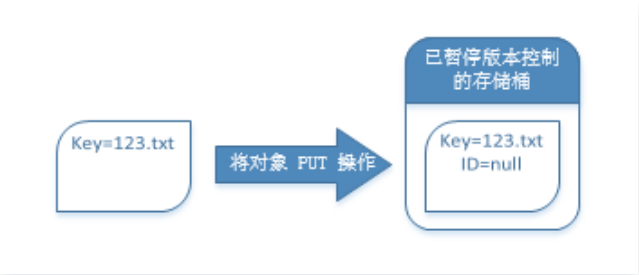
1. 将对象的早期版本复制到同一存储桶中
   复制的对象将成为该对象的当前版本，且所有对象版本都保留。
2. 永久删除对象的当前版本
   当您删除当前对象版本时，实际上会将前一个版本转换为该对象的当前版本。

管理暂停版本控制状态下的对象

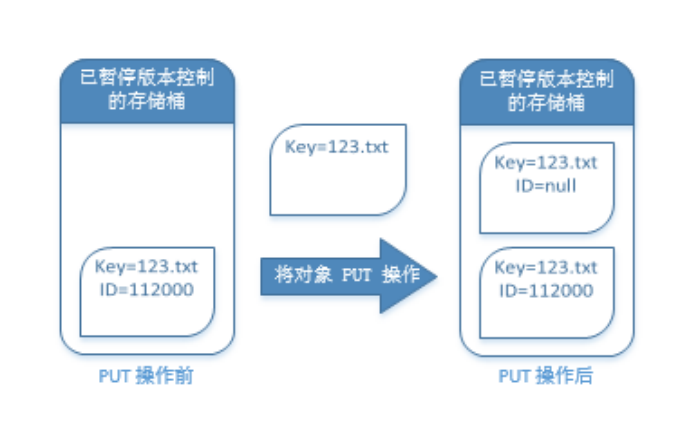
暂停版本控制时，存储桶中的现有对象不会更改。更改的是对象存储在以后的请求中处理对象的方式。以下将介绍在已暂停版本控制的存储桶中如何管理对象。

上传对象

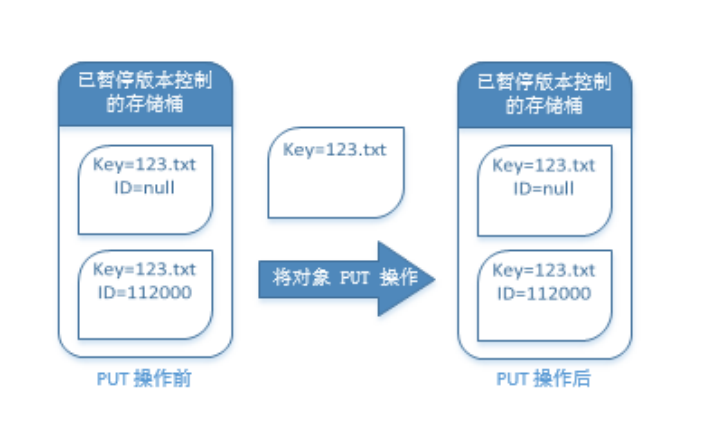
在存储桶上暂停版本控制后，当用户执行 PUT、POST 或 COPY 操作时，COS 自动将版本 ID 为 null 添加到存放到该存储桶中的对象。如下图所示：



如果存储桶中存在版本控制的对象，则上传到存储桶的对象将成为当前版本，并且版本 ID 为 null。如下图所示：



如果存储桶中已存在空版本，则该空版本将被覆盖，原有的对象内容也会相应被替换，如下图所示：



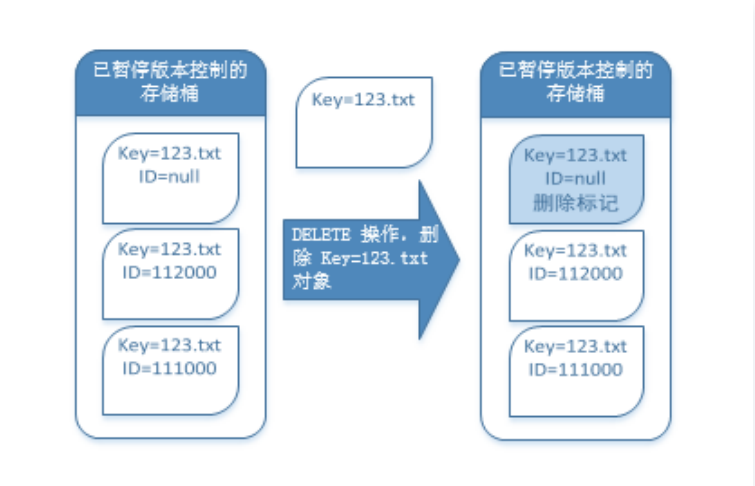
查询数据元版本

在已暂停版本控制的存储桶上，用户发出 GET Object 请求将返回对象的当前版本。

删除对象

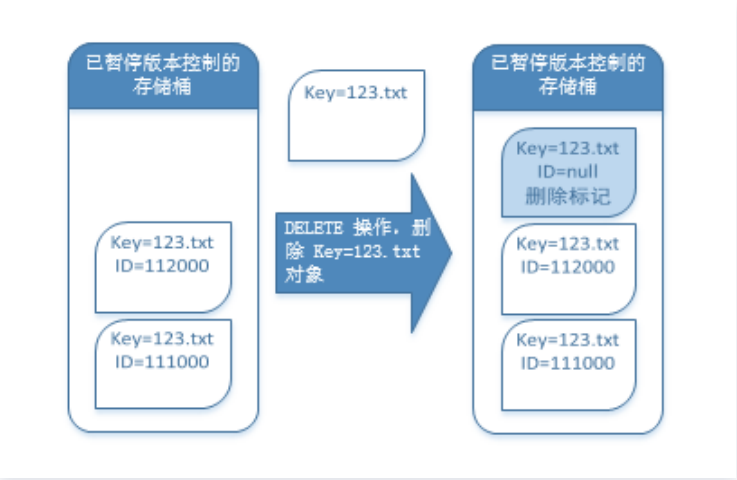
如果暂停了版本控制，执行 DELETE 请求有以下情况：

- 存储桶中存在空版本的对象，将删除其版本 ID 为 null 的对象。
  如下图所示，用户执行一般 DELETE 操作时，COS 会为空版本的对象插入删除标记。

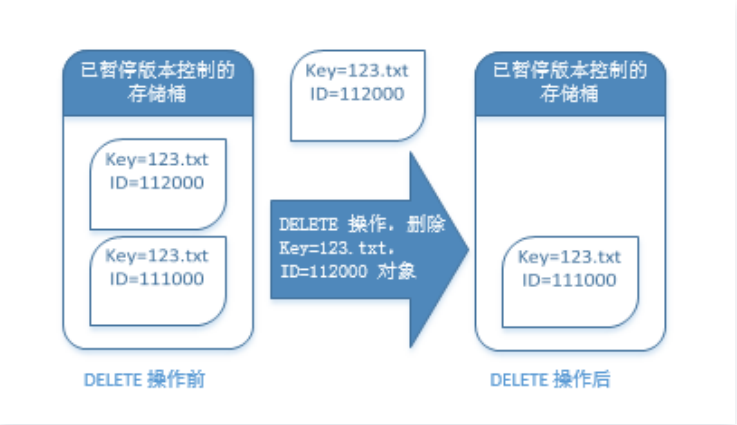


> **注意**
> 删除标记不存在内容，在删除标记替换空版本时，空版本原先的内容会丢失。

- 存储桶中没有空版本的对象，存储桶中会新添加一个删除标记。
  如下图所示，在存储桶不存在空版本的情况下，用户执行 DELETE 操作不会删除任何内容，对象存储仅插入删除标记。



- 即使是在已暂停版本控制的存储桶中，主账号也可以永久删除指定版本。
  如下图所示，删除指定的对象版本将永久删除该对象。



> **注意**
> 只有主账号或被主账号授权的账号才可以删除指定的对象版本。

删除标记

最近更新时间：2023-09-21 20:29:25

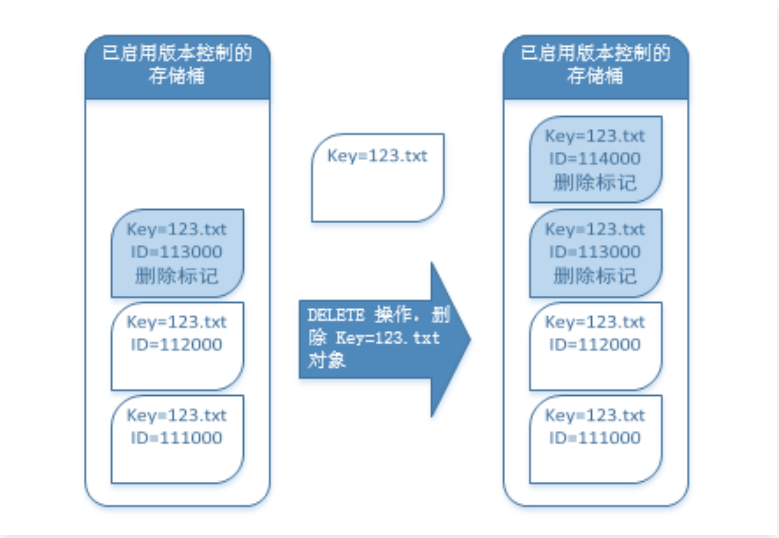
简介

删除标记用于版本控制的对象，删除标记在对象存储（Cloud Object Storage，COS）中可认为是“对象已被删除”的标记。删除标记与对象同样拥有对象键（Key）和版本 ID。区别在于以下几点：

- 删除标记的内容为空。
- 删除标记不存在 ACL 值。
- 删除标记执行 GET 请求会返回404错误。
- 删除标记只支持 DELETE 操作（需要主账号下发出请求）。

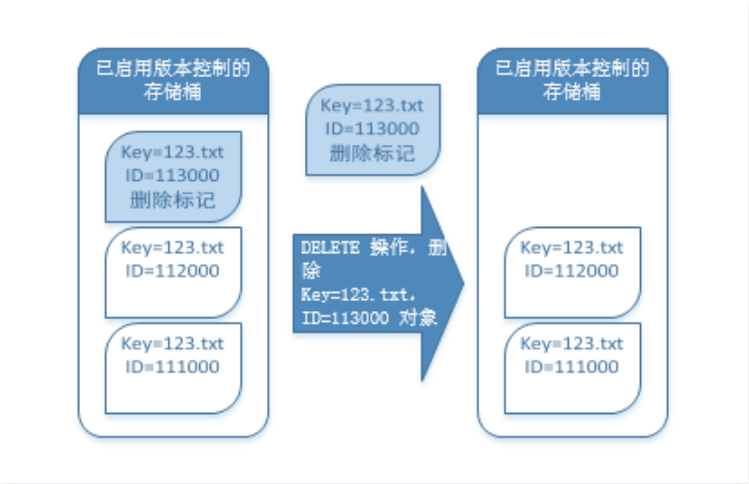
删除“删除标记”

用户如需删除“删除标记”，则可以在 DELETE Object versionId 请求中指定它的版本 ID，实现永久删除“删除标记”。如果您未指定删除标记的版本 ID，对删除标记发出 DELETE 请求，COS 将不会删除该删除标记，而是再插入一个新的删除标记。
如下图所示，对删除标记执行一般 DELETE 请求，不会删除任何内容，而在存储桶里新增了一个新的删除标记。



在已启用版本控制的存储桶中，新增的删除标记将具有唯一的版本 ID。因此，在一个存储桶中，同一个对象可能有多个删除标记。要永久删除“删除标记”，必须在 DELETE Object versionId 请求中包含其版本 ID。

如下图所示，执行 DELETE Object versionId 请求永久删除“删除标记”。



注意

只有经过主账号授权 `DeleteObject` 操作后才可删除“删除标记”。

永久删除“删除标记”的步骤：

1. 设置 versionId 为删除标记的版本 ID。
2. 发送 DELETE Object versionId 请求。

使用版本控制
版本控制配置

最近更新时间：2023-08-09 18:08:52

使用场景

利用版本控制功能，您可以在存储桶中存放对象的多个版本，并且实现检索、删除和还原指定版本对象。
了解版本控制详情信息，请参见 版本控制概述 文档。

> **注意**
> 只有主账号和被授权的子账号可以配置存储桶的版本控制状态。

使用方法

使用对象存储控制台

您可以使用对象存储控制台开启版本控制功能，详情请参见 设置版本控制 控制台指南文档。

使用 REST API

您可以直接使用 REST API 配置存储桶的版本控制和管理版本控制状态下存储桶中的对象，请参见以下 API 文档：

- PUT Bucket versioning
- GET Buket versioning
- GET Bucket Object versions
- PUT Object
- GET Object
- DELETE Object
- DELETE Multiple Objects

使用 SDK

您可以直接调用 SDK 的版本控制方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK

- Python SDK
- 小程序 SDK

存储桶复制
存储桶复制概述

最近更新时间：2025-06-16 16:48:02

简介

存储桶复制是针对存储桶的一项配置，通过配置存储桶复制规则，可以在不同存储桶中自动、异步地复制**增量对象**。启用存储桶复制后，对象存储（Cloud Object Storage，COS）将精确复制源存储桶中的对象内容（如对象元数据和版本 ID 等）到目标存储桶中，复制的对象副本拥有完全一致的属性信息。此外，源存储桶中对于对象的操作，如添加对象、删除对象等操作，也将被复制到目标存储桶中。

> **注意：**
> - 开启存储桶复制功能需要同时对源存储桶和目标存储桶均启用版本控制功能。
> - 启用存储桶复制时，除非您在复制数据时明确指定了配置对象副本的存储类型，否则对象副本将保持与源对象相同的存储类型。
> - COS 复制时将复制源存储桶的访问控制列表（ACL），目前 COS 不支持两个不同账号的存储桶互为源存储桶和目标存储桶。

适用场景

- **异地容灾：**COS 为对象数据提供了12个9的可用性，但仍然存在各种不可抗因素如战争、自然灾害等因素导致数据丢失。如果您无法忍受因数据丢失带来的损失，希望显式地在不同存储桶维护一份数据副本，那么您可以通过存储桶复制实现数据的异地容灾，当某个数据中心因为不可抗因素损毁时，另一个存储桶的数据中心仍然可以提供副本数据以供您使用。
- **合规性要求：**COS 默认在物理盘中为数据提供多副本和纠删码等方式保障数据的可用性，但某些行业中可能存在合规性要求，规定您需要在不同的存储桶间保存数据副本。因此启用存储桶复制，可以实现在不同存储桶间复制数据以满足这些合规性要求。
- **减少访问延迟：**当您的客户在不同地理位置访问对象时，您可以通过存储桶复制，在与客户地理位置最近的存储桶中维护对象副本，最大限度上缩短客户的访问延迟，有利于提高您的产品体验。
- **操作原因：**如果您在两个不同存储桶中均具有计算集群，且这些计算集群需要处理同一套数据，则您可以通过存储桶复制在这两个不同的存储桶中维护对象副本。
- **数据迁移与备份：**您可以根据业务发展需要，将业务数据从一个存储桶复制到另一个存储桶，实现数据迁移和数据备份。

注意事项

费用说明

- 复制对象时，会产生读取和写入请求。COS 会计算请求次数并收取请求费用，详情请参见 请求费用 。
- 同地域复制，流量免费；跨地域复制，会产生跨地域复制流量费用。COS 会计算跨地域复制流量大小，并按照源存储桶所属地域的单价计费，详情请参见 流量费用 。
- 复制对象完成后，会产生存储容量费用。COS 会计算对象大小，存储容量费用按照目标对象的存储类型和所属地域计费，详情请参见 存储容量费用 。

- 由于存储桶复制需要用户开启版本控制功能。版本控制功能会使得存储桶中存在对象的多个历史版本，产生较多的存储消耗。如果您希望减少由于存储桶复制和版本控制带来的成本，或者自定义数据保留方法，您可以结合您的行业背景，通过 生命周期管理 手段实现存储成本控制或者自定义数据保留方式。

复制时间限制

COS 复制对象所需的时间取决于对象大小、存储桶地域间的距离，以及对象的上传方式等因素。同步时间根据上述因素差异，在几分钟、几小时或者几天内不等。

- 对象大小。大型对象的复制需要消耗更多的时间，对于大型对象而言，建议使用分块上传的方式以减少对象的上传和同步时间。
- 存储桶地域间的距离。地域间距离更远的同步需要消耗更多的数据传输时间。
- 对象上传方式。简单上传方式不能做并发，只能在一条连接上串行地上传或者下载数据，分块上传方式能够做并发，因此大文件上传时通过分块上传能够加速上传及存储桶复制。有关对象上传方式的详细介绍请参见 简单上传 和 分块上传 文档。

生命周期相关

由于存储桶复制需要用户开启版本控制功能。版本控制功能会使得存储桶中存在对象的多个历史版本，产生较多的存储消耗。如果您希望减少由于存储桶复制和版本控制带来的成本，可以通过 生命周期 定期删除、沉降文件。

- **源桶生命周期的沉降、删除操作不会自动同步到目标桶中。**如果您希望目标存储桶中的对象副本能够遵循和源存储桶中一样的生命周期规则，请在目标存储桶添加与源存储桶相同的生命周期规则。
- 如果您在源存储桶中设置了生命周期规则，某个对象正在进行存储桶复制的同时需要被生命周期规则删除，则对象的存储桶复制仍会完成，目标存储桶中的对象副本仍然保留。
- 对象创建时间。设置了存储桶规则后，目标桶内对象副本的创建时间和源桶保持一致，而非其出现在目标存储桶中的时间。如果您对目标存储桶设置了生命周期规则，生命周期规则是依据此创建时间执行的。

版本控制相关

存储桶复制配置需要用户在源存储桶和目标存储桶中均配置版本控制功能，版本控制功能的详细内容请参见 版本控制概述 。开启版本控制后，需要注意关闭版本控制对存储桶复制功能的影响：

- 如果您尝试在已开启存储桶复制功能的存储桶中禁用版本控制，则 COS 会返回错误并提示您需要先删除存储桶复制规则后再禁用版本控制。
- 如果您尝试在一个目标存储桶中禁用版本控制，则 COS 会提示您关闭版本控制后存储桶复制功能将受影响，如果您仍确认关闭版本控制，则 COS 以该存储桶作为目标存储桶的存储桶复制规则将失效。

复制行为说明

最近更新时间：2024-11-29 14:22:03

本文档介绍用户在对存储桶启用了存储桶复制功能后，对象存储（Cloud Object Storage，COS）会复制的内容和不复制的内容。

复制的内容

在启用了存储桶复制功能的源存储桶中，对象存储将会复制以下内容：

- 添加存储桶复制规则后，用户往源存储桶中新上传的任何对象。
- 对象的元数据和版本 ID 等对象属性信息。
- 有关对象的操作信息，如新增同名对象（等同于新增对象），删除对象等。

> **说明**
> - 如果您在源存储桶中指定删除某个对象版本，即指定了版本 ID，则该操作不会被复制。
> - 如果您在源存储桶中添加了存储桶级别配置，如生命周期规则，则因这些配置引起的对象操作也不会复制到目标存储桶中。

存储桶复制下的删除操作

如果从源存储桶中删除对象，则存储桶复制行为如下所示：

- 不指定对象版本 ID 执行 DELETE 请求时，COS 将源存储桶中添加删除标记，如果选择了同步删除标记， 则存储桶复制会将该标记复制到目标存储桶；选择了不同步删除标记，则目标存储桶不会新增删除标记。两种情况下，目标存储桶均不会删除对应文件，用户可通过指定版本 ID 访问对象历史版本。 关于版本控制和删除标记的详细信息请参见 版本控制概述 文档。
- 指定对象版本 ID 执行 DELETE 请求时，COS 将删除源存储桶中指定的对象版本，但不会在目标存储桶中复制这一删除操作，即 COS 不会在目标存储桶中删除指定的对象版本。此行为可防止恶意删除数据。

不会复制的内容

当源存储桶启用了存储桶复制功能，对象存储不会复制以下内容：

- 启用存储桶复制功能之前已存在的对象内容，即存量数据。
- 已加密的对象的加密信息，即加密对象被复制后将失去加密信息。
- 源存储桶中新增的数据是来自其它存储桶复制的对象数据。
- 源存储桶中新增的归档存储、深度归档存储类型的对象数据。
- 存储桶级别的配置更新行为。
- 生命周期配置执行后的结果。

> **说明**
> - 对象数据在存储桶间的存储桶复制不具备传递性，如果您同时设置了 A 存储桶为源存储桶，B 存储桶为目标存储桶和 B 存储桶为源存储桶，C 存储桶为目标存储桶的两条存储桶复制规则，那么 A 存储桶中的新增对象数据仅会复制到 B 存储桶中，而不会进一步复制到 C 存储桶中。
> - 例如生命周期的配置，当您更新了源存储桶的生命周期配置，COS 不会将这一生命周期配置同步应用到目标存储桶。

- 如果您只对源存储桶配置了生命周期规则，对象存储会为过期对象添加删除标记，而目标存储桶不会复制这些标记。如果您希望目标存储桶能够删除过期对象，则需要您单独对目标存储桶配置与源存储桶相同的生命周期规则。

存储桶复制配置

最近更新时间：2023-08-08 18:13:11

适用场景

通过配置存储桶复制规则，用户可以将对象数据从源存储桶复制到另一指定的目标存储桶中。存储桶复制功能适用于异地容灾、满足行业合规性要求、数据迁移与备份、降低客户访问延迟、方便不同地域的集群访问数据等场景。
特殊场景：

- 多地域备份：支持针对源存储桶配置多条复制规则，复制对象到不同地域的存储桶，实现多地域备份容灾。
- 双向复制：支持通过在源存储桶和目标存储桶中分别创建复制规则，实现两个存储桶的双向复制，实现存储桶数据的同步。

> **注意**
> 开启了版本控制后，新上传的同名对象将会产生历史版本并占用存储空间，因此这些版本的对象同样收取存储费用。

使用方法

使用对象存储控制台

您可以通过对象存储控制台设置存储桶复制规则，请参见 设置存储桶复制 控制台指南文档。

使用 REST API

您可以直接使用 REST API 配置和管理存储桶的存储桶复制规则，具体可参考以下 API 文档：

- PUT Bucket replication
- GET Bucket replication
- DELETE Bucket replication

使用 SDK

您可以直接调用 SDK 的存储桶复制方法，详情请参见下列各语言 SDK 文档：

- Android SDK
- C SDK
- C++ SDK
- .NET SDK
- Go SDK
- iOS SDK
- Java SDK
- JavaScript SDK
- Node.js SDK
- PHP SDK
- Python SDK
- 小程序 SDK

多 AZ 特性概述

最近更新时间：2025-06-25 14:20:22

简介

多 AZ（Multiple Availability Zones）是由 腾讯云对象存储 推出的多 AZ 存储架构，该架构将客户数据分散存储在城市中多个不同的数据中心，当某个数据中心因自然灾害、断电等极端情况导致整体故障时，仍能为客户提供稳定可靠的存储服务。多 AZ 存储提供99.9999999999%（12个9）的数据设计可靠性和高达99.995%的服务设计可用性。用户在上传数据时，只需指定对象的存储类型，即可将对象存放到多 AZ 的地域。

> **注意：**
> - 目前仅支持在**北京、广州、上海、中国香港、新加坡**地域创建多 AZ 存储桶，其他公有云地域后续将陆续支持。
> - 多 AZ 存储产生的存储容量费用比单 AZ 高，详情请参见各地域 产品定价 。
> - 开启存储桶的多 AZ 配置后，不支持关闭。

使用场景

多 AZ 存储架构提供数据中心级别的容灾能力，以下是多 AZ 存储的几个使用场景：

- **灾难恢复：**在发生自然灾害如地震、洪水等不可预测事件时，多 AZ 存储能够在不同地理位置的数据中心之间实现数据的自动备份和恢复，减少灾难带来的损失。
- **高可用性应用：**对于电商平台、在线游戏、社交网络等需要保证高可用性的互联网服务，多 AZ 存储可以提供更稳定的服务，即使在某个数据中心出现问题时，也能快速切换到其他数据中心，保障用户体验。
- **关键业务数据保护：**对于金融、医疗、政府等对数据安全性和业务连续性要求高的行业，多 AZ 存储可以确保在某个数据中心发生故障时，其他数据中心的数据依然安全、业务不中断。

优势介绍

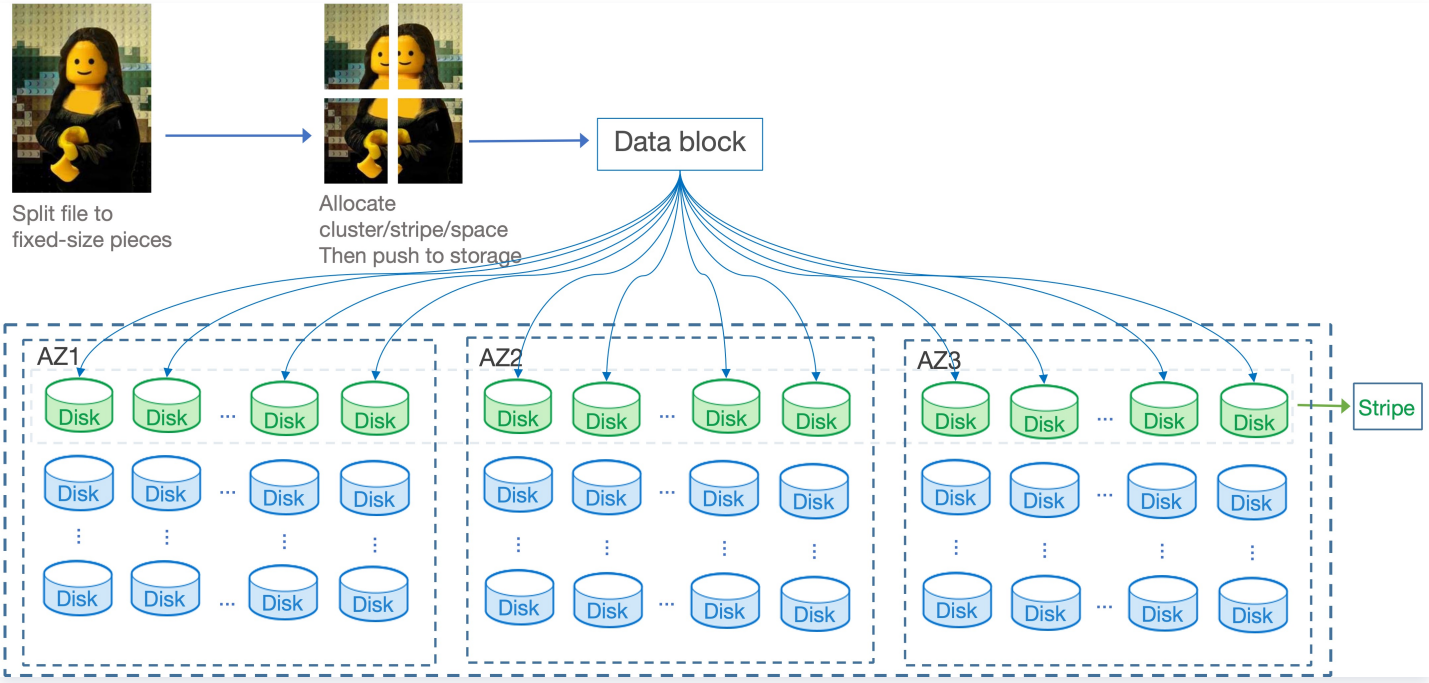
当您将数据存储在多 AZ 地域时，数据会被打散成若干个分块，同时按照纠删码算法计算出对应的校验码分块。原始数据分块和校验码分块会被打散并均分存储到该地域的不同数据中心中，实现同城容灾。当某个数据中心不可用时，另外其他数据中心的数据依旧可以正常读取和写入，保障客户数据持久存储不丢失，维持客户业务数据连续性和高可用。使用对象存储多 AZ 具有以下优势：

- **同城容灾：**提供跨数据中心的容灾。多 AZ 存储架构下，对象数据会被存储在同个地域不同数据中心的不同设备中。当一个数据中心出现故障时，冗余数据中心保持可用，用户业务不受影响，数据不丢失。
- **稳定持久：**采用纠删码冗余存储的方式，提供了高达99.9999999999%的数据设计可靠性；数据分块存储，并发读写，提供高达99.995%的服务设计可用性。
- **便捷易用：**通过对象存储类型指定您的数据存储于何种存储架构，您可以指定存储桶内的任意对象存储到多 AZ 架构中，让使用更为简单。

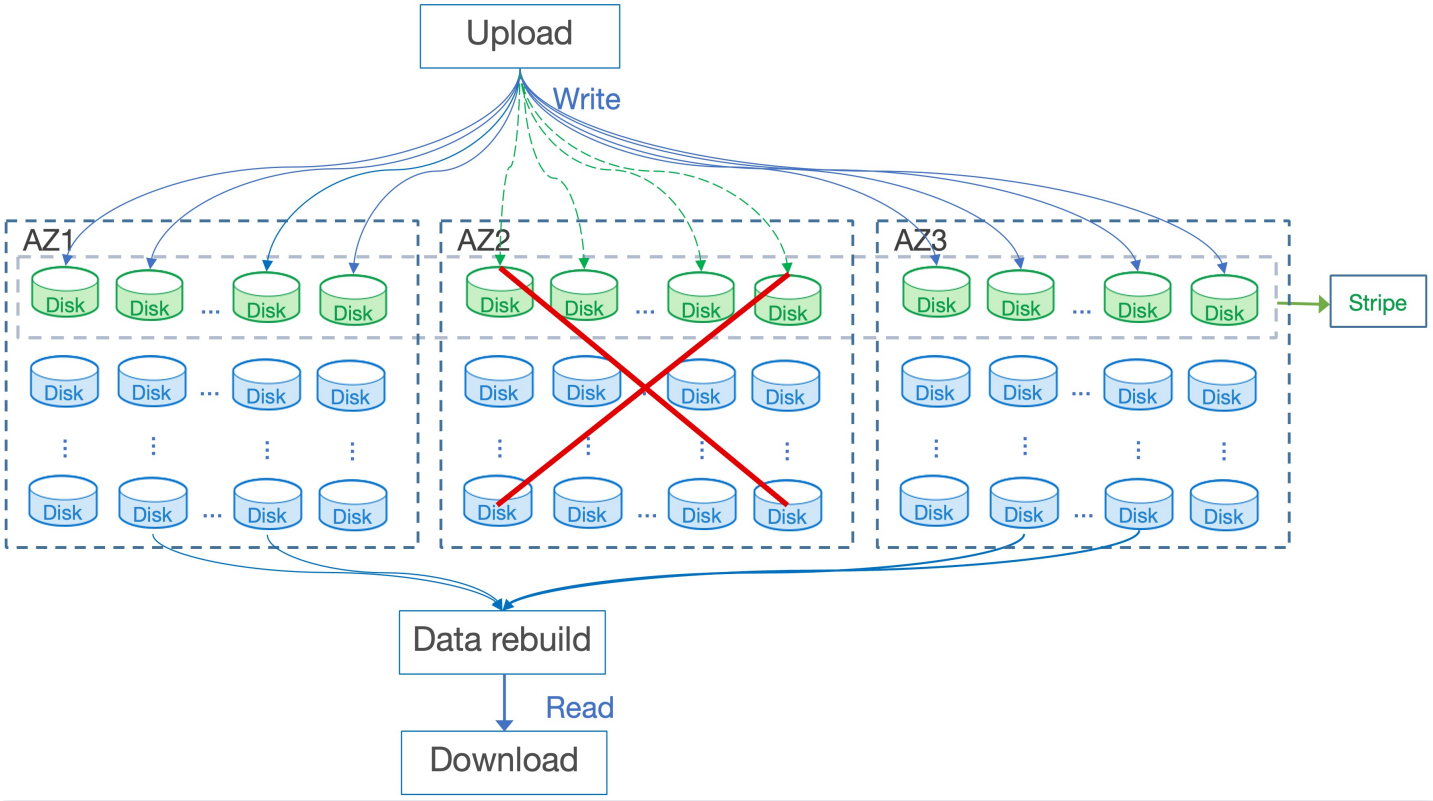
原理介绍

AZ 是 Availability Zone （可用区）的缩写，指的是云服务提供的在物理上隔离、电力和网络独立的区域。多个 AZ 之间互相独立，因此跨 AZ 部署能提高服务的可用性和容灾能力。
对于多 AZ 存储，COS 存储引擎部署在3个环境独立的 AZ 上。用户上传的数据将被切片存储在3个 AZ 的多个节点中。基于 EC 纠删码算法，原始数据会被切分为 N 个数据块，再由算法生成 M 个校验块，由此产生了多个存储分块。COS 将这些存储

分块打散分布在3个 AZ 的不同机架服务器上，就实现了多 AZ 存储。



当机房故障导致部分节点、可用区损坏时，部分分块不可用，多 AZ 存储仍可以利用备 AZ 其他分块保证用户读、写数据成功，确保客户业务连续不中断。



多 AZ 存储和单 AZ 存储对比

多 AZ 存储和单 AZ 存储的 数据持久性、服务可用性、地域、存储类型对比和部分功能限制如下表所示。

对比项	多 AZ 存储	单 AZ 存储	相关文档

数据设计持久性		99.9999999999% （12个9）	99.999999999%（11个9）	–
服务设计可用性		高达99.995%	高达99.99%	–
地域		当前仅支持北京、上海、广州、中国香港、新加坡地域	全部地域	–
存储类型		标准存储（多 AZ）（MAZ_STANDARD） 低频存储（多 AZ）（MAZ_STANDARD_IA） 归档存储（多 AZ）（MAZ_ARCHIVE） 智能分层存储（多 AZ）（MAZ_INTELLIGENT_TIERING）	标准存储（STANDARD） 低频存储（STANDARD_IA） 归档存储（ARCHIVE） 深度归档存储（DEEP_ARCHIVE） 智能分层存储（INTELLIGENT_TIERING）	存储类型概述
功能限制	版本控制	开启版本控制后，不支持暂停	开启版本控制后，支持暂停	• 版本控制概述 • 设置版本控制
	回收站	开启回收站后，不支持暂停	开启回收站后，支持暂停	–
	对象锁定	当前暂不支持开启对象锁定	支持开启对象锁定，当前仅对白名单客户开放	• 对象锁定概述 • 设置对象锁定
	批量处理	暂不支持批量恢复归档类型对象	支持批量数据复制、批量恢复归档类型对象、批量删除	• 批量处理概述 • 批量处理
	Select	不支持	支持	• Select 概述 • 检索数据
	API	不支持用 JSON API 上传/下载对象，请升级至 XML API	XML API、JSON API 均支持，推荐使用 XML API	API 文档简介

使用方法

用户可以为存储桶开启多 AZ 配置，在已开启多 AZ 配置的存储桶中上传对象，可以将对象的存储类型设置为多 AZ。用户在上传对象时，指定对象的存储类型即可将对象存放到多 AZ 存储架构中。

简而言之，您只需要执行以下两步即可将文件存储到多 AZ 架构中：

1. 创建存储桶，并在创建时开启多 AZ 配置。存储桶的创建指引，可参见 创建存储桶 文档。
2. 上传文件，并在上传时指定文件存储类型。文件的上传指引可参见 上传对象 文档。

说明：

- 存储桶多 AZ 配置开启后，无法进行修改，请谨慎配置。原先已创建的存储桶默认未开启多 AZ 配置，仅新创建的存储桶可以开启。
- 如果您希望将存量数据存储于多 AZ 存储桶中，可以新建一个开启多 AZ 配置的存储桶，同时使用 COS Batch 的批量复制功能，将存量存储桶中的文件批量复制到新存储桶中。COS Batch 的使用指引，可参见 批量处理 操作文档。

数据安全
服务端加密概述

最近更新时间：2025-06-16 18:00:52

概述

对象存储（Cloud Object Storage，COS）在数据写入数据中心内的磁盘之前，支持在对象级别上应用数据加密的保护策略，并在访问数据时自动解密。加密和解密这一操作过程都是在服务端完成，这种服务端加密功能可以有效保护静态数据。

> **注意：**
> - 访问加密对象与访问未加密的对象在体验上并无差别，但前提是用户已拥有对象的访问权限。
> - 服务端加密仅加密对象数据而不加密对象元数据。
> - 在列出存储桶中对象时，列表会返回所有对象的列表，无论对象是否加密。

适用场景

- **私密数据存储场景：**对于私密数据的存储，服务端加密可以对存储的数据进行加密，保证用户的隐私，用户访问时会自动解密。
- **私密数据传输场景：**对于私密数据的传输，COS 提供用 HTTPS 部署 SSL 证书实现加密的功能，在传输链路层上建立加密层，确保数据在传输过程中不会被窃取及篡改。

加密方式

COS 支持多种服务端加密方式：SSE-COS、SSE-KMS、SSE-C。用户可以自行选择合适的加密方式对存放到 COS 中的数据进行加密。

SSE-COS 加密

SSE-COS 加密即 COS 托管密钥的服务端加密。由腾讯云 COS 托管主密钥和管理数据。用户通过 COS 直接对数据进行管理和加密。SSE-COS 采用了多因素强加密，确保使用唯一的密钥加密每个对象，并且会通过定期轮换的主密钥来对密钥本身进行加密。
SSE-COS 模式支持 AES256和 SM4两种加密算法，默认使用 AES256算法，上传对象时可以通过服务端加密头部 x-cos-server-side-encryption 指定加密算法。

> **注意：**
> 对于使用预签名 URL 上传的对象，则无法使用 SSE-COS 加密。只能使用 COS 控制台或 HTTP 请求头部指定服务端加密。

SSE-KMS 加密

SSE-KMS 加密即使用 KMS 托管密钥的服务端加密。KMS 是腾讯云推出的一款安全管理类服务，使用经过第三方认证的硬件安全模块 HSM（Hardware Security Module）来生成和保护密钥。它能够帮助用户轻松创建和管理密钥，满足用户多应用多业务的密钥管理需求以及满足监管和合规要求。

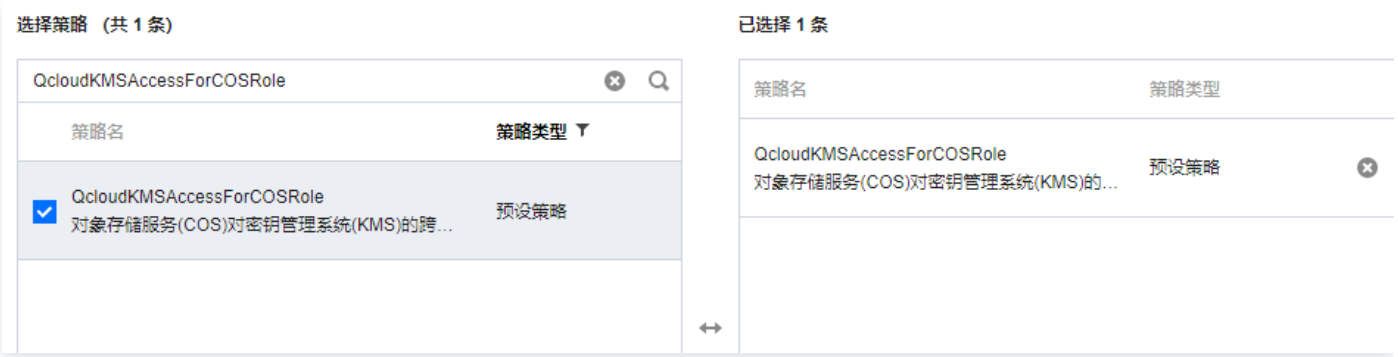
SSE-KMS 模式支持 AES256和 SM4两种加密算法，默认使用 AES256算法，上传对象时可以通过服务端加密头部 x-cos-server-side-encryption-cos-kms-algorithm 指定加密算法。
首次使用 SSE-KMS 加密，需要 开通 KMS 服务，开通 KMS 服务后，系统会自动为您创建一个默认主密钥（CMK）。您也可以通过 KMS 控制台 自主创建密钥，定义密钥策略及使用方法，KMS 支持用户自主选择密钥材料来源为 **KMS 或外部**，更多信息请参见 创建密钥 和 外部密钥导入 。

> **注意：**
> - SSE-KMS 仅加密对象数据，不会加密任何对象元数据。
> - 目前 SSE-KMS 仅支持北京、上海、广州、上海金融、中国香港、曼谷地域。
> - 使用 SSE-KMS 加密，会产生额外费用，由 KMS 收取，详情请参见 KMS 计费概述 。
> - 使用 SSE-KMS 加密的对象必须使用有效签名访问，不可被匿名用户访问。

注意事项

若您未使用过 **COS 控制台**进行 SSE-KMS 加密，而只使用 **API** 的方式进行 SSE-KMS 加密时，您需先创建 CAM 角色，具体创建步骤如下：

1. 登录访问管理控制台，进入 角色 列表页面。
2. 单击**新建角色**，选择角色载体为**腾讯云产品服务**。
3. 选择支持角色的服务为**对象存储**，单击**下一步**。
4. 配置角色策略，搜索并勾选 **QcloudKMSAccessForCOSRole**，单击**下一步**。



5. 标记角色的标签键和标签值，单击**下一步**。
6. 输入指定角色名称：COS_QCSRole。

> **说明：**
> 访问管理 CAM 角色列表中，存量的角色名称“COS_QcsRole”仍可继续使用，生效状态不受影响。

7. 单击**完成**即可创建完毕。

SSE-C 加密

SSE-C 加密即用户自定义密钥的服务端加密。加密密钥由用户自己提供，用户在上传对象时，COS 将使用用户提供的加密密钥对用户的数据进行加密。SSE-C 模式支持 AES256和 SM4两种加密算法，默认使用 AES256算法，上传对象时可以通过服务端加密头部 x-cos-server-side-encryption-customer-algorithm 指定加密算法。

> **注意：**

> COS 不存储用户提供的加密密钥，而是存储加密密钥添加了随机数据的 HMAC 值，该值用于验证用户访问对象的请求。COS 无法使用随机数据的 HMAC 值来推导出加密密钥的值或解密加密对象的内容。因此，如果用户丢失了加密密钥，则无法再次获取到该对象。

使用方法

使用对象存储控制台

用户可以参见 设置对象加密 控制台文档，了解如何通过控制台对对象进行 SSE-COS、SSE-KMS 加密。

> **注意：**
> - SSE-C 加密不支持控制台操作，仅能通过 API 使用。
> - SSE-COS、SSE-KMS 加密的 SM4算法当前不支持控制台、SDK调用，仅能通过 API 使用。

使用 REST API

对于 SSE-COS、SSE-KMS 加密，支持以下 API 接口：

- PUT Object
- Initiate Multipart Upload
- PUT Object - Copy
- POST Object

> **说明：**
> 如果使用非 POST 请求上传对象，可通过提供 `x-cos-server-side-encryption-*` 头部来应用服务端加密；使用 POST 请求，则需在表单字段中提供 `x-cos-server-side-encryption-*` 字段，详情请参见 POST Object。

对于 SSE-C 加密，支持以下 API 接口：

- PUT Object
- Initiate Multipart Upload
- Upload Part
- POST Object
- PUT Object - Copy

> **说明：**
> - 如果使用非 POST 请求上传对象，可通过提供 `x-cos-server-side-encryption-*` 头部来应用服务端加密；使用 POST 请求，则需在表单字段中提供 `x-cos-server-side-encryption-*` 字段，详情请参见 POST Object。
> - 当通过 GET Object 和 HEAD Object 接口请求 SSE-C 加密的对象时，需要提供 `x-cos-server-side-encryption-*` 头部来解密指定对象，详情请参见 公共请求头部 - SSE-C。

使用 SDK

- Android
- C
- .NET(C#)
- Go
- iOS
- Java
- JavaScript
- Node.js
- PHP
- Python
- 小程序

存储桶加密概述

最近更新时间：2023-05-17 11:25:45

简介

存储桶加密是针对存储桶的一项配置，通过设置存储桶加密，可对新上传至该存储桶的所有对象默认以指定的加密方式进行加密。
目前存储桶支持的加密方式有：

- SSE-COS 加密：即由对象存储（Cloud Object Storage，COS）托管密钥的服务端加密。
- SSE-KMS 加密：即使用 KMS 托管密钥的服务端加密。

有关服务端加密的更多信息，请参见 服务端加密概述 。

使用方法

使用对象存储控制台

您可以使用对象存储控制台设置存储桶加密，详情请参见 设置存储桶加密 控制台指南文档。

使用 REST API

您可以通过以下 API 配置存储桶加密：

- PUT Bucket encryption
- GET Bucket encryption
- DELETE Bucket encryption

注意事项

在加密存储桶中上传对象

对于需要配置存储桶加密功能的存储桶，需要注意以下几点：

- 存储桶加密不会对存储桶中已存在的对象产生加密操作。
- 配置了存储桶加密后，对于上传至该存储桶的对象：
  - 如果您的 PUT 请求中不包含加密信息，则上传的对象会以存储桶加密配置来加密。
  - 如果您的 PUT 请求中包含了加密信息，则上传的对象会以 PUT 请求中的加密信息来加密。
- 配置了存储桶加密后，对于投递至该存储桶的清单报告：
  - 若清单本身未配置加密，则投递的清单会以存储桶的加密配置来加密。
  - 若清单本身设置了加密，则投递的清单会以清单的加密配置来加密。
- 配置了存储桶加密后，对于回源至该存储桶的数据，会默认以存储桶的加密配置来加密。

对已配置跨地域复制规则的存储桶进行加密

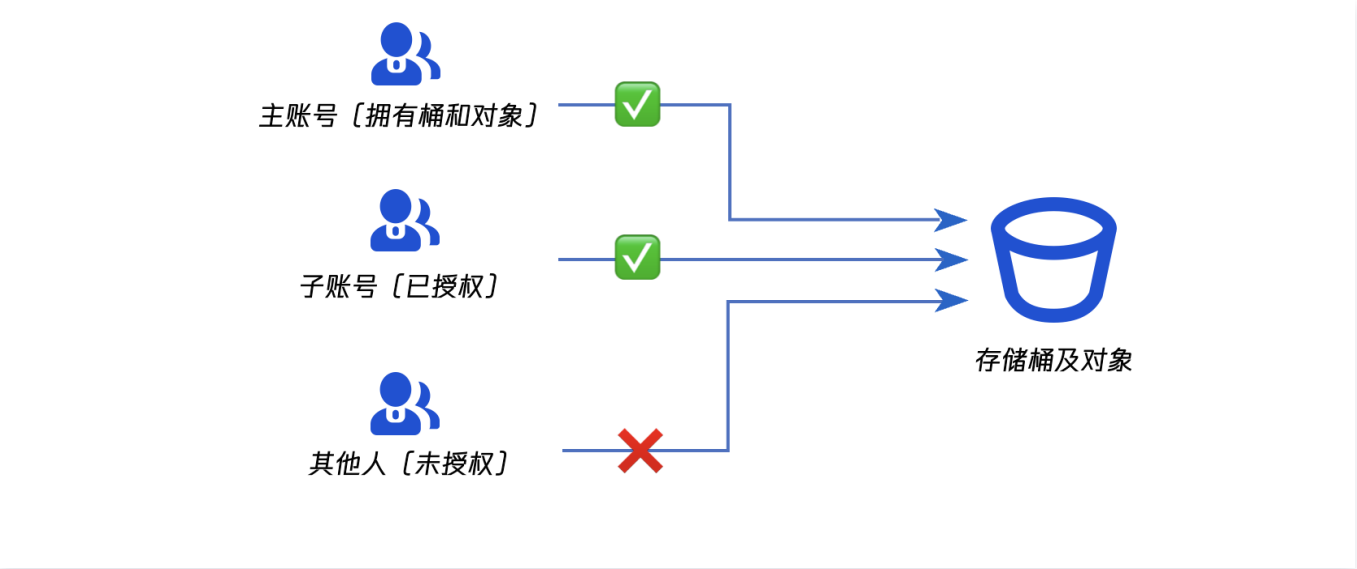
对于已配置跨地域复制规则的目标存储桶，如果再进行配置存储桶加密，需要注意以下几点：

- 如果源存储桶中的对象未加密过，则将对目标存储桶中的副本对象进行默认加密设置。
- 如果源存储桶中的对象已经加密过，则目标存储桶中的副本对象继承源存储桶中的加密，不执行存储桶加密设置。

访问管理
访问权限配置说明
访问控制概述
访问控制基本概念

最近更新时间：2025-01-17 15:48:53

在腾讯云对象存储（Cloud Object Storage，COS）中，资源（存储桶和对象）默认是私有的。只有资源拥有者（腾讯云主账号）可以访问和修改这些资源。其他用户（例如子账号、匿名用户等）在未授权的情况下无法直接通过 URL 访问对象。



资源访问权限概述

- **私有原则**

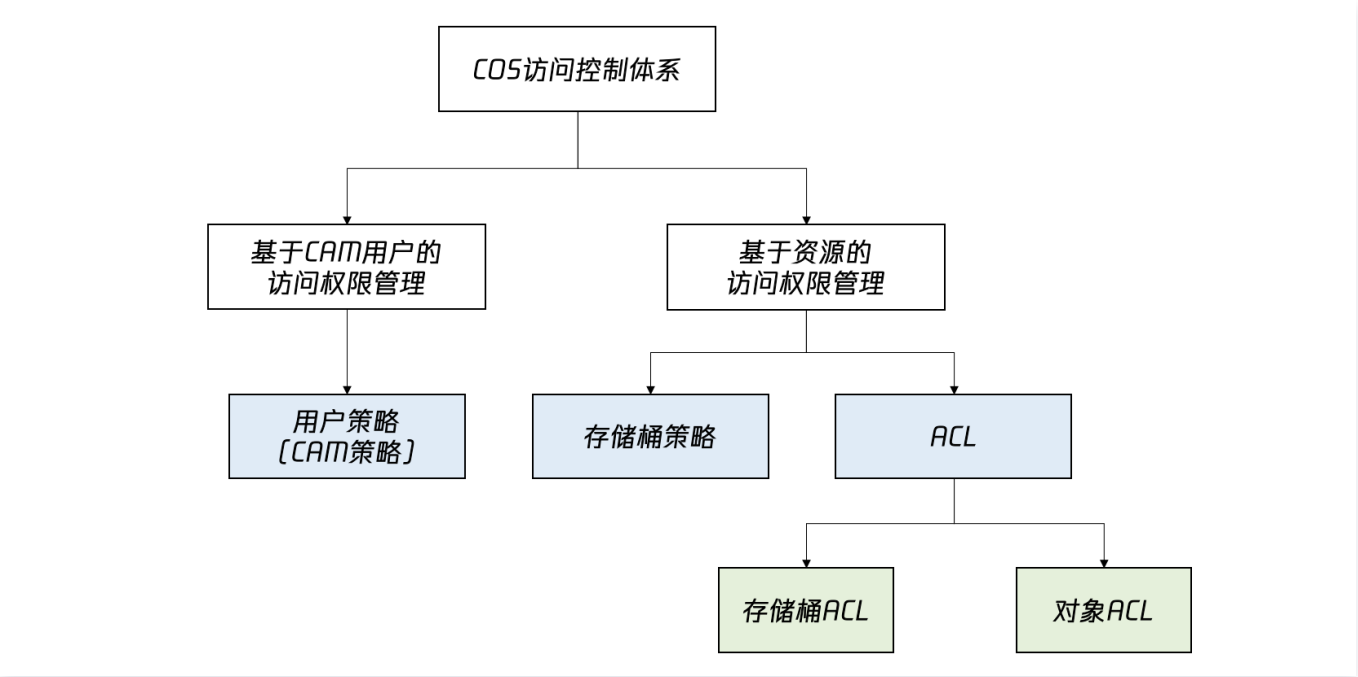
  默认情况下，所有 COS 资源均为私有。
- **资源拥有者权限**
  - 资源拥有者（创建存储桶的主账号）具备最高权限。创建存储桶或上传对象所使用的账号（包括通过腾讯云 访问管理 CAM 所创建的账号）的父级主账号即为资源拥有者。
  - 资源拥有者可创建和编辑访问策略，为其他用户或匿名用户授予访问权限。
- **子账号授权**

  存储桶拥有者可以授权其他腾讯云主账号上传对象的权限（跨账号上传），对象的拥有者依然是存储桶拥有者的主账号。
- **公开权限设置**

  若需将资源开放给非腾讯云用户，可通过设置资源（存储桶、对象、目录）的公共权限（例如公有读）实现。

访问控制体系

COS 访问控制体系如下图所示：



分为基于 CAM 用户的访问权限管理和基于资源的访问权限管理两种维度：

- 基于 CAM 用户的访问权限管理，则对应的是用户策略（CAM 策略）。
- 基于资源的访问权限管理，则对应的是存储桶策略和 ACL ，其中 ACL 包括存储桶 ACL 和对象 ACL。

授权机制

基于用户授权

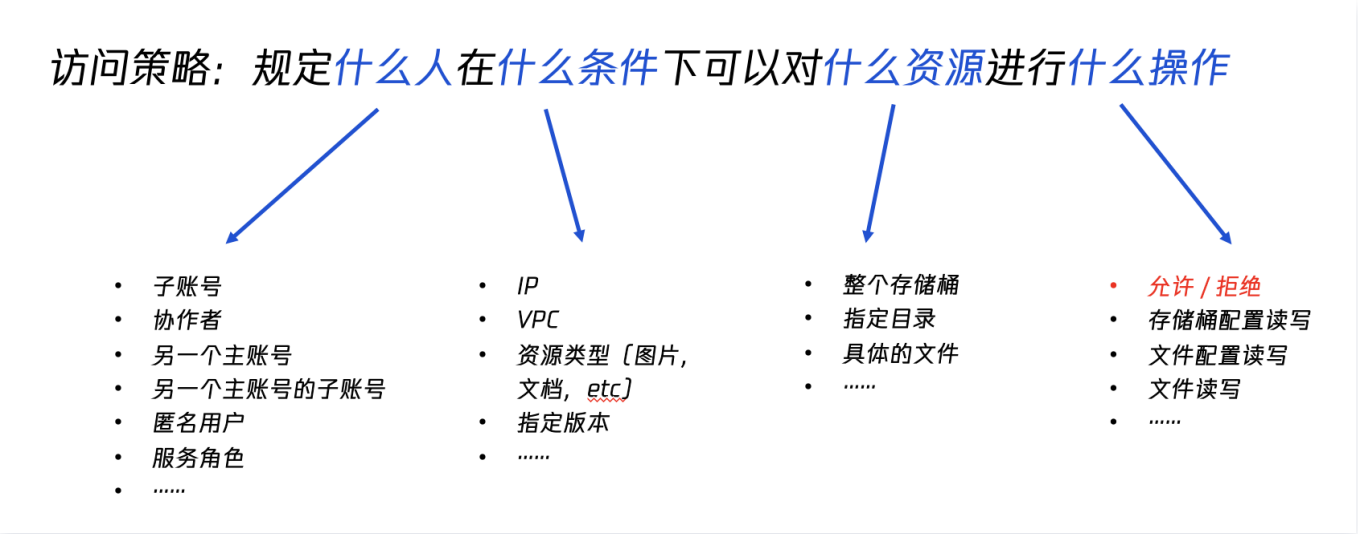
用户策略（CAM 策略）：以用户为中心的权限管理。配置策略时无需指定用户，需要指定资源、操作和条件。可在 CAM 控制台 进行设置。

基于资源授权

资源策略：将权限直接关联到具体资源上，包括存储桶策略、存储桶 ACL 和对象 ACL。可通过 COS 控制台或 COS API 进行配置。

访问控制的元素

授予访问权限指资源拥有者可以决定什么人、在何种条件下、对哪些资源、执行具体操作的控制能力组合。因此描述一个访问权限行为，通常包括四个元素：**身份、资源、操作、条件（可选）**。



腾讯云的身份（Principal）

用户申请腾讯云账号时，系统会创建一个用于登录腾讯云服务的主账号身份。腾讯云主账号可通过用户管理功能对具有不同职责的分类用户进行管理。用户类型包括 **协作者、消息接收人、子用户和角色** 等，具体定义请参见访问管理的 用户类型 和 词汇表 文档。

> **说明**
> 假设您希望为企业的某位同事进行授权，首先需要在 访问管理控制台 创建一个子用户，选择 存储桶策略 、ACL 或 用户策略 中的一个或多个途径为子用户设置具体的权限。

对象存储的资源（Resource）

存储桶 Bucket 和对象 Object 是对象存储的基本资源，其中文件夹是一种特殊的对象，您可以通过文件夹对文件夹下的对象进行授权，详情请参见 设置文件夹权限 。

此外，存储桶和对象都有与之相关的子资源。

资源维度	子资源
存储桶	• acl 和 policy：存储桶的访问控制信息。 • website：存储桶的静态网站托管配置。 • tagging：存储桶的标签信息。 • cors：存储桶的跨域配置信息。 • lifecycle：存储桶的生命周期配置信息。
对象	• acl：对象的访问控制信息。 • restore：归档类型对象的恢复配置。

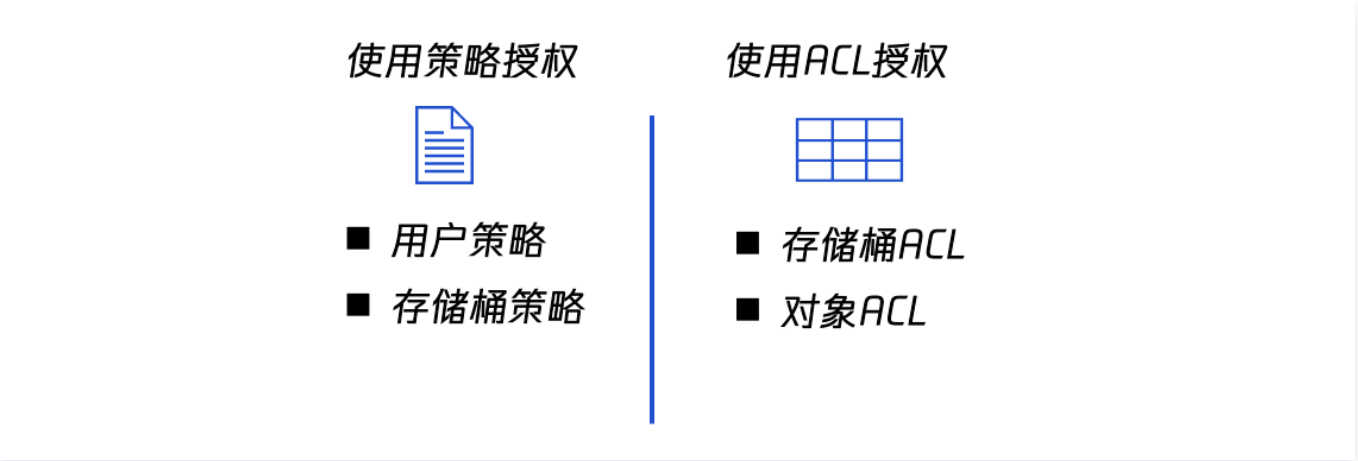
对象存储的操作（Action）

对象存储提供了一系列针对资源的 API 操作，详情请参见 操作列表 文档。

对象存储的条件（Condition，可选）

可选项，权限生效的条件，例如 vpc、vip 等，详情请参见访问管理的 生效条件 。

访问控制方式



- 策略：用户策略（CAM 策略）和存储桶策略，都是基于完整的策略语法进行授权的，支持细化到授权每个 API 操作，支持指定允许/拒绝效力。
- ACL（访问控制列表）：是与资源关联的一个指定被授权者和授予权限的列表，对应经过整理和抽象的权限，仅支持指定允许效力。ACL 分为以下两种：
  - 存储桶 ACL：与存储桶关联的权限列表，指定被授权者及其权限，仅支持允许效力。
  - 对象 ACL：针对对象的权限控制，功能类似于存储桶 ACL。

基于资源的策略

基于资源的策略包括存储桶策略、访问控制列表 ACL，支持在**存储桶**和**对象**级别进行访问控制，具体请参照下表：

维度	类型	描述方式	支持的身份	支持的资源粒度	支持的操作	支持的效力
Bucket	访问策略语言（Policy）	JSON	子账号、角色、腾讯云服务、其他主账号、匿名用户等	存储桶、对象、前缀等	每一个具体的操作	允许/显式拒绝
Bucket	访问控制列表（ACL）	XML	其他主账号、子账号、匿名用户	存储桶	经过整理的读写权限	仅允许
Object	访问控制列表（ACL）	XML	其他主账号、子账号、匿名用户	对象	经过整理的读写权限	仅允许

存储桶策略（Bucket Policy）

存储桶策略（Bucket Policy）使用 JSON 语言描述，支持向匿名身份或腾讯云任何 CAM 账户授予对存储桶、存储桶操作、对象或对象操作的权限，在腾讯云 COS 中存储桶策略可以用于管理该存储桶内的几乎所有操作，推荐您使用存储桶策略来管理通过 ACL 无法表述的访问策略。更多内容可参考 存储桶策略 文档。

> **注意**
> 腾讯云主账号具备对其名下资源（包括存储桶）的最大权限，您虽然可以在存储桶策略中限制几乎所有操作，但主账号始终具备 PUT Bucket Policy 操作的权限，主账号调用该操作不检查存储桶策略。

以下是一个许可匿名用户访问位于广州的存储桶 examplebucket-1250000000中所有对象的策略，无需签名校验即可下载存储桶中的所有对象（GetObject），即任何知晓 URL 的匿名用户都可以下载对象（类似于公有读方式）：

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": ["cos:GetObject"],
      "Resource": ["qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-
1250000000/*"]
    }
  ],
  "Version": "2.0"
}
```

访问控制列表（ACL）

访问控制列表（ACL）使用 XML 语言描述，是与资源关联的一个指定被授权者和授予权限的列表，每个存储桶和对象都有与之关联的 ACL，支持向匿名用户或其他腾讯云的主账号授予基本的读写权限。更多内容可参考 ACL 文档。

> **注意**
> 资源的拥有者始终对资源具备 FULL_CONTROL 权限，而无论下发的 ACL 中是否描述了此项。

存储桶 ACL

描述了存储桶拥有者（用户 UIN：100000000001）的完全控制权限：

```
<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="RootAccount">
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
      </Grantee>
```

```
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

对象 ACL

描述了对象拥有者（用户 UIN：100000000001）的完整控制权限，并授予了所有人可读（匿名用户公有读）的权限：

```
<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee>
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee>
        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

基于用户的策略

用户可以在 CAM 中，对于主账号名下的不同类型用户，授予不同的权限。
用户策略与存储桶策略的最大差别是：用户策略只描述效力（Effect）、操作（Action）、资源（Resource）和条件（Condition，可选），不描述身份（Principal）。因此用户策略需要撰写完成后，再对子用户、用户组或角色执行关联操作，以及用户策略不支持将操作和资源权限授予匿名用户。
您可以 使用预设策略进行关联授权，也可以 自行撰写用户策略 后关联到指定的身份，来实现对名下用户的访问管理。更多内容可参考 用户策略 文档。
以下是一个授权位于广州的存储桶 examplebucket-1250000000 所有 COS 操作的策略，您需要将策略保存后再关联到 CAM 子用户、用户组或角色方可生效：

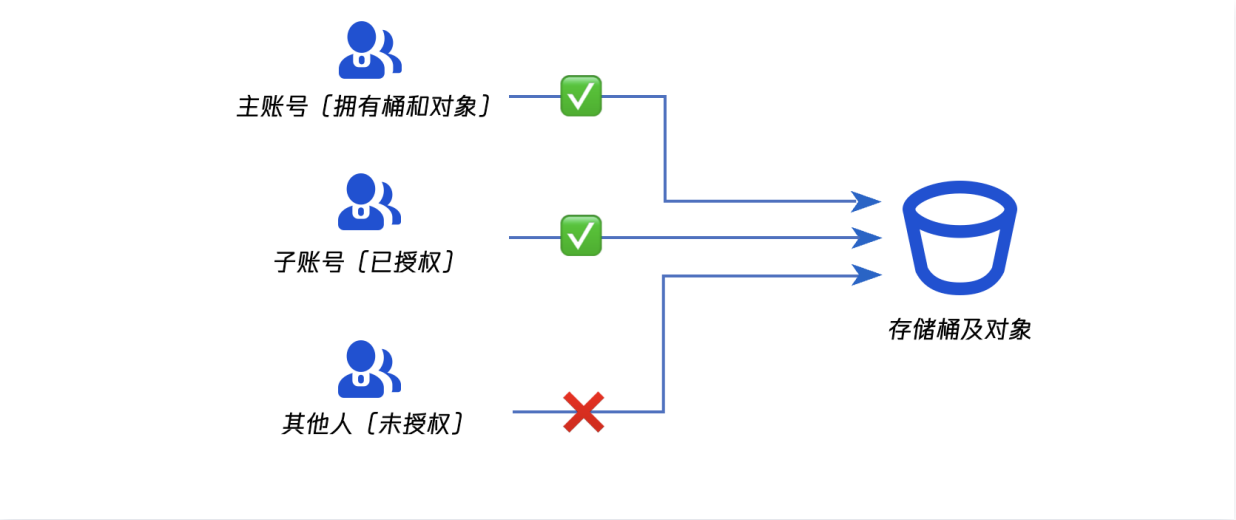
```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["cos:*"],
      "Resource": [
          "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*",
```

```
            "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/"
        ]
      }
    ],
    "Version": "2.0"
  }
```

COS 授权与身份认证流程

最近更新时间：2025-01-17 15:48:53

默认情况下，对象存储（Cloud Object Storage，COS）的资源（存储桶、对象）都是私有读写，即使获取了对象 URL，匿名用户由于未携带签名也无法通过 URL 访问您的资源内容。



操作步骤

从 注册腾讯云账号 开始，COS 的授权与身份认证流程需要经过这五步：注册腾讯云账号、开通 COS 服务、创建授权身份、为身份设置权限、开始访问（身份认证）。

第一步：注册腾讯云账号

注册腾讯云账号 后，您的账号将作为主账号存在，主账号拥有最高权限。

第二步：开通 COS 服务

开通 COS 服务 后，您创建的所有存储桶，都为主账号所拥有。主账号拥有所有资源的最高使用权限，并有权建立子账号并为子账号授权。

第三步：创建授权身份

> **注意**
> 除非将存储桶或对象开放为公有读，否则访问 COS 必须经过身份认证。

通过主账号，您可以创建多种身份，并授予不同资源的不同使用权限。

- 当您需要为指定的用户授权时，例如公司同事、特定部门的用户等，可以在 访问管理（Cloud Access Management，CAM）控制台 为这些用户创建子账号，然后通过存储桶策略、用户策略（CAM 策略）、存储桶 ACL、对象 ACL 等多种授权方式，为子账号授予指定资源的指定访问权限。
- 当您需要为匿名用户授权时，例如允许其他人不经过任何身份认证直接通过 URL 下载对象，您需要将资源权限由默认的私有读修改为公有读。
- 当您需要腾讯云其他服务（例如 CDN 等）使用 COS 存储桶时，也需要遵循同样的授权流程，在您的允许下，这些服务将通过服务角色合法访问 COS，您可以在 CAM 控制台查看已创建的服务角色。

针对跨腾讯云账号授权的情况，若您只需要为一个 COS 存储桶授权，可以直接通过存储桶策略或存储桶 ACL 为另一个主账号授权；若您需要为多个 COS 存储桶或多个腾讯云资源授权，可以通过 访问管理（Cloud Access Management，CAM）控制台 创建协作者身份，进行更大范围的授权。

第四步：为身份设置权限

COS 支持多种权限配置方式，包括 存储桶策略 、用户策略（CAM 策略） 、存储桶 ACL 和 对象 ACL ，您可以根据自己的使用场景选择合适的授权方式。

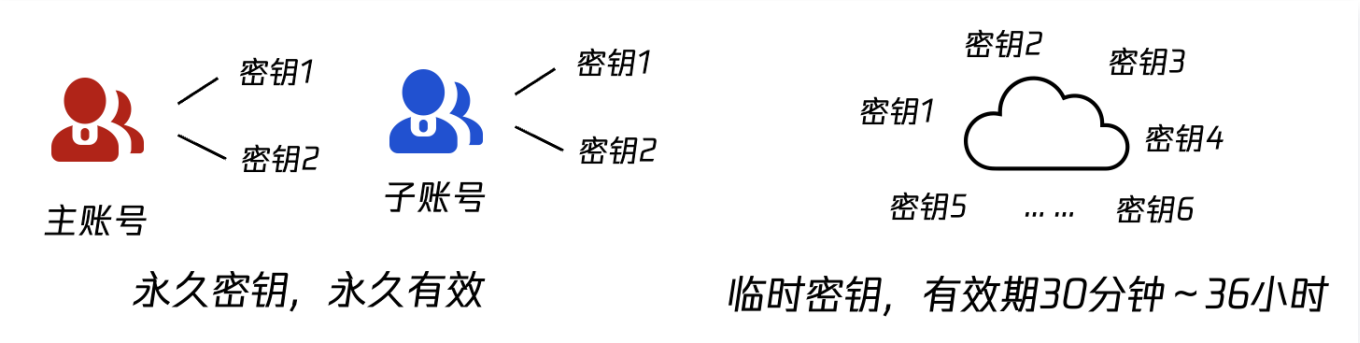
第五步：开始访问（身份认证）

您可以通过控制台、API 请求、SDK 等多种途径访问 COS。出于安全的考虑，存储桶默认为私有读，无论通过哪种途径都需要经过身份认证。对于控制台，使用账号密码即可登录。对于 API 请求和 SDK，用户都需要使用密钥（SecretId/SecretKey）进行身份认证。COS 身份认证方式如下：

默认情况下，COS 存储桶为私有的，无论是通过密钥（永久密钥、临时密钥）访问 COS，还是使用预签名 URL 访问，都要经过身份认证的环节。出于特殊需要，您也可以将存储桶开放为公有读，这是一种有风险的操作，任何用户都可以通过对象 URL 直接下载对象，不需要经过身份认证。

使用永久密钥访问

密钥（SecretId/SecretKey），是用户访问腾讯云 API 进行身份验证时需要用到的安全凭证，可在 API 密钥管理 中获取，每个主账号和子账号都可以创建多个密钥。



永久密钥包括 SecretId 和 SecretKey，每个主账号和子账号都可以生成两对永久密钥，永久密钥代表了账号的永久身份，如果不删除则长期有效。详情可参考 使用永久密钥访问 COS 。

使用临时密钥访问

临时密钥包括 SecretId、SecretKey 和 Token，每个主账号和子账号都可以生成多个临时密钥。相比永久密钥，临时密钥具有有效期（默认1800秒），主账号最长可设定有效期为7200秒，子账号最长可设定有效期为129600秒，详情参考 获取联合身份临时访问凭证 。
临时密钥适用于前端直传等临时授权场景，相比永久密钥，分发临时密钥给不受信任的用户，安全性更高，详情参考 使用临时密钥访问 COS 。

使用临时 URL 访问（预签名 URL）

更多内容和使用说明可参考 使用预签名 URL 访问 COS 。

下载对象

当您希望第三方可以从存储桶下载对象，又不希望对方使用 CAM 账户或临时密钥等方式时，您可以使用预签名 URL 的方式将签名提交给第三方，以供完成临时的下载操作。收到有效预签名 URL 的任何人都可以下载对象。

- 从控制台或 COSBrowser 获取临时下载链接（有效期1 – 2小时）

  您可以直接从控制台或 COSBrowser 快速获取对象的临时下载链接，在浏览器直接输入这个临时链接，即可下载对象。详情参考 快速获取临时链接 文档。
- 使用 SDK 生成预签名 URL

  使用 SDK 可以批量获取自定义有效期的预签名 URL，详情参考 使用 SDK 批量获取预签名 URL 文档。
- 使用签名工具生成预签名 URL

  适合对编程不熟悉的用户，获取自定义有效期的预签名 URL，详情参考 使用签名工具 文档。
- 自行拼接签名链接

  预签名 URL 实际上就是在对象 URL 之后拼接了签名；因此您也可以通过 SDK、签名生成工具等，自行生成签名，将 URL 与签名拼接成签名链接。然而，由于签名生成算法较为复杂，一般情况下不推荐这种使用方式。

上传对象

如果您希望第三方可以上传对象到存储桶，又不希望对方使用 CAM 账户或临时密钥等方式时，您可以使用预签名 URL 的方式将签名提交给第三方，以便完成临时的上传操作。收到有效预签名 URL 的任何人都可以上传对象。

- 途径一：使用 SDK 生成预签名 URL

  各语言 SDK 提供了生成上传预签名 URL 的方法，生成方法可参考 预签名授权上传 ，选择您熟悉的开发语言。
- 途径二：自行拼接签名链接

  预签名 URL 实际上就是在对象 URL 之后拼接了签名。因此，您也可以通过 SDK、签名生成工具 等自行生成签名，将 URL 与签名拼接成签名链接用于对象上传。需注意的是，由于签名生成算法较为复杂，一般情况下不推荐这种使用方式。

匿名访问

默认情况下，COS 存储桶为私有的，无论是通过密钥（永久密钥、临时密钥）访问 COS，还是使用预签名 URL 访问，都要经过身份认证的环节。
出于特殊需要，您也可以将存储桶或对象开放为公有读，任何用户都可以通过对象 URL 直接下载对象，而不经过任何身份认证。

> **注意**
> 将资源开放为公有读具有安全风险，资源链接一旦泄露，任何人均可访问，可能被恶意用户盗刷流量。

将存储桶开放为公有读

您可以在控制台将整个存储桶设置为公有读。这种情况下，存储桶中的每个对象都可以通过对象 URL 被直接下载。配置方式可参考 设置存储桶访问权限 。

将对象开放为公有读

您可以在控制台将单个对象设置为公有读。这种情况下，只有该对象可以直接通过 URL 被下载，其他对象不受影响。配置方式可参考 设置对象的访问权限 。

将文件夹开放为公有读

您可以在控制台将文件夹设置公有读。这种情况下，该文件夹下的所有对象都可以直接通过 URL 被下载，文件夹之外的对象不受影响。配置方式可参考 设置文件夹权限 。

最小权限原则说明

最近更新时间：2024-09-18 19:25:31

概述

在使用对象存储（Cloud Object Storage，COS）时，您可能需要使用临时密钥授予用户相应的资源操作权限，或者为您的子用户或者协作者授予相应的用户策略，允许他们协助您操作 COS 上的资源，或者需要为存储桶添加相应的存储桶策略，允许指定的用户能够在您的存储桶中进行指定的操作，操作指定的资源。在进行这些权限配置的时候，请务必遵循**最小权限原则**，以保障您的数据资产安全。
最小权限原则指的是，当您在授权的时候，请务必明确权限范围，需要明确授予**指定用户**在**何种条件**下，执行**何种操作**，访问**何种资源**的权限。

注意事项

当您在授权的时候，建议严格遵循最小权限，限定用户执行受限的操作（例如授权 `action:GetObject` ），访问受限的资源（例如授权 `resource:examplebucket-1250000000/exampleobject.txt` ）。
为避免授予过大的权限，导致预期外的越权操作，引起数据安全风险。我们强烈建议您尽量避免授予用户访问所有资源（例如 `resource:*` ），或者进行所有操作（例如 `action:*` ）的权限。
潜在的数据安全风险举例如下：

- 数据泄露：授予用户下载指定资源，例如下载 `examplebucket-1250000000/data/config.json` 和 `examplebucket-1250000000/video/` ，但如果在权限策略中配置了 `examplebucket-1250000000/*` ，将导致该存储桶下的所有对象均被允许下载，出现越权操作行为，导致预期外的数据被泄漏。
- 数据被覆盖写：授予用户上传资源权限，例如上传 `examplebucket-1250000000/data/config.json` 和 `examplebucket-1250000000/video/` ，但如果在权限策略中配置了 `examplebucket-1250000000/*` ，将导致该存储桶下的所有对象均被允许上传，出现越权操作行为，导致预期外的对象被覆盖。防范该风险，除了遵循最小权限原则，您可以通过 版本控制 来保留数据的所有历史版本以便追溯。
- 权限泄露：授予用户列出存储桶的对象列表 `cos:GetBucket` ，但如果在权限策略中配置了 `cos:*` ，将导致该存储桶的所有操作均被允许，包括对存储桶再授权、删除对象和删除存储桶，您的数据将存在巨大风险。

使用指引

在最小权限原则下，您需要在策略中明确指定以下信息：

- 委托人（principal）：必须明确指定您需要授予哪些子用户（需要填写用户 ID）、协作者（需要填写用户 ID）、匿名用户或者用户组权限。如果您使用临时密钥进行访问，则不需要指定此项。
- 语句（statement）：在以下几项参数中明确填写相应的参数。
  - 效力（effect）：必须明确声明该策略是“允许”还是“显式拒绝”，包括 allow 和 deny 两种情况。
  - 操作（action）：必须明确声明该策略允许或拒绝的操作。操作可以是单个 API 操作或者多个 API 操作的集合。
  - 资源（resource）：必须明确声明该策略授权的具体资源。资源是用六段式描述。您可以指定资源范围为指定的文件，例如 `exampleobject.jpg` ，或者指定的目录，例如 `examplePrefix/*` 。除非业务需要，否则请不要随意授予用户访问所有资源的权限，即通配符 `*` 。
  - 条件（condition）：描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。

临时密钥最小权限指引

申请临时密钥过程中，您可以通过设置权限策略 Policy 字段，限制操作和资源，将权限限制在指定的范围内。有关临时密钥的生成说明，可参见 临时密钥生成和使用指引 文档。

授权示例

使用 Java SDK 授权访问某个对象的权限

假设需要使用 Java SDK 授权下载存储桶 `examplebucket-1250000000` 中对象 `exampleObject.txt` 的权限，则相应的需要配置代码如下：

```
// 根据 github 提供的 maven 集成方法导入 java sts sdk
import java.util.*;
import com.tencent.cloud.CosStsClient;
import com.tencent.cloud.Response;

public class Demo {
    public static void main(String[] args) {
        TreeMap<String, Object> config = new TreeMap<String, Object>();

        try {
            String secretId = System.getenv("secretId");//用户的 SecretId，建议使用子
账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取可参考
https://cloud.tencent.com/document/product/598/37140
            String secretKey = System.getenv("secretKey");//用户的 SecretKey，建议使
用子账号密钥，授权遵循最小权限指引，降低使用风险。子账号密钥获取可参考
https://cloud.tencent.com/document/product/598/37140
            // 替换为您的 SecretId
            config.put("SecretId", secretId);
            // 替换为您的 SecretKey
            config.put("SecretKey", secretKey);

            // 临时密钥有效时长，单位是秒，默认1800秒，最长可设定有效期为7200秒
            config.put("durationSeconds", 1800);

            // 换成您的 bucket
            config.put("bucket", "examplebucket-1250000000");
            // 换成 bucket 所在地区
            config.put("region", "ap-guangzhou");

            // 这里改成允许的路径前缀，可以根据自己网站的用户登录态判断允许上传的具体路径，例
子：a.jpg 或者 a/* 或者 * 。
            // 如果填写了“*”，将允许用户访问所有资源；除非业务需要，否则请按照最小权限原则授予
用户相应的访问权限范围。
            config.put("allowPrefix", "exampleObject.txt");

            // 密钥的权限列表。简单上传、表单上传和分片上传需要以下的权限，其他权限列表请看
https://cloud.tencent.com/document/product/436/31923
            String[] allowActions = new String[] {
                    // 下载数据
```

```
                    "name/cos:GetObject"
            };
            config.put("allowActions", allowActions);

            Response response = CosStsClient.getCredential(config);
            //成功返回临时密钥信息，如下打印密钥信息
            System.out.println(response.credentials.tmpSecretId);
            System.out.println(response.credentials.tmpSecretKey);
            System.out.println(response.credentials.sessionToken);
        } catch (Exception e) {
            //失败抛出异常
            throw new IllegalArgumentException("no valid secret !");
        }
    }
}
```

使用 API 授权访问某个对象的权限

假设需要使用 API 授权下载存储桶 `examplebucket-1250000000` 中对象 `exampleObject.txt` 和目录为 `examplePrefix` 下所有对象的权限，则相应的需要写入的 Policy 如下：

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "name/cos:GetObject"
      ],
      "effect": "allow",
      "resource": [
        "qcs::cos:ap-beijing:uid/1250000000:examplebucket-
1250000000/exampleObject.txt",
        "qcs::cos:ap-beijing:uid/1250000000:examplebucket-
1250000000/examplePrefix/*"
      ]
    }
  ]
}
```

预签名最小权限指引

您可以通过预签名 URL 的方式实现临时上传和下载操作。此外也可将预签名 URL 提交给任何人，收到有效预签名 URL 的任何人都可以上传或下载对象。

> **注意**

> 临时密钥和永久密钥都可以用于生成预签名 URL，但强烈建议您遵循最小权限指引原则 生成临时密钥，并使用临时密钥计算预签名；尽量不要使用权限过大的永久密钥，以免造成安全风险。

授权示例

授予用户使用预签名 URL 下载对象的权限

使用临时密钥生成一个带签名的下载链接，并设置覆盖要返回的一些公共头部（例如 content-type，content-language），以下为 Java 示例代码：

```
// 传入获取到的临时密钥 (tmpSecretId, tmpSecretKey, sessionToken)
String tmpSecretId = "SECRETID";
String tmpSecretKey = "SECRETKEY";
String sessionToken = "TOKEN";
COSCredentials cred = new BasicSessionCredentials(tmpSecretId, tmpSecretKey,
sessionToken);
// 设置 bucket 的区域, COS 地域的简称请参照
https://cloud.tencent.com/document/product/436/6224
// clientConfig 中包含了设置 region, https(默认 http), 超时, 代理等 set 方法, 使用可参见
源码或者常见问题 Java SDK 部分
Region region = new Region("COS_REGION");
ClientConfig clientConfig = new ClientConfig(region);
// 如果要生成一个使用 https 协议的 URL，则设置此行，推荐设置。
// clientConfig.setHttpProtocol(HttpProtocol.https);
// 生成 cos 客户端
COSClient cosClient = new COSClient(cred, clientConfig);
// 存储桶的命名格式为 BucketName-APPID
String bucketName = "examplebucket-1250000000";
// 此处的key为对象键，对象键是对象在存储桶内的唯一标识
String key = "exampleobject";
GeneratePresignedUrlRequest req =
        new GeneratePresignedUrlRequest(bucketName, key, HttpMethodName.GET);
// 设置下载时返回的 http 头
ResponseHeaderOverrides responseHeaders = new ResponseHeaderOverrides();
String responseContentType = "image/x-icon";
String responseContentLanguage = "zh-CN";
// 设置返回头部里包含文件名信息
String responseContentDispositon = "filename=\"exampleobject\"";
String responseCacheControl = "no-cache";
String cacheExpireStr =
        DateUtils.formatRFC822Date(new Date(System.currentTimeMillis() + 24L *
3600L * 1000L));
responseHeaders.setContentType(responseContentType);
responseHeaders.setContentLanguage(responseContentLanguage);
responseHeaders.setContentDisposition(responseContentDispositon);
responseHeaders.setCacheControl(responseCacheControl);
responseHeaders.setExpires(cacheExpireStr);
req.setResponseHeaders(responseHeaders);
```

```
// 设置签名过期时间(可选)，若未进行设置，则默认使用 ClientConfig 中的签名过期时间(1小时)
// 这里设置签名在半个小时后过期
Date expirationDate = new Date(System.currentTimeMillis() + 30L * 60L * 1000L);
req.setExpiration(expirationDate);
URL url = cosClient.generatePresignedUrl(req);
System.out.println(url.toString());
cosClient.shutdown();
```

用户策略最小权限指引

用户策略是指在 CAM 控制台 添加的用户权限策略，用于授予用户访问 COS 资源的权限。用户访问策略概述的配置说明，可参见 访问策略语言概述 文档。

授权示例

授予账户访问某个对象的权限

假设需要授予账户 UIN 为 `100000000001` ，下载存储桶 `examplebucket-1250000000` 中对象 `exampleObject.txt` 的权限，则相应的访问策略为：

```
{
   "version": "2.0",
   "principal": {
      "qcs": [
         "qcs::cam::uin/100000000001:uin/100000000001"
      ]
   },
    "statement": [
        {
            "action": [
                "name/cos:GetObject"
            ],
            "effect": "allow",
            "resource": [
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000.ap-guangzhou.myqcloud.com/exampleObject.txt"
            ]
        }
    ]
}
```

授予子账户访问某个目录的权限

假设需要授予子账户 UIN 为 `100000000011` （主账户 UIN 为 `100000000001` ），下载存储桶 `examplebucket-1250000000` 中目录为 `examplePrefix` 下的对象的权限，则相应的访问策略为：

```
{
   "version": "2.0",
```

```
    "principal": {
       "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000011"
       ]
    },
     "statement": [
         {
             "action": [
                 "name/cos:GetObject"
             ],
             "effect": "allow",
             "resource": [
                 "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000.ap-
guangzhou.myqcloud.com/examplePrefix/*"
             ]
         }
     ]
}
```

存储桶策略最小权限指引

存储桶策略是指在存储桶中配置的访问策略，允许指定用户对存储桶及桶内的资源进行指定的操作。配置存储桶策略可参见 添加存储桶策略 文档。

授权示例

授予子账号访问特定对象的权限

假设需要授予子账户 UIN 为 `100000000011`（主账户 UIN 为 `100000000001`），下载存储桶 `examplebucket-1250000000` 中的对象 `exampleObject.txt` 和目录 `examplePrefix` 下所有对象的权限，则相应的访问策略为：

```
{
  "Statement": [
    {
      "Action": [
        "name/cos:GetObject"
      ],
      "Effect": "allow",
      "Principal": {
        "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000011"
        ]
      },
      "Resource": [
        "qcs::cos:ap-beijing:uid/1250000000:examplebucket-
1250000000/exampleObject.txt",
```

```
        "qcs::cos:ap-beijing:uid/1250000000:examplebucket-
1250000000/examplePrefix/*"
      ]
    }
  ],
  "version": "2.0"
}
```

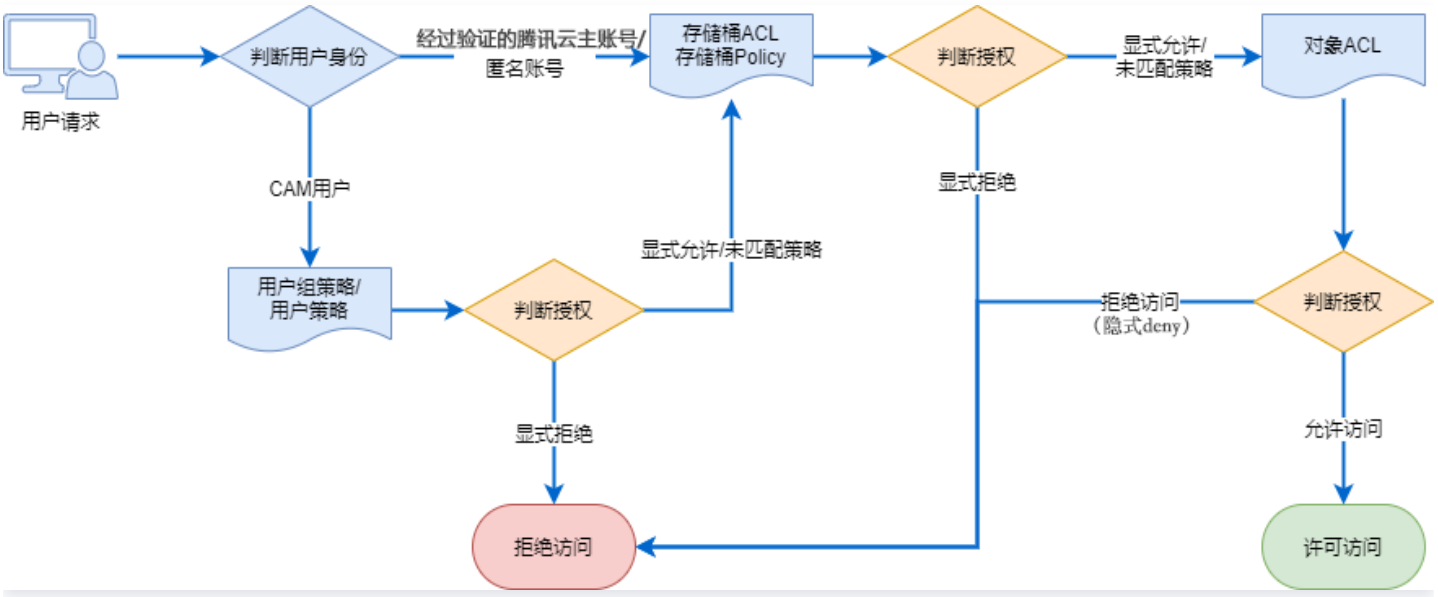
访问策略评估流程

最近更新时间：2024-10-15 09:51:31

访问对象存储（Cloud Object Storage，COS）存储桶及存储桶内资源时需要经过授权后才可进行访问。在腾讯云的权限体系中，资源所属的主账号默认对存储桶及存储桶内的资源拥有所有管理权限。访问管理（Cloud Access Management，CAM）用户（其他主账号、协作者、子账号）以及匿名用户等其他类型用户，需要经过主账号授权后才可进行访问。
账号中的访问策略包括用户组策略、用户策略、存储桶访问控制列表（ACL）和存储桶策略（Policy）等不同的策略类型。在访问策略的评估中，有以下几点关键因素：

1. 用户身份认证：当用户访问 COS 上的资源时，有以下两种情况。
   - 如果携带请求签名，COS 会从请求签名中解析出用户的账号信息，然后将请求转发给访问管理 CAM 进行身份认证。
   - 如果未携带签名请求，则会认为是匿名用户进入到下一环节的认证中。
2. 访问策略的类别：访问策略包括用户组、用户、存储桶等多种类型策略，访问策略的类别决定了访问策略的顺序。
3. 策略上下文信息：在处理用户资源访问请求的时候，会根据用户组策略，用户策略，存储桶策略等多种策略内记录的权限详情进行联合判断，最终决定是否请求通过。

访问策略评估流程



访问策略评估流程

当腾讯云 COS 收到请求时，首先会确认请求者身份，并验证请求者是否拥有相关权限。验证的过程包括检查用户策略、存储桶访问策略和基于资源的访问控制列表，对请求进行鉴权。
腾讯云 COS 收到请求时，将会首先进行身份认证，根据身份认证的结果，将对请求者身份进行分类，不同的身份类别可能会有不同的表现：

1. 经过验证的腾讯云主账号：主账号具备对其所属资源的所有操作权限。而对于非其所属的资源，则需要评估资源权限，若鉴权通过则被许可访问资源。
2. 经过验证的 CAM 用户（子账号或者协作者）：评估用户策略 —— CAM 用户必须具备其父级主账号的授权，才被许可发起相关的访问。如果 CAM 用户需要访问其他主账号所属的资源，则需要评估 CAM 用户所属主账号的资源权限，若鉴权通过则被许可访问资源。

3. 不具备身份特征的匿名用户：评估资源权限 —— 评估在存储桶访问策略或存储桶、对象的访问控制列表的权限，若鉴权通过则被许可访问资源。
4. 上述几类用户以外的请求者：拒绝其访问。

访问策略评估依据

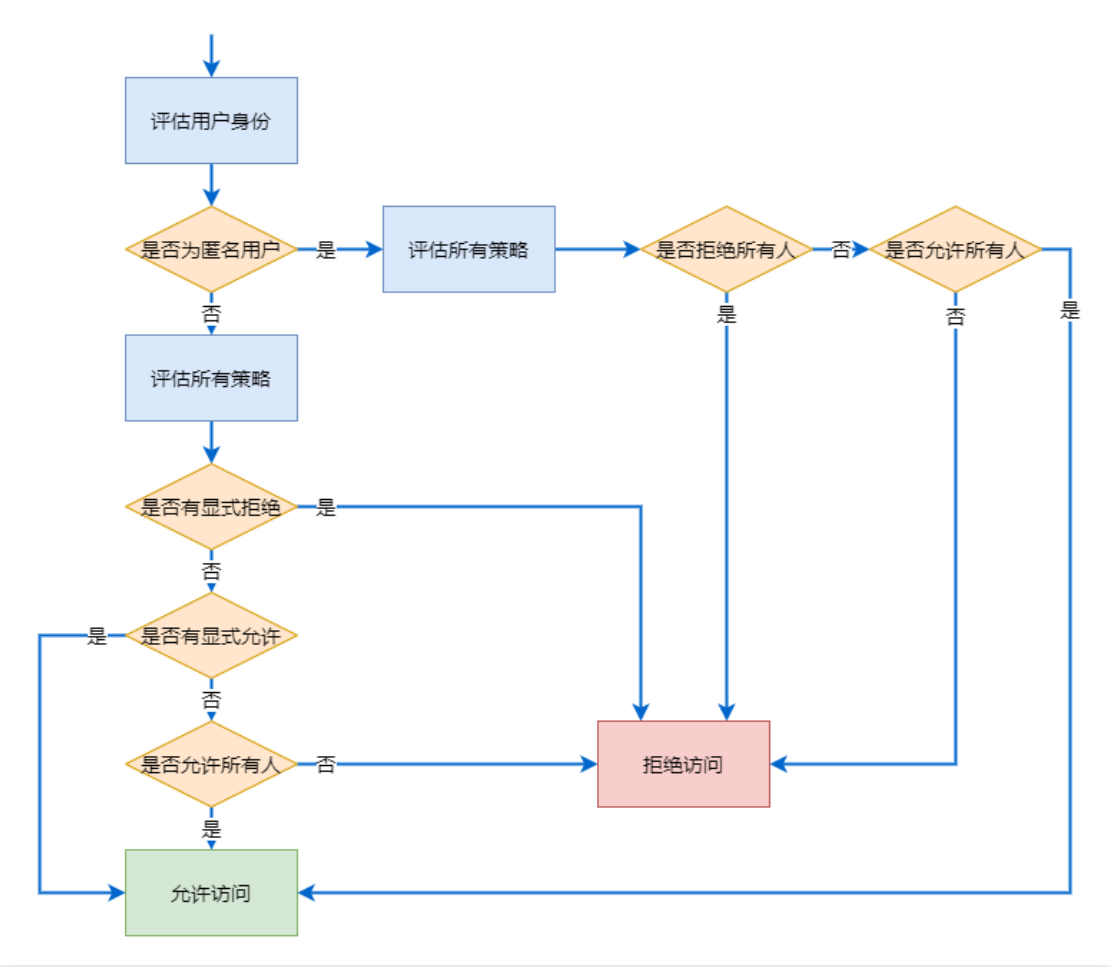
在腾讯云权限体系中，在访问策略评估流程中，全程根据策略上下文信息进行权限评估，同时有以下几项基本原则：

1. 在默认情况下，所有请求都被隐式拒绝（deny）；主账号默认享有账户下所有资源的访问权限。
2. 如果在用户组策略、用户策略、存储桶策略或者存储桶/对象访问控制列表中存在显式允许时，将覆盖此默认值。
3. 任何策略中的显式拒绝将覆盖任何允许。
4. 生效的权限范围为基于身份的策略（用户组策略、用户策略）和基于资源的策略（存储桶策略或者存储桶/对象访问控制列表）的并集。

说明

- 显式拒绝：在用户策略、用户组策略、存储桶 Policy 中针对特定用户有明确的 Deny 策略；如主账号在用户策略中明确配置 Deny 子用户 UIN 100000000011进行 `GET Object` 操作，那该子用户无法下载该主账号下的对象资源。
- 显式允许：在用户策略、用户组策略、存储桶 Policy、存储桶 ACL 中通过 `grant-\*` 明确指定特定用户针对特定用户有明确的 Allow 策略。
- 拒绝所有人：在存储桶 Policy 中明确指定 `Deny anyone`；拒绝所有人后，任意不带签名的请求会被拒绝掉；携带签名的请求会以基于身份的策略进行鉴权。
- 允许所有人：在存储桶 Policy 中明确指定 `Allow anyone`，或者在存储桶 ACL 中明确指定 `public-\*`。
- 生效权限范围为基于身份策略和基于资源策略的并集：在一次完整的鉴权中，COS 将会首先解析用户身份，按照用户身份进行权限校验他有权限访问的资源；同时根据基于资源策略，将用户视为匿名用户进行权限校验；两次校验中有一次校验成功即可允许访问。

访问策略评估依据如下图所示，首先会根据请求中是否携带签名，评估用户是否为匿名用户。如果用户为匿名用户，则会评估策略中是否有拒绝所有人或者允许所有人的策略，并据此判定允许访问或者拒绝访问。如果用户为合法的 CAM 用户或者拥有资源的主账号，则会评估策略中是否有显式拒绝、显式允许或者允许所有人的策略，并据此判定允许访问或者拒绝访问。



策略上下文信息

策略上下文信息指的是策略中记录的权限详情，在 最小权限原则 下，用户需要在策略中明确指定以下信息：

- 委托人（principal）：必须明确指定您需要授予哪些子用户（需要填写用户 ID）、协作者（需要填写用户 ID）、匿名用户或者用户组权限。如果您使用临时密钥进行访问，则不需要指定此项。
- 语句（statement）：在以下几项参数中明确填写相应的参数。
- 效力（effect）：必须明确声明该策略是“允许”还是“显式拒绝”，包括 allow 和 deny 两种情况。
- 操作（action）：必须明确声明该策略允许或拒绝的操作。操作可以是单个 API 操作或者多个 API 操作的集合。
- 资源（resource）：必须明确声明该策略授权的具体资源。资源是用六段式描述。您可以指定资源范围为指定的文件，例如 exampleobject.jpg，或者指定的目录，例如 examplePrefix/*。除非业务需要，否则请不要随意授予用户访问所有资源的权限，即通配符*。
- 条件（condition）：描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。

撰写策略需要按照一定的策略语法撰写，可参见 访问策略语言概述 按照业务所需撰写。用户策略和存储桶策略的编写示例，可分别参见 用户策略语法结构 和 存储桶策略示例 。

访问策略评估示例

假设主账号 UIN 100000000001为子账号 UIN 100000000011关联了一项用户预设策略，允许该子账号只读主账号下的资源；该用户策略详情如下。此项用户策略允许了该子账号执行所有的 `List` 、 `Get` 、 `Head` 操作以及 `OptionsObject` 操作。

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cos:List*",
                "cos:Get*",
                "cos:Head*",
                "cos:OptionsObject"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

同时，主账号在一个私有读写的存储桶 `examplebucket-1250000000` 中添加了以下存储桶策略：

```
{
  "Statement": [
    {
      "Principal": {
        "qcs": [
          "qcs::cam::anyone:anyone"
        ]
      },
      "Effect": "Deny",
      "Action": [
        "name/cos:GetObject"
      ],
      "Resource": [
        "qcs::cos:ap-guangzhou:uid/100000000011:examplebucket-1250000000/*"
      ]
    }
  ],
  "version": "2.0"
}
```

此项存储桶策略显式拒绝了所有用户执行下载对象（ `GetObject` ）的操作。因此，在访问策略评估流程中：

1. 如果该子账号携带签名参数请求 `GetObject` 时，会根据该请求表示的用户身份匹配对应的用户策略，访问策略评估验证通过。
2. 如果该子账号不携带签名参数请求 `GetObject` 时，会被系统判定为匿名请求，会被存储桶策略**拒绝**访问。

权限控制方式介绍
存储桶策略

最近更新时间：2025-05-15 18:11:42

存储桶策略作用于配置的存储桶和存储桶内对象，您可以通过存储桶策略为 CAM 子账号、其他主账号、匿名用户授权存储桶及存储桶内对象的操作权限。

概述

> **注意：**
> 腾讯云主账号具备对其名下资源（包括存储桶）的最大权限，您虽然可以在存储桶策略中限制几乎所有操作，但主账号始终具备 PUT Bucket Policy 操作的权限，主账号调用该操作不检查存储桶策略。

存储桶策略（Bucket Policy）使用 JSON 语言描述，支持向匿名身份或腾讯云任何 CAM 账户授予对存储桶、存储桶操作、对象或对象操作的权限，在腾讯云 COS 中存储桶策略可以用于管理该存储桶内的几乎所有操作，推荐您使用存储桶策略来管理通过 ACL 无法表述的访问策略。

适用场景

> **注意：**
> 创建存储桶和获取存储桶列表这两个服务级的操作权限，需要通过 访问管理控制台 配置。

当您关心谁能访问这个 COS 桶时，推荐使用存储桶策略。可通过查找存储桶，并检查存储桶策略来了解谁能访问。推荐场景有：

- 针对某个存储桶进行授权。
- 相比 ACL，存储桶策略更加灵活。
- 相比用户策略，存储桶策略支持跨账号授权及匿名用户授权。

存储桶策略组成

存储桶策略使用 JSON 语言描述，语法遵循 访问策略语言 的统一规范，包括委托人（principal）、效力（effect）、操作（action）、资源（resource）、条件（condition）等基本元素。详情可参考 访问策略语言概述 。
其中，存储桶策略的资源范围被限制在该存储桶之内，可以针对整个存储桶、指定目录、指定对象授权。

> **注意：**
> 添加存储桶策略时，请务必根据业务需要，按照最小权限原则进行授权。如果您直接授予其他用户所有资源 `(resource:*)` ，或者所有操作 `(action:*)` 权限，则存在由于权限范围过大导致数据安全风险。

控制台配置示例

> **说明：**
> - 使用对象存储控制台配置存储桶策略时，您需要授予用户拥有存储桶的相关权限，例如获取存储桶标签和列出存储桶的权限。

- 存储桶策略的大小限制为20KB。

示例：授予子账号拥有存储桶特定目录的所有权限。配置信息如下：

配置项	配置值
效力	允许
委托人	子账号，子账号的 UIN，该子账号必须为当前主账号下的子账号，例如 100000000011
资源	特定目录前缀，例如 `folder/sub-folder/*`
操作	所有操作
条件	无

控制台支持 图形化配置 和 策略配置 两种方式添加、管理存储桶策略。

图形化配置

进入目标存储桶的**权限管理**，选择 **Policy 权限设置** > **图形设置**，单击**添加策略**，在弹窗中进行策略配置。

步骤1：选择模板（可选）

通过选择不同的被授权用户、资源范围组合，COS 为您提供了多种策略模板，帮助您快速配置存储桶策略。若模板不符合您的需要，可以跳过此步，或在 步骤2：配置策略 中添加或删除授权操作。

被授权用户

- **所有用户（可匿名访问）**：当您希望为匿名用户开放操作权限时，选择此项。在下一步配置策略时会为您自动添加所有用户，表示为*。
  - 当被授权用户指定为**所有用户（可匿名访问）**，请求对象时无需携带签名，所有用户可直接通过链接访问对象，您的数据会存在泄露风险，请谨慎设置。
  - 选择**拒绝所有用户**只能**拒绝匿名用户**，不会拒绝 CAM 用户，如需拒绝**子账号**，请在存储桶策略中选择**指定用户**并指定具体子账号 UIN。
- **指定用户**：当您希望为指定子账户、主账户或云服务开放操作权限时，选择此项。在下一步配置策略中，您需要进一步指定具体的账户 UIN。

资源范围

- 整个存储桶：当您希望授予存储桶配置项相关权限，或者将资源范围指定为整个存储桶，选择此项，在第二步配置策略时会为您自动添加整个存储桶为资源。
- 指定目录：当您希望将资源范围限定到指定文件夹，选择此项。在步骤2：配置策略，您需要进一步指定具体的目录。

模板

您希望授权的操作集合。根据您选择的被授权用户和资源范围，COS 为您提供了推荐的策略模板，若模板不符合您的需要，可以跳过此步，或在下一步“配置策略”中添加或删除授权操作。

- 自定义策略（不提供预设配置）：如您不需要使用模板，可选择此项，在第二步“配置策略”中根据您的需要自行添加策略。
- 其他模板：根据您选择的被授权用户和资源范围的不同组合，COS 为您提供不同的推荐模板。勾选相应的模板后，在第二步配置策略中，COS 会为您自动添加相应的操作。

模板说明请参见下表：

被授权用户	资源范围	策略模板	说明
所有组合		自定义策略	对于任意被授权用户、资源范围组合，选择此模板不提供任何预设策略，您可以直接在第二步配置策略中自行添加策略。
所有用户（可匿名访问）	整个存储桶	只读对象（不含列出对象列表）	对于匿名用户，COS 为您提供读文件（例如下载）、写文件（例如上传、修改）的推荐模板。COS 推荐模板不包括列出您存储桶内的所有对象、和读写权限、存储桶配置等其他敏感权限，避免开放其他多余权限提高数据安全。如
		读写对象（不含列出对象列表）	

	指定目录	只读对象（不含列出对象列表）	您有需要，可以在后续步骤自行添加、删除动作权限。
		读写对象（不含列出对象列表）	
指定用户	整个存储桶	只读对象（不含列出对象列表）	对于指定用户和整个存储桶组合，COS 提供了最多的推荐模板。除了读、写文件和列出文件，COS 还包括以下敏感权限模板，适用于给受信任的用户使用： • 读写存储桶与对象 ACL：获取、修改存储桶 ACL、对象 ACL。包括 GetObjectACL、PutObjectACL、GetBucketACL、PutBucketACL • 存储桶一般配置项：存储桶标签、跨域、回源等非敏感权限。 • 存储桶敏感配置项：涉及存储桶策略、存储桶 ACL、删除存储桶等敏感权限，需要谨慎使用。
		只读对象（含列出对象列表）	
		读写对象（不含列出对象列表）	
		读写对象（含列出对象列表）	
		读写存储桶与对象 ACL	
		存储桶一般配置项	
		存储桶敏感配置项	
	指定目录	只读对象（不含列出对象列表）	对于指定用户和指定目录组合，COS 除了读文件（例如，下载）、写文件（例如上传、修改）之外，还提供了包含列出对象列表的权限的推荐模板。当您需要为指定用户开放指定文件夹的读、写、列出文件的权限时，推荐选择此组合。如您有需要，可以在后续步骤自行添加、删除动作权限。
		只读对象（含列出对象列表）	
		读写对象（不含列出对象列表）	
		读写对象（含列出对象列表）	

步骤2：配置策略

针对您在步骤1选择的被授权用户、指定目录和模板组合，COS 为您在配置策略中自动添加了对应的操作、被授权用户、资源等。其中，当您选择指定用户、指定目录时，需要在配置策略时指定具体的用户 UIN 和目录。

> **说明：**
> 需要注意，对目录授权，输入的资源路径需要在后面加 `/*` 。例如，对目录 test 授权，则需输入 `test/*` 。

当 COS 提供的推荐模板不符合您的需要时，您也可以在这一步对策略内容进行调整，添加、删除被授权用户、资源和操作。如下图所示：

配置项说明如下：

- **效力**：支持选择**允许**或**拒绝**，对应策略语法中的 “allow” 和 “deny” 。
- **用户**：支持添加、删除被授权用户，包括所有用户(`*`)、主账户、子账户和云服务。
- **资源**：支持添加整个存储桶或指定目录资源。
- **操作**：添加、删除您需要授权的操作。
- **条件**：授予权限时指定条件，例如限制用户来访 IP。

> **注意：**
> 当**效力**为**拒绝**时，**所有用户**的生效范围仅针对匿名用户。匿名用户将**无法访问存储桶**，**请谨慎操作**。

策略语法

除了使用图形化配置，熟悉存储桶策略的用户可以直接使用 JSON 语言编写策略。
编写好存储桶策略后，单击**保存**即可。如下图所示。此外您也可以通过 API 或 SDK 添加存储桶策略。

JSON 策略示例

以下策略示例描述为：允许属于主账号 ID 为100000000001（APPID 为1250000000）下的子账号 ID 100000000011，对北京地域的存储桶 examplebucket-bj 下的目录 `folder/sub-folder` 中的对象，授权所有操作。

```
{
  "statement": [
    {
      "principal": {
        "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000011"
        ]
      },
      "effect": "allow",
      "action": [
        "name/cos:*"
      ],
      "resource": [
        "qcs::cos:ap-beijing:uid/1250000000:examplebucket-bj-
1250000000/folder/sub-folder/*"
      ]
    }
  ],
  "version": "2.0"
}
```

操作方式

COS 支持使用控制台、API、SDK 等多种方式添加存储桶策略。控制台支持图形化配置和常用的授权模板，方便不熟悉策略语言的用户快速添加策略。

操作方式		说明
控制台		Web 页面，直观易用
API		RESTful API，直接请求 COS
SDK	Android	丰富的 SDK demo，支持各类开发语言。
	iOS	
	C++	
	.NET(C#)	
	Go	
	Java	
	JavaScript	
	Node.js	
	Python	
	小程序	

更多存储桶策略示例

- 通过存储桶策略（Policy）授权案例
- 授予其他主账号下的子账号操作名下存储桶的权限

> **注意：**
> 添加存储桶策略时，请务必根据业务需要，按照最小权限原则进行授权。如果您直接授予其他用户所有资源 `(resource:*)` ，或者所有操作 `(action:*)` 权限，则存在由于权限范围过大导致数据安全风险。

下面介绍限制子网、委托人和 VPC ID 的存储桶策略示例。

示例一

拒绝**匿名用户**（ `qcs::cam::anyone:anyone` ）来自子网 10.1.1.0/24 网段 和 vpcid 为 aqp5jrc1的请求，语法示例如下。

> **注意：**
> 这个拒绝策略的效力仅针对匿名用户生效，如果需要限制 CAM 用户（子账号、主账号），需要在 principal 字段中明确指定该账号的 Arn。

```
{
  "statement": [
    {
```

```
      "action": [
        "name/cos:*"
      ],
      "condition": {
        "ip_equal": {
          "qcs:ip": [
            "10.1.1.0/24"
          ]
        },
        "string_equal": {
          "vpc:requester_vpc": [
            "vpc-aqp5jrc1"
          ]
        }
      },
      "effect": "deny",
      "principal": {
        "qcs": [
          "qcs::cam::anyone:anyone"
        ]
      },
      "resource": [
        "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
      ]
    }
  ],
  "version": "2.0"
}
```

示例二

允许子账号（`qcs::cam::uin/100000000001:uin/100000000002`）在访问来源 vpcid 为 aqp5jrc1 时访问指定存储桶，语法示例如下：

```
{
  "statement": [
    {
      "action": [
        "name/cos:*"
      ],
      "condition": {
        "string_equal": {
          "vpc:requester_vpc": [
            "vpc-aqp5jrc1"
          ]
        }
      },
      "effect": "allow",
```

```
      "principal": {
        "qcs": [
          "qcs::cam::uin/100000000001:uin/100000000002"
        ]
      },
      "resource": [
        "qcs::cos:ap-beijing:uid/1250000000:examplebucket-1250000000/*"
      ]
    }
  ],
  "version": "2.0"
}
```

用户策略

最近更新时间：2025-05-15 18:11:42

腾讯云主账户可以在 访问管理（Cloud Access Management，CAM） 控制台创建 CAM 用户，并关联策略，授予 CAM 用户使用腾讯云资源的权限。

概述

用户可以在 CAM 中，对于主账号名下的不同类型用户，授予不同的权限。这些权限通过访问策略语言描述，并以用户为出发点进行授权，因此被称为**用户策略**。

用户策略与存储桶策略的区别

用户策略与存储桶策略的最大差别是：用户策略只描述效力（Effect）、操作（Action）、资源（Resource）和条件（Condition，可选），不描述身份（Principal）。因此，用户策略的使用方式为：

- 用户策略需要撰写完成后，再对子用户、用户组或角色执行关联操作。
- 用户策略**不支持将操作和资源权限授予匿名用户**。

预设策略和自定义策略

用户策略包括两类，预设策略和自定义策略，您可以使用 预设策略进行关联授权，也可以 自行撰写用户策略 再进行关联授权，详见访问管理的 授权指南 。

适用场景

当您关心用户能做什么，推荐用户策略时，可通过查找 CAM 用户，并检查其所属用户组的权限来了解用户能做什么。推荐场景有：

- 要配置对象存储（Cloud Object Storage，COS）服务级权限时，例如创建桶（PutBucket）、列举桶（GetService）。
- 需要使用主账号下所有 COS 桶和对象。
- 要对主账号下的大量 CAM 用户授予相同权限。

用户策略语法

策略语法

和存储桶策略一样，用户策略使用 JSON 语言描述，遵循 访问策略语言 的统一规范（委托人、效力、操作、资源、条件等）。但由于用户策略是直接关联到用户/用户组上的，因此用户策略不需要填写委托人（Principal）。
下表是用户策略和存储桶策略的区别对比：

元素	用户策略	存储桶策略
委托人	**不填**	必填
效力	必填	必填
操作	必填	必填
资源	必填	**该存储桶内资源**

条件	选填	选填

策略示例

以下是一个典型的用户策略示例，策略含义为：授权位于广州的存储桶 examplebucket-1250000000所有 COS 操作的策略。您需要将策略保存后再关联到 CAM 子用户、用户组或角色方可生效：

```
{
  "statement": [
    {
      "effect": "allow",
      "action": ["cos:*"],
      "resource": [
          "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
      ]
    }
  ],
  "version": "2.0"
}
```

通过用户策略授权子账号访问 COS

前提条件

已创建 CAM 子账号，创建方法可参考 创建子账号 。

配置步骤

CAM 提供了 预设策略和自定义策略 。预设策略为 CAM 提供的系统预设策略，COS 相关策略见 预设策略 ；自定义策略支持用户自定义资源、操作等元素，更加灵活。下面说明如何新建一个自定义策略，为子账户授权：

1. 登录 CAM 控制台 。
2. 选择**策略 > 新建自定义策略 > 按策略语法创建**，进入策略创建页面。
3. 您可按照实际需求选择**空白模板**自定义授权策略，或选择与 COS 相关联的**系统模板**。这里以选择**空白模板**为例。
4. 选择**空白模板**，单击**下一步**，输入您的策略语法。需要包括以下基本元素：
   - **resource：授权资源。**
     - 所有资源（ `"*"` ）
     - 指定存储桶(`"qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"`)
     - 存储桶内指定目录或对象(`"qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/test/*"`)
   - **action：授权操作。**
   - **effect：效力。**选择 `"allow"` (允许)或 `"deny"` （拒绝）。
   - **condition：生效条件。**可选项。

   COS 提供了用户策略示例，您也可以参考以下文档，直接将策略内容复制粘贴到**策略内容**编辑框内，确认输入无误后单击**完成**即可。
   - 授权子账号访问 COS

- COS API 授权策略使用指引

5. 创建完成后，您可在 CAM 控制台 的**策略** > **自定义策略**中查看已创建的自定义策略，并将策略关联到子账号。
6. 勾选子账号并单击**确定**授权后，即可使用子账号访问所限定的 COS 资源。

预设策略

1. CAM 提供了一些预设策略，您可以在 CAM 控制台 的**策略** > **预设策略**中查看，搜索“COS”筛选。
2. 单击策略名，进入**策略语法** > **JSON** 查看具体的策略内容。预设策略的资源（ `resource` ）被设置为 COS 所有资源(`"*"`)，且不支持修改。若您需要对部分 COS 存储桶、对象授权，可以复制 JSON 的预设策略，创建 自定义策略 。

表1和表2列出了 CAM 提供的 COS 相关的预设策略及说明。

表1：COS 预设策略

预设策略	说明	JSON策略
QcloudCOSBucketConfigRead	拥有该权限的用户可以读取 COS 存储桶配置	`{ "version": "2.0", "statement": [ { "effect": "allow", "action": [ "cos:GetBucket*", "cos:HeadBucket" ], "resource": "*" } ]`

		`}`
QcloudCOSBucketConfigWrite	拥有该权限的用户可以修改 COS 存储桶配置	`{ "version": "2.0", "statement": [ { "effect": "allow", "action": [ "cos:PutBucket*" ], "resource": "*" } ] }`
QcloudCOSDataFullControl	包含 COS 存储桶内数据读、写、删除、列出的访问权限	`{ "version": "2.0", "statement": [{ "effect": "allow", "action": ["cos:GetService", "cos:GetBucket", "cos:ListMultipartUploads", "cos:GetObject*", "cos:HeadObject", "cos:GetBucketObjectVersions", "cos:OptionsObject", "cos:ListParts", "cos:DeleteObject", "cos:PostObject", "cos:PostObjectRestore", "cos:PutObject*", "cos:InitiateMultipartUpload", "cos:UploadPart", "cos:UploadPartCopy", "cos:CompleteMultipartUpload", "cos:AbortMultipartUpload", "cos:DeleteMultipleObjects"], "resource": "*" }]`

```
}
```

表2：COS 操作与预设策略的关系

说明	操作	QcloudCOSBucketConfigRead	QcloudCOSBucketConfigWrite	QcloudCOSDataFullControl	QcloudCOSDataReadOnly	QcloudCOSDataWriteOnly	QcloudCOSFullAccess	QcloudCOSGetServiceAccess	QcloudCOSListOnly	QcloudCOSReadOnlyAccess
列举桶	GetService	✗	✗	✓	✗	✗	✓	✓	✓	✓
创建桶	PutBucket	✗	✓	✗	✗	✗	✓	✗	✗	✗
删除桶	DeleteBucket	✗	✗	✗	✗	✗	✓	✗	✗	✗
获取桶基本信息	HeadBucket	✓	✗	✗	✗	✗	✓	✗	✓	✓
获取桶的配置项	GetBucket*	✓	✗	✗	✗	✗	✓	✗	✗	✓
修改桶的配置项	PutBucket*	✗	✓	✗	✗	✗	✓	✗	✗	✗

获取桶的访问权限	GetBucketAcl	✔	✘	✘	✘	✘	✔	✘	✘	✔
修改桶的访问权限	PutBucketAcl	✘	✔	✘	✘	✘	✔	✘	✘	✘
列举桶内对象	GetBucket	✔	✘	✔	✘	✘	✔	✘	✔	✔
列举桶内对象的所有版本	GetBucketObjectVersions	✔	✘	✔	✘	✘	✔	✘	✔	✔
上传对象	PutObject	✘	✘	✔	✘	✔	✔	✘	✘	✘
分块上传	ListParts InitiateMultipartUpload UploadPart UploadPartCopy CompleteMultipartU	✘	✘	✔	✘	✔	✔	✘	✘	✘

	pload AbortMultipartUpload ListMultipartUploads									
下载对象	GetObject	✕	✕	✓	✓	✕	✓	✕	✕	✕
查看对象元数据	HeadObject	✕	✕	✓	✓	✕	✓	✕	✕	✓
跨域（CORS）预检	OptionsObject	✕	✕	✓	✓	✕	✓	✕	✕	✓

更多用户策略示例

- 授权子账号访问 COS
- COS API 授权策略使用指引

ACL

最近更新时间：2024-10-14 16:45:21

基本概念

访问控制列表（ACL）使用 XML 语言描述，是与资源关联的一个指定被授权者和授予权限的列表，每个存储桶和对象都有与之关联的 ACL，支持向匿名用户或其他腾讯云的主账号授予基本的读写权限。

> **注意**
> 使用与资源关联的 ACL 管理有一些限制：
> - 资源的拥有者始终对资源具备 FULL_CONTROL 权限，无法撤销或修改。
> - 匿名用户无法成为资源拥有者，此时对象资源的拥有者属于存储桶的创建者（腾讯云主账号）。
> - 仅可对腾讯云访问管理（Cloud Access Management，CAM）主账号或预设用户组授予权限，无法授予自定义用户组权限，不推荐授予子用户权限。
> - 不支持对权限附加条件。
> - 不支持显示拒绝的权限。
> - 一个资源最多可以拥有100条 ACL 策略。

适用场景

> **注意**
> 开放匿名用户访问（公有读）属于高危操作，有流量盗刷的风险；必须使用公有读时，可 设置防盗链 进行安全防护。

当您仅需要为存储桶和对象设置一些简单的访问权限或开放匿名访问，可以选择 ACL。但在更多的情况下，推荐您优先使用存储桶策略或用户策略，灵活程度更高。ACL 的适用场景包括：

- 仅设置简单的访问权限。
- 在控制台快速设置访问权限。
- 需要将某个对象、目录或存储桶开放给所有互联网匿名用户访问，ACL 操作更为便捷。

ACL 的元素

身份 Grantee

支持的被授权身份可以是某个 CAM 主账号，或者是某个预设的 CAM 用户组。

> **注意**
> - 当您授予了其他腾讯云主账号访问权限时，这个被授权的主账号可以授权其名下的子用户、用户组或角色的访问权限。
> - 对象存储（Cloud Object Storage，COS）完全不建议您对匿名用户或 CAM 用户组授予 WRITE、WRITE_ACP 或 FULL_CONTROL 权限。一旦授权许可后，用户组可以对您的资源进行上传、下载、删除等行为，这将会给您带来数据丢失、扣费等风险。

在存储桶或对象的 ACL 中支持授予的身份包括：

- 跨账号：请使用主账号的 ID，通过**账号中心**的 账号信息 获得账号 ID，例如100000000001。
- 预设用户组：请使用 URI 标签标记预设的用户组，支持的用户组包括：
  - 匿名用户组 – `http://cam.qcloud.com/groups/global/AllUsers` 该组代表了任何人都可以无需授权而访问资源，无论请求已签名或者未签名。
  - 认证用户组 – `http://cam.qcloud.com/groups/global/AuthenticatedUsers` 该组代表所有经过腾讯云 CAM 账户认证的用户都可以访问资源。

操作 Permission

腾讯云 COS 在资源 ACL 上支持的操作实际上是一系列的操作集合，对于存储桶和对象 ACL 来说分别代表不同的含义。

存储桶的操作

下表列出了支持在存储桶 ACL 中设置的操作列表：

操作集	描述	许可的行为
READ	列出对象	HeadBucket、GetBucketObjectVersions、ListMultipartUploads
WRITE	上传、覆盖和删除对象	PutObject、PutObjectCopy、PostObject、InitiateMultipartUpload、UploadPart、UploadPartCopy、CompleteMultipartUpload、DeleteObject
READ_ACP	读取存储桶的 ACL	GetBucketACL
WRITE_ACP	写入存储桶的 ACL	PutBucketACL
FULL_CONTROL	以上四种权限的集合	以上所有行为的集合

> **注意**
> 请谨慎授予存储桶 WRITE、WRITE_ACP 或 FULL_CONTROL 权限。授予存储桶 WRITE 权限将允许被授权者覆盖或删除已有的任何对象。

对象的操作

下表列出了支持在对象 ACL 中设置的操作列表：

操作集	描述	许可的行为
READ	读取对象	GetObject、GetObjectVersion、HeadObject
READ_ACP	读取对象的 ACL	GetObjectACL、GetObjectVersionACL
WRITE_ACP	写入对象的 ACL	PutObjectACL、PutObjectVersionACL
FULL_CONTROL	以上三种权限的集合	以上所有行为的集合

版权所有：腾讯云计算（北京）有限责任公司

> **说明**
> 对象不支持授予 WRITE 操作集。

预设的 ACL

COS 支持一系列预设的 ACL 进行授权，方便简单权限的描述。使用预设 ACL 描述时，需要在 PUT Bucket/Object 或 PUT Bucket/Object acl 中携带 x-cos-acl 头部并描述所需权限，如果同时在请求正文中携带了 XML 的描述内容，我们将优先选择头部中的描述并忽略请求正文中的 XML 描述。

存储桶的预设 ACL

预设名称	描述
private	创建者（主账号）具备 FULL_CONTROL 权限，其他人没有权限（默认）
public-read	创建者具备 FULL_CONTROL 权限，匿名用户组具备 READ 权限
public-read-write	创建者和匿名用户组都具备 FULL_CONTROL 权限，通常不建议授予此权限
authenticated-read	创建者具备 FULL_CONTROL 权限，认证用户组具备 READ 权限

对象的预设 ACL

预设名称	描述
default	空描述，此时根据各级目录的显式设置及存储桶的设置来确定是否允许请求（默认）
private	创建者（主账号）具备 FULL_CONTROL 权限，其他人没有权限
public-read	创建者具备 FULL_CONTROL 权限，匿名用户组具备 READ 权限
authenticated-read	创建者具备 FULL_CONTROL 权限，认证用户组具备 READ 权限
bucket-owner-read	创建者具备 FULL_CONTROL 权限，存储桶拥有者具备 READ 权限
bucket-owner-full-control	创建者和存储桶拥有者都具备 FULL_CONTROL 权限

> **说明**
> 对象不支持授予 public-read-write 权限。

示例

存储桶的 ACL

在创建存储桶时，COS 将创建一个默认的 ACL 以赋予资源拥有者对资源的完全控制权限（FULL_CONTROL），示例如下：

```
<AccessControlPolicy>
  <Owner>
```

```
    <ID>Owner-Cononical-CAM-User-Id</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee>
        <ID>Owner-Cononical-CAM-User-Id</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

对象的 ACL

在创建对象时，COS 默认不会创建 ACL，此时对象的拥有者为存储桶拥有者。对象继承存储桶的权限，与存储桶的访问权限一致。由于对象没有默认的 ACL，其将遵循存储桶策略（Bucket Policy）中对访问者和其行为的定义，来判断请求是否被许可。详情请参见 访问策略语言概述 文档。
如果您需要对对象授予其他访问权限，您可以在此基础上添加更多的 ACL 来描述对象的访问权限。例如授予匿名用户只读单个对象的权限，示例如下：

```
<AccessControlPolicy>
  <Owner>
    <ID>Owner-Cononical-CAM-User-Id</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee>
        <ID>Owner-Cononical-CAM-User-Id</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee>
        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

基于标签管理项目资源

最近更新时间：2023-04-10 09:42:05

简介

项目管理是基于项目维度对资源进行集中管理。您可以将支持项目功能的云产品资源添加至项目，并通过访问管理（Cloud Access Management，CAM）控制台生成项目策略，然后将项目策略关联至项目相关用户或用户组，以允许用户或用户组拥有项目资源的操作权限。
对象存储（Cloud Object Storage，COS）在历史版本控制台上基于项目为用户提供了相关权限管理操作，然而项目策略包含对添加至项目的所有产品下所有资源的完全访问权限，不仅**无法满足多维度的标记和归类需求**，也**无法进行精细化的权限管理**。
COS 利用标签服务对原项目功能进行了兼容。在标签服务的体系下，项目是特殊的标签，它的标签键为 `project` 。
对于在历史版本控制台上使用过项目的部分老用户，您依然可以在项目控制台，新建项目并在项目下创建存储桶，COS 会在您创建存储桶时将存储桶的项目归属关系自动双写一份到标签中，以便进行控制台的展示。

> **说明**
> - COS 新版本（V5）控制台仅支持基于标签的方式进行项目资源的权限管理。
> - 如果您有分类管理存储桶的需求，我们推荐您直接通过标签管理您的存储桶，实现权限控制和分账等任务，而不是通过项目的途径进行管理。有关如何在控制台上增加标签，请参见 设置存储桶标签 。
> - 了解项目策略说明和操作指引，请参见 CAM 的 项目与标签 。如需了解标签服务，请参见 标签 产品文档。

授予子账号拥有项目的访问权限

> **注意**
> 以下操作步骤仅适用于在历史版本控制台上使用过项目的部分老用户，通过项目鉴权方式授予子账号访问权限。

授予子账号拥有项目的访问权限，您可以按照以下步骤进行操作：

1. 登录 项目管理控制台 ，新建一个项目，自定义项目名称并提交，然后选择在该项目下创建一个存储桶或者云服务器等资源。如果您已有项目，且项目已有归属的存储或计算资源，可跳过这一步。
2. 创建完项目并绑定好相应资源后，进入 策略管理 页面，单击**新建自定义策略 > 按标签授权**，选择添加对象存储服务和授权的操作权限，选择相应的项目标签，授予子账号访问该项目标签下的所有资源。
3. 单击**下一步**，选择将此权限授权给用户/用户组/角色，最后单击**完成**即可。

> **说明**
> - 默认策略内容为授予子账号访问该项目标签下的所有资源。如果您只希望用户仅能对标签下的部分资源进行指定操作，您可以参见 语法结构 文档，修改策略语法中的 `action` （设置指定操作）和 `resource` （设定可操作资源）。
> - 如果您希望子账号能够创建存储桶，则还需要为子账号授予 `PUT Bucket` 的操作权限。您可以在 策略管理 页面单击**新建自定义策略 > 按策略生成器创建**，授权子账号相应的权限。

如何选择授权方式

最近更新时间：2024-03-27 11:21:41

存储桶策略、用户策略、存储桶 ACL 和对象 ACL 的差异点

用户策略、存储桶策略、存储桶 ACL 和对象 ACL 之间的差异点，如表1所示。

- 用户策略是基于用户的授权方式，存储桶策略、存储桶 ACL 和对象 ACL 是基于资源的授权方式。
- 用户策略和存储桶策略的授权均基于访问策略语言，权限控制的灵活程度更高，授权可以具体到每一个操作（action）且授权效果（effect）可以是 deny 或 allow；存储桶 ACL 和对象 ACL 均基于访问控制列表（ACL）进行授权，权限控制的灵活程度相对较低，使用更加简单，但只支持授予基本的读写权限。
- 用户策略不支持向匿名用户授权，存储桶策略、存储桶 ACL、对象 ACL 支持匿名用户授权。
- 用户策略在访问管理（Cloud Access Management，CAM）控制台配置，存储桶策略、存储桶 ACL、对象 ACL 在对象存储（Cloud Object Storage，COS）控制台配置。

表1：不同授权方式的差异点对比

<table>
<tr><th colspan="2">对比项</th><th>用户策略</th><th>存储桶策略</th><th>存储桶 ACL</th><th>对象 ACL</th></tr>
<tr><td colspan="2">分类</td><td>基于用户的授权</td><td>基于资源的授权</td><td>基于资源的授权</td><td>基于资源的授权</td></tr>
<tr><td colspan="2">权限控制灵活程度</td><td>灵活度高</td><td>灵活度高</td><td>灵活度低</td><td>灵活度低</td></tr>
<tr><td colspan="2">控制台配置</td><td>CAM 控制台</td><td>COS 控制台</td><td>COS 控制台</td><td>COS 控制台</td></tr>
<tr><td rowspan="6">访问控制元素</td><td rowspan="3">用户</td><td rowspan="2">本账号管理的所有 CAM 身份（子账号、角色等）</td><td>子账号、主账号、匿名用户</td><td rowspan="2">子账号、其他主账号、匿名用户</td><td rowspan="2">子账号、其他主账号、匿名用户</td></tr>
<tr><td>子账号腾讯云服务、角色等</td></tr>
<tr><td>跨账号授权需要先添加为协作者</td><td colspan="3">直接支持跨账号授权</td></tr>
<tr><td>效果</td><td>允许+拒绝</td><td>允许+拒绝</td><td>仅允许</td><td>仅允许</td></tr>
<tr><td>资源</td><td>所有云资源，COS 所有存储桶、指定存储桶（前缀、对象等）</td><td>指定存储桶（前缀、对象等）</td><td>整个存储桶</td><td>指定对象</td></tr>
<tr><td>动作</td><td>每个具体动作</td><td>每个具体动作（不包括创建桶、列出桶）</td><td>简化的读写权限</td><td>简化的读写权限</td></tr>
<tr><td>条件</td><td>支持</td><td>支持</td><td>不支持</td><td>不支持</td></tr>
</table>

如何选择合适的授权方式?

通过表1列出的差异点，可以明显看出不同授权方式的优劣，您可以根据自己的需要选择合适的授权方式。
但无论选择哪种方式，建议尽可能保持统一。随着存储桶策略、用户策略和 ACL 的增加，权限维护难度将越来越大。

何时选择 ACL?

> **注意**
> 开放匿名用户访问（公有读）属于高危操作，有流量盗刷的风险；必须使用公有读时，可 设置防盗链 进行安全防护。

当您仅需要为存储桶和对象设置一些简单的访问权限或开放匿名访问时，可以选择 ACL。但在更多的情况下，推荐您优先使用存储桶策略或用户策略，灵活程度更高。

- 仅设置简单的访问权限。
- 在控制台快速设置访问权限。
- 需要将某个对象、目录或存储桶开放给所有互联网匿名用户访问，ACL 操作更为便捷。
- 每个账户下仅支持设置1000条 ACL，超过此上限请选择存储桶策略或用户策略替代。

何时选择用户策略?

当您关心用户能做什么时，推荐用户策略。可通过查找 CAM 用户，并检查其所属用户组的权限来了解用户能做什么。推荐场景有：

- 要配置 COS 服务级权限时，如创建桶、列举桶。
- 需要使用主账号下所有 COS 桶和对象。
- 要对主账号下的大量 CAM 用户授予相同权限。

何时选择存储桶策略?

当您关心谁能访问这个 COS 桶时，推荐使用桶策略。可通过查找桶，并检查桶策略来了解谁能访问。推荐场景有：

- 针对某个存储桶进行单独授权。
- 相比 ACL，授权具体到某个操作（action），授权效果（effect）需要为 deny 或 allow。
- 相比用户策略，存储桶策略支持跨账号授权及匿名用户授权。

访问策略语言
访问策略语言概述

最近更新时间：2024-10-08 18:54:41

> **注意：**
> 在给子用户或者协作者添加访问策略时，请务必根据业务需要，按照最小权限原则进行授权。如果您直接授予子用户或者协作者所有资源 `(resource:*)` ，或者所有操作 `(action:*)` 权限，则存在由于权限范围过大导致数据安全风险。

概述

访问策略可用于授予访问对象存储（Cloud Object Storage，COS）资源的权限。访问策略使用基于 JSON 的访问策略语言。您可以通过访问策略语言授权指定委托人（principal）对指定的 COS 资源执行指定的操作。
访问策略语言描述了存储桶策略（Bucket Policy）的基本元素和用法，有关策略语言的说明请参见 CAM 元素参考 。

访问策略中的元素

访问策略语言包含以下基本意义的元素：

- **委托人（principal）**：描述策略授权的实体。例如用户（主账号、子账号、匿名用户）、用户组等。该元素对于存储桶访问策略有效，对用户访问策略则不应添加。
- **语句（statement）**：描述一条或多条权限的详细信息。该元素包括效力、操作、资源、条件等多个其他元素的权限或权限集合。一条策略有且仅有一个语句元素。
- **效力（effect）**：描述声明产生的结果是“允许”还是“显式拒绝”，包括 allow 和 deny 两种情况。该元素是必填项。
- **操作（action）**：描述允许或拒绝的操作，操作一般为 API（以 name 前缀描述）。该元素是必填项。
- **资源（resource）**：描述授权的具体数据。资源是用六段式描述。每款产品的资源定义详情会有所区别。有关如何指定资源的信息，请参阅您编写的资源声明所对应的产品文档。该元素是必填项。
- **条件（condition）**：描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。有些服务允许您在条件中指定其他值。该元素是非必填项。

元素用法

指定委托人

委托人 principal 元素用于指定被允许或拒绝访问资源的用户、账户、服务或其他实体。元素 principal 仅在存储桶中起作用；用户策略中不必指定，因为用户策略直接附加到特定用户。下面是指定 principal 的示例。

```
"principal": {
"qcs": [
"qcs::cam::uin/100000000001:uin/100000000001"
]
}
```

授予匿名用户权限：

```
"principal": {
"qcs": [
"qcs::cam::anonymous:anonymous"
]
}
```

授权主账户 UIN 100000000001 权限：

```
"principal": {
"qcs": [
"qcs::cam::uin/100000000001:uin/100000000001"
]
}
```

授权子账户 UIN 100000000011（主账户 UIN 为 100000000001）权限：

> **注意：**
> 操作前需确保子账号已被添加到主账号的子账号列表中。

```
"principal": {
"qcs": [
"qcs::cam::uin/100000000001:uin/100000000011"
]
}
```

指定效力

如果没有显式授予（允许）对资源的访问权限，则隐式拒绝访问。您也可显式拒绝（deny）对资源的访问，这样可确保用户无法访问该资源，即使有其他策略授予了访问权限的情况下也是如此。下面是指定允许效力的示例。

```
"effect" : "allow"
```

指定操作

COS 定义了可在策略中指定某一个特定的 COS 操作，指定的操作与发起的 API 请求操作完全一致，以下列举了部分存储桶操作和对象操作，更多具体操作请参见 API 操作列表 文档。

存储桶操作

描述	对应的 API 接口
name/cos:GetService	GET Service
name/cos:GetBucket	GET Bucket (List Objects)

name/cos:PutBucket	PUT Bucket
name/cos:DeleteBucket	DELETE Bucket

对象操作

描述	对应的 API 接口
name/cos:GetObject	GET Object
name/cos:PutObject	PUT Object
name/cos:HeadObject	HEAD Object
name/cos:DeleteObject	DELETE Object

指定允许操作的示例如下：

```
"action": [
"name/cos:GetObject",
"name/cos:HeadObject"
]
```

指定资源

资源（resource）元素描述一个或多个操作对象，如 COS 存储桶或对象等。所有资源均可采用下述的六段式描述方式。

```
qcs:project_id:service_type:region:account:resource
```

参数说明如下：

参数	描述	是否必选
qcs	是 qcloud service 的简称，表示是腾讯云的云服务。	是
project_id	描述项目信息，仅为了兼容 CAM 早期逻辑。	否
service_type	描述产品简称，如 COS。	是
region	描述地域信息，可参阅腾讯云 COS 支持的 可用地域 。	否
account	描述资源拥有者的主账号信息。目前支持两种方式描述的资源拥有者。一种方式是 uin 方式，即主账号的 UIN 账号，表示为 `uin/${OwnerUin}` ，如 uin/100000000001。另外一种方式是 uid 方式，即主账号的 APPID，表示为 `uid/${appid}` ，如 uid/1250000000。目前 COS 的资源拥有者统一使用 uid 的方式表述，即主账号的开发商 APPID。	否

resource	描述具体资源详情，在 COS 服务中使用存储桶 XML API 访问域名来描述。	是

下面是指定存储桶 examplebucket-1250000000 的示例。

```
"resource": ["qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"]
```

下面是指定存储桶 examplebucket-1250000000 中的 /folder/ 文件夹下所有对象的示例。

```
"resource": ["qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-
1250000000/folder/*"]
```

下面是指定存储桶 examplebucket-1250000000 中的 /folder/exampleobject 对象的示例。

```
"resource": ["qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-
1250000000/folder/exampleobject"]
```

下面是指定存储桶 examplebucket-1250000000 中所有以 .txt 为后缀对象的示例。

```
"resource": ["qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-
1250000000/*.txt"]
```

指定条件

访问策略语言可使您在授予权限时指定条件。例如限制用户访问来源，限制授权时间等。下面列出了目前支持的条件操作符列表以及通用的条件键和示例等信息。

条件操作符	含义	条件名	示例
ip_equal	IP 等于	qcs:ip	{"ip_equal":{"qcs:ip ":"10.121.2.0/24"}}
ip_not_equal	IP 不等于	qcs:ip	{"ip_not_equal":{"qcs:ip ":["10.121.1.0/24", "10.121.2.0/24"]}}

下面是满足来访 IP 为 10.121.2.0/24 网段内的示例。

```
"ip_equal":{"qcs:ip ":"10.121.2.0/24"}
```

下面是满足来访 IP 为 101.226.100.185 和 101.226.100.186 的示例。

```
"ip_equal": {
    "qcs: ip": [
        "101.226.100.185",
        "101.226.100.186"
    ]
```

```
}
```

实际案例

当主账号允许匿名用户，在访问来源 IP 为 101.226.100.185 或 101.226.100.186 时，对广州存储桶 examplebucket-1250000000 中的对象，执行 GET（下载）和 HEAD 操作，而无需鉴权。更多案例请参阅 权限设置相关案例 。

```
{
    "version": "2.0",
    "principal": {
        "qcs": [
            "qcs::cam::anonymous:anonymous"
        ]
    },
    "statement": [
        {
            "action": [
                "name/cos:GetObject",
                "name/cos:HeadObject"
            ],
            "condition": {
                "ip_equal": {
                    "qcs: ip": [
                        "101.226.100.185",
                        "101.226.100.186"
                    ]
                }
            },
            "effect": "allow",
            "resource": [
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000.ap-
guangzhou.myqcloud.com/*"
            ]
        }
    ]
}
```

生效条件

最近更新时间：2025-07-10 18:03:12

简介

生效条件是访问策略语言中的一部分，一个完整的生效条件包括如下元素：

- 条件键：表示生效条件的具体类型，例如用户访问来源 IP、授权时间等。
- 条件操作符：表示生效条件的判断方法。
- 条件值：指条件键的值。

详情可参见 访问管理 CAM 生效条件 。

> **说明：**
> - 当您使用条件键编写策略时，请务必遵循最小权限原则，仅为适用请求（action）添加相应的条件键，避免在指定操作（action）时使用通配符“*”，导致请求失败。
> - 当您使用访问管理 CAM 控制台创建策略时，请注意语法格式，version、principal、statement、effect、action、resource、condition 语法元素需保持全小写。

生效条件示例

以下存储桶策略示例中的生效条件（condition）表示用户必须在 10.217.182.3/24 或者 111.21.33.72/24 网段才能完成授权 `cos:PutObject` 操作。其中，

- **条件键为** `qcs:ip` ，表示生效条件的类型是 IP。
- **条件操作符为** `ip_equal` ，表示生效条件的判断方法是判断 IP 地址是否相等。
- **条件值为数组** `["10.217.182.3/24","111.21.33.72/24"]` ，表示生效条件判断的规定值。若用户处于数组中任意一个 IP 所在的网段，条件判断都为 true。

```
{
    "version":"2.0",
    "statement":[
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"allow",
            "action":[
                "name/cos:PutObject"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ],
            "condition":{
                "ip_equal":{
```

```
                    "qcs:ip":[
                        "10.217.182.3/24",
                        "111.21.33.72/24"
                    ]
                }
            }
        }
    ]
}
```

COS 支持的条件键

对象存储（Cloud Object Storage，COS）支持的条件键包括两类，一类是所有请求都适用的，包括 IP、VPC 和 HTTPS；一类是来自请求头部和请求参数的条件键，一般只适用于需要携带该请求头部或者请求参数的请求。关于这些条件键的说明和使用实例可参见 条件键说明及使用示例 文档。

> **说明：**
> 生效条件、条件键等概念都是针对用户请求的访问管理进行的。由于生命周期、存储桶复制规则生效时的删除、复制等动作是由 COS 发起，不是用户发起的请求，条件键对生命周期、存储桶复制规则发起的动作无效。

适用于大部分请求的条件键

第一类，大部分请求都适用的条件键，如下表所示。

条件键	适用请求	含义	类型
cos:secure-transport	所有请求	检查请求是否使用了 HTTPS 协议	Boolean
qcs:ip	所有请求	请求来源的 IP 网段	IP
cos:tls-version	所有 https 请求	https 请求使用的 TLS 版本	Numeric
cos:host	所有请求	请求的 Host 头部	String

来自请求头部和请求参数的条件键

第二类，来自请求头部（Header）和请求参数（Param）的条件键。由于不同的请求具有不同的请求头部和请求参数，因此这些条件键一般只适合在包含这类头部或请求参数的请求中使用。
例如，条件键 `cos:content-type` 适用于需要使用请求头部 `Content-Type` 的上传类请求（PutObject 等），条件键 `cos:response-content-type` 只适用于 GetObject 请求，因为只有该请求支持请求参数 `response-content-type` 。
下表列出了 COS 目前支持的来自请求头部和请求参数的条件键和这些条件键适用的请求。

条件键	适用请求	检测请求头部/请求参数	类型

cos:x-cos-storage-class	PutObject PostObject InitiateMultipartUpload	请求头部：x-cos-storage-class	String
cos:versionid	GetObject DeleteObject PostObjectRestore PutObjectTagging GetObjectTagging DeleteObjectTagging HeadObject	请求参数：versionid	String
cos:prefix	GetBucket GetBucketObjectVersions ListMultipartUploads ListLiveChannels	请求参数：prefix	String
cos:x-cos-acl	PutObject PutObject-Copy PostObject PutObjectACL PutBucket PutBucketACL InitiateMultipartUpload	请求头部：x-cos-acl	String
cos:content-length	该请求头适用范围广，关注代表性请求，例如带请求体的请求	请求头部：Content-Length	Numeric
cos:content-type	该请求头适用范围广，关注代表性请求，例如带请求体的请求	请求头部：Content-Type	String
cos:response-content-type	GetObject	请求参数：response-content-type	String
qcs:request_tag	PutBucket PutBucketTagging	请求头部：x-cos-tagging 请求参数：tagging	String
cos:x-cos-forbid-overwrite	PutObject PutObject-Copy InitiateMultipartUpload CompleteMultipartUpload	请求头部：x-cos-forbid-overwrite	String
cos:object-lock-mode	PutObject PutObject-Copy InitiateMultipartUpload PutObjectRetention	请求头部：x-cos-object-lock-mode 请求体字段：PutObjectRetention 请求体的 Retention.Mode	String
cos:object-lock-remaining-retention-days	PutObject PutObject-Copy InitiateMultipartUpload PutObjectRetention	请求头部：x-cos-object-lock-remaining-retention-days 请求体字段：PutObjectRetention 请求体的 Retention.RetainUntilDate	Timestamp

cos:object-lock-retain-until-date	PutObject PutObject-Copy InitiateMultipartUpload PutObjectRetention	请求头部：x-cos-object-lock-remaining-retention-days 请求体字段：PutObjectRetention 请求体的 Retention.RetainUntilDate	Timestamp
x-cos-grant-read	PutObject PutObject-Copy PostObject PutObjectACL PutBucket PutBucketACL InitiateMultipartUpload	请求头部：x-cos-grant-read	String
x-cos-grant-read-acp	PutObject PutObject-Copy PostObject PutObjectACL PutBucket PutBucketACL InitiateMultipartUpload	请求头部：x-cos-grant-read-acp	String
x-cos-grant-write	PutObject PutObject-Copy PostObject PutObjectACL PutBucket PutBucketACL InitiateMultipartUpload	请求头部：x-cos-grant-write	String
x-cos-grant-write-acp	PutObject PutObject-Copy PostObject PutObjectACL PutBucket PutBucketACL InitiateMultipartUpload	请求头部：x-cos-grant-write-acp	String
x-cos-grant-full-control	PutObject PutObject-Copy PostObject PutObjectACL PutBucket PutBucketACL InitiateMultipartUpload	请求头部：x-cos-grant-full-control	String

条件操作符

COS 的条件键支持以下条件操作符，适用于字符串（String）、数值型（Numeric）、布尔型（Boolean）和 IP 等不同类型的条件键。

条件操作符	含义	类型
string_equal	字符串等于（区分大小写）	String
string_not_equal	字符串不等于（区分大小写）	String
string_like	字符串相似（区分大小写），当前支持在字符串前后添加通配符 `*`，例如 `image/*`	String
ip_equal	IP 等于	IP
ip_not_equal	IP 不等于	IP
numeric_equal	数值等于	Numeric
numeric_not_equal	数值不等于	Numeric
numeric_greater_than	数值大于	Numeric
numeric_greater_than_equal	数值大于等于	Numeric
numeric_less_than	数值小于	Numeric
numeric_less_than_equal	数值小于等于	Numeric

_if_exist 的含义

以上所有条件操作符都支持在后面添加 `_if_exist` 成为一个条件操作符。例如 `string_equal_if_exist` 。条件操作符是否包含 `_if_exist` 的区别在于请求不带条件键对应的请求头或请求参数时如何处理。

- 条件操作符不含有 `_if_exist` ，例如 `string_equal` ，当请求不带对应的请求头/请求参数时，默认不命中条件，即为 `False` 。
- 条件操作符含有 `_if_exist` ，例如 `string_equal_if_exist` ，当请求不带对应的请求头/请求参数时，默认命中条件，即为 `True` 。

示例

示例1：允许下载指定的对象版本

例如，对于下面这个存储桶策略，效力为 allow，表示允许请求参数 versionid 为 “MTg0NDUxNTc1NjIzMTQ1MDAwODg” 的 GetObject 请求通过。如果命中了条件（True），根据 allow 的授权策略，请求会通过；如果没有命中条件（False），根据 allow 的授权策略，请求未获得授权，请求会失败。

```
{
    "version":"2.0",
    "statement":[
```

```
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"allow",
            "action":[
                "name/cos:GetObject"
            ],
            "condition":{
                "string_equal":{
                    "cos:versionid":"MTg0NDUxNTc1NjIzMTQ1MDAwODg"
                }
            },
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ]
        }
    ]
}
```

当条件操作符是 `string_equal` 或 `string_equal_if_exist`，condition 命中情况和请求是否通过如下表所示。

条件操作符	请求	是否命中 condition	请求是否通过
string_equal	不带 versionid	FALSE	不通过
string_equal_if_exist	不带 versionid	TRUE	通过
string_equal	带 versionid，是指定的	TRUE	通过
string_equal_if_exist	带 versionid，是指定的	TRUE	通过
string_equal	带 versionid，不是指定的	FALSE	不通过
string_equal_if_exist	带 versionid，不是指定的	FALSE	不通过

示例2：拒绝下载指定的对象版本

以下存储桶策略示例，效力为 deny，表示拒绝请求参数 versionid为 “MTg0NDUxNTc1NjIzMTQ1MDAwODg” 的 GetObject 请求。如果命中了条件（True），根据 deny 的授权策略，请求会失败；如果没有命中条件（False），根据 deny 的授权策略，请求不会被拒绝。

```
{
    "version":"2.0",
    "statement":[
        {
            "principal":{
```

```
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"deny",
            "action":[
                "name/cos:GetObject"
            ],
            "condition":{
                "string_equal":{
                    "cos:versionid":"MTg0NDUxNTc1NjIzMTQ1MDAwODg"
                }
            },
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ]
        }
    ]
}
```

当条件操作符是 `string_equal` 或 `string_equal_if_exist` ，condition 命中情况和请求是否被拒绝如下表所示。

条件操作符	请求	是否命中 condition	请求拒绝/不拒绝
string_equal	不带 versionid	FALSE	不拒绝
string_equal_if_exist	不带 versionid	TRUE	拒绝
string_equal	带 versionid，是指定的	TRUE	拒绝
string_equal_if_exist	带 versionid，是指定的	TRUE	拒绝
string_equal	带 versionid，不是指定的	FALSE	不拒绝
string_equal_if_exist	带 versionid，不是指定的	FALSE	不拒绝

相关说明

特殊字符需经过 urlencode

请求参数中的特殊字符都需要经过 urlencode，因此您在存储桶策略中使用来自请求参数的条件键时，需要先经过 urlencode。例如，您在使用 `cos:response-content-type` 条件键时，条件值 "image/jpeg" 必须经过 urlencode 转换为 "image%2Fjpeg"，再填入存储桶策略中。

遵循最小权限原则，避免使用*号

使用条件键时，请遵循最小权限原则，只添加您需要设置权限的 action，避免使用通配符“*”。滥用通配符“*”，可能导致部分请求失败。例如下面这个例子，GetObject 之外的其他请求均不支持使用请求参数 response-content-type。deny + string_equal_if_exist 条件操作符在请求中缺失这一条件键时，默认按照 true 处理。因此，当您发起 PutObject、PutBucket 等请求时会命中这个 deny statement，请求被拒绝。

allow + string_equal 在请求中缺失这一条件键时，默认按照 false 处理。因此，当您发起 PutObject、PutBucket 等请求时无法命中 allow statement 时，请求不被允许。

```
{
    "version":"2.0",
    "statement":[
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"allow",
            "action":[
                "*"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ],
            "condition":{
                "string_equal":{
                    "cos:response-content-type":"image%2Fjpeg"
                }
            }
        },
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"deny",
            "action":[
                "*"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ],
            "condition":{
                "string_not_equal_if_exist":{
                    "cos:response-content-type":"image%2Fjpeg"
                }
            }
        }
    ]
}
```

另一种方案，使用 allow+string_equal_if_exist 和 deny + string_not_equal，允许不带 response-content-type 请求参数的请求。
deny + string_equal 条件操作符在请求中缺失这一条件键时，默认按照 false 处理。因此，当您发起 PutObject、PutBucket 等请求时不会命中这个 deny statement，请求不被拒绝。
allow + string_equal_if_exist 在请求中缺失这一条件键时，默认按照 true 处理。因此，当您发起 PutObject、PutBucket 等请求时可以命中 allow statement，请求获得授权。
但这样使用条件操作符，您将无法对 GetObject 是否携带 response-content-type 进行限制。当 GetObject 不携带 response-content-type 请求参数时，它和其他请求一样，将被默认允许通过。只有当 GetObject 携带了 response-content-type 请求参数时，您才可以按照自己规定的条件，检查请求参数的内容是否与您预期的一致，从而实现有条件的授权。

```
{
    "version":"2.0",
    "statement":[
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"allow",
            "action":[
                "*"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ],
            "condition":{
                "string_equal_if_exist":{
                    "cos:response-content-type":"image%2Fjpeg"
                }
            }
        },
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"deny",
            "action":[
                "*"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ],
            "condition":{
```

```
                "string_not_equal":{
                    "cos:response-content-type":"image%2Fjpeg"
                }
            }
        }
    ]
}
```

因此，更加安全的方法是，遵循最小权限原则，不使用通配符“*”，将 action 限定在 GetObject。
如下面这个例子，策略生效条件被严格限定为：GetObject 请求必须携带 response-content-type，且请求参数的值必须为 "image%2Fjpeg"，才能获得授权。
对于其他请求，不受下面例子中策略的影响，您可以遵循最小权限原则，额外单独授权。

```
{
    "version":"2.0",
    "statement":[
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"allow",
            "action":[
                "name/cos:GetObject"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
            ],
            "condition":{
                "string_equal":{
                    "cos:response-content-type":"image%2Fjpeg"
                }
            }
        },
        {
            "principal":{
                "qcs":[
                    "qcs::cam::uin/1250000000:uin/1250000001"
                ]
            },
            "effect":"deny",
            "action":[
                "name/cos:GetObject"
            ],
            "resource":[
                "qcs::cos:ap-guangzhou:uid/1250000000:examplebucket-1250000000/*"
```

```
            ],
            "condition":{
                "string_not_equal_if_exist":{
                    "cos:response-content-type":"image%2Fjpeg"
                }
            }
        }
    ]
}
```

请求方式介绍
使用永久密钥访问 COS

最近更新时间：2024-08-16 17:47:11

背景介绍

通过 RESTful API 对对象存储（Cloud Object Storage，COS）可以发起 HTTP 匿名请求或 HTTP 签名请求。匿名请求一般用于需要公开访问的场景，例如托管静态网站；此外，绝大部分场景都需要通过签名请求完成。
签名请求相比匿名请求，多携带了一个签名值，签名是基于密钥（SecretId/SecretKey）和请求信息加密生成的字符串。
SDK 会自动计算签名，您只需要在初始化用户信息时设置好密钥，无需关心签名的计算；对于通过 RESTful API 发起的请求，需要按照签名算法计算签名并添加到请求中。

获取永久密钥

您可以在访问管理控制台中的 API 密钥管理 页面获取永久密钥，永久密钥包括 SecretId 和 SecretKey，代表了账号的永久身份，不会过期。

- SecretId：用于标识 API 调用者身份。
- SecretKey：用于加密签名字符串和服务端验证签名字符串的密钥。

操作指引请参见以下文档：

- 主账号访问密钥管理
- 子账号访问密钥管理

使用永久密钥访问 COS

通过 API 请求访问 COS

在使用 API 请求时，对于私有桶您必须使用签名请求。通过永久密钥生成签名，放入 Authorization 头部中，形成签名请求；请求发送到 COS，COS 会验证签名与请求是否一致。

> **说明：**
> 由于签名生成算法较为复杂，建议您直接使用 SDK 发起请求省略这一环节。

1. 通过永久密钥生成签名
   签名算法的介绍可参见 请求签名 文档，COS 同时提供了 签名生成工具，也可以通过 SDK 生成签名，参考 SDK 签名实现；您也可以自行编写程序生成签名，但签名算法较为复杂，一般不推荐这种方式。
2. 填入 Authorization 头部
   发起 API 请求时，将签名填入标准 Http Authorization 头部，以下为 GetObject 请求的示例：

```
GET /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
```

```
Authorization: q-sign-algorithm=sha1&q-ak=SecretId&q-sign-time=KeyTime&q-key-time=KeyTime&q-header-list=HeaderList&q-url-param-list=UrlParamList&q-signature=Signature
```

通过 SDK 工具访问 COS

1. 通过永久密钥初始化身份信息
   安装 SDK 工具完成后，首先需要初始化用户身份信息，写入主账号或子账号的永久密钥（SecretId 和 SecretKey）。
2. 直接使用 SDK 请求 COS
   初始化之后，您可以直接使用 SDK 工具进行上传、下载等基本操作，而无需像 API 请求一样自行生成签名，因为 SDK 工具代替您通过密钥生成了签名，向 COS 发起请求。

例如，如下 Java SDK 的代码，更多语言 demo 可参考 SDK 概览 的快速入门文档。

```
// 1 初始化用户身份信息（secretId, secretKey）。
// SECRETID和SECRETKEY请登录访问管理控制台 https://console.cloud.tencent.com/cam/capi进行查看和管理
String secretId = "SECRETID";
String secretKey = "SECRETKEY";
COSCredentials cred = new BasicCOSCredentials(secretId, secretKey);
// 2 设置 bucket 的地域, COS 地域的简称请参照 https://cloud.tencent.com/document/product/436/6224
// clientConfig 中包含了设置 region, https(默认 http), 超时, 代理等 set 方法, 使用可参见源码或者常见问题 Java SDK 部分。
Region region = new Region("COS_REGION");
ClientConfig clientConfig = new ClientConfig(region);
// 这里建议设置使用 https 协议
// 从 5.6.54 版本开始，默认使用了 https
clientConfig.setHttpProtocol(HttpProtocol.https);
// 3 生成 cos 客户端。
COSClient cosClient = new COSClient(cred, clientConfig);
```

使用临时密钥访问 COS

最近更新时间：2024-11-29 11:00:53

临时密钥简介

临时密钥是由 安全凭证服务（Security Token Service,STS） 提供的临时访问凭证。临时密钥由 TmpSecretId、TmpSecretKey 和 Token 三部分组成，相比于永久密钥，临时密钥具有以下特点：

- 有效时间短（30min - 36h），不必暴露永久密钥，降低账号泄露风险。
- 在获取临时密钥时，可通过传入 policy 参数设置临时权限来进一步约束使用者的权限范围。

因此，临时密钥适用于前端直传等临时授权场景，相比永久密钥，分发临时密钥给不受信任的用户，安全性更高。

获取临时密钥

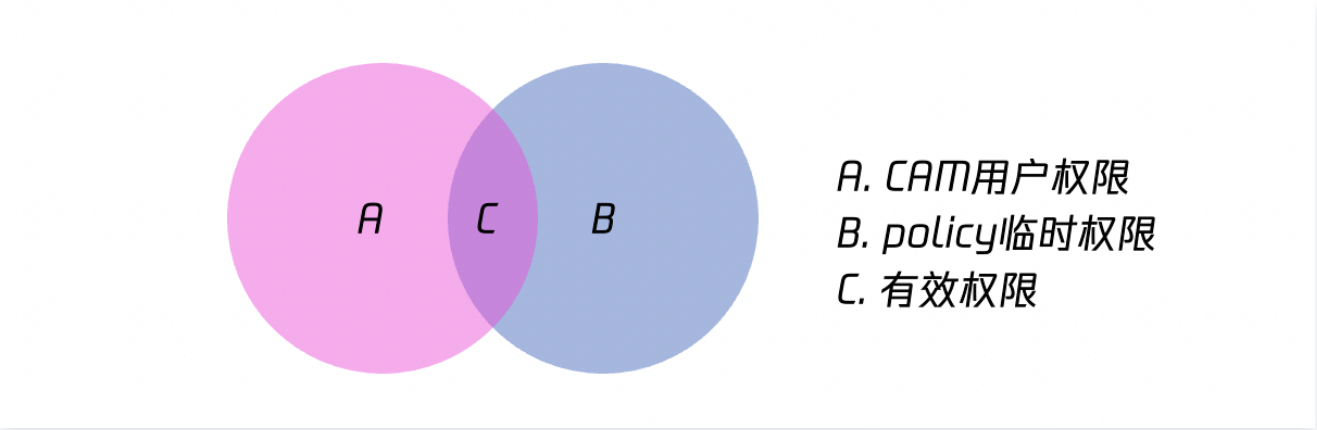
获取临时密钥，可以通过我们提供的 COS STS SDK 方式获取，也可以直接通过 STS 云 API 的方式获取。
详情可参考 临时密钥生成及使用指引 。

临时密钥的权限

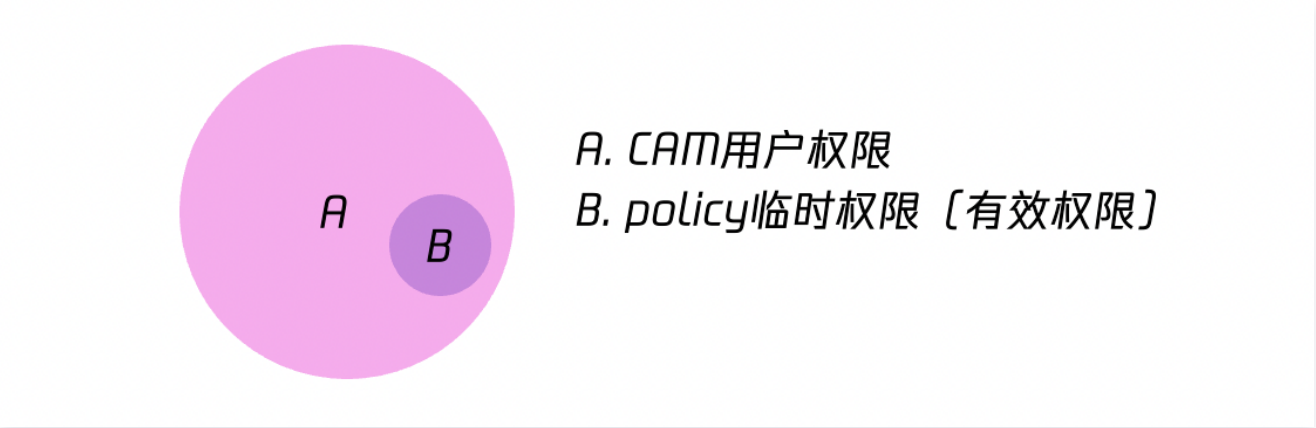
申请临时密钥之前，您必须拥有访问管理（Cloud Access Management，CAM）用户（腾讯云主账号或子账号），可以通过设置 Policy 参数，为临时密钥增加临时策略约束使用者的权限。

- 若不设置 policy 参数，获取的临时密钥具有与 CAM 用户相同的权限。
- 若设置了 policy 参数，获取的临时密钥会在 CAM 用户权限的基础上，进一步将权限限制在 policy 设置的范围以内。

假设，“A”代表 CAM 用户的原有权限，“B”代表通过 policy 参数为临时密钥设置的权限，“A”和“B”的交集代表了临时密钥最终的有效权限。
如下图，CAM 用户权限和 policy 临时权限的交集为有效权限：



如下图，policy 在 CAM 用户权限以内，policy 为有效权限：



使用临时密钥访问 COS

临时密钥包括 SecretId、SecretKey 和 Token，每个主账号和子账号都可以生成多个临时密钥。相比永久密钥，临时密钥的有效期只有30分钟 – 36小时。临时密钥适用于前端直传等临时授权场景，相比永久密钥，分发临时密钥给不受信任的用户，安全性更高，详情参考 临时密钥生成及使用指引 及 用于前端直传的临时密钥使用指引 。

- 发起 API 请求
  类似永久密钥，您也可以通过临时密钥生成签名，填入请求头部 Authorization，形成签名请求。COS 接收到请求后，会校验签名是否有效，以及临时密钥是否过期。
  签名算法的介绍请参见 请求签名 ，COS 同时提供了 签名生成工具 ，也可以通过 SDK 生成签名，请参见 SDK 签名实现 。
- 使用 SDK 工具
  安装 SDK 工具后，除了使用临时密钥初始化用户身份信息，您也可以使用临时密钥（SecretId、SecretKey、Token）初始化 COSClient，直接使用 SDK 进行上传、下载等操作，而无需生成签名。临时密钥生成可参考临时密钥生成及使用指引。

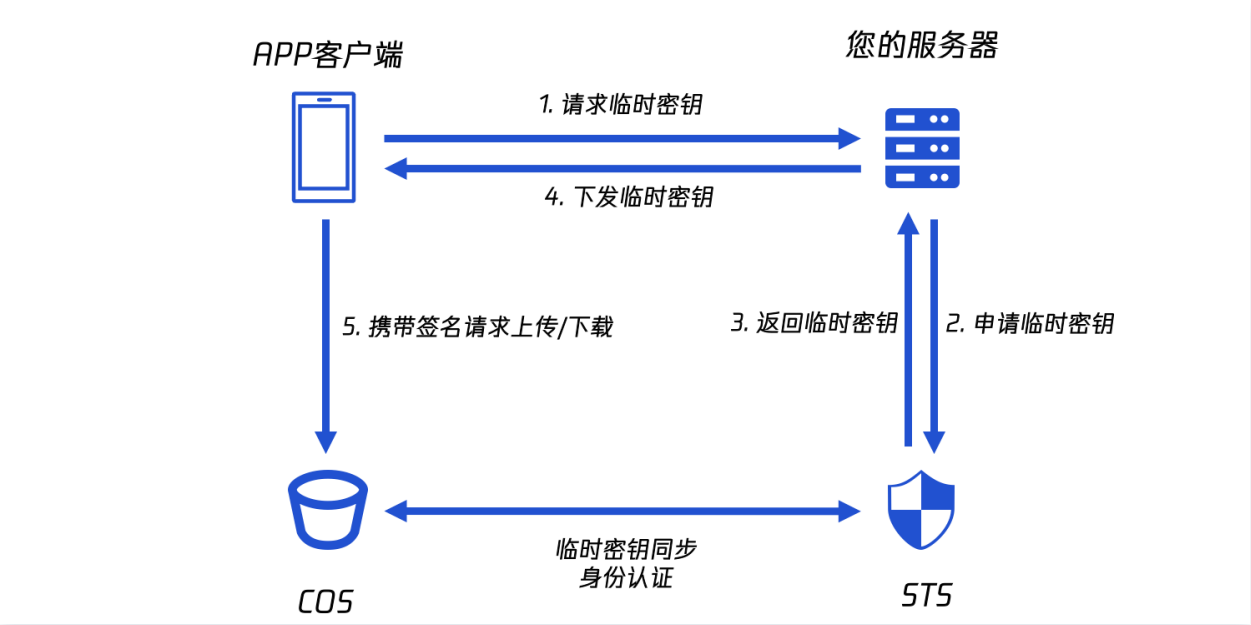
Java SDK 参考示例如下，更多语言 demo 可参考 SDK 概览 。

```
// 1 传入获取到的临时密钥 (tmpSecretId, tmpSecretKey, sessionToken)
String tmpSecretId = "SECRETID";
String tmpSecretKey = "SECRETKEY";
String sessionToken = "TOKEN";
BasicSessionCredentials cred = new BasicSessionCredentials(tmpSecretId,
tmpSecretKey, sessionToken);
// 2 设置 bucket 的地域, COS 地域的简称请参阅
https://cloud.tencent.com/document/product/436/6224
```

```
// clientConfig 中包含了设置 region, https(默认 http), 超时, 代理等 set 方法, 使用可参阅
源码或者常见问题 Java SDK 部分
Region region = new Region("COS_REGION");
ClientConfig clientConfig = new ClientConfig(region);
// 3 生成 cos 客户端
COSClient cosClient = new COSClient(cred, clientConfig);
```

临时密钥的使用场景

临时密钥主要用于授权第三方临时访问 COS。例如，用户开发了客户端 App，将数据存储在 COS 存储桶上，此时将永久密钥直接存放在 App 客户端上是不安全的，但又需要授予客户端上传、下载的权限，针对这种场景可使用临时密钥。



如上图所示，用户开发了 App 客户端，用户服务器上存放有永久密钥，使用临时密钥进行前端直传需要经过以下几步：

1. App 客户端向用户服务器请求临时密钥，用于上传、下载数据。
2. 用户服务器使用永久密钥身份，向 STS 服务器申请临时密钥。
3. STS 服务返回临时密钥给用户服务器。
4. 用户服务器将临时密钥下发到 App 客户端。
5. App 客户端使用临时密钥生成签名请求，向 COS 请求上传、下载数据。

临时密钥适用于前端数据直传的使用场景，您可以参考以下最佳实践使用临时密钥：

- Web 端直传实践
- 小程序直传实践
- 移动应用直传实践

使用预签名 URL 访问 COS

最近更新时间：2024-06-21 11:13:01

对象存储（Cloud Object Storage，COS）支持使用预签名 URL 进行对象的上传、下载，原理是将签名嵌入 URL 生成签名链接。并可以通过签名的有效期，控制预签名 URL 的生效时间。

使用场景

- 使用预签名 URL 进行下载，获取临时 URL 用于临时分享文件、文件夹。或使用长期有效的预签名 URL 用于长期分享文件，详情请参见 文件分享 。
- 使用预签名 URL 进行上传，详情请参见 上传文件 。

文件分享（下载文件）

COS 支持对象的分享，您可以使用预签名 URL 将文件、文件夹限时分享给其他用户。其原理是将签名嵌入拼接在对象 URL 之后，得到**预签名 URL**，有关签名生成算法请参见 请求签名 。其中签名携带了身份信息，因此得到预签名 URL 的用户可以下载对象。
例如，对象 URL 为： `https://test-12345678.cos.ap-beijing.myqcloud.com/test.png`
拼接了签名值的对象 URL 预签名为：

```
https://test-12345678.cos.ap-beijing.myqcloud.com/test.png?q-sign-algorithm=sha1&q-ak=xxxxx&q-sign-time=1638417770;1638421370&q-key-time=1638417770;1638421370&q-header-list=host&q-url-param-list=&q-signaturexxxxxfxxxxxx6&x-cos-security-token=xxxxxxxxxxxx
```

注意事项

- 存储桶默认为私有读，直接通过对象 URL 下载会提示访问失败。
- 如果您一定要使用永久密钥来生成预签名，建议永久密钥的权限范围仅限于上传或下载操作，以规避风险。并且所生成的签名有效时长设置为完成本次上传或下载操作所需的最短期限，因为，当指定预签名 URL 的有效时间过期后，请求会中断。
- 申请新的签名后，需要重新执行失败请求，不支持断点续传。

以下提供的文件分享方法，本质上都是在自动生成签名，并拼接到对象 URL 后面，生成后为临时链接，可以直接用于下载或预览。

方式一：快速获取临时链接作为预签名 URL

您可以通过控制台或 COSBrowser 工具快速获取对象的临时链接。有效期为1 – 2小时。

控制台（Web 页面）

1. 登录 COS 控制台，单击存储桶名称，进入“文件列表”，单击对象详情。

2. 进入对象详情页面，复制临时链接，有效期为1小时。

COSBrowser（桌面端）

参见 COSBrowser 桌面端使用说明中的 生成文件链接 获取对象的临时链接。使用主账号密钥可获取最长2小时的临时链接，使用子账号密钥可获取最长1.5天的临时链接。

方式二：获取自定义时长的临时链接作为预签名 URL

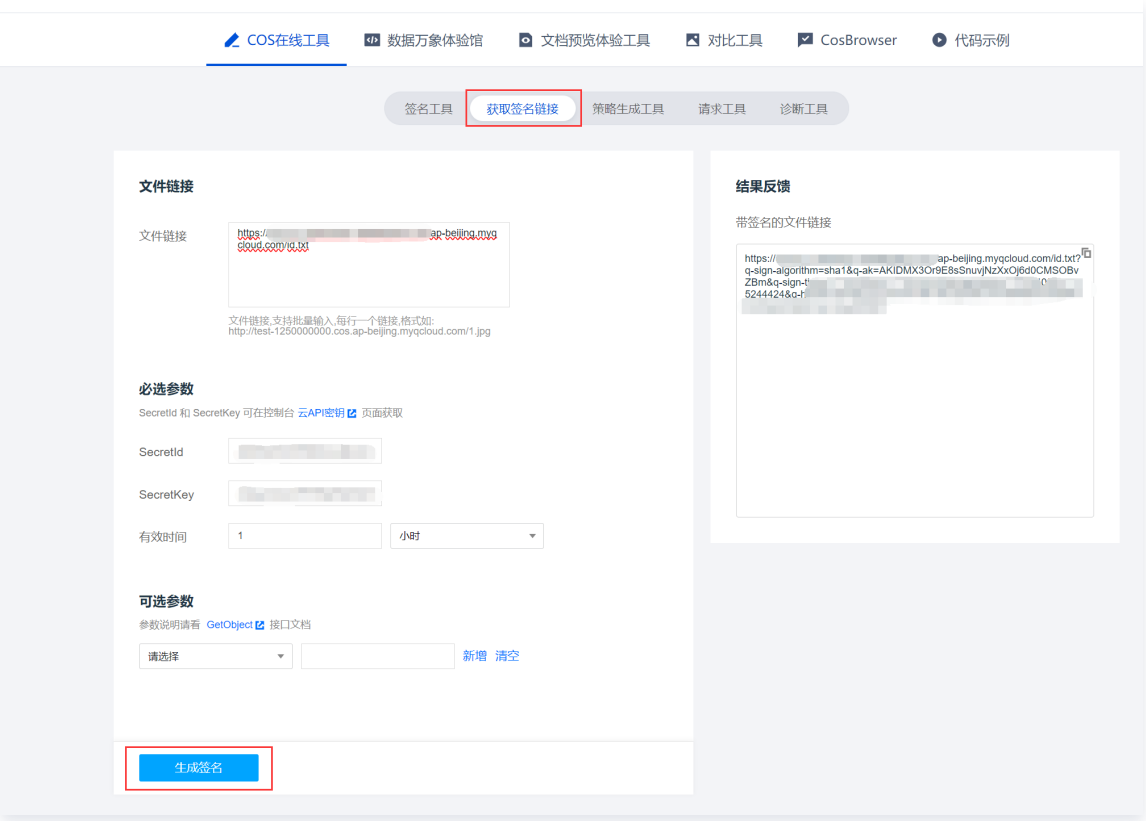
您可以通过两种方式获取自定义时长的对象临时链接：使用签名工具、使用 SDK 批量获取预签名 URL。

使用签名工具

适合对编程不熟悉的用户，操作步骤如下：

1. 获取文件链接。登录 COS 控制台，单击存储桶名称，进入“文件列表”，单击对象详情。进入对象详情页面，复制不带签名的“对象地址”。
2. 从 API 密钥管理 中获取 SecretId 和 SecretKey。

3. 单击 COS 签名工具，获取签名链接。有效时间支持天、小时、分钟、秒级别，默认选中小时。



4. 点击生成签名，在右侧的结果反馈中可获取带签名的临时文件链接。

使用 SDK 批量获取预签名 URL

适合批量获取临时链接、有编程基础的用户。您可以使用临时密钥或永久密钥用于生成预签名 URL。两者的区别在于，临时密钥的最长时效不超过36小时，永久密钥不会过期，这间接影响了预签名 URL 的有效期。
若需要时间更长的临时链接，可以使用 SDK 通过控制签名时长实现生成预签名 URL。生成方法请参见 预签名授权下载，可根据您熟悉的开发语言选择合适的实现方式。

使用永久密钥生成预签名 URL（任意时长）

永久密钥不会过期，预签名 URL 的有效期取决于您设置的签名有效期。您可以直接调用 SDK 的预签名 URL 方法。操作步骤如下：

1. 输入 secret_id、secret_key、region 等初始化 client。
2. 输入您的存储桶名称、对象名称、签名有效期，生成自定义时长的预签名 URL。详情请参见下列各语言 SDK 文档：

Android SDK	C SDK	C++ SDK	.NET SDK
Go SDK	iOS SDK	Java SDK	JavaScript SDK
Node.js SDK	PHP SDK	Python SDK	小程序 SDK

使用临时密钥生成预签名 URL（不超过36小时）

在前端直传的场景中，经常需要使用到临时密钥。关于临时密钥的说明和生成指引您可以参考：

- 使用临时密钥访问COS
- 临时密钥生成及使用指引

- 用于前端直传 COS 的临时密钥安全指引

临时密钥最长为36小时，预签名 URL 的有效期会取您设置的签名有效期和临时密钥有效期的最小值。假设您设置的签名有效期为 X，临时密钥的有效期为 Y，链接的实际生效时间为 T：

```
T=min(X,Y)；由于 X<=36，所以 T<=36。
```

使用临时密钥生成预签名 URL，需要两步：

1. 获取临时密钥 。
2. 获取临时密钥后，可以使用与永久密钥类似的函数生成预签名 URL。需要注意的是，使用临时密钥初始化 client，不仅需要输入 SecretId、SecretKey 还需要输入 token，并且携带参数 `x-cos-security-token` 。详情请参见下列各语言 SDK 文档：

Android SDK	C SDK	C++ SDK	.NET SDK
Go SDK	iOS SDK	Java SDK	JavaScript SDK
Node.js SDK	PHP SDK	Python SDK	小程序 SDK

文件夹分享

文件夹是一种特殊的对象，原理同文件分享一致，您可以通过控制台或 COSBrowser 工具获取临时链接分享文件夹，详情请参见 分享文件夹 。

上传文件

如果您希望第三方可以上传对象到存储桶，又不希望对方使用 CAM 账户或临时密钥等方式时，您可以使用预签名 URL 的方式将签名提交给第三方，以供完成临时的上传操作。收到有效预签名 URL 的任何人都可以上传对象。

> **说明：**
> 如果您一定要使用永久密钥来生成预签名，建议永久密钥的权限范围仅限于上传或下载操作，以规避风险。并且所生成的签名有效时长设置为完成本次上传或下载操作所需的最短期限，因为，当指定预签名 URL 的有效时间过期后，请求会中断；申请新的签名后，需要重新执行失败请求，不支持断点续传。

方式一：使用 SDK 生成预签名 URL（推荐）

各语言 SDK 提供了生成上传预签名 URL 的方法，生成方法可参考 预签名授权上传 ，选择您熟悉的开发语言。

方式二：自行拼接预签名 URL

预签名 URL 实际上就是在对象 URL 之后拼接了签名；因此，您也可以通过 SDK、签名生成工具等，自行生成签名，将 URL 与签名拼接成签名链接用于对象上传。然而，由于签名生成算法较为复杂，一般情况下不推荐这种使用方式。

匿名访问 COS

最近更新时间：2023-09-21 20:29:25

对象存储（Cloud Object Storage，COS）存储桶默认为私有的，访问 COS 必须经过身份验证，通过对象 URL 访问 COS 必须携带签名。但当资源（存储桶、对象、文件夹）开放为公有读后，将允许匿名访问，即可以直接通过对象 URL 下载资源。
根据开放权限的范围，COS 支持在存储桶级别、对象级别、文件夹级别设置公有读。

开放存储桶为公有读

将存储桶开放为公有读私有写，存储桶中所有对象都可被匿名访问。配置方式可参考 设置存储桶访问权限 。

开放对象为公有读

将指定对象开放为公有读私有写，该对象可直接通过对象 URL 访问。配置方式可参考 设置对象的访问权限 。

开放文件夹为公有读

将文件夹开放为公有读私有写，该文件夹下所有文件可被匿名访问。配置方式可参考 设置文件夹权限 。

公有读权限评估机制

COS 权限的评估机制可参考 访问策略评估流程 ，针对存储桶、文件夹、对象级别的公有读权限设置出现冲突时，优先级排序为：
针对某个对象的 ACL，对象 ACL 优先级最高，若对象 ACL 为继承权限，以文件夹 ACL 为准，若文件夹 ACL 为继承权限，以存储桶 ACL 为准。

使用 CDN 加速访问
CDN 加速概述

最近更新时间：2023-08-03 10:16:21

使用内容分发网络（Content Delivery Network，CDN）加速对象存储（Cloud Object Storage，COS），可将存储桶中的内容进行大范围的下载、分发，特别是对于相同内容反复下载的使用场景。通过回源鉴权功能，可以实现使用 CDN 加速私有读存储桶中的内容。通过 CDN 鉴权功能，使内容只能被合法用户下载，避免开放下载可能造成的数据安全与流量成本等问题。

> **说明**
> 用户开启 CDN 加速域名后，使用 CDN 加速域名进行数据的下载访问，将会产生 CDN 回源流量、CDN 流量，详情请参见 COS 作为 CDN 源站时产生的流量 。

内容分发网络

CDN 的定义

CDN 是在现有 Internet 中增加的一层新的网络架构，由遍布全球的高性能加速节点构成。这些高性能的服务节点都会按照一定的缓存策略存储您的业务内容，当您的用户向您的某一业务内容发起请求时，请求会被调度至最接近用户的服务节点，直接由服务节点快速响应，有效降低用户访问延迟，提升可用性。
CDN 会进行缓存和回源行为，即用户在访问某个 URL 的时候，如果被解析到的边缘节点没有命中需要响应的缓存内容，或者缓存已经到期，就会返回到源站去获取需响应的内容。

适用场景

- 对响应延时和下载速度有较高要求的场景。
- 需跨地区、国家、大洲传输数 GB 至数 TB 的数据的场景。
- 需高密集地反复下载相同的内容的场景。

安全类型

- 回源鉴权：当用户请求的数据在边缘节点没有命中缓存时，CDN 需要回源获取数据内容。使用 COS 作为源站并开启回源鉴权后，CDN 边缘节点将使用特殊的服务身份访问 COS 源站，以实现获取并缓存私有访问存储桶中的数据。
- CDN 服务授权：CDN 边缘节点可通过添加 CDN 服务授权，来使用特殊的服务身份访问 COS源站，添加 CDN 服务授权后才可开启回源鉴权。
- CDN 鉴权配置：当用户通过访问边缘节点获取缓存数据的时候，边缘节点会根据鉴权配置规则，校验访问 URL 中的身份验证字段，以此防范非授权的访问，实现防盗链，提高边缘节点缓存数据的安全性和可靠性。

对象存储的访问节点

访问节点定义

访问节点是用户创建存储桶时，依据存储桶的地域和名称，系统自动划分给存储桶的访问域名，可通过该域名访问存储桶中的数据。
开启静态网站功能后，您将额外获得一个静态网站的访问节点，用于呈现与默认访问节点表现不同的、特殊配置的响应内容。

访问节点

- 访问节点：用户在创建一个存储桶后，COS 会自动为存储桶分配一个XML 访问节点，该访问节点形式为 `<BucketName-APPID>.cos.<Region>.myqcloud.com` ，适用于 RESTful API 访问。用户可以通过访问域名，并查看 API 文档 的执行规范，对存储桶进行配置，或进行对象的上传、下载操作。
- 静态网站节点：在控制台的存储桶基本配置界面可以开启托管静态网站的功能，开启后将提供一个访问节点，访问节点形式为 `<BucketName-APPID>.cos-website.<Region>.myqcloud.com` 。静态网站支持特殊的索引页（IndexPage）、错误页（ErrorPage）和跳转等，仅支持针对对象的下载类操作，用户可以通过静态网站域名获取内容。

访问权限

- 公有读：将存储桶设为公有读，任何人都可以通过存储桶的访问域名对其进行访问。如果用户将公有读存储桶作为源站回源，可直接开启 CDN 加速，无须使用 CDN 鉴权和回源鉴权。
- 私有读：将存储桶设为私有读，用户可以通过编写访问策略管理访问者，包括管理 CDN 服务授权等。如果用户将私有读存储桶作为源站回源，开启回源鉴权但未开启 CDN 鉴权，将导致非权限范围内的人群，可通过 CDN 直接访问存储桶，所以对于私有读存储桶，**强烈建议同时开启 CDN 鉴权和回源鉴权保障数据的安全**。

使用 CDN 加速 COS 访问

用户可以通过自定义 CDN 加速域名来对 COS 进行加速访问。首先用户自行准备已备案的自定义域名，并指向源站为 COS 存储桶，即可实现使用自定义 CDN 加速域名加速访问存储桶中的对象。

> **说明**
> 腾讯云 CDN 加速域名默认不提供 IP 地址。如需了解域名的解析情况，可以使用 dig 命令进行查询。

公有读存储桶

将存储桶设置为允许公共访问，配置 CDN 回源到 COS 访问节点时，无需开启回源鉴权，CDN 边缘节点即可获取并缓存存储桶中的对象数据。
您仍然可以通过在 CDN 控制台开启 鉴权配置 **有限度的**保护存储桶中的数据，因为无论是否开启 CDN 中的该功能，知晓存储桶访问域名的用户仍可以访问到存储桶中的所有对象。对于不同 CDN 鉴权配置，域名对公有读存储桶的访问能力见下表：

CDN 鉴权配置	CDN 加速域名访问	COS 域名访问	常见场景
关闭（默认）	可访问	可访问	全站许可公共访问，通过 CDN 或源站均可访问
开启	需使用 URL 鉴权	可访问	对 CDN 访问开启防盗链，但不保护源站访问，不推荐

私有读存储桶

存储桶默认为私有读时，配置 CDN 回源到 COS 访问节点，CDN 边缘节点将**无法获取和缓存任何数据**。因此，需要将 CDN 服务身份加入到存储桶访问策略（Bucket Policy）中，并许可该身份可以执行以下操作：

- GET Object：下载对象
- HEAD Object：查询对象元数据
- OPTIONS Object：预请求跨域配置

在 CDN 控制台 和 COS 控制台 均提供了一键授权的功能，单击**添加 CDN 服务授权**即可完成。完成授权操作后，需要开启**回源鉴权**选项，此时 CDN 边缘将会使用其服务身份访问 COS 中的数据。详情可参见以下文档：

- CDN 加速 COS 资源
- 开启自定义 CDN 加速域名

> **注意**
> - 如果存储桶被设置为私有读，则必须添加授权并开启回源鉴权，否则 COS 将拒绝访问。
> - CDN 边缘会根据每个主账户生成一个服务账号，因此账户授权只对加速域名所属的主账户有效，跨账户绑定加速域名将会被拒绝访问。

对于私有读存储桶，同时开启回源鉴权和 CDN 服务授权会导致通过 CDN 访问源站时无需携带签名，CDN 缓存资源将进行公网分发，导致数据的安全性受到影响，建议**开启 CDN 鉴权配置**保护存储桶中的数据。具体配置步骤请参见 配置说明 文档中的鉴权配置部分。
对于不同 CDN 鉴权配置，域名对私有读存储桶的访问能力见下表：

CDN 鉴权配置	CDN 加速域名访问	COS 域名访问	常见场景
关闭（默认）	可访问	需使用 COS 鉴权	可直接访问 CDN 域名，保护源站数据
开启	需使用 URL 鉴权	需使用 COS 鉴权	全链路保护访问，支持 CDN 鉴权防盗链

CDN 加速配置

最近更新时间：2024-11-29 17:46:43

适用场景

- 对响应延时和下载速度有较高要求的场景。
- 需跨地区、国家、大洲传输数 GB 至数 TB 数据的场景。
- 需高密集地反复下载相同的内容的场景。

> **注意：**
> 当您的下载请求来源于腾讯云 VPC 内（例如使用腾讯云 CVM 访问存储桶），建议直接使用 COS 标准域名。若使用自定义 CDN 加速域名，用户相当于通过外网访问 CDN 节点，将会产生额外的 CDN 回源流量费用、CDN 流量费用等。

相关说明

关于域名定义、CDN 回源鉴权和 CDN 鉴权配置已在 CDN 加速概述 文档中介绍，此处不再赘述。
CDN 回源鉴权、CDN 鉴权配置会影响自定义 CDN 加速域名和 COS 域名对源站存储桶的访问方式，具体情况如下表：

存储桶访问权限	是否开启 CDN 回源鉴权	是否开启 CDN 鉴权配置	通过自定义 CDN 加速域名是否可访问源站	通过 COS 源站域名是否可访问源站	适用场景
公有读	关闭	关闭	可访问	可访问	全站公共访问
公有读	开启	关闭	可访问	可访问	不推荐
公有读	关闭	开启	需使用 URL 鉴权	可访问	不推荐
公有读	开启	开启	需使用 URL 鉴权	可访问	不推荐
私有读+CDN 服务授权	开启	开启	需使用 URL 鉴权	需使用 COS 鉴权	全链路保护
私有读+CDN 服务授权	关闭	开启	需使用 URL 鉴权	需使用 COS 鉴权	不推荐
私有读+CDN 服务授权	开启	关闭	可访问	需使用 COS 鉴权	源站保护
私有读+CDN 服务授权	关闭	关闭	不可访问	需使用 COS 鉴权	不推荐
私有读	关闭	开启或关闭	不可访问	需使用 COS 鉴权	无法使用 CDN

> **注意：**

> 以上表格中的**源站保护**场景，若未开启 CDN 鉴权配置，可能导致您缓存在 CDN 边缘节点上的数据被恶意拉取，因此强烈建议同时开启 CDN 鉴权配置以保障数据的安全。

配置 CDN 加速

用户可以通过 COS 控制台为存储桶绑定自定义域名，然后为自定义域名开启 CDN 加速，最后通过自定义域名加速访问存储桶。绑定自定义域名时，用户需自行在自定义域名的服务提供商处添加 CNAME 解析。

> **注意：**
> 目前 COS 使用自定义加速域名必须开启 CDN，请根据您的情况进行判断。

1. 若您的域名接入国内 CDN，需要备案。但不要求必须通过腾讯云备案，保证接入的域名已备案即可。
2. 若您的域名接入海外 CDN，不需要备案。但需要注意，您在腾讯云上存放的数据和操作行为仍需遵守相关国家的法律法规，以及 《腾讯云服务协议》。

前提条件

1. 注册域名。您可以通过腾讯云 域名注册 或其他服务商注册域名。
2. 域名备案。若域名接入国内 CDN，需要完成域名备案，具体介绍请参见 备案指引 。

操作步骤

> **注意：**
> COS 控制台和 CDN 控制台均可完成自定义域名添加及开启 CDN 加速，若要从 CDN 控制台添加自定义域名，请参见 接入域名 。

1. 选择存储桶

登录 对象存储控制台 ，在左侧导航栏中，单击**存储桶列表**，单击需要开启 CDN 加速的存储桶，进入存储桶。

2. 添加自定义 CDN 加速域名

单击**域名与传输管理 > 自定义 CDN 加速域名**，在**自定义 CDN 加速域名**配置项下，单击**添加域名**，进入可配置状态。

- **域名**：输入待绑定的自定义域名（例如 `www.example.com` ）。请确保输入的域名已备案，并已在 DNS 服务商处设置好对应的 CNAME，详情请参见 CNAME 配置 。若您在接入的自定义 CDN 加速域名为以下情况，则需要进行域名归属权验证，详情请参见 域名归属验证 文档。
  - 首次接入该域名。
  - 该域名已被其他用户接入。
  - 接入域名为泛域名。
- **加速地域**：支持中国境内、中国境外和全球加速，其中全球加速指支持所有地域之间的存储桶加速。
- **源站类型**：默认是**默认源站**，若作为源站的存储桶开启了静态网站，并且希望为静态网站加速，可以将源站类型设置为**静态网站源站**，具体详情请参阅 CDN 加速概述 。
- **回源鉴权**：开启回源鉴权。对于私有读存储桶，请**开启回源鉴权**以保护源站。

> **注意：**
> 对于私有读存储桶，同时开启回源鉴权和 CDN 服务授权会导致通过 CDN 访问源站时无需携带签名，CDN 缓存资源将进行公网分发，导致数据的安全性受到影响，建议开启 CDN 鉴权。

- 开启回源鉴权。

  回源鉴权是用来验证 CDN 边缘节点的服务身份以阻止非法访问，具体情况如下：
  - 公有读存储桶：CDN 边缘节点无需任何授权即可直接访问存储桶，无需开启回源鉴权。
  - 私有读存储桶：CDN 边缘节点需经回源鉴权验证服务身份，验证通过的 CDN 边缘节点才能访问存储桶中对象，选择开启**回源鉴权**。

开启**回源鉴权**后，在右侧单击**保存**，等待5分钟左右，自定义域名的添加与 CDN 加速即可部署完成。

- 开启 CDN 鉴权。

> **注意：**
> 您在为自定义域名启用 CDN 加速之后，任何人都可以通过此域名直接访问源站，如果您的数据有一定的私密性，请您务必开启 CDN 鉴权配置来保护您的源站数据。

自定义域名部署完成后，CDN 鉴权栏会出现 CDN 鉴权功能设置链接，单击**设置**可直接进入 CDN 控制台进行 CDN 鉴权配置，具体操作方式请参见 鉴权配置 。

- **CDN 缓存自动刷新**：开启后，触发 COS 存储桶更新文件规则时，将自动刷新 CDN 缓存，可前往 COS 函数计算配置，操作指引请参见 设置 CDN 缓存刷新 。
- **HTTPS 证书**：如需为自定义 CDN 加速域名添加 HTTPS 证书，可前往 CDN 控制台 进行配置。

> **说明：**
> 更多配置项说明，请参见 开启自定义 CDN 加速域名 。

3. 解析域名

在自定义域名接入 CDN 后，系统会自动为您分配一个 CNAME 域名（以 `.cdn.dnsv1.com` 为后缀），您需要在域名服务提供商处完成 CNAME 的配置，具体请见 CNAME 配置 。

> **注意：**
> CNAME 域名不能直接访问。

4. 关闭功能

以上步骤完成后，即可通过自定义域名加速访问存储桶内资源。如需关闭自定义 CDN 加速，可通过以下方式关闭：
在自定义 CDN 加速域名管理界面，单击**编辑**可更改域名状态，将域名状态由**已上线**改为**已下线**，单击**保存**后等待部署，时间大约为5分钟即可关闭功能， CDN 控制台上相对应的自定义加速域名的状态由**已启动**改为**已关闭**。

> **注意：**
> 如果要删除自定义域名，在自定义域名状态为**已上线**时，无法直接删除域名的，必须将状态改为**已下线**才可删除域名。
> 您可在 CDN 控制台 对域名进行关闭或删除操作，详情请参见 域名操作 。

COS 源站域名升级须知

最近更新时间：2024-10-15 19:06:11

简介

对象存储（Cloud Object Storage，COS）域名已全面升级，对于 V4 版本的域名将不会支持后续 COS 的新增特性与功能。同时，最新的 V5 版本域名可为用户提供更好的访问性能，建议仍在使用 V4 版本域名的用户尽快升级到 V5 版本。V4 和 V5 版本域名区分如下：

- V4 域名形如：examplebucket-1250000000.cossh.myqcloud.com
- V5 域名形如：examplebucket-1250000000.cos.ap-shanghai.myqcloud.com

部分用户在使用内容分发网络（Content Delivery Network，CDN）加速 COS 源站时，依然在使用 V4 版本的 COS 域名，如果您也在使用 V4 版本的域名作为源站，请仔细阅读本文档，确认您的业务属于以下哪种情况，并根据提示进行升级前的准备工作，以避免域名升级可能对您造成的潜在影响。

如何升级到 V5 版本的源站域名

当您进入 CDN 加速配置编辑状态后再次保存，我们会自动为您升级到 V5 版本的 COS 域名，您也可以单击取消，退出编辑状态而不进行升级。

- 当您的存储桶权限为公有读时
  升级到 V5 版本的域名不会对您现有的业务造成影响，升级前后 CDN 业务都可以正常回源您的存储桶。
- 当您的存储桶权限为私有读时
  在使用 V4 域名时，若通过 CDN 回源私有读 COS 存储桶，则需要您的客户端计算好签名，再来访问 CDN。如果不携带签名，则无法访问。而升级到 V5 后，分为以下几种情况：
  - 如果您的 COS 源站为私有，并且希望匿名用户使用 CDN 域名直接进行访问，那么您可以按照以下步骤开启 CDN 回源鉴权功能：

    a. 进行 CDN 服务授权，允许 CDN 服务访问您的指定的存储桶。
    b. 开启 CDN 回源鉴权。
    开启服务授权后，CDN 服务即拥有了访问您私有读存储桶的权限，而匿名用户访问 CDN 域名无需带上任何签名信息。此时 COS 源站为私有，适用　于仅把 CDN 作为公有分发渠道的场景。
  - 如果您开启了 CDN 回源鉴权，匿名用户获取 URL 后均可进行访问。为了避免一些恶意访问的情况，您可以通过设置高级时间戳鉴权来进行盗刷防护，详情请参见 CDN 鉴权配置 进行开启。
  - 如果您希望保持 COS V4 域名的使用习惯，并对回源 COS 的请求进行较强的权限控制，您无需开启回源鉴权功能，也无需进行任何更改，V5 域名依然支持对通过 CDN 回源透传的签名进行权限校验实现访问控制。但是请注意，区别于 COS V4 域名，使用 V5 域名作为 CDN 源站时，如果 CDN 中缓存了您想访问的文件，此时访问 CDN 加速域名不会进行任何的权限校验，即 CDN 的缓存允许匿名请求。

综合上述场景，请您仔细判断您的业务场景，并根据您的实际业务需求选择不同的方案，如有任何问题，也可以 联系我们 获取技术支持。

单链接限速

最近更新时间：2024-11-14 14:22:02

单链接限速

对象存储（Cloud Object Storage，COS）支持上传、下载文件时进行流量控制，以保证您其他应用的网络带宽，您可以在 PUT Object、POST Object、GET Object、Upload Part 请求时携带 x-cos-traffic-limit 参数，并设置限速值，COS 会根据设置的限速值来控制本次请求的网络带宽。

使用说明

- 用户在 PUT Object、POST Object、GET Object、Upload Part 请求时携带 x-cos-traffic-limit 请求头部（对于 POST Object 请求为请求表单字段）来指定本次请求的限速值，该参数可以设置到 header、请求参数中，或者使用表单上传接口时在表单域中。
- x-cos-traffic-limit 参数的值必须为数字，单位默认为 bit/s。
- 限速值设置范围为819200 – 838860800，单位默认为 bit/s，即800Kb/s – 800Mb/s，如果超出该范围将返回400错误。

> **说明**
> 单位换算公式：1MByte=1024KByte=1048576Byte=8388608bit。

API 使用示例

如下为简单上传的 API 示例，限速值为1048576bit/s，即128KB/s：

```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Content-Length: 13
Authorization: q-sign-algorithm=sha1&q-ak=**********************************&q-
sign-time=1561109068;1561116268&q-key-time=1561109068;1561116268&q-header-
list=content-length;content-md5;content-type;date;host&q-url-param-list=&q-
signature=************************************************
x-cos-traffic-limit: 1048576
```

批量处理
批量处理概述

最近更新时间：2025-02-27 19:07:32

对象存储（Cloud Object Storage，COS）批量处理功能可以让您对存储桶内指定的对象范围或对象列表执行指定的操作。

- 有清单模式：您可以通过清单功能生成一份对象清单作为指定的对象列表，或者将您需要处理的对象依照清单文件的格式记录在一份 CSV 格式的文件中，COS 批量处理功能将根据这份对象清单文件进行批量处理。如需进一步了解清单功能，请参见 清单功能概述 。
- 无清单模式（自动生成即时清单）：若您没有清单报告，您可根据需求设置清单信息，例如文件范围、对象版本、筛选标签及筛选时间等，批量处理服务会根据您的设置自动生成即时清单，并用于执行批量处理。

当前 COS 批量处理功能仅支持以下指定操作：

- 批量复制对象
- 批量恢复归档对象
- 批量删除对象

您可以通过 COS 控制台来使用 COS 批量处理功能，具体可参见 批量处理 。

原理

- 当您需要执行批量处理操作时，首先需要建立批量处理任务，在批量处理任务中包含了要对对象列表执行指定操作所需的所有信息，您可以使用清单作为对象列表。
- 当您提供了对象清单，建立并启动批量处理任务后，批量处理功能将对清单内的对象逐一执行您指定的操作。任务正在执行时，您可以通过 COS 控制台监控该任务的执行状态，同时您也可以指定其在完成任务后输出相应的任务报告。任务报告将详细描述您本次任务中所执行的每一项操作的状态。

> **注意：**
> 批量处理功能仅对当前存储桶内的对象有效，如您需要批量处理其他存储桶的对象数据，请在相应存储桶内开启批量处理功能。

对象清单

对象清单是一份记录了待处理的所有对象的列表。如您需要创建一项批量处理任务，首先需提供一份对象清单，以告知 COS 您需要处理的对象。您需要将这一份对象清单文件存放在存储桶中，并提供这一份文件的名称、ETag 和 VersionID（如果有）等信息。您可以通过以下两种方式创建对象清单：

- **COS 清单功能**：该功能输出 CSV 格式的对象清单。有关对象清单的详细信息，您可以参见 清单功能概述 。如您的对象清单中包含对象的版本 ID 信息，那么 COS 在执行批量处理时将处理相应版本 ID 的对象。
- **配置 CSV 文件**：该文件中每一行均必须包含存储桶名称、待处理的对象名称，如果存储桶同时开启了版本控制，则还需包含对象版本 ID。如您未曾开启过版本控制功能，可忽略对象版本 ID 信息。CSV 文件配置的格式如下：

```
examplebucket-appid, exampleobject, PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
examplebucket-appid, exampleobject, jbo9_jhdPEyB4RrmOxWS0kU0EoNrU_oI
```

> **注意：**
> - 如果您的存储桶已开启或者曾经开启过版本控制，并且希望对指定的对象版本执行批量处理，您必须在对象清单中提供该对象的版本 ID 信息。
> - 如果您的存储桶已开启或者曾经开启过版本控制，但您在对象清单中未指定版本 ID，COS 将默认对最新版本的对象进行操作。
> - 当您在创建任务之前上传了与待处理对象同名的对象文件，COS 将默认处理最新版本的对象，而不是创建对象清单时的对象。避免此类操作的方式是开通版本控制功能，并在对象清单中指定版本 ID。
> - 您可以在对象清单中包含存储桶内所有对象，但需注意 COS 在处理海量对象时，任务执行过程会花费较长的时间。

批量处理任务

本部分详细描述如何创建批量处理任务，以及当您创建批量处理任务完成后，系统的反馈情况。
创建批量处理任务需要提供以下信息：

类型	说明
操作	您需要明确对清单中的待处理对象执行何种操作。每一个操作均可以配置相应的参数，COS 将会根据这些操作的配置信息，依次处理清单中的对象。
对象清单	对象清单是一份记录了待处理的所有对象的文件。 • 有清单模式：您可以通过清单功能创建一份对象清单，请参见 清单功能概述 。或者您自行将待处理的对象按照清单文件的格式记录在一份 CSV 格式的文件中，并以此作为对象清单。 • 无清单模式：如您不方便生成清单，也可按需设置需要处理的对象反问，系统会按照您的需要，自动生成即时清单，用于后续的批量处理任务。
优先级	您可以使用优先级来标识当前批量处理任务相对于其他批量处理任务的优先状态。任务优先级并不直接决定您任务完成的先后顺序。如您需要控制多个任务顺序执行，需要您自行检测任务执行状态，在一个任务结束后开始下一个任务。
规则权限	当您配置批量处理任务后，需要保证您的账号具有相应的 IAM 权限执行批量处理操作。例如，您配置了批量处理任务以批量执行 `PUT Object-copy` 操作，此时您应该确保您在源存储桶中具备 `Get Object` 的授权，同时也应当确保您在目标存储桶中具备 `PUT Object` 的授权。此外，对于任意批量处理任务，您都应当保证您具有读取对象清单和写入任务报告的权限。有关权限配置的详细信息，请参见 权限配置 和 存储桶访问策略 。
任务报告	您可以配置在任务完成后输出任务报告。如果您需要输出任务报告，您需要在创建批量处理任务时填写相应的参数，以确保系统能将您的任务报告正确输出至指定存储桶。必填的信息包括存放任务报告的存储桶，任务报告的格式，任务报告是否包含所有任务信息等，任务报告的文件前缀是可选项。
任务描述（可选）	您可以为您创建的批量处理任务提供256字节的任务描述，帮助您跟踪和监控您的任务。COS 将在您的控制台上展示任务描述的详细信息。您可以根据任务描述方便地排列或者过滤您创建的任务。任务描述可以是重复的内容，您可以为相近的任务配置同一任务描述（例如，每周同步复制日志数据），以实现同类任务的管理。

批量处理操作
批量复制对象

最近更新时间：2024-10-14 20:05:31

批量复制对象操作可以用于复制清单中的对象。您可以将指定对象从源存储桶中批量复制到目标存储桶中。目标存储桶所属地域可以相同或者不同。批量复制对象操作支持自定义配置 `PUT Object-copy` 的相关参数，这些配置信息可以影响到副本的元数据信息或者存储类型等信息。有关 PUT Object-copy 的更多内容，请参见 PUT Object-copy 。

相关限制

- 所有待处理对象均需处于同一存储桶中。
- 在一项批处理任务中，仅可配置一个目标存储桶。
- 您需要被授予从源存储桶中读取对象的权限以及往目标存储桶中写入对象的权限。
- 待处理的对象不得超过5GB。
- 不支持检查 ETags 和使用用户自定义密钥的服务端加密。
- 如果目标存储桶中未开启版本控制，且存在同名对象文件，对象存储（Cloud Object Storage，COS）将执行覆盖对象文件的操作。
- 如果清单中的对象具有多个版本，仅支持复制一个版本，在不指定版本号的情况下，默认复制最新版本。

批量恢复归档对象

最近更新时间：2024-08-22 11:17:41

批量恢复归档操作可以用于恢复清单中归档类型的对象。该操作支持自定义配置 POST Object restore 的相关参数，这些配置信息会影响到副本的恢复时间和过期时间。有关 POST Object restore 的更多介绍，请参见 POST Object restore 。
恢复操作支持两种归档类型：归档存储类型和深度归档存储类型。关于这两种类型的更多介绍，请参见 存储类型概述 。
要创建批量恢复归档任务，需指定以下两个参数：

- 恢复模式：分别为标准取回模式和批量取回模式。有关恢复模式的更多信息，请参见 恢复归档对象 。
- 副本有效期：归档类型的对象被恢复后，会产生一个临时副本（用户可指定有效期，最长为365天），该副本会在指定时间后自动过期删除。有关副本有效期的更多信息，请参见 恢复归档对象 。

> **注意：**
> 如果您希望将归档类型的对象永久更改为标准类型，则需要完成以下两步操作：
> 1. 恢复归档对象：将归档类型的对象进行恢复。在恢复完成直到标准副本过期删除这个期间，可对其修改存储类型。
> 2. 修改存储类型：通过 复制对象 的方法，将已完成恢复的对象修改为标准存储类型。

注意事项

- 如果批量恢复归档任务中包含了已完成恢复的对象副本，则开启批量恢复归档任务会更新这些对象的副本有效期，以保证任务中的所有的对象副本的有效期相同。
- 批量恢复任务仅用于发起对象恢复请求。当所有请求发送完成后，批量处理页面将显示该任务进度为已完成。但对象存储（Cloud Object Storage，COS）不会通知您对象什么时候已恢复完成。您可以配置事件通知来接收通知。有关更多信息，请参见 事件通知 。

批量删除对象

最近更新时间：2024-10-22 10:28:52

批量删除操作可以用于删除清单中的对象，有关删除操作的说明可参考 DELETE Object 。存储桶的版本控制状态会影响到批量删除的执行结果，详情可参考 版本控制概述 。

注意事项

1. 根据清单中是否包含了对象的 versionid，决定了执行批量删除任务时发起的删除请求是否携带 versionid：
   - 当清单仅保留当前版本时，会执行不带 versionid 的删除操作，即 `DELETE /example.txt`
   - 当清单列出历史版本时，会执行带 versionid 的删除操作，即 `DELETE /example.txt?versionid=xxx`
2. 存储桶处于未开启、暂停、开启版本控制三种不同的状态时，不带 versionid 的删除操作会有不同的执行结果，带 versionid 的删除则会保证真实删除对象，详情可参考 版本控制概述 。不同情况下，批量删除对象的执行结果如下：

操作	清单1（仅保留当前版本）	清单2（列出历史版本）
删除操作	不带 versionid	带 versionid
未开启版本控制	真实删除	真实删除
已开启版本控制	添加删除标记，不会真实删除	根据 versionid 真实删除
暂停版本控制	添加删除标记，若存储桶中有 null 版本对象，该对象将被真实删除，非 null 版本对象不会被删除	根据 versionid 真实删除

版权所有：腾讯云计算（北京）有限责任公司

管理批量处理任务

最近更新时间：2024-08-02 18:02:42

您可以通过控制台管理批量处理任务，本文详细介绍批量处理任务的相关管理操作。

筛选任务

您可以通过 List Jobs API 列出近90天内创建的批量处理任务。批量处理任务列表包含每一项批量处理任务的信息，例如任务ID，任务描述，任务优先级，任务状态和任务执行情况等信息。您可以通过任务状态，从批量处理任务列表中筛选处于相同状态的任务。当您在控制台上进行筛选操作时，也可以通过任务描述或者任务 ID 进行筛选。

查询任务状态

例如您需要获取更多有关任务的信息，可以通过 DescribeJob API 获取单个任务的所有信息。这一接口会返回指定任务的操作配置，对象清单信息，任务报告等信息。通过这一接口，您可以了解指定任务的详情。

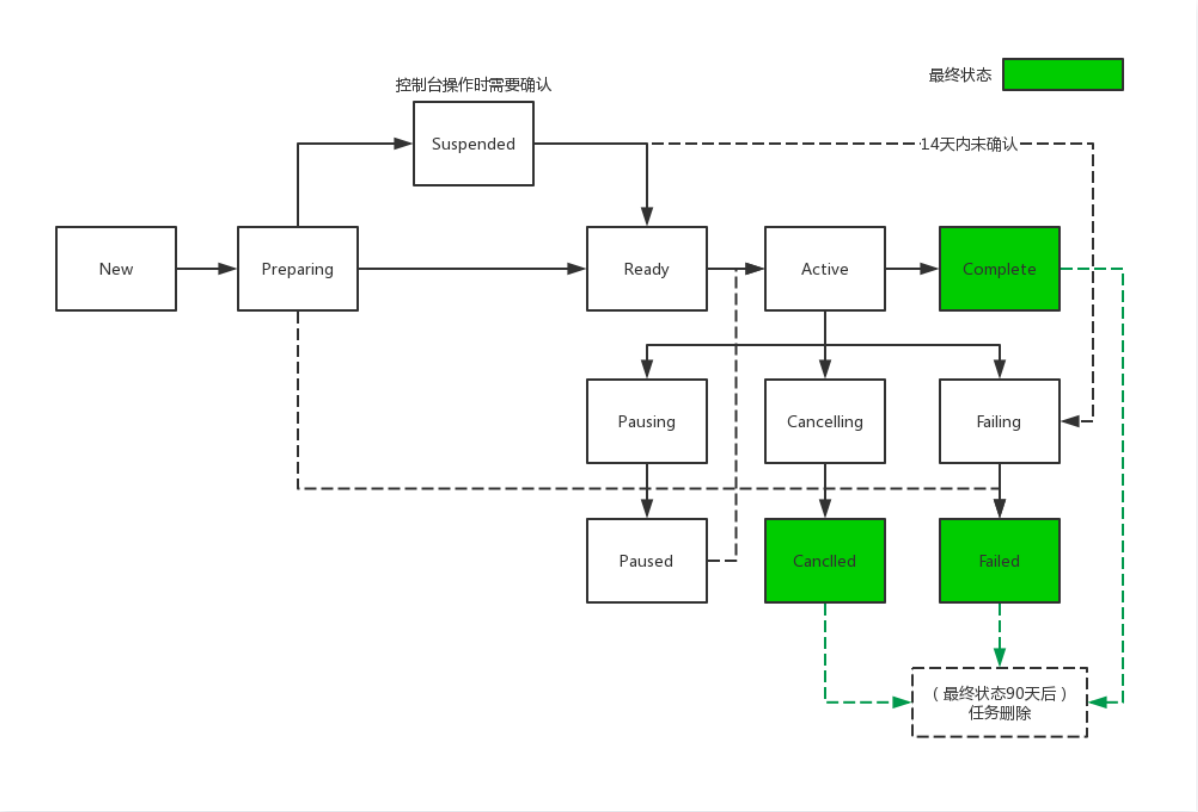
分配任务优先级

您可以为您的任务配置相应的优先级。对象存储（Cloud Object Storage，COS）将根据您所设置的优先级处理您的批量处理任务。高优先级的任务将会被优先执行。优先级以整数型数值表示，级别以降序排列，即数值越大，代表优先级越高。您可以在任务执行过程中修改优先级。如果您在任务执行过程中新增了一项优先级更高的任务，您可以通过暂停低优先级的任务以执行高优先级的任务。

> **注意：**
> 尽管高优先级的批量处理任务通常比低优先级的任务优先执行，但优先级并不作为顺序执行的一项标准。如果您需要配置多项批量处理任务依次执行，请自行监测各项任务的执行状态以实现此目的。

任务状态

当您创建一项任务后，任务状态将会在以下几项状态中切换。状态切换图如下所示：



下表详细介绍了不同任务状态的含义。

任务状态	描述	下一状态
New	当一项任务刚被创建时，任务的状态即为 New。	可转换至 Preparing，代表 COS 开始解析您任务中的清单。
Preparing	COS 开始解析批量处理任务中配置的清单及其他信息时，任务的状态即 Preparing。	可转换至 Ready 状态或者 Suspended 状态。转换至 Ready 状态代表您的批量处理任务配置信息已由 COS 解析完毕，COS 即将根据您的配置对清单中的对象执行指定操作。转换至 Suspended 状态代表您的批量处理任务配置信息已由 COS 解析完毕，但 COS 仍需要您的确认才可执行。这一状态在控制台配置批量处理任务时出现。
Suspended	批量处理任务需要等候您的确认才可以执行，这种待确认的状态即 Suspended ，这一状态是在控制台上创建任务时产生。当经过您确认，COS 开始执行任务时，这一状态切换到下一状态，且不会再转换为该状态。	当您确认执行任务时，任务状态会切换为 Ready。
Ready	COS 解析完您批量处理任务中的对象清单和任务配置信息，即将执行批量处理任务时，任务状态为 Ready。	可转换至 Active 状态，此时 COS 开始执行任务。如果您有更高优先级的任务正在执行中，那么 COS 将保持您的任务状态为 Ready ，直到高优先级的任务状态转换为 Complete。
Active	COS 正在根据您批量处理任务中配置的信息对清单中的对象进行操作时，任务状态为	可转换至 Complete、Failing、Pausing 或者 Cancelling 等状态。当您的任务成功结束或者失败

	Active，您可以通过控制台或者调用 DescribeJob API 等方式查询任务进展。	时，或者由于您的中止命令，例如取消任务时，状态会根据具体的原因进行切换。
Pausing	COS 中断正在处理的批量处理任务，切换至 Paused 的中间状态即 Pausing。	可切换至 Paused 状态。当 COS 确保已中止您正在执行的批量处理任务时，状态即切换至 Paused。
Paused	当您创建一项更高优先级的批量处理任务时，您当前正在执行的批量处理任务将会被切换至 Paused 状态。	当一项高优先级的任务完成、失败或者等待确认时，Paused 状态的任务可自行切换至 Active 状态。
Complete	当批量处理任务已成功完成所有清单内对象的批量操作或者失败时，任务状态切换为 Complete。 如您配置了生成任务报告，COS 将在任务状态切换为Complete 时向您指定存储桶中投递任务报告。	Complete 状态是最终状态，当一项任务切换 Complete 状态时，将不再切换至其他状态。
Cancelling	COS 取消正在处理的批量处理任务，切换至 Cancelled 的中间状态即 Cancelling。	可切换至 Cancelled 状态。当 COS 确保已取消您正在执行的批量处理任务时，状态即切换至 Cancelled。
Cancelled	当批量处理任务已被成功取消时，任务状态为 Cancelled，此时您将无法对任务状态进行任何修改。	Cancelled 状态是最终状态，当一项任务切换 Cancelled 状态时，将不再切换至其他状态。
Failing	COS 切换至 Failed 的中间状态即 Failing。	可切换至 Failed 状态。
Failed	任务失败时，任务状态即 Failed 。有关任务失败的详细信息，可参见 跟踪失败任务 。	Failed 状态是最终状态，当一项任务切换 Failed 状态时，将不再切换至其他状态。

跟踪失败任务

如果批量处理任务执行过程中遇到问题，例如无法成功解析对象清单，则批量处理任务执行失败，COS 将返回相应的错误码及错误原因。COS 将为您保存任务失败原因，您可以通过 DescribeJob API 获取任务失败的详细信息。您也可以通过任务报告获取有关任务失败的原因及其他相关信息。
COS 为每一项批量处理任务提供了操作失败阈值以避免您创建的任务中出现大量失败操作。当一项任务中存在1000个以上的操作时，COS 将会监控操作失败率。任意时刻下，操作失败率（当前执行的失败操作除以全部已执行的操作数）超过50%的阈值时，COS 将会终止任务并返回失败状态。您可以自行筛查导致操作失败率超过阈值的原因，如在对象清单中包含了大量不存在的对象信息，修复错误后重新创建任务。

> **注意：**
> COS 批量处理任务以异步方式运行，并且对对象的操作不一定按清单中列出的对象顺序执行。因此，您不能依据对象清单中的对象顺序来判断任务是否已经操作到某个对象，并据此判断操作的成功或者失败情况。但您可以从任务报告中获取操作的成功或者失败信息。

任务报告

您在创建任务时可以配置是否输出任务报告，如果您配置了输出任务报告，COS 将在任务成功、失败或者被取消时输出一份报告，您可以在这份报告上查询到任务涉及的所有成功或失败操作情况。

任务报告包含任务指定操作的配置参数、执行状态等信息，此外还包括被处理对象的对象名、版本 ID、操作状态码和错误描述等内容。

任务报告生成后，将投递至您所指定的存储桶或具体路径下，任务报告所在的路径格式为如下：

```
# 路径格式
{报告投递存储桶}/{报告投递前缀}/job-{批量任务ID}/results/csv文件
# 示例
examplebucket-1250000000/batch-report/job-84773273-c802-4599-b451-
956992d38d04/results/cMjVlBJEuXFLwzOOKVmxSEuDCQtOeclLHYnxeKfPCADtiudozgVFfwzUlPbXX
XX.csv
```

全球加速
全球加速概述

最近更新时间：2025-03-25 11:49:22

腾讯云对象存储（Cloud Object Storage，COS）的全球加速功能，借助腾讯全局流量调度的负载均衡系统，智能路由解析用户请求，选择最优网络访问链路，实现请求就近接入。利用全球分布的云机房，可帮助全球各地用户快速访问您的存储桶，提升您的业务访问成功率，进一步保障您的业务稳定和提升您的业务体验。此外 COS 的全球加速功能还可以实现数据上传加速和下载加速。

> **注意：**
> - 全球加速功能支持 COS 公有云地域。金融云地域由于网络隔离暂不支持。
> - 使用全球加速功能时，由于请求信息会通过腾讯云内网专线加速传输，因此需要进行收费。具体的价格详情可参见 产品定价 。

使用方法

您可以通过对象存储控制台、API 等方式开启全球加速功能。

使用对象存储控制台

您可以通过对象存储控制台为您的存储桶开启全球加速功能，请参见 开启全球加速 控制台文档。

使用 REST API

您可以直接通过以下 API 开启全球加速功能：
- PUT Bucket Accelerate
- GET Bucket Accelerate

访问域名

开启全球加速功能后，您可以通过两种域名访问您的 COS 文件：
- **存储桶默认域名：**格式如 `<BucketName-APPID>.cos.<Region>.myqcloud.com` ，详情可参见 地域与访问域名 。
- **全球加速域名：**格式如 `<BucketName-APPID>.cos.accelerate.myqcloud.com` 。

以广州地域的存储桶 `examplebucket-1250000000` 为例，假设您已经为该存储桶开启了全球加速功能，当您的业务需要从北京向该存储桶中上传文件 `exampleObject.txt` 时，您有两种上传方式：
- **使用全球加速域名访问：**您在上传时需要指定域名为 `examplebucket-1250000000.cos.accelerate.myqcloud.com` ，通过该域名上传对象时，COS 服务将会根据网络状况为您智能解析并就近接入，例如将请求转发至北京接入层，此后再通过内网专线传输到广州存储层，实现加速的效果。
- **使用存储桶默认域名访问：**您在上传时需要指定域名为 `examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com` ，通过该域名上传对象时，您的请求会被直接转发至广州接入层，此后再到广州存储层，此时外网链路长，可能存在传输不稳定的情况。

> **注意：**

使用全球加速功能会额外产生费用，因此我们建议您根据业务实际情况审慎评估：

- 如果您的业务写入多于读取，且需要从偏远地区上传数据到云机房时（例如 PUT Object、POST Object、Multipart Upload 等操作），我们建议您使用全球加速域名。
- 如果您的业务读取多于写入，涉及的主要场景以文件下载居多（例如 GET Object 操作），我们建议您综合评估 使用 CDN 加速 的方案，选择成本最优的方案。
- 如果您的业务以配置操作为主，或者是检索文件，我们建议您使用存储桶默认域名。
- 如果您的业务是通过同地域的内网环境访问同地域的存储桶，或者通过专线进行访问，我们建议您使用存储桶默认域名。

配置 DNS 智能解析

若企业需要在全球范围内进行访问，可通过 DNS 智能解析的配置，判断用户处于境内或境外，以便更快速响应用户的访问。操作步骤如下：

1. 请登录 DNS 解析控制台，单击**添加域名**，添加需要配置智能解析的自定义域名，进入该域名的**记录管理**页面。
2. 单击**添加记录**，创建2条 CNAME 记录（例如，主机记录设置为 www），**线路类型**分别设置为**境外**和**境内**，**记录值**分别设置为全球加速域名 `examplebucket-1250000000.cos.accelerate.myqcloud.com` 、存储桶的默认域名 `examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com` 。

   如下图所示：

   配置后可以实现的效果：
   - 若您本地 DNS 出口 IP 来源于境外，DNS 查询获取的地址为全球加速域名。
   - 若您本地 DNS 出口 IP 来源于非境外，DNS 查询获取的地址为存储桶的默认域名。

注意事项

使用全球加速域名有如下注意事项：

- 开启全球加速域名后，预计在15分钟左右即可生效，请稍等片刻以待域名生效。
- 开启全球加速域名后，单个存储桶使用加速域名访问的最大带宽将根据全网业务量进行分配。
- 开启全球加速域名后，只有使用加速域名的请求才有加速效果，存储桶默认域名仍然可以正常使用。
- 使用加速域名时，只有请求链路属于加速链路时才会产生加速计费。例如，您使用加速域名从北京上传数据到北京存储桶，此时链路未加速，该请求不产生加速计费。
- 使用加速域名时，您可以指定 HTTP/HTTPS 传输协议，当请求信息在内网专线上传输时，COS 会视情况选择是否需要用 HTTPS 协议来传输以便保障数据传输安全。

计费示例

使用全球加速域名上传数据或访问存储桶将会产生额外的加速计费，加速费用按**天**结算，相关计费项说明与定价可参见 计费概述 和 产品定价 。以下展示使用加速域名和存储桶默认域名的费用对比：

业务场景1

用户业务场景主要是将视频文件上传至 COS，对传输成功率要求高，存储桶设置为广州地域，用户每天需要从新疆和新加坡办公区分别上传1GB视频数据到广州存储桶，则30天的费用如下：

- 使用**加速域名**上传消耗上传费用：30 x 1GB x（0.5元/GB + 1.25元/GB）= 52.5元
- 使用**存储桶默认域名**上传消耗上传费用：30 x 1GB x（0元/GB）= 0元

> **说明：**
> 境内上传加速费用单价为0.5元/GB，境外上传加速费用为1.25元/GB，使用存储桶默认域名上传文件时不需要收取上传流量费用。

业务场景2

用户业务场景主要是加速文件跨境下载，对传输成功率要求高，存储桶设置为广州地域，用户每天需要从新加坡办公区下载1GB视频数据，则30天的费用如下：

- 使用**加速域名**下载消耗下载加速流量费用：30 x 1GB x 1.25元/GB = 37.5元
- 使用**加速域名**下载消耗外网下行流量费用：30 x 1GB x 0.5元/GB = 15元

综合以上，总下载流量费用为37.5 + 15 = 52.5元。

> **说明：**
> 跨境下载加速费用单价为1.25元/GB，使用全球加速域名下载文件时需要收取**外网下行流量费用**和**全球加速下行流量费用**。

内网全球加速

最近更新时间：2024-10-22 10:28:52

腾讯云对象存储服务的内网全球加速能力，借助腾讯云全局流量调度能力，智能路由解析用户请求，选择最优网络访问链路，可帮助用户实现一键式的跨地域内网访问能力。利用内网全球加速能力，可以充分利用腾讯云内网骨干线路的稳定传输质量，提升数据跨地域传输的稳定性和性能。
许多业务具有跨地域内网拉取数据的需求。例如在容器镜像分发场景下，镜像仓库往往采取集中式托管存储的方式存储在某个地域，容器集群则可能根据业务需要部署在不同地域；部署应用时，容器集群需要跨地域拉取镜像，会产生大量的跨地域请求。使用内网全球加速域名可以为这类场景提供稳定的内网跨城专线传输线路，提升镜像分发的稳定性。

> **注意**
> 使用内网全球加速功能时，由于请求信息会通过腾讯云内网专线加速传输，因此需要进行收费。具体的价格详情可参见 产品定价 。

使用方法

您可以通过对象存储控制台、API 等方式开启全球加速功能。

使用对象存储控制台

您可以通过对象存储控制台为您的存储桶开启全球加速功能，详情请参见 开启全球加速 控制台文档。

使用 REST API

您可以直接通过以下 API 开启全球加速功能：

- PUT Bucket Accelerate
- GET Bucket Accelerate

访问域名

开启全球加速功能后，您可以通过内网全球加速域名跨地域内网访问 COS 文件。内网全球加速域名格式形如：`<BucketName-APPID>.cos-internal.accelerate.tencentcos.cn` 。
以某业务为例，该业务的应用镜像存储在广州地域的存储桶 `examplebucket-1250000000` 中，容器集群分别部署在广州、北京和上海地域。该业务团队为了加速容器镜像分发，为存储桶 `examplebucket-1250000000` 开启了全球加速能力，因此北京、上海、广州地域的容器集群可以通过内网全球加速域名 `examplebucket-1250000000.cos-internal.accelerate.tencentcos.cn` 内网拉取镜像包：

- 广州容器集群通过该域名加速拉取文件时：COS 会将请求智能解析到广州地域的内网接入层，直接从广州地域的存储集群中拉取文件。
- 北京/上海地域的容器集群通过该域名加速拉取文件时：COS 会将请求智能解析到北京/上海地域的内网接入层，由内网接入层设备通过跨城骨干网专线，到广州地域的存储集群中拉取文件。

> **注意**
> - 使用全球加速功能会额外产生费用，因此我们建议您根据业务实际情况审慎评估。
> - 内网全球加速域名只能在腾讯云内网环境中使用，如果您的请求源不在腾讯云内网环境下将无法连通。此时您可以考虑使用默认源站域名或者默认的全球加速域名。

- 通过其他腾讯云产品（例如云服务器、容器服务、云函数等）使用内网全球加速域名，云产品所属地域需在 COS 可用地域范围内，否则无法使用。关于 COS 可用地域，请参见 地域和访问域名 。

注意事项

使用全球加速域名有如下注意事项：

- 开启全球加速域名后，预计在15分钟左右即可生效，请稍等片刻以待域名生效。
- 开启全球加速域名后，单个存储桶使用加速域名访问的最大带宽将根据全网业务量进行分配。
- 开启全球加速域名后，只有使用加速域名的请求才有加速效果，存储桶默认域名仍然可以正常使用。
- 使用加速域名时，只有请求链路属于加速链路时才会产生加速计费。例如，您使用加速域名从北京上传数据到北京存储桶，此时链路未加速，该请求不产生加速计费。
- 使用加速域名时，您可以指定 HTTP/HTTPS 传输协议，当请求信息在内网专线上传输时，COS 会视情况选择是否需要用 HTTPS 协议来传输以便保障数据传输安全。

数据工作流
数据工作流概述

最近更新时间：2023-10-18 11:29:13

简介

数据工作流是对象存储全新推出的数据处理服务，其中包含工作流和任务两项功能。

- 工作流：通过工作流，您可以快速、灵活、按需搭建数据处理流程。每个工作流与输入存储桶的一个路径绑定，当文件**上传**至该路径时，该工作流就会被**自动触发**，执行指定的处理操作，并将处理结果自动保存至输出存储桶的指定路径下。
- 任务：针对已存在于存储桶中的文件，您可创建任务进行数据处理操作。
- 模板：您在使用数据工作流时，通常需要设定一系列参数，模板的作用就是将这一系列参数组合在一起。设置模板能**简化您的操作**，使得您设置的参数可被复用而无需重复填写。

> **说明**
> 针对音视频转码、音视频拼接、视频截帧和视频转动图功能，您需要在创建任务或工作流时，指定使用的模板，模板包含**系统预设模板**和**自定义模板**。

- 队列与回调：当您开通数据工作流相关的服务时，系统会为您**自动创建**一个用户队列。当您提交任务后，任务会先进入队列中进行排列，根据优先级和提交顺序依次执行。您还可设置**回调规则**，及时了解任务或工作流进度，同时系统会将处理结果状态提交到您的指定地址。

当前数据工作流中支持的处理操作：

操作名称	说明
音视频转码	提供音频、视频等媒体类文件的转码能力，是文件码流转换成另一个码流的过程。通过转码，可以改变原始码流的编码格式、分辨率和码率等参数，从而适应不同终端和网络环境的播放。
极速高清转码	提供让视频更小更清晰的转码能力，集成了画质修复与增强、内容自适应参数选择、V265编码器等一整套视频处理解决方案，能够保证低网络资源消耗的同时带给用户视觉上更佳体验。
广电专业格式转码	支持 XAVC ProRes 等特殊格式转码。
视频增强	通过细节增强、色彩增强、SDR to HDR 上变换等系列功能，对视频画质进行优化，提升视觉效果。
超分辨率	识别视频的内容与轮廓高清重建视频的细节与局部特征，通过一系列低分辨率的图像来得到一幅高分辨率的图像，可与视频增强搭配使用升级老旧片源。
精彩集锦	精准地提取视频中的精彩片段，对视频的内容、动作姿态、场景进行多维度识别与聚合，以匹配专业编辑的水准迅速剪辑生成视频中精彩集锦。
人声分离	可将指定的视频（或音频）中人声与背景声进行分离生成独立音频素材，便于后期实现其他风格艺术加工。
HLS 自适应打包	提供将单个源视频一步生成多码率自适应文件的能力，让视频适配不同播放终端与网络情况。

HDR to SDR 变换	使变换后视频的画面细节最大程度贴近原视频，适配不同类型的终端设备，避免画面出现失真、灰暗的情况。
视频截帧	提供针对视频某一时间节点的截图功能，可对截帧开始时间点、截帧间隔、截帧数量、输出图片尺寸、输出格式等进行自定义设置，满足多种截帧需求。
音视频拼接	可将指定的视频（或音频）片段拼接在视频（或音频）文件的开头或结尾，生成一个新的视频（或音频）文件。
音视频分段	可将指定的视频（或音频）按指定时长切分成若干个片段。
视频转动图	可将视频格式文件转为动图格式文件，您可选择视频指定时间段转换、视频抽帧方式、输出动图帧率、动图尺寸、动图格式等，满足不同场景下的动图需求。
智能封面	集成了腾讯云先进的 AI 技术，通过对视频内容的理解，智能分析视频帧的质量、精彩程度、内容相关度，提取最优帧生成截图作为封面，提升内容吸引力。
语音识别	语音识别功能可针对录音文件进行识别，异步返回识别文本，目前支持语言类型包括中文普通话、英语和粤语。同时，数据万象支持对识别结果进行处理，包括脏词屏蔽、语气词过滤、阿拉伯数字智能转换等，满足多种语音识别需求。
文档预览	文档预览支持近30多种类型文件以图片或 HTML 方式进行预览，最大程度保留在线浏览时源文档样式，解决不同终端对某些文档类型不支持的问题，满足 PC、App 等多端的文档在线浏览需求。
图片处理	支持灵活的图像编辑，例如图片旋转、裁剪、转码、缩放等，并且提供 Guetzli 压缩、TPG 转码、HEIF 转码等图片瘦身解决方案，图片或文字水印、独有盲水印等版权保护解决方案，满足多种业务场景下的图片需求。

适用场景

多终端适配

内容平台通常使用多终端，因此针对不同用户不同终端需要提供不同格式的媒体文件。音视频转码功能覆盖绝大多数音视频转码需求，同时提供多样的压缩功能以提高压缩效率、减小文件体积，从而减少卡顿并节省存储空间和流量费用。

视频平台

传统视频平台对于视频封面需经过人工浏览视频而后进行筛选，耗费大量人力，延长视频上线时间。
智能封面功能可快速选出最精彩的画面作为封面，节省人力资源，提高视频上新速度。
视频转动图功能可以选择视频中的精彩片段转为动图作为视频预览，用户无需点击播放全视频即可了解视频精彩部分。与传统的静态视频封面相比，动图封面可提高用户点击率，从而提升视频播放量。

视频字幕生成

针对音频文件，可生成单个词语及对应时间戳列表，可便捷地为对应视频增加字幕。

会议语音资料转写

大型会议记录工作内容繁杂，若会议时长较长、参会人员较多，则更加难以完整记录。对象存储的语音识别功能可对中文普通话、英语和粤语进行识别，减少与会人员会议记录工作量，提升会议效果。

在线教育

对于在线教育中的课件、随堂讲义等文档，文档预览功能可帮助实现在线浏览，降低学生使用门槛，提高体验。

网站转码

网站显示文档内容常受限于浏览器规则，对象存储的文档预览功能支持多种文件类型生成图片格式进行预览，可以解决文档内容的页面展示问题。

使用方法

使用对象存储控制台

您可以通过对象存储控制台使用数据工作流服务，详情请参见 配置工作流 和 配置任务 文档。

使用 REST API

您可以使用 API 对工作流及任务进行管理，详情请参见 任务与工作流 API 文档。

监控与告警

最近更新时间：2025-02-26 17:08:12

概述

对象存储 COS 的读写请求量、流量等数据是基于 腾讯云可观测平台 来进行统计和展示的。您可在对象存储控制台或腾讯云可观测平台的 控制台 查看 COS 的读写请求量、流量等详细的监控数据。

> **说明：**
> - 本文主要讲述 COS 控制台获取统计数据的场景，如需了解如何使用数据接口获取更为详细的信息，请使用腾讯云可观测平台的接口，详情请参见 腾讯云可观测平台 产品文档。
> - 目前上报到腾讯云可观测平台的指标都支持所有 COS 地域，详情请参见 地域和访问域名 。

基本功能

腾讯云可观测平台为对象存储提供以下入口实现监控与告警功能。

模块	能力	主要功能
监控概览	展示产品的当前状态	提供总体概况、告警概况、总体监控信息一览
告警管理	支持告警管理与配置	支持新增对象存储的告警策略、自定义消息以及触发条件模板
监控平台	监控流量以及查看用户自定义的监控指标数据	查看用户整体带宽信息，用户可预先自定义监控指标及上报的数据
云产品监控	查看对象存储下的存储桶监控视图	查询当前各存储桶下的读写请求量、流量等监控视图以及数据

使用场景

- **日常管理场景：**登录腾讯云可观测平台控制台，实时查看对象存储的运行状态。
- **异常处理场景：**在监控数据达到告警阈值时发送告警信息，让您能够快速获取异常通知，查询异常原因，并及时处理异常情况。

通过控制台设置和查询

- 您可以通过 腾讯云可观测平台控制台 为对象存储创建告警策略。当监控指标达到设定值，将收到告警提醒，相关操作指引请参见 设置监控告警 。
- 您如需查看监控数据，可在**云产品监控** > 对象存储 页面，查看所有存储桶的监控数据、健康状态和告警策略数。此外您也可以通过对象存储控制台查看，操作指引请参见 查看数据概览 和 查询数据监控 。

通过调用接口查询

您可以通过调用 云产品监控相关接口 查看对象存储的监控数据。如需了解对象存储的监控接口详情，请参见 对象存储监控指标 文档。

监控指标

对象存储监控指标

对象存储监控指标详情，请参见 对象存储监控指标 文档。

> **说明：**
> - 由于对象存储使用的是通用地域，所以无论存储桶所属地域是哪里，拉取对象存储监控指标数据时 Region 请统一选择“广州”地域。
    - 使用 API Explorer 拉取数据时，Region 字段统一选择“华南地区(广州)”。
    - 使用 SDK 拉取数据时，Region 字段统一填写“ap-guangzhou”。
> - 每个指标的统计粒度（Period）可取值不一定相同，可通过 DescribeBaseMetrics 接口获取每个指标支持的统计粒度。
> - 具体的3xx、4xx、5xx状态码详情请参见 错误码表 。

数据处理监控指标

您可以通过调用 云产品监控相关接口 查看数据万象的监控数据，如需了解数据万象的监控接口详情，请参见 数据万象监控指标 文档。

各维度对应参数总览

参数名称	维度名称	维度解释	格式
&Instances.N.Dimensions.0.Name	appid	主账号 APPID 的维度名称	输入 String 类型维度名称：appid
&Instances.N.Dimensions.0.Value	appid	主账号的具体 APPID	输入主账号 APPID，例如：1250000000
&Instances.N.Dimensions.1.Name	bucket	存储桶维度名称	输入 String 类型维度名称：bucket
&Instances.N.Dimensions.1.Value	bucket	存储桶具体名称	输入存储桶具体名称，例如：examplebucket-1250000000

入参说明

查询对象存储监控数据，入参取值如下：
&Namespace = QCE/COS
&Instances.N.Dimensions.0.Name = appid
&Instances.N.Dimensions.0.Value = 主账号的 APPID
&Instances.N.Dimensions.1.Name = bucket
&Instances.N.Dimensions.1.Value = 存储桶名称

监控说明

- **监控间隔**：腾讯云可观测平台提供实时、近24小时、近7天、自定义日期等多种统计区间来监控数据，时间粒度支持1分钟、5分钟、1小时、1天。
- **数据存储**：1分钟粒度监控数据，存储15天；5分钟粒度监控数据，存储31天。1小时粒度监控数据，存储93天。1天粒度监控数据，存储186天。
- **告警展示**：腾讯云可观测平台集成了对象存储的监控数据，数据展示为易读的图表形式，可根据产品预先定义好的告警指标发出告警，有利于用户掌握整体的运行情况。
- **告警设置**：设置监控指标界限值。当监控数据触发告警条件时，腾讯云可观测平台能及时发送告警信息至关心的群体中，详情请参见 告警概述 和 设置监控告警 。