

目录

目录	1
开通KS3服务	3
注册金山云账户	3
开通KS3服务	3
获取AK/SK	3
说明:	3
创建存储空间	3
存储空间限制	3
操作步骤	3
上传文件	4
文件上传	4
操作步骤:	4
其他实现方式	5
分享与下载文件	5
文件权限为私密的情况下, 短暂分享文件操作步骤:	5
文件下载	6
操作方法:	6
其他实现方式	6
删除文件	6
删除单个文件	6
批量删除文件	7
其他实现方式	7
删除风险	8
如何避免误删	8
删除存储空间	8
KS3Util命令行工具快速入门	8
前提条件	8
创建Bucket	8
上传文件	8
下载文件	8
删除文件	8
删除Bucket	8
KS3Finder图形化界面工具快速入门	9
前提条件	9
创建存储空间	9
上传、下载文件/目录	9
删除文件	9
删除存储空间	9
Java SDK快速入门	10
前提条件	10
创建存储空间	10
上传文件	10
下载文件	10
删除文件	11
删除存储空间	11
Go SDK快速入门	11
前提条件	11
创建存储空间	11
上传文件	12
下载文件	12

删除文件	13
删除存储空间	13
Python SDK快速入门	13
前提条件	13
创建存储空间	13
上传文件	14
下载文件	14
删除文件	14
删除存储空间	14

开通KS3服务

注册金山云账户

在使用金山云 KS3 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，系统会在您开通 KS3 时提示您注册账号。

步骤1：注册金山云账号 [注册金山云](#)，如果已在金山云注册，可忽略此步骤。

步骤2：完成实名认证 如果已完成，可忽略此步骤。 [完成实名认证](#)，详细认证流程请参见 [实名认证介绍](#)。

开通KS3服务

1. 登录[KS3控制台](#)
2. 点击[提交工单](#)或联系[在线客服](#)开通KS3服务。



您尚未开通KS3服务

如需体验产品，请 [提交工单](#) 或联系 [在线客服](#) 进行开通

 400-1070-808

产品介绍

金山云KS3 (Kingsoft Standard Storage Service) 为您提供基于网络的数据存取服务，使用KS3，您可以通过网络方便地存储和获取图片、音频、视频、文本等各类数据文件。

KS3会将您上传的数据文件以对象 (Object) 的形式组织到存储空间 (Bucket) 里，然后提供地址 (域名) 给您用于分享或下载数据文件。

[快速入门 >](#)
[产品定价 >](#)



获取AK/SK

1. 鼠标悬停在页面右上角的用户名上，选择下拉菜单中的Access Keys选项，进入AK密钥管理页面。
2. 点击[新建密钥](#)创建AK。

注意： Access Key及密钥是您访问金山云API的凭证，具有该账户的全部的权限，请您妥善保管。并养成定时更换密钥的习惯，及时删除以前的密钥。

说明：

- 如果您是子用户，需要您联系主账号开通KS3服务后赋予权限，再使用KS3服务。

创建存储空间

注册开通KS3后，您就可以开始使用KS3管理控制台创建存储空间 (Bucket)。KS3中的每个文件 (Object) 都存储在存储空间 (Bucket) 中。必须先创建一个存储空间，然后才能在KS3中存储数据。

存储空间限制

- 同一用户创建的存储空间总数不能超过100个。
- 存储空间的名在KS3服务内全局唯一。
- 存储空间一旦创建成功，名称和所处地域不能修改。
- 单个存储空间的容量无限制。

操作步骤

1. 进入 [KS3控制台](#)。
2. 点击左侧导航栏中的**存储空间**，点击上**创建空间**按钮。

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。

存储空间的命名规则如下：

- 仅包含小写英文字母 (a-z)，数字，中线，即： abcdefghijklmnopqrstuvwxyz012345678-9
- 必须由字母或数字开头和结尾
- 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

3. 在弹出的创建空间对话框中选择**地区**，输入**空间名称**，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。

4. 不支持更换地域。如需要通过云服务器 KEC 内网访问 KS3，需要选择与您 KEC 相同的地域。更多信息请参阅[访问域名](#)。

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“标准存储”、“低频存储”或“归档存储”。标准存储空间适用于频繁访问的业务场景；低频存储空间适用于较低访问频率的业务场景；归档存储空间适用于存储长期保存、访问频率极低的文件，需要将文件解冻后，才可以访问。

7. 在 **访问控制** 单选框中，选择对应的权限。

- **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，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。

8. 单击**确定**，存储空间创建完成。

上传文件

文件上传

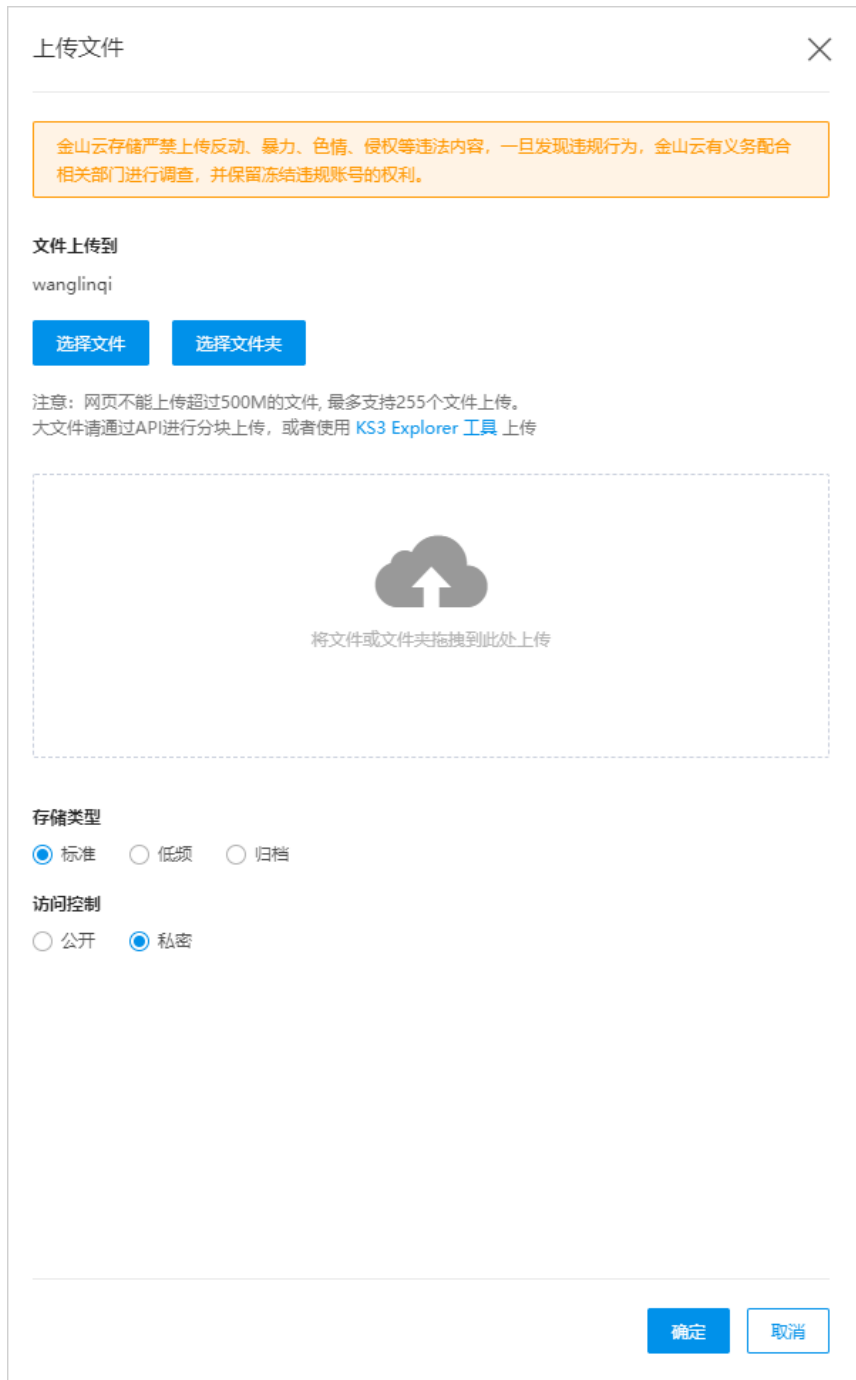
您可以上传任何文件到存储空间中。需要注意的是：控制台不能上传超过1GB的文件，大文件请通过API进行[分块上传](#)，或通过[KS3工具](#)进行上传。当文件数量超过1000个时，建议通过[API](#)或[KS3 Finder](#)工具上传。

操作步骤：

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，点击上方**上传**按钮。



4. 右侧弹出上传对话框，支持多文件上传。选择文件/文件夹，或直接将文件/文件夹拖拽到对话框中。



5. 选择文件的**存储类型**。详情请详见[存储类型介绍](#)。

- 6. 选择文件的访问控制权限。
- 7. 点击**确认**，开始上传文件。

其他实现方式

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	

分享与下载文件

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. 进入 [KS3控制台](#)。
 - 2. 点击左侧导航栏中的**存储空间**，点击您想分享文件所在的空间名称。
 - 3. 点击**内容管理**页签。
 - 4. 在文件列表中，点击想要分享的文件名，或点击文件右侧的**详情**。
-
- 5. 右侧出现文件详情弹窗，在文件详情弹窗中**设置外链地址过期时间**。

文件详情



文件名

KS3产品白皮书V1.1.pdf

存储类型

标准

ACL

私密

设置外链地址过期时间

0

分

60

秒

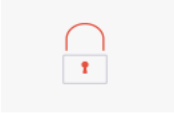
确定

外链地址

复制

- 6. 设置完成后，点击**确定**，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。

文件详情



文件名

KS3产品白皮书V1.1.pdf

存储类型

标准

ACL

私密

设置外链地址过期时间

0

分

60

秒

确定

外链地址

https://

;-cn-beijing.ksyuncs.com/%E6%B5%8B%E8%AF%95/KS3%E

复制

注意：如果是归档文件，需要解冻后，才可以进行文件分享操作。

文件下载

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**，点击您想下载文件所在的空间名称。
3. 点击**内容管理**页签。
4. 在文件列表页中，点击想要下载文件的右侧**下载**，即可完成该文件下载。

其他实现方式

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	

删除文件

如果您不再需要存储所上传的文件，请将其删除以免进一步产生费用。您可以通过控制台删除单个文件或批量删除文件。也可以通过[KS3工具](#)、[SDK](#)、[生命周期管理](#)来批量删除。

删除单个文件

1. 进入 [KS3控制台](#)。
2. 点击左侧导航栏中的**存储空间**，点击您想删除文件所在的空间名称。
3. 点击**内容管理**页签。
4. 在文件列表页中，找到需要删除的文件，点击右侧**更多** > **删除**。

5. 弹出删除对话框，单击**确定**。

?

你确认要删除此文件吗？

确定

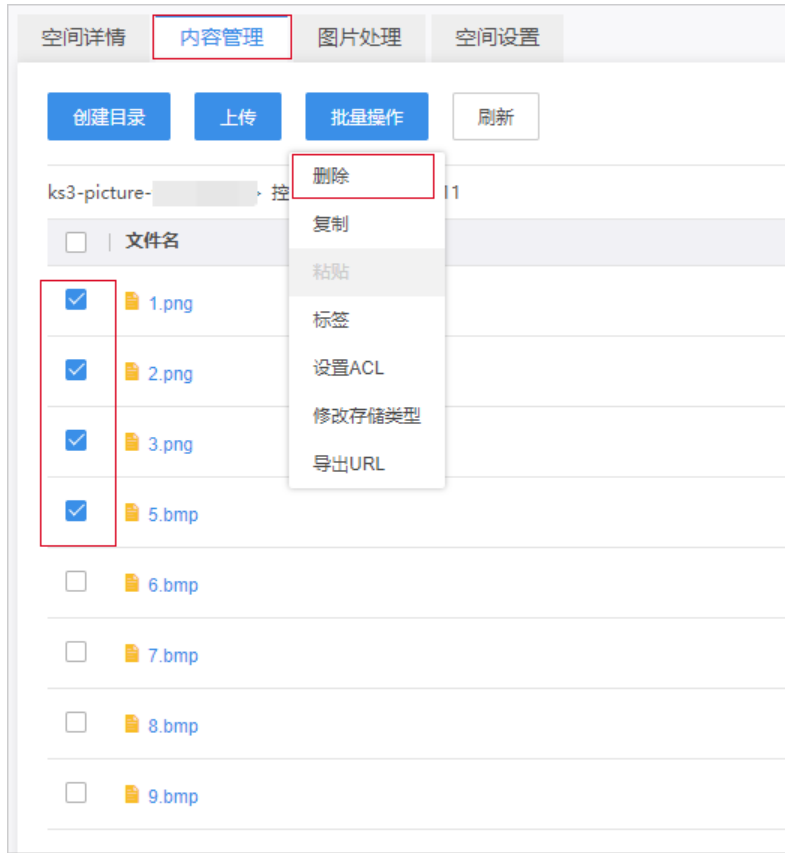
取消

金山云

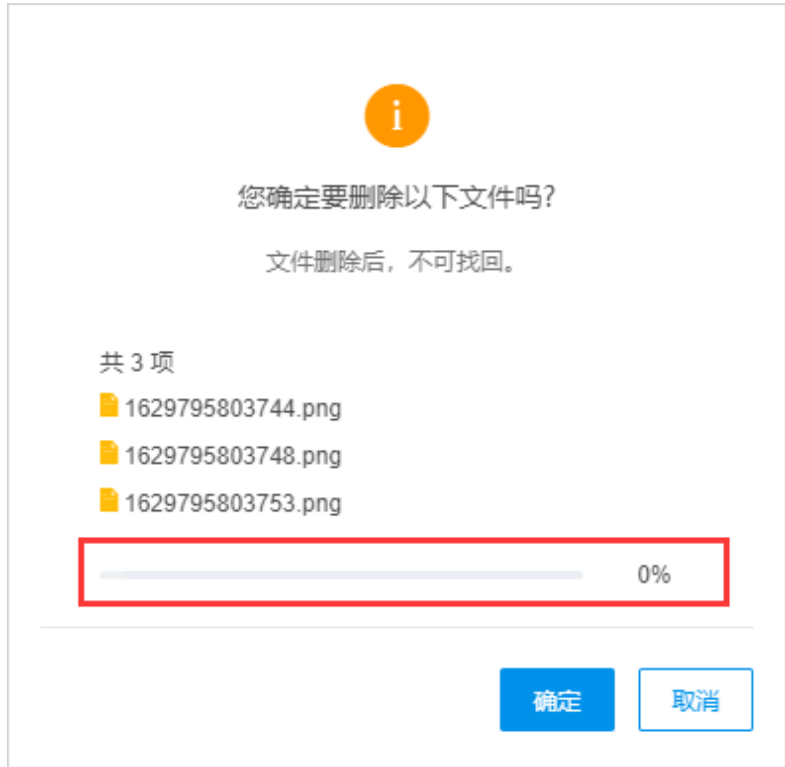
6/14

批量删除文件

1. 进入 [KS3控制台](#)。
2. 点击左侧导航栏中的**存储空间**，点击您想删除文件所在的空间名称。
3. 点击**内容管理**页签。
4. 在文件列表页中，选定多个需要删除的文件，点击**批量操作** > **删除**。



5. 弹出删除对话框，单击**确认**，用户可以看到删除的进度。



注意：删除的文件总数量不可以超过1000个。如果超过1000个，需要使用KS3工具、SDK去删除。

其他实现方式

1. 命令行工具Util

操作步骤参见文档[KS3Util新版命令行工具](#)。

2. 可视化工具Finder

操作步骤参见文档[KS3Finder图形化界面工具](#)。

3. SDK

操作步骤参见文档[SDK概览](#)。

4. 生命周期管理

若客户在生命周期规则中配置了定期删除文件，KS3会根据生命周期的配置定期删除符合条件的文件。

操作步骤参见文档[生命周期管理](#)。

删除风险

以下情况均会导致您的数据被误删，请谨慎操作。

- 在使用上述删除方式时操作失误，导致误删。
- 没有正确地配置Bucket访问权限，导致文件被他人恶意删除。

如何避免误删

为了避免数据被误删，需要配置正确的访问权限，配置时注意遵循以下原则：

- 不使用主账号访问KS3。
- 读写分离，对于只需要读数据的业务，只使用具有读权限的子账号。
- 针对不同的业务，授予“够用且最小的范围”的KS3权限。如对子账号设置用户策略(User Policy)，限制用户调用DeleteObject删除文件权限。
- 妥善保管数据访问的凭据，如账号密码、IAM子账号访问凭据。配置权限详情参见文档[权限概述](#)。

删除存储空间

删除存储空间之前请确保其中存储的文件和配置信息（包括自定义域名、空间策略等）已经全部删除，否则无法删除存储空间。

具体步骤如下：

1. 进入 [KS3控制台](#)。
2. 选择左侧列表中的**存储空间**，找到需要删除的空间。
3. 在页面右侧点击**删除**。
4. 弹出删除空间对话框，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）。

KS3Util命令行工具快速入门

本文旨在引导您通过KS3Util命令行工具快速进行创建存储空间（Bucket）、将本地文件上传至Bucket、删除文件、删除存储空间一系列基本操作。

前提条件

已安装KS3Util。[点击下载](#)

创建Bucket

命令格式

```
./ks3util mb ks3://bucketname
```

使用示例

创建名为samplebucket的存储空间

```
./ks3util mb ks3://samplebucket
```

上传文件

命令格式

```
./ks3util cp local_file ks3://bucketname
```

使用示例

上传单个文件sample.txt至名为samplebucket的存储空间

```
./ks3util cp sample.txt ks3://samplebucket
```

下载文件

命令格式

```
./ks3util cp cloud_url file_url
```

使用示例

将单个文件sample.txt从存储空间samplebucket下载到本地local-dir

```
./ks3util cp ks3://samplebucket/src-prefix/sample.txt local-dir/
```

删除文件

命令格式

```
./ks3util rm ks3://bucketname[/prefix]
```

使用示例

删除存储空间samplebucket内的单个文件sample.txt

```
./ks3util rm ks3://samplebucket/sample.txt
```

删除Bucket

命令格式

```
./ks3util rm ks3://bucketname -b
```

使用示例

删除存储空间samplebucket

```
./ks3util rm ks3://samplebucket -b
```


关于使用KS3Util 命令行工具的更多细节操作，请参见[KS3Util 命令行工具](#)

KS3Finder图形化界面工具快速入门

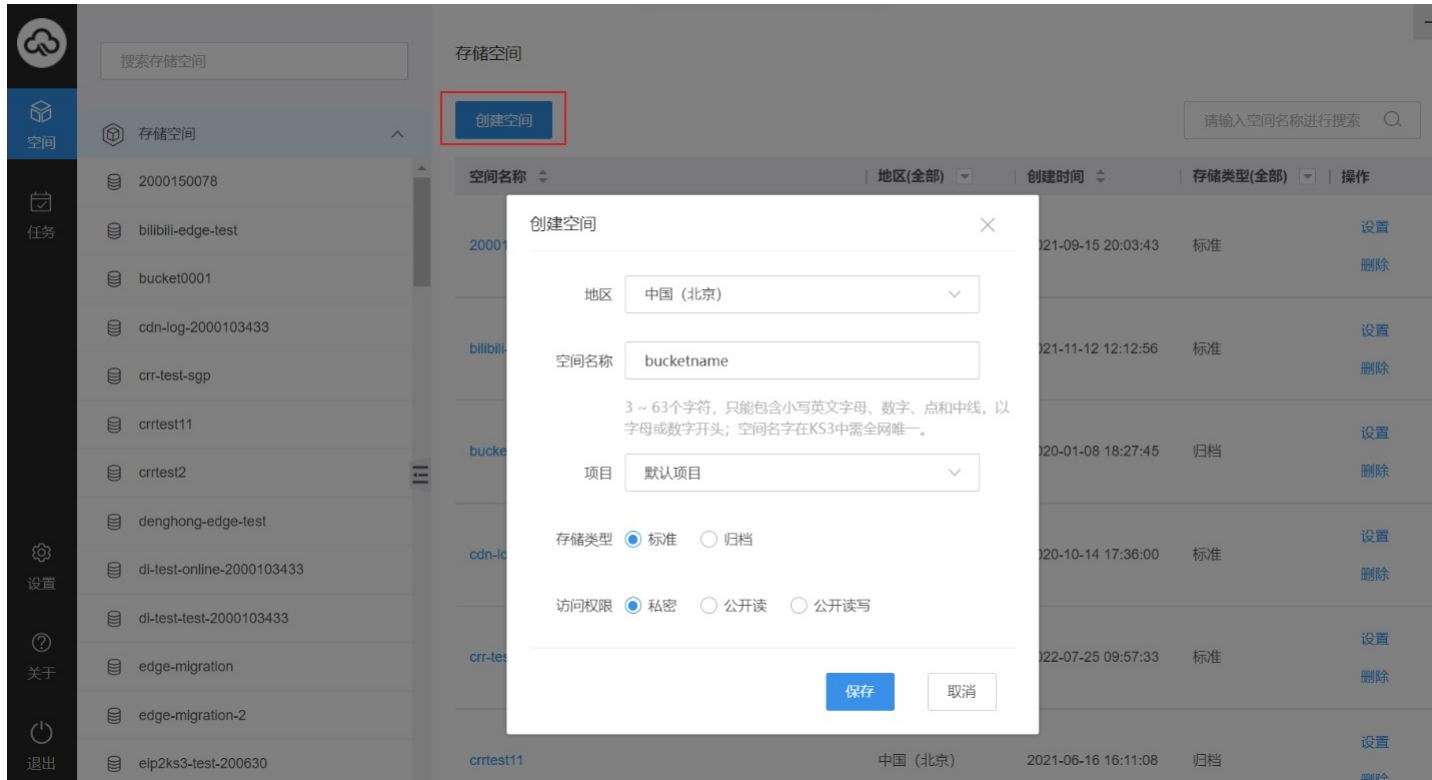
KS3 Finder是金山云对象存储KS3推出的桌面端可视化界面工具，使用KS3Finder，您可以快速完成存储空间（Bucket）和文件（Object）的相关操作。

前提条件

已安装并登录KS3Finder。[点击下载](#)

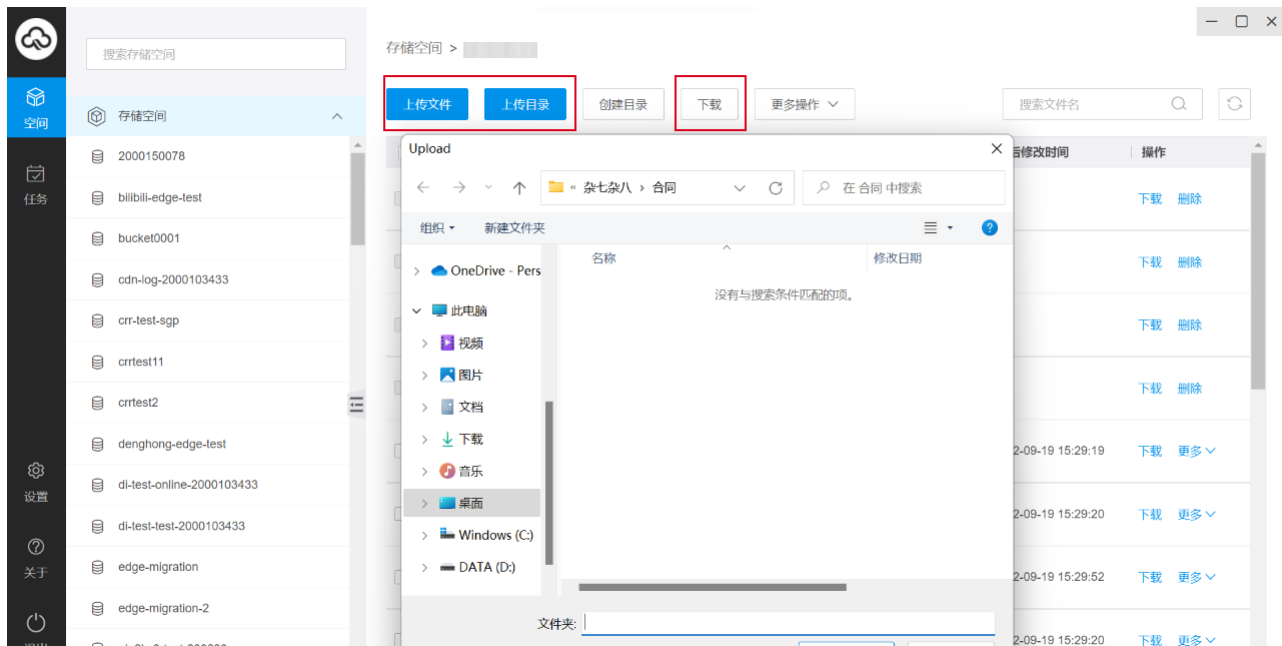
创建存储空间

在桌面端左侧选中【空间】进入“存储空间”页面，点击【创建空间】并正确填写“地区”、“空间名称”、“项目”、“存储类型”和“访问权限”，单击【保存】即可创建完成。



上传、下载文件/目录

KS3 Finder支持单个上传、批量上传、增量上传文件或目录至存储空间，支持单个下载、批量下载、增量下载文件或目录至本地。

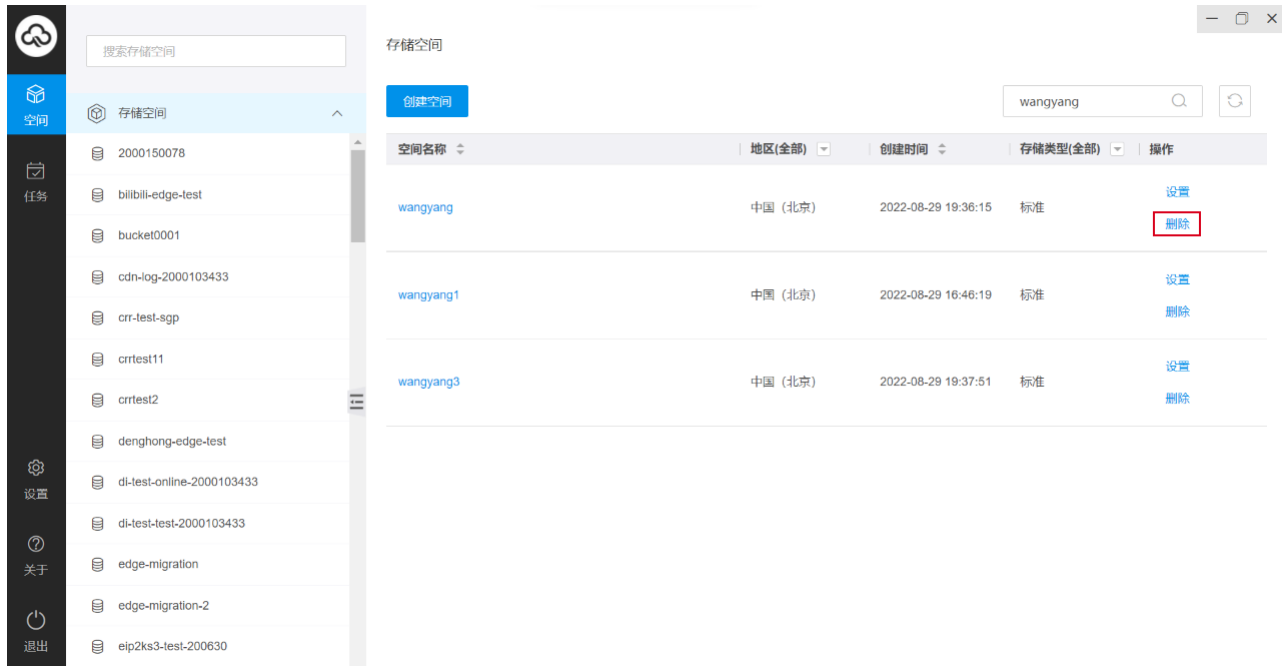


删除文件

选中需要删除的文件，在右侧“操作栏”中点击【更多】 【删除】进行删除，或是批量选中要删除的文件，在【更多操作】中找到【删除】进行一键删除。 注意：对象一旦被删除将不可恢复，需要谨慎操作。

删除存储空间

点击右侧“操作栏”中【删除】完成删除空间。



注意：删除存储空间前需要将桶内对象全部删除，对象一旦被删除将不可恢复，需要谨慎操作。

关于使用KS3Finder图形化界面工具的更多细节操作，请参见[KS3Finder图形化界面工具](#)。

Java SDK快速入门

本文介绍如何快速使用JAVA SDK完成常见操作，如创建存储空间（Bucket）、上传文件（Object）、下载文件等。

前提条件

已安装JAVA SDK。详情请参见[安装](#)。

创建存储空间

Bucket是存放Object的容器，所有的Object都必须存放在特定的Bucket中。 以下代码用于创建一个Bucket

```
public void createBucketSimple() {
    // yourEndpoint填写Bucket所在地域对应的Endpoint。以中国（北京）为例，Endpoint填写为ks3-cn-beijing.ksyuncs.com。如果使用自定义域名，设置endpoint为自定义域名，同时设置domainMode为true
    String endpoint = "yourEndpoint";
    // 金山云账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
    String accessKeyId = "yourAccessKeyId";
    String accessKeySecret = "yourAccessKeySecret";
    // 创建Ks3ClientConfig 实例。
    Ks3ClientConfig config = new Ks3ClientConfig();
    // 设置域名
    config.setEndpoint(endpoint);
    // 创建Ks3Client实例
    Ks3 client = new Ks3Client(accessKeyId, accessKeySecret, config);
    // 创建桶
    client.createBucket("<您的bucket名称>");
}
```

上传文件

以下代码用于通过文件上传的方式将文件上传到KS3。

```
// yourEndpoint填写Bucket所在地域对应的Endpoint。以中国（北京）为例，Endpoint填写为ks3-cn-beijing.ksyuncs.com。如果使用自定义域名，设置endpoint为自定义域名，同时设置domainMode为true
String endpoint = "yourEndpoint";
// 金山云账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
String accessKeyId = "yourAccessKeyId";
String accessKeySecret = "yourAccessKeySecret";
// 创建Ks3ClientConfig 实例。
Ks3ClientConfig config = new Ks3ClientConfig();
// 设置域名
config.setEndpoint(endpoint);
/**
 * false: （推荐）使用三级域名: {bucketName}/{endpoint}/{objectKey}的形式访问
 * true: 使用二级域名: {endpoint}/{bucketName}/{objectKey}的形式访问
 * 如果domainMode设置为true，则pathStyleAccess可忽略设置
 */
config.setPathStyleAccess(false);

// 创建Ks3Client实例
Ks3 client = new Ks3Client(accessKeyId, accessKeySecret, config);
// 填写本地文件的完整路径。如果未指定本地路径，则默认从示例程序所属项目对应本地路径中上传文件。
File file = new File("D:\\localpath\\examplefile.txt");
// 依次填写Bucket名称（例如examplebucket）和Object完整路径（例如exampledir/exampleobject.txt）。Object完整路径中不能包含Bucket名称。
client.putObject("examplebucket", "exampledir/exampleobject.txt", file);
```

下载文件

以下代码用于流式下载examplebucket中的exampleobject.txt文件。

```
public GetObjectResult getObject() {
    // yourEndpoint填写Bucket所在地域对应的Endpoint。以中国（北京）为例，Endpoint填写为ks3-cn-beijing.ksyuncs.com。如果使用自定义域名，设置endpoint为自定义域名，同时设置domainMode为true
    String endpoint = "yourEndpoint";
    // 金山云账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
    String accessKeyId = "yourAccessKeyId";
    String accessKeySecret = "yourAccessKeySecret";
    // 创建Ks3ClientConfig 实例。
    Ks3ClientConfig config = new Ks3ClientConfig();
    // 设置域名
    config.setEndpoint(endpoint);
    /**
     * false: （推荐）使用三级域名: {bucketName}/{endpoint}/{objectKey}的形式访问
     * true: 使用二级域名: {endpoint}/{bucketName}/{objectKey}的形式访问
     * 如果domainMode设置为true，则pathStyleAccess可忽略设置
     */
    config.setPathStyleAccess(false);
```

```
// 创建Ks3Client实例
Ks3 client = new Ks3Client(accessKeyId, accessKeySecret, config);
// 填写Bucket名称。
String bucketName = "examplebucket";
// 填写Object的完整路径。Object完整路径中不能包含Bucket名称。
String objectName = "exampleobject.txt";
// 创建GetObject请求，指定
GetObjectRequest request = new GetObjectRequest(bucketName, objectName);
// 可以重写返回的header
// ResponseHeaderOverrides overrides = new ResponseHeaderOverrides();
// overrides.setContentType("text/html");
// .....
// request.setOverrides(overrides);
// 只接受数据的0-10字节。通过控制该项可以实现分块下载
// request.setRange(0,10);

// 如果文件是用户提供key的方式进行服务端加密的
// SecretKey sourceKey= ???当初加密时用的key
// request.setSseCustomerKey(new SSECustomerKey(sourceKey));
// ks3Object包含文件所在的存储空间名称、文件名称、文件元信息以及一个输入流。
Ks3Object ks3Object = client.getObject(request).getObject();
// 读取文件内容。
System.out.println("Object content:");
BufferedReader reader = new BufferedReader(new InputStreamReader(ks3Object.getObjectContent()));
while (true) {
    String line = reader.readLine();
    if (line == null) break;
    System.out.println("\n" + line);
}
// 数据读取完成后，获取的流必须关闭，否则会造成连接泄漏，导致请求无连接可用，程序无法正常工作。
reader.close();
}
```

删除文件

以下代码用于删除文件。

```
/**
 * 将<bucket名称>这个存储空间下的<object key>删除
 */

public void deleteObject() {
    // yourEndpoint填写Bucket所在地域对应的Endpoint。以中国（北京）为例，Endpoint填写为ks3-cn-beijing.ksyuncs.com。如果使用自定义域名，设置endpoint为自定义域名，同时设置domainMode为true
    String endpoint = "yourEndpoint";
    // 金山云账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
    String accessKeyId = "yourAccessKeyId";
    String accessKeySecret = "yourAccessKeySecret";
    // 创建Ks3ClientConfig 实例。
    Ks3ClientConfig config = new Ks3ClientConfig();
    // 设置域名
    config.setEndpoint(endpoint);
    // 创建Ks3Client实例
    Ks3 client = new Ks3Client(accessKeyId, accessKeySecret, config);
    // 删除指定桶内的指定文件
    client.deleteObject("<bucket名称>","<object key>");
}
```

删除存储空间

以下代码用于删除名为test-bucket的存储空间。

```
public void deleteBucket() {
    // yourEndpoint填写Bucket所在地域对应的Endpoint。以中国（北京）为例，Endpoint填写为ks3-cn-beijing.ksyuncs.com。如果使用自定义域名，设置endpoint为自定义域名，同时设置domainMode为true
    String endpoint = "yourEndpoint";
    // 金山云账号AccessKey拥有所有API的访问权限，风险很高。强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
    String accessKeyId = "yourAccessKeyId";
    String accessKeySecret = "yourAccessKeySecret";
    // 创建Ks3ClientConfig 实例。
    Ks3ClientConfig config = new Ks3ClientConfig();
    // 设置域名
    config.setEndpoint(endpoint);
    // 创建Ks3Client实例
    Ks3 client = new Ks3Client(accessKeyId, accessKeySecret, config);
    // 删除桶名为test-bucket的桶
    client.deleteBucket("test-bucket");
}
```

Go SDK快速入门

本文介绍如何快速使用GO SDK完成常见操作，如创建存储空间（Bucket）、上传文件（Object）、下载文件等。

前提条件

已安装GO SDK。详情请参见[安装](#)。

创建存储空间

Bucket是存放Object的容器，所有的Object都必须存放在特定的Bucket中。

以下代码用于创建一个Bucket

```
package bucket_sample

import (
    "fmt"
    "github.com/ks3sdklib/aws-sdk-go/aws"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsutil"
    "github.com/ks3sdklib/aws-sdk-go/aws/credentials"
    "github.com/ks3sdklib/aws-sdk-go/service/s3"
)

var bucket = string("yourbucket")
var key = string("yourkey")
var key_encode = string("yourkey")
var key_copy = string("yourkey")
var content = string("content")

// 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
// 强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
// 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。
var cre = credentials.NewStaticCredentials("ak", "sk", "") //online
var svc = s3.New(&aws.Config{
    Region:    "BEIJING",
    Credentials: cre,
    Endpoint:  "ks3-cn-beijing.ksyuncs.com",
    DisableSSL: true,
    LogLevel:  1,
    S3ForcePathStyle: false,
    LogHTTPBody: true,
})

//创建bucket并关联项目
func TestCreateBucket(svc *s3.S3) {
    resp, _ := svc.CreateBucket(&s3.CreateBucketInput{
        ACL:      aws.String("public-read"),
        Bucket:   aws.String(bucket),
        ProjectId: aws.String("1232"), //项目ID
    })
}
```

```
    })
    fmt.Println("结果: \n", awsutil.StringValue(resp))
}

上传文件

以下代码用于上传文件到K3。

package ks3test

import (
    "bufio"
    "bytes"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsserr"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsutil"
    "net/url"
    "strconv"
    "strings"
    "testing"
    "time"
    // "io"
    "fmt"
    "github.com/ks3sdklib/aws-sdk-go/aws"
    "github.com/ks3sdklib/aws-sdk-go/aws/credentials"
    "github.com/ks3sdklib/aws-sdk-go/service/s3"
    "net/http"
    "os"
)

var bucket = string("yourbucket")
var key = string("yourkey")
var key_encode = string("yourkey")
var key_copy = string("yourkey")
var content = string("content")
var prefix = "test/" //目录名称

// 金山云主账号 AccessKey 拥有所有API的访问权限, 风险很高。
// 强烈建议您创建并使用子账号账号进行 API 访问或日常运维, 请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
// 通过指定 host(Endpoint), 您可以在指定的地域创建新的存储空间。
var cre = credentials.NewStaticCredentials("ak", "sk", "") //online
var svc = s3.New(&aws.Config{
    //Region 可参考 [https://docs.ksyun.com/documents/6761] (https://docs.ksyun.com/documents/6761)
    Region: "BEIJING",
    Credentials: cre,
    //Endpoint 可参考 [https://docs.ksyun.com/documents/6761] (https://docs.ksyun.com/documents/6761)
    Endpoint: "ks3-cn-beijing.ksyuncs.com",
    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
    LogHTTPBody: true, //是否把HTTP请求body打入日志
    S3ForcePathStyle: true,
    Logger: nil, //打日志的位置
})

//上传文件
func putObj(svc *s3.S3) {

    //获取本地文件FD
    fd, _ := os.Open("D:\\suiyi.jpg")

    //指定目标Object对象标签, 可同时设置多个标签, 如: TagA=A&TagB=B。
    //说明 Key和Value需要先进行URL编码, 如果某项没有 "=", 则看作Value为空字符串。详情请见对象标签 ([https://docs.ksyun.com/documents/39576] (https://docs.ksyun.com/documents/39576)) 。
    v := url.Values{}
    v.Add("name", "yz")
    v.Add("age", "11")
    XAmzTagging := v.Encode()

    input := s3.PutObjectInput{
        Bucket: aws.String(bucket),
        Key: aws.String(key),
        ACL: aws.String("public-read"),
        Body: fd,
        XAmzTagging: aws.String(XAmzTagging),
    }
    resp, _ := svc.PutObject(&input)
    fmt.Println("结果: \n", awsutil.StringValue(resp))
}

下载文件

以下代码用于下载文件。

package ks3test

import (
    "bufio"
    "bytes"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsserr"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsutil"
    "net/url"
    "strconv"
    "strings"
    "testing"
    "time"
    // "io"
    "fmt"
    "github.com/ks3sdklib/aws-sdk-go/aws"
    "github.com/ks3sdklib/aws-sdk-go/aws/credentials"
    "github.com/ks3sdklib/aws-sdk-go/service/s3"
    "net/http"
    "os"
)

var bucket = string("yourbucket")
var key = string("yourkey")
var key_encode = string("yourkey")
var key_copy = string("yourkey")
var content = string("content")
var prefix = "test/" //目录名称

// 金山云主账号 AccessKey 拥有所有API的访问权限, 风险很高。
// 强烈建议您创建并使用子账号账号进行 API 访问或日常运维, 请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
// 通过指定 host(Endpoint), 您可以在指定的地域创建新的存储空间。
var cre = credentials.NewStaticCredentials("ak", "sk", "") //online
var svc = s3.New(&aws.Config{
    //Region 可参考 [https://docs.ksyun.com/documents/6761] (https://docs.ksyun.com/documents/6761)
    Region: "BEIJING",
    Credentials: cre,
    //Endpoint 可参考 [https://docs.ksyun.com/documents/6761] (https://docs.ksyun.com/documents/6761)
    Endpoint: "ks3-cn-beijing.ksyuncs.com",
    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
    LogHTTPBody: true, //是否把HTTP请求body打入日志
    S3ForcePathStyle: false,
    Logger: nil, //打日志的位置
})

//下载文件
func TestGetObject(svc *s3.S3) {

    resp, _ := svc.GetObject(&s3.GetObjectInput{
        Bucket: aws.String(bucket),
        Key: aws.String(key),
    })
}
```

```
        ResponseContentType: aws.String("application/pdf"), //响应类型
        Range:                aws.String("bytes=0-1"),      //rang分块下载
    })
    //读取数据
    br := bufio.NewReader(resp.Body)
    //转成字符串
    _, _ = br.ReadString("\n")
    fmt.Println("结果: \n", awsutil.StringValue(resp))
}
```

删除文件

以下代码用于删除文件。

```
package ks3test

import (
    "bufio"
    "bytes"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsserr"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsutil"
    "net/url"
    "strconv"
    "strings"
    "testing"
    "time"
    // "io"
    "fmt"
    "github.com/ks3sdklib/aws-sdk-go/aws"
    "github.com/ks3sdklib/aws-sdk-go/aws/credentials"
    "github.com/ks3sdklib/aws-sdk-go/service/s3"
    "net/http"
    "os"
)

var bucket = string("yourbucket")
var key = string("yourkey")
var key_encode = string("yourkey")
var key_copy = string("yourkey")
var content = string("content")
var prefix = "test/" //目录名称

// 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
// 强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
// 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。
var cre = credentials.NewStaticCredentials("ak", "sk", "") //online
var svc = s3.New(&aws.Config{
    //Region 可参考 [https://docs.ksyun.com/documents/6761] (https://docs.ksyun.com/documents/6761)
    Region: "BEIJING",
    Credentials: cre,
    //Endpoint 可参考 [https://docs.ksyun.com/documents/6761] (https://docs.ksyun.com/documents/6761)
    Endpoint: "ks3-cn-beijing.ksyuncs.com",
    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
    LogHTTPBody: true, //是否把HTTP请求body打入日志
    S3ForcePathStyle: false,
    Logger: nil, //打日志的位置
})

//删除对象
func TestDelObject(svc *s3.S3) {

    resp, _ := svc.DeleteObject(&s3.DeleteObjectInput{
        Bucket: aws.String(bucket),
        Key:    aws.String(key),
    })

    fmt.Println("结果: \n", awsutil.StringValue(resp))
}
```

删除存储空间

以下代码用于删除存储空间。

```
package bucket_sample

import (
    "fmt"
    "github.com/ks3sdklib/aws-sdk-go/aws"
    "github.com/ks3sdklib/aws-sdk-go/aws/awsutil"
    "github.com/ks3sdklib/aws-sdk-go/aws/credentials"
    "github.com/ks3sdklib/aws-sdk-go/service/s3"
)

var bucket = string("yourbucket")
var key = string("yourkey")
var key_encode = string("yourkey")
var key_copy = string("yourkey")
var content = string("content")

// 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
// 强烈建议您创建并使用子账号账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
// 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。
var cre = credentials.NewStaticCredentials("ak", "sk", "") //online
var svc = s3.New(&aws.Config{
    Region: "BEIJING",
    Credentials: cre,
    //Endpoint:"ks3-sgp.ksyun.com",
    Endpoint: "ks3-cn-beijing.ksyuncs.com",
    DisableSSL: true,
    LogLevel: 1,
    LogHTTPBody: true,
})

//删除bucket
func TestDeleteBucket(svc *s3.S3) {
    resp, _ := svc.DeleteBucket(&s3.DeleteBucketInput{
        Bucket: aws.String(bucket),
    })

    fmt.Println("结果: \n", awsutil.StringValue(resp))
}
```

Python SDK快速入门

本文介绍如何快速使用Python SDK完成常见操作，如创建存储空间（Bucket）、上传文件（Object）、下载文件等。

前提条件

已安装Python SDK。详情请参见[安装](#)。

创建存储空间

Bucket是存放Object的容器，所有的Object都必须存放在特定的Bucket中。

以下代码用于创建一个Bucket

```
from ks3.connection import Connection
```

```
# 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
# 强烈建议您创建并使用子账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
# 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。
c = Connection('<YOUR_ACCESS_KEY>', '<YOUR_SECRET_KEY>', host='YOUR_REGION_ENDPOINT')

# 这里如果出现409 conflict错误，说明请求的bucket name有冲突，因为bucket name是全局唯一的。
# 默认创建私有访问权限的存储空间。
b = c.create_bucket('<YOUR_BUCKET_NAME>')
```

上传文件

以下代码用于将指定目录下某一个文件上传到KS3。

```
from ks3.connection import Connection

# 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
# 强烈建议您创建并使用子账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
# 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。
c = Connection('<YOUR_ACCESS_KEY>', '<YOUR_SECRET_KEY>', host='YOUR_REGION_ENDPOINT')

# 获取存储空间实例。
b = c.get_bucket('<YOUR_BUCKET_NAME>')

# 将指定目录下某一个文件上传，同时可以指定文件 ACL
# 填写Object完整路径。Object完整路径中不能包含Bucket名称。
k = b.new_key('<yourKeyName>')

# x-kss-storage-class有效值为"STANDARD"、"STANDARD_IA"。"STANDARD"表示标准存储，"STANDARD_IA"表示低频存储，如果不指定，默认为标准存储。
headers = {"x-kss-storage-class": "STANDARD_IA"}

# 填写本地文件的完整路径。
# object policy：'private' or 'public-read'
ret = k.set_contents_from_filename("<yourSourceFilePath>", policy="private", headers=headers)

# 请求ID。请求ID是本次请求的唯一标识，强烈建议在程序日志中添加此参数。
print(ret.headers['x-kss-request-id'])

# ETag是put_object方法返回值特有的属性，用于标识一个Object的内容。
print(eval(ret.headers['ETag']))

# HTTP返回码。
if ret and ret.status == 200:
    print("上传成功")
```

下载文件

以下代码用于流式下载文件。

```
from ks3.connection import Connection

# 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
# 强烈建议您创建并使用子账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
# 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。host(Endpoint) 以北京为例，其它 Region 请按实际情况填写。
conn = Connection('<yourAccessKeyId>', '<yourAccessKeySecret>', host='ks3-cn-beijing.ksyuncs.com')

# 获取存储空间实例
b = conn.get_bucket('<yourBucketName>')

# 下载 Object，并且保存到文件中
# 填写Object完整路径。Object完整路径中不能包含Bucket名称。
k = b.get_key('<yourKeyName>')
# 按照字节大小读取，比如300
bytes = k.read(300)
```

删除文件

以下代码用于删除文件。

```
from ks3.connection import Connection

# 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
# 强烈建议您创建并使用子账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
# 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。host(Endpoint) 以北京为例，其它 Region 请按实际情况填写。
conn = Connection('<yourAccessKeyId>', '<yourAccessKeySecret>', host='ks3-cn-beijing.ksyuncs.com')

# 获取存储空间实例
b = conn.get_bucket('<yourBucketName>')

# 删除文件。<yourObjectName>表示删除KS3文件时需要指定包含文件后缀在内的完整路径。如 images/test.jpg
b.delete_key('<yourKeyName>')
```

删除存储空间

以下代码用于删除存储空间。

```
from ks3.connection import Connection

# 金山云主账号 AccessKey 拥有所有API的访问权限，风险很高。
# 强烈建议您创建并使用子账号进行 API 访问或日常运维，请登录 https://uc.console.ksyun.com/pro/iam/#/user/list 创建子账号。
# 通过指定 host(Endpoint)，您可以在指定的地域创建新的存储空间。
c = Connection('<YOUR_ACCESS_KEY>', '<YOUR_SECRET_KEY>', host='YOUR_REGION_ENDPOINT')

# 删除指定存储空间。
c.delete_bucket('<YOUR_BUCKET_NAME>')
```