contabo对象存储使用

德国制造2003 年成立（20年了）的contabo，号称高性价比的云服务。contabo产品比较单一，没有ali / aws云那么花里胡哨，只提供服务器和对象存储。这里研究对象存储。

contabo对象存储

特点

- **S3 兼容 API**
- **始终在线的 DDos 防护**
- **保证三重冗余高可用**
- **适用于任何服务器**
- **完全控制数据位置**
- **链接分享**
- **自动缩放限制**
- **方便管理的Web界面**
- **访问控制列表管理**

与其他云性价对比

计费很简单，所有按月按全包费率计费，迁移到其他供应商无限制的免费传输。

以250G容量/月为最小计算单位，亚洲 $3.39 250G/月，其他地区 $2.99 250G/月。更大的空间，基于此单位计算则可，1T则是$13.56 月，10T是$135.6 月，单个账号最大容量25T。

以10T/月为例，罗列表格

收费科目\ 云平台	**平台**	**费用解释**	**月费用$**	**半年费用$**	**年费用$**
存储费用	aliyun	数据占用的存储空间大小计费	单价：173 存储包：143.36	单价：1038 存储包：713.93	单价：2076 存储包：1290.24
	aws	-	250	1500	3000
	contabo	-	135.6	813.6	1627.2
流量费用（10T)	aliyun	通过OSS进行数据传输产生的费用，包含下行流量费用，传输加速费用上行流量和通过内网访问的流量免费阶梯累计计费	单价：1070 流量包：942.08	单价：6420 流量包：4691.56	单价：12840 流量包：8478.72
	aws		1140	6840	13680
	contabo		全包	全包	全包
接口调用费用(1000万次)	aliyun	根据调用OSS API的次数计费，所有对 OSS的操作最终都是对OSS API的调用 (包括OSS控制台、工具、阿里云其他云产品调用OSS)	3.11037	18.66222	37.32444
	aws		47+3.7=50.7	304.2	608.4
	contabo		全包	全包	全包
数据增值服务	aliyun	图片处理 传输加速 跨区域复制 打标签 DDos防护 依使用情况，单独计量计费，不使用不计费。	-	-	-
	aws	复制 打标签/标记 Lambada查询转换	-	-	-
	contabo	DDos防护 3重备份 界面管理操作	全包	全包	全包

收费科目 \ 云平台	平台	费用解释	月费用$	半年费用$	年费用$
总计费用	aliyun		1088.55037	5424.15222	9806.28444
	aws		1440.7	8644.2	17288.4
	contabo		135.6	813.6	1627.2

定价参考资料：

阿里云定价：https://www.alibabacloud.com/zh/product/oss/pricing?spm=a3c0i.15008052.8239932550..4c053aeceNsTVO

aws定价：https://aws.amazon.com/cn/s3/pricing/?nc1=h_ls

contabo定价：https://contabo.com/en/object-storage/

可行性可用性

虽然contabo价格便宜，但是要注意几个限制，这几个限制是为了确保对象存储系统的所有用户获得良好且公平的性能，可以根据当前使用情况随时调整限制。列出几个对我们可能有影响的限制：

- ★★★ S3 API 请求数默认限制为 250 个请求/秒

  会影响到请求速度，C端走CDN缓存可能是个办法
- ★★★ 带宽是有限的并且定期调整。默认限制为 10 MByte/s (=80 Mbit/s)

  会影响到请求速度，可能需要联系调整一下。
- 默认情况下每个客户最大100个桶

  影响不是很大，但是需要了解。
- 公开共享的对象设置为私有，由于缓存，它们仍然可以公开使用长达一小时

  影响不是很大，但是需要了解，例如真想这样调整，那就需要提前一小时操作。

其他方面的限制影响不大。

迁移历史数据

contabo支持一系列的工具来执行一系列操作，其中非常推荐使用rclone工具。这使用rclone来完成数据从其他平台迁移到contabo的作业。

从aliyun迁移到contabo

添加Aliyun配置

通过运行启动 rclone 配置 `rclone config`。然后 `n` 选择 `New remote`.

1. 选择远程存储的名称，例如 `aliYunOss`
2. 选择您要添加的存储类型，对于 Amazon S3，这是 `Amazon S3 Compliant Storage Providers including AWS, ...`
3. 选择您的 S3 提供商： `Alibaba`

4. `Enter AWS credentials in the next step` 通过选择并输入您的 `access_key_id` 和 `secret_access_key` 在接下来的提示中输入您的凭据
5. 选择地区
6. 指定 S3 API 的端点。跳过 (`return`) 此步骤以使用上一步中指定的区域的默认 AWS 终端节点
7. 您可以跳过 (`return`) 以下步骤 `acl` ， `server_side_encryption` 因为 `sse_kms_key_id` 您不想在 Amazon S3 存储上创建文件
8. 选择您的对象存储类别。如果您有默认 S3 存储，请选择 `Default` （ `Glacier Storage class` 如果您有 S3 Glacier 存储，请选择)
9. 如果系统提示您编辑高级配置，请 `n` 选择 `No`
10. 验证信息并确认 `y`

```
# 取名
No remotes found, make a new one?
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> aliYunOss

# 配置存储类型
Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
[snip]
 4 / Amazon S3 Compliant Storage Providers including AWS, Alibaba, Ceph, ChinaMobile, ArvanCloud, DigitalOcean, Dreamhost, Huawei OBS, IBM COS, Liara, Minio, and Tencent COS
   \ "s3"
[snip]
Storage> s3

# oss配置的云服务提供者
Choose your S3 provider.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
  1 / Amazon Web Services (AWS) S3
   \ (AWS)
 2 / Alibaba Cloud Object Storage System (OSS) formerly Aliyun
   \ (Alibaba)
 3 / Ceph Object Storage
   \ (Ceph)
 4 / China Mobile Ecloud Elastic Object Storage (EOS)
   \ (ChinaMobile)
 5 / Cloudflare R2 Storage
   \ (Cloudflare)
 6 / Arvan Cloud Object Storage (AOS)
   \ (ArvanCloud)
 7 / DigitalOcean Spaces
   \ (DigitalOcean)
 8 / Dreamhost DreamObjects
   \ (Dreamhost)
 9 / Huawei Object Storage Service
   \ (HuaweiOBS)
10 / IBM COS S3
```

```
   \ (IBMCOS)
11 / IDrive e2
   \ (IDrive)
12 / IONOS Cloud
   \ (IONOS)
13 / Seagate Lyve Cloud
   \ (LyveCloud)
14 / Liara Object Storage
   \ (Liara)
15 / Minio Object Storage
   \ (Minio)
16 / Netease Object Storage (NOS)
   \ (Netease)
17 / RackCorp Object Storage
   \ (RackCorp)
18 / Scaleway Object Storage
   \ (Scaleway)
19 / SeaweedFS S3
   \ (SeaweedFS)
20 / StackPath Object Storage
   \ (StackPath)
21 / Storj (S3 Compatible Gateway)
   \ (Storj)
22 / Tencent Cloud Object Storage (COS)
   \ (TencentCOS)
23 / Wasabi Object Storage
   \ (Wasabi)
24 / Qiniu Object Storage (Kodo)
   \ (Qiniu)
25 / Any other S3 compatible provider
   \ (Other)
[snip]
provider> Alibaba

# 环境变量设置，只有在access_key_id和secret_access_key为空有用，我们直接在下一步输入
Get AWS credentials from runtime (environment variables or EC2/ECS meta data if
no env vars).
Only applies if access_key_id and secret_access_key is blank.
Enter a boolean value (true or false). Press Enter for the default ("false").
Choose a number from below, or type in your own value
 1 / Enter AWS credentials in the next step
   \ "false"
 2 / Get AWS credentials from the environment (env vars or IAM)
   \ "true"
env_auth> 1

# access_key_id
AWS Access Key ID.
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
access_key_id> accesskeyid

# secret_access_key
AWS Secret Access Key (password)
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
secret_access_key> secretaccesskey
```

```
# 选择正确的endpoint
Endpoint for OSS API.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
Option endpoint.
Endpoint for OSS API.
Choose a number from below, or type in your own string value.
Press Enter for the default (oss-accelerate.aliyuncs.com).
 1 / Global Accelerate
   \ (oss-accelerate.aliyuncs.com)
 2 / Global Accelerate (outside mainland China)
   \ (oss-accelerate-overseas.aliyuncs.com)
 3 / East China 1 (Hangzhou)
   \ (oss-cn-hangzhou.aliyuncs.com)
 4 / East China 2 (Shanghai)
   \ (oss-cn-shanghai.aliyuncs.com)
 5 / North China 1 (Qingdao)
   \ (oss-cn-qingdao.aliyuncs.com)
 6 / North China 2 (Beijing)
   \ (oss-cn-beijing.aliyuncs.com)
 7 / North China 3 (Zhangjiakou)
   \ (oss-cn-zhangjiakou.aliyuncs.com)
 8 / North China 5 (Hohhot)
   \ (oss-cn-huhehaote.aliyuncs.com)
 9 / North China 6 (Ulanqab)
   \ (oss-cn-wulanchabu.aliyuncs.com)
10 / South China 1 (Shenzhen)
   \ (oss-cn-shenzhen.aliyuncs.com)
11 / South China 2 (Heyuan)
   \ (oss-cn-heyuan.aliyuncs.com)
12 / South China 3 (Guangzhou)
   \ (oss-cn-guangzhou.aliyuncs.com)
13 / West China 1 (Chengdu)
   \ (oss-cn-chengdu.aliyuncs.com)
14 / Hong Kong (Hong Kong)
   \ (oss-cn-hongkong.aliyuncs.com)
15 / US West 1 (Silicon Valley)
   \ (oss-us-west-1.aliyuncs.com)
16 / US East 1 (Virginia)
   \ (oss-us-east-1.aliyuncs.com)
17 / Southeast Asia Southeast 1 (Singapore)
   \ (oss-ap-southeast-1.aliyuncs.com)
18 / Asia Pacific Southeast 2 (Sydney)
   \ (oss-ap-southeast-2.aliyuncs.com)
19 / Southeast Asia Southeast 3 (Kuala Lumpur)
   \ (oss-ap-southeast-3.aliyuncs.com)
20 / Asia Pacific Southeast 5 (Jakarta)
   \ (oss-ap-southeast-5.aliyuncs.com)
21 / Asia Pacific Northeast 1 (Japan)
   \ (oss-ap-northeast-1.aliyuncs.com)
22 / Asia Pacific South 1 (Mumbai)
   \ (oss-ap-south-1.aliyuncs.com)
23 / Central Europe 1 (Frankfurt)
   \ (oss-eu-central-1.aliyuncs.com)
24 / West Europe (London)
   \ (oss-eu-west-1.aliyuncs.com)
25 / Middle East 1 (Dubai)
   \ (oss-me-east-1.aliyuncs.com)
```

```
endpoint> 1

# 访问OSS文件的权限使用默认的
Canned ACL used when creating buckets and storing or copying objects.
Note that this ACL is applied when server-side copying objects as S3
doesn't copy the ACL from the source but rather writes a fresh one.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
 1 / Owner gets FULL_CONTROL. No one else has access rights (default).
   \ "private"
 2 / Owner gets FULL_CONTROL. The AllUsers group gets READ access.
   \ "public-read"
   / Owner gets FULL_CONTROL. The AllUsers group gets READ and WRITE access.
[snip]
acl> 1

# 选默认即可
The storage class to use when storing new objects in OSS.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
 1 / Default
   \ ()
 2 / Standard storage class
   \ (STANDARD)
 3 / Reduced redundancy storage class
   \ (REDUCED_REDUNDANCY)
 4 / Standard Infrequent Access storage class
   \ (STANDARD_IA)
 5 / One Zone Infrequent Access storage class
   \ (ONEZONE_IA)
 6 / Glacier storage class
   \ (GLACIER)
 7 / Glacier Deep Archive storage class
   \ (DEEP_ARCHIVE)
 8 / Intelligent-Tiering storage class
   \ (INTELLIGENT_TIERING)
 9 / Glacier Instant Retrieval storage class
   \ (GLACIER_IR)
storage_class> 1

# 不开启高级选项
Edit advanced config? (y/n)
y) Yes
n) No
y/n> n

# 最终配置结果如下
Remote config
Configuration complete.
Options:
- type: s3
- provider: Alibaba
- access_key_id: xxxxxxx你的access_key_id
- secret_access_key: yyyyyy你的secret_access_key
- endpoint: oss-accelerate.aliyuncs.com
- acl: private
- bucket_acl: private
Keep this "aliYunOss" remote?
```

```
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d>
```

添加contabo配置

启动 rclone 配置 (`rclone config`) 并创建一个新的远程（选择 `n` 新的）。

1. 选择远程存储的名称，例如 `contabo-storage`
2. 因为对象存储可以与S3兼容，所以选择 `Amazon S3 Compliant Storage Providers including AWS, ...`
3. 选择 `Ceph Object Storage` 作为您的提供商
4. `Enter AWS credentials in the next step` 通过选择并在接下来的提示中输入您的 `access_key_id` 和来输入您的凭据。 `secret_access_key` 如果您没有这些密钥，请首先在 Contabo 客户面板中生成它们。
5. 按跳过区域选择 `return`
6. 指定您可以在对象存储概述中找到的 S3 API 端点（例如 `https://eu2.contabostorage.com` ）
7. 您可以跳过 (`return`) 以下步骤 `acl` 、 `server_side_encryption` 和 `sse_kms_key_id`
8. 如果系统提示您编辑高级配置，请 `n` 选择 `No`
9. 验证信息并确认 `y`

```
Microsoft Windows [Version 10.0.20348.1726]
(c) Microsoft Corporation. All rights reserved.

D:\kit\rclone-v1.62.2>rclone config
Current remotes:

Name                 Type
====                 ====
aliYunOss            s3
awsS3                s3

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> n

Enter name for new remote.
name> contabo-storage

Option Storage.
Type of storage to configure.
Choose a number from below, or type in your own value.
 1 / 1Fichier
   \ (fichier)
 2 / Akamai NetStorage
   \ (netstorage)
 3 / Alias for an existing remote
   \ (alias)
 4 / Amazon Drive
   \ (amazon cloud drive)
 5 / Amazon S3 Compliant Storage Providers including AWS, Alibaba, Ceph, China
Mobile, Cloudflare, ArvanCloud, DigitalOcean, Dreamhost, Huawei OBS, IBM COS,
IDrive e2, IONOS Cloud, Liara, Lyve Cloud, Minio, Netease, RackCorp, Scaleway,
SeaweedFS, StackPath, Storj, Tencent COS, Qiniu and Wasabi
   \ (s3)
 6 / Backblaze B2
   \ (b2)
 7 / Better checksums for other remotes
   \ (hasher)
 8 / Box
   \ (box)
 9 / Cache a remote
   \ (cache)
10 / Citrix Sharefile
   \ (sharefile)
11 / Combine several remotes into one
   \ (combine)
12 / Compress a remote
   \ (compress)
13 / Dropbox
   \ (dropbox)
14 / Encrypt/Decrypt a remote
   \ (crypt)
15 / Enterprise File Fabric
   \ (filefabric)
16 / FTP
   \ (ftp)
17 / Google Cloud Storage (this is not Google Drive)
   \ (google cloud storage)
18 / Google Drive
   \ (drive)
19 / Google Photos
   \ (google photos)
20 / HTTP
   \ (http)
21 / Hadoop distributed file system
   \ (hdfs)
22 / HiDrive
   \ (hidrive)
23 / In memory object storage system.
   \ (memory)
24 / Internet Archive
   \ (internetarchive)
25 / Jottacloud
   \ (jottacloud)
26 / Koofr, Digi Storage and other Koofr-compatible storage providers
   \ (koofr)
27 / Local Disk
   \ (local)
28 / Mail.ru Cloud
   \ (mailru)
29 / Mega
   \ (mega)
30 / Microsoft Azure Blob Storage
```

```
   \ (azureblob)
31 / Microsoft OneDrive
   \ (onedrive)
32 / OpenDrive
   \ (opendrive)
33 / OpenStack Swift (Rackspace Cloud Files, Memset Memstore, OVH)
   \ (swift)
34 / Oracle Cloud Infrastructure Object Storage
   \ (oracleobjectstorage)
35 / Pcloud
   \ (pcloud)
36 / Put.io
   \ (putio)
37 / QingCloud Object Storage
   \ (qingstor)
38 / SMB / CIFS
   \ (smb)
39 / SSH/SFTP
   \ (sftp)
40 / Sia Decentralized Cloud
   \ (sia)
41 / Storj Decentralized Cloud Storage
   \ (storj)
42 / Sugarsync
   \ (sugarsync)
43 / Transparently chunk/split large files
   \ (chunker)
44 / Union merges the contents of several upstream fs
   \ (union)
45 / Uptobox
   \ (uptobox)
46 / WebDAV
   \ (webdav)
47 / Yandex Disk
   \ (yandex)
48 / Zoho
   \ (zoho)
49 / premiumize.me
   \ (premiumizeme)
50 / seafile
   \ (seafile)
Storage> s3

Option provider.
Choose your S3 provider.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / Amazon Web Services (AWS) S3
   \ (AWS)
 2 / Alibaba Cloud Object Storage System (OSS) formerly Aliyun
   \ (Alibaba)
 3 / Ceph Object Storage
   \ (Ceph)
 4 / China Mobile Ecloud Elastic Object Storage (EOS)
   \ (ChinaMobile)
 5 / Cloudflare R2 Storage
   \ (Cloudflare)
 6 / Arvan Cloud Object Storage (AOS)
```

```
   \ (ArvanCloud)
 7 / DigitalOcean Spaces
   \ (DigitalOcean)
 8 / Dreamhost DreamObjects
   \ (Dreamhost)
 9 / Huawei Object Storage Service
   \ (HuaweiOBS)
10 / IBM COS S3
   \ (IBMCOS)
11 / IDrive e2
   \ (IDrive)
12 / IONOS Cloud
   \ (IONOS)
13 / Seagate Lyve Cloud
   \ (LyveCloud)
14 / Liara Object Storage
   \ (Liara)
15 / Minio Object Storage
   \ (Minio)
16 / Netease Object Storage (NOS)
   \ (Netease)
17 / RackCorp Object Storage
   \ (RackCorp)
18 / Scaleway Object Storage
   \ (Scaleway)
19 / SeaweedFS S3
   \ (SeaweedFS)
20 / StackPath Object Storage
   \ (StackPath)
21 / Storj (S3 Compatible Gateway)
   \ (Storj)
22 / Tencent Cloud Object Storage (COS)
   \ (TencentCOS)
23 / Wasabi Object Storage
   \ (Wasabi)
24 / Qiniu Object Storage (Kodo)
   \ (Qiniu)
25 / Any other S3 compatible provider
   \ (Other)
provider> 3

Option env_auth.
Get AWS credentials from runtime (environment variables or EC2/ECS meta data if
no env vars).
Only applies if access_key_id and secret_access_key is blank.
Choose a number from below, or type in your own boolean value (true or false).
Press Enter for the default (false).
 1 / Enter AWS credentials in the next step.
   \ (false)
 2 / Get AWS credentials from the environment (env vars or IAM).
   \ (true)
env_auth> 1

Option access_key_id.
AWS Access Key ID.
Leave blank for anonymous access or runtime credentials.
Enter a value. Press Enter to leave empty.
access_key_id> d05d88307f38726058d5ffd7d33f8146
```

```
Option secret_access_key.
AWS Secret Access Key (password).
Leave blank for anonymous access or runtime credentials.
Enter a value. Press Enter to leave empty.
secret_access_key> ca798fbc839926afde01386d66847791

Option region.
Region to connect to.
Leave blank if you are using an S3 clone and you don't have a region.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
   / Use this if unsure.
 1 | Will use v4 signatures and an empty region.
   \ ()
   / Use this only if v4 signatures don't work.
 2 | E.g. pre Jewel/v10 CEPH.
   \ (other-v2-signature)
region>

Option endpoint.
Endpoint for S3 API.
Required when using an S3 clone.
Enter a value. Press Enter to leave empty.
endpoint> https://sin1.contabostorage.com

Option location_constraint.
Location constraint - must be set to match the Region.
Leave blank if not sure. Used when creating buckets only.
Enter a value. Press Enter to leave empty.
location_constraint>

Option acl.
Canned ACL used when creating buckets and storing or copying objects.
This ACL is used for creating objects and if bucket_acl isn't set, for creating
buckets too.
For more info visit https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-
overview.html#canned-acl
Note that this ACL is applied when server-side copying objects as S3
doesn't copy the ACL from the source but rather writes a fresh one.
If the acl is an empty string then no X-Amz-Acl: header is added and
the default (private) will be used.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
   / Owner gets FULL_CONTROL.
 1 | No one else has access rights (default).
   \ (private)
   / Owner gets FULL_CONTROL.
 2 | The AllUsers group gets READ access.
   \ (public-read)
   / Owner gets FULL_CONTROL.
 3 | The AllUsers group gets READ and WRITE access.
   | Granting this on a bucket is generally not recommended.
   \ (public-read-write)
   / Owner gets FULL_CONTROL.
 4 | The AuthenticatedUsers group gets READ access.
   \ (authenticated-read)
   / Object owner gets FULL_CONTROL.
 5 | Bucket owner gets READ access.
   | If you specify this canned ACL when creating a bucket, Amazon S3 ignores
it.
   \ (bucket-owner-read)
   / Both the object owner and the bucket owner get FULL_CONTROL over the
object.
 6 | If you specify this canned ACL when creating a bucket, Amazon S3 ignores
it.
   \ (bucket-owner-full-control)
acl> 1

Option server_side_encryption.
The server-side encryption algorithm used when storing this object in S3.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / None
   \ ()
 2 / AES256
   \ (AES256)
 3 / aws:kms
   \ (aws:kms)
server_side_encryption> 1

Option sse_kms_key_id.
If using KMS ID you must provide the ARN of Key.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / None
   \ ()
 2 / arn:aws:kms:*
   \ (arn:aws:kms:us-east-1:*)
sse_kms_key_id> 1

Edit advanced config?
y) Yes
n) No (default)
y/n>

Configuration complete.
Options:
- type: s3
- provider: Ceph
- access_key_id: youAccessKey
- secret_access_key: youSecretAccessKey
- endpoint: https://sin1.contabostorage.com
- acl: private
Keep this "contabo-storage" remote?
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d>
Current remotes:

Name                 Type
====                 ====
aliYunOss            s3
awsS3                s3
contabo-storage      s3
```

```
e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q>
```

执行命令

```
# rclone copy [source] [destination]
rclone copy aliYunOss:bucket-name contabo-storage:bucket-name --s3-no-head
```

从AWS迁移到contabo

https://docs.contabo.com/docs/products/Object-Storage/HowTo/migrate-to-contabo-object-storage

添加AWS配置

通过运行启动 rclone 配置`rclone config`。然后`n`选择`New remote`.

1. 选择远程存储的名称，例如`aws-s3`
2. 选择您要添加的存储类型，对于 Amazon S3，这是`Amazon S3 Compliant Storage Providers including AWS, ...`
3. 选择您的 S3 提供商：`Amazon Web Services (AWS) S3`
4. `Enter AWS credentials in the next step`通过选择并输入您的`access_key_id`和`secret_access_key`在接下来的提示中输入您的凭据
5. 选择地区
6. 指定 S3 API 的端点。跳过 (`return`) 此步骤以使用上一步中指定的区域的默认 AWS 终端节点
7. 您可以跳过 (`return`) 以下步骤`acl`，`server_side_encryption`因为`sse_kms_key_id`您不想在 Amazon S3 存储上创建文件
8. 选择您的对象存储类别。如果您有默认 S3 存储，请选择`Default`（`Glacier Storage class`如果您有 S3 Glacier 存储，请选择)
9. 如果系统提示您编辑高级配置，请`n`选择`No`
10. 验证信息并确认`y`

```
e/n/d/r/c/s/q> n

Enter name for new remote.
name> aws-s3

Option Storage.
Type of storage to configure.
Choose a number from below, or type in your own value.
 1 / 1Fichier
   \ (fichier)
 2 / Akamai NetStorage
   \ (netstorage)
 3 / Alias for an existing remote
   \ (alias)
 4 / Amazon Drive
   \ (amazon cloud drive)
 5 / Amazon S3 Compliant Storage Providers including AWS, Alibaba, Ceph, China Mobile, Cloudflare, ArvanCloud, DigitalOcean, Dreamhost, Huawei OBS, IBM COS, IDrive e2, IONOS Cloud, Liara, Lyve Cloud, Minio, Netease, RackCorp, Scaleway, SeaweedFS, StackPath, Storj, Tencent COS, Qiniu and Wasabi
   \ (s3)
 6 / Backblaze B2
   \ (b2)
 7 / Better checksums for other remotes
   \ (hasher)
 8 / Box
   \ (box)
 9 / Cache a remote
   \ (cache)
10 / Citrix Sharefile
   \ (sharefile)
11 / Combine several remotes into one
   \ (combine)
12 / Compress a remote
   \ (compress)
13 / Dropbox
   \ (dropbox)
14 / Encrypt/Decrypt a remote
   \ (crypt)
15 / Enterprise File Fabric
   \ (filefabric)
16 / FTP
   \ (ftp)
17 / Google Cloud Storage (this is not Google Drive)
   \ (google cloud storage)
18 / Google Drive
   \ (drive)
19 / Google Photos
   \ (google photos)
20 / HTTP
   \ (http)
21 / Hadoop distributed file system
   \ (hdfs)
22 / HiDrive
   \ (hidrive)
23 / In memory object storage system.
   \ (memory)
24 / Internet Archive
   \ (internetarchive)
25 / Jottacloud
   \ (jottacloud)
26 / Koofr, Digi Storage and other Koofr-compatible storage providers
   \ (koofr)
27 / Local Disk
   \ (local)
28 / Mail.ru Cloud
   \ (mailru)
29 / Mega
```

```
   \ (mega)
30 / Microsoft Azure Blob Storage
   \ (azureblob)
31 / Microsoft OneDrive
   \ (onedrive)
32 / OpenDrive
   \ (opendrive)
33 / OpenStack Swift (Rackspace Cloud Files, Memset Memstore, OVH)
   \ (swift)
34 / Oracle Cloud Infrastructure Object Storage
   \ (oracleobjectstorage)
35 / Pcloud
   \ (pcloud)
36 / Put.io
   \ (putio)
37 / QingCloud Object Storage
   \ (qingstor)
38 / SMB / CIFS
   \ (smb)
39 / SSH/SFTP
   \ (sftp)
40 / Sia Decentralized Cloud
   \ (sia)
41 / Storj Decentralized Cloud Storage
   \ (storj)
42 / Sugarsync
   \ (sugarsync)
43 / Transparently chunk/split large files
   \ (chunker)
44 / Union merges the contents of several upstream fs
   \ (union)
45 / Uptobox
   \ (uptobox)
46 / WebDAV
   \ (webdav)
47 / Yandex Disk
   \ (yandex)
48 / Zoho
   \ (zoho)
49 / premiumize.me
   \ (premiumizeme)
50 / seafile
   \ (seafile)
Storage> s3

Option provider.
Choose your S3 provider.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / Amazon Web Services (AWS) S3
   \ (AWS)
 2 / Alibaba Cloud Object Storage System (OSS) formerly Aliyun
   \ (Alibaba)
 3 / Ceph Object Storage
   \ (Ceph)
 4 / China Mobile Ecloud Elastic Object Storage (EOS)
   \ (ChinaMobile)
 5 / Cloudflare R2 Storage
```

```
   \ (Cloudflare)
 6 / Arvan Cloud Object Storage (AOS)
   \ (ArvanCloud)
 7 / DigitalOcean Spaces
   \ (DigitalOcean)
 8 / Dreamhost DreamObjects
   \ (Dreamhost)
 9 / Huawei Object Storage Service
   \ (HuaweiOBS)
10 / IBM COS S3
   \ (IBMCOS)
11 / IDrive e2
   \ (IDrive)
12 / IONOS Cloud
   \ (IONOS)
13 / Seagate Lyve Cloud
   \ (LyveCloud)
14 / Liara Object Storage
   \ (Liara)
15 / Minio Object Storage
   \ (Minio)
16 / Netease Object Storage (NOS)
   \ (Netease)
17 / RackCorp Object Storage
   \ (RackCorp)
18 / Scaleway Object Storage
   \ (Scaleway)
19 / SeaweedFS S3
   \ (SeaweedFS)
20 / StackPath Object Storage
   \ (StackPath)
21 / Storj (S3 Compatible Gateway)
   \ (Storj)
22 / Tencent Cloud Object Storage (COS)
   \ (TencentCOS)
23 / Wasabi Object Storage
   \ (Wasabi)
24 / Qiniu Object Storage (Kodo)
   \ (Qiniu)
25 / Any other S3 compatible provider
   \ (Other)
provider> 1

Option env_auth.
Get AWS credentials from runtime (environment variables or EC2/ECS meta data if
no env vars).
Only applies if access_key_id and secret_access_key is blank.
Choose a number from below, or type in your own boolean value (true or false).
Press Enter for the default (false).
 1 / Enter AWS credentials in the next step.
   \ (false)
 2 / Get AWS credentials from the environment (env vars or IAM).
   \ (true)
env_auth> 1

Option access_key_id.
AWS Access Key ID.
Leave blank for anonymous access or runtime credentials.
```

```
Enter a value. Press Enter to leave empty.
access_key_id> AKIAX7FROTZ7XEZW7EOM

Option secret_access_key.
AWS Secret Access Key (password).
Leave blank for anonymous access or runtime credentials.
Enter a value. Press Enter to leave empty.
secret_access_key> JjvwzOCOYAEkHNnOPcRu3HqJLOXClQYylHEoFbms

Option region.
Region to connect to.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
   / The default endpoint - a good choice if you are unsure.
 1 | US Region, Northern Virginia, or Pacific Northwest.
   | Leave location constraint empty.
   \ (us-east-1)
   / US East (Ohio) Region.
 2 | Needs location constraint us-east-2.
   \ (us-east-2)
   / US West (Northern California) Region.
 3 | Needs location constraint us-west-1.
   \ (us-west-1)
   / US West (Oregon) Region.
 4 | Needs location constraint us-west-2.
   \ (us-west-2)
   / Canada (Central) Region.
 5 | Needs location constraint ca-central-1.
   \ (ca-central-1)
   / EU (Ireland) Region.
 6 | Needs location constraint EU or eu-west-1.
   \ (eu-west-1)
   / EU (London) Region.
 7 | Needs location constraint eu-west-2.
   \ (eu-west-2)
   / EU (Paris) Region.
 8 | Needs location constraint eu-west-3.
   \ (eu-west-3)
   / EU (Stockholm) Region.
 9 | Needs location constraint eu-north-1.
   \ (eu-north-1)
   / EU (Milan) Region.
10 | Needs location constraint eu-south-1.
   \ (eu-south-1)
   / EU (Frankfurt) Region.
11 | Needs location constraint eu-central-1.
   \ (eu-central-1)
   / Asia Pacific (Singapore) Region.
12 | Needs location constraint ap-southeast-1.
   \ (ap-southeast-1)
   / Asia Pacific (Sydney) Region.
13 | Needs location constraint ap-southeast-2.
   \ (ap-southeast-2)
   / Asia Pacific (Tokyo) Region.
14 | Needs location constraint ap-northeast-1.
   \ (ap-northeast-1)
   / Asia Pacific (Seoul).
15 | Needs location constraint ap-northeast-2.
```

```
   \ (ap-northeast-2)
   / Asia Pacific (Osaka-Local).
16 | Needs location constraint ap-northeast-3.
   \ (ap-northeast-3)
   / Asia Pacific (Mumbai).
17 | Needs location constraint ap-south-1.
   \ (ap-south-1)
   / Asia Pacific (Hong Kong) Region.
18 | Needs location constraint ap-east-1.
   \ (ap-east-1)
   / South America (Sao Paulo) Region.
19 | Needs location constraint sa-east-1.
   \ (sa-east-1)
   / Middle East (Bahrain) Region.
20 | Needs location constraint me-south-1.
   \ (me-south-1)
   / Africa (Cape Town) Region.
21 | Needs location constraint af-south-1.
   \ (af-south-1)
   / China (Beijing) Region.
22 | Needs location constraint cn-north-1.
   \ (cn-north-1)
   / China (Ningxia) Region.
23 | Needs location constraint cn-northwest-1.
   \ (cn-northwest-1)
   / AWS GovCloud (US-East) Region.
24 | Needs location constraint us-gov-east-1.
   \ (us-gov-east-1)
   / AWS GovCloud (US) Region.
25 | Needs location constraint us-gov-west-1.
   \ (us-gov-west-1)
region>

Option endpoint.
Endpoint for S3 API.
Leave blank if using AWS to use the default endpoint for the region.
Enter a value. Press Enter to leave empty.
endpoint> s3.ap-northeast-1.amazonaws.com

Option location_constraint.
Location constraint - must be set to match the Region.
Used when creating buckets only.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / Empty for US Region, Northern Virginia, or Pacific Northwest
   \ ()
 2 / US East (Ohio) Region
   \ (us-east-2)
 3 / US West (Northern California) Region
   \ (us-west-1)
 4 / US West (Oregon) Region
   \ (us-west-2)
 5 / Canada (Central) Region
   \ (ca-central-1)
 6 / EU (Ireland) Region
   \ (eu-west-1)
 7 / EU (London) Region
   \ (eu-west-2)
 8 / EU (Paris) Region
   \ (eu-west-3)
 9 / EU (Stockholm) Region
   \ (eu-north-1)
10 / EU (Milan) Region
   \ (eu-south-1)
11 / EU Region
   \ (EU)
12 / Asia Pacific (Singapore) Region
   \ (ap-southeast-1)
13 / Asia Pacific (Sydney) Region
   \ (ap-southeast-2)
14 / Asia Pacific (Tokyo) Region
   \ (ap-northeast-1)
15 / Asia Pacific (Seoul) Region
   \ (ap-northeast-2)
16 / Asia Pacific (Osaka-Local) Region
   \ (ap-northeast-3)
17 / Asia Pacific (Mumbai) Region
   \ (ap-south-1)
18 / Asia Pacific (Hong Kong) Region
   \ (ap-east-1)
19 / South America (Sao Paulo) Region
   \ (sa-east-1)
20 / Middle East (Bahrain) Region
   \ (me-south-1)
21 / Africa (Cape Town) Region
   \ (af-south-1)
22 / China (Beijing) Region
   \ (cn-north-1)
23 / China (Ningxia) Region
   \ (cn-northwest-1)
24 / AWS GovCloud (US-East) Region
   \ (us-gov-east-1)
25 / AWS GovCloud (US) Region
   \ (us-gov-west-1)
location_constraint>

Option acl.
Canned ACL used when creating buckets and storing or copying objects.
This ACL is used for creating objects and if bucket_acl isn't set, for creating
buckets too.
For more info visit https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-
overview.html#canned-acl
Note that this ACL is applied when server-side copying objects as S3
doesn't copy the ACL from the source but rather writes a fresh one.
If the acl is an empty string then no X-Amz-Acl: header is added and
the default (private) will be used.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
   / Owner gets FULL_CONTROL.
 1 | No one else has access rights (default).
   \ (private)
   / Owner gets FULL_CONTROL.
 2 | The AllUsers group gets READ access.
   \ (public-read)
   / Owner gets FULL_CONTROL.
 3 | The AllUsers group gets READ and WRITE access.
```

```
   | Granting this on a bucket is generally not recommended.
   \ (public-read-write)
   / Owner gets FULL_CONTROL.
 4 | The AuthenticatedUsers group gets READ access.
   \ (authenticated-read)
   / Object owner gets FULL_CONTROL.
 5 | Bucket owner gets READ access.
   | If you specify this canned ACL when creating a bucket, Amazon S3 ignores
it.
   \ (bucket-owner-read)
   / Both the object owner and the bucket owner get FULL_CONTROL over the
object.
 6 | If you specify this canned ACL when creating a bucket, Amazon S3 ignores
it.
   \ (bucket-owner-full-control)
acl> 1

Option server_side_encryption.
The server-side encryption algorithm used when storing this object in S3.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / None
   \ ()
 2 / AES256
   \ (AES256)
 3 / aws:kms
   \ (aws:kms)
server_side_encryption>

Option sse_kms_key_id.
If using KMS ID you must provide the ARN of Key.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / None
   \ ()
 2 / arn:aws:kms:*
   \ (arn:aws:kms:us-east-1:*)
sse_kms_key_id> 1

Option storage_class.
The storage class to use when storing new objects in S3.
Choose a number from below, or type in your own value.
Press Enter to leave empty.
 1 / Default
   \ ()
 2 / Standard storage class
   \ (STANDARD)
 3 / Reduced redundancy storage class
   \ (REDUCED_REDUNDANCY)
 4 / Standard Infrequent Access storage class
   \ (STANDARD_IA)
 5 / One Zone Infrequent Access storage class
   \ (ONEZONE_IA)
 6 / Glacier storage class
   \ (GLACIER)
 7 / Glacier Deep Archive storage class
   \ (DEEP_ARCHIVE)
 8 / Intelligent-Tiering storage class
```

```
   \ (INTELLIGENT_TIERING)
 9 / Glacier Instant Retrieval storage class
   \ (GLACIER_IR)
storage_class> 1

Edit advanced config?
y) Yes
n) No (default)
y/n> n

Configuration complete.
Options:
- type: s3
- provider: AWS
- access_key_id: 你的access_key_id
- secret_access_key: 你的secret_access_key
- endpoint: s3.ap-northeast-1.amazonaws.com
- acl: private
Keep this "new-aws-s3" remote?
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d>

Current remotes:

Name                 Type
====                 ====
aliYunOss            s3
awsS3                s3
contabo-storage      s3
aws-s3           s3

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q>
```

添加contabo配置

启动 rclone 配置 (`rclone config`) 并创建一个新的远程（选择 `n` 新的）。

1. 选择远程存储的名称，例如 `contabo-storage`
2. 因为对象存储可以与S3兼容，所以选择 `Amazon S3 Compliant Storage Providers including AWS, ...`
3. 选择 `Ceph Object Storage` 作为您的提供商
4. `Enter AWS credentials in the next step` 通过选择并在接下来的提示中输入您的 `access_key_id` 和来输入您的凭据。 `secret_access_key` 如果您没有这些密钥，请首先在 Contabo 客户面板中生成它们。
5. 按跳过区域选择 `return`

6. 指定您可以在对象存储概述中找到的 S3 API 端点（例如 `https://eu2.contabostorage.com`）
7. 您可以跳过 (`return`) 以下步骤 `acl` 、 `server_side_encryption` 和 `sse_kms_key_id`
8. 如果系统提示您编辑高级配置，请 `n` 选择 `No`
9. 验证信息并确认 `y`

执行命令

```
# rclone copy [source] [destination]
rclone copy aws-s3:bucket-name contabo-storage:bucket-name --s3-no-head

rclone copy -P -vv --transfers 32 --include "{test}/**" awsS3:/bucket-20230527
contabo-storage:bucket-name
```

应用编程接口

Ceph

contabo云平台基于Ceph实现的对象存储服务，Ceph则是用来向云平台提供对象存储/块设备/文件系统服务的一套框架。Ceph对象存储支持两种接口：**S3 兼容** / Swift 兼容。我们这里值挂住S3兼容。

遵循AWS的S3标准

contabo借助Ceph通过与 Amazon S3 RESTful API 的大部分子集兼容的接口提供对象存储功能。S3的标准请参考：AWS的S3标准。

Java编程示例

https://docs.ceph.com/en/latest/radosgw/s3/java/

pom依赖

```
    <dependencyManagement>
        <dependencies>
            <!-- AWS SDK -->
            <dependency>
                <groupId>software.amazon.awssdk</groupId>
                <artifactId>bom</artifactId>
                <version>2.17.42</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>

    <dependencies>
        <!-- AWS SDK -->
        <dependency>
            <groupId>software.amazon.awssdk</groupId>
            <artifactId>s3</artifactId>  <!-- S3 dependency -->
            <exclusions>
                <exclusion>
                    <groupId>software.amazon.awssdk</groupId>
                    <artifactId>netty-nio-client</artifactId>
                </exclusion>
                <exclusion>
                    <groupId>software.amazon.awssdk</groupId>
                    <artifactId>apache-client</artifactId>
                </exclusion>
            </exclusions>
        </dependency>

        <dependency>
            <groupId>software.amazon.awssdk</groupId>
            <artifactId>sso</artifactId> <!-- Required for identity center authentication. -->
        </dependency>

        <dependency>
            <groupId>software.amazon.awssdk</groupId>
            <artifactId>ssooidc</artifactId> <!-- Required for identity center authentication. -->
        </dependency>

        <dependency>
            <groupId>software.amazon.awssdk</groupId>
            <artifactId>apache-client</artifactId> <!-- HTTP client specified. -->
            <exclusions>
                <exclusion>
                    <groupId>commons-logging</groupId>
                    <artifactId>commons-logging</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>
```

YML配置

```
oss:
  integration:
    account:
      # 启用的oss账号，填写为resource-map的key值，必填
      # 例如，这里配置了ali/aws两个账号信息，但是只启用了aws账号
      enabled-accounts: [contabo]
      # 账号资源map，以key-value的形式装入应用
      resource-map:
        contabo:
          platform: contabo
          # 备份账号类型，目前还不支持多个主/副账号，只能配置一个。
          type: PRIMARY
          secretId: d05d88307f38726058d5ffd7d33f8146
          secretKey: ca798fbc839926afde01386d66847791
          endpoint: https://sin1.contabostorage.com
          region: Singapore
          bucket: bucket-20230807
          domainName: https://sin1.contabostorage.com/bucket-20230807
```

如何获取访问密钥：https://docs.contabo.com/docs/products/Object-Storage/HowTo/access_secret_key

如何获得endpoint：https://docs.contabo.com/docs/products/Object-Storage/s3-connection-settings

示例代码

```
package com.maya.oss.integration.oss.client;

import cn.hutool.core.collection.CollectionUtil;
import com.maya.oss.integration.config.OssAccountResourceConfig;
import com.maya.oss.integration.oss.OssAccount;
import lombok.Data;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.util.CollectionUtils;
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.StaticCredentialsProvider;
import software.amazon.awssdk.awscore.exception.AwsErrorDetails;
import software.amazon.awssdk.awscore.exception.AwsServiceException;
import software.amazon.awssdk.awscore.internal.AwsErrorCode;
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.*;
import software.amazon.awssdk.services.s3.paginators.ListObjectsV2Iterable;

import javax.annotation.PostConstruct;
import java.io.*;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

@Data
@Slf4j
@Service("contaboOssClient")
public class ContaboS3Client implements OssClient {
    private S3Client ossClient;
    private OssAccount account;
    @Autowired
    private OssAccountResourceConfig ossAccountResourceConfig;

    @PostConstruct
    public void post(){
        OssAccount account =
ossAccountResourceConfig.getResourceMap().get("contabo");
        if(account == null){return;}
        init(account);
    }

    public void init(OssAccount account){
        this.account = account;
        if(ossClient == null) {
            ossClient = createClient();
        }

        // JVM退出时，优雅关闭资源
        Runtime.getRuntime().addShutdownHook(new Thread(()->{
            try {
                close();
            } catch (Exception e) {
                log.error(e.getMessage(), e);
            }
        }));
    }

    @Override
    public OssAccount getOssAccount() {
        return account;
    }

    /**
     * 创建AWS S3客户端
     * @return 返回客户端对象
     */
    private S3Client createClient(){
        AwsCredentialsProvider credentialsProvider =
StaticCredentialsProvider.create(
                AwsBasicCredentials.create(account.getSecretId(),
account.getSecretKey())
        );
        Region region = Region.of(account.getRegion());
        S3Client client = null;
        try {
            client = S3Client.builder()
                    .endpointOverride(new URI(account.getEndpoint()))
                    .httpClientBuilder(ApacheHttpClient.builder())
                    .region(region)
                    .serviceConfiguration(srvcConf -> {
                        srvcConf.pathStyleAccessEnabled();
                    })
                    .credentialsProvider(credentialsProvider)
                    .build();
        } catch (URISyntaxException e) {
            log.error(e.getMessage(), e);
        }
        return client;
    }

    /**
     * 处理文件上传
     */
    public boolean uploadToOSS(String key, File localFile) {
        try {
            return uploadToOSS(key, new FileInputStream(localFile), 0);
        } catch (FileNotFoundException e) {
            log.error(e.getMessage(), e);
        }
        return false;
    }

    @Override
    public boolean uploadToOSS(String key, InputStream inputStream) {
        try {
            return uploadToOSS(key, inputStream, 5);
        }finally {
            try {
                inputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    @Override
    public void copy(String sourceKey, String destinationKey) {
        CopyObjectRequest copyReq = CopyObjectRequest.builder()
                .sourceBucket(account.getBucket())
                .sourceKey(sourceKey)
                .destinationBucket(account.getBucket())
                .destinationKey(destinationKey)
                .build();
        ossClient.copyObject(copyReq);
    }

    @Override
    public InputStream downloadFile(String objectKey) {
        GetObjectRequest objectRequest = GetObjectRequest
                .builder()
                .key(objectKey)
                .bucket(account.getBucket())
                .build();

        ResponseBytes<GetObjectResponse> objectBytes =
ossClient.getObjectAsBytes(objectRequest);
        return objectBytes == null ? null : objectBytes.asInputStream();
    }

    @Override
    public List<String> page(String nextContinuationToken, String prefix) {
        ListObjectsV2Request listReq = ListObjectsV2Request.builder()
                .bucket(account.getBucket())
                .prefix(prefix)
                .continuationToken(nextContinuationToken)
                .maxKeys(MAX_KEYS)
                .build();

        ListObjectsV2Iterable listRes =
ossClient.listObjectsV2Paginator(listReq);
        List<String> result = new ArrayList<>();
        listRes.forEach(e->{
            for (S3Object content : e.contents()) {
                result.add(content.key());
            }
        });
        return result;
    }
    public List<String> list(String nextContinuationToken, String prefix) {
        ListObjectsRequest request = ListObjectsRequest.builder()
                .bucket(account.getBucket())
                .prefix(prefix)
                .maxKeys(MAX_KEYS)
                .build();
        ListObjectsResponse loResponse = ossClient.listObjects(request);
        if(loResponse != null &&
CollectionUtil.isNotEmpty(loResponse.contents())) {
            return loResponse.contents().stream().map(e-
>e.key()).collect(Collectors.toList());
        }
        return null;
    }

    @Override
    public void delete(List<String> keys) {
        if(CollectionUtils.isEmpty(keys)) {
            return;
        }

        List<ObjectIdentifier> removeKeys = keys.stream().map(
                e->ObjectIdentifier.builder()
                        .key(e).build()
        ).collect(Collectors.toList());

        Delete del = Delete.builder()
                .objects(removeKeys)
                .build();
        DeleteObjectsRequest multiObjectDeleteRequest =
                DeleteObjectsRequest.builder()
                        .bucket(account.getBucket())
                        .delete(del)
                        .build();

        DeleteObjectsResponse delObjRes =
ossClient.deleteObjects(multiObjectDeleteRequest);
        for (DeletedObject deletedObject : delObjRes.deleted()) {
            log.debug("aws云端文件已删除。{}", deletedObject.key());
        }
    }

    public boolean uploadToOSS(String key, InputStream inputStream, int
tryCount) {
        try {
            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(account.getBucket())
                    .key(key)
                    //.metadata(metadata)
                    .build();

            PutObjectResponse putObjectResponse = ossClient.putObject(putOb,
RequestBody.fromInputStream(inputStream, inputStream.available()));
            return true;
        } catch (Exception e) {
            log.error(String.format("上传%s文件到oss发生异常：", key), e);
            if(e instanceof AwsServiceException) {
                AwsServiceException awsServiceException = (AwsServiceException)
e;
                AwsErrorDetails awsErrorDetails =
awsServiceException.awsErrorDetails();
                if(awsErrorDetails != null &&
AwsErrorCode.THROTTLING_ERROR_CODES.contains(awsErrorDetails.errorCode())) {
                    if(tryCount < 5){
                        tryCount++;
                        Double pow = Math.pow(tryCount, tryCount);
                        int sleepTime = pow.intValue() + 5;
                        log.info("等待{}秒后，将进行第{}次重上传 {}", sleepTime,
tryCount, key);
                        try {
                            TimeUnit.SECONDS.sleep(sleepTime);
                        } catch (InterruptedException ex) {
                            Thread.interrupted();
                        }
                        log.info("开始第{}次重上传 {}", tryCount, key);
                        return uploadToOSS(key, inputStream, tryCount);
                    }
                }
            }
            return false;
        }
    }


    @Override
    public void close() throws Exception {
        if(ossClient != null) {
            ossClient.close();
        }
    }
}
```

单元测试

```
package com.maya.oss.integration.oss;

import com.maya.oss.integration.BaseTest;
import com.maya.oss.integration.oss.client.ContaboS3Client;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;

import java.util.List;

public class ContaboS3ClientTest extends BaseTest {
    @Autowired
    private ContaboS3Client client;


    @Test
    public void page() {
        List<String> result = client.page(null, null);
        System.out.println(result);
    }

    @Test
    public void test() {
        File file = new File("D:\\data\\Snipaste_2023-06-01_22-33-53.png");
        boolean b = client.uploadToOSS("Snipaste_2023-06-01_22-33-53.png",
file);
        System.out.println("uploadToOss "+b);
        client.copy("Snipaste_2023-06-01_22-33-53.png", "123456.png");
        List<String> pageResult = client.page(null, null);
        System.out.println(pageResult);
        client.delete(Lists.newArrayList("123456.png"));
    }
}
```