

目录

目录	1
Endpoint与Region的对应关系	3
KS3访问域名使用规则	3
KS3域名构成规则	3
通过外网访问KS3服务	3
通过内网访问KS3服务	3
通过IPV6地址访问KS3	4
KEC实例通过KS3内网地址访问KS3资源	4
获取KS3内网地址	4
同地域KEC实例访问KS3资源	4
存储类型介绍	5
根据数据的访问频度和成本的不同需求，KS3提供标准、低频、归档三种存储类型：	5
存储类型对比	6
存储类型转换	6
自动转换Object的存储类型	6
手动转换Object的存储类型	6
注意事项	7
创建存储空间	7
命名规则	7
操作方式	7
权限控制	7
查看存储空间列表	8
操作方式	8
删除存储空间	8
操作方式	8
项目管理	8
防盗链	8
Referer为空	9
在防盗链设置中，允许空Referer和不允许空Referer的区别如下。	9
日志设置	9
操作方式	9
域名绑定	9
CORS配置	10
操作方式	10
设置文件名功能	10
对象标签	10
使用场景	10
使用说明	11
对象标签和生命周期管理	11
重定向回源（原镜像）	12
文件预推	12
生命周期	12
细节说明	13
状态：	13
策略：	13
过期策略	13
规则执行	13
转换为低频存储文件	13
转换为归档存储文件	14
过期删除	14

约束和限制	14
存储类型转换费用说明	14
生命周期规则匹配逻辑	14
镜像回源	14
操作方式	15
镜像方式	15
多条规则匹配情况举例：	16
跨区域复制	16
使用场景	16
操作方式	16
使用说明	17
限制说明	17
回源	17
1. 镜像回源	17
2. 重定向回源，即原镜像	18

Endpoint与Region的对应关系

地区包含：中国（北京）、中国（上海）、中国（广州）、中国（香港）、俄罗斯、新加坡、金融专区（北京）、金融专区（上海）、政务专区（北京）。

Region中文名称	Region英文名称	外网域名(endpoint)	内网域名(endpoint)
中国（北京）	BEIJING	ks3-cn-beijing.ksyun.com	ks3-cn-beijing-internal.ksyun.com
中国（上海）	SHANGHAI	ks3-cn-shanghai.ksyun.com	ks3-cn-shanghai-internal.ksyun.com
中国（广州）	GUANGZHOU	ks3-cn-guangzhou.ksyun.com	ks3-cn-guangzhou-internal.ksyun.com
中国（香港）	HONGKONG	ks3-cn-hk-1.ksyun.com	ks3-cn-hk-1-internal.ksyun.com
俄罗斯	RUSSIA	ks3-rus.ksyun.com	ks3-rus-internal.ksyun.com
新加坡	SINGAPORE	ks3-sgp.ksyun.com	ks3-sgp-internal.ksyun.com
金融专区（北京）	JR_BEIJING	ks3-jr-beijing.ksyun.com	ks3-jr-beijing-internal.ksyun.com
金融专区（上海）	JR_SHANGHAI	ks3-jr-shanghai.ksyun.com	ks3-jr-shanghai-internal.ksyun.com
政务专区（北京）	GOV_BEIJING	ks3-gov-beijing.ksyun.com	ks3-gov-beijing-internal.ksyun.com

说明

- 如果不配置成对应的外网域名将返回307，会跳转到Bucket所在的Region。
- 金融专区与政务专区的使用，需联系您的商务，或者提交工单开通对应权限。

KS3访问域名使用规则

KS3会为每一个存储空间（Bucket）分配默认的访问域名，本文介绍KS3访问域名的构成规则及使用方式。

KS3域名构成规则

针对KS3的网络请求，除了GetService这个API以外，其他所有请求的域名都是带有指定Bucket信息的三级域名组成的。

访问域名结构：BucketName.Endpoint。BucketName为您的存储空间名称，Endpoint为存储空间对应的地域域名。

通过外网访问KS3服务

外网指的是互联网。通过外网访问产生的流入流量（写）是免费的，流出流量（读）是收费的。

外网访问KS3有如下两种方式：

- 访问方式一：访问时以URL的形式来表示KS3的资源。KS3的URL构成如下：

```
<Schema>://<Bucket>.<外网Endpoint>/<Object>
```

- Schema：HTTP或者为HTTPS。
- Bucket：KS3存储空间名称。
- 外网Endpoint：Bucket所在地区供外网访问的Endpoint，各地区Endpoint详情请参见 [Endpoint与Region的对应关系](#)。
- Object：上传到KS3上的文件的访问路径。

示例1：如果您的Region为北京（ks3-cn-beijing），Bucket名称为test，Object访问路径为aaa/bbb.txt，那么您的外网访问地址为 test.ks3-cn-beijing.ksyun.com/aaa/bbb.txt。

示例2：您可以直接将访问路径为aaa/ddd.png的Object的URL放入HTML中使用，例如：

```

```

- 访问方式二：通过KS3 SDK配置外网访问域名。

KS3 SDK会对您的每一个操作拼接访问域名。但您在对不同地域的Bucket进行操作的时候需要设置不同的Endpoint。

以Java SDK为例，对北京地区的Bucket进行操作时，需要在对类实例化时设置Endpoint：

```
Ks3ClientConfig config = new Ks3ClientConfig();
config.setEndpoint("ks3-cn-beijing.ksyun.com");
Ks3 client = new Ks3Client("<您的AccessKeyID>", "<您的AccessKeySecret>", config);
```

通过内网访问KS3服务

内网指的是金山云产品之间的内网通信网络，例如您通过KEC云服务器访问KS3服务。内网产生的流入和流出流量均免费，但是

请求次数仍会计费。

内网访问KS3有如下两种方式：

- 访问方式一：在访问的时候以URL的形式来表示KS3的资源。KS3的URL构成如下：

```
<Schema>://<Bucket>.<内网Endpoint>/<Object>
```

- Schema：HTTP或者为HTTPS。
- Bucket：KS3存储空间名称。
- 内网Endpoint：Bucket所在地区供同地区KEC访问的内网Endpoint，各地区Endpoint详情请参见 [Endpoint与Region的对应关系](#)。
- Object：上传到KS3上的文件的访问路径。

示例：如果您的Region为北京，Bucket名称为test，Object访问路径为aaa/bbb.txt，那么您的内网访问地址为test.ks3-cn-beijing-internal.ksyun.com/aaa/bbb.txt。

- 访问方式二：通过KEC使用KS3 SDK配置内网Endpoint。

以Java SDK为例，对北京地区的Bucket进行操作时，需要将Endpoint设置为北京地区的内网Endpoint：

```
Ks3ClientConfig config = new Ks3ClientConfig();
config.setEndpoint("ks3-cn-beijing-internal.ksyun.com");
Ks3 client = new Ks3Client("<您的AccessKeyID>", "<您的AccessKeySecret>", config);
```

通过IPV6地址访问KS3

IPv6是IETF（互联网工程任务组，Internet Engineering Task Force）设计的用于替代现行版本IP协议（IPv4）的下一代IP协议。

KS3目前已支持通过IPv6/IPv4双栈域名访问。

您的IPv6/IPv4客户端均可以使用KS3提供的统一双栈域名访问您的存储空间。您的DNS服务器将按照您使用的协议版本解析对应协议的KS3服务器地址给您。

目前可以通过IPv6协议访问的地域有：

- 北京
- 上海
- 广州

如有需要使用新版双栈域名，请提交工单或者联系商务开通权限。

KEC实例通过KS3内网地址访问KS3资源

当您使用KEC实例可以直接通过内网访问同地区具有读取权限的KS3资源，不收取流量费用。本文介绍KEC实例如何通过KS3内网地址访问KS3资源。

获取KS3内网地址

- 通过KS3控制台获取

登录KS3管理控制台，打开指定Bucket的概览页面，在访问域名区域查看Bucket的Endpoint和Bucket域名。

- 通过固定格式获取

KS3的访问地址为固定格式：Endpoint/BucketName。其中，Endpoint为存储空间所在的地区对应的访问域名，BucketName为您的存储空间名称。详情请参见 [KS3访问域名使用规则](#)。

同地域KEC实例访问KS3资源

与KS3同地域的KEC实例可以通过以下方式使用内网访问KS3资源：

- 通过URL直接访问KS3资源

您可以直接使用KS3资源的内网地址访问有权限的KS3资源。例如，北京地区某Bucket名为aaa，根目录下有个Object名为bbb.png，处于公共读状态。此时，北京地区的KEC实例均可以使用http://ks3-cn-beijing-internal.ksyun.com/aaa/bbb.png访问此Object。因此，您可以将KS3资源的访问URL嵌入到您的网站中，提供给同地区的KEC用户。

- 通过KS3Util访问KS3资源

您可以在配置KS3Util访问参数的时候，将Endpoint设置为KS3的内网Endpoint地址。详情请参见[KS3Util命令行工具](#)。

- 通过SDK访问KS3资源

SDK初始化client的时候，Endpoint配置KS3内网对应的Endpoint即可。

- Java SDK

```
Ks3ClientConfig config = new Ks3ClientConfig();
config.setEndpoint("ks3-cn-beijing-internal.ksyun.com");
Ks3Client client = new Ks3Client("<您的AccessKeyID>", "<您的AccessKeySecret>", config);
```

更多详情请参见[Java SDK初始化](#)。

- PHP SDK

```
$client = new Ks3Client("<您的AccessKeyID>", "<您的AccessKeySecret>", "endpoint")
```

更多详情请参见[PHP SDK初始化](#)。

- Python SDK

```
from ks3.connection import Connection
ak = 'YOUR_ACCESS_KEY'
sk = 'YOUR_SECRET_KEY'
c = Connection(ak, sk, host='YOUR_REGION_ENDPOINT', is_secure=False, domain_mode=False)
```

更多详情请参见[Python SDK初始化](#)。

- C# SDK

```
String accessKey = "YOUR ACCESS KEY";
String secretKey = "YOUR SECRET KEY";
String bucketName = "YOUR BUCKET NAME";
String objKeyName = "YOUR OBJECT KEY";
String endPoint = "ks3-cn-beijing-internal.ksyun.com"; //此处以北京region为例
ks3Client = new KS3Client(accessKey, secretKey);
ks3Client.setEndpoint(endPoint);
```

更多详情请参见[C# SDK初始化](#)。

- GO SDK

```
credentials := credentials.NewStaticCredentials("<AccessKeyID>", "<AccessKeySecret>", "")
client := s3.New(&aws.Config{
Region: "BEIJING",
Credentials: credentials,
Endpoint: "ks3-cn-beijing-internal.ksyun.com", //ks3地址
DisableSSL: true, //是否禁用https
LogLevel: 1, //是否开启日志, 0为关闭日志, 1为开启日志
S3ForcePathStyle: false, //是否强制使用path_style方式访问
LogHTTPBody: true, //是否把HTTP请求body打入日志
Logger: os.Stdout, //打日志的位置
})
```

更多详情请参见[GO SDK初始化](#)。

存储类型介绍

根据数据的访问频度和成本的不同需求，KS3提供标准、低频、归档三种存储类型：

- **标准存储类型**：提供通用的对象存储服务，适合频繁访问、有热点存在的各类音视频、图片、网站静态资源的数据，较低的延迟和较高的吞吐量性能的数据。
- **低频访问存储类型**：适用于较低访问频率的业务场景，适合长期保存、较少访问的数据。低频存储在降低存储价格的基础上同时保持访问延时在毫秒级，保证了用户在取回数据的场景下，无需等待，高速读取。但对存储时长有一定要求，存储时间短于30天的文件提前删除会产生提前删除费用。无论是在内网还是外网，数据获取都会产生数据取回费用。
- **归档存储类型**：适用于数据存储后访问频率极低的业务场景，适合保存需要长期存储、极少访问的数据。对比标准存储和低频存储，归档存储能提供相同的数据可靠性和服务可用性，同时又大幅降低了存储的成本，但数据进入到可读取状态需要1~10分钟的解冻时间，适合需要长期保存的医疗影像、档案、业务日志及影像素材等。归档存储类型有最短存储时间，存储时间短于90天的Object提前删除会产生提前删除费用，另外，读取归档存储中的数据时，会产生数据取回费用。有关金山云KS3各类型存储价格的相关信息，请参见[产品报价](#)。

存储类型对比

对比指标	标准存储类型	低频访问存储	归档存储类型
数据可靠性	99.999999999%	99.999999999%	99.999999999%
访问可用性	99.95%	99.9%	99%（数据解冻后）
最低存储时间	无	30天	90天
访问延时	实时访问 ms延迟	实时访问 ms延迟	解冻时间1~10分钟，解冻后可正常访问
图片处理	支持	支持	解冻后支持
适应场景	音视频分发源站、图片分享、大型网站、移动应用	企业应用与数据库备份，监控摄像头数据，影视素材	医疗影像、档案、业务日志，影像素材

- 有关金山云KS3低频存储的接口访问细节，请参见[低频存储](#)。
- 有关金山云KS3归档存储的接口访问细节，请参见[归档存储](#)和[解冻接口](#)。

存储类型转换

KS3支持标准存储、低频访问、归档存储三种存储类型，不同存储类型之间可以相互转换。

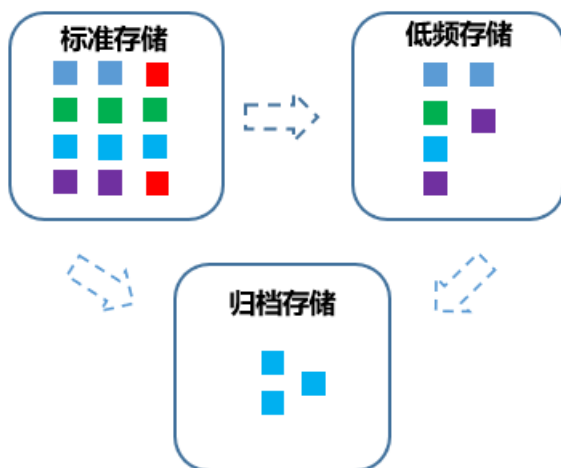
KS3支持通过以下方式转换对象（Object）的存储类型：

- 方式一：通过生命周期规则自动转换Object的存储类型
- 方式二：通过控制台、SDK、API手动转换Object的存储类型

自动转换Object的存储类型

生命周期管理（Lifecycle），支持以下存储类型的自动转换：

- 标准存储类型转换为低频访问或归档存储类型
- 低频访问类型转换为归档存储类型



如图：

生命周期管理的功能具体操作请参见：[生命周期管理](#)。

您可以通过控制台完成以上Lifecycle策略的配置。具体操作请参见[设置生命周期](#)。

手动转换Object的存储类型

手动转换Object存储类型实际是通过覆写操作，将Object转储为指定的存储类型。所以，若转换的Object是低频访问或归档存储类型，且存储未满足指定天数的，会产生数据提前删除费用，详情请参见[计费方式说明](#)和[计费项说明](#)。

对于归档存储类型的Object，需要先执行解冻（Restore）操作，解冻成可读取状态后，才可以修改存储类型。详情请参见[归档存储类型](#)。

手动转换Object存储类型的操作方式如下：

1. 通过 `put object copy` 修改文件存储类型。

- API参考: [put object copy](#)
- SDK参考: [JAVA SDK-修改文件存储类型](#)
- 控制台修改: [控制台-修改文件存储类型](#)

更多SDK 支持持续更新中。

注意事项

Object存储类型转换后，会按照转换后的存储类型的存储单价计算存储费用。低频访问类型和归档存储类型需要特别注意：

- 最短存储期限

低频类型的Object需要至少保存30天；归档存储类型的Object需至少保存90天。提前删除这些Object，需要补足，到达最短存储期限的剩余天数的存储费用。

- 手动转换Object存储类型时，Object的存储天数重新计算。例如，test.txt作为标准存储类型已经在ks3中存储了10天，手动将Object转换为低频访问类型，则需继续存储30天才满足最少存储30天的要求。

- 归档存储类型的Restore时间

归档存储类型Object恢复到可读取状态（Restore）需要1分钟到10分钟的解冻时间，所以如果业务场景上需要实时读取，建议只转换成低频访问存储类型，而不应转换成归档存储类型，避免转换成归档存储类型后，数据无法实时读取。

- 数据取回费用

访问低频访问类型和归档存储类型的Object时会额外收取数据取回费用，此费用与外网下行流量费用是两个独立计费项。如果每个Object平均访问频率高于每月1次，不建议转换成低频访问或者归档存储类型。

创建存储空间

在上传任何文件到 KS3 之前，您需要首先创建存储空间以用来存储文件。存储空间具有各种配置属性，包括其地理区域、访问权限以及其他元数据。

说明：创建存储空间API详情请参见[PUT Bucket](#)。

注意：存储空间的命名必须符合命名规范。所选定的存储空间名称在KS3的所有现有存储空间名称中必须具有唯一性。创建后不支持更改存储空间名称。

命名规则

- 仅包含小写英文字母（a-z），数字，点（.），中线，即： abcdefghijklmnopqrstuvwxyz0123456789.-
- 必须由字母或数字开头
- 长度在3和63个字符之间
- 不能是IP的形式，类似192.168.0.1
- 不能以kss开头

操作方式

操作方式	说明
控制台	Web 应用程序，直观易用
JAVA SDK	
PHP SDK	
Python SDK	
Android SDK	
IOS SDK	
JavaScript SDK	丰富、完整的各类语言 SDK demo
Node.js SDK	
C# SDK	
C/C++ SDK	
GO SDK	

权限控制

您可以在创建存储空间的时候设置相应的存储空间权限（ACL），也可以在创建之后修改 ACL。如果不设置 ACL，默认值为私有。更多信息，请参见[ACL](#)。

查看存储空间列表

存储空间创建之后，您可以通过KS3 API的Get Service接口获取存储空间列表信息。

说明：查看存储空间API详情请参见[GET Service](#)。

操作方式

操作方式	说明
控制台	Web 应用程序，直观易用 图形化工具，易操作
GUI图形化界面工具 KS3 Explorer	
JAVA SDK	丰富、完整的各类语言 SDK demo
PHP SDK	
Python SDK	
Android SDK	
IOS SDK	
JavaScript SDK	
Node.js SDK	
C# SDK	
C/C++ SDK	
GO SDK	

删除存储空间

如果您不再需要存储空间，请将其删除以免进一步产生费用。

说明：删除存储空间之前请确保其中存储的文件已经全部清空，否则无法删除存储空间。

删除存储空间API详情请参见[DELETE Bucket](#)。

操作方式

操作方式	说明
控制台	Web 应用程序，直观易用
JAVA SDK	
PHP SDK	丰富、完整的各类语言 SDK demo
Python SDK	
Android SDK	
IOS SDK	
JavaScript SDK	
Node.js SDK	
C# SDK	
C/C++ SDK	
GO SDK	

项目管理

用户可以将存储空间（Bucket）分配到不同的项目组中，方便按照项目粒度进行权限分配、费用识别和统一管理。

在将存储空间（Bucket）分配到不同的项目组中之前，用户需要先维护项目信息，详见文档[项目管理](#)。

未指派项目信息的存储空间(Bucket)将默认分配到【默认项目组】中。

仅支持控制台设置，操作步骤请参见[控制台空间管理-项目管理](#)。

防盗链

KS3提供的防盗链通过黑白名单的方式控制，其中黑名单用于添加禁止访问的来源域名，白名单用于添加允许访问的来源域

名。防盗链功能可手动设置关闭。

录入域名的时候需要注意以下规则：

- 域名之间用逗号(,)分开，不需要写 http:// 或者 https://
- 支持域名前使用通配符 *，可用于指代当前域名下的多级子域名

由于有些合法的请求是不会带referer来源头部的，所以有时候不能拒绝来源头部（referer）为空请求，在KS3中，我们还提供了手动设置referer是否为空的选项。

具体说明如下：

Referer为空

空Referer指的是HTTP请求中Referer头部的内容为空，或者HTTP请求中不包含Referer头部。如果有下面两种情况则Referer为空。1. 当请求并不是由链接触发产生。比如，直接把地址输入地址栏里打开页面。2. 从https页面上的链接访问到非加密的http页面时，在http页面上是检查不到Referer的。

在防盗链设置中，允许空Referer和不允许空Referer的区别如下。

在防盗链的白名单设置中，如果指明白名单中包含空的Referer，那么通过浏览器地址栏直接访问该资源URL是可以访问到的。如果不指明需要包含空的Referer，那么通过浏览器直接访问也是被禁止的。

控制台操作步骤请参见[控制台空间管理-防盗链](#)。

日志设置

您可以利用管理控制台为存储空间启用和禁用日志记录。您可以在已启用日志记录的同一存储空间中存储日志，也可以在同一Region内另外创建新存储空间来存储日志。

KS3中日志记录功能默认为不开启状态，您可以手动开启，并手动设置存储位置。

日志文件格式为：ks3log_空间名_yyyy_MM_dd.log

日志设置相关API请参见[API文档-GET Bucket logging](#)。

操作方式

操作方式	说明
控制台	Web 应用程序，直观易用
JAVA SDK	
PHP SDK	
Node.js SDK	丰富、完整的各类语言 SDK demo
GO SDK	

域名绑定

KS3支持用户将自定义的域名绑定到KS3存储空间，这样就可以用自定义域名访问存储空间下的文件，能够支持用户无缝地将存储迁移到KS3上，也可以避免域名安全问题。域名绑定成功后，添加CNAME记录指向存储空间对应的外网域名，就可以使用自定义域名访问KS3。

控制台操作步骤请参见[控制台空间管理-域名绑定](#)。

注意：

1. 要绑定的自定义域名需要在工信部备案，否则绑定会失败。
2. 要绑定的自定义域名不能同时绑定到当前账户下同一Region的两个或两个以上的存储空间（bucket）。
3. 每个Bucket最多可以绑定50个域名。
4. 每个Bucket最多可以有50个域名绑定记录（包括成功绑定的记录和由于未备案、域名冲突等原因导致的错误绑定记录）。
5. 绑定成功后，使用自定义域名访问对应存储空间（Bucket）的文件时，访问地址不需要写存储空间（Bucket）名称。

比如：用户在北京region创建了名为mybucket的存储空间（Bucket），绑定了自定义域名mydomain.com。

- 在绑定之前：mybucket.ks3-cn-beijing.ksyun.com/hello.jpg

- 在绑定之后: mydomain.com/hello.jpg

以下是错误的URL:

mybucket.mydomain.com/hello.jpg

mydomain.com/mybucket/hello.jpg

6. 目前自定义域名绑定证书需在CDN控制台进行, 暂不支持在KS3控制台直接绑定自定义域名证书。

CORS 配置

跨域资源共享 (Cross-Origin Resource Sharing, 简称 CORS), 是 HTML5 提供的标准跨域解决方案。CORS允许WEB端的应用程序访问不属于本域的资源。开发者可以利用KS3提供的接口控制跨域访问的各种权限, 开发灵活的WEB应用程序。

KS3会根据跨域请求匹配相对应的bucket下的CORS规则, 根据规则设定的权限进行检查, 并依次匹配每一条规则, 根据规则的设置来允许请求并返回相对应的header。

CORS相关API请参见[API文档-CORS相关](#)。

操作方式

操作方式

[控制台](#)

[JAVA SDK](#)

[PHP SDK](#)

说明

Web 应用程序, 直观易用

丰富、完整的各类语言 SDK demo

设置文件名功能

当启用该功能后用户可以自定义上传到金山云中的文件名, 当上传成功后, 将根据用户设定的命名规则, 重新命名新的文件名。**说明:** 该功能只适用于通过API、SDK调用Put Object请求上传文件时使用, 对于通过控制台和使用Post Object接口请求上传的文件, 该规则不起作用。

控制台操作步骤请参见[控制台空间管理-设置文件名](#)。

对象标签

KS3支持使用对象标签 (Object Tagging) 对存储的对象 (Object) 进行分类, 您可以针对指定标签的Object设置生命周期规则。

说明: 对象标签功能目前处于公测期, 公测期间免费, 公测期结束后使用对象标签会产生一定的费用, 计费方式参考: [计费方式和计量项说明](#)。

对象标签使用一组键值对 (Key-Value) 标记对象, 您可以在上传文件时添加标签, 也可以为现有文件添加标签。

每个对象最多添加10个标签。

Key命名规则:

- 支持大小写字母、数字、空格和符号 + - = . _ : /
- 1-128字节, 区分大小写, 不能以空格开头或结尾, 不容许为空
- 不允许设置系统保留字段, ksc:与kss:开头

value设置规则:

- 支持大小写字母、数字、空格和符号 + - = . _ : /
 - 1-256字节, 区分大小写, 不能以空格开头或结尾
- 只有Bucket拥有者和授权用户拥有读写对象标签的权限。
- 跨区域复制时, 对象标签也会复制到目的端。

使用场景

对象标签不受文件目录限制, 您可以批量操作拥有指定标签的对象, 例如:

- 设置针对指定标签的生命周期规则。例如，周期性生成的非长期保存的文件，可以在上传时设置指定的标签，之后通过生命周期规则，将拥有这个标签的文件定期删除。

使用说明

- 对象标签涉及的API接口
 - [PutObjectTagging](#): 设置对象的标签。若对象已有标签，则覆盖原标签。
 - [GetObjectTagging](#): 读取对象的标签。
 - [DeleteObjectTagging](#): 删除某个对象关联的标签。
 - [PutObject](#): 用户可在上传对象时通过 `x-kss-tagging` 请求头指定对象标签。
 - [PostObject](#): 用户可在表单上传对象时通过 `tagging` 指定对象标签。
 - [InitiateMultipartUpload](#): 用户可在初始化分片上传时，通过 `x-kss-tagging` 请求头指定对象标签。
 - [PutObjectCopy](#): 复制对象时可通过 `x-kss-tagging-directive` 设置是否覆盖源对象标签，`x-kss-tagging` 请求头指定目标对象标签。
 - [PutObjectFetch](#): 用户拉取文件，并上传至KS3某个 bucket 中，可通过 `x-kss-tagging` 请求头指定目标对象标签。
 - [GetObject](#): 当用户拥有读取对象标签的权限时，响应头中会带有 `x-kss-tagging-count` 标识对象的标签个数。
 - [HeadObject](#): 当用户拥有读取对象标签的权限时，响应头中会带有 `x-kss-tagging-count` 标识对象的标签个数。
- 对象标签涉及的控制台操作：
 - 控制台使用手册 > 内容管理 > [对象标签管理](#)
- 对象标签涉及的SDK文档：
 - [JAVA](#)
 - [GO](#)
 - 权限说明

能够进行Tag相关操作的用户、角色、服务，必须具有如下相关权限，您可以通过Bucket policy 与IAM 实现如下权限授权管理：

- [GetObjectTagging](#): 获取对象标签的权限。拥有此权限，可以查看到文件的已有标签。
- [PutObjectTagging](#): 设置对象标签的权限。拥有此权限，可以为文件设置标签。
- [DeleteObjectTagging](#): 删除对象标签的权限。拥有此权限，可以删除文件的标签。

例如：

- 当为已经存在的对象设置标签或者通过上传时指定对象的标签，您需要具有 [PutObjectTagging](#) 权限
- 当您通过CopyObject接口，复制一个带有标签的对象，您需要同时具有：
 - 源文件权限的[GetObject](#)，[GetObjectTagging](#) 权限
 - 目标文件权限[PutObject](#)，[PutObjectTagging](#)，权限
 - 但是如果指定 `x-kss-tagging-directive` 为REPLACE且新指定tag为空，则目标权限只需[PutObject](#) 权限即可
 - 删除对象上标签，您需要具有[DeleteObjectTagging](#) 权限
 - 但是若您删除对象则会将对象附属标签自动删除，您仅需[DeleteObject](#) 权限

更多接口权限细节请参考，[对象标签涉及的API接口](#)

对象标签和生命周期管理

在生命周期规则配置中，您可以指定生命周期规则生效的条件。生命周期规则可针对前缀或对象标签生效，您也可以同时指定两者作为条件。

- Tag条件中，标签的Key和Value必须同时匹配。
- 同一个规则中，若同时配置 前缀和多个对象标签，则对象需满足前缀且同时匹配规则中所有对象标签，才视为适用于该规则。

示例：

```
<LifecycleConfiguration>
  <Rule>
    <ID>1</ID>
    <Filter>
      <And>
        <Prefix>documents</Prefix>
        <Tag>
          <Key>age</Key>
          <Value>21</Value>
        </Tag>
        <Tag>
          <Key>name</Key>
          <Value>li</Value>
        </Tag>
      </And>
    </Filter>
  </Rule>
</LifecycleConfiguration>
```

```

</Filter>
<Status>Enabled</Status>
<Expiration>
  <Days>30</Days>
</Expiration>
</Rule>
<Rule>
  <ID>2</ID>
  <Filter>
    <And>
      <Prefix>pic</Prefix>
      <Tag>
        <Key>age</Key>
        <Value>20</Value>
      </Tag>
    </And>
  </Filter>
  <Status>Enabled</Status>
  <Transition>
    <Days>60</Days>
    <StorageClass>Archive</StorageClass>
  </Transition>
</Rule>
</LifecycleConfiguration>

```

以上规则中：

- 以documents为前缀且同时拥有标签age=21和name=li的对象在30天后被删除。
- 以pic为前缀且拥有标签age=20的对象在60天后被转换为Archive（归档存储）。

说明： 更多信息请参见[管理文件生命周期](#)。

重定向回源（原镜像）

镜像回源主要用于无缝迁移数据到KS3，即当您的服务已经在自己建立的源站或者在其他云产品上运行，需要迁移到KS3上，但是又不能停止服务，此时可利用镜像回源功能实现。

当Bucket配置了镜像回源后，用户向KS3访问一个文件（Object），而该文件并没有保存在KS3中时，KS3会返回给用户一个Location指向源站的302重定向请求；同时，KS3也会异步向回源地址请求该文件，并保存至KS3中。

注意：

- 利用镜像回源进行数据迁移，单个文件最大支持50GB。
- 当用户访问KS3上没有的文件时，会触发KS3异步向源站下载该文件，KS3会尽快将文件下载下来并保存在KS3，但KS3不保证时间和成功率。
- 支持同时配置多个镜像源，镜像源之间以英文“;”隔开，回源时按照源站配置顺序拉取，拉取成功后，不再从剩余源站尝试拉取。
- 若镜像源还有一个301到另一个源的过程，那么此源上的文件不会被异步拉取到KS3上（支持302）
- 回源地址支持输入域名与IP，多个回源地址之间以英文“;”隔开，地址最大长度为512个字符。

文件预推

您可根据模板中的资源列表，将文件从镜像源站拉取到KS3中，并根据所设定的规则进行存储。

注意：

- 当配置好镜像回源时，才会出现文件预推的配置，归档存储类型的bucket不支持文件预推设置。
- 当设置文件名和文件预推功能同时存在时，设置文件名对文件预推功能无效。
- 控制台操作步骤请参见[控制台空间管理-镜像](#)。

说明：若镜像源还有一个301到另一个源的过程，那么此源上的文件不会被异步拉取到KS3上（支持302）

生命周期

KS3支持在特定时间点或对象最后一次被修改后的一段时间，进行批量删除对象或者批量转换对象的存储类型。为实现此目的，用户需要在存储空间上创建生命周期管理规则。

规则创建：

1. 选定需要设置生命周期的bucket。
2. 设置规则，分为针对整个bucket生效或者是对特定前缀的对象生效两种。

3. 设置策略，可以指定特定过期日期或过期天数，即在特定日期或者在文件最后修改时间后多少天后进行删除或者转换。

规则创建后，用户可以启用或者停用规则。

每个存储空间最多可以创建1000条规则。

用户在2017年4月12日之前上传的文件不支持存储类型转换和过期删除。

- 控制台操作步骤 > [控制台空间管理-生命周期管理](#)
- 生命周期相关API > [API文档-生命周期](#)

细节说明

状态：

指定本条规则的状态，分别表示启用和停用该规则。

策略：

匹配范围：KS3支持匹配方式：

- 按前缀匹配：
 - 按指定前缀匹配Object。可创建多条规则匹配不同的前缀，**多条前缀不能重叠**，例如，同一bucket配置了两条规则，一条前缀是pics/，一条前缀是pics/program，那么KS3会返回错误。
 - 全部文件与任意前缀的文件均为重叠关系，故当您设置全部文件范围的规则后，不能再创建文件前缀的规则。
 - 设置前缀时不要以“/”开始，否则规则不会匹配到任何object。
 - 如果规则设置到整个bucket上，那么该bucket下所有对象都适用于该规则；如果规则设置为某个前缀，那么该bucket下只有匹配该前缀的Object才适用这个规则。
- 按标签匹配：
 - 按指定标签的Key和Value匹配Object。
 - 单条规则可配置多个标签，KS3对所有拥有这些标签的对象执行生命周期规则。
 - 按前缀+标签匹配：按指定前缀和一个或多个标签的筛选条件匹配对象。
 - 配置到整个Bucket：匹配整个Bucket内的所有Object和碎片。此种方式只能创建一条规则。

例如：一个Bucket有如下几个Object：

```
pics/dog.gif
pics/cat.gif
pics/bird.gif
docs/readme.txt
```

如果一个规则指定的前缀是pics/，那么该规则就适用于前三个以pics/开头的object；如果前缀是docs/readme.txt，那么这条规则就只对docs/readme.txt起作用。

过期策略

用户可以指定过期天数或者过期日期：

- 如果指定过期天数，那么从object最后一次被修改当天算起，指定天数之后，数据将进行指定的删除或者转化存储类型操作。

例如：用户设置对所有对象，设置过期天数为3后执行删除操作，若object1最后修改时间为2018-08-01 18:01:26，那么3天后，也就是2018-08-05 00:00:00，对象将被删除；若object2最后修改时间为2018-08-03 18:01:26，那么3天后，也就是2018-08-07 00:00:00，对象将被删除。
- 如果指定过期日期，那么object最后修改时间在指定日期之前的对象会被执行操作。例如：用户设置过期日期为2018-01-01T00:00:00.000Z，对所有对象执行转化低频存储操作，若object1的最后修改时间为2017-08-01 18:01:26，则会被转化存储类型为低频，若object2的最后修改时间为2018-08-02 18:01:26，则不会被转化。

规则执行

转换为低频存储文件

可以选择过期日期天数或者不启用，规则配置在一天内生效，比如下午三点启用的规则，将会在第二天0点生效。从转换条件满足，转换操作被触发，KS3需要一段时间才能完成转换操作，这段时间与对象的数量规模有关。

转换为归档存储文件

可以选择过期日期天数或者不启用，规则配置在一天内生效，比如下午三点启用的规则，将会在第二天0点生效。

注意：

从转换条件满足，转换操作被触发(过期删除同理)，KS3需要一段时间才能完成转换操作，这段时间与对象的数量规模有关。

过期删除

规则配置在一天内生效，比如下午三点启用的规则，将会在第二天0点生效。从删除条件满足，删除行为被触发，KS3需要一段时间才能完成删除操作，这段时间与被删除对象的数量规模有关。

约束和限制

- 如果配置了转低频规则和转归档规则，会有如下限制：转低频时间或日期+30<=转归档时间或日期。
- 如果配置了转低频规则和删除规则，会有如下限制：转低频时间或日期<删除时间或日期。
- 如果配置了转归档规则和删除规则，会有如下限制：转归档时间或日期<删除时间或日期。
- 如果配置了转低频、转归档和删除规则，会有如下限制：转低频时间或日期+30<=转归档时间或日期<删除文件时间或日期。
- 如果只配置三条中的一条，则没有任何限制。

存储类型转换费用说明

- 低频存储最短存储期限为30天，早于30天删除、转换的文件，需要补足未满30天的剩余天数的存储费用，超过30天不需要补。
- 归档存储最短存储期限为90天，早于90天删除的Object，需要补足未满90天的剩余天数的存储费用，超过90天不需要补。

生命周期规则匹配逻辑

- 前缀规则生效说明 只要Object名称前缀和一条规则的前缀匹配，那么该规则就适用于它。例如，一个Bucket有如下几个Object： logs/program.log.1 logs/program.log.2 logs/program.log.3 doc/readme.txt 如果一个规则指定的前缀是logs/，那么该规则就适用于前三个以logs/开头的Object；如果前缀是doc/readme.txt，那么这条规则就只对doc/readme.txt起作用。
- 规则冲突行为 若您设置匹配规则，存在某时刻同时满足两种以上不同的执行操作，KS3优先级处理方式为：删除 > 转化为归档 > 转化为低频。 举例说明： 当您配置了如下规则

rule序号	prefix	tagging	action
rule1	123	a=1	20天后删除
rule2	12345	a=1	20天后转为归档存储
rule3	1233456	a=1	20天后转为低频存储

当满足前缀为123，且拥有a=1的对象会在20天后执行删除操作，此时文件已经不存在，所已规则二、规则三没有意义。

镜像回源

KS3提供同步镜像回源功能，当您的用户向KS3请求的文件不存在时，KS3会向回源地址请求这个文件，返回给用户。对于完整文件的请求，KS3会同时请求部分数据返回给用户，并将该文件存入KS3；对于Range请求，KS3会同时请求部分数据返回给用户，然后异步拉取数据存储至KS3。

注意：

1. 您最多可以配置20条回源规则，系统将按照规则顺序依次执行。
2. KS3镜像回源单个文件最大支持50GB。
3. 该功能目前仅北京地区支持。
4. 使用该功能需相应的权限，可联系商务或[提交工单](#)申请权限。

特别说明：

1. 拉取到KS3的数据默认以标准存储类型存储。
2. 拉取到KS3的文件不支持用KS3提供的设置文件名功能来修改文件名。
3. 源站拉取完成后，KS3不感知源站的文件变化。
4. 若用户设置了原图保护，在同步回源时失效。
5. 若用户设置了文件服务端加密/客户端传输密钥加密，在同步回源时失效。

操作方式

- 控制台操作步骤请参见 [控制台使用手册-镜像](#)
- API参见
 - [PUT BucketMirror](#): 为目标桶设置与更新一组镜像回源规则。
 - [GET BucketMirror](#): 获得源存储空间的镜像回源与重定向回源配置。
 - [DELETE BucketMirror](#): 删除某个Bucket的BucketMirror设置。

镜像方式

同步镜像回源流程

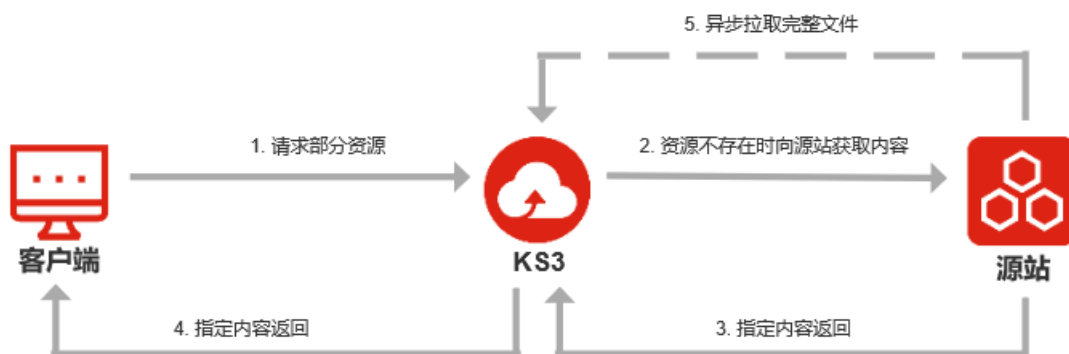
当您的客户向Bucket请求的某个Object不存在时，KS3会向回源地址请求这个Object，并且，为了避免等待整个文件需要较长时间，KS3会向源站流式拉取部分数据，源站返回数据后，KS3返回给用户，并且存在对应Bucket中。具体流程如下图所示。

1. 完整文件请求流程



细节说明

2. 部分（range）文件请求流程：



- 同步回源规则触发条件

当客户端Get Object请求的文件不存在时，且用户配置了该文件的同步镜像回源规则，并且该文件满足规则时，触发同步镜像回源。

- 请求失败返回规则

如果镜像源也不存在此文件，即镜像源返回给KS3的HTTP状态码为404，那么KS3也将返回404给用户。如果镜像源返回给KS3的HTTP status为其他非200的状态码（包括因为网络原因等获取不到文件的错误情况），KS3将返回424给用户，错误码为

MirrorFailed。如果源站返回3xx（3XX包含302与301），根据用户的回源规则配置，决定是否follow源站3xx重定向请求获取到资源。

- 同步回源文件更新规则

如果文件通过镜像回源，存储到了KS3中，之后源站里该文件发生了变化，KS3不会更新已存在于KS3上的该文件。

- 同步回源文件元信息

```
Content-Type
Content-Disposition
Cache-Control
Expires
```

- HTTP请求规则

传给KS3的Header信息不会传递给源站，QueryString信息是否会传递给源站取决于控制台回源规则中的配置。Header传递规则控制台输入显示参见[控制台使用手册-镜像回源]()。

如果源站是chunked编码返回，那么KS3首次回源时返回给用户的也是chunked编码。

多条规则匹配情况举例：

当客户存在多条规则且回源条件相同与部分重叠时：

规则一：回源条件全部文件404时回test.com。

规则二：回源条件全部文件404时回141.0.0.12。

规则三：回源条件文件名前缀为333的文件404时回back.com。

KS3的处理说明：

1. 请求优先匹配到规则一，同时test.com可以正常拉取数据，则不再匹配规则二。
2. 请求优先匹配到规则一，但是KS3回源服务前往 test.com时，该源站不可用，则KS3再去前往规则二配置的141.0.0.12站点。
3. 若规则二配置的141.0.0.12也恰巧不可用时，如果文件名符合前缀为333，则KS3继续前往规则三配置back.com；如果文件名前缀不是333，则KS3给用户返回404。

跨区域复制

跨区域复制（Cross-Region Replication）是在不同地域（Region）的KS3存储空间之间自动复制文件（Object），它会将Object的创建、更新和删除等操作从源存储空间复制到不同区域的目标存储空间。目标端对象是源存储空间的精确副本，它们具有相同的对象名、元数据以及内容，例如创建时间、拥有者、用户定义的元数据、Object ACL、对象内容等。支持复制未加密的对象和使用服务器端加密的对象。

- 说明
 - 开启跨区域复制功能后，主要从两个区域的存储空间在复制文件时会产生跨区域间的数据流量，KS3会收取跨区域复制流量费用。每同步1个Object，KS3会累计请求次数并按量计费，若源文件是低频存储类型文件还会正常收取低频数据取回费用。
 - 目前跨区域复制流量仅支持按量付费，不提供资源包。计量计费方式详情见[计费方式和计量项说明](#)。
 - 目前支持中国各地域（包括香港）之间进行跨区域复制。
 - 跨区域复制开启后，有一定的生效时间（约1-6分钟），生效后的复制就是近实时的。
 - 当前跨区域复制仅能支持对开启复制后的新增数据进行复制，**暂不支持对历史数据的复制**。

使用场景

- 安全合规：虽然KS3默认对每个存储的对象在物理盘上有多个副本，但安全合规要求所规定的数据需要具有一定距离的远程容灾能力。通过跨区域复制，KS3可以在远距离的数据中心之间复制数据以满足安全性要求。
- 减少延迟：为了最大限度缩短访问对象时的延迟，可以选择在地理位置与用户较近的两个数据中心之间复制对象副本。
- 数据备份与容灾：如果您对数据的安全性和可用性有极高的要求，以备发生如地震、海啸等灾难导致一个KS3数据中心损毁时，还能及时启用另一个数据中心的备份数据以快速恢复业务。
- 数据复制：由于业务原因，需要将数据从一个数据中心迁移到另一个数据中心，或者使用不同数据中心的副本。

操作方式

操作方式	操作指南	说明
------	------	----

控制台 [跨区域复制设置](#) Web应用程序, 直观易用
 SDK [JAVA SDK 跨区域复制相关](#) Java SDK demo

使用说明

如果您拥有的两个存储空间分属不同地域，可以通过配置同步规则，将源存储空间的数据实时复制到目标存储空间。目前支持以下特性：

- 实时同步数据：数据实时复制，对于数据的增加、删除、修改能够实时同步到目标地域存储空间。对于2MB文件，能够做到分钟级别信息复制，保证两边数据的最终一致。
- 互通同步：您可以配置存储空间 A到存储空间 B的复制，然后配置存储空间 B到存储空间 A的复制，实现双向的互通复制。
- 传递同步：您可以配置存储空间 A到存储空间 B的复制，然后配置存储空间 B到存储空间 C的复制。
- 加密复制：支持复制未加密的对象和使用服务器端加密的对象。

限制说明

- 对于处于同步状态的两个存储空间，源存储空间复制过去的Object可能存在覆盖目标存储空间中同名Object的风险。
- 跨区域复制采用异步（近实时）复制，数据复制到目标存储空间需要一定的时间，通常几分钟到几小时不等，取决于文件的大小。
- 跨区域复制的仅单向、互通和传递这三种复制模式，且一个存储空间不能同时作为多个复制关系的源空间或目的空间，如空间A不能同时复制到空间B和空间C，同时空间 A也不能同时作为空间 B和空间 C的目的端。
- 开启跨区域复制的多个存储空间必须分属不同地域，同地域的存储空间之间不能进行数据复制。
- 每个数据仅能被复制一次，若建议桶A->B->C的复制规则，B中通过CRR复制得到的文件将不会再复制给C。
- 如果用户对目标存储空间内文件被复制完成之前进行主动修改，则复制状态无法被保证。

回源

KS3的回源规则允许请求者向存储空间（Bucket）请求不存在的文件时，从回源规则设置的源站获取目标文件。回源规则分为镜像回源和重定向回源两种，可以满足您对于数据热迁移、特定请求的重定向等需求。

1. 镜像回源

当客户端通过GetObject请求Bucket中的某个文件不存在时，KS3会向回源地址请求这个文件。获取到该文件后，KS3会将文件存入目标Bucket，并返回给用户。具体流程如下图所示。



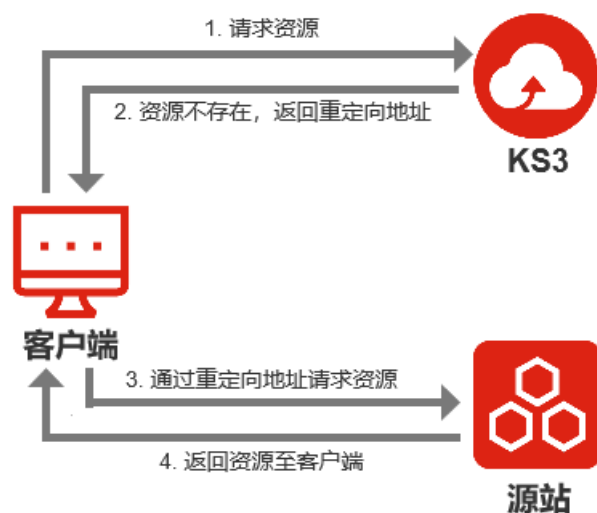
- 使用场景 镜像回源主要用于数据无缝迁移到KS3的场景。例如某服务已经在自己建立的源站或者在其他云产品上运行。现因业务发展，需要将数据迁移到KS3上，但是又不能停止服务，此时可以在迁移数据的同时，使用镜像回源功能保证业务的正常进行。
- 细节分析
 - 同步回源规则触发条件 当客户端Get Object请求的文件不存在时，且用户配置了该文件的同步镜像回源规则，并且该文件满足规则时，触发同步镜像回源。
 - 请求失败返回规则 如果镜像源也不存在此文件，即镜像源返回给KS3的HTTP状态码为404，那么KS3也将返回404给用户。如果镜像源返回给KS3的HTTP status为其他非200的状态码（包括因为网络原因等获取不到文件的错误情况），KS3将返回424给用户，错误码为MirrorFailed。如果源站返回3xx（3XX包含302与301），根据用户的回源规则配置，决定是否follow源站3xx重定向请求获取到资源。
 - 同步回源文件更新规则 如果文件通过镜像回源，存储到了KS3中，之后源站里该文件发生了变化，KS3不会更新已存在于KS3上的该文件。
 - 同步回源文件元信息：

Content-Type

Content-Disposition
Cache-Control
Expires

2. 重定向回源，即原镜像

重定向功能的作用是根据设置的回源条件，以及相应的跳转的配置，向用户返回一个3xx跳转。具体流程如下图所示。



- 使用场景:

- 其他数据源KS3的无缝迁移 用户异步的从自己的数据源向KS3迁移数据，在此过程中未迁移到KS3的数据通过URL rewrite的方式返回给用户一个302重定向请求，用户的客户端根据302中的Location从自己的数据源读回数据。
- 配置页面跳转功能 例如用户希望隐藏自己的某些前缀开头的Object，给访问者返回一个特殊的页面。
- 配置发生404错误时的跳转页面 发生以上错误的时候用户可以看到一个预先设定的页面，不至于系统发生错误的时候向用户完全暴露KS3的错误。