

目录

目录	1
Endpoint与Region的对应关系	6
KS3访问域名使用规则	6
KS3域名构成规则	6
通过外网访问KS3服务	6
通过内网访问KS3服务	6
通过IPV6地址访问KS3	7
KEC实例通过KS3内网地址访问KS3资源	7
获取KS3内网地址	7
同地域KEC实例访问KS3资源	7
存储类型介绍	8
KS3的三种存储类型	8
存储类型对比	8
存储类型转换	8
自动转换Object的存储类型	9
手动转换Object的存储类型	9
注意事项	9
归档存储使用指南	9
开始使用归档存储	9
开通KArchive服务	10
控制台操作指南	10
创建归档存储空间	10
上传文件	10
分享文件	10
删除文件	11
删除存储空间	11
基本操作	11
空间创建	11
操作步骤	11
说明	11
空间概况	11
空间删除	11
操作方式	11
权限设置	12
注意：	12
设置方法	12
原图保护功能	12
生命周期管理	12
使用场景	12
生命周期创建流程	12
注意事项	13
控制台操作流程	13
编辑、删除、停用及启动规则	14
细节说明	14
状态：	14
策略：	14
过期策略	15
约束和限制	15
存储类型转换费用说明	15
生命周期规则匹配逻辑	15

空间策略设置	16
权限判断逻辑	16
方式一：图形化配置Bucket Policy	16
Condition详解	17
方式二：通过策略语法配置Bucket Policy	18
复制设置	19
使用说明	19
使用场景	19
限制说明	19
具体操作	19
开启复制	19
关闭复制	20
日志管理	21
操作步骤	21
日志文件格式	24
注意事项：	25
操作方式	25
CORS配置	25
背景信息	25
操作步骤	26
参数设置说明：	26
操作方式	26
回源	26
防盗链设置	26
防盗链简介	26
操作步骤	26
具体配置说明如下：	27
黑名单/白名单配置规则	27
配置允许Referer为空/不允许Referer为空	27
项目管理	27
创建存储空间（Bucket）时，指定项目组	27
查看存储空间（Bucket）的项目组	28
更改存储空间（Bucket）的项目组	28
按照项目列举存储空间（Bucket）	28
域名绑定	28
使用限制	28
操作步骤	28
验证域名所有权	29
使用说明	29
设置文件名	29
命名规范	29
命名示例	29
操作步骤	30
目录管理	31
操作步骤	31
文件管理	32
文件上传	32
操作步骤：	32
权限设置	32
设置单个文件：	32
批量设置文件权限：	32
文件分享	32

文件权限为私密的情况下，短暂分享文件操作步骤：	32
文件下载	33
操作方法：	33
文件删除	33
删除单个文件	33
批量删除文件	33
删除目录	33
文件解冻	33
文件修改存储类型	33
具体操作如下：	33
修改文件名	34
操作步骤：	34
对象标签	34
使用场景	34
操作步骤	34
单对象操作：	34
批量操作：	35
使用说明	36
对象标签涉及的API接口	36
对象标签涉及的SDK文档：	36
权限说明	36
对象标签和生命周期管理	36
对象搜索	37
操作方法：	37
存储容量	37
查询条件	37
结果展示	37
折线图	37
统计项占比排序	37
存储流量	37
查询条件	37
结果展示	37
带宽	37
查询条件	38
结果展示	38
请求次数	38
查询条件	38
结果展示	38
折线图	38
统计项占比排序	38
数据取回量	38
查询条件	38
结果展示	38
折线图	38
统计项占比排序	38
服务质量	38
查询条件	38
结果展示	38
折线图	38
统计项占比排序	39
业务分析	39
查询条件	39

结果展示	39
权限概述	39
访问策略的元素	39
KS3资源	39
被授权人	39
条件	40
资源操作	40
访问策略	40
术语说明	41
访问控制列表 (ACL)	41
被授权者	41
授予的权限	42
预设ACL权限	42
存储空间预设ACL	42
文件预设ACL	42
操作方式	43
空间策略 (Bucket Policy)	43
指定效果	44
指定资源	44
指定被授权人	44
指定权限	44
指定条件	45
IP地址 (ksc:SourceIp) 支持的条件运算符如下:	45
请求头 (ksc:RequestHeader) 支持的条件运算符如下:	45
VPC子网ID (ksc:SubnetID) 支持的条件运算符如下:	45
操作方式	45
空间策略示例	45
用户策略 (User Policy)	45
在策略中指定资源	46
在策略中指定被授权人	46
在策略中指定权限	46
Service级别	46
Bucket级别	46
Object级别	47
在策略中指定条件	47
用户策略示例	47
访问策略对比	47
使用不同访问策略的准则	47
请求授权	47
账号设置	48
查看方式:	48
敏感操作保护	48
图像处理API	48
图像样式管理	48
云监控概览	48
使用云监控	48
术语表	48
KS3监控项	49
KS3 Bucket Region和云监控endpoint对应表	49
粒度和延迟说明	49
使用Java SDK访问云监控	49
目录	49

开发前准备	50
安装KS3 Java SDK	50
安装云监控Java SDK	50
运行环境	50
Demo演示	50
使用 Go SDK访问云监控	51
目录	51
开发前准备	51
安装KS3 GO SDK	51
安装云监控 GO SDK	51
运行环境	51
初始化配置	51
Demo演示	51
使用Python SDK访问云监控	52
目录	52
开发前准备	52
安装KS3 Python SDK	52
安装云监控Python SDK	52
运行环境	52
初始化配置	53
Demo演示	53

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.ksyuncs.com	ks3-cn-beijing-internal.ksyuncs.com
中国（上海）	SHANGHAI	ks3-cn-shanghai.ksyuncs.com	ks3-cn-shanghai-internal.ksyuncs.com
中国（广州）	GUANGZHOU	ks3-cn-guangzhou.ksyuncs.com	ks3-cn-guangzhou-internal.ksyuncs.com
中国（香港）	HONGKONG	ks3-cn-hk-1.ksyuncs.com	ks3-cn-hk-1-internal.ksyuncs.com
俄罗斯	RUSSIA	ks3-rus.ksyuncs.com	ks3-rus-internal.ksyuncs.com
新加坡	SINGAPORE	ks3-sgp.ksyuncs.com	ks3-sgp-internal.ksyuncs.com
金融专区（北京）	JR_BEIJING	ks3-jr-beijing.ksyuncs.com	ks3-jr-beijing-internal.ksyuncs.com
金融专区（上海）	JR_SHANGHAI	ks3-jr-shanghai.ksyuncs.com	ks3-jr-shanghai-internal.ksyuncs.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目前已上线新域名ksyuncs.com，与原有使用方式相同，不会对业务产生影响。

KS3访问域名使用规则

KS3会为每一个存储空间（Bucket）分配默认的申请域名，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.ksyuncs.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.ksyuncs.com");
Ks3Client 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.ksyuncs.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.ksyuncs.com");
Ks3Client 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

目前仅支持企业实名用户使用IPv6, 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://aaa.ks3-cn-beijing-internal.ksyuncs.com/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.ksyuncs.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.ksyuncs.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.ksyuncs.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

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

KS3的三种存储类型

- 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一定费用，另外，读取低频存储中的数据时，会产生一部分数据取回费用。
- 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提前删除会产生一定费用，另外，读取归档存储中的数据时，会产生一部分数据取回费用。

进入 [KS3控制台](#) 开始使用。

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）
最短存储时间	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各类型存储价格的相关信息，请参见[产品报价](#)。
- 有关金山云KS3不同存储类型接口访问细节，请参见[存储类型](#)。
- 有关金山云KS3解冻接口访问细节，请参见[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，全面覆盖从热到冷的各种数据存储场景，您可以根据您的实际需求，选择合适的文

件存储类型。不同存储类型之间可以相互转换。关于存储类型详细介绍请参见文档：[存储类型介绍](#)。

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](#)。

手动转换Object的存储类型

通过控制台修改文件存储类型时，单次批量操作最多支持30个文件。具体操作步骤如下所示。

1. 进入 [KS3控制台](#) 界面。
2. 选择左侧存储空间，单击目标存储空间名称，进入该存储空间详情页面。
3. 单击 [内容管理](#) 页签。
4. 鼠标选中目标文件后，点击文件列表上方 **批量操作** 中的 **修改存储类型**。

 5. 选择您希望修改的存储类型后，单击 **确定**。



其他方式

- API参考：[PUT Object Copy](#)
- JAVA SDK参考：[Java SDK-修改文件存储类型](#)
- Python SDK参考：[Python SDK-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
 - 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。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。提前转换存储类型或删除这些Object，需要补足到达最短存储期限的剩余天数的存储费用。详情请参见[计费方式说明](#)。
 - 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。
- 归档存储类型的解冻（Restore）时间 归档存储类型的文件需要解冻后才可以修改存储类型，解冻过程需要1分钟到10分钟的时间，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。详情请参见[存储类型](#)。
- 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。

归档存储使用指南

开始使用归档存储

- [开通KArchive服务](#)

控制台操作指南

- [创建归档存储空间](#)
- [上传文件](#)
- [分享文件](#)
- [删除文件](#)
- [删除存储空间](#)

API相关

开始使用归档存储

开始使用归档存储来存储文档，您可以通过以下三种方式：

1. **控制台** 通过浏览器访问[KS3控制台](#)，方便用户进行上传归档存储文件、解冻归档存储文档、存储类型转换等操作。控制台的使用帮助请参考[控制台操作指南](#)。
2. **通过API** 通过调用接口使用归档存储，灵活地满足各种开发需求，详情请参考[存储类型相关API](#)。
3. **通过生命周期规则** 通过生命周期转化，设置存储类型转换规则，系统将按照规则自动将到期的对象存储(KS3)文件沉降至归档存储类型。关联产品[KS3对象存储](#)，详情请参考[生命周期管理](#)。

开通KArchive服务

1. **注册金山云账户** 在使用金山云-归档存储(KArchive)服务之前，请确保您已经注册了金山云账号并完成实名认证。如果您还没有创建金山云账号，系统会提示您注册账号并完成[实名认证](#)，实名认证详细流程参见[实名认证入口](#)。
2. **登录KS3控制台**。

控制台操作指南

创建归档存储空间

1. 登录[KS3控制台](#)，单击左侧导航 **存储空间**。
2. 单击 **创建空间**按钮。
3. 弹出创建空间的弹窗，输入空间的名称（名称必须符合存储空间命名规则；创建后不支持更改存储空间名称）。

存储空间命名规则：

- 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

4. 在 **地区** 框中，下拉选择该存储空间的数据中心。存储空间创建后不支持更换地域。
5. 在 **项目组** 框中，下拉选择该存储空间所在的项目组。在创建后，用户可以根据实际需要重新设置存储空间所在的项目组。
6. 在 **存储类型** 单选框中，选择**归档存储**。
7. 在 **访问控制** 单选框中，选择对应的权限。

说明：

公共读写：任何人（包括匿名访问）都可以向存储空间上传文件和List文件，所有这些操作产生的费用由该存储空间的创建者承担，请慎用该权限。

公共读：只有该存储空间的创建者可以对该存储空间内的文件进行写操作，任何人（包括匿名访问）可以对该存储空间中的文件进行List操作。

私密：只有该存储空间的创建者可以上传文件和List文件，其他人无法访问该存储空间内的文件。

上传文件

1. 登录[KS3控制台](#)，单击左侧导航 **存储空间**，点击一个归档存储空间名称，进入该空间。
2. 进入 **内容管理** 页面，单击上方**上传**按钮。
3. 出现弹窗，可上传一个或多个文件/文件夹。
4. 在 **存储类型** 单选框中，点击选择 **归档**。
5. 在 **访问控制** 单选框中，选择文件的访问控制权限。
6. 点击**确定**，开始上传文件。

分享文件

在分享归档类型的文件前，需要先将其解冻。解冻过程需要约1-5分钟时间。解冻成功后，文件变为解冻状态，可以下载/分享文件。

注意：解冻会产生数据取回费用，详情请参见[计费方式说明](#)。

1. 登录[KS3控制台](#)，单击左侧导航**存储空间**，点击选择一个归档存储空间。
2. 进入 **内容管理** 页面，单击要解冻的文件名，或者单击文件名右侧的 **详情**。
3. 出现文件详情侧边弹窗，单击**解冻**按钮。
4. 出现二次确认弹窗，单击**确定**按钮。

删除文件

1. 登录[KS3控制台](#)，单击左侧导航 **存储空间**，选择一个归档存储空间进入。
2. 进入 **内容管理** 页面，单击要删除的文件名右侧的**更多** > **删除**。也可以批量删除文件，勾选中选择多个或单个要删除的文件，单击上方**批量操作** > **删除**。
3. 出现二次确认弹窗，单击 **确定**。

删除存储空间

1. 登录[KS3控制台](#)，单击左侧导航**存储空间**，单击需要删除的存储空间右侧的**删除**。
2. 出现二次确认弹窗，单击**确定**。

注意：删除存储空间前，需确保存储空间里的所有文件已经删除，如需要删除文件，请参考[删除文件](#)。

基本操作

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。

在使用KS3服务过程中，您可以在控制台查看空间概况进行管理，并在不需要空间时删除该空间。

空间创建

操作步骤

1. 进入 [KS3控制台](#)。
2. 点击左侧**存储空间**。
3. 单击**创建空间**。
4. 在弹出的创建空间对话框中选择**地区**、输入**空间名称**、选择空间所在的**项目组**、**存储类型**、**访问控制**，单击**确定**，空间即创建完成。

说明

- 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。 存储空间的命名规则如下：
 - 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
 - 标准存储和低频存储不需要解冻就可以访问文件；归档存储空间存储长期保存、访问频率极低的文件，需要将文件解冻后，才可以访问。
 - 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](#)。

创建存储空间API详情请参见[PUT Bucket](#)。

空间概况

- 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。
- 进入空间的首页，可看到空间的使用概况。
- 各图表的详细信息，将在报表中心章节做细致介绍。

空间删除

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。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。 具体步骤如下：

1. 进入 [KS3控制台](#)。
2. 点击左侧**存储空间**。
3. 找到要删除的存储空间，点击右侧**删除**。
4. 在弹出的删除空间对话框中，点击**确认**即可删除空间（删除空间之前，需要先删除完空间内的所有文件和配置信息（包括自定义域名、空间策略等））。

删除存储空间API详情请参见[DELETE Bucket](#)。

操作方式

操作方式	说明
Java SDK	丰富、完整的各类语言 SDK demo
Go SDK	
Python SDK	
Node.js SDK	
JavaScript SDK	
PHP SDK	
iOS SDK	
C# SDK	
C/C++ SDK	
Android SDK	

权限设置

权限设置是针对已有的存储空间或对象进行读写权限的设置，每一个存储空间或文件/对象都可以设置独立的访问权限。存储空间三种：公开读写、公开读和私密，文件(对象)的权限都分为公开读和私密。

权限	存储空间	文件/对象
公开读写	所有用户将获得对该空间的读写权限（上传文件、删除文件、list文件），即：允许所有用户列出存储空间中的所有文件/对象，并允许创建、覆盖和删除存储空间中的任意文件/对象。通常不建议在存储空间上授予该权限	不适用
公开读	存储空间所有者将获得公开读写权限（上传文件、删除文件、list文件）；其他用户（包括匿名用户）将获得对该空间的读权限（list文件），即：允许其他用户（包括匿名用户）列出存储空间中的所有文件/对象	所有用户将获得对该对象的读权限（下载文件），即：允许所有用户读取文件/对象数据及其元数据
私密	存储空间所有者将获得公开读写权限（上传文件、删除文件、list文件）；其他用户（包括匿名用户）没有对存储空间的读写权限	文件/对象所有者将获得该文件/对象的读权限（下载文件）；其他用户（包括匿名用户）没有该文件/对象的访问权限

注意：

文件/对象权限不继承空间权限，两者作用范围不同。将存储空间的权限设置为公开，并不意味着存储空间下的文件是公开的，若希望任何人都可访问某文件/对象，需设置文件/对象权限为公开。

文件/对象设置如果设置为**私密**，可以在控制台对文件生成一个临时外链，任何人可以在外链有效时间之内访问文件。

设置方法

登录[KS3控制台](#)，进入需要设置的空间中，点击空间设置页签，空间权限设置位于基础设置中。

原图保护功能

- 打开原图保护后，原图将不能被直接访问，需通过[样式](#)。目前样式只支持图片处理，无法处理其它格式，一旦需要开启此功能，请保持该空间内存储的均为图片。
- 空间的部分文件在访问的过程中会被云加速缓存，最长需要24小时才能失效，需要您主动调用云加速缓存刷新功能清理缓存，但这会影响您客户的访问速度。

生命周期管理

使用场景

在KS3中存储的文件一般会经过新建、使用、归档和删除等一系列生命周期过程。在新建后，文档通常会被频繁访问，随后访问次数逐渐变少，最终会被删除。面对大规模的数据，如果用户手工去维护数据的生命周期，会浪费很多时间人力成本；但如果不去维护，数据始终存放在标准存储里则会不断产生费用。KS3提供生命周期管理帮助用户自动批量完成数据生命周期管理，实现数据从创建到归档到删除的自动流程，从而节约人力和金钱成本。

生命周期管理的适用具体场景有：

- 日志记录管理：可对周期性上传的日志数据设置生命周期，使日志数据定期归档和过期删除；
- 低频访问文档管理：对于在一段时间内经常访问，但是超过一定时间后访问频率非常低的文档，可以设置生命周期，在一定时间后自动转化为低频存储，归档存储或者删除；
- 存档文件管理：金融数据、医疗记录、媒体数据、数据库备份数据等常常需要长期保存，使用生命周期功能对历史版本的文件进行管理；
- 批量删除大规模数据。手动删除对象有数量限制，在桶中配置一条生命周期管理规则，设置定时删除所有文件即可实现批量删除大规模数据；

生命周期创建流程

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。

注意事项

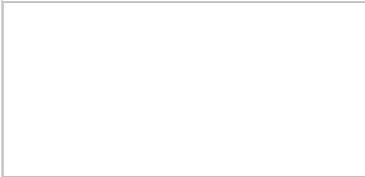
每个存储空间最多可以创建1000条规则。

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。

控制台操作流程

1. 登录[KS3控制台](#)，点击左侧导航栏**存储空间**，进入存储空间列表页。
2. 点击需要设置生命周期的存储空间，点击**空间设置** > **生命周期**。
3. 点击**添加规则**，在对话页中进行规则设置，配置项说明如下：

(1) 状态：可设置规则设置的状态为启用或停用。



(2) 策略：可选择按前缀匹配或配置到整个bucket。

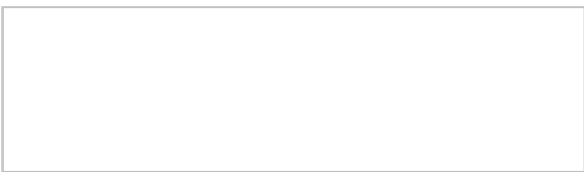
- 如果规则设置到整个bucket上，那么该bucket下所有对象都适用于该规则。



- 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。

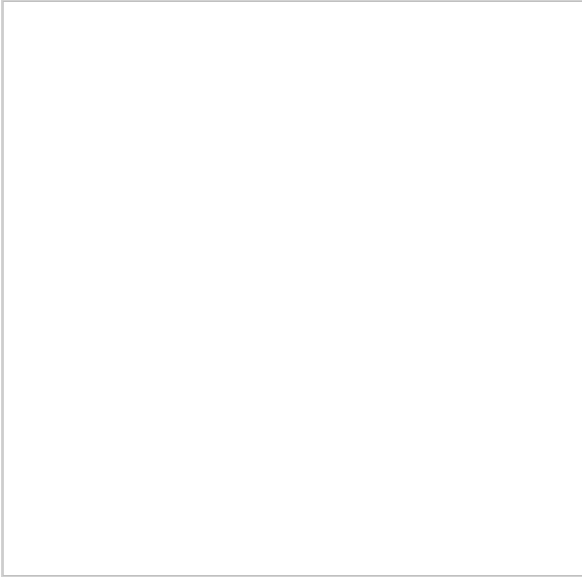


(3) 标签：可针对有特定标签的Object，配置生命周期规则，配置时需要先勾选单选框激活KEY、VALUE输入框，之后输入相应的键、值，输入完毕之后，点击**添加**（若不点击添加则该标签视为废弃），若需设置多个标签同理。

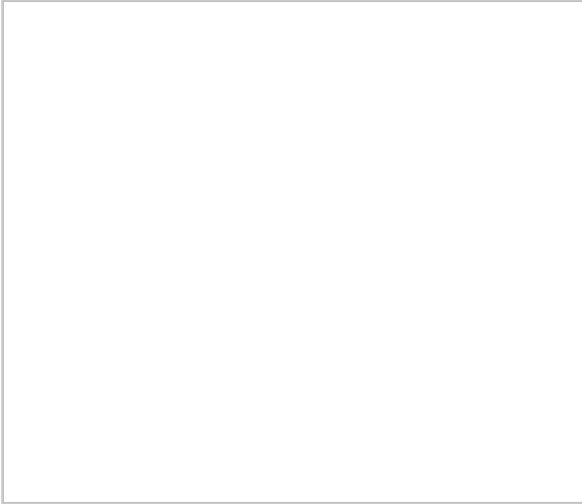


(4) 过期策略：

- 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。



- 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。



4. 点击**确定**，完成生命周期配置，该条规则将在第二天00:00生效。

编辑、删除、停用及启动规则

1. 点击 **空间设置** > **生命周期**
2. 在规则列表右侧点击相应的按钮即可进行编辑规则、删除、停用和启用规则的操作。



API操作方式: [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:

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

1. 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。

2. 指定过期后删除或改变文件的存储类型：

- 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：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。

空间策略设置

当您是空间的所有者时，您可对KS3资源（空间和对象）设置访问控制策略。

其他用户对空间和空间内对象的访问请求都将受到空间策略的控制，这种访问控制表现为接受或拒绝请求。对访问请求是接受还是拒绝，取决于您在空间策略中的描述。

权限判断逻辑

Policy在做权限判断时，每条Statement（策略）会有3种结果，explicit deny、allow和default deny。Bucket Policy对于Policy中的多条Statement采用以下规则进行判定：Bucket Policy对Policy中包含的每条Statement都要进行Explicit Deny、Allow和Default Deny的判断，最终的判决结果遵循explicit deny>allow>default deny的规则：

- 如果没有显式的Deny和Allow，则该条Statement请求权限判别为Default Deny
- 显式的Deny覆盖Allow
- Allow覆盖默认的Default Deny
- Statement的顺序没有影响
- 如果多条Statement判断的最后结果是Default Deny，最终资源是否可以访问，还需要看资源的ACL是否公开。

方式一：图形化配置Bucket Policy

1. 登录[KS3控制台](#)，点击左侧**存储空间**，点击相应的空间名，点击**空间设置 > 空间策略**，点击**添加策略**。
2. 选择配置此条策略的效果是允许还是禁止：

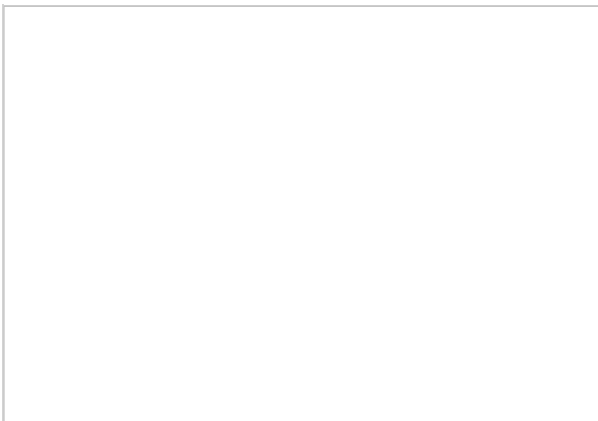
3. 添加被授权人，最多可以添加10条：

注意不同用户类型对应的账号ID填写格式： 1. 所有用户用*表示； 2. 主账号格式accountid； 3. 子用户格式accountid/userName；accountld表示主账号id，userName表示用户用户名 4. 角色名格式accountid/roleName；accountld表示主账号id，roleName表示角色名

4. 选择资源，可以选择整个存储桶或者指定资源：



5. 指定权限:



其中，操作关键字和操作的对应关系可以参考下表：

权限关键字	对应的KS3操作	操作级别
ks3:ListBucket	列举bucket下的文件和查询bucket信息	Bucket 级别
ks3>DeleteBucket	删除当前Bucket	Bucket 级别
ks3:GetBucketAcl	获取Bucket的ACL信息	Bucket 级别
ks3:PutBucketAcl	设置Bucket的ACL信息	Bucket 级别
ks3:GetBucketCORS	获取bucket的CORS配置信息	Bucket 级别
ks3:PutBucketCORS	设置Bucket的CORS	Bucket 级别
ks3ListBucketMultipartUploads	列出分段上传	Bucket 级别
ks3:PutObject	上传文件，包含post、PUT和分块上传	Object 级别
ks3>DeleteObject	删除文件	Object 级别
ks3:GetObject	GET Object 和 HEAD Object	Object 级别
ks3:GetObjectAcl	获取文件的ACL信息	Object 级别
ks3:PutObjectAcl	设置文件的ACL	Object 级别
ks3ListMultipartUploadParts	列出分段	Object 级别
ks3:AbortMultipartUpload	取消分块上传	Object 级别
ks3:PostObjectRestore	解冻归档存储对象	Object 级别

权限关键字	对应的KS3操作	操作级别
ks3:ListBucket	列举bucket下的文件和查询bucket信息	Bucket 级别
ks3>DeleteBucket	删除当前Bucket	Bucket 级别
ks3:GetBucketAcl	获取Bucket的ACL信息	Bucket 级别
ks3:PutBucketAcl	设置Bucket的ACL信息	Bucket 级别
ks3:GetBucketCORS	获取bucket的CORS配置信息	Bucket 级别
ks3:PutBucketCORS	设置Bucket的CORS	Bucket 级别
ks3ListBucketMultipartUploads	列出分段上传	Bucket 级别
ks3:PutObject	上传文件，包含post、PUT和分块上传	Object 级别
ks3>DeleteObject	删除文件	Object 级别
ks3:GetObject	GET Object 和 HEAD Object	Object 级别
ks3:GetObjectAcl	获取文件的ACL信息	Object 级别
ks3:PutObjectAcl	设置文件的ACL	Object 级别
ks3ListMultipartUploadParts	列出分段	Object 级别
ks3:AbortMultipartUpload	取消分块上传	Object 级别
ks3:PostObjectRestore	解冻归档存储对象	Object 级别

注意：不同级别的操作需要指定与之对等的资源。如您想授权ks3:ListBucket操作则其对应的资源必须为bucket，例如：karn:ksc:ks3:::bucket01代表名为bucket01的存储空间；如果您想授权ks3:PutObject操作，则需要指定文件资源，例如：karn:ksc:ks3:::bucket01/*，代表bucket01中的所有文件；需要同时授权Bucket与Object级别权限需要同时指定两种对应的资源。

6. 添加条件，目前支持请求者的IP地址（ksc:SourceIp）、请求头（ksc:RequestHeader）和请求者所在VPC的子网ID作为条件：



Condition详解

- IP地址（ksc:SourceIp）支持的条件运算符如下：

条件运算符	value取值范围	说明
IpAddress	严格的IP地址格式和CIDR格式	只支持IPV4 客户请求的源IP地址是value中指定的IP地址或范围内的地址，策略生效
NotIpAddress	严格的IP地址格式和CIDR格式	只支持IPV4 客户的请求的源IP地址是指定IP地址和范围外的IP地址，策略生效

- 请求头 (ksc:RequestHeader) 支持的条件运算符如下:

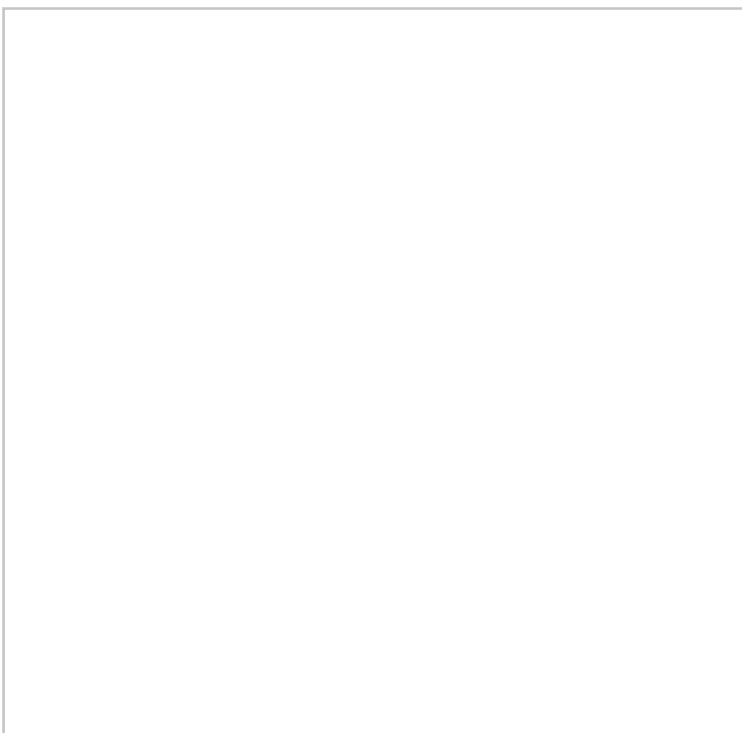
条件运算符	value取值范围	说明
StringEquals	键值对形式的字符串, 如"x-kss-cdn:kingsoftcdn"	请求中带有指定header, 并且请求头的value值恰好匹配 (区分大小写), 策略生效
StringNotEquals	键值对形式的字符串, 如"x-kss-cdn:kingsoftcdn"	请求中带有指定header, 并且请求头的value值不匹配 (区分大小写), 策略生效
StringEqualsIgnoreCase	键值对形式的字符串, 如"x-kss-cdn:kingsoftcdn"	请求中带有指定header, 并且请求头的value值恰好匹配 (不区分大小写), 策略生效
StringNotEqualsIgnoreCase	键值对形式的字符串, 如"x-kss-cdn:kingsoftcdn"	请求中带有指定header, 并且请求头的value值不匹配 (不区分大小写), 策略生效
StringLike	键值对形式的字符串, 可包括字符串中任何一个多字符匹配的通配符 (*) 或单字符匹配的通配符 (?), 如"x-kss-cdn:*"	KS3的请求中带有指定header, 并且请求头的value值可模糊匹配 (区分大小写), 策略生效
StringNotLike	键值对形式的字符串, 可包括字符串中任何一个多字符匹配的通配符 (*) 或单字符匹配的通配符 (?), 如"x-kss-cdn:*"	请求中带有指定header, 并且请求头的value值不模糊匹配 (不区分大小写), 策略生效

- VPC子网ID (ksc:SubnetID) 支持的条件运算符如下:

条件运算符	value取值范围	说明
StringEquals	严格的Account ID格式和SubnetID格式	客户的请求来自value中指定的SubnetID对应的VPC子网, 策略生效
StringNotEquals	严格的Account ID格式和SubnetID格式	客户的请求不是来自value中指定的VPC子网, 策略生效

方式二: 通过策略语法配置Bucket Policy

- 登录[KS3控制台](#), 点击左侧存储空间, 点击相应的空间名, 点击空间设置 > 空间策略, 点击策略语言。
- 在弹出的策略编辑器中编辑Bucket Policy:



下面是一条空间策略示例, 此策略允许所有用户在ip为10.1.1.1的条件下对整个存储桶进行 ks3:listbucket 和 ks3:getobject 操作。

```
{
  "Statement": [
    {
      "Condition": {
        "IpAddress": {
          "ksc:SourceIp": [
            "10.1.1.1"
          ]
        }
      },
      "Resource": [
        "krn:ksc:ks3::examplebucket",
        "krn:ksc:ks3::examplebucket/*"
      ],
      "Principal": {
        "KSC": [
          "*"
        ]
      }
    }
  ]
}
```

```
    "Action": [
      "ks3:GetObject",
      "ks3:ListBucket"
    ],
    "Effect": "Allow"
  }
}
```

3. 点击**确定**即可完成配置。

复制设置

复制功能支持在同地域（Region）与不同地域的存储空间之间自动、异步（近实时）复制文件（Object），将源存储空间（Bucket）中Object的改动（新建、更新、删除等操作）从源存储空间复制到目标存储空间。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。根据复制规则选择的不同可支持同区域复制以及跨区域复制。

使用说明

- 开启跨区域复制功能后，不同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。
- 目前同区域复制功能不收取流量费用。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）之间进行复制。
- 复制功能采用异步（近实时）复制，复制功能开启后，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，会受到当前待复制文件数量、网络、文件大小等因素影响。
- 当前复制功能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**。
- 您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。
 - 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。

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不同副本。

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。
- 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。
- 每个源存储空间只能设置一条跨区域复制规则，但是一个源空间可以同时作为另一个复制规则的目的空间。
- 跨区域复制的数据只会被复制一次，若设置了桶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，同理对于双向复制不会出现数据循环复制的情况。
- 如果用户在目标文件被复制完成之前进行了主动修改，则复制状态无法被保证。
- 跨区域复制规则最多添加10条前缀匹配规则，且前缀之间不能重叠。
- 要删除处于复制关系中的存储空间，必须先关闭该复制关系才能将该存储空间删除。

具体操作

开启复制

1. 进入[KS3管理控制台](#)界面。
2. 点击左侧**存储空间**，单击需要设置的存储空间名称，进入存储空间详情页面。
3. 单击**空间设置**。
4. 单击空间设置的**复制**标签页，并单击**开启同步**按钮。



5. 在弹出的侧边栏中选择复制方式跨区域复制或同区域复制。
6. 在弹出的侧边栏中选择复制的目标区域和目标存储空间。
7. 可以进行数据同步对象、数据同步策略等设置。



8. 设置完毕之后点击确定。

开启完毕之后在空间设置>复制页签下会显示您所设置的信息以及复制情况概览，复制情况概览包括最近复制时间点、最近复制文件信息、待复制文件总数、待复制文件总大小等字段，点击刷新按钮可实现信息的刷新显示。

关闭复制

您可以通过单击空间设置>复制>关闭同步来停止复制功能。



关闭同步后，已复制的数据将被保留在目标存储空间中，源存储空间中在关闭时间点后的增量数据将不再复制到目标端。请谨慎操作。

其他操作方式：[API](#)、[JAVA SDK](#)

日志管理

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](#)。

操作步骤

1. 登录[KS3控制台](#)，点击左侧 **存储空间**，单击目标Bucket名称。
2. 点击 **空间设置** > **基础设置** > **日志状态设置**。
3. 点击 **授权**，跳转至“角色授权页”进行“KsyunKS3BucketLoggingProcessingRole”授权，您必须完成授权后才能开始使用服务。

注意：只有主账号或具有PutBucketLogging和GetBucketLogging权限的子账号才能开通该服务。

图1: 首次开通日志服务



4. 授权成功后, 点击 设置。

图2: 日志服务开通授权



5. 点击 **设置** 后出现如下信息，在“日志存储位置”下拉框内选择日志文件的存储空间，默认存储在当前空间内，也可选择其他存储空间（只支持选择同一Region的存储空间）。在“路径前缀”输入框内输入日志存放的桶内具体路径，默认路径为：KS3-access-log，可进行修改。若当前路径在桶内不存在，会自动生成对应路径文件夹。点击 **保存** 完成日志设置。

图3: 日志存放路径设置



日志文件格式

日志文件的格式组成：以下名称从左至右，以制表符分隔。

字段	记录示例	备注
bucket_owner	NzMOMDEzMzI=	经过base64编码的用户ID
bucket	bucket_test	bucket名
time	[11/Jan/2016:15:05:35 +0800]	时间
remote_ip	219.141.176.229	请求IP

requester	Anonymous	如果请求者是匿名用户，记录值为Anonymous； 如果请求者是主账户，记录值为用户ID的base64编码； 如果请求者是子用户，记录值为用户ID/子用户名的base64编码
request_id	1e6ac821e4904e4ca21d3a9e0ce52e25	请求ID
operation	REST.POST.OBJECT	操作，本例是Post Object
key	-	Object Key
request_url	"POST /chenjin.jpg HTTP/1.1"	请求的URL
http_status	200	请求状态码
error_code	-	错误码
bytes_sent	461	发送的字节数
object_size	139008	被访问资源的大小
total_time	2	请求总时长，单位为秒，精确到毫秒
turn_around_time	2	nginx到api server往返时长，单位为秒，精确到毫秒
referer	"http://ks3.ksyun.com/console.html"	请求的Referer
user_agent	"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36"	用户代理
version_id	-	版本ID，暂不使用
storageclass	Standard	被访问资源的存储类型
usertype	-	用户类型:普通用户默认不显示；cdn用户显示"kc"
nginx_port	80	nginx端口:8000、4433为内网访问；80为外网访问

注意事项:

- 日志延迟时间为一天，即在第二天下午可以看到第一天的日志。
- 日志中的任何一个字段，都可能出现-，用于表示未知数据或对于当前请求该字段无效。
- 根据需求，KS3的日志格式未来会在尾部添加一些新字段，请开发者开发日志处理工具时考虑兼容性的问题。

操作方式

操作方式	说明
Java SDK	丰富、完整的各类语言 SDK demo
Go SDK	
Node.js SDK	
PHP SDK	

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。

背景信息

同源策略是浏览器出于安全考虑而设置的一个限制，是用于隔离潜在恶意文件的关键安全机制。在同源策略的限制下，当A、B两个网站属于不同域时，来自于A网站页面中的JavaScript代码访问B网站时，浏览器会拒绝该访问。同协议、同域名（或IP）、以及同端口视为同域。两个页面的协议、域名和端口（若指定了端口）相同，则视为同源。下表给出了相对http://www.ksyun.com/org/test.html的同源检测示例：

URL	访问是否成功	原因
http://www.ksyun.com/org/other.html	是	协议、域名、端口相同
http://www.ksyun.com/org/internal/page.html	是	协议、域名、端口相同
https://www.ksyun.com/page.html	否	协议不同（HTTPS）
http://www.ksyun.com:22/dir/page.html	否	端口不同（22）
http://help.ksyun.com/dir/other.html	否	域名不同

从上表中可以看出，协议、域名或者端口不同的情况下，浏览器会拒绝该来源的访问。如果要允许这些来源的访问，需要设置跨域资源共享规则。

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，根据规则的设定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](#)。

操作步骤

1. 登录[KS3控制台](#)，点击左侧 **存储空间**，单击目标Bucket名称。
2. 点击 **空间设置 > CORS配置**。



3. 单击 **添加规则**，出现如下添加规则弹窗，根据参数设置说明填写规则，点击 **确定** 完成添加。

参数设置说明：

- Allow Origin 设定允许跨境请求的来源，多个域名以英文逗号分隔，支持通配符星号（*），*每条匹配规则允许使用最多一个星号（*）。*
- Method 设定允许的跨域请求方法，包含：GET、POST、DELETE、PUT、HEAD。
- Allow Header 设定允许的跨域请求Header，多个Header以英文逗号分隔。
- Exposed Header 设定允许从应用程序进行访问的响应头部。
- Cache Time 设定浏览器对特定资源的预取（OPTIONS）请求返回结果的缓存时间。

注：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，根据规则的设定来允许请求并返回相对应的Header。

操作方式

操作方式

[Java SDK](#)
[PHP SDK](#)

说明

丰富、完整的各类语言 SDK demo

回源

防盗链设置

防盗链简介

通过配置防盗链能够避免KS3上的资源被不良网站盗用。

防盗链配置通过黑白名单的方式对访问来源进行控制，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开启或关闭。

HTTP协议中通过表头字段referer可以检测目标网页访问的来源网页，防盗链就是通过检测请求来源的referer字段信息是否与白名单或黑名单匹配来进行防盗操作，若与白名单匹配成功则允许请求访问，否则将阻止请求访问。

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）为空的请求，在KS3中，我们还提供了手动设置referer的选项。

注意：1. Bucket在不同ACL权限下都会检查防盗链设置。 2. 当通过签名URL或者匿名访问Object时，将进行防盗链验证。 3. 当请求的Header中包含Authorization字段，不进行防盗链验证。

操作步骤

1. 登录[KS3控制台](#)，点击左侧存储空间，进入存储空间列表页。
2. 找到需要设置防盗链的存储空间，单击其名称，进入存储空间详情页。
3. 单击**空间设置**页签，点击**防盗链**页签，点击编辑按钮可进入防盗链编辑状态。



4. 选择需要设置名单类型，选择白名单后代表只允许白名单内的域名地址访问存储空间，选择黑名单则代表禁止黑名单内的域名地址访问存储空间。

5. 填写访问域名。



6. 选择空Referer选项。

具体配置说明如下：

黑名单/白名单配置规则

- 若选择配置黑名单，则将拒绝黑名单内的域名访问，黑名单之外的域名允许访问；若选择配置白名单，则只允许白名单内的域名访问，白名单之外的域名将拒绝访问。
- 录入的域名之间用逗号(,)分开，不需要写 http:// 或者 https://
- 支持域名前使用通配符 *，通配符 *表示使用星号代替0个或多个字符,可用于指代当前域名下的多级子域名。例如若Referer白名单配置为*.example.com，则help.example.com、www.example.com等域名均可以访问KS3资源。

配置允许Referer为空/不允许Referer为空

- 若选择“允许Referer为空”，则HTTP请求中Referer头部为空也可以访问KS3中的资源，例如使用浏览器地址栏可以直接访问到该资源；
- 若选择“不允许Referer为空”，则Referer头部为空的HTTP请求中将被禁止访问；

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。一般下面两种情况则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。

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之前，用户需要先维护项目信息，详见文档[访问控制-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。

创建存储空间（Bucket）时，指定项目组

1. 进入 [KS3控制台](#)，点击左侧存储空间，单击**创建空间**。
2. 在弹出的创建空间对话框中选择地区，输入空间的名称、选择空间所在的项目组，单击**确定**，空间即创建完成。



注意：主账户登录后，能创建任何项目下的空间；子用户登录后，只能创建该子用户所在项目下的空间（在授予子用户创建空间的前提下）。

查看存储空间（Bucket）的项目组

选择需要查看的存储空间 > **空间详情**，可以在基本信息中看到存储空间所在的项目组。



注意：主账户登录后，能够查看任何存储空间（Bucket）的项目信息；子用户登录后，只能查看该子用户所在项目下的存储空间（Bucket）的项目信息（在授予子用户查询bucket列表的前提下）。

更改存储空间（Bucket）的项目组

进入需要设置的空间，点击**空间设置** > **基础设置**，点击下方**项目管理**中的**编辑**。选择项目组，点击**保存**。



注意：主账户登录后，能够将存储空间（Bucket）设置为任何项目；子用户登录后，只能将存储空间（Bucket）设置为子用户所在的项目。

按照项目列举存储空间（Bucket）

进入 [KS3控制台](#)，点击左侧存储空间，列表的右侧第二列为项目组（ALL），选择下拉框项目组（ALL），点击需要查看的项目名称。



注意：主账户登录后，能够列举所有项目下的存储空间（Bucket）；子用户登录后，只能列举子用户所在项目的存储空间（Bucket）。

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。

使用限制

- 要绑定的自定义域名需要在工信部备案，否则绑定会失败。
- 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）。
- 如果自定义域名被其它账户的存储空间(Bucket)绑定，需要通过修改域名服务器的TXT记录，进行域名所有权的验证，来删除之前的绑定关系，强制绑定到当前的存储空间（Bucket）。具体操作详见[验证域名所有权](#)。
- 每个Bucket最多可以绑定50个域名。
- 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）。
- 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。
- 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。

操作步骤

1. 登录[KS3控制台](#)，点击左侧 **存储空间**，单击目标Bucket名称。
2. 点击**空间设置** > **域名绑定**。



3. 单击 **添加域名**，出现如下添加域名弹窗，输入要绑定的域名，点击 **确定** 完成提交。

添加域名

需要绑定的域名:

4. 在域名绑定成功后，用户需在自己的域名服务器或服务商的域名服务器上，编辑一条CNAME记录，记录值为要绑定Bucket的外网域名。这样，所有对用户自定义域名的访问都会被发送到 KS3 的存储空间（Bucket）。

验证域名所有权

1. 如果域名被其它账户的存储空间（Bucket）绑定，会提示详情，并出现获取随机串的按钮。

2. 点击按钮，会显示随机生成的TXT记录。

```

```

3. 登录域名服务器，添加对应的TXT记录。

4. 在控制台点击**我已修改TXT记录，重新绑定**，在弹出的对话框点击验证。如果系统检测到域名的TXT记录一致，验证通过，将会把之前的绑定关系解除，同时绑定到当前的存储空间（Bucket）。

使用说明

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.ksyuncs.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

设置文件名

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、控制台进行单个文件上传时使用，对于通过分块上传请求上传的文件，该规则不起作用。

命名规范

文件名规则最多由五个字段组成，至少需要选择一个字段，如选择两个以上字段，则根据对字段的选择顺序进行排序。

字段规则：

- 第一个字段由0-9，a-z和A-Z组成，最多允许20个字符。
- 第二个字段由系统根据当前时间自动生成，格式为YYYY-MM-DD HH:MM:SS, YYYYMMDDHHMMSS, YYYY年MM月DD日HH时MM分SS秒。
- 第三个字段由三位随机数组成。
- 第四个字段由系统自动生成MD5校验位。
- 第五个字段由系统自动生成的Sha1值。
- 各个字段之间由连接符进行链接，包括：-，_，+，*，\。

注意：

当您使用设置文件名功能时，不建议使用以下三种字段组合，因为在文件并发上传文件时，可能出现同名文件被覆盖的风险：

- 仅选择字段1（固定字符串）
- 仅选择字段2（时间）
- 仅选择字段1（固定字符串）与字段2（时间）

命名示例

按下图中所选生成的文件名为：aaa-yyyy年MM月dd日HH时mm分ss秒-三位随机数-MD5校验-SHA1

设置文件名状态: 开启 关闭

文件名组成

字段1: 3/20

字段2: ▼

字段3: 三位随机数

字段4: MD5校验

字段5: SHA1

分隔符

各字段的分隔符: ▼

字段顺序

字段1 字段2 字段3 字段4 字段5

字段1 字段2 字段3 字段4 字段5

文件名示例

示例: aaa-yyyy年MM月dd日HH时mm分ss秒-三位随机数-MD5校验-SHA1

操作步骤

1. 登录[KS3控制台](#)，点击左侧 **存储空间**，单击目标Bucket名称。
2. 点击 **空间设置** > **设置文件名**。您可以通过选择开启和关闭服务，来设定是否自定义文件名。

3. 点击**开启**，出现如下设置文件名信息，根据说明填写字段，点击 **保存** 完成添加

设置文件名状态: 开启 关闭

文件名组成

字段1: 0/20

字段2: ▼

字段3: 三位随机数

字段4: MD5校验

字段5: SHA1

分隔符

各字段的分隔符: ▼

字段顺序

字段1 字段2 字段3 字段4 字段5

文件名示例

示例:

目录管理

您可以在控制台上对目录进行创建操作。

操作步骤

1. 进入 [KS3控制台](#)，点击左侧导航栏中的存储空间，点击您想上传文件的空间名称，点击内容管理 页签，点击创建目录按钮。



2. 在弹出的对话框中输入目录名称并提交，目录创建完成。

创建目录

* 目录名: 0/1023

注: 名称不可以为'..' ,且不准包含以下字符: '/' '@'

注意:

- 目录名称不可以为“..”，且不准包含以下字符：“/”、“@”
- 目录名称不能超过1024字节

文件管理

通用前置操作

- 进入 [KS3控制台](#) 界面。
- 点击左侧 [存储空间](#) 按钮，查看存储空间列表。单击目标存储空间名称，打开该存储空间 [内容管理](#) 页面。

文件上传

操作步骤:

1. 在 [内容管理](#) 页面中点击 [上传](#) 按钮。
2. 弹出上传对话框，支持多文件上传。
3. 选择文件，也可以直接将文件或文件夹拖拽到对话框。
4. 选择文件的存储类型，存储类型介绍，详见 [存储类型介绍](#)。
5. 选择文件的访问控制权限。
6. 点击确认，开始上传文件。

权限设置

设置单个文件:

1. 在 [内容管理](#) 页选定需要设置权限的文件，点击 [更多](#) > [编辑](#) 按钮。
2. 弹出编辑弹窗。
3. 修改文件权限，提交即可生效。

批量设置文件权限:

1. 在 [内容管理](#) 页选择一个或多个文件，点击 [批量操作](#) > [编辑](#) 按钮。
2. 弹出修改对话框。
3. 选择文件的访问控制权限，点击确认后，用户可以看到修改的进度。

文件分享

一般情况下，文件在上传时选择权限为公开的情况下，文件上传完毕后会自动显示文件外部链接，可以直接分享。

文件权限为私密的情况下，短暂分享文件操作步骤:

1. 在 [内容管理](#) 页选定需要分享的文件，点击文件名称，或右侧的 [详情](#)。
2. 设置 [外链地址过期时间](#)。
3. 设置之后，出现可分享的外链地址，点击右侧 [复制](#) 按钮，即可将文件外链复制下来用于分享。

文件下载

操作方法：

在**内容管理**页选定需要下载的文件，点击**下载**按钮，即可完成文件下载。

文件删除

删除单个文件

1. 在**内容管理**页选定需要删除的文件，点击 **更多** > **删除** 按钮。
2. 弹出删除对话框。
3. 点确认即完成删除。

批量删除文件

1. 在**内容管理**页选择一个或多个文件，点击 **批量操作** > **删除** 按钮。
2. 弹出删除对话框。
3. 点确认后，进行删除，用户可以看到删除的进度。

删除目录

删除目录支持遵循以下规则：

1. 一次只能删除一个目录，不支持同时删除多个目录；
2. 不支持同时删除目录和文件；
3. 目录下的文件不可以超过1000个。如果超过1000个，需要使用KS3工具、SDK去删除。

在**内容管理**页选择满足以上条件要删除的目录，点击 **删除** 按钮，在弹出的删除确认框中，点击 **确认** 按钮后，进行删除，用户可以看到删除的进度。

文件解冻

文件解冻操作只限于归档存储文件。

文件上传到归档存储空间，自动存储为归档文件，默认处于冷冻状态，在访问文件之前，需要执行解冻操作，才可以访问文件。

操作如下：

1. 在**内容管理**页选择要解冻的文件，点击右侧的 **详情** 按钮。



2. 在详情页中点击 **解冻** 按钮。



3. 在弹出的确认框，点击 **确认** 按钮。
4. 在提交解冻状态后，文件将处于解冻中的状态，会在原解冻按钮位置处显示**解冻中**，服务端执行解冻。解冻中状态会持续1到10分钟。
5. 待服务端执行完成解冻任务后，会在原解冻按钮位置处显示**解冻完成**，文件进入解冻状态，此时用户可以读取文件。解冻状态默认持续24小时，24小时内调用Restore Object接口则解冻状态会自动延长24小时，最多可延长7天，之后，文件又回到初始时的冷冻状态。

说明：

- 文件解冻会产生数据取回费用，详情请参见[计费方式说明](#)。
- 关于归档存储类型的介绍请参见[归档存储使用指南](#)。

文件修改存储类型

具体操作如下：

1. 在左侧存储空间列表中，单击目标存储空间名称，打开该存储空间概览页面。 2. 单击**内容管理**页签。 3. 鼠标勾选目标文件后，点击文件列表上方的 **批量操作** > **修改存储类型** 。



4. 选择您希望修改的存储类型后，单击 **确定** 。



说明：修改文件存储类型实际是通过覆写操作，将文件修改为指定的存储类型。所以，若修改的文件是低频访问或归档存储类型，且存储未满足指定天数的，补足未满足天数的存储费用，详情请参见[计费方式说明](#)。

修改文件名

操作步骤：

1. 在[内容管理页](#)，找到需要修改名字的文件，[更多](#) > [编辑](#)。
2. 出现弹窗，输入需要修改的文件名。
3. 单击 [确定](#) ，修改完毕。

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，您可以批量管理拥有相同标签的对象，例如针对相同标签的对象设置生命周期规则等。详情请参见[生命周期](#)。

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。

说明：

- 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](#)。
- 单个文件最多可设置10个标签，Key不可重复。

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。

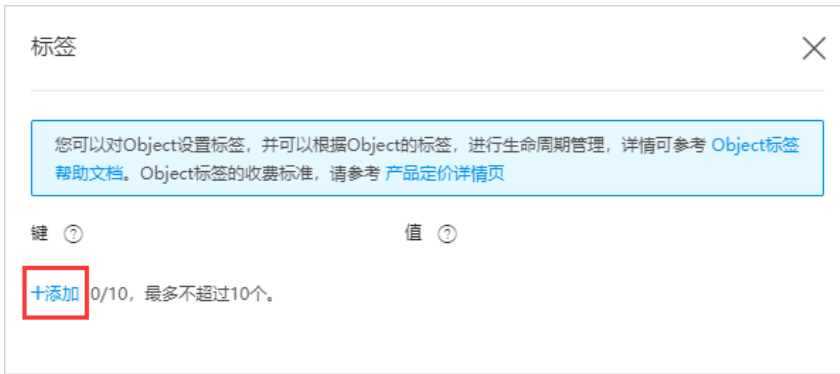
操作步骤

单对象操作：

1. 登录[KS3控制台](#)。
2. 单击左侧[存储空间](#)，之后单击目标Bucket名称。
3. 单击[内容管理](#)页签。
4. 单击Object右侧的[更多](#)，在下拉菜单中单击[标签](#)。



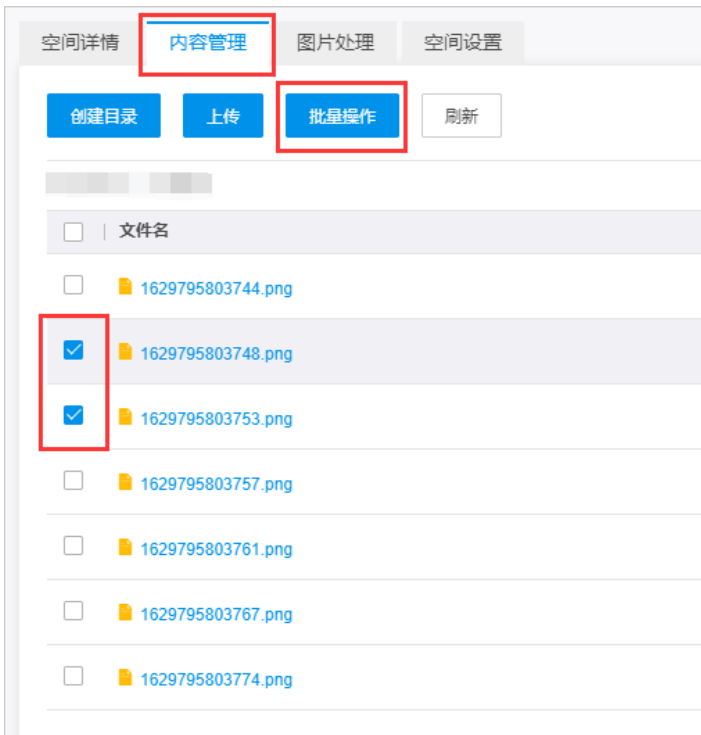
5. 在弹出的[标签](#)页面设置对象标签。



6. 单击**确定**。

批量操作：

1. 登录[KS3控制台](#)。
2. 单击左侧**存储空间**，之后单击目标Bucket名称。
3. 单击**内容管理**页签。
4. 在选中多个想设置标签的文件之后，单击上方的**批量操作**按钮，在下拉菜单中单击**标签**。



注意：在批量管理标签时，若Object已经存在标签，则将覆盖已存在标签，请确定对您的业务无影响再进行操作。

5. 在弹出的**标签**页面设置对象标签。



6. 单击**确定**。

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>
    <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）。

对象搜索

操作方法：

1. 进入 [KS3控制台](#)。
2. 点击左侧导航栏中的**存储空间**，点击您想上传文件的空间名称。
3. 点击**内容管理**页签。
4. 在**内容管理**页面右上角的空间搜索栏中进行对象搜索。

注意：

- 目前空间搜索和对象搜索仅支持资源前缀匹配，不支持通配符和后缀匹配
示例：在新建的空间中输入”example“，返回的检索结果如下图

存储容量

查询条件

包括项目组（project），地区（region），空间（bucket），存储类型和时间。您可以手动选择某个项目中某个region的一个或多个空间，统计周期默认为近30天，可选择近7天、近30天、昨天，您也可以自定义起止日期。

结果展示

折线图

存储容量报表页以折线图形式展现指定时间内的存储总量、新增量以及删除量，您可以点击相应的文字可使对应数据展现或隐藏。点击右侧导出按钮支持以excel格式导出。

注意： 报表页展示的存储容量数据为前一天的数据，每天0点更新。

统计项占比排序

除折线图外，还有各存储空间容量占比排序表，用于展现各存储空间容量占比排序，默认按照占比量从高到低排序，也可选择从低到高排序方式。

存储流量

查询条件

包括项目（project），地区（region），存储空间（bucket），流量类型，存储类型，时间和粒度。您可以手动选择某个项目中某个region的一个或多个空间；统计周期默认为近30天，可选择今天、近7天、近30天、昨天，您也可以自定义起止日期。时间粒度可选天、小时、分钟。

结果展示

存储流量报表页以折线图形式展现指定时间内的CDN、外网、内网、跨区域复制（CRR）的上传流量和下载流量，您可点击相应的文字可使对应的数据展现或隐藏。点击右侧导出按钮支持以excel格式导出。

带宽

查询条件

包括项目组（project），地区（region），空间（bucket），带宽类型，时间和粒度。

您可以手动选择某个项目中某个region的一个或多个空间；统计周期默认为近30天，可选择今天、近7天、近30天、昨天，您也可以自定义起止日期。时间粒度可选天、小时、分钟。

结果展示

带宽报表页以折线图形式展现指定时间内CDN、外网、内网、跨区域复制（CRR）的上传带宽峰值和下载带宽峰值，您可点击相应的文字可使对应的数据展现或隐藏。点击右侧**导出**按钮支持以excel格式导出。

请求次数

查询条件

包括项目组（project），地区（region），空间（bucket），存储类型，时间和粒度。您可以手动选择某个项目中某个region的一个或多个空间；统计周期默认为近30天，可选择近7天、近30天、昨天，您也可以自定义起止日期。时间粒度支持天、小时、分钟。

结果展示

折线图

请求次数报表页以折线图形式展现指定时间内的GET请求数和PUT请求数，点击相应的文字可使对应数据展现或隐藏。点击右侧**导出**按钮支持以excel格式导出。

统计项占比排序

除折线图外，还有三个统计项占比排序表，用于展现各isp占比、各地区占比、各bucket占比排序，默认按照占比量从高到低排序，也可选择从低到高排序方式。

- 各isp占比：统计用户账户下各个isp的请求次数占比排序
- 各地区占比：根据IP统计用户账户下不同区域的请求次数占比排序
- 各bucket占比：统计用户账户下各个空间的请求次数占比排序

数据取回量

支持查询低频存储和归档存储的数据取回量

查询条件

包括项目组（project），地区（region），空间（bucket），存储类型和时间。您可以手动选择某个具体空间，统计周期默认为近30天，您可以选择近7天、近30天、昨天，也可以自定义起止日期。

结果展示

折线图

数据取回量报表页以折线图形式展现指定时间内的低频存储的数据取回量，您可点击相应的文字可使对应数据展现或隐藏。点击右侧**导出**按钮支持以excel格式导出。

统计项占比排序

除折线图外，还有各存储空间容量占比排序表，用于展现各存储空间容量占比排序，默认按照占比量从高到低排序，也可选择从低到高排序方式。

服务质量

查询条件

包括项目组（project），地区（region），空间（bucket），存储类型和时间。您可以手动选择某个项目中某个region的一个或多个空间，统计周期默认为近30天，可选择近7天、近30天、昨天，您也可以自定义起止日期。

结果展示

折线图

服务质量报表页以折线图形式展现指定时间内的服务情况。

统计项占比排序

除折线图外，还有四个统计项占比排序表，用于展现返回各状态码占比、异常返回各Bucket占比、异常返回各isp占比、异常返回各地区占比排序，默认按照占比量从高到低排序，也可选择从低到高排序方式。

业务分析

查询条件

包括项目组（project），地区（region），空间（bucket）统计，排序依据和时间。您可以手动选择某个项目中某个region的一个或多个空间，时间默认为当天，您也可以自定义日期。

业务分析报表，您可以手动选择某个region下的某个具体空间，业务分析报表支持Object\Referer\IP\UA的流量&请求统计。

结果展示

业务分析以表格形式展示指定时间的Object\Referer\IP\UA的流量&请求统计的排序情况。 点击右侧导出按钮支持以excel格式导出。

权限概述

默认情况下，所有 KS3 资源都是私有的，包括存储空间（Bucket）、对象(Object)和相关子资源（例如：Lifecycle 配置和 ACL 配置）。只有资源拥有者，即创建该资源的 KS3 账户可以访问该资源。存储空间的创建者也具有该存储空间下所有资源的访问权限。

资源拥有者可以选择通过编写访问策略授予他人访问权限。授予访问权限，指的是资源拥有者可以决定哪些资源、对什么人、在何种条件下、授予执行什么操作的权限。因此描述一个访问策略，通常包括四个元素：资源、被授权人、条件(非必需)、操作。下面依次说明访问策略的各个元素和具体的访问策略类型。

访问策略的元素

KS3资源

存储空间和对象/文件是KS3的主要资源，他们都有相关联的资源。 存储空间的子资源包括：

- Bucket Policy - 空间策略
- ACL - 存储空间访问控制列表
- Lifecycle - 存储生命周期配置信息
- CORS（跨域资源共享）- 存储空间跨域请求配置
- Logging - KS3的存储空间访问日志

对象/文件的子资源包括：

- ACL - 存储对象访问权限列表。

被授权人

- 账户（主账户）

账户或称主账户是客户在金山云资源归属、资源计量、资源计费的主体。任何客户在使用金山云的服务前，都需要首先注册生成一个金山云账户，一般使用用户名作为账户的登录标识。

账户是其名下所有云计算资源的所有者，拥有名下全部资源的控制权限，拥有资源的订单、账单；云计算资源可被所属账户随意操作访问。

- IAM用户（子用户）

IAM用户是账户下的授权实体，也是归属于账户的一种资源。IAM用户不拥有任何云计算资源、不能独立计量和计费，只能被主账户授权管理其名下的各种资源，其所管理的资源归属于主账户（由主账户付费），且没有独立的账单。

IAM用户在获得主账户的授权后，能够被设置密码和访问密钥，从而登录控制台和使用openAPI管理主账户的资源。

- 角色（Role）

IAM角色是一种虚拟用户（或影子账户），它是IAM用户类型的一种。这种虚拟用户有确定的身份，也可以被赋予一组权限(Policy)，但它没有确定的身份认证密钥（登录密码或AccessKey）。与普通IAM用户的差别主要在使用方法上，IAM角色需要被一个授信的实体用户扮演，扮演成功后实体用户将获得IAM角色的临时安全令牌，使用这个临时安全令牌就能以角色身份访问被授权的资源。

- 关于资源归属的说明

默认情况下，所有KS3资源都是私有的，只有资源拥有者才能访问资源，资源拥有者是指创建资源的KS3账户。

- 创建存储空间(Bucket)和对象/文件(Object)的 KS3 账户拥有这些资源的所有权限
- IAM子用户不拥有任何资源，如果一个IAM子用户再被授权后，上传了一个对象/文件，那么该IAM用户所属的父账户拥有此对象/文件。
- 存储空间拥有者可以向其他KS3账户授予上传对象的跨账户权限。在这种情况下，上传对象的KS3账户拥有这些对象。存储空间拥有者

对其他账户拥有的对象也同时拥有所有权限，账单也是由存储空间拥有者支付。

条件

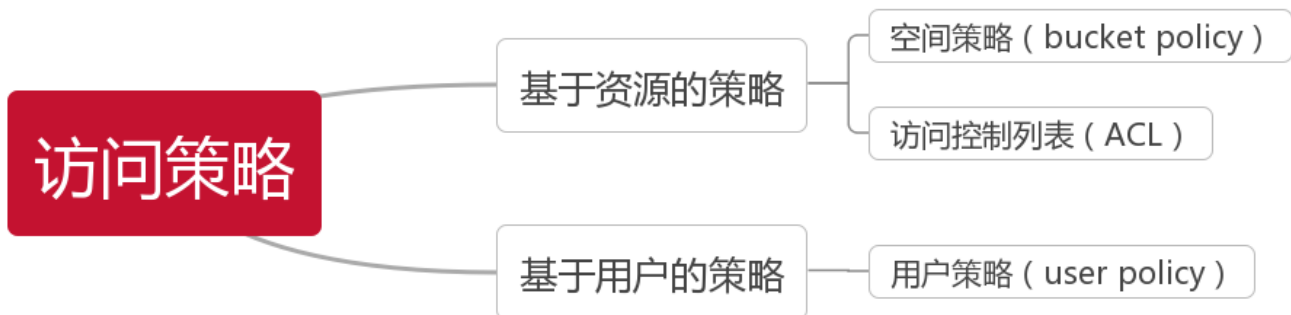
在授予权限时指定的条件。访问请求只有在满足指定条件时，访问策略才可以生效。KS支持指定IP地址、指定请求中带有请求头等条件设置。

资源操作

KS3 提供一系列针对KS3资源的API操作，详情请见[API概览](#)。

访问策略

访问策略分为基于资源的策略和用户策略两类。附加到资源（存储空间和文件）的访问策略称为基于资源的策略，例如，存储空间策略（bucket policy）和访问控制列表（ACL）就是基于资源的策略；访问策略附加到您账户中的用户，这些策略称为用户策略（user policy）。



- **ACL访问策略：** 每一个存储空间和对象都关联一个ACL。ACL可以向其他的KS3账户授予基本的读写权限，ACL使用XML格式来表示，以下的例子指定存储空间的拥有者授予另外一个用户Read权限。

```

<AccessControlPolicy>
<Owner>
  <ID>Owner-User-Id</ID>
  <DisplayName>Owner-User-Name</DisplayName>
</Owner>
<AccessControlList>
  <Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
      <ID>User2-Id</ID>
      <DisplayName>User2-Name</DisplayName>
    </Grantee>
    <Permission>READ</Permission>
  </Grant>
  ...
</AccessControlList>
</AccessControlPolicy>
  
```

- **空间策略 (Bucket Policy)：** 用户可以在存储空间设置空间策略，向其他KS3账户或者IAM子用户（包含本账户下的子用户和其他账户下的子用户）授予存储空间以及其中文件/对象的权限。空间策略可以很好补充ACL访问策略，授权的权限更多。空间策略使用JSON文件来表示。下面为一个空间策略的例子，允许账户ID为11123的用户可以对mybucket和mybucket下所有文件做任何操作。

```

{
  "Version": "2008-10-17",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"KSC": ["krc:ksc:iam::11123:root"]},
    "Action": ["ks3:*"],
    "Resource": ["krc:ksc:ks3::mybucket", "krc:ksc:ks3::mybucket/*"]
  }]
}
  
```

- **用户策略 (User Policy)：**

用户可以使用 身份与访问控制中心 (Identity and Access Mutual, 简称:IAM) 管理对 KS3 资源的访问权限。使用 IAM, 用户可以在主账户中创建 子用户 (IAM 用户)、组和角色, 并通过附加访问策略授予它们对KS3资源的访问权限。

```

{
  "Version": "2015-11-01",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["ks3:*"],
    "Resource": ["krc:ksc:ks3::mybucket", "krc:ksc:ks3::mybucket/*"]
  }]
}
  
```


}

术语说明

术语	全称	中文	说明
Account	Member Account	账户（账号）	指一个金山云账户，账户是资源的拥有者，任何资源均隶属于某个账户
User	IAM User	IAM用户	指一个金山云账户下的IAM用户（又称：子用户），子用户不拥有资源，能够操作主账户授予权限的资源
Role	IAM Role	角色	指一个金山云账户下创建的角色
Group (IAM)	IAM Group	用户群组	金山云多个相同职能的用户（IAM用户的集合）
AccountId	Ksyun Account Id	金山云账户ID	指一个金山云账户的ID，为最长20位的数字标识 访问策略选项大致可分为基于资源的策略和用户策略两类
Policy	Policy	访问策略	<ul style="list-style-type: none"> 基于资源的策略：附加到资源（存储桶和对象）的访问策略，如空间策略和访问控制列表（ACL） 用户策略：访问策略附加到账户中的用户
ACL	Access Control List	访问控制列表	KS3使用ACL（访问控制列表）方便用户管理Bucket和Object的访问权限，每个Bucket和Object都拥有一个ACL
Bucket Policy	Bucket Policy	空间策略	可以通过添加空间策略向其他主账户或IAM子用户授予对相应存储空间（Bucket）及其中文件（Object）的权限。存储桶策略相比ACL更加灵活，授权的粒度更细
User Policy	User Policy	用户策略	可以在账户中创建IAM子用户、用户组和角色，并通过附加访问策略授予它们对包括KS3在内的所有金山云资源的访问权限
KRN	Kingsoft Resource Name	金山云资源名称	指金山云全局唯一的资源标识名称，用于在策略文档中标识资源，形如krn:ksc::::/。例如：krn:ksc:iam::(11233):user/peter，表示11233账户下的peter子用户
AK	AccessKeyId	访问密钥ID	指一个访问密钥的唯一标识，标识访问者的身份
SK	SecretAccessKey	私有访问密钥	用于加密签名字符串和KS3用来验证签名字符串的密钥，需要用户保管好，不能泄露。
STS	Security Token Service	安全凭证（Token）管理系统	用来授予临时的访问权限

访问控制列表（ACL）

KS3使用ACL（Access Control List：访问控制列表）方便用户管理存储空间（Bucket）和文件（Object）的访问权限，每个Bucket和Object都拥有一个ACL，可以通过设置ACL定义向哪些用户授予哪些访问权限。

当KS3收到针对某个资源的请求后，将检查相应的ACL以验证请求者是否拥有所需的访问权限。

ACL使用XML格式来表示，示例：

```
<AccessControlPolicy>
  <Owner>
    <ID>Owner-User-Id</ID>
    <DisplayName>Owner-User-Name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>User2-Id</ID>
        <DisplayName>User2-Name</DisplayName>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    ...
  </AccessControlList>
</AccessControlPolicy>
```

示例中Owner元素中需要设置资源拥有者的账户ID和用户名，Grant元素中需要设置被授权人的账户ID和用户名，以及所授予的权限。下面依次对被授权人、授予的权限、预设ACL的方式以及KS3支持操作ACL的方式进行说明。

被授权者

被授权者可以是KS3账户也可以是所有用户（包括匿名用户），但不能是IAM子用户。

- **KS3账户**：在授权语句中，使用账户ID和用户名来表示被授权的KS3账户。

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser"><ID>ID</ID><DisplayName>GranteesEmail</DisplayN
ame>
</Grantee>
```

- **所有用户（包含匿名用户）**：在授权语句中，使用<http://acs.ksyun.com/groups/global/AllUsers>表示授权所有用户。

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group"><URI>http://acs.ksyun.com/groups/global/AllUsers</URI></Grantee>
```

授予的权限

下表列出了KS3在ACL中支持的权限集。

ACL权限	对存储空间授权	对文件授权
READ	允许被授权者列出存储空间中的文件	允许被授权者读取文件数据及其元数据
WRITE	允许被授权者创建、覆盖和删除存储空间（Bucket）中的任意文件	不适用
FULL_CONTROL	授予被授权者在存储空间（Bucket）上的READ、WRITE 权限	授予被授权者在文件上的READ权限

注意：1、对于文件ACL和存储空间ACL，ACL权限集相同，但这些ACL权限授予允许被授权者进行的具体操作不同，例如存储空间ACL的READ权限允许被授权者列出存储空间中的文件，而文件ACL的权限则是允许被授权者读取文件数据及其元数据。

2、文件ACL不支持WRITE权限。

ACL的每个权限允许一个或多个 KS3 操作，例如：文件的读权限，包含了Head Object和GET Object等KS3操作。下表展示了每个ACL权限允许的具体操作。ACL 主要用于授予基本读/写权限，这与文件系统权限类似。

ACL权限	在Bucket授予ACL权限时允许的操作	在Object授予ACL权限时允许的操作
READ	List Bucket, List Multipart Upload	Get Object, Head Object, List Parts
WRITE	Put Object, Post Object, Put Object Copy, Upload Part Copy, Delete Object, Initiate Multipart Upload, Upload Part, Complete Multipart Upload, Abort Multipart Upload	不适用
FULL_CONTROL	等同于授予READ和WRITE ACL权限	等同于授予READ ACL权限。

预设ACL权限

KS3支持在创建桶或上传对象时通过头域设置桶或对象的权限控制策略，这种方式设置的权限称为预设ACL。使用预设 ACL 时，需要在 PUT Bucket/Object 或 PUT Bucket/Object acl 中携带相应的头部，这些头部中x-kss-acl只能对所有用户进行授权，x-kss-grant-read、x-kss-grant-write和x-kss-grant-full-control可以更细粒度地对指定用户授权。

• 存储空间预设ACL

存储空间预设ACL可选择的头部有4个，描述和示例见下表：

头域	描述	示例
x-kss-acl	设置任何用户（包括匿名用户）的操作权限，取值包括：private、public-read、public-read-write，默认值为private； <ul style="list-style-type: none"> private: 私有，空间所有者拥有所有的访问权限。其他人没有访问权限（默认）； public-read: 公开读，空间所有者拥有所有的访问权限，任何用户（包括匿名用户）都拥有列举空间下所有文件的权限； public-read-write: 公开读写，任何用户（包括匿名用户）都可以向存储空间写入文件、删除文件、列举空间下所有文件的权限； 	指定任何用户（包括匿名用户）都拥有列举空间下所有文件的权限； <ul style="list-style-type: none"> x-kss-acl:public-read
x-kss-grant-read	为若干用户授予READ权限	给id为1234578和3344211的两个用户授予READ权限： <ul style="list-style-type: none"> x-kss-grant-read:id=“1234578”,id=“3344211”
x-kss-grant-write	为若干用户授予WRITE权限	给id为1234578和3344211的两个用户授予WRITE权限： <ul style="list-style-type: none"> x-kss-grant-write:id=“1234578”,id=“3344211”
x-kss-grant-full-control	为若干用户授予FULL_CONTROL权限	给id为1234578和3344211的两个用户授予FULL_CONTROL权限： <ul style="list-style-type: none"> x-kss-grant-full-control:id=“1234578”,id=“3344211”

• 文件预设ACL

文件预设ACL可选择的头部有3个，描述和示例见下表：

头域	描述	示例
----	----	----

设置任何用户（包括匿名用户）的操作权限，取值包括：private、public-read，默认值为private：

x-kss-acl

- private: 私有，文件所有者拥有所有的访问权限。其他人没有访问权限（默认）；
- public-read: 公开读，所有用户（包括匿名用户）拥有文件的访问权限；

指定任何用户（包括匿名用户）都拥有文件的访问权限：

- x-kss-acl:public-read

x-kss-grant-read 为若干用户授予READ权限

给id为1234578和3344211的两个用户授予READ权限：

- x-kss-grant-read:id=“1234578”,id=“3344211”

xx-kss-grant-full-control 为若干用户授予FULL_CONTROL权限

给id为1234578和3344211的两个用户授予FULL_CONTROL权限：

- x-kss-grant-full-control:id=“1234578”,id=“3344211”

操作方式

支持控制台和API两种操作方式：	操作方式	参考文档
控制台	<ul style="list-style-type: none"> • 控制台空间管理-权限设置 • 控制台内容管理-文件管理 <p>用户在创建存储空间或上传对象时：用户可以在请求头中通过设置以下的请求头来指定ACL，详情请见：</p> <ul style="list-style-type: none"> • PUT Bucket API • PUT Object API 	
API	<p>用户在已有资源上设置ACL：可以在请求头或者在请求内容中中设置，详情请见：</p> <ul style="list-style-type: none"> • PUT Bucket ACL API • PUT Object ACL API 	
SDK	<ul style="list-style-type: none"> • JAVA • PHP • Python • Android • IOS • JavaScript • Node.js • C# • Go 	

空间策略（Bucket Policy）

空间策略作用于所配置的KS3空间及空间内对象, 包含以下几个元素：

- **效果（Effect）**：代表本条策略的效果，可以是允许或拒绝，对应Effect的值为Allow或者Deny。该元素是必填项。
- **资源（Resource）**：代表本条策略针对哪些资源起作用，可以设置为存储空间(Bucket)、对象(Object)以及相对应的子资源。该元素是必填项。
- **被授权人（Principal）**：代表授权给哪些用户和账户。该元素是必填项。
- **权限（Action）**：代表对于指定的资源授权了哪些权限。该元素是必填项。
- **条件（Condition）**：在授予权限时指定的条件。访问请求只有在满足指定条件时，访问策略才可以生效。该元素是非必填项。

下面是一条空间策略示例, 此策略允许账户accountid下的子用户Dave在ip为101.226.XXX.185的条件下对examplebucket存储桶具有ks3:listbucket 和 ks3:getobject 的权限。

```
{
  "Version": "2015-11-01",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {"krc:ksc:iam::(accountid):user/Dave"},
      "Action": [
        "ks3:ListBucket",
        "ks3:GetObject"
      ],
      "Resource": ["krc:ksc:ks3::examplebucket", "krc:ksc:ks3::examplebucket/*"],
      "Condition": {
        "IpAddress": {
          "ksc:SourceIp": ["101.226.XXX.185"]
        }
      }
    }
  ]
}
```

```

    }
  }
}

```

下面依次对空间策略包含的元素的配置方式进行说明。

指定效果

包含允许 (Allow) 和拒绝 (Deny) 两种授权效果，允许 (Allow) 代表允许设定的用户对指定资源的访问权限，拒绝 (Deny) 可确保设定的用户无法访问指定资源，即使有其他策略授予了访问权限的情况下也是如此。

指定资源

在KS3中，使用资源名称 (KRN) 来标识资源，KRN的格式如下：

```

krn:ksc:ks3:::bucketname
krn:ksc:ks3:::bucketname/keyname

```

其中bucketname表示存储空间名称，keyname表示对象/文件 (Object) 的名称，支持通配符* (匹配多个字符) 和? (匹配单个字符)。

以下为 KS3 资源KRN的示例：

- 表示 mybucket 存储空间

```
krn:ksc:ks3:::mybucket
```

- 表示 mybucket存储空间中全部对象

```
krn:ksc:ks3:::mybucket/*
```

- 表示您拥有的所有存储空间与对象

```
krn:ksc:ks3:::*
```

注意：

- 在实际使用中，为了简化输入，在控制台写入Resource时，只需要写入空间名称和文件名称，可省略前面的krn:ksc:ks3:::。后台程序会自动将简化输入转换为标准输入。
- Bucket与Object级别的操作需要对应与之相匹配的资源表示，详见[指定权限](#)。

指定被授权人

Principal 元素用于指定被允许或拒绝访问资源的子用户、账户、服务或其他实体。元素 Principal 仅在空间策略中起作用；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。

- 1、*表示所有用户。
- 2、krn:ksc:iam::(accountid):root //标识主账号，accountid是主账户id，root是固定值。
- 3、krn:ksc:iam::(accountid):user/(userName) userName表示子账号，其中user是固定值，accountid是归属主账户id。
- 4、krn:ksc:iam::(accountid):role/(ruleNAME) ruleNAME表示角色名称，其中role是定值，accountid是归属主账户id。

注意：在实际使用中，为了简化输入，在控制台写入Principal时，对于主账户只需要填入主账户ID即可；对于子用户，写入格式为“accountid/userName”，其中accountid为用户主账号ID，userName为子用户名。后台程序会自动将简化输入转换为标准输入。

指定权限

KS3在策略中指定的一组权限。每一个权限，都会映射到特定 KS3 操作。

权限关键字	对应的KS3操作	操作级别
ks3:ListBucket	列举bucket下的文件和查询bucket信息	Bucket 级别
ks3>DeleteBucket	删除当前Bucket	Bucket 级别
ks3:GetBucketAcl	获取Bucket的ACL信息	Bucket 级别
ks3:PutBucketAcl	设置Bucket的ACL信息	Bucket 级别
ks3:GetBucketCORS	获取bucket的CORS配置信息	Bucket 级别
ks3:PutBucketCORS	设置Bucket的CORS	Bucket 级别
ks3:ListBucketMultipartUploads	列出分段上传	Bucket 级别
ks3:PutObject	上传文件，包含post、PUT和分块上传	Object 级别
ks3>DeleteObject	删除文件	Object 级别
ks3:GetObject	GET Object 和 HEAD Object	Object 级别
ks3:GetObjectAcl	获取文件的ACL信息	Object 级别
ks3:PutObjectAcl	设置文件的ACL	Object 级别
ks3:ListMultipartUploadParts	列出分段	Object 级别
ks3:AbortMultipartUpload	取消分块上传	Object 级别
ks3:PostObjectRestore	解冻归档存储对象	Object 级别

ks3:PutObjectTagging	添加/更新对象的标签	Object 级别
ks3:GetObjectTagging	查询对象的标签	Object 级别
ks3>DeleteObjectTagging	删除对象的标签	Object 级别

注意： 不同级别的操作需要指定与之对等的资源。 1. 如您想授权ks3:ListBucket操作则其对应的资源必须为bucket，例如：karn:ksc:ks3:::bucket01代表名为bucket01的存储空间； 2. 如果您想授权ks3:PutObject操作，则需要指定文件资源，例如：karn:ksc:ks3:::bucket01/*,代表bucket01中的所有文件； 3. 需要同时授权Bucket与Object级别权限需要同时指定两种对应的资源。

指定条件

访问策略语言支持在授予权限时指定该策略生效的条件。在Condition 元素（或 Condition 块）中，可以指定策略生效的条件。Condition 元素是可选的。

如果在策略中指定了条件，那么用户的请求必须满足条件，策略才可以生效。

KS3支持的Condition如下：

Condition	功能
ksc:SourceIp	指定IP地址
ksc:RequestHeader	指定请求中带有请求头
ksc:SubnetID	指定请求来自SubnetID对应的VPC子网

IP地址（ksc:SourceIp）支持的条件运算符如下：

条件运算符	value取值范围	说明
IpAddress	严格的IP地址格式和CIDR格式，只支持IPv4	客户请求的源IP地址是value中指定的IP地址或范围内的地址，策略生效
NotIpAddress	严格的IP地址格式和CIDR格式，只支持IPv4	客户的请求的源IP地址是指定IP地址和范围外的IP地址，策略生效

请求头（ksc:RequestHeader）支持的条件运算符如下：

条件运算符	value取值范围	说明
StringEquals	键值对形式的字符串，如"x-kss-cdn:kingsoftcdn"	请求中带有指定header，并且请求头的value值恰好匹配（区分大小写），策略生效
StringNotEquals	键值对形式的字符串，如"x-kss-cdn:kingsoftcdn"	请求中带有指定header，并且请求头的value值不匹配（区分大小写），策略生效
StringEqualsIgnoreCase	键值对形式的字符串，如"x-kss-cdn:kingsoftcdn"	请求中带有指定header，并且请求头的value值恰好匹配（不区分大小写），策略生效
StringNotEqualsIgnoreCase	键值对形式的字符串，如"x-kss-cdn:kingsoftcdn"	请求中带有指定header，并且请求头的value值不匹配（不区分大小写），策略生效
StringLike	键值对形式的字符串，可包括字符串中任何一个多字符匹配的通配符(*)或单字符匹配的通配符(?),如"x-kss-cdn:*"	KS3的请求中带有指定header，并且请求头的value值可模糊匹配（区分大小写），策略生效
StringNotLike	键值对形式的字符串，可包括字符串中任何一个多字符匹配的通配符(*)或单字符匹配的通配符(?),如"x-kss-cdn:*"	请求中带有指定header，并且请求头的value值不模糊匹配（不区分大小写），策略生效

VPC子网ID（ksc:SubnetID）支持的条件运算符如下：

条件运算符	value取值范围	说明
StringEquals	严格的AccountID格式和SubnetID格式	客户的请求来自value中指定的SubnetID对应的VPC子网，策略生效
StringNotEquals	严格的AccountID格式和SubnetID格式	客户的请求不是来自value中指定的VPC子网，策略生效

操作方式

支持控制台和API两种操作方式：	操作方式	参考文档
控制台	空间策略	
API	<ul style="list-style-type: none"> Put Bucket Policy Get Bucket Policy Delete Bucket Policy 	
SDK	<ul style="list-style-type: none"> Python Android 	

空间策略示例

用户策略(User Policy)

用户策略（User Policy）是基于用户的授权策略。通过设置用户策略，您可以集中管理您的用户，以及控制用户可以访问您名下哪些资源的权限，用户策略包含以下几个元素：

- **Effect**: 当用户请求特定操作（可以是允许或拒绝）时的效果。如果没有显式授予（允许）对资源的访问权限，则说明此条策略不生效。也可显式拒绝对资源的访问，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。
- **资源**: 存储空间(Bucket)、对象(Object)以及相对应的子资源。
- **操作**: 对于每个资源，所支持的一组操作。
- **指定条件**: 在授予权限时指定的条件。访问请求只有在满足指定条件时，访问策略才可以生效。

下面是一条用户策略示例。此策略允许对 examplebucket 存储桶具有ks3:listbucket和ks3:getobject的权限。创建完此策略后，将策略授权给用户，子用户就拥有了相对应的权限。其他示例请参阅[用户权限示例](#)。

```
{
  "Version": "2015-11-01",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Action": [
        "ks3:ListBucket",
        "ks3:GetObject"
      ],
      "Resource": ["krn:ksc:ks3::examplebucket", "krn:ksc:ks3::examplebucket/*"]
    }
  ]
}
```

注意：和空间策略(Bucket policy)不同，用户策略中不必指定被授权人，因为用户策略直接附加到特定用户。

在策略中指定资源

在KS3中，使用资源名称（KRN）来标识资源，KRN的格式如下：

```
krn:ksc:ks3::bucketname
krn:ksc:ks3::bucketname/keyname
```

其中bucketname表示存储空间名称，keyname表示对象/文件（Object）的名称，支持通配符*（匹配多个字符）和?（匹配单个字符）。

以下为 KS3 资源KRN的示例：

- 该KRN表示 mybucket 存储桶

```
krn:ksc:ks3::mybucket
```

在策略中指定被授权人

用户策略中不必指定，因为用户策略直接附加到特定用户。如何将自定义策略授权给IAM子用户或用户组或角色，请参照[访问控制](#)。

在策略中指定权限

KS3在策略中指定的一组权限。每一个权限，都会映射到特定 KS3 操作。权限分为Service级别操作、Bucket级别操作以及Object级别的操作。各API接口功能描述的更多信息，请参见[API概览](#)。

Service级别

Service级别操作对应的是ks3:ListBuckets，用来列举所有属于该用户的Bucket列表。

权限关键字 对应的KS3操作

ks3:ListBuckets 查询Bucket列表

注意：ListBuckets是服务级别的操作，与Bucket级别ListBucket不同，ListBuckets能查看到所有的Bucket列表，但不能看到Bucket中具体的Object内容，而ListBucket可以看到Bucket下的Object信息。

Bucket级别

Bucket级别操作的对象均为Bucket，权限关键字与对应的操作如下：

权限关键字	对应的KS3操作
ks3:GetBucketLocation	查询Bucket归属region
ks3:PutBucket	新建Bucket
ks3>DeleteBucket	删除Bucket
ks3:ListBucket	列举Bucket下的文件和查询bucket信息
ks3:GetBucketAcl	获取Bucket的ACL信息
ks3:PutBucketAcl	设置Bucket的ACL信息
ks3:PutBucketPolicy	创建Bucket的空间策略信息
ks3:GetBucketPolicy	获取Bucket的空间策略信息
ks3>DeleteBucketPolicy	删除Bucket的空间策略信息
ks3:GetBucketCORS	获取Bucket的CORS配置信息
ks3:PutBucketCORS	设置Bucket的CORS
ks3>DeleteBucketMirror	删除（清空）Bucket回源规则

ks3:GetBucketMirror
ks3:PutBucketMirror

获取Bucket的CORS配置信息
设置Bucket回源规则

Object级别

Object级别操作的对象均为Object, 权限关键字与对应的操作如下: 权限关键字 对应的KS3操作

ks3:PutObject	上传文件, 包含post、PUT和分块上传
ks3>DeleteObject	删除文件
ks3:GetObject	GET Object 和 HEAD Object
ks3:GetObjectAcl	获取文件的ACL信息
ks3:PutObjectAcl	设置文件的ACL
ks3:ListBucketMultipartUploads	列出分段上传
ks3:ListMultipartUploadParts	列出分段
ks3:AbortMultipartUpload	取消分块上传
ks3:PutBucketLifecycle	创建生命周期管理规则
ks3>DeleteBucketLifecycle	删除生命周期管理规则
ks3:GetBucketLifecycle	查看生命周期管理规则
ks3:PostObjectRestore	解冻归档文件
ks3:PutObjectTagging	添加/更新对象的标签
ks3:GetObjectTagging	查询对象的标签
ks3>DeleteObjectTagging	删除对象的标签

在策略中指定条件

访问策略语言可使您在授予权限时指定条件。在 Condition 元素（或 Condition 块）中，可以指定策略生效的条件。

KS3支持的Condition如下：

Condition 功能

ksc:SourceIp 指定IP地址

用户策略示例

访问策略对比

策略类型	ACL	空间策略 (Bucket Policy)	用户策略 (User Policy)
可支持的资源操作	基于资源的策略 较少, 只支持简单的读写操作	基于资源的策略 较多, 但不支持service操作, 如不支持查询bucket列表	基于用户的策略 最多, 支持service操作, 如: ks3:ListBuckets (查询bucket列表)。
是否支持授权给其它账户 (账号)	支持	支持	不支持, 但可通过创建角色, 选择授信账户来实现跨账户的授权。
是否支持授权给IAM子用户	不支持	支持	支持
是否支持授权给角色	不支持	支持	支持

使用不同访问策略的准则

- 建议使用ACL的场景
 - 如果不需要实现复杂的授权逻辑, 只是简单的设置存储空间 (Bucket) 和文件 (Object) 公开还是私密, 建议使用ACL。
- 建议使用Bucket Policy的场景
 - 如果想方便的将资源细粒度的授权给其它账户, 实现跨账号访问, 推荐使用存储空间策略 (Bucket Policy)。
 - 如果想将资源细粒度的授权给IAM子用户, IAM子用户不需要登录控制台, 可以使用存储空间策略 (Bucket Policy)。
- 建议使用User Policy的场景
 - 如果想将资源授权给主账户下不同的IAM子用户, IAM子用户需要登录控制台, 需要使用用户策略 (User Policy)。
 - 如果希望让IAM子用户扮演角色, 使用临时权限, 需要使用用户策略 (User Policy)。


请求授权

当 KS3 收到请求, 例如, 存储空间 (Bucket) 或文件 (Object) 操作时, 它首先验证请求者是否拥有必要的权限。

KS3 对所有相关访问策略、用户策略和基于资源的策略（空间策略、存储空间 ACL、对象 ACL）进行评估，以决定是否对该请求进行授权。如果任何一个策略现实拒绝（Deny）用户的请求，无论其它的策略是否授权，KS3都会拒绝用户的这次请求。

以下是一些示例说明：

- 如果请求者是IAM子用户，则 KS3 需要确定该IAM子用户所属的父账户是否通过用户策略（User Policy）授予该用户执行操作的必要权限；但是，如果父账户是资源所有者，也可以通过空间策略(Bucket Policy) 授予该IAM子用户执行操作的必要权限。
- IAM子用户除了需要来自父账户的授权外，还需要得到资源所有者向IAM子用户（使用空间策略）或父账户（使用空间策略、存储桶 ACL 或对象 ACL）授予权限；但是如果父账户是资源所有者，在通过用户策略授权后，不再需要空间策略的授权。
- 如果请求者是主账户，则KS3只需要判断是否为资源所有者；如果不是资源所有者，必须得到资源所有者向该主账户（使用空间策略、存储空间 ACL 或对象 ACL）授予权限。
- 如果请求者是匿名用户，则KS3需要判断ACL是否公开，或者资源所有者通过空间策略进行了匿名授权。

流程图如下：

- 用户上下文：如果请求者是IAM子用户，则该子用户必须拥有来自其所属的父账户的授权，同时KS3还要判断其父账户是否也拥有对应的权利。

注意：如果父账户同时也是资源所有者，那么通过空间权限(bucket policy或ACL) 授予该IAM子用户执行操作的必要权限也是可以的。

- 资源上下文：请求者必须拥有来自资源所有者的权限才能执行特定的操作。资源所有者可以通过用户策略、空间策略、ACL向主账号和IAM子用户授予权限。

注意：如果IAM子用户的父账户是资源所有者，已经通过用户策略获得了权限，在空间策略没有显示拒绝但是没有显示授权的情况下，也是可以通过资源上下文的验证。

账号设置

1. KS3主账号的访问密钥管理于2017年4月27日迁移到身份与访问控制（IAM），KS3主账号可以继续使用之前创建的KS3访问密钥，但仅支持禁用、删除操作，不支持禁用后的重新启用操作和创建操作；

查看方式：

若KS3主账号有旧的KS3访问密钥，可以登录[KS3控制台](#)之后，点击右上角账户名，点击下拉菜单中的【accesskeys】查看。



2. 推荐客户使用【AK密钥】，KS3的访问密钥【对象存储密钥】独立于金山云其他产品，仅限于访问KS3，不能用于访问金山云其他产品。



3. 在[访问控制 \(IAM\)](#) 中创建的主账户访问密钥可以访问包括KS3在内的所有金山云服务。建议客户前往身份与访问控制管（IAM），管理主账号及IAM用户。请参考[访问控制 \(IAM\) 文档](#)。

敏感操作保护

用户可以开启敏感操作保护功能，在进行特定操作时验证身份，保护数据；启用该功能后，用户在进行生命周期规则的创建、编辑、删除、启动、停用以及空间策略的创建、编辑、删除等敏感操作时，需要按照流程通过二次验证。

详情说明请参见[用户账户操作-敏感操作保护](#)。

图像处理API

图像样式管理

云监控概览

使用云监控

用户可以使用云监控 API 对云监控服务进行相关操作，如读取监控数据，也可以调用相关的SDK进行读取监控数据。

术语表

名称	中文	说明
----	----	----

Namespace	命名空间	一个命名空间表示一类资源，在金山云中命名空间与云产品一一对应。KS3对象存储对应的命名空间为：“KS3”
Metric	指标	指标是监控的变量，监控数据是该变量随时间变化的数值。详见 KS3监控项
Instance	实例	云资源实例，最小监控单元。KS3云监控的实例为Bucket，使用bucket name来表示实例ID
endpoint	云监控接入域名	云监控的服务接入地址：monitor. {region}. api. ksyun. com， {region}根据KS3 Bucket所在的Region不同而不同，详见 KS3 Bucket Region和云监控endpoint对应表

KS3监控项

KS3云监控服务为客户提供计量方面的监控数据指标，借助云监控服务，客户可以实时的洞察KS3上资源使用情况。

监控指标	描述	单位
ks3.bucket.capacity.total.sd	标准存储量总量	byte
ks3.bucket.capacity.add.sd	标准存储量增量	byte
ks3.bucket.capacity.del.sd	标准存储量删除量	byte
ks3.bucket.capacity.total.ia	低频存储量总量	byte
ks3.bucket.capacity.add.ia	低频存储量增量	byte
ks3.bucket.capacity.del.ia	低频存储量删除量	byte
ks3.bucket.capacity.total.ar	归档存储总量	byte
ks3.bucket.capacity.add.ar	归档存储增量	byte
ks3.bucket.capacity.del.ar	归档存储删除量	byte
ks3.bucket.flow.down.sd	标准存储流量（外网下载流量+CDN回源流量）	byte
ks3.bucket.flow.onet.down.sd	标准存储外网流量（不包含CDN下载量的公网下行流量）	byte
ks3.bucket.flow.cdn.down.sd	标准存储CDN流量（下载）	byte
ks3.bucket.flow.down.ia	低频存储流量（外网下载流量+CDN回源流量）	byte
ks3.bucket.flow.onet.down.ia	低频存储外网流量（不包含CDN下载量的公网下行流量）	byte
ks3.bucket.flow.cdn.down.ia	低频存储CDN流量（下载）	byte
ks3.bucket.flow.down.ar	归档存储流量（外网下载流量+CDN回源流量）	byte
ks3.bucket.flow.onet.down.ar	归档存储外网流量（不包含CDN下载量的公网下行流量）	byte
ks3.bucket.flow.cdn.down.ar	归档存储CDN流量（下载）	byte
ks3.bucket.bandwidth.down	下行带宽（不区分外网带宽和CDN回源带宽）	bps
ks3.bucket.getcount.sd	标准get次数	次
ks3.bucket.putcount.sd	标准put次数	次
ks3.bucket.getcount.ia	低频get次数	次
ks3.bucket.putcount.ia	低频put次数	次
ks3.bucket.getcount.ar	归档get次数	次
ks3.bucket.putcount.ar	归档put次数	次
ks3.bucket.flow.up.ia	低频存储取回量	byte
ks3.bucket.flow.up.ar	归档存储取回量	byte

KS3 Bucket Region和云监控endpoint对应表

KS3会将不同Region下的Bucket的监控数据，发到对应的云监控服务地址。用户在查询某个Bucket的监控数据时，需要查询Bucket所在的Region，然后去云监控对应的服务地址查询相关的监控项。

Region	云监控endpoint
北京 (BEIJING)、金融专区 (北京)、政务专区 (北京)	monitor.cn-beijing-6.api.ksyun.com
上海 (SHANGHAI)、金融专区 (上海)	monitor.cn-shanghai-2.api.ksyun.com
广州 (GUANGZHOU)	monitor.cn-guangzhou-1.api.ksyun.com
香港 (HONGKONG)	monitor.cn-hongkong-2.api.ksyun.com
俄罗斯 (RUSSIA)	monitor.eu-east-1.api.ksyun.com
新加坡 (SINGAPORE)	monitor.ap-singapore-1.api.ksyun.com

粒度和延迟说明

KS3云监控指标的按照天粒度采集聚合的，延迟为10个小时左右。建议用户在次日的13:00之后查询昨日数据。

使用Java SDK访问云监控

目录

- [开发前准备](#)
 - [安装KS3 Java SDK](#)
 - [安装云监控 Java SDK](#)
 - [运行环境](#)
- [Demo演示](#)

开发前准备

安装KS3 Java SDK

因为用户在查询某个Bucket的监控数据时，需要查询Bucket所在的Region，去云监控对应的服务地址查询相关的监控项，所以需要安装KS3 SDK。详见[KS3 SDK For Java使用指南](#)。

安装云监控Java SDK

详见[云监控 SDK说明](#)。

运行环境

适用于Java 5 以上开发环境

Demo演示

```
import org.apache.log4j.Logger;

import com.ksc.auth.AWSCredentials;
import com.ksc.auth.BasicAWSCredentials;
import com.ksc.monitor.KSCMonitorClient;
import com.ksc.monitor.model.GetMetricStatisticsRequest;
import com.ksc.monitor.model.GetMetricStatisticsResponse;

/**
 * Created by cutedandan on 2017/7/3.
 */
public class ks3Size {
    private static final Logger log = Logger.getLogger(ks3Size.class);

    public static void main(String[] args) {
        String ak = "<your ak>";
        String sk = "<your sk>";

        AWSCredentials credentials = new BasicAWSCredentials(ak, sk);
        KSCMonitorClient client = new KSCMonitorClient(credentials);

        # 不同Region的bucket需要设置对应的参数去查询
        # 北京region monitor.cn-beijing-6.api.ksyun.com
        # 上海region monitor.cn-shanghai-2.api.ksyun.com
        # 香港region monitor.cn-hongkong-2.api.ksyun.com
        # 俄罗斯region monitor.eu-east-1.api.ksyun.com
        client.setEndpoint("monitor.cn-beijing-6.api.ksyun.com");

        String bucket="<YourBucketName>";
        String startTime="2017-09-04T016:00:00Z";
        String endTime="2017-09-05T016:00:00Z";
        //获取标准存储的总量
        GetMetricStatisticsRequest request=new GetMetricStatisticsRequest();
        request.setNamespace("KS3"); //固定设置为KS3
        request.setInstanceId(bucket); // 设置对应要查询的bucket name
        request.setMetricName("ks3.bucket.capacity.total.sd"); //要查询的参数
        request.setStartTime(startTime); // 设置起始和截止时间为查询7月2日的数据
        request.setEndTime(endTime);
        request.setPeriod(24*60*60);
        request.setAggregate("Max");
        request.setVersion("2010-05-25");

        GetMetricStatisticsResponse result = client.getMetricStatistics(request);
        String lable = result.getGetMetricStatisticsResult().getLabel();

        System.out.println("result: " + result);
        System.out.println("lable: " + lable);
        System.out.println("Datapoints: " + result.getGetMetricStatisticsResult().getDatapoints().get(0).getMax());

        // 获取一天中标准存储的增量
        request = new GetMetricStatisticsRequest();
        request.setNamespace("KS3");
        request.setInstanceId(bucket);
        request.setMetricName("ks3.bucket.capacity.add.sd");
        request.setStartTime(startTime);
        request.setEndTime(endTime);
        request.setPeriod(24*60*60);
        request.setAggregate("Max");
        request.setVersion("2010-05-25");

        result= client.getMetricStatistics(request);
        lable = result.getGetMetricStatisticsResult().getLabel();

        System.out.println("result: " + result);
        System.out.println("lable: " + lable);
        System.out.println("Datapoints: " + result.getGetMetricStatisticsResult().getDatapoints().get(0).getMax());

        //获取一天中标准存储的删除量
        request = new GetMetricStatisticsRequest();
```

```

request.setNamespace("KS3");
request.setInstanceId(bucket);
request.setMetricName("ks3.bucket.capacity.del.sd");
request.setStartTime(startTime);
request.setEndTime(endTime);
request.setPeriod(24*60*60);
request.setAggregate("Max");
request.setVersion("2010-05-25");

result= client.getMetricStatistics(request);
lable = result.getGetMetricStatisticsResult().getLabel();

System.out.println("result: " + result);
System.out.println("lable: " + lable);
System.out.println("Datapoints: " + result.getGetMetricStatisticsResult().getDatapoints().get(0).getMax());

System.out.println(result.getGetMetricStatisticsResult().getDatapoints().get(0).getTimestamp());
}
}

```

使用 Go SDK访问云监控

目录

- [开发前准备](#)
 - [安装KS3 GO SDK](#)
 - [安装云监控 GO SDK](#)
 - [运行环境](#)
 - [初始化配置](#)
- [Demo演示](#)

开发前准备

安装KS3 GO SDK

因为用户在查询某个Bucket的监控数据时，需要查询Bucket所在的Region，去云监控对应的服务地址查询相关的监控项，所以需要安装KS3 SDK。详见[KS3 SDK for GO使用指南](#)。

安装云监控 GO SDK

详见[云监控 SDK说明](#)。

运行环境

基于aws-sdk-go改造，适用于golang 1.8 开发环境。

初始化配置

1、用户首先需要在金山云控制台申请安全凭证，安全凭证包括access_key_id和secret_access_key。access_key_id 是用于标识API调用者的身份，secret_access_key是用于加密签名字符串和服务器端验证签名字符串的密钥。secret_access_key 必须严格保管，避免泄露。获取安全凭证方法：[点击查看](#)。

2、将获取到的access_key_id和secret_access_key（AK和SK）在程序进行配置。

Demo演示

此Demo演示了：如果获取某一账号下所有bukcet的标准存储量。

```

package main

import (
    "encoding/json"
    "fmt"
    "github.com/ks3sdklib/aws-sdk-go/aws"
    "github.com/ks3sdklib/aws-sdk-go/aws/credentials"
    "github.com/ks3sdklib/aws-sdk-go/service/s3"
    "github.com/ksc/ksc-sdk-go/ksc"
    "github.com/ksc/ksc-sdk-go/ksc/utils"
    "github.com/ksc/ksc-sdk-go/service/monitor"
)

func query_monitor(bucket string, region string, ak string, sk string) {
    //debug模式的话 打开这个开关
    svc := monitor.SdkNew(ksc.NewClient(ak, sk /*true*/), &ksc.Config{Region: &region}, &utils.UrlInfo{
        UseSSL: true,
    })
    //*****获取监控项(ListMetrics)*****
    m_metrics := make(map[string]interface{})
    m_metrics["Namespace"] = "KS3"

```

```
m_metrics["InstanceID"] = bucket
m_metrics["MetricName"] = "ks3.bucket.capacity.total.sd"
m_metrics["Period"] = "86400"
//如果要查询带宽, MetricName设置为"ks3.bucket.bandwidth.down", Period设置为300
//m_metrics["MetricName"] = "ks3.bucket.bandwidth.down"
//m_metrics["Period"] = "300"
m_metrics["StartTime"] = "2019-06-22T00:00:00Z"
m_metrics["EndTime"] = "2019-06-23T00:00:00Z"
m_metrics["Aggregate"] = "Max, min, avg"
resp, err := svc.GetMetricStatistics(&m_metrics)

if err != nil {
    fmt.Println(err.Error())
}
if resp != nil {
    str, _ := json.Marshal(&resp)
    fmt.Printf("%+v\n", string(str))
}
}

func main() {
    ak := "your_ak"
    sk := "your_sk"

    region_map := map[string]string {
        "BEIJING": "cn-beijing-6",
        "SHANGHAI": "cn-shanghai-2",
        "GUANGZHOU": "cn-guangzhou-1",
        "HONGKONG": "cn-hongkong-2",
        "RUSSIA": "eu-east-1",
        "SINGAPORE": "ap-singapore-1",
    }

    var cre = credentials.NewStaticCredentials(ak, sk, "")
    var svcKs3 = s3.New(&aws.Config{
        Region: "BEIJING",
        Credentials: cre,
        Endpoint: "ks3-cn-beijing.ksyun.com",
        DisableSSL: true,
        LogLevel: 1,
        S3ForcePathStyle: true,
        LogHTTPBody: true,
    })

    out, err := svcKs3.ListBuckets(nil)
    if err != nil {
        fmt.Println(err)
        return
    }

    buckets := out.Buckets
    for i:=0; i<len(buckets); i++{
        fmt.Println(*buckets[i].Name)
        fmt.Println(*buckets[i].Region)
        query_monitor(*buckets[i].Name, region_map[*buckets[i].Region], ak, sk)
    }
}
```

使用Python SDK访问云监控

目录

- [开发前准备](#)
 - [安装KS3 Python SDK](#)
 - [安装云监控 Python SDK](#)
 - [运行环境](#)
 - [初始化配置](#)
- [Demo演示](#)

开发前准备

安装KS3 Python SDK

因为用户在查询某个Bucket的监控数据时, 需要查询Bucket所在的Region, 去云监控对应的服务地址查询相关的监控项, 所以需要安装KS3 SDK。详见[KS3 SDK for Python使用指南](#)。

安装云监控Python SDK

详见[云监控 SDK说明](#)。

运行环境

支持Python2.6, 2.7, 3.7 版本。

初始化配置

1. 用户首先需要在金山云控制台申请安全凭证，安全凭证包括access_key_id和secret_access_key。access_key_id 是用于标识API调用者的身份，secret_access_key是用于加密签名字符串和服务器端验证签名字符串的密钥。secret_access_key 必须严格保管，避免泄露。获取安全凭证方法：[点击查看](#)。
2. 通过文件配置及管理密钥，参考examples内示例：

所在位置：'/etc/kscore.cfg' 或 './.kscore.cfg' 或 'C:\kscore.cfg'

注意：使用相对路径时，需与运行目录保持一致。

```
[Credentials]
ks_access_key_id=<your ak>
ks_secret_access_key=<your sk>
```

注意：配置文件中ak和sk不要加引号

3. 用户也可以在程序运行时配置：

```
from kscore.session import get_session
# 密钥
ACCESS_KEY_ID = "<your ak>"
SECRET_ACCESS_KEY = "<your sk>"

s = get_session()
client = s.create_client("monitor", "cn-beijing-6", use_ssl=True, ks_access_key_id=ACCESS_KEY_ID, ks_secret_access_key=SECRET_ACCESS_KEY)
```

Demo演示

```
from kscore.session import get_session
import json

if __name__ == "__main__":
    s = get_session()

    # 不同Region的bucket需要设置对应的参数去查询
    # 北京region cn-beijing-6
    # 上海region cn-shanghai-2
    # 香港region cn-hongkong-2
    # 俄罗斯region eu-east-1

    #GetMetricStatistics
    client = s.create_client("monitor", "cn-beijing-6", use_ssl=True)

    #获取一天的标准存储总量
    m=client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.total.sd", StartTime="2018-07-03T00:00:00Z", EndTime="2018-07-04T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)

    #获取一天的标准存储增量
    m=client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.add.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的标准存储量删除量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.del.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的低频存储总量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.total.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的低频存储增量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.add.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的低频存储量删除量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.del.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的归档存储总量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.total.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的归档存储增量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.add.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的归档存储量删除量
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.capacity.del.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
    # 获取一天的标准存储流量（外网下载流量+CDN回源流量）
    m = client.get_metric_statistics(InstanceID="<YourBucketName>", Namespace="KS3", MetricName="ks3.bucket.flow.down.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
    print json.dumps(m, sort_keys=True, indent=4)
```

```
# 获取一天的标准存储外网流量（不包含CDN下载量的公网下行流量）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.onet.down.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的标准存储CDN流量（下载）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.cdn.down.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的低频存储流量（外网下载流量+CDN回源流量）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.down.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的低频存储外网流量（不包含CDN下载量的公网下行流量）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.onet.down.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的低频存储CDN流量（下载）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.cdn.down.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的归档存储流量（外网下载流量+CDN回源流量）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.down.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的归档存储外网流量（不包含CDN下载量的公网下行流量）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.onet.down.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的归档存储CDN流量（下载）
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.cdn.down.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的标准存储的get请求数
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.getcount.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的标准存储的put请求数
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.putcount.sd", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的低频存储的get请求数
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.getcount.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的低频存储的put请求数
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.putcount.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的归档get次数
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.getcount.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的归档put次数
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.putcount.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的低频存储的数据取回量
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.up.ia", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
# 获取一天的归档存储取回量
m = client.get_metric_statistics(InstanceID=<YourBucketName>, Namespace="KS3", MetricName="ks3.bucket.flow.up.ar", StartTime="2018-03-18T00:00:00Z", EndTime="2018-03-19T00:00:00Z", Period="86400", Aggregate="Max")
print json.dumps(m, sort_keys=True, indent=4)
```