

ALIBABA CLOUD

Alibaba Cloud

对象存储

開發指南

Document Version: 20201013

 Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1. 使用前須知	06
2. 基本概念介紹	07
3. 儲存類型	11
3.1. 儲存類型介紹	11
4. 管理儲存空間	15
4.1. 建立儲存空間	15
4.2. 設定儲存空間讀寫權限 (ACL)	15
4.3. 獲取儲存空間資訊	16
4.4. 查看儲存空間列表	16
4.5. 綁定自訂網域名	16
4.6. 設定跨域訪問	17
4.7. 刪除儲存空間	18
5. 靜態網站託管	19
5.1. 配置靜態網站託管	19
5.2. 教程樣本：使用自訂網域名設定靜態網站託管	19
6. 監控服務	25
6.1. 監控服務概覽	25
6.2. 使用監控服務	26
6.3. 使用報警服務	30
6.4. 訪問監控數據	33
6.5. 監控指標參考	39
6.6. 監控、診斷和故障排除	45
7. 雲端資料處理	53
8. 錯誤處理	54
9. OSS錯誤響應	55
10. 接入OSS	62

10.1. 快速開始	62
10.2. 基於OSS的移動開發	62

1. 使用前須知

本文列出您在使用阿里雲Object Storage Service前需要瞭解的內容。

如果您初次使用阿里雲OSS，請參閱阿里雲[OSS快速入門系列文檔](#)，協助您瞭解OSS並快速使用OSS。

如果您已經充分瞭解OSS，您也可以通過下列資源快速使用OSS的其他各項功能：

資源	描述
阿里雲OSS開發人員指南	本文檔為您講解阿里雲OSS服務的核心概念、所有功能介紹與操作步驟，以及如何使用API和SDK的有效樣本。
阿里雲OSS最佳實務	詳細介紹阿里雲OSS的各種使用情境與配置實踐。
阿里雲OSS SDK參考	介紹主流語言的SDK開發操作和參數。
阿里雲OSS API參考	詳細探討了阿里雲OSS支援的RESTful API操作和相關的樣本。
阿里雲OSS 控制台使用者指南	阿里雲OSS管理主控台可讓您通過介面執行OSS的部分功能。本文檔為您介紹了基於阿里雲OSS管理主控台的所有操作。
阿里雲OSS圖片處理指南	詳細探討了阿里雲OSS提供的圖片處理服務的詳細內容與操作方式。
阿里雲OSS遷移工具	您可能需要將您本地或第三方雲端儲存體服務上的檔案同步到阿里雲OSS上，阿里雲為此提供了完善的遷移解決方案。

2. 基本概念介紹

本部分將向您介紹本產品中涉及的幾個基本概念，以便於您更好地理解 Object Storage Service 產品。

儲存空間 (Bucket)

儲存空間是您用於儲存物件 (Object) 的容器，所有的對象都必須隸屬於某個儲存空間。您可以設定和修改儲存空間屬性用來控制地域、存取權限、生命週期等，這些屬性設定直接作用於該儲存空間內所有對象，因此您可以通過靈活建立不同的儲存空間來完成不同的管理功能。

- 同一個儲存空間的內部是扁平的，沒有檔案系統的目錄等概念，所有的對象都直接隸屬於其對應的儲存空間。
- 每個使用者可以擁有多個儲存空間。
- 儲存空間的名稱在 OSS 範圍內必須是全域唯一的，一旦建立之後無法修改名稱。
- 儲存空間內部的對象數目沒有限制。

儲存空間的命名規範如下：

- 只能包括小寫字母、數字和短橫線 (-)。
- 必須以小寫字母或者數字開頭和結尾。
- 長度必須在 3-63 位元組之間。

對象/檔案 (Object)


對象是 OSS 儲存資料的基本單元，也被稱為 OSS 的檔案。對象由元資訊 (Object Meta)，使用者資料 (Data) 和檔案名 (Key) 組成。對象由儲存空間內部唯一的 Key 來標識。對象元資訊是一個鍵值對，表示了對象的一些屬性，比如最後修改時間、大小等資訊，同時使用者也可以在元資訊中儲存一些自訂的資訊。

對象的生命週期是從上傳成功到被刪除為止。在整個生命週期內，對象資訊不可變更。重複上傳同名的對象會覆蓋之前的對象，因此，OSS 不支援修改檔案的部分內容等操作。

OSS 提供了追加上傳功能，使用者可以使用該功能不斷地在 Object 尾部追加寫入資料。

對象的命名規範如下：

- 使用 UTF-8 編碼。
- 長度必須在 1-1023 位元組之間。
- 不能以正斜線 (/) 或者反斜線 (\) 開頭。

 說明 對象名稱需要區分大小寫。如無特殊說明，本文檔中的對象、檔案稱謂等同於 Object。

Region (地域)

Region 表示 OSS 的資料中心所在的地域，物理位置。使用者可以根據費用、請求來源等綜合選擇資料存放區的 Region。一般來說，距離使用者更近的 Region 訪問速度更快。詳情請查看 [OSS 已經開通的 Region](#)。

Region 是在建立 Bucket 的時候指定的，一旦指定之後就不允許更改。該 Bucket 下所有的 Object 都儲存在對應的資料中心，目前不支援 Object 等級的 Region 設定。

Endpoint (訪問網域名稱)

Endpoint 表示 OSS 對外服務的訪問網域名稱。OSS 以 HTTP RESTful API 的形式對外提供服務，當訪問不同的 Region 的時候，需要不同的網域名稱。通過內網和外網訪問同一個 Region 所需要的 Endpoint 也是不同的。例如杭州 Region 的外網 Endpoint 是 `oss-cn-hangzhou.aliyuncs.com`，內網 Endpoint 是 `oss-cn-hangzhou-internal.aliyuncs.com`。具體的內容請參見各個 Region 對應的 Endpoint。

AccessKey (存取金鑰)

AccessKey，簡稱 AK，指的是訪問身分識別驗證中用到的 AccessKeyId 和 AccessKeySecret。OSS 通過使用 AccessKeyId 和 AccessKeySecret 對稱式加密的方法來驗證某個請求的寄件者身份。AccessKeyId 用於標識使用者，AccessKeySecret 是使用者用於加密簽名字元串和 OSS 用來驗證簽名字元串的密鑰，其中 AccessKeySecret 必須保密。對於 OSS 來說，AccessKey 的來源有：

- Bucket 的擁有者申請的 AccessKey。
- 被 Bucket 的擁有者通過 RAM 授權給第三方要求者的 AccessKey。
- 被 Bucket 的擁有者通過 STS 授權給第三方要求者的 AccessKey。

更多 AccessKey 介紹請參見[存取控制](#)。

強一致性

Object 操作在 OSS 上具有原子性，操作要麼成功要麼失敗，不會存在有中間狀態的 Object。OSS 保證使用者一旦上傳完成之後讀到的 Object 是完整的，OSS 不會返回給使用者一個部分上傳成功的 Object。

Object 操作在 OSS 上同樣具有強一致性，使用者一旦收到了一個上傳 (PUT) 成功的響應，該上傳的 Object 就已經立即可讀，並且資料的三份副本已經寫成功。不存在一種上傳的中間狀態，即 read-after-write 卻無法讀取到資料。對於刪除操作也是一樣的，使用者刪除指定的 Object 成功之後，該 Object 立即變為不存在。

強一致性方便了使用者架構設計，可以使用跟傳統存放裝置同樣的邏輯來使用 OSS，修改立即可見，無需考慮最終一致性帶來的各種問題。

OSS與檔案系統的對比

對比項	OSS	檔案系統
資料模型	OSS 是一個分布式的 Object Storage Service 服務，提供的是一個 Key-Value 對形式的 Object Storage Service 服務。	檔案系統是一種典型的樹狀索引結構。
資料獲取	根據 Object 的名稱 (Key) 唯一的獲取該 Object 的內容。 雖然使用者可以使用類似 <code>test1/test.jpg</code> 的名字，但是這並不表示使用者的 Object 是保存在 <code>test1</code> 目錄下面的。對於 OSS 來說， <code>test1/test.jpg</code> 僅僅只是一個字元串，和 <code>a.jpg</code> 這種並沒有本質的區別。因此不同名稱的 Object 之間的訪問消耗的資源是類似的。	一個名為 <code>test1/test.jpg</code> 的檔案，訪問過程需要先訪問到 <code>test1</code> 這個目錄，然後再在該目錄下尋找名為 <code>test.jpg</code> 的檔案。

對比項	OSS	檔案系統
優勢	支援海量的使用者並發訪問。	支援檔案的修改，比如修改指定位移位置的內容、截斷檔案尾部等。也支援檔案夾的操作，比如重新命名目錄、刪除目錄、移動目錄等非常容易。
劣勢	<p>OSS 保存的 Object 不支援修改（追加寫 Object 需要調用特定的介面，生成的 Object 也和正常上傳的 Object 類型上有差別）。使用者哪怕是僅僅需要修改一個位元組也需要重新上傳整個 Object。</p> <p>OSS 可以通過一些操作來類比類似檔案夾的功能，但是代價非常昂貴。比如重新命名目錄，希望將 test1 目錄重新命名成 test2，那麼 OSS 的實際操作是將所有以 test1/ 開頭的 Object 都重新複製成以 test2/ 開頭的 Object，這是一個非常消耗資源的操作。因此在使用 OSS 的時候要盡量避免類似的操作。</p>	受限於單個裝置的效能。訪問越深的目錄消耗的資源也越大，操作擁有很多檔案的目錄也會非常慢。

因此，將 OSS 映射為檔案系統是非常低效的，也是不建議的做法。如果一定要掛載成檔案系統的話，建議盡量只做寫新檔案、刪除檔案、讀取檔案這幾種操作。使用 OSS 應該充分發揮其優點，即海量資料處理能力，優先用來儲存海量的非結構化資料，比如圖片、視頻、文檔等。

以下是 OSS 與檔案系統的概念對比：

Object Storage Service	檔案系統
Object	檔案
Bucket	主目錄
Region	無
Endpoint	無
AccessKey	無
無	多級目錄
GetService	獲取主目錄列表
GetBucket	獲取檔案清單
PutObject	寫檔案
AppendObject	追加寫檔案
GetObject	讀檔案

Object Storage Service	檔案系統
DeleteObject	刪除檔案
無	修改檔案內容
CopyObject (目的和源相同)	修改檔案屬性
CopyObject	複製檔案
無	重新命名檔案

3. 儲存類型

3.1. 儲存類型介紹

OSS提供標準、低頻訪問、歸檔三種儲存類型，全面覆蓋從熱到冷的各種資料存放區場景。

標準儲存類型 (Standard)

OSS標準儲存類型提供高可靠、高可用、高性能的對象儲存服務，能夠支援頻繁的數據訪問。OSS高吞吐和低延時的服務響應能力能夠有效支援各種熱點類型數據的訪問。標準儲存類型是各種社交、分享類的圖片、音視頻應用、大型網站、大資料分析的合適選擇。

關鍵特性：

- 99.999999999%的數據可靠性
- 按照99.99%服務可用性設計
- 低延時、高吞吐的訪問性能
- 支援HTTPS加密傳輸
- 支援圖片處理

低頻訪問儲存類型 (Infrequent Access)

OSS低頻訪問儲存類型適合長期保存不經常訪問的數據（平均每月訪問頻率1到2次）。儲存單價低於標準類型，適合各類行動裝置 App、智慧型裝置、企業數據的長期備份，支援即時數據訪問。低頻訪問儲存類型的Object有最短儲存時間，儲存時間短於30天的Object提前被刪除會產生一定費用。低頻訪問儲存Object有最小計量空間，Object大小低於64KB，會按照64KB計算儲存空間，數據獲取會產生費用。

關鍵特性：

- 99.999999999%的數據可靠性
- 按照99.99%服務可用性設計
- 支援即時訪問
- 支援HTTPS加密傳輸
- 支援圖片處理
- 有最短儲存時間和最小計量空間

Archive Storage類型 (Archive)

OSSArchive Storage類型在三種儲存類型中單價最低，適合需要長期保存（建議半年以上）的歸檔數據，在儲存周期內極少被訪問，數據進入到可讀取狀態需要1分鐘的解凍時間。適合需要長期保存的檔案數據、醫學影像、科學資料、影視素材。Archive Storage類型的Object有最短儲存時間，儲存時間短於60天的Object提前刪除會產生一定費用。歸檔類型儲存Object有最小計量空間，Object大小低於64KB，會按照64KB計算儲存空間，數據獲取會產生費用。

關鍵特性：

- 99.999999999%的數據可靠性
- 按照99.99%服務可用性設計
- 已經儲存的數據從冷凍狀態恢復到可讀取狀態需要1分鐘的等待時間
- 支援HTTPS加密傳輸
- 支援圖片處理，但需要先解凍
- 有最短儲存時間和最小計量空間

儲存類型對比

對比指標	標準儲存類型	低頻訪問儲存類型	Archive Storage類型
數據可靠性	99.999999999%	99.999999999%	99.999999999%
服務設計的可用性	99.99%	99.99%	99.99% (數據解凍之後)
對象最小計量大小	按照對象實際大小計算	64KB	64KB
最少儲存時間	無最短儲存實際要求	30天	60天
數據取回費用	不收取數據取回費用	按實際獲取的數據收取，單位GB	按實際解凍的數據量收取，單位GB
數據訪問特點	即時訪問 ms延遲	即時訪問 ms延遲	數據需要先解凍，解凍完成後才能讀取，解凍時間需要1分鐘時間
圖片處理	支援	支援	支援，但需要先解凍

🔍 說明

“數據取回費用”中的數據是從底層分布式儲存系統讀取的數據量，在公網傳輸的數據量會計入到流出流量的計費項中。

儲存類型支援的API

API	標準儲存類型	低頻訪問儲存類型	Archive Storage類型
Bucket建立、刪除、查詢			
PutBucket	支援	支援	支援
GetBucket	支援	支援	支援
DeleteBucket	支援	支援	支援
Bucket ACL設定相關			
PutBucketAcl	支援	支援	支援
GetBucketAcl	支援	支援	支援
Bucket日誌功能			
PutBucketLogging	支援	支援	支援
GetBucketLogging	支援	支援	支援
Bucket預設靜態頁面設定			

API	標準儲存類型	低頻訪問儲存類型	Archive Storage類型
PutBucketWebsite	支援	支援	不支援
GetBucketWebsite	支援	支援	不支援
Bucket Referer防盜鏈			
PutBucketReferer	支援	支援	支援
GetBucketReferer	支援	支援	支援
Bucket生命週期			
PutBucketLifecycle	支援	支援	支援, 只支援數據回收
GetBucketLifecycle	支援	支援	支援
DeleteBucketLifecycle	支援	支援	支援
Bucket跨區域複製			
PutBucketReplication	支援	支援	支援
Bucket跨網域設定			
PutBucketcors	支援	支援	支援
GetBucketcors	支援	支援	支援
DeleteBucketcors	支援	支援	支援
Object操作			
PutObject	支援	支援	支援
PutObjectACL	支援	支援	支援
GetObject	支援	支援	支援, 需要先解凍
GetObjectACL	支援	支援	支援
GetObjectMeta	支援	支援	支援
HeadObject	支援	支援	支援
CopyObject	支援	支援	支援
OptionObject	支援	支援	支援
DeleteObject	支援	支援	支援
DeleteMultipleObjects	支援	支援	支援

API	標準儲存類型	低頻訪問儲存類型	Archive Storage類型
PostObject	支援	支援	支援
PutSymlink	支援	支援	支援
GetSymlink	支援	支援	支援
RestoreObject	不支援	不支援	支援
Multipart操作			
InitiateMultipartUpload	支援	支援	支援
UploadPart	支援	支援	支援
UploadPartCopy	支援	支援	支援
CompleteMultipartUpload	支援	支援	支援
AbortMultipartUpload	支援	支援	支援
ListMultipartUpload	支援	支援	支援
ListParts	支援	支援	支援
圖片處理	支援	支援	支援

4. 管理儲存空間

4.1. 建立儲存空間

您可以選擇在已有的地域建立儲存空間。同時需要注意有下列限制：

- 同一用戶建立的儲存空間總數不能超過 30 個。
- 每個儲存空間的名字全域唯一，否則會建立失敗。
- 儲存空間的名稱需要符合命名規範。
- 儲存空間一旦建立成功，名稱和所處地域不能修改。

OSS 提供 ACL (Access Control List) 許可權控制方法，您可以在建立儲存空間的時候設定相應的儲存空間許可權 (ACL)，也可以在建立之後修改 ACL。如果不設定 ACL，預設值為私有讀寫。更多資訊，請參見[設定儲存空間讀寫權限](#)。

功能使用參考

- 控制台：[建立儲存空間](#)
- SDK: Java SDK-[Bucket](#)中新建儲存空間
- API: [PutBucket](#)

4.2. 設定儲存空間讀寫權限 (ACL)

除了在建立儲存空間的時候能夠對儲存空間的 ACL 進行設定，也可以之後根據自己的業務需求對儲存空間的 ACL 進行修改。這個操作只有該儲存空間的建立者有許可權執行。目前儲存空間有三種存取權限：

許可權值	中文名稱	許可權對訪問者的限制
public-read-write	公共讀寫	任何人（包括匿名訪問）都可以對該儲存空間中的檔案進行讀寫操作，所有這些操作產生的費用由該儲存空間的建立者承擔，請慎用該許可權。
public-read	公共讀，私有寫	只有該儲存空間的擁有者或者授權對象可以對該儲存空間內的檔案進行寫操作，任何人（包括匿名訪問）可以對該儲存空間中的檔案進行讀操作。
private	私有讀寫	只有資源擁有者或者授權對象可以對該儲存空間內的檔案進行讀寫操作，其他人在未經授權的情況下無法訪問該儲存空間內的檔案。

詳細解釋請參見[存取權限存取權限"?>](#)。

功能使用參考

設定儲存空間 ACL

- 控制台：[設定存取權限](#)
- SDK: Java SDK-[Bucket](#)中設定Bucket ACL
- API: [Put BucketACL](#)

獲取儲存空間 ACL

- 控制台：登入後可以在儲存空間屬性中查看
- SDK：Java SDK-[Bucket](#)中獲取Bucket ACL
- API：[Get BucketACL](#)

4.3. 獲取儲存空間資訊

您可以獲取儲存空間所屬的地域，即資料中心的物理位置資訊。返回的 Location 欄位顯示儲存空間所在的地域資訊，比如華東 1 (原杭州) 的Location 欄位資訊顯示為 oss-cn-hangzhou。請參見 OSS 提供的[訪問網域名稱](#)。

功能使用參考

- 控制台：進入控制台後在儲存空間屬性中直接顯示地域資訊
- API：[Get Bucket Location](#)
- SDK：Java SDK-[Bucket](#)中獲取Bucket地址

4.4. 查看儲存空間列表

查看您建立的所有儲存空間列表。

功能使用參考

- 控制台：進入控制台後預設顯示您建立的儲存空間列表
- API：[GetService](#)
- SDK：Java SDK-[Bucket](#)中列舉Bucket

相關參考連結

- [建立Bucket](#)

4.5. 綁定自訂網域名

您的檔案上傳到OSS後，會自動生成該檔案的訪問地址。您可以使用此地址訪問OSS檔案。如果您想要通過自訂網域名訪問OSS檔案，需要將自訂網域名訪問綁定在檔案所在的Bucket上，即CNAME。按照中國《互連網管理條例》的要求，所有需要開通這項功能的用戶，必須提供工信部備案號，網域名稱持有人身份證等有效資料，經由阿里雲審批通過後才可以使用。在開通CNAME功能後，OSS將自動處理對該網域名稱的訪問請求。

應用場景

例如，用戶A擁有一個網域名稱為abc.com的網站，網站的網頁中的連結為 `http://img.abc.com/logo.png`。用戶A此時需要將對網站中圖片的請求遷移到OSS，並且不想修改任何網頁的代碼，也就是對外連結不變，CNAME功能特別適合這種場景。流程如下：

1. 用戶A在OSS上建立一個名為abc-img的Bucket，並上傳了其網站上的圖片。
2. 用戶A通過OSS控制台，提交將img.abc.com這個自訂的網域名稱綁定在abc-img上的申請，並提供相應的材料。
3. 通過阿里雲審核後，OSS後台會將img.abc.com做一個映射到abc-img（此處會做許可權驗證）。
4. 用戶A在自己的網域名稱伺服器上，添加一條CNAME規則，將img.abc.com映射成abc-img.oss-cn-hangzhou.aliyuncs.com（即abc-img的OSS網域名稱）。

5. `http://img.abc.com/logo.png` 請求到達OSS後，OSS找到http://img.abc.com/logo.png 的訪問，經過OSS後，實際上訪問的是 `http://abc-img.oss-cn-hangzhou.aliyuncs.com/logo.png`。

	CNAME綁定前	CNAME綁定後
流程對比	<ol style="list-style-type: none"> 1. 訪問 <code>http://img.abc.com/logo.png</code>。 2. DNS解析到使用者服務器 IP。 3. 訪問使用者服務器上的 <code>logo.png</code>。 	<ol style="list-style-type: none"> 1. 訪問 <code>http://img.abc.com/logo.png</code>。 2. DNS解析到abc-img.oss-cn-hangzhou.aliyuncs.com。 3. 訪問OSS上 abc-img 裡的 <code>logo.png</code>。

功能使用參考

控制台：[網域名稱管理](#)

4.6. 設定跨域訪問

跨域訪問，或者說JavaScript的跨域訪問問題，是瀏覽器出於安全考慮而設定的一個限制，即同源策略。當來自於A網站的頁面中的JavaScript代碼希望訪問B網站的時候，瀏覽器會拒絕該訪問，因為A、B兩個網站是屬於不同的域。

在實際應用中，經常會有跨域訪問的需求，比如用戶的網站www.a.com，後端使用了OSS。在網頁中提供了使用JavaScript實現的上傳功能，但是在該頁面中，只能向www.a.com發送請求，向其他網站發送的請求都會被瀏覽器拒絕。這樣就導致用戶上傳的數據必須從www.a.com中轉。如果設定了跨域訪問的話，用戶就可以直接上傳到OSS而無需從www.a.com中轉。

跨域資源共用的實現

跨域資源共用（Cross-Origin Resource Sharing），簡稱CORS，是HTML5提供的標準跨域解決方案，OSS支援CORS標準來實現跨域訪問。具體的CORS規則可以參考[W3C CORS規範](#)。其實現如下：

1. CORS通過HTTP請求中附帶Origin的Header來表明自己來源域，比如上面那個例子，Origin的Header就是www.a.com。
2. 伺服器端接收到這個請求之後，會根據一定的規則判斷是否允許該來源域的請求。如果允許的話，伺服器在返回的響應中會附帶上Access-Control-Allow-Origin這個Header，內容為www.a.com來表示允許該次跨域訪問。如果伺服器允許所有的跨域請求，將Access-Control-Allow-Origin的Header設定為*即可，
3. 瀏覽器根據是否返回了對應的Header來決定該跨域請求是否成功，如果沒有附加對應的Header，瀏覽器將會攔截該請求。如果是非簡單請求，瀏覽器會先發送一個OPTIONS請求來獲取伺服器的CORS配置，如果伺服器不支援接下來的操作，瀏覽器也會攔截接下來的請求。

OSS提供了CORS規則的配置從而根據需求允許或者拒絕相應的跨域請求。該規則是配置在Bucket等級的。詳情可以參考[PutBucketCORS](#)。

這裡有幾個要點：

- CORS相關的Header附加等都是瀏覽器自動完成的，用戶不需要有任何額外的操作。CORS操作也只在瀏覽器環境下有意義。
- CORS請求的通過與否和OSS的身分識別驗證是完全獨立的，即OSS的CORS規則僅僅是用來決定是否附加

CORS相關的Header的一個規則。是否攔截該請求完全由瀏覽器決定。

- 使用跨域請求的時候需要關注瀏覽器是否開啟了Cache功能。當運行在同一個瀏覽器上分別來源於www.a.com和www.b.com的兩個頁面都同時請求同一個跨域資源的時候，如果www.a.com的請求先到達伺服器，伺服器將資源帶上Access-Control-Allow-Origin的Header為www.a.com返回給用戶。這個時候www.b.com又發起了請求，瀏覽器會將Cache的上一次請求返回給用戶，此時Header的內容和CORS的要求不匹配，就會導致後面的請求失敗。

功能使用參考

- API: [跨域資源共用](#)
- SDK: Java SDK-[跨域資源共用](#)
- 控制台: [跨域資源共用](#)

4.7. 刪除儲存空間

您可以刪除您建立的儲存空間。如果儲存空間不為空（儲存空間中有檔案或者是尚未完成的分區上傳），則儲存空間無法刪除，必須刪除儲存空間中的所有檔案和未完成的分區檔案後，儲存空間才能成功刪除。如果想刪除儲存空間內部所有的檔案，推薦使用[生命週期管理](#)。

功能使用參考

- API: [Delete Bucket](#)
- SDK: Java SDK-[Bucket](#)中刪除Bucket
- 控制台: [刪除儲存空間](#)

5. 靜態網站託管

5.1. 配置靜態網站託管

OSS支援靜態網站託管。用戶可以通過[OSS 控制台](#)將自己的儲存空間配置成靜態網站託管模式。

配置生效後，假如這個Bucket在杭州，那麼這個靜態網站的訪問網域名稱為：

```
http://<Bucket>.oss-cn-hangzhou.aliyuncs.com/
```

為了使用戶更方便地管理在OSS上託管的靜態網站，OSS支援兩種功能：

- 靜態頁面支援 (Index Document Support)

靜態頁是指當用戶直接存取靜態網站根網域名稱時，OSS返回的預設靜態頁（相當於網站的index.html）。如果您為一個Bucket配置了靜態網站託管模式，就必須指定一個靜態頁。

- 錯誤頁面支援 (Error Document Support)

錯誤頁面是指在用戶訪問該靜態網站時，如果遇到HTTP 4XX錯誤時（最典型的是404 “NOT FOUND” 錯誤），OSS返回給用戶的錯誤頁面。通過指定錯誤頁面，您可以為您的用戶提供恰當的出錯提示。

例如：使用者佈建索引頁面為index.html，錯誤頁面為error.html，Bucket為oss-sample，Endpoint為oss-cn-hangzhou.aliyuncs.com，那麼：

- 用戶訪問 `http://oss-sample.oss-cn-hangzhou.aliyuncs.com/` 和 `http://oss-sample.oss-cn-hangzhou.aliyuncs.com/directory/` 的時候，相當於訪問 `http://oss-sample.oss-cn-hangzhou.aliyuncs.com/index.html`
- 用戶訪問 `http://oss-sample.oss-cn-hangzhou.aliyuncs.com/object` 的時候，如果object不存在，OSS會返回 `http://oss-sample.oss-cn-hangzhou.aliyuncs.com/error.html`

細節分析

- 所謂靜態網站是指所有的網頁都由靜態內容構成，包括客戶端執行的指令碼或直譯式程式，例如JavaScript。OSS不支援涉及到需要伺服器端處理的內容，例如PHP，JSP，APS.NET等。
- 如果您想使用自己的網域名稱來訪問基於Bucket的靜態網站，可以通過[綁定自訂網域名CNAME](#)來實現。
- 將一個Bucket設定成靜態網站託管模式時，必須指定索引頁面，錯誤頁面是可選的。
- 將一個Bucket設定成靜態網站託管模式時，指定的索引頁面和錯誤頁面必須是該Bucket內的一個Object。
- 在將一個Bucket設定成靜態網站託管模式後，對靜態網站根網域名稱的匿名訪問，OSS將返回索引頁面；對靜態網站根網域名稱的簽名訪問，OSS將返回Get Bucket結果。
- Bucket設定靜態網站託管模式後，對於靜態網站根網域名稱的訪問或者訪問不存在的Object會返回給用戶設定的Object，對此返回的流量和請求將會計費。

功能使用參考

- API: [PutBucketWebsite](#)
- 控制台: [靜態網站託管](#)

5.2. 教程樣本：使用自訂網域名設定靜態網站託管

假設您要在阿里雲 OSS 上託管靜態網站。您註冊了網域名稱（例如，examplewebsite.com）並且想要從 OSS 內容響應對 `http://examplewebsite.com` 和 `http://www.examplewebsite.com` 的請求。不論您是已經有了要在 OSS 上託管的靜態網站，還是要從頭開始建立，都可以通過以下樣本了解如何在阿里雲 OSS 上託管網站。

準備工作

本教程涉及到以下服務：

- 網域名稱註冊

如果您沒有註冊網域名稱（如 `examplewebsite.com`），請選擇一個註冊商進行註冊。阿里雲也提供網域名稱註冊服務。詳情請參考 [阿里雲網域名稱服務 \(DNS\)](#)。

- 阿里雲 OSS

您可以使用阿里雲 OSS 建立 bucket，上傳樣本網頁，配置許可權允許所有人查看內容，然後配置 bucket 的網站託管功能。在本樣本中，因為要允許對 `http://examplewebsite.com` 和 `http://www.examplewebsite.com` 的請求，所以需要建立兩個 bucket。您可以僅在一個 bucket 中託管內容，再配置另一個 bucket 將請求重新導向到託管內容的 bucket。

- 雲解析

您可以將阿里雲解析配置為網域名稱解析服務 (DNS) 供應商。在本樣本中，您可以將網域名稱添加到雲解析並定義 CNAME 記錄，這樣能夠使用您的網域名稱而不是 OSS 分配的訪問網域名稱來訪問 OSS bucket。

- 在本樣本中，我們使用阿里雲解析。但您可以通過大多數註冊商定義指向 OSS bucket 的 CNAME 記錄。

🔍 說明

本教程使用 `examplewebsite.com` 作為網域名稱。請使用您註冊的網域名稱替換此網域名稱。

步驟 1：註冊網域名稱

如果您已有註冊網域名稱，可以跳過該步驟。如果您從未託管過網站，請先註冊一個網域名稱，例如 `examplewebsite.com`。您可以使用阿里雲網域名稱服務 (DNS) 註冊一個網域名稱。

詳情請參考 [購買網域名稱](#)。

步驟 2：建立和配置 bucket 並上傳數據

您可以建立兩個 bucket，以同時支援來自根網域名稱（如 `examplewebsite.com`）和子網域名稱（如 `http://www.examplewebsite.com`）的請求。一個 bucket 用於儲存內容，另一個 bucket 用於將請求重新導向到儲存內容的 bucket。

步驟 2.1：建立兩個 bucket

在該步驟中，使用阿里雲帳戶登入 OSS 控制台，並建立以下兩個 bucket：

- `originbucket`：用於儲存內容
- `redirectbucket`：用於將請求重新導向到 `originbucket`
 1. 登入 [OSS 控制台](#)。
 2. 建立兩個 bucket（例如 `originbucket` 和 `redirectbucket`）。將兩個 bucket 的讀寫權限設定為公共讀

取，以便所有人都可以查看 bucket 的內容。

詳細流程，請參考[建立 bucket](#)。

3. 記下 **originbucket** 和 **redirectbucket** 的訪問網域名稱。在後續步驟中將會用到它們。您可以在 bucket 的概覽頁面上找到 bucket 的訪問網域名稱，如下圖所示。

□

4. 將網站數據上傳到 **originbucket**。

您將在根域 bucket **originbucket** 外託管內容，並且將針對子域 bucket **redirectbucket** 的請求重新導向到根域 bucket **originbucket**。您可以在任一 bucket 中儲存內容。

本樣本將內容託管在 **originbucket** bucket 中。內容可以是任何類型的檔案，例如文字檔、照片和視頻。如果您尚未建立網站，您在本樣本中僅需要兩個檔案。一個檔案用作網站首頁，另一個檔案用作網站的錯誤頁面。

例如，使用以下 HTML 建立一個名為 `index.html` 的檔案，並將其上傳到 bucket。在後續步驟中，將使用此檔案名作為網站的預設首頁。

```
<html>
  <head>
    <title>Hello OSS!</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>開始阿里雲 OSS 託管</p>
    <p>這是索引頁面</p>
  </body>
</html>
```

您可以使用以下 HTML 建立另一個名為 `error.html` 的檔案，並將其上傳到 bucket。此檔案用作網站的 404 錯誤頁面。在後續步驟中，將使用此檔案名作為網站的預設 404 頁面。

```
<html>
<head>
  <title>Hello OSS!</title>
  <meta charset="utf-8">
</head>
<body>
  <p>這是 404 錯誤頁面</p>
</body>
</html>
```

步驟 2.2: 配置 bucket 的網站託管功能

在配置 bucket 的網站託管功能後，可以通過 OSS 分配的訪問網域名稱訪問該網站。

在此步驟中，將 **originbucket** 配置為網站。

1. 登入 **OSS 控制台**。
2. 從 bucket 名稱列表中，選擇 *originbucket*。

3. 單擊**基礎設定**頁籤，找到靜態頁面區域。
4. 單擊**設定**，然後輸入以下資訊：
 - **預設首頁**：索引頁面（相當於網站的 `index.html`）。只能使用已儲存在 bucket 中的 HTML 檔案。在本樣本中，輸入 `index.html`。
 - **預設 404 頁**：當訪問錯誤路徑時，返回預設的 404 頁面。只能使用已儲存在 bucket 中的 HTML 和影像檔。在本樣本中，輸入 `error.html`。
5. 單擊**保存**。

步驟 2.3: 配置索引頁面的重新導向功能

配置 `originbucket` 的預設首頁和錯誤頁面之後，您還需要配置 `redirectbucket` 的預設首頁。

參考以下步驟配置索引頁面的重新導向功能：

1. 登入 **OSS 控制台**。
2. 從 bucket 名稱列表中，選擇 `redirectbucket`。
3. 單擊**基礎設定**頁籤，找到靜態頁面區域。
4. 單擊**設定**，然後在預設首頁文字框中輸入 `index.html`。
5. 單擊**保存**。

步驟 3: 將網域名稱綁定到 OSS bucket

現在，您已有了根網域名稱 `examplewebsite.com` 和 OSS bucket `originbucket`，接下來您需要將網域名稱綁定到 OSS bucket，以便能夠使用您的網域名稱而不是 OSS 分配的網域名稱來訪問 OSS bucket。

在本樣本中，在將網域名稱 `examplewebsite.com` 綁定到 OSS bucket `originbucket` 之前，必須使用 OSS 分配的網域名稱 `originbucket.oss-cn-beijing.aliyuncs.com` 訪問 bucket `originbucket`。在綁定網域名稱 `examplewebsite.com` 後，可以使用此 `examplewebsite.com` 來訪問 OSS bucket。

同樣，還需要將子網域名稱 `www.examplewebsite.com` 綁定到 OSS bucket `redirectbucket`，以便能夠使用 `www.examplewebsite.com` 而不是 OSS 分配的網域名稱 `originbucket.oss-cn-beijing.aliyuncs.com` 來訪問 OSS bucket。

參考以下步驟將根網域名稱 `examplewebsite.com` 綁定到 OSS bucket `originbucket`：

1. 登入 **OSS 控制台**。
2. 從 bucket 名稱列表中，選擇 `originbucket`。
3. 單擊**網域名稱管理**頁籤。
4. 單擊**綁定用戶網域名稱**。
5. 在用戶網域名稱文字框中，輸入根網域名稱 `examplewebsite.com`。
6. 將 CNAME 記錄定義為 `originbucket`。
 - 如果已在阿里雲帳戶下解析網域名稱，則可以開啟**自動添加 CNAME 記錄**開關。然後單擊**提交**。
 - 如果未在阿里雲主帳戶下解析網域名稱，**自動添加 CNAME 記錄**開關處於禁用狀態。執行以下步驟，手動添加 CNAME 記錄，然後單擊**提交**。
 - a. 在雲解析中添加網域名稱。
 - 如果網域名稱是在阿里雲中註冊的，它會自動添加到雲解析列表中。您可以跳過該步驟。
 - b. 在雲解析控制台中，找到網域名稱。

- c. 單擊網域名稱。
- d. 單擊添加解析。
- e. 在添加解析對話方塊中，從記錄類型下拉框中選擇 *CNAME*，然後在記錄值文字框中輸入 bucket 的 OSS 網域名稱。在本樣本中，輸入 `originbucket.oss-cn-beijing.aliyuncs.com`。
- f. 單擊確認。

7. 參考上述步驟，將子網域名稱 `www.examplewebsite.com` 綁定到 OSS bucket `redirectbucket`。

步驟 4：配置網站重新導向功能

您已配置 bucket 的網站託管功能並將自訂網域名綁定到 OSS bucket，接下來需要配置 `redirectbucket`，將 `http://www.examplewebsite.com` 的所有請求重新導向到 `http://examplewebsite.com`。

參考以下步驟配置網站的重新導向功能：

1. 登入 [OSS 控制台](#)。
2. 從 bucket 列表中，選擇 `redirectbucket`。
3. 單擊基礎設定頁籤，找到鏡像回源區域。
4. 單擊設定，然後單擊建立規則。
5. 建立 404 重新導向規則，具體步驟如下：
 - i. 在回源類型區域，選擇 *重新導向*。
 - ii. 在回源條件區域，設定 *HTTP 狀態碼* 為 404。
 - iii. 在回源地址區域，選擇添加前尾碼，輸入 `originbucket` 的網域名稱。在本樣本中，輸入 `examplewebsite.com`。
 - iv. 單擊確定。

步驟 5：（可選）使用阿里雲 CDN 加快網站速度

您可以使用阿里雲 CDN 改善網站性能。CDN 讓您的網站檔案（如 HTML、映像和視頻）可供全球各地的資料中心（即，邊緣節點）使用。當訪問者從您的網站請求檔案時，CDN 自動將請求重新導向到最近邊緣節點上的檔案副本。因此下載速度要快於訪問者從較遠的資料中心請求內容。

CDN 在邊緣節點緩存內容的時間長度由您指定。如果訪問者請求的內容的緩存時間超過了到期日，CDN 會檢查原始伺服器，看看是否有新版本的內容可用。如果有新版本，CDN 將新版本複製到邊緣節點。您對原始內容所做的更改會在訪問者請求內容時複製到邊緣節點。但在到期日前，內容仍為之前的版本。我們建議開啟 *CDN 緩存自動刷新開關*，以便您對原始內容所做的更改可以在 CDN 緩存中自動即時刷新。

參考以下步驟使用 CDN 加快 `originbucket` 速度：

1. 在 CDN 控制台中添加 CDN 加速網域名稱，並複製該網域名稱的 *CNAME* 地址。詳情請參考 [CDN 快速入門](#) 中的步驟 2：添加加速網域名稱。
2. 在雲解析控制台中定義 *CNAME* 記錄。詳情請參考 [配置 CNAME 流程](#)。
3. 開啟 *CDN 緩存自動刷新開關*。
 - i. 登入 [OSS 控制台](#)。
 - ii. 從 bucket 列表中，選擇 `originbucket`。
 - iii. 單擊網域名稱管理頁籤。
 - iv. 開啟 *CDN 緩存自動刷新開關*。
4. 參考上述步驟，使用 CDN 為 `redirectbucket` 加速。

步驟 6：測試網站

要驗證網站是否正常運行，請在瀏覽器中嘗試以下 URL：

URL	結果
<code>http://examplewebsite.com</code>	顯示originbucket中的索引文檔。
originbucket中不存在的檔案的 URL，例如 <code>http://examplewebsite.com/abc</code>	顯示originbucket中的錯誤頁面。
<code>http://www.examplewebsite.com</code>	將您的請求重新導向到 <code>http://examplewebsite.com</code> 。
<code>http://www.examplewebsite.com/abc</code>	將您的請求重新導向到 <code>http://examplewebsite.com/abc</code> 。

 說明 某些情況下，可能需要清除網頁瀏覽器的緩存才能看到預期行為。

清理

如果您建立靜態網站只是為了練習，別忘了刪除分配的阿里雲資源以免產生不必要的費用。刪除阿里雲資源後，網站不再可用。

參考以下步驟進行清理：

1. 禁用阿里雲 CDN 控制台中的網域名稱，然後將其刪除。
2. 刪除雲解析控制台中的 CNAME 記錄。
3. 刪除阿里雲 OSS 控制台中的 OSS 檔案和 bucket。

6. 監控服務

6.1. 監控服務概覽

OSS監控服務為您提供系統基本運行狀態、性能以及計量等方面的監控數據指標，並且提供自訂報警服務，幫助您跟蹤請求、分析使用方式、統計業務趨勢，及時發現以及診斷系統的相關問題。

OSS監控指標分類詳細，主要可以歸類為基礎服務指標、性能指標和計量指標，詳見[OSS監控指標參考](#)。

高即時性

高即時性能夠暴露可能隱藏的峰穀問題，顯示出實際的波動情況，有助於分析和評估業務場景。OSS監控指標的即時性（除了計量指標）是按照分鐘粒度採集彙總的，輸出延時不超過1分鐘，即每分鐘內的用戶資訊都會彙總成一個值，並在一分鐘內輸出，代表這一分鐘的監控情況。

計量指標相關說明

為了保持和計費策略的統一，計量指標的收集和展現存在一定的特殊性，說明如下：

- 計量指標數據是按照小時粒度輸出的，即每個小時內的資源計量資訊都會彙總成一個值，代表這個小時總的計量情況。
- 計量指標數據會有近半個小時的延時輸出。
- 計量指標數據的數據時間是指該數據所統計時間區間的開始時間。
- 計量採集截止時間是當月最後一條計量數據所統計時間區間的結束時間，如果當月沒有產生任何一條計量監控數據，那麼計量資料擷取截止時間為當月1號0點。
- 計量指標數據的展示都是盡最大可能推送的，準確計量請參考費用中心—[使用記錄](#)

舉個例子，假設用戶只使用PutObject這個請求上傳數據，每分鐘平均10次。那麼在2018-05-10 08:00:00到2018-05-10 09:00:00這一個小時時間區間內，用戶的PUT類請求數的計量數據值為600次（10*60分鐘），數據時間為2018-05-10 08:00:00，這條數據將會在2018-05-10 09:30:00左右被輸出。如果這條數據是從2018-05-01 00:00:00開始到現在的最後一條計量監控數據，那麼當月的計量資料擷取截止時間就是2018-05-10 09:00:00。如果2018年5月該用戶沒有產生任何的計量數據，那麼計量採集截止時間為2018-05-01 00:00:00。

OSS報警服務

每個帳號最多能夠設定1000項報警規則。除計量指標和統計指標，其他的監控指標均可配置為報警規則加入報警監控，並且一個監控指標可以配置為多個不同的報警規則。

- 報警服務相關概念參考[報警服務概覽](#)。
- OSS報警服務使用指南詳見[使用報警服務](#)。
- OSS具體監控指標詳見[監控指標參考](#)。

監控數據保留原則

監控數據保留31天，過期自動清除，如果需要離線分析監控數據或者長期下載並保存歷史監控，需要使用工具或者編寫代碼來讀取Cloud Monitor資料存放區，請參考[OpenAPI訪問監控數據](#)。

控制台展示最近7天的數據，如果希望查詢7天以上的歷史數據，建議使用Cloud Monitor提供的SDK進行查詢，參考[OpenAPI訪問監控數據](#)。

OpenAPI訪問監控數據

OSS服務的相關監控指標數據可以通過Cloud Monitor提供的OpenAPI訪問，使用方法可以參考如下文檔：

- [Cloud MonitorAPI參考](#)
- [Cloud MonitorSDK參考](#)
- [訪問監控數據](#)

監控、診斷和故障排除

監控診斷和故障排除通過詳細介紹以下各個方面的內容幫助您更好的了解OSS服務的運行狀態並進行自主診斷和故障排除：

- **服務監控**
介紹如何使用監控服務持續監控OSS儲存服務的健全狀態和性能。
- **跟蹤診斷**
介紹如何使用OSS監控服務和logging記錄功能診斷問題；另外，還介紹如何關聯各種記錄檔中的相關資訊進行跟蹤診斷。
- **故障排除**
提供常見的問題場景和故障排除方法。

注意事項

OSS Bucket全域唯一，如果刪掉Bucket之後再建立同名的Bucket，那麼被刪掉的Bucket的監控以及報警規則會作用在新的同名Bucket上。

6.2. 使用監控服務

OSS監控服務入口

OSS監控服務處於Cloud Monitor控制台中。可以通過如下兩種方式進入。

- 登入[OSS管理主控台](#)，在OSS概覽頁右邊單擊設定報警規則，進入OSS監控服務。
 -
- 直接進入[Cloud Monitor控制台](#)查看OSS監控服務。
 -

OSS監控服務頁面

OSS監控服務首頁的主體由如下三部分組成。

- 用戶概況
- Bucket列表
- 報警規則

該頁面沒有自動刷新功能，可以單擊右上方的刷新按鈕自動更新數據資訊。

單擊前往[OSS控制台](#)可以直接進入OSS控制台介面。

用戶概況

用戶概況頁面從用戶層級監控用戶相關的資訊。主要包括用戶監控資訊、當月計量統計和用戶層級監控指標三大部分。


- 用戶監控資訊
該模組主要展示該帳號所擁有的Bucket總數以及相關的報警規則情況。


-
- ■ 單擊Bucket數量的數字，連結到Bucket列表Tab頁。
 - 單擊報警規則總數的數字，連結到報警規則Tab頁。
 - 單擊處於警示狀態的數字，連結到報警規則Tab頁，並且此時該頁展示的報警規則均處於警示狀態。
 - 單擊已禁用規則數的數字，連結到報警規則Tab頁，並且此時該頁展示的報警規則均被禁用。
 - 單擊警鈴表徵圖下面的數字，連結到報警規則Tab頁，並且此時該頁展示的報警規則均處於警示狀態。
- 當月計量統計

當月計量統計展示了從當月1號0點開始，到採集截止時間為止，這段時間內所使用的OSS服務的計費相關的資源資訊，包括如下指標：

 - 儲存大小
 - 公網流出流量
 - Put類請求數
 - Get類請求數
-
- 各個計量框中展示的數據根據量級自動調整單位，滑鼠停留在數字上方會顯示精確的數值。
-
- 用戶層級監控指標

該模組主要展示具體的用戶層級的監控圖表，主要包括服務監控總覽和請求狀態詳情兩部分，下面會詳細介紹。

 -
 - 圖表展現提供了快速時間範圍選擇按鈕和自訂時間框。
 - 快速時間範圍選擇按鈕提供1小時、6小時、12小時、1天和7天的時間範圍選擇，預設為1小時。
 - 自訂時間框可以自訂起始時間和結束時間，精確到分鐘等級。注意，不支援查詢8天以前的數據。
 -
 - 圖表展示還支援以下功能：
 - 單擊相關圖例可以將該指標曲線隱去，如下圖：
 -
 - 單擊圖形右上表徵圖
 - 

可以將圖形放大展示。注意，表格不支援放大展示。
 - 單擊圖形右上表徵圖
 - 

可以對該圖中展示的指標項設定相關報警規則。詳見[報警服務使用指南](#)。注意，表格和計量參考指標不支援報警設定。
 - 滑鼠按住圖形曲線區域拖放，可以進行時間範圍快速調整放大，單擊恢復縮放迴歸到拖放之前的時間範圍。
 -
 - 服務監控總覽

服務監控總覽頁面主要包括下面監控指標圖：

 - 用戶層級可用性/有效請求率：包括可用性和有效請求率兩項指標。

- 用戶層級總請求數/有效請求數：包括總請求數和有效請求數兩項指標。
- 用戶層級流量：包括公網流出流量、公網流入流量、內網流出流量、內網流入流量、CDN流出流量、CDN流入流量、跨區域複製流出流量和跨區域複製流入流量八項指標。
- 用戶層級請求狀態分布：該表格中展示選定時間範圍內各個請求類型的個數以及佔比。
-
- 請求狀態詳情

請求狀態詳情是對請求狀態分布統計的一個具體監控，主要包括下面的監控指標圖：
- 用戶層級服務端錯誤請求數。
- 用戶層級服務端錯誤請求佔比。
- 用戶層級網路錯誤請求數。
- 用戶層級網路錯誤請求佔比。
- 用戶層級客戶端錯誤請求數：包括資源不存在錯誤請求數、授權錯誤請求數、客戶端逾時錯誤請求數和客戶端其他錯誤請求數四項指標。
- 用戶層級客戶端錯誤請求佔比：包括資源不存在錯誤請求佔比、授權錯誤請求佔比、客戶端逾時錯誤請求佔比和客戶端其他錯誤請求佔比四項指標。
- 用戶層級有效請求數：包括成功請求數和重新導向請求數兩項指標。
- 用戶層級有效請求佔比：包括成功請求佔比和重新導向請求佔比兩項指標。
-

Bucket列表

- Bucket列表資訊

列表展現該帳號所擁有的Bucket的名稱、所屬地域、建立時間、當月計量數據統計資訊以及相關操作。

-
- 當月計量統計包括每個Bucket的儲存量、公網流出流量、Put類請求數和Get類請求數。
- 單擊監控圖表或者對應的Bucket名稱，能夠進入具體的Bucket監控視圖頁。
- 單擊報警規則，進入報警規則Tab頁，並且展現所有屬於該Bucket的報警規則。
- 通過上面的搜尋方塊能夠模糊比對快速找到具體的Bucket。
- 選中Bucket複選框，並單擊設定報警規則可以批量設定報警規則，詳見[報警服務使用指南](#)。

- Bucket層級監控視圖

單擊Bucket列表中具體的Bucket行中的監控圖表，就能進入對應的Bucket的監控視圖頁，如下圖：

-
- Bucket監控視圖頁按指標分組進行展示監控圖，主要包含六個指標分組：
 - ■ 服務監控總覽
 - 請求狀態詳情
 - 計量參考
 - 平均延時
 - 最大延時
 - 成功請求操作分類

除了計量參考，所有的指標項都是分鐘等級彙總展示的。不同於用戶層級預設時間展現為最近1小時，Bucket層級的監控展示預設為6小時。單擊上方的返回Bucket列表能夠回到Bucket列表Tab頁。

○ 服務監控總覽

該指標分組同用戶層級的服務監控總覽，只是從具體的Bucket進行監控，主要包括下面監控指標圖：

- 可用性/有效請求率：包括可用性和有效請求率兩項指標。
- 總請求數/有效請求數：包括總請求數和有效請求數兩項指標。
- 流量：包括公網流出流量、公網流入流量、內網流出流量、內網流入流量、cdn流出流量、cdn流入流量、跨區域複製流出流量和跨區域複製流入流量八項指標。
- 請求狀態分布：該表格中展示選定時間範圍內各個請求類型的個數以及佔比。

□

○ 請求狀態詳情

該指標分組同用戶層級的請求狀態詳情，只是從具體的Bucket進行監控，主要包括下面監控指標圖：

- 服務端錯誤請求數
- 服務端錯誤請求佔比
- 網路錯誤請求數
- 網路錯誤請求佔比
- 客戶端錯誤請求數：包括資源不存在錯誤請求數、授權錯誤請求數、客戶端逾時錯誤請求數和客戶端其他錯誤請求數四項指標。
- 客戶端錯誤請求佔比：包括資源不存在錯誤請求佔比、授權錯誤請求佔比、客戶端逾時錯誤請求佔比和客戶端其他錯誤請求佔比四項指標。
- 有效請求數：包括成功請求數和重新導向請求數兩項指標。
- 有效請求佔比：包括成功請求佔比和重新導向請求佔比兩項指標。

□

○ 計量參考

計量參考分組展示各個計量相關的指標資訊，以小時粒度收集展現，如下圖所示：

□

包含以下計量指標監控圖：

- 儲存大小
- 公網流出流量
- 計費請求數：包括Get類請求數和Put類請求數兩項指標項。

如果新建Bucket，需要到目前時間點的下一個整小時點才會採集到新數據，然後在半個小時內展示出來。

○ 平均延時

該分組包含分API類型監控的各項平均延時指標，包含如下幾個指標圖：

- GetObject請求平均延時
- HeadObject請求平均延時
- PutObject請求平均延時
- PostObject請求平均延時
- AppendObject請求平均延時
- UploadPart請求平均延時
- UploadPartCopy請求平均延時

每個指標圖中都包含對應的平均E2E延時和平均伺服器延時，如下圖所示：

□

- 最大延時

該分組包含分API類型監控的各項最大延時指標，包含如下幾個指標圖：

- GetObject請求最大延時
- HeadObject請求最大延時
- PutObject請求最大延時
- PostObject請求最大延時
- AppendObject請求最大延時
- UploadPart請求最大延時
- UploadPartCopy請求最大延時

每個指標圖中都包含對應的最大E2E延時和最大伺服器延時，如下圖所示：

□

- 成功請求操作分類

該分組包含分API類型監控的各項成功請求數指標，包含如下幾個指標圖：

- GetObject成功請求
- HeadObject成功請求
- PutObject成功請求
- PostObject成功請求
- AppendObject成功請求
- UploadPart成功請求
- UploadPartCopy成功請求
- DeleteObject成功請求
- DeleteObjects成功請求

如下圖所示：

□

報警規則

報警規則Tab頁能夠展示和管理報警規則，如下圖所示：

□

報警規則頁的使用和相關說明請參見[使用報警服務](#)。

監控關注事項以及使用指導

監控關注點以及使用指南請參見[監視診斷和故障排除](#)。

6.3. 使用報警服務

在介紹OSS監控服務控制台之前，請先閱讀Cloud Monitor提供的監控服務文檔，了解基本概念並進行報警連絡人和報警聯繫組的配置。

- [報警服務概覽](#)
- [報警連絡人和聯繫組](#)

因為OSS的報警規則是根據OSS監控項設定的，所以類似於OSS監控項的維度分類，將其分成兩個報警維度：用戶層級和Bucket層級。

報警規則頁

報警規則頁是OSS監控報警相關的規則管理頁面，您可以查看、修改、啟用、禁用和刪除對應的報警規則，而且能夠查看該對應報警規則的歷史報警情況。

-
- 單擊對應報警規則的修改，就可以對該報警規則進行修改。
- 單擊對應報警規則的刪除，就可以刪除該報警規則。選中多條報警規則，然後單擊表格最下方的刪除按鈕，可以大量刪除報警規則。
- 如果報警規則處於已啟用狀態，單擊該報警規則的禁用，可以禁用該報警規則，報警規則失效，用戶不能再收到對應的警示資訊。選中多條報警規則，然後單擊表格最下方的禁用按鈕，可以批量禁用報警規則。
- 如果報警規則處於已禁用狀態，單擊該報警規則的啟用，可以啟用該報警規則，報警規則重新生效，能檢測並發出對應的警示資訊。選中多條報警規則，然後單擊表格最下方的啟用按鈕，可以批量啟用報警規則。
- 單擊對應報警規則的報警歷史，可以查看該報警規則歷史發生的所有的警示情況。

□

這裡解釋一下報警歷史的相關概念：

- 報警歷史指的是該報警規則的狀態變化歷史，比如從正常變成警示狀態，是一個狀態變化；從警示變成正常也是狀態變化。另外，還有一個特殊的狀態變化：通道沉默。
- 當通知對象為通道沉默時，表示該報警規則引發警示之後的24小時內一直滿足報警觸發狀態(即一直在警示，沒有恢復到正常狀態)。這個時候系統是不會再向用戶發送警示資訊進行通知的，一直持續到24小時之後才會有新的報警資訊發送到通知對象。
- 報警歷史資訊能夠保存一個月，即一個月之前的警示資訊會被自動清理。查詢時一次最多隻能查詢3天的數據，但不支援查詢31天前的數據。

另外，單擊具體報警規則的通知對象旁邊的查看，可以顯示該通知對象（報警聯繫組）的成員以及每個成員接收警示資訊的方式（SMS、郵箱或者旺旺），如下圖所示：

□

查看報警規則

- 快速定位報警規則

根據報警規則頁中的下面控制項資訊能夠快速定位到被搜索的報警規則：

- 報警維度下拉框：全部和Bucket層級。當選項為全部時，顯示所有用戶層級和Bucket層級的報警規則。
 -
 - Bucket下拉框：當報警維度下拉框為Bucket層級時，這裡可以羅列該帳號下所有的Bucket。選擇對應的Bucket，可以展示屬於該Bucket的所有報警規則。
 -
 - 監控項下拉框：羅列所有的OSS的監控項，包括用戶層級和Bucket層級的監控項。當選項為監控項時，顯示用戶層級或者Bucket層級所有監控項的報警規則。
 - 報警狀態下拉框：報警狀態、警示和正常。當為報警狀態時顯示所有狀態的報警規則。
 - 啟用狀態下拉框：啟用狀態、已禁用和已啟用。當為啟用狀態時，顯示所有狀態的報警規則。
 - 查看全部報警規則
- 單擊報警規則Tab頁進入報警規則也預設顯示全部報警規則。將報警維度下拉框設定為全部，即能顯示全部報警規則。配合監控項、報警狀態和啟用狀態這三個下拉框能夠更精確定位全部維度下的報警規則。
- 查看特定Bucket的報警規則

如果需要查看某個特定的Bucket的報警規則，需要設定報警維度下拉框為Bucket層級，然後在Bucket下拉框中選擇對應的Bucket名稱。通過Bucket列表中的對應Bucket的報警規則進入的報警頁，就能顯示該Bucket的所有的報警規則。配合監控項、報警狀態和啟用狀態這三個，即能顯示該報警維度下更精確的報警規則。

- 查看與具體監控項相關的報警規則
設定監控項下拉框為具體的某項監控項，就能顯示對應監控項的所有報警規則。
- 查看指定報警狀態的報警規則
設定報警狀態下拉框為指定報警狀態，如警示，就能展示所有處於該警示狀態的報警規則。
- 查看指定啟用狀態的報警規則
設定啟動狀態下拉框為指定啟用狀態，如已禁用，就能顯示已經被禁用的所有的報警規則。

添加報警規則

在Bucket列表Tab頁選中指定的Bucket之後單擊設定報警規則按鈕，或者在用戶概況Tab頁亦或某個Bucket監控視圖頁中的監控圖形上的



都可以開啟批量設定報警規則視窗，進行報警規則的設定。

下面示範一下報警規則設定的過程（注意，下面出現的名詞概念請詳見Cloud Monitor[報警服務概覽](#)。）

1. 首先進入如下設定報警規則步驟：

- - 報警維度：提示用戶目前設定的報規則屬於哪個維度監控。如果是Bucket層級，則會提示目前為哪個具體的Bucket設定報警規則，圖例中顯示的是為用戶層級。
 - 監控項：該多選框顯示屬於該報警維度所有的監控項選項。可以通過快速搜尋框迅速定位：
 - 統計周期：間隔多久進行一次數據統計，預設為5min。
 - 連續次數：指連續幾個統計周期監控項的值持續超過閾值後觸發報警。
 - 統計方法：指定該監控項中的具體的統計指標，OSS監控服務的統計方法均為**監控值**。
 - 添加報警規則按鈕：單擊按鈕可以設定更多監控項的報警規則。
 - 刪除：單擊對應報警規則後的刪除，可以刪除該報警規則。

2. 設定完成之後，單擊下一步，進入設定通知對象步驟，如下圖：


- 如果事先已經按照[報警連絡人和聯繫組](#)設定好報警聯繫組，那麼就會像圖例中展示出已經存在的報警連絡人群組。如果沒有設定過報警連絡人群組，則單擊**快建立連絡人群組**，按照提示建立即可。

3. 單擊確定，完成報警規則的設定。

□

● Bucket列表添加報警規則

在Bucket列表Tab頁中可以為多個Bucket大量新增相同的報警規則。選中需要配置報警規則的Bucket，單擊設定報警規則按鈕，進入報警規則設定頁面。

 **說明** 批量設定時報警維度是確定的，監控項都是Bucket層級的。

● 具體監控圖表中添加報警規則

在用戶概況Tab頁或者具體Bucket的監控視圖頁中，單擊具體的監控圖右上方的



可以為該監控圖用的監控指標項設定報警規則。

說明 通過單擊監控圖的報警表徵圖進入報警規則頁面時，報警維度都是確定的，而且只能設定該圖中的監控項的報警規則。

注意事項

目前屬於某個Bucket的報警規則存在性並沒有與該Bucket的存在性強關聯，即如果刪除了某個Bucket，屬於這個Bucket的報警規則依賴存在。建議用戶在刪除Bucket之前先刪除對應的報警規則。

6.4. 訪問監控數據

本章內容為OpenAPI（或者Cloud Monitor提供的SDK）查詢OSS監控服務指標數據的使用提供相關參數依據。

關於Project

OSS監控服務指標項的數據都使用相同的Project名稱：acs_oss。

使用Java SDK設定程式碼範例如下：

```
QueryMetricRequest request = new QueryMetricRequest();
request.setProject("acs_oss");
```

關於StartTime和EndTime

Cloud Monitor的時間參數取值範圍採用左開右閉的形式，即(StartTime, EndTime]，即StartTime的數據不會被查詢到，而EndTime的數據會被查詢到。

另外，Cloud Monitor數據保留時間為31天，設定的StartTime和EndTime的時間間距不能大於31天，並且不能夠查詢31天以前的數據。

其他時間參數資訊詳見Cloud MonitorAPI參考。

使用Java SDK設定程式碼範例如下：

```
request.setStartTime("2016-05-15 08:00:00");
request.setEndTime("2015-05-15 09:00:00");
```

關於Dimensions

OSS監控服務根據用戶使用場景，將監控項分為用戶和Bucket兩個不同的層級進行監控。針對不同的層級監控數據的訪問，Dimensions不同：

- 用戶層級數據不需要設定Dimensions。
- Bucket層級數據的Dimensions設定為：

```
{"BucketName": "your_bucket_name"}
```

其中，your_bucket_name為屬於該用戶的某個需要查詢的Bucket名稱。

注意，Dimenisons是一個JSON字元串，OSS監控指標的Dimensions只有一對Key-Value。

使用Java SDK設定程式碼範例如下：

```
request.setDimensions("{\"BucketName\":\"your_bucket_name\"});
```

關於Period

OSS監控指標除了計量指標，其他所有的指標的彙總粒度均為60s。計量指標的彙總粒度為3600s。

使用Java SDK設定程式碼範例如下：

```
request.setPeriod("60");
```

關於Metric

OSS 監控指標參考中詳細介紹的各項指標項對應的Metric名稱和層級如下表所示：

Metric	對應指標項名稱	單位	層級
UserAvailability	用戶層級可用性	%	用戶層級
UserRequestValidRate	用戶層級有效請求率	%	用戶層級
UserTotalRequestCount	用戶層級總請求數	次數	用戶層級
UserValidRequestCount	用戶層級有效請求數	次數	用戶層級
UserInternetSend	用戶層級公網流出流量	位元組	用戶層級
UserInternetRecv	用戶層級公網流入流量	位元組	用戶層級
UserIntranetSend	用戶層級內網流出流量	位元組	用戶層級
UserIntranetRecv	用戶層級內網流入流量	位元組	用戶層級
UserCdnSend	用戶層級cdn流出流量	位元組	用戶層級
UserCdnRecv	用戶層級cdn流入流量	位元組	用戶層級
UserSyncSend	用戶層級跨區域複製流出流量	位元組	用戶層級
UserSyncRecv	用戶層級跨區域複製流入流量	位元組	用戶層級
UserServerErrorCount	用戶層級服務端錯誤請求總數	次數	用戶層級
UserServerErrorRate	用戶層級服務端錯誤請求佔比	%	用戶層級

Metric	對應指標項名稱	單位	層級
UserNetworkErrorCount	用戶層級網路錯誤請求總數	次數	用戶層級
UserNetworkErrorRate	用戶層級網路錯誤請求佔比	%	用戶層級
UserAuthorizationErrorCount	用戶層級客戶端授權錯誤請求總數	次數	用戶層級
UserAuthorizationErrorRate	用戶層級客戶端授權錯誤請求佔比	%	用戶層級
UserResourceNotFoundErrorCount	用戶層級客戶端資源不存在錯誤請求總數	次數	用戶層級
UserResourceNotFoundErrorRate	用戶層級客戶端資源不存在錯誤請求佔比	%	用戶層級
UserClientTimeoutErrorCount	用戶層級客戶端逾時錯誤請求總數	次數	用戶層級
UserClientOtherErrorRate	用戶層級客戶端逾時錯誤請求佔比	%	用戶層級
UserClientOtherErrorCount	用戶層級客戶端其他錯誤請求總數	次數	用戶層級
UserClientOtherErrorRate	用戶層級客戶端其他錯誤請求佔比	%	用戶層級
UserSuccessCount	用戶層級成功請求總數	次數	用戶層級
UserSuccessRate	用戶層級成功請求佔比	%	用戶層級
UserRedirectCount	用戶層級重新導向請求總數	次數	用戶層級
UserRedirectRate	用戶層級重新導向請求佔比	%	用戶層級
Availability	可用性	%	Bucket層級
RequestValidRate	有效請求率	%	Bucket層級
TotalRequestCount	總請求數	次數	Bucket層級
ValidRequestCount	有效請求數	次數	Bucket層級
InternetSend	公網流出流量	位元組	Bucket層級
InternetRecv	公網流入流量	位元組	Bucket層級

Metric	對應指標項名稱	單位	層級
IntranetSend	內網流出流量	位元組	Bucket層級
IntranetRecv	內網流入流量	位元組	Bucket層級
CdnSend	cdn流出流量	位元組	Bucket層級
CdnRecv	cdn流入流量	位元組	Bucket層級
SyncSend	跨區域複製流出流量	位元組	Bucket層級
SyncRecv	跨區域複製流入流量	位元組	Bucket層級
ServerErrorCount	服務端錯誤請求總數	次數	Bucket層級
ServerErrorRate	服務端錯誤請求佔比	%	Bucket層級
NetworkErrorCount	網路錯誤請求總數	次數	Bucket層級
NetworkErrorRate	網路錯誤請求佔比	%	Bucket層級
AuthorizationErrorCount	客戶端授權錯誤請求總數	次數	Bucket層級
AuthorizationErrorRate	客戶端授權錯誤請求佔比	%	Bucket層級
ResourceNotFoundErrorCode	客戶端資源不存在錯誤請求總數	次數	Bucket層級
ResourceNotFoundErrorCodeRate	客戶端資源不存在錯誤請求佔比	%	Bucket層級
ClientTimeoutErrorCount	客戶端逾時錯誤請求總數	次數	Bucket層級
ClientOtherErrorRate	客戶端逾時錯誤請求佔比	%	Bucket層級
ClientOtherErrorCount	客戶端其他錯誤請求總數	次數	Bucket層級
ClientOtherErrorRate	客戶端其他錯誤請求佔比	%	Bucket層級
SuccessCount	成功請求總數	次數	Bucket層級
SuccessRate	成功請求佔比	%	Bucket層級
RedirectCount	重新導向請求總數	次數	Bucket層級
RedirectRate	重新導向請求佔比	%	Bucket層級
GetObjectE2eLatency	GetObject請求平均E2E延時	毫秒	Bucket層級

Metric	對應指標項名稱	單位	層級
GetObjectServerLatency	GetObject請求平均伺服器延時	毫秒	Bucket層級
MaxGetObjectE2eLatency	GetObject請求最大E2E延時	毫秒	Bucket層級
MaxGetObjectServerLatency	GetObject請求最大伺服器延時	毫秒	Bucket層級
HeadObjectE2eLatency	HeadObject請求平均E2E延時	毫秒	Bucket層級
HeadObjectServerLatency	HeadObject請求平均伺服器延時	毫秒	Bucket層級
MaxHeadObjectE2eLatency	HeadObject請求最大E2E延時	毫秒	Bucket層級
MaxHeadObjectServerLatency	HeadObject請求最大伺服器延時	毫秒	Bucket層級
PutObjectE2eLatency	PutObject請求平均E2E延時	毫秒	Bucket層級
PutObjectServerLatency	PutObject請求平均伺服器延時	毫秒	Bucket層級
MaxPutObjectE2eLatency	PutObject請求最大E2E延時	毫秒	Bucket層級
MaxPutObjectServerLatency	PutObject請求最大伺服器延時	毫秒	Bucket層級
PostObjectE2eLatency	PostObject請求平均E2E延時	毫秒	Bucket層級
PostObjectServerLatency	PostObject請求平均伺服器延時	毫秒	Bucket層級
MaxPostObjectE2eLatency	PostObject請求最大E2E延時	毫秒	Bucket層級
MaxPostObjectServerLatency	PostObject請求最大伺服器延時	毫秒	Bucket層級
AppendObjectE2eLatency	AppendObject請求平均E2E延時	毫秒	Bucket層級
AppendObjectServerLatency	AppendObject請求平均伺服器延時	毫秒	Bucket層級

Metric	對應指標項名稱	單位	層級
MaxAppendObjectE2eLatency	AppendObject請求最大E2E延時	毫秒	Bucket層級
MaxAppendObjectServerLatency	AppendObject請求最大伺服器延時	毫秒	Bucket層級
UploadPartE2eLatency	UploadPart請求平均E2E延時	毫秒	Bucket層級
UploadPartServerLatency	UploadPart請求平均伺服器延時	毫秒	Bucket層級
MaxUploadPartE2eLatency	UploadPart請求最大E2E延時	毫秒	Bucket層級
MaxUploadPartServerLatency	UploadPart請求最大伺服器延時	毫秒	Bucket層級
UploadPartCopyE2eLatency	UploadPartCopy請求平均E2E延時	毫秒	Bucket層級
UploadPartCopyServerLatency	UploadPartCopy請求平均伺服器延時	毫秒	Bucket層級
MaxUploadPartCopyE2eLatency	UploadPartCopy請求最大E2E延時	毫秒	Bucket層級
MaxUploadPartCopyServerLatency	UploadPartCopy請求最大伺服器延時	毫秒	Bucket層級
GetObjectCount	GetObject成功請求數	次數	Bucket層級
HeadObjectCount	HeadObject成功請求數	次數	Bucket層級
PutObjectCount	PutObject成功請求數	次數	Bucket層級
PostObjectCount	PostObject成功請求數	次數	Bucket層級
AppendObjectCount	AppendObject成功請求數	次數	Bucket層級
UploadPartCount	UploadPart成功請求數	次數	Bucket層級
UploadPartCopyCount	UploadPartCopy成功請求數	次數	Bucket層級
DeleteObjectCount	DeleteObject成功請求數	次數	Bucket層級
DeleteObjectsCount	DeleteObjects成功請求數	次數	Bucket層級

計量類指標的Metric如下表所示，注意彙總粒度為3600s。

Metric	對應指標項名稱	單位	層級
MeteringStorageUtilization	儲存大小	位元組	若設定Dimensions，則為Bucket層級；如果不設定，那麼返回的指標為整個用戶層級數據。
MeteringGetRequest	Get類請求數	次數	若設定Dimensions，則為Bucket層級；如果不設定，那麼返回的指標為整個用戶層級數據。
MeteringPutRequest	Put類請求數	次數	若設定Dimensions，則為Bucket層級；如果不設定，那麼返回的指標為整個用戶層級數據。
MeteringInternetTX	公網流出計量流量	位元組	若設定Dimensions，則為Bucket層級；如果不設定，那麼返回的指標為整個用戶層級數據。
MeteringCdnTX	cdn流出計量流量	位元組	若設定Dimensions，則為Bucket層級；如果不設定，那麼返回的指標為整個用戶層級數據。
MeteringSyncRX	跨區域複製流入計量流量	位元組	若設定Dimensions，則為Bucket層級；如果不設定，那麼返回的指標為整個用戶層級數據。

使用Java SDK設定程式碼範例如下：

```
request.setMetric("UserAvailability");
```

6.5. 監控指標參考

根據用戶使用場景，將OSS的指標分為用戶層級和Bucket（儲存空間）層級兩個層級維度進行監控。

另外，為了更好地觀察監控數據以及匹配計費策略，除了一般監控項的時間序指標外，OSS對現有的監控指標項進行統計分析，提供了一段時間內的統計指標，如請求狀態分布統計和當月計量統計。

除了計量指標和統計指標，所有的指標（時間序指標）都是分鐘等級的數據匯總（如求和、求最大值或者求均值等等）。而計量指標是按小時的數據進行匯總的時間序指標。

用戶層級指標

用戶層級指標是指從用戶的賬戶等級對OSS系統使用的總體情況進行監控的指標資訊，是對該賬戶下的所有的Bucket相關監控數據的匯總。其中包括當月計量統計、服務監控總覽和請求狀態詳情三個方面。

- 服務監控總覽

服務監控總覽指標屬於基礎服務指標，具體指標項詳見下表。注意，下面所有的指標都是在用戶層級監控的。

服務監控總覽指標名稱	單位	描述
可用性	%	儲存服務的系統可用性衡量指標。通過公式 $1 - \text{服務端錯誤請求（返回狀態碼為5xx）佔總請求的百分比}$ 而得到
有效請求率	%	有效請求佔總請求數的百分比，有效請求的介紹見下面說明
總請求數	次數	被OSS服務端接收並處理的請求總數
有效請求數	次數	返回狀態碼為2xx和3xx的請求總數
公網流出流量	位元組	通過互連網網路的下行流量
公網流入流量	位元組	通過互連網網路的上行流量
內網流出流量	位元組	通過服務系統內部網路的下行流量
內網流入流量	位元組	通過服務系統內部網路的上行流量
cdn流出流量	位元組	開通cdn加速服務之後，通過cdn產生的下行流量，即回源流量
cdn流入流量	位元組	開通cdn加速服務之後，通過cdn產生的上行流量
跨區域複製流出流量	位元組	開通跨區域複製功能之後，數據複製過程產生的下行流量
跨區域複製流入流量	位元組	開通跨區域複製功能之後，數據複製過程產生的上行流量

除了以上具體的監控指標，還提供一段時間內的請求狀態分布統計，主要是根據返回的狀態碼或者OSS錯誤碼進行分類的請求的統計資訊（被觀察時間段內的請求次數總和以及佔比），相關的監控指標項資訊詳見以下請求狀態詳情的介紹。

- 請求狀態詳情

請求狀態詳情指標是指根據請求返回狀態碼或者OSS錯誤碼進行分類的請求的監控資訊，屬於基礎服務指標，具體指標項詳見下表。注意，下面所有的指標都是在用戶層級監控的。

請求狀態詳情指標名稱	單位	描述
服務端錯誤請求總數	次數	返回狀態碼為5xx的系統級錯誤請求總數
服務端錯誤請求佔比	%	服務端錯誤請求總數佔總請求數的百分比

請求狀態詳情指標名稱	單位	描述
網路錯誤請求總數	次數	HTTP狀態碼為499的請求總數，
網路錯誤請求佔比	%	網路錯誤請求數佔總請求數的百分比
客戶端授權錯誤請求總數	次數	返回狀態碼403的請求總數
客戶端授權錯誤請求佔比	%	授權錯誤請求數佔總請求數的百分比
客戶端資源不存在錯誤請求總數	次數	返回狀態碼為404的請求總數
客戶端資源不存在錯誤請求佔比	%	資源不存在錯誤請求數佔總請求數百分比
客戶端逾時錯誤請求總數	次數	返回狀態碼為408或者返回的OSS錯誤碼為RequestTimeout的請求總數
客戶端逾時錯誤請求佔比	%	網路錯誤請求總數佔總請求數的百分比
客戶端其他錯誤請求總數	次數	除了以上提到的客戶端錯誤請求之外的其他返回狀態碼為4xx的請求總數
客戶端其他錯誤請求佔比	%	客戶端其他錯誤請求數佔總請求數的百分比
成功請求總數	次數	返回狀態碼為2xx的請求總數
成功請求佔比	%	成功請求數佔總請求數的百分比
重新導向請求總數	次數	返回狀態碼為3xx的請求總數
重新導向請求佔比	%	重新導向請求數佔總請求數的百分比

● 當月計量統計

當月計量統計指標是指從當月的1號0點開始，到當月計量採集截止時間為止，這段時間內計量指標的統計數據。

目前統計的計量指標如下：

當月計量統計指標名稱	單位	描述
儲存大小	位元組	在計量採集截止時間前屬於該用戶的所有Bucket佔用的儲存總大小
公網流出流量	位元組	從本月1號0點開始累積到計量採集截止時間為止，用戶所使用的所有公網流出流量的總和。

當月計量統計指標名稱	單位	描述
Put類請求數	次數	從本月1號0點開始累積到計量採集截止時間為止，用戶所使用的所有Put類請求的總和。
Get類請求數	次數	從本月1號0點開始累積到計量採集截止時間為止，用戶所使用的所有Get類請求的總和。

Bucket層級指標

Bucket層級指標是指對具體的儲存空間的OSS操作情況進行監控的指標資訊，具有更強的業務場景，所以除了類似從用戶層面可以監控的服務監控總覽和請求狀態詳情這些基礎服務指標項和當月計量統計之外，還有計量參考、延時和成功請求操作分類等計量指標和性能指標。

- 服務監控總覽

監控項指標含義同用戶層級的服務監控總覽，從具體的Bucket進行監控。

- 請求狀態詳情

監控項指標含義同用戶層級的請求狀態詳情，從具體的Bucket進行監控。

- 當月計量統計

統計方式同用戶層級的當月計量統計，從具體的Bucket資源使用方式進行統計。

當月計量統計指標名稱	單位	描述
儲存大小	位元組	在計量採集截止時間前該Bucket佔用的儲存大小
公網流出流量	位元組	從本月1號0點開始累積到計量採集截止時間為止，該Bucket的公網流出流量的總和。
Put類請求數	次數	從本月1號0點開始累積到計量採集截止時間為止，該Bucket的所有Put類請求的總和。
Get類請求數	次數	從本月1號0點開始累積到計量採集截止時間為止，該Bucket的所有Get類請求的總和。

- 計量參考

計量指標的時間序監控，具體如下：

當月計量統計指標名稱	單位	描述
儲存大小	位元組	該Bucket每小時使用的平均儲存大小
公網流出流量	位元組	該Bucket每小時的公網流出流量的總和

當月計量統計指標名稱	單位	描述
Put類請求數	次數	該Bucket每小時的Put類請求的總和
Get類請求數	次數	該Bucket每小時的Get類請求的總和

● 延時

請求延時是系統性能的直觀反映。監控服務提供了分鐘等級的平均延時和最大延時兩類指標，分別反映系統平均響應能力和系統抖動情況。並且根據OSS API請求操作類型進行分類，更細粒度地反應系統應對不同操作的性能狀況。目前只對關於Bucket的操作並且涉及數據操作（不包含對meta操作）的API進行監控。

另外，延時監控指標分別從E2E和伺服器兩條不同的鏈路進行收集，便於分析性能熱點以及環境問題，其中：

- E2E延時是指向OSS系統發出的成功請求的端到端延隔時間，包括在OSS系統中讀取請求、發送響應以及接受響應確認所需的處理時間。
- 伺服器延時是指OSS系統成功處理請求所使用的延隔時間，不包括E2E延時中的網路延隔時間。

注意，性能相關指標都是對成功請求（返回狀態碼為2xx）進行的監控。

具體的監控指標項如下表：

延時指標名稱	單位	描述
GetObject請求平均E2E延時	毫秒	請求API為GetObject的成功請求的平均端到端延時
GetObject請求平均伺服器延時	毫秒	請求API為GetObject的成功請求的平均伺服器延時
GetObject請求最大E2E延時	毫秒	請求API為GetObject的成功請求的最大端到端延時
GetObject請求最大伺服器延時	毫秒	請求API為GetObject的成功請求的最大伺服器延時
HeadObject請求平均E2E延時	毫秒	請求API為HeadObject的成功請求的平均端到端延時
HeadObject請求平均伺服器延時	毫秒	請求API為HeadObject的成功請求的平均伺服器延時
HeadObject請求最大E2E延時	毫秒	請求API為HeadObject的成功請求的最大端到端延時
HeadObject請求最大伺服器延時	毫秒	請求API為HeadObject的成功請求的最大伺服器延時
PutObject請求平均E2E延時	毫秒	請求API為PutObject的成功請求的平均端到端延時

延時指標名稱	單位	描述
PutObject請求平均伺服器延時	毫秒	請求API為PutObject的成功請求的平均伺服器延時
PutObject請求最大E2E延時	毫秒	請求API為PutObject的成功請求的最大端到端延時
PutObject請求最大伺服器延時	毫秒	請求API為PutObject的成功請求的最大伺服器延時
PostObject請求平均E2E延時	毫秒	請求API為PostObject的成功請求的平均端到端延時
PostObject請求平均伺服器延時	毫秒	請求API為PostObject的成功請求的平均伺服器延時
PostObject請求最大E2E延時	毫秒	請求API為PostObject的成功請求的最大端到端延時
PostObject請求最大伺服器延時	毫秒	請求API為PostObject的成功請求的最大伺服器延時
AppendObject請求平均E2E延時	毫秒	請求API為AppendObject的成功請求的平均端到端延時
AppendObject請求平均伺服器延時	毫秒	請求API為AppendObject的成功請求的平均伺服器延時
AppendObject請求最大E2E延時	毫秒	請求API為AppendObject的成功請求的最大端到端延時
AppendObject請求最大伺服器延時	毫秒	請求API為AppendObject的成功請求的最大伺服器延時
UploadPart請求平均E2E延時	毫秒	請求API為UploadPart的成功請求的平均端到端延時
UploadPart請求平均伺服器延時	毫秒	請求API為UploadPart的成功請求的平均伺服器延時
UploadPart請求最大E2E延時	毫秒	請求API為UploadPart的成功請求的最大端到端延時
UploadPart請求最大伺服器延時	毫秒	請求API為UploadPart的成功請求的最大伺服器延時
UploadPartCopy請求平均E2E延時	毫秒	請求API為UploadPartCopy的成功請求的平均端到端延時
UploadPartCopy請求平均伺服器延時	毫秒	請求API為UploadPartCopy的成功請求的平均伺服器延時

延時指標名稱	單位	描述
UploadPartCopy請求最大E2E延時	毫秒	請求API為UploadPartCopy的成功請求的最大端到端延時
UploadPartCopy請求最大伺服器延時	毫秒	請求API為UploadPartCopy的成功請求的最大伺服器延時

- 成功請求操作分類

配合延時監控，成功請求的監控一定程度上反應了系統處理訪問請求的能力。目前只監控關於Bucket的操作中涉及數據操作的API。詳細的指標項如下：

成功請求操作分類指標名稱	單位	描述
GetObject成功請求數	次數	請求API為GetObject的成功請求數
HeadObject成功請求數	次數	請求API為HeadObject的成功請求數
PutObject成功請求數	次數	請求API為PutObject的成功請求數
PostObject成功請求數	次數	請求API為PostObject的成功請求數
AppendObject成功請求數	次數	請求API為AppendObject的成功請求數
UploadPart成功請求數	次數	請求API為UploadPart的成功請求數
UploadPartCopy成功請求數	次數	請求API為UploadPartCopy的成功請求數
DeleteObject成功請求數	次數	請求API為DeleteObject的成功請求數
DeleteObjects成功請求數	次數	請求API為DeleteObjects的成功請求數

6.6. 監控、診斷和故障排除

相對於傳統應用程式，開發雲端應用雖然降低了用戶在基礎設施搭建、運維等方面的成本，但卻增大了監控、診斷和故障排查的難度。OSS儲存服務為您提供了豐富的監控和日誌資訊，幫助您深刻洞察程式行為，及時發現並快速定位問題。

概述

本文主要描述如何使用OSS監控服務、日誌記錄功能以及其他第三方工具來監控、診斷和排查應用業務使用OSS儲存服務時遇到的相關問題，幫助您達到如下目標：

- 即時監控OSS儲存服務的健全狀態和性能，並及時報警通知。
- 獲取有效方法和工具來定位問題。

- 根據相關問題的處理和操作指南，快速解決與OSS相關的問題。

本文包括如下內容：

- **服務監控**，介紹如何使用OSS監控服務持續監控OSS儲存服務的健全狀態和性能。
- **跟蹤診斷**，介紹如何使用OSS監控服務和logging記錄功能診斷問題；另外，還介紹如何關聯各種記錄檔中的相關資訊進行跟蹤診斷。
- **故障排除**，提供常見的問題場景和故障排除方法。

服務監控

監視總體健全狀態

- 可用性和有效請求率

可用性和有效請求率是有關系統穩定性和用戶是否正確使用系統的最重要指標，指標小於100%說明某些請求失敗。

可能因為一些系統優化因素出現暫時性的低於100%，比如為了負載平衡而出現分區遷移，此時OSS的SDK能夠提供相關的重試機制無縫處理這類間歇性的失敗情況，使得業務端無感知。

另外，對於有效請求率低於100%的情況，用戶需要根據自己的使用方式進行分析，可以通過請求分布統計或者請求狀態詳情確定錯誤請求的具體類型，**跟蹤診斷**確定原因，並**故障排除**。當然，對於一些業務場景，出現有效請求率低於100%是符合預期的，比如，用戶需要先check訪問的object是否存在，然後根據object的存在性採取一定的操作，如果object不存在，檢測object存在性的讀取請求必然會收到404錯誤（資源不存在錯誤），那麼必然會導致該指標項低於100%。

對於系統可用性指標要求較高的業務，可以設定其報警規則，當低於預期閾值時，進行報警。

- 總請求數和有效請求數

該指標從總訪問量角度來反應系統運行狀態。當有效請求數不等於總請求數時表明某些請求失敗。

用戶可以關注總請求數或者有效請求數的波動狀況，特別是突然上升或者下降的情況，需要進行跟進調查，可以通過設定報警規則進行及時通知，對於存在周期性狀況的業務，可以設定為環比報警規則（環比報警方式即將上線），詳見**報警服務使用指南**。

- 請求狀態分布統計

當可用性或者有效請求率低於100%（或者有效請求數不等於總請求數時），可以通過查看請求狀態分布統計快速確定請求的錯誤類型，該統計監控指標的詳細介紹參見**OSS監控指標參考手冊**。

監視請求狀態詳情

請求狀態詳情在請求狀態分布統計的基礎上進一步細化請求監控狀態，可以更加深入、具體地監視某類請求。

監視性能

監控服務提供了以下幾個監控項用來監控性能相關的指標。

- 平均延時，包括E2E平均延時和伺服器平均延時
- 最大延時，包括E2E最大延時和伺服器最大延時
- 成功請求操作分類

- ◻
- 流量

- ◻ 以上各個監控項（除流量）都分別從API操作類型進行分類監控：

- GetObject
- HeadObject
- PutObject
- PostObject
- AppendObject
- UploadPart
- UploadPartCopy

延時指標顯示API操作類型處理請求所需的平均和最大時間。其中E2E延時是端到端延遲指標，除了包括處理請求所需的時間外，還包括讀取請求和發送響應所需的時間以及請求在網路上傳輸的延時；而伺服器延時只是請求在伺服器端被處理的時間，不包括與客戶端通訊的網路延時。所以當出現E2E延時突然升高的情況下，如果伺服器延時並沒有很大的變化，那麼可以判定是網路的不穩定因素造成的性能問題，排除OSS系統內部故障。

成功請求操作分類除了以上提到的API之外，還提供以下兩個API操作類型請求數量的監控：

- DeleteObject
- DeleteObjects

流量指標從用戶或者具體的Bucket層級的總體情況進行監控，關注公網、內網、cdn回源以及跨域複製等場景中的網路資源佔用狀況。

對於性能類指標項，需要重點關注其突發的異常變化，例如，平均延時突然出現尖峰，或者長時間處於高出正常請求延時的基準上方等等。用戶可以通過對性能指標設定對應的報警規則，對於低於或者超過閾值時及時通知到相關人員。對於有業務周期性的業務，可以設定環比報警規則(環比一周、環比一天或者環比一個小時，環比報警方式即將上線)。

監視計量

撰寫本文檔時，OSS監控服務只支援監控儲存區大小、公網流出流量、Put類請求數和Get類請求數（不包括跨區域複製流出流量和cdn流出流量），而且不支援對計量數據的報警設定和OpenAPI的讀取。

OSS監控服務對Bucket層級的計量監控數據進行小時等級的粒度採集，可以在具體的Bucket監控視圖中查看其持續的監控趨勢圖。用戶可以根據監控視圖分析其業務的儲存服務資源使用趨勢並且進行成本的預估等。如下所示：

- ◻ OSS監控服務還提供了用戶層級和Bucket層級兩個不同維度當月消費的資源數統計。即從當月1號開始到目前的賬戶或者某個Bucket所消耗的OSS資源總量，每小時更新，幫助用戶即時了解本月使用資源的情況並計算消費。

OSS的消費計算請參考[計量項和計費項](#)。

② 說明 監控服務中提供的計量數據是盡最大可能推送的，可能會與實際消費賬單中產生差異，請以費用中心的數據為準。

跟蹤診斷

問題診斷

- 診斷性能

對於應用程式的性能判斷多帶有主觀因素，用戶需要根據具體的業務場景確定滿足業務需求的基準線，來判定性能問題。另外，從客戶端發起的請求，能引起性能問題的因素貫穿整個請求鏈路。比如OSS儲存服務load過大、客戶端tcp配置問題或網路基礎結構中存在流量瓶頸等。

因此診斷性能問題首先需要設定合理的基準線，然後通過監控服務提供的性能指標確定性能問題可能的根源位置，然後根據日誌查到詳細的資訊以便進一步診斷並且排除故障。

在下文[故障排除](#)中例舉了常見的性能問題和排查措施，可以參考。

- 診斷錯誤

用戶端應用程式會在請求發生錯誤時接收到服務端返回的相關錯誤資訊，監控服務也會記錄並顯示各種錯誤類型請求的計數和佔比。用戶也可以通過檢查伺服器端日誌、客戶端日誌和部落格來獲取相關單個請求的詳細資料。通常，響應中返回的HTTP狀態碼和OSS錯誤碼以及OSS錯誤資訊都指示請求失敗的原因。

相關錯誤響應資訊請參考 [OSS錯誤響應](#)。

- 使用Logging功能

OSS儲存服務為用戶的請求提供服務端日誌記錄功能，能幫助用戶記錄端到端的詳細請求日誌跟蹤。

有關如何開啟並使用logging功能，請參考[設定日誌](#)。

有關Log Service的命名規則以及記錄格式等詳細資料，請參見[設定訪問日誌記錄](#)。

- 使用部落格記錄工具

在許多情況下，通過Logging功能記錄的儲存日誌和用戶端應用程式的日誌數據已足以診斷問題，但在某些情況下，可能需要更詳細的資訊，這時需要使用部落格記錄工具。

捕獲客戶端和伺服器之間的流量，可以更詳細地獲取客戶端和伺服器之間交換的數據以及底層網路狀況的詳細資料，幫助問題的調查。例如，在某些情況下，用戶請求可能會報告一個錯誤，而伺服器端日誌中卻看不到任何該請求的訪問情況，這時就可以使用OSS的Log Service功能記錄的日誌來調查該問題的原因是否出在客戶端上，或者使用網路監視工具來調查網路問題。

最常用的部落格分析工具之一是[Wireshark](#)。該免費的協議分析器運行在數據包等級，能夠查看各種網路通訊協定的詳細數據包資訊，從而可以排查丟失的數據包和串連問題。

更詳細的WireShark操作請參見[WireShark用戶指南](#)。

E2E跟蹤診斷

請求從客戶端應用進程發起，通過網路環境，進入OSS服務端被處理，然後響應從服務端回到網路環境，被客戶端接收。整個過程，是一個端到端的跟蹤過程。關聯用戶端應用程式日誌、網路追蹤記錄檔和服務端日誌，有助於排查問題的詳細資料根源，發現潛在的問題。

在OSS儲存服務中，提供了RequestID作為關聯各處日誌資訊的標識符。另外，通過日誌的時間戳記，不僅可以迅速尋找和定位日誌範圍，還能夠了解在請求發生時間點範圍內，客戶端應用、網路或者服務系統發生的其他事件，有利於問題的分析和調查。

- RequestID

OSS服務始會為接收的每個請求分配唯一的伺服器請求ID，即RequestID。在不同的日誌中，RequestID位於不同的欄位中：

- 在OSS logging功能記錄的服務端日誌記錄中，RequestID出現在“Request ID”列中。
- 在網路跟蹤（如WireShark捕獲的資料流）中，RequestID作為x-oss-request-id標題值出現在響應消息中。
- 在客戶端應用中，需要客戶端code實現的時候將請求的RequestID自行列印到客戶端日誌中。在攥寫本文的時候，最新版本的JAVA SDK已經支援列印正常請求的RequestID資訊，可以通過各個API介面返回的Result結果的getRequestId這個方法獲取。OSS的各個版本SDK都支援列印出異常請求的RequestID，可以通過調用OSSException的getRequestId方法獲取。

- 時間戳記

使用時間戳來尋找相關日誌項，需要注意客戶端和伺服器之間可能存在的時間偏差。在客戶端上基於時間戳記搜索logging功能記錄的伺服器端日誌條目時，應加/減15分鐘。

故障排除

性能相關常見問題

- 平均E2E延時高，而平均服務端延時低

前面介紹了平均E2E延時與平均伺服器延時的區別。所以產生高E2E延時、低伺服器延時可能的原因有兩個：

- 用戶端應用程式響應慢。
- 網路原因導致。

用戶端應用程式響應速度慢的可能原因包括：

- 可用串連數或可用線程數有限。
 - 對於可用串連數問題，可以使用相關命令確定系統是否存在大量TIME_WAIT狀態的串連。如果是，可以通過調整核心參數解決。
 - 對於可用線程數有限，可以先查看客戶端CPU、記憶體、網路等資源是否已經存在瓶頸，如果沒有，適當調大並發線程數。
 - 如果還不能解決問題，那麼就需要通過優化客戶端代碼，比如，使用非同步訪問方式等。也可以使用性能分析功能分析用戶端應用程式熱點，然後具體優化。
- CPU、記憶體或網路頻寬等資源不足。
 - 對於這類問題，需要先使用相關系統的資源監控查看客戶端具體的資源瓶頸在哪裡，然後通過優化代碼使其對資源的使用更為合理，或者擴容客戶端資源（使用更多的核心或者記憶體）。

網路延遲的可能原因是：

通常，因網路導致的端到端高延遲是由暫時狀況導致的。可以使用Wireshark調查臨時和持久網路問題，例如數據包丟失問題。

- 平均E2E延時低，平均服務端延時低，但客戶端請求延時高

客戶端出現請求延時高的情況，最可能的原因是請求還未達到服務端就出現了延時。所以應該調查來自客戶端的請求為什麼未到達伺服器。

對於客戶端延遲發送請求，可能的客戶端的原因有兩個：

- 可用串連數或可用線程數有限，在上面章節已經描述過解決方案。

- 客戶端請求出現多次重試，如果遇到這種情況，需要根據重試資訊具體調查重試的原因再解決。可以通過下面方式確定客戶端是否出現重試：
 - 檢查客戶端日誌。詳細日誌記錄會指示重試已發生過。以OSS的JAVA SDK為例，可以搜索如下日誌提示，warn或者info的等級。如果存在該日誌，說明可能出現了重試。

```
[Server]Unable to execute HTTP request:
或者
[Client]Unable to execute HTTP request:
```

- 如果客戶端的記錄層級為debug，以OSS的JAVA SDK為例，可以搜索如下日誌，如果存在，那麼肯定出現過重試。

```
Retrying on
```

如果客戶端沒有問題，則應調查潛在的網路問題，例如數據包丟失。可以使用工具（如 Wireshark）調查網路問題。

- 平均服務端延時高

對於下載或者上傳出現服務端高延時的情況，可能的原因有2個：

- 大量客戶端頻繁訪問同一個小Object。

這種情況，可以通過查看logging功能記錄的服務端日誌資訊來確定是否在一段時間內，某個或某組小Object被頻繁訪問。

對於下載場景，建議用戶為該Bucket開通CDN服務，利用CDN來提升性能，並且可以節約流量費用；對於上傳場景，用戶可以考慮在不影響業務需求的情況下，收回Object或Bucket的寫存取權限。
- 系統內部因素。

對於系統內部問題或者不能通過優化方式解決的問題，請提供客戶端日誌或者Logging功能記錄的日誌資訊中的RequestID，聯繫系統工作人員協助解決。

服務端錯誤問題

對於服務端錯誤的增加，可以分為兩個場景考慮：

- 暫時性的增加

對於這一類問題，用戶需要調整用戶端程式中的重試策略，採用合理的退讓機制，例如指數退避。這樣不僅可以有效避免因為優化或者升級等系統操作（如為了系統負載平衡進行分區遷移等）暫時導致的服務不可用問題，還可以避開業務峰值的壓力。
- 永久性的增加

如果服務端錯誤持續在一個較高的水平，那麼請提供客戶端日誌或者Logging功能記錄的RequestID，聯繫後台工作人員協助調查。

網路錯誤問題

網路錯誤是指服務端正在處理請求時，串連在非伺服器端斷開而不及返回HTTP要求標頭的情況。此時系統會記錄該請求的HTTP狀態碼為499。以下幾種情況會導致伺服器記錄請求的狀態碼變為499：

- 伺服器在收到讀寫請求處理之前，會檢查串連是否可用，不可用則為499。
- 伺服器正在處理請求時，客戶端提前關閉了串連，此時請求被記錄為499。

總之，在請求過程中，客戶端主動關閉請求或者客戶端網路斷掉都會產生網路錯誤。對於客戶端主動關閉請求的情況，需要調查客戶端中的代碼，了解客戶端斷開與儲存服務串連的原因和時間。對於客戶端網路斷掉的情況，用戶可以使用工具（如Wireshark）調查網路連接問題。

客戶端錯誤問題

● 客戶端授權錯誤請求增加

當監控中的客戶端授權錯誤請求數增加，或者用戶端程式接收到大量的403請求錯誤，那麼最常見的可能原因有如下幾個：

- 用戶訪問的Bucket網域名稱不正確。
 - 如果用戶直接用第三層網域名或者次層網域訪問，那麼可能的原因就是用戶的Bucket並不屬於該網域名稱所指示的region內，比如，用戶建立的Bucket的地域為杭州，但是訪問的網域名稱卻為Bucket.oss-cn-shanghai.aliyuncs.com。這時需要確認Bucket的所屬區域，然後更正網域名稱資訊。
 - 如果用戶開啟了cdn加速服務，那麼可能的原因是cdn綁定的回源網域名稱錯了，請檢查cdn回源網域名稱是否為用戶Bucket的第三層網域名。
- 如果用戶使用javascript客戶端遇到403錯誤，可能的原因就是CORS（跨域資源共用）的設定問題，因為Web瀏覽器實施了“同源策略”的安全限制。用戶需要先檢查所屬Bucket的CORS設定是否正確，並進行相應的更正。設定CORS參考[跨域資源共用](#)。
- 存取控制問題，可以分為下面四種：
 - 用戶使用主AK訪問，那麼用戶需要檢查是否AK設定出錯，使用了無效AK。
 - 用戶使用RAM子帳號訪問，那麼用戶需要確定RAM子帳號是否使用了正確的子AK，或者對應子帳號的相關操作是否已經授權。
 - 用戶使用STS臨時Token訪問，那麼用戶需要確認一下這個臨時Token是否已經過期。如果過期，需要重新申請。
 - 如果Bucket或者Object設定了存取控制，這個時候需要查看用戶所訪問的Bucket或者Object是否支援相關的操作。
- 授權第三方下載，即用戶使用簽名URL進行OSS資源訪問，如果之前訪問正常而突然遇到403錯誤，最大可能的原因是URL已經過期。
- RAM子帳號使用OSS周邊工具的情況也會出現403錯誤。這類周邊工具如ossftp、ossbrowser、OSS控制台客戶端等，在填寫相關的AK資訊登入時就拋出錯誤，此時如果您的AK是正確填寫的，那麼您需要查看使用的AK是否為子帳號AK，該子帳號是否有GetService等操作的授權等。

● 客戶端資源不存在錯誤請求增加

客戶端收到404錯誤說明用戶試圖訪問不存在的資源資訊。當看到監控服務上資源不存在錯誤請求增加，那麼最大可能是以下問題導致的：

- 用戶的業務使用方式，比如，用戶需要先檢查Object是否存在來進行下一步動作，這時他會調用doesObjectExist（以JAVA SDK為例）方法，如果Object不存在，在客戶端則收到false值，但是這時在伺服器端實際上會產生一個404的請求資訊。所以，這種業務場景下，出現404是正常的業務行為。
- 客戶端或其他進程以前刪除了該對象。這種情況可以通過搜索logging功能記錄的服務端日誌資訊中的相關對象操作即可確認。

- 網路故障引起丟包重試。舉個例子，客戶端發起一個刪除操作刪除某個Object，此時請求達到服務端，執行刪除成功，但是響應在網路環境中丟包，然後客戶端發起重試，第二次的刪除操作可能就會遇到404錯誤。這種由於網路問題引起的404錯誤可以通過客戶端日誌和服務端日誌確定：
 - 查看客戶端應用日誌是否出現重試請求。
 - 查看服務端日誌是否對該Object有兩次刪除操作，前一次的刪除操作HTTP Status為2xx。
 - 有效請求率低且客戶端其他錯誤請求數高
- 有效請求率為請求返回的HTTP狀態碼為2xx/3xx的請求數佔總請求的比例。狀態碼為4XX和5XX範圍內的請求將計為失敗並降低該比例。客戶端其他錯誤請求是指除服務端錯誤請求（5xx）、網路錯誤請求（499）、客戶端授權錯誤請求（403）、客戶端資源不存在錯誤請求（404）和客戶端逾時錯誤請求（408或者OSS錯誤碼為RequestTimeout的400請求）這些錯誤請求之外的請求。

可以通過查看Logging功能記錄的服務端日誌確定這些錯誤的具體類型，可以在OSS錯誤響應上找到儲存服務返回的常見錯誤碼的列表，然後檢查客戶端代碼中尋找具體原因進行解決。

儲存容量異常增加

儲存容量異常增加，如果不是上傳類請求量增多，一般常見的原因應該是清理操作出現了問題，可以根據下面兩個方面進行調查：

- 客戶端應用使用特定的進程定期清理來釋放空間。針對這種請求的調查步驟是：
 - i. 查看有效請求率指標是否下降，因為失敗的刪除請求會導致清理操作沒能按預期完成。
 - ii. 定位請求有效率降低的具體原因，查看具體是什麼錯誤類型的請求導致。然後還可以結合具體的客戶端日誌定位更詳細的錯誤資訊（例如，用於釋放空間的STS臨時Token已過期）。
- 客戶端通過設定LifeCycle來清理空間：針對這種請求，需要通過控制台或者API介面查看目前Bucket的LifeCycle是否為之前設定的預期值。如果不是，可以直接更正目前配置；進一步的調查可以通過Logging功能記錄的服務端日誌記錄查詢以前修改的具體資訊。如果LifeCycle是正常的，但是卻沒有生效，請聯繫OSS系統管理員協助調查。

其他儲存服務問題

如果前面的故障排除章節未包括您遇到的儲存服務問題，則可以採用以下方法來診斷和排查您的問題：

1. 查看OSS監控服務，了解與預期的基準行為相比是否存在任何更改。監控視圖可能能夠確定此問題是暫時的還是永久性的，並可確定此問題影響哪些儲存操作。
2. 使用監控資訊來幫助您搜索Logging功能記錄的服務端日誌數據，獲取相關時間點發生的任何錯誤資訊。此資訊可能會幫助您排查和解決該問題。
3. 如果伺服器端日誌中的資訊不足以成功排查此問題，則可以使用客戶端日誌來調查用戶端應用程式，或者配合使用網路工具（Wireshark等）調查您的網路問題。

7. 雲端資料處理

圖片服務(IMG)

請參見 [圖片處理指南](#)。

ApsaraVideo for Media Processing

ApsaraVideo for Media Processing 是為多媒體數據提供的轉碼計算服務。它以經濟、易用、彈性和高可擴充的音視頻轉換方法，幫助您將儲存於OSS的音視頻轉碼成適合在PC、TV以及移動終端上播放的格式。

ApsaraVideo for Media Processing 基於阿里雲雲端運算服務構建，它改變了以往進行轉碼時需要購買、搭建、管理轉碼軟硬體的高昂投入以及配置優化、轉碼參數適配等複雜性問題。同時，藉助雲端運算服務的Auto Scaling特性，可以按需提供轉碼能力，從而最大限度的滿足業務轉碼需求、避免資源浪費。

ApsaraVideo for Media Processing 功能包括Web管理主控台、服務API和軟體開發套件。您可以通過它們使用和管理ApsaraVideo for Media Processing 服務，也可以將ApsaraVideo for Media Processing 功能整合到您自己的應用和服務中。

ApsaraVideo for Media Processing 功能包括：

- 轉碼
- 管道
- 截圖
- 媒體資訊
- 浮水印
- 預置模版
- 自訂模版
- 剪輯輸出
- 解析度按比例縮放
- M3U8輸出自訂切片時長
- 音視頻抽取
- 視頻畫面旋轉
- 視頻轉GIF

更多功能及詳細介紹請參見 [ApsaraVideo for Media Processing 文檔](#)。

8. 錯誤處理

9.OSS錯誤響應

當用戶訪問OSS出現錯誤時，OSS會返回給用戶相應的錯誤碼和錯誤資訊，便於用戶定位問題，並做出適當的處理。

OSS的錯誤響應格式

當用戶訪問OSS出錯時，OSS會返回給用戶一個合適的3xx，4xx或者5xx的HTTP狀態碼；以及一個application/xml格式的消息體。

錯誤響應的消息體例子：

```
<?xml version="1.0" ?>
<Error xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
  <Code>
    AccessDenied
  </Code>
  <Message>
    Query-string authentication requires the Signature, Expires and OSSAccessKeyId parameters
  </Message>
  <RequestId>
    1D842BC5425544BB
  </RequestId>
  <HostId>
    oss-cn-hangzhou.aliyuncs.com
  </HostId>
</Error>
```

所有錯誤的消息體中都包括以下幾個元素：

- Code：OSS返回給用戶的錯誤碼。
- Message：OSS給出的詳細錯誤資訊。
- RequestId：用於唯一標識該次請求的UUID；當你無法解決問題時，可以憑這個RequestId來請求OSS開發工程師的幫助。
- HostId：用於標識訪問的OSS叢集，與用戶請求時使用的Host一致。

其他特殊的錯誤資訊元素請參照每個請求的具體介紹。

OSS的錯誤碼

OSS的錯誤碼列表如下：

錯誤碼	描述	HTTP狀態碼	說明
AccessDenied	拒絕訪問	403	原因及排除請參看 許可權問題及排查

錯誤碼	描述	HTTP狀態碼	說明
BucketAlreadyExists	Bucket已經存在	409	CreateBucket 指定的BucketName已經使用，請選擇新的BucketName
BucketNotEmpty	Bucket不為空	409	DeleteBucket 前請先刪除檔案和未完成的分區上傳任務
CallbackFailed	上傳回調失敗	203	原因及排除請參看 上傳回調錯誤及排除
EntityTooLarge	實體過大	400	Post請求消息長度超過5GB，原因及排除請參看 PostObject錯誤及排查
EntityTooSmall	實體過小	400	Post請求消息長度太短，排除請參看 PostObject錯誤及排查
FieldItemTooLong	Post請求中表單域過大	400	除了 file 的表單域大小不要超過4KB，排除請參看 PostObject錯誤及排查
FilePartIntegrity	檔案Part已改變	400	讀分區數據時探索資料與校驗和不符
FilePartNotExist	檔案Part不存在	400	CompleteMultipartUpload 提交的分區沒有上傳
FilePartStale	檔案Part過時	400	讀分區數據時探索資料與長度不符
IncorrectNumberOfFilesInPOSTRequest	Post請求中檔案個數非法	400	Post請求表單域中只能有一個 file 域，排除請參看 PostObject錯誤及排查
InvalidArgument	參數格式錯誤	400	參數格式不符合要求，請對照相應API
InvalidAccessKeyId	AccessKeyId不存在	403	AccessKeyId無效或過期，排除請參看 403錯誤及排查
InvalidBucketName	無效的Bucket名字	400	Bucket命名規則請參看 開發人員指南
InvalidDigest	無效的摘要	400	指定的MD5校驗值與檔案不符，MD5的計算方法請參見 PutObject

錯誤碼	描述	HTTP狀態碼	說明
InvalidEncryptionAlgorithmError	指定的熵編碼密碼編譯演算法錯誤	400	目前只支援 AES256 密碼編譯演算法，詳見PutObject
InvalidObjectName	無效的Object名字	400	ObjectName命名規則請參看開發人員指南
InvalidPart	無效的Part	400	CompleteMultipartUpload提交的Part無效， PartNumber 或 ETag 錯誤
InvalidPartOrder	無效的part順序	400	CompleteMultipartUpload提交的Part需按照 PartNumber 升序排列
InvalidPolicyDocument	無效的Policy文檔	400	Post請求中 Policy 無效，排除請參看PostObject錯誤及排查
InvalidTargetBucketForLogging	Logging操作中有無效的目標bucket	400	存放Logging的目標bucket不存在，請更換
InternalServerError	OSS內部發生錯誤	500	請重試
MalformedXML	XML格式非法	400	請求中 XML 非法，請根據具體請求排除DeleteObjects、CompleteMultipartUpload、PutBucketLogging、PutBucketWebsite、PutBucketLifecycle、PutBucketReferer、PutBucketCORS
MalformedPOSTRequest	Post請求的body格式非法	400	表單域格式非法，排除請參看PostObject錯誤及排查
MaxPOSTPreDataLengthExceededError	Post請求上傳檔案內容之外的body過大	400	除了 file 的表單域大小不要超過4KB，排除請參看PostObject錯誤及排查
MethodNotAllowed	不支援的方法	405	以OSS不支援的操作來訪問資源
MissingArgument	缺少參數	411	請參看具體的API對照解決

錯誤碼	描述	HTTP狀態碼	說明
MissingContentLength	缺少內容長度	411	消息即非chunked encoding又沒有攜帶 Content-Length
NoSuchBucket	Bucket不存在	404	
NoSuchKey	Object不存在	404	
NoSuchUpload	Multipart Upload ID不存在	404	沒有初始化分區上傳或者初始化的分區上傳過期
NotImplemented	無法處理的方法	400	OSS不支援的操作
ObjectNotAppendable	不是可追加檔案	409	OSS有三類檔案normal、appendable、multipart，只有appendable類型的檔案才能執行AppendObject
PositionNotEqualToLength	Append的位置和檔案長度不相等	409	詳見AppendObject
PreconditionFailed	預先處理錯誤	412	下載條件不符合，詳見GetObject
RequestTimeTooSkewed	發起請求的時間和伺服器時間超出15分鐘	403	排除請參看403錯誤及排查
RequestTimeout	請求逾時	400	請重試
RequestIsNotMultiPartContent	Post請求content-type非法	400	排除請參看PostObject錯誤及排查
DownloadTrafficRateLimitExceeded	下載流量超過限制	503	預設上限是5Gbps，包括內外網，有調整需求請提交工單
UploadTrafficRateLimitExceeded	上傳流量超過限制	503	預設上限是5Gbps，包括內外網，有調整需求請提交工單
SignatureDoesNotMatch	簽名錯誤	403	排除請參看Header中簽名、URL中簽名
TooManyBuckets	Bucket數目超過限制	400	預設上限是10，有調整需求請提交工單

OSS常見錯誤及排除

- [上傳回調錯誤及排除](#)
- [403錯誤及排查](#)
- [PostObject錯誤及排查](#)
- [許可權問題及排查](#)
- [跨域資源共用CORS錯誤及排除](#)
- [防盜鏈Referer配置及錯誤排除](#)
- [STS常見問題及排查](#)

SDK/Tool常見錯誤及排除

- [Java SDK](#)
- [Python SDK](#)
- [ossfs](#)
- [ossftp](#)

OSS不支援的操作


如果試圖以OSS不支援的操作來訪問某個資源，返回405 Method Not Allowed錯誤。

錯誤請求樣本：

```
ABC /1.txt HTTP/1.1
Host: bucketname.oss-cn-shanghai.aliyuncs.com
Date: Thu, 11 Aug 2016 03:53:40 GMT
Authorization: signatureValue
```

返回樣本：

```
HTTP/1.1 405 Method Not Allowed
Server: AliyunOSS
Date: Thu, 11 Aug 2016 03:53:44 GMT
Content-Type: application/xml
Content-Length: 338
Connection: keep-alive
x-oss-request-id: 57ABF6C8BC4D25D86CBA5ADE
Allow: GET DELETE HEAD PUT POST OPTIONS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>MethodNotAllowed</Code>
  <Message>The specified method is not allowed against this resource.</Message>
  <RequestId>57ABF6C8BC4D25D86CBA5ADE</RequestId>
  <HostId>bucketname.oss-cn-shanghai.aliyuncs.com</HostId>
  <Method>abc</Method>
  <ResourceType>Bucket</ResourceType>
</Error>
```

 說明 如果訪問的資源是 /bucket/，ResourceType應該是bucket，如果訪問的資源是 /bucket/object，ResourceType應該是object。

OSS操作支援但參數不支援的操作

如果在OSS合法的操作中，添加了OSS不支援的參數（例如在PUT的時候，加入If-Modified-Since參數），OSS會返回400 Bad Request錯誤

錯誤請求樣本：

```
PUT /abc.zip HTTP/1.1
Host: bucketname.oss-cn-shanghai.aliyuncs.com
Accept: */*
Date: Thu, 11 Aug 2016 01:44:50 GMT
If-Modified-Since: Thu, 11 Aug 2016 01:43:51 GMT
Content-Length: 363
```

返回樣本：

```
HTTP/1.1 400 Bad Request
Server: AliyunOSS
Date: Thu, 11 Aug 2016 01:44:54 GMT
Content-Type: application/xml
Content-Length: 322
Connection: keep-alive
x-oss-request-id: 57ABD896CCB80C366955187E
x-oss-server-time: 0
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NotImplemented</Code>
  <Message>A header you provided implies functionality that is not implemented.</Message>
  <RequestId>57ABD896CCB80C366955187E</RequestId>
  <HostId>bucketname.oss-cn-shanghai.aliyuncs.com</HostId>
  <Header>If-Modified-Since</Header>
</Error>
```

10. 接入OSS

10.1. 快速開始

1. 登入[OSS管理主控台](#)，開通OSS。
2. 建立一個Bucket。
3. 上傳和下載檔案。

具體請參見[開始使用阿里雲 OSS](#)文檔。

快速了解OSS的上傳下載

開始使用SDK之前，請先參考開發人員指南關於上傳下載檔案的功能介紹。

OSS是使用RESTful API來操作的，所有的請求都是標準的HTTP請求。

- OSS提供了多種類型的上傳檔案的方法，如使用單次PUT請求完成的[簡單上傳](#)、使用網頁表單直接上傳的[表單上傳](#)、以及用於大檔案上傳的[分區上傳](#)還有適用於視頻監控等領域使用的[追加上傳](#)。
- 同樣也可以使用多種方法從OSS下載檔案，[簡單下載](#)和下載大檔案的[斷點續傳下載](#)。

基於SDK快速開始

1. 開通OSS後，從控制台上獲取AccessKeyId和AccessKeySecret。
2. 下載各種開發語言SDK。
3. 根據SDK的文檔描述，完成上傳、下載檔案等操作。

具體請參見[SDK文檔開發](#)。

10.2. 基於OSS的移動開發

開發架構圖

典型的基於OSS的移動開發有四個組件：

- OSS：提供上傳、下載、上傳回調等功能。
- 開發人員的移動客戶端（App或者網頁應用），簡稱客戶端：通過開發人員提供的服務，間接使用OSS。
- 應用伺服器：客戶端互動的伺服器。也是開發人員的商務服務器。
- 阿里雲STS：頒發臨時憑證。

最佳實踐

- [30分鐘快速搭建行動裝置 App直傳服務](#)
- [移動端回調案例分析及代碼下載](#)
- [輕鬆搞定移動端安全性原則](#)

開發商務程序

- 臨時憑證授權的上傳

如圖所示：

□

具體步驟如下：

- i. 客戶端嚮應用伺服器發出上傳檔案到OSS的請求。
- ii. 應用伺服器向STS伺服器請求一次，獲取臨時憑證。
- iii. 應用伺服器回複客戶端，將臨時憑證返回給客戶端。
- iv. 客戶端獲取上傳到OSS的授權（STS的AccessKey以及Token），調用OSS提供的移動端SDK上傳。
- v. 客戶端成功上傳數據到OSS。如果沒有設定回調，則流程結束。如果設定了回調功能，OSS會調用相關的介面。

這裡有幾個要點：

- 客戶端不需要每次都嚮應用伺服器請求授權，在第一次授權完成之後可以緩存STS返回的臨時憑證直到超過失效時間。
- STS提供了強大的許可權控制功能，可以將客戶端的存取權限制到Object等級，從而實現不同客戶端在OSS上傳的Object的完全隔離，極大的提高了安全係數。

更多資訊請參見[授權給第三方上傳](#)。

- 簽名URL授權的上傳和表單上傳

如圖所示：

□

具體步驟如下：

- i. 客戶端嚮應用伺服器發出上傳檔案到OSS的請求。
- ii. 應用伺服器回複客戶端，將上傳憑證（簽名URL或者表單）返回給客戶端。
- iii. 客戶端獲取上傳到OSS的授權（簽名URL或者表單），調用OSS提供的移動端SDK上傳或者直接表單上傳。
- iv. 客戶端成功上傳數據到OSS。如果沒有設定回調，則流程結束。如果設定了回調功能，OSS會調用相關的介面。

更多資訊請參見[授權給第三方上傳](#)。

- 臨時憑證授權的下載

跟臨時憑證授權的上傳過程類似：

- i. 客戶端嚮應用伺服器發出下載OSS檔案的請求。
- ii. 應用伺服器向STS伺服器請求一次，獲取臨時憑證。
- iii. 應用伺服器回複客戶端，將臨時憑證返回給客戶端。
- iv. 客戶端獲取下載OSS檔案的授權（STS的AccessKey以及Token），調用OSS提供的移動端SDK下載。
- v. 客戶端成功從OSS下載檔案。

這裡有幾個要點：


- 和上傳類似，客戶端對於臨時憑證也可以進行緩存從而提高訪問速度。
- STS同樣也提供了精確到Object的下載許可權控制，和上傳許可權控制結合在一起可以實現各移動端在OSS上儲存空間的完全隔離。

- 簽名URL授權的下載

跟簽名URL授權的上傳類似：

- i. 客戶端嚮應用伺服器發出下載OSS檔案的請求。
- ii. 應用伺服器回複客戶端，將簽名URL返回給客戶端。

- iii. 客戶端獲取下載OSS檔案的授權（簽名URL），調用OSS提供的移動端SDK下載。
- iv. 客戶端成功從OSS下載檔案。

 說明 客戶端不能儲存開發人員的AccessKey，只能獲取應用伺服器簽名的URL或者是通過STS頒發的臨時憑證（也就是STS的AccessKey和Token）。

最佳實踐

- [RAM和STS使用指南](#)

功能使用參考

- SDK: Android SDK [上傳檔案](#)
- SDK: iOS SDK [上傳檔案](#)